



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# Modelo de Planejamento Multiagente com Verificação, Transformação e Validação de Planos

Leonardo Henrique Moreira

Dissertação apresentada como requisito parcial  
para conclusão do Mestrado em Informática

Orientadora  
Prof.<sup>a</sup> Dr.<sup>a</sup> Célia Ghedini Ralha

Brasília  
2017

Universidade de Brasília — UnB  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Mestrado em Informática

Coordenador: Prof. Dr. Bruno Luigi Macchiavello Espinoza

Banca examinadora composta por:

Prof.<sup>a</sup> Dr.<sup>a</sup> Célia Ghedini Ralha (Orientadora) — CIC/UnB  
Prof. Dr. George Luiz Medeiros Teodoro — CIC/UnB  
Prof.<sup>a</sup> Dr.<sup>a</sup> Leliane Nunes de Barros — IME/USP

### **CIP — Catalogação Internacional na Publicação**

Moreira, Leonardo Henrique.

Modelo de Planejamento Multiagente com Verificação, Transformação e Validação de Planos / Leonardo Henrique Moreira. Brasília : UnB, 2017.

95 p. : il. ; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2017.

1. Sistemas Multiagente, 2. Planejamento Automatizado,  
3. Planejamento Multiagente, 4. Tomada de Decisão

CDU 004

Endereço: Universidade de Brasília  
Campus Universitário Darcy Ribeiro — Asa Norte  
CEP 70910-900  
Brasília-DF — Brasil



# Dedicatória

A Michele, a melhor parte da minha vida. E a nossa maior Vitória.

# Agradecimentos

Primeiramente agradeço a Deus por apontar-me a direção certa, colocar as pessoas certas ao meu lado e principalmente por estar presente em todos momentos da minha vida, desde os mais agradáveis até os mais difíceis.

Entre essas pessoas, destaco minha esposa, Michele, e minha filha Vitória que provavelmente foram as mais afetadas durante esse período de dedicação aos estudos no qual tive que estar mais ausente e portanto, menos participativo que o normal; sem o carinho, atenção e compreensão delas, essa jornada teria sido muito mais difícil.

Agradeço também aos meus pais e irmãos por serem exemplos em tudo para mim e os motivos pelos quais tento sempre ser melhor do que fui ontem.

A todos mestres com os quais tive o prazer e a honra de ser aluno nesses dois rápidos, intensos e produtivos anos. Em especial, agradeço aos professores George Luiz Medeiros Teodoro e Leliane Nunes de Barros por aceitarem fazer parte da banca, contribuindo de maneira significativa com críticas, sugestões e principalmente com o tempo deles.

Agradeço ao Cel Andersonn Kohl e ao Maj Marcelo José Camilo, meus chefes no Centro de Desenvolvimento de Sistemas do Exército Brasileiro, por me incentivarem e ajudarem durante todas as fases desse curso, desde a solicitação junto ao escalão superior até a liberação, e por não medirem esforços para gerar as melhores condições que permitiram minha dedicação exclusiva ao aprimoramento profissional.

Por fim, gostaria de agradecer a Prof<sup>a</sup> Célia Ghedini Ralha, mais um exemplo de pessoa abençoada que Deus coloca na minha vida. Obrigado por, desde a primeira reunião, tratar-me com respeito, desafiar-me e confiar no meu trabalho; essa é maneira mais eficiente de motivar uma pessoa. Obrigado por sempre estar disponível para reuniões, mesmo quando isso significava abdicar de momentos de lazer com a família e durante finais de semana. Obrigado pela parceria na escrita dos artigos; com certeza isso contribuiu muito para minha maturidade científica. Enfim, obrigado pela orientação e paciência durante esses dois anos; sem sua ajuda nada disso teria acontecido.

# Resumo

Planejamento multiagente é um tema desafiador e atual na área de pesquisa de Inteligência Artificial. A complexidade e necessidade de interação entre os agentes tem como consequência elevados valores para métricas relacionadas ao tempo de execução e volume de comunicação. Considerando esses fatores como interessantes questões de pesquisa, esse trabalho propõe um modelo de planejamento multiagente com verificação, transformação e validação de planos. O modelo proposto foi denominado LCMAP - *Lightweight Coordination Multiagent Planning*, por utilizar um modelo de coordenação eficiente reduzindo o nível de troca de mensagens entre os agentes. No LCMAP a verificação e transformação do problema são realizadas em uma fase inicial, com o intuito de facilitar o processo de planejamento diminuindo a interação entre os agentes. As capacidades dos agentes são analisadas quanto à possibilidade de atingirem os objetivos de forma total ou parcial. Para simplificar o problema original, o mesmo é transformado em um conjunto de instâncias menores através da atribuição dos objetivos aos agentes com as capacidades necessárias. Desta forma, o processo de planejamento é realizado de maneira individual, minimizando o tempo necessário e o volume de mensagens trocadas. Por fim, os planos calculados são validados, garantindo que suas execuções possam ocorrer de maneira paralela. As principais contribuições do trabalho incluem: (i) realização de testes antes da fase de planejamento para avaliar a possibilidade de encontrar uma solução; (ii) utilização de técnicas de execução paralela, sendo este aspecto pouco explorado nos trabalhos relacionados; e (iii) alocação eficiente dos recursos disponíveis através de estratégias que visam minimizar o número de agentes ou balancear a carga entre os mesmos.

**Palavras-chave:** Sistemas Multiagente, Planejamento Automatizado, Planejamento Multiagente, Tomada de Decisão

# Abstract

Multiagent planning is a challenging and modern theme in the Artificial Intelligence research area. The complexity and requirements of interactions among agents lead to high values of metrics related to execution time and communication volume. Concerning these points as interesting research questions, this work proposes a multiagent planning model with plan verification, transformation, and validation. The proposal was defined as LCMAP - Lightweight Coordination Multiagent Planning due to its use of an efficient coordination process that aims to reduce the message exchanged level among agents. In LCMAP, the problem verification and validation are carried out in an initial phase in order to ease the planning process by diminishing agent interactions. The agent's capabilities are analyzed about the possibility of reaching the goals in a total or partial way. The original problem is transformed to smaller instances set by assigning goals to agents with the required capabilities. In this sense, the planning process is performed in an individual way, minimizing the necessary time and the message exchanged volume. Finally, the plans found are validated in order to grant that their executions can occur parallel. The main contributions of these works include: (i) tests carried out before the planning phase in order to evaluate the possibility of finding a solution; (ii) use of parallel execution techniques, which are rarely exploited in related works; and (iii) efficient allocation of available resources using strategies that aim to minimize the number of agents or balance the load among them.

**Keywords:** Multiagent Systems, Automated Planning, Multiagent Planning, Decision Making

# Lista de Figuras

2.1	Relações e objetos do cenário de planejamento. . . . .	7
2.2	Agente orientado a objetivos. . . . .	16
2.3	Taxonomia de coordenação. . . . .	17
2.4	Exemplo de problema de planejamento no domínio <i>rovers</i> . . . . .	19
2.5	Exemplo de problema de planejamento do tipo <i>logistics</i> . . . . .	20
2.6	Classificação de MAP. . . . .	21
3.1	Primeiro nível da árvore de busca. . . . .	26
3.2	Taxonomia para MAP. . . . .	32
4.1	Interação entre agentes. . . . .	35
4.2	Modelo conceitual. . . . .	38
4.3	Paralelização do processo de instanciação dos agentes. . . . .	41
5.1	Comparação dos tempos obtidos a partir dos modelos - <i>Satellite</i> . . . . .	57
5.2	Comparação dos tempos obtidos a partir dos modelos - <i>Rovers</i> . . . . .	58
5.3	Comparação dos tempos obtidos a partir dos modelos - <i>Zeno-travel</i> . . . . .	59
5.4	Comparação dos tempos obtidos a partir dos modelos - <i>Depots</i> . . . . .	60
5.5	Comparação dos tempos obtidos a partir dos modelos - <i>Elevators</i> . . . . .	61
5.6	Comparação dos tempos obtidos a partir dos modelos - <i>Logistics</i> . . . . .	62
5.7	Análise de escalabilidade para <i>rovers</i> . . . . .	65
5.8	Análise de escalabilidade para <i>logistics</i> . . . . .	66
A.1	Análise dos resultados de identificação de pesquisa. . . . .	81

# Lista de Tabelas

3.1	Taxonomia para trabalhos de MAP. . . . .	31
3.2	Trabalhos classificados de acordo com a taxonomia proposta. . . . .	32
4.1	Contribuição e características do modelo proposto. . . . .	34
5.1	Resultados para <i>Satellite</i> . . . . .	57
5.2	Resultados para <i>Rovers</i> . . . . .	58
5.3	Resultados para <i>Zeno-travel</i> . . . . .	59
5.4	Resultados para <i>Depots</i> . . . . .	60
5.5	Resultados para <i>Elevators</i> . . . . .	61
5.6	Resultados para <i>Logistics</i> . . . . .	62
5.7	Resultados para <i>Rovers</i> . . . . .	64
5.8	Resultados para <i>Logistics</i> . . . . .	65
A.1	Fases e atividades de uma revisão sistemática. . . . .	78
A.2	Protocolo de pesquisa: estratégia de busca. . . . .	80
A.3	Identificação de pesquisa. . . . .	80
A.4	Conjunto de perguntas do critério de avaliação. . . . .	81
A.5	Avaliação de qualidade dos trabalhos incluídos. . . . .	83
A.6	Sumarização dos principais trabalhos. . . . .	84

# Sumário

<b>Lista de Figuras</b>	<b>viii</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos . . . . .	3
1.2 Metodologia e Organização do Documento . . . . .	4
<b>2 Fundamentação Teórica</b>	<b>5</b>
2.1 Planejamento Automatizado . . . . .	5
2.2 Linguagens e Domínios de Planejamento . . . . .	12
2.3 Sistema Multiagente . . . . .	14
2.4 Planejamento Multiagente . . . . .	18
<b>3 Trabalhos Correlatos</b>	<b>24</b>
3.1 Principais Trabalhos . . . . .	24
3.1.1 A Formal Analysis of Required Cooperation in Multi-Agent Planning	24
3.1.2 FMAP: Distributed Cooperative Multi-Agent Planning . . . . .	25
3.1.3 MADLA: Planning With Distributed and Local Search . . . . .	27
3.1.4 MAPJA: Multi-agent planning with joint actions . . . . .	27
3.1.5 Multi-Agent Planning by plan Reuse . . . . .	28
3.1.6 Transforming Multi-Agent Planning Into Single-Agent Planning Using Best-cost Strategy . . . . .	29
3.2 Taxonomia . . . . .	30
<b>4 Modelo proposto</b>	<b>33</b>
4.1 Visão Geral . . . . .	34
4.2 Modelo Arquitetural . . . . .	37
4.2.1 Camada de tratamento da entrada . . . . .	39
4.2.2 Camada de coordenação e planejamento . . . . .	45
4.2.3 Conjunto mínimo de agentes . . . . .	49
<b>5 Experimentos e Análise dos Resultados</b>	<b>53</b>
5.1 Escalabilidade . . . . .	62
5.2 Análise dos Resultados . . . . .	66
<b>6 Conclusões</b>	<b>68</b>

<b>Referências</b>	<b>71</b>
<b>A Revisão Sistemática</b>	<b>77</b>
A.1 Aplicação . . . . .	78

# Capítulo 1

## Introdução

Planejamento é o processo de deliberação associado à realização de tarefas. Ele é baseado na escolha e organização de ações através da antecipação de seus efeitos esperados. Esse processo visa satisfazer, da melhor maneira possível, os objetivos definidos. Planejamento automatizado é uma área da Inteligência Artificial que estuda esse processo de deliberação sob a ótica da computação [1].

Independente da representação utilizada para o problema de planejamento, seja baseada em predicado lógico e/ou variáveis de estado, a complexidade computacional é equivalente a  $EXPSPACE^1$ . Em ambos os casos, o tempo necessário para resolver um problema pode ser exponencial em função do tamanho da entrada. Entretanto, essa complexidade é a do pior caso, logo existem problemas que podem ser considerados menos complexos [1].

O planejamento automatizado tem como característica a presença de um único agente responsável pelo planejamento ou execução das ações, o que limita a modelagem de problemas do mundo real. O processo de planejamento realizado na presença de múltiplos agentes é chamado de planejamento multiagente.

Planejamento multiagente é um desafio no campo de Inteligência Artificial que tem ganhado uma crescente importância. Devido à complexidade computacional, a necessidade de interação entre os agentes, elevados tempos de execução e volumes de comunicação são problemas ainda considerados em aberto. Além disso, essa área de pesquisa abrange outros fatores do problema, tais como: presença de objetivos individuais ou globais, processo de coordenação, comunicação e interação entre os agentes, e escalabilidade.

Pode-se dizer que a finalidade de um modelo de planejamento multiagente, segundo de Weerd et al. [2], de Weerd and Clement [3], Traverso et al. [4], é encontrar uma sequência de ações que, quando executada, é capaz de atingir um conjunto de objetivos a partir de um estado inicial. A dificuldade em computar essa coleção de ações depende de vários fatores que incluem a quantidade de objetivos, de ações disponíveis, de agentes e do processo de coordenação. Além disso, a complexidade computacional do problema depende da forma pela qual ele é representado e pode atingir níveis exponenciais em tempo e espaço [4].

Os principais trabalhos propostos na área podem ser divididos em duas classes de abordagem. A primeira tenta transformar o problema original em várias instâncias meno-

---

<sup>1</sup>Conjunto de todos os problemas de decisão resolvíveis por uma Máquina de Turing determinística em espaço  $O(2^{p(n)})$ , onde  $p(n)$  é uma função polinomial de  $n$

res e mais simples, utilizando a estratégia de delegação e distribuição dos objetivos entre os agentes [5], [6], [7]. Entretanto, um fato importante sobre os trabalhos que utilizam a estratégia de transformação é que não é realizado nenhum tipo de validação dos planos, ou seja, se eles podem ser executados de maneira simultânea. A segunda classe apresenta soluções que exploram o espaço de busca de maneira colaborativa, usando várias entidades e continuamente refinando os planos até que uma solução seja encontrada [8, 9].

Uma característica em comum entre os trabalhos é o fato que nenhum deles verifica se as condições necessárias para a solução do problema estão disponíveis antes de tentar resolvê-lo, o que causa um desperdício de recursos quando tenta-se resolver um plano não exequível.

Essa deficiência motivou a comunidade científica a lançar, em 2016, um desafio denominado *Unsolvability International Planning Competition*<sup>2</sup> cujo objetivo é testar a habilidade de planejadores automatizados detectarem quando um planejamento de tarefas não tem solução.

O processo de coordenação utilizado também configura um desafio na área de planejamento multiagente, pois o volume de mensagens trocadas entre os agentes, a fim de controlar o progresso do trabalho, pode ser tão grande que impossibilite a escalabilidade do sistema.

Por fim, poucas abordagens exploram técnicas de execução paralela e quando o fazem não apresentam explicitamente as decisões de projeto e suas implementações. Com o uso das ferramentas existentes em programação paralela, é possível diminuir o tempo gasto durante o processo de planejamento além de melhor utilizar todos os recursos computacionais disponíveis nas plataformas que instanciam sistemas multiagentes. Considerando as lacunas referentes à verificação de condições para solução do problema e ao processo de coordenação, a questão de pesquisa dessa dissertação é: *Quais são os efeitos da execução de um pré-planejamento para um planejamento multiagente?*

Dessa forma, o problema a ser investigado nesse trabalho é a análise da possibilidade de aprimorar os modelos de planejamento multiagente. Essa melhoria está baseada na introdução de fases de verificação e transformação do problema inicial e de uma fase final de validação da independência dos planos. Outra característica importante do modelo é a capacidade de selecionar os agentes que participarão do processo de planejamento segundo estratégias que visem o balanceamento de carga ou o menor número de recursos (agentes) empregados. Essa capacidade difere dos trabalhos existentes pois esses envolvem todos os recursos disponíveis, o que pode afetar a escalabilidade.

Enfim acreditamos que as fases de verificação, transformação e validação são capazes de reduzir o tempo transcorrido durante o processo de tomada de decisão pelas seguintes razões. Primeiramente, a análise do problema e uma etapa anterior ao planejamento evita que recursos sejam empregados na busca de soluções que não são factíveis, considerando as condições vigentes. A transformação do problema em instâncias menores e independentes entre si possibilita processos de planejamento mais simples, com espaços de busca menores e, conseqüentemente, mais rápidos de serem resolvidos. Por fim, a validação dos planos calculados visa garantir que as execuções possam ocorrer de forma paralela e sem concorrência e competição entre os agentes.

Além disso, as técnicas de paralelismo, quando possível, permitem que os recursos computacionais possam ser explorados na plenitude. A união dessas abordagens produ-

---

<sup>2</sup><http://unsolve-ipc.eng.unimelb.edu.au>

zem uma economia do tempo consumido durante o processo, permitindo que a resposta seja apresentada ao usuário mais rapidamente. Concomitantemente, a seleção apenas dos agentes potencialmente capazes de resolver o problema pode diminuir o número de mensagens trocadas, favorecendo tanto a escalabilidade do modelo computacional quanto o processo de descentralização do planejamento, delegando objetivos apenas aos agentes com condições de resolvê-los.

Portanto, o desenvolvimento de um modelo de planejamento multiagente, que ofereça as capacidades de verificar se um problema possui as condições para sua solução antes de iniciar sua exploração, de selecionar eficientemente os elementos a serem envolvidos e de coordenar as atividades de suas entidades trocando um baixo número de mensagens, configura uma contribuição interessante para a área de planejamento multiagente.

Ainda, esse modelo de planejamento multiagente pode auxiliar processos decisórios de problemas de mundo real, seja de natureza civil ou militar. Ao final desse processo, o decisor terá um plano que representa seu planejamento e os preparativos para atender a um determinado acontecimento ou intenção. Além da característica essencial de apresentação de uma sequência de atividades, um plano deve também possuir outras peculiaridades, das quais três são destacáveis: (i) deve possibilitar o cumprimento da missão, atendendo a todos objetivos planejados; (ii) deve propiciar o emprego adequado dos recursos disponíveis, evitando que estes sejam utilizados em tarefas inexecutáveis; e (iii) deve ser simples, ou seja, apenas os elementos essenciais para o sucesso da missão devem ser empregados.

A partir desse contexto, uma das primeiras atividades que devem ser executadas pelo decisor é a análise dos três pontos supracitados: eficiência, viabilidade e simplicidade. Antes de iniciar o processo de apresentação das atividades que definirão o plano, é preciso saber se aquela missão é possível de ser executada. Sendo possível, uma seleção eficiente dos elementos necessários para o cumprimento das tarefas é uma condição necessária para que ocorra a descentralização do processo decisório, além de garantir um uso adequado dos recursos. Nesse contexto, o modelo proposto visa atender a esses três pontos através de fases de verificação, transformação e validação, executadas antes do planejamento propriamente dito.

## 1.1 Objetivos

O objetivo geral desse trabalho é desenvolver um modelo de planejamento multiagente que executa um pré-planejamento através de fases de verificação, transformação e validação de planos. Dessa forma, é esperado obter um uso mais eficiente dos recursos disponíveis, isto é, tempo de execução do algoritmo de planejamento e número de agentes utilizados.

Para tanto, os seguintes objetivos específicos são destacados:

1. Identificar e adaptar algoritmos e protocolos de planejamento multiagente existentes na literatura, os quais serão executados pelos componentes do modelo proposto;
2. Validar o modelo proposto a partir de *benchmarks* utilizados pela comunidade de planejamento automatizado;
3. Comparar resultados obtidos pelo modelo com aqueles oriundos dos trabalhos FMAP (Torreno et al. [8]) e MADLA (Štolba and Komenda [10]).

## 1.2 Metodologia e Organização do Documento

Para a realização desse trabalho foram identificadas fases importantes que por sua vez definiram a sequência da atividades realizadas.

Inicialmente, buscou-se revisar conceitos de planejamento automatizado, sistemas multiagentes e planejamento multiagente. Assim, no Capítulo 2 dessa dissertação são apresentados os principais conceitos relacionados ao planejamento em um ambiente multiagente, destacando as premissas assumidas no planejamento clássico e como suas técnicas podem ser aplicadas a sistemas multiagente garantindo autonomia das entidades.

Em seguida, uma revisão sistemática da literatura foi efetuada para identificar os principais trabalhos correlatos e conseqüentemente, as oportunidades de melhorias e pontos fortes. No Capítulo 3 é apresentada a metodologia de uma revisão sistemática e como ela foi aplicada no contexto dessa pesquisa. Em seguida, é destacada uma comparação dos principais trabalhos existentes que possibilitaram a proposta de uma taxonomia de classificação. Essa análise permitiu a identificação das lacunas e dos pontos fortes, orientando assim o desenvolvimento do modelo proposto.

Após a consolidação dessas duas fases, o modelo de planejamento multiagente foi definido com base nas características destacadas no estado da arte. No Capítulo 4 são apresentados os detalhes do modelo, explicitando seu modelo conceitual, o esquema de interação entre os agentes e os componentes que definem as características dos agentes.

Em seguida, a validação do modelo foi realizada e no Capítulo 5 é apresentado um conjunto de experimentos e comparações dos resultados com outros trabalhos existentes. Esses experimentos tem como finalidade prover maneiras que permitam a discussão dos desempenhos dos trabalhos referentes a tempo de planejamento, número de agentes empregados e tamanho dos planos. Além disso, a escalabilidade e o processo de coordenação do modelo proposto são analisados.

Por fim, no Capítulo 6 são apresentadas as conclusões do trabalho e algumas propostas de trabalhos futuros que podem enriquecer o conjunto de contribuições oferecidas por essa pesquisa.

# Capítulo 2

## Fundamentação Teórica

Neste capítulo são apresentados os conceitos básicos das áreas de estudo envolvidas no trabalho incluindo planejamento automatizado, sistemas multiagentes, e planejamento multiagente, sendo que todas estão relacionadas à área de Inteligência Artificial.

### 2.1 Planejamento Automatizado

Planejamento de forma geral é um processo que visa definir uma sequência de ações cujo objetivo é atingir um conjunto de estados desejáveis. As entradas para a execução desse processo, tais como estados iniciais, recursos disponíveis e restrições afetam a complexidade da computação das etapas e conseqüentemente o tempo de resposta. Em Inteligência Artificial, este tópico é chamado de Planejamento Automatizado [1].

Os algoritmos de planejamento automatizado podem ser classificados de acordo com a forma pela qual o espaço de busca é explorado. Em uma abordagem de espaço de estados (*state-space*), cada nó desse espaço representa um estado do ambiente e um plano é o caminho entre os nós. A transição entre os estados é realizada pela execução das ações, cuja ordenação é dada pelo próprio plano. A definição formal de estado será apresentada no prosseguimento do texto.

Na abordagem de espaço de planos (*plan-space*), a busca é baseada em planos parciais, isto é, conjunto de ações parcialmente ordenadas. A cada estágio de exploração, os planos são atualizados pela adição de novas ações ou por restrições, até que uma solução seja encontrada ou todo espaço tenha sido avaliado. A busca pela sequência de ações, que é capaz de transformar o estado inicial nos objetivos finais, pode ser vista como um refinamento do conjunto de possíveis sequências denominadas planos parciais [2]. Um planejador pode reduzir o número de planos possíveis neste conjunto pela adição de ações, de condições que permitam a execução de ações e de restrições para um ou mais planos parciais. A esse processo de redução é dado o nome de refinamento de plano.

De forma geral, as abordagens de planejamento executam seus algoritmos a partir de tuplas compostas por conjuntos de estados, ações e eventos, além de uma função de transição entre esses estados. A abordagem seguida nesse trabalho é a mesma de um sistema de transição de estados, cujo formalismo é apresentado em Ghallab et al. [1] e descrito na Definição 1.

**Definição 1.** *Um sistema de transição de estados é uma tupla  $\Sigma = (S, A, \gamma)$ , onde:*

- $S$ : um conjunto finito de estados que definem o ambiente;
- $A$ : um conjunto finito de ações que podem ser executadas (no prosseguimento do texto será visto que as ações são executadas por agentes);
- $\gamma : S \times A \rightarrow S$ : uma função que a partir de um estado e uma ação leva a um novo estado.

Além dos itens  $S, A, \gamma$ , outro conceito utilizado é o de custo da transição de estado para outro através da execução de uma ação. Formalmente, o custo é apresentado como uma função  $custo : S \times A \rightarrow [0, \infty)$ .

Quando a propriedade de finitude dos conjuntos é combinada com restrições adicionais, um modelo de planejamento pode ser definido, assumindo hipóteses, como por exemplo aquelas definidas em Ghallab et al. [1]:

1. O conjunto finito  $S$  de estados;
2. O conjunto de estados  $S$  são considerados *totalmente observáveis*, o que indica que é possível ter conhecimento sobre qualquer estado do ambiente;
3. O conjunto  $A$  de ações *determinísticas*, isto é, os efeitos causados pelas suas execuções são sempre as mesmas, sem qualquer incerteza;
4. O conjunto de eventos é considerado *estático* e vazio, isto é, as mudanças no ambiente são causadas estritamente pela execução de ações;
5. Os objetivos são *restritos* o que garante que os planejadores lidem apenas com estados finais explicitamente definidos, ignorando qualquer condição ou restrição adicional sobre eles;
6. A solução, isto é, o plano final, sempre é uma sequência *finita e ordenada* de ações;
7. A duração da execução das ações e da ocorrência dos eventos (apesar de ter um conjunto vazio) são *instantâneas*, ou seja, a transição de estado causada a partir deles (ações e eventos) é imediata;
8. O planejamento é realizado de forma *offline* o que implica que qualquer mudança no conjunto de estados seja ignorada durante o processo de busca pelo plano;
9. Existe apenas um agente.

Tradicionalmente uma abordagem de planejamento que siga essas nove hipóteses é dita *clássica* (*Classical Planning*).

No espaço de estados, cada elemento  $s \in S$  define propriedades dos diversos objetos que pertencem e definem o ambiente. Dessa maneira, é comum utilizar tais propriedades para descrever o cenário de planejamento. Um exemplo simples é a caracterização da navegabilidade em um ambiente através da definição de objetos referentes às posições ou coordenadas e das relações entre eles, isto é, se são adjacentes, visíveis e alcançáveis entre si. Para facilitar o entendimento a Figura 2.1 apresenta um cenário hipotético composto por quatro posições (A, B, C e D) e pelas relações entre elas. Por exemplo, uma aresta direcionada de A para B representa que A está relacionado (adjacente, visível ou alcançável) com B. Nas condições descritas e supondo que para deslocar-se de um ponto

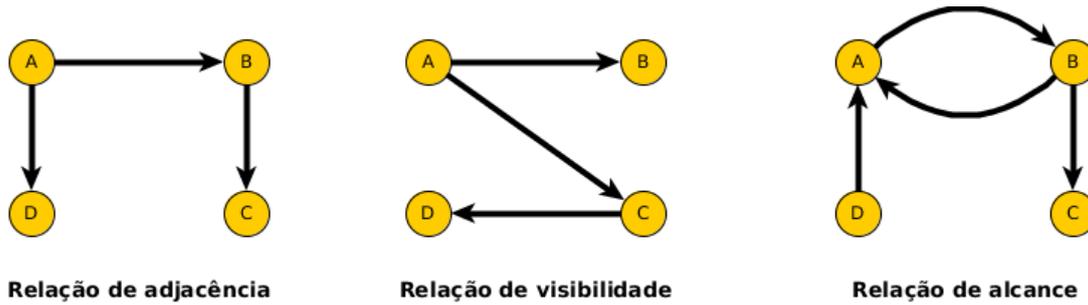


Figura 2.1: Relações e objetos do cenário de planejamento.

a outro sejam necessárias a ocorrência das três relações, o único movimento possível é alcançar B a partir de A.

Um fato importante dessas propriedades que definem cenários é a possibilidade de serem imutáveis, ou seja, elas são mantidas em qualquer estado de  $S$ . Para formalizar esse conceito é necessário definir variável de estado, que segundo Ghallab et al. [1] pode ser descrito conforme a Definição 2.

**Definição 2.** *Uma variável de estado é um termo  $x = id(o_1, o_2, o_3, \dots, o_n)$ , onde  $id$  representa um identificador e cada  $o_i$  é um objeto pertencente ao ambiente.*

Cabe ressaltar que as variáveis de estado podem ser entendidas tanto como proposições cujas imagens são valores lógicos de *Verdadeiro* e *Falso* quanto funções mapeadas em outros objetos. Dessa forma, na Figura 2.1, as variáveis de estado são por exemplo  $adjacente(A, B)$ ,  $visivel(A, C)$ ,  $alcance(D, A)$ , quando utilizadas como proposições lógicas. No segundo caso, apenas como ilustração, as variáveis podem ser  $adjacente(A) = \{B, D\}$ ,  $visivel(C) = \{D\}$ ,  $alcance(B) = \{A, C\}$ .

Uma função que mapeia cada variável de estado  $x$  a um valor da sua imagem é definida como *função de atribuição de variável*. Com isso, segundo Ghallab et al. [1], o conceito de *espaço de variável de estado* pode ser formalmente descrito conforme a Definição 3.

**Definição 3.** *Um espaço de variável de estado é um conjunto  $\nu$  de funções de atribuição de variável sobre um conjunto de variáveis de estado  $X$ . Cada uma das funções em  $S$  é definida como um estado pertencente a  $S$ .*

A Figura 2.1 representa um estado  $s_0 = \{adjacente(A, B) = Verdadeiro, \dots, alcance(D, A) = Verdadeiro\}$  no qual existem as seguintes variáveis de estado e funções de atribuição:

- Variáveis de estado:
  - $adjacente(A, B)$ ;
  - $adjacente(A, D)$ ;
  - $adjacente(B, C)$ ;
  - $visivel(A, B)$ ;
  - $visivel(A, C)$ ;
  - $visivel(C, D)$ ;

$alcance(A, B)$ ;

$alcance(B, A)$ ;

$alcance(B, C)$ ;

$alcance(D, A)$ ;

- Funções de atribuição:

$adjacente(A, B) = Verdadeiro$ ;

$adjacente(A, D) = Verdadeiro$ ;

$adjacente(B, C) = Verdadeiro$ ;

$visivel(A, B) = Verdadeiro$ ;

$visivel(A, C) = Verdadeiro$ ;

$visivel(C, D) = Verdadeiro$ ;

$alcance(A, B) = Verdadeiro$ ;

$alcance(B, A) = Verdadeiro$ ;

$alcance(B, C) = Verdadeiro$ ;

$alcance(D, A) = Verdadeiro$ ;

No caso das proposições que não estão explicitamente descritas, assume-se que às variáveis de estados são atribuídos o valor *Falso*, definindo o que conhecido como *hipótese do mundo fechado* [4]. Dessa forma, o conjunto de funções de atribuição do estado  $s_0$  possui ainda  $adjacente(B, A) = Falso$ ,  $visivel(B, D) = Falso$ , etc.

Retornando à discussão a cerca das propriedades entre os objetos e estendo-a às definições de estado e funções de atribuição de variável, é possível formalizar o conceito de rigidez apresentado em Ghallab et al. [1] conforme a Definição 4.

**Definição 4.** *Seja  $r$  uma relação entre objetos do ambiente de planejamento em um estado de  $S$  e  $f(x) \in S$  uma função de atribuição da variável de estado  $x$ ,  $r$  é dita rígida (ou invariante) se para  $\forall s_i \in S$ ,  $r$  se mantém inalterada. De forma análoga,  $f(x)$  é dita rígida se  $f(x) = constante$  para qualquer  $s_i \in S$ .*

Assumindo, por exemplo, um conjunto de estados  $S = \{s_0, s_1, \dots, s_n\}$  e que a função de atribuição  $adjacente(A, B) = Verdadeiro$  seja mantida em qualquer  $s_i \in S$ , essa função apresenta a propriedade de rigidez.

O segundo item da Definição 1 trata do conceito de ação. Porém, antes de defini-lo é necessário formalizar *literal* ou *átomo*, de acordo com a Definição 5 apresentada em Ghallab et al. [1].

**Definição 5.** *Um literal ou átomo é uma expressão contendo  $r(o_1, o_2, \dots, o_n)$  ou  $id_v(x_1, x_2, \dots, x_m) = x_0$ , onde  $r$  é uma relação entre objetos,  $id_v$  o identificador de alguma variável de estado e  $x_i$  representa variáveis que podem assumir qualquer valor ou constante, com exceção a uma variável de estado.*

Quando em um literal todas as ocorrências de variáveis  $x_i$  foram substituídas por constantes ou objetos do domínio, ou seja, não existem variáveis, esse átomo é chamado de proposição ou instância (*ground*); caso contrário ele é classificado como não instanciado (*unground*).

Um modelo de ação (ou *template*) pode ser construído a partir de literais *unground* e quando todas variáveis forem substituídas por valores, uma ação é dita instanciada a partir do modelo, contendo apenas átomos do tipo *ground*. A instância da ação é dada o nome de *operador* [1].

**Definição 6.** Um modelo de ação  $\alpha$  é da forma:  $\alpha = \langle def(\alpha), pre(\alpha), eff(\alpha) \rangle$ , onde:

- $def(\alpha)$  define um nome para a ação e um conjunto de parâmetros, isto é, variáveis;
- $pre(\alpha)$  representa um conjunto finito de literais indicando condições para a execução da ação, ou simplesmente, um conjunto de pré-condições que devem ser satisfeitos no estado em que a ação for executada;
- $eff(\alpha)$  representa um conjunto finito de efeitos, cada um da forma  $id_v(x_1, \dots, x_j) \leftarrow x_0$  e onde a variável de estado descrita assumirá o valor  $x_0$  após a execução da ação.

Adicionado ao cenário descrito na Figura 2.1 o literal  $atual(obj) = pos1$ , que representa a posição atual do objeto, um exemplo de modelo de ação que descreve a capacidade de um objeto atualizar sua posição pode ser descrito a seguinte forma:

$$mover = \langle def(mover), pre(mover), eff(mover) \rangle,$$

onde:

- $def(mover) = mover(obj, pos1, pos2)$ ;
- $pre(mover) = \{adjacente(pos1, pos2), visivel(pos1, pos2), alcance(pos1, pos2), atual(obj) = pos1\}$ ;
- $eff(mover) = \{atual(obj) \leftarrow pos2\}$ .

Conforme apresentado na Definição 6, o conjunto  $pre(\alpha)$  determina quais condições devem ser satisfeitas para que uma instância daquela ação possa ser executada. Portanto, para o operador  $\alpha$  ser aplicável em um estado  $s$  é preciso que todos literais determinados por  $pre(\alpha)$  sejam satisfeitos pelas funções de atribuição de variável de estado presentes em  $s$ . Quando  $s$  satisfaz  $pre(a)$ , a transição de estado causada pelo operador  $a$  é definida pela Equação 2.1. Consequentemente, o último elemento apresentado na tupla da Definição 1, o qual define um sistema de transição, isto é, a função  $\gamma$ , está formalizado conforme [1].

$$\gamma(s, a) = eff(a) \cup \{id_v(x_i, \dots, x_j) = x_0 \in s \mid id_v(x_i, \dots, x_j) \leftarrow x_k \notin eff(a), x_k \neq x_0\} \quad (2.1)$$

Dessa forma, considerando o estado  $s_0$  descrito na Figura 2.1 e o operador  $mover(obj, A, B)$  tem-se que o novo estado  $s_1 = \gamma(s_0, mover)$  será a união de  $\{atual(obj) = pos2\}$  com todas demais variáveis de estado inalteradas.

De posse do formalismo herdado dos sistemas de transição é possível especializar os conceitos para o cenário de planejamento automatizado e definir *domínio de planejamento* conforme a Definição 7 apresentada em Ghallab et al. [1].

**Definição 7.** Um domínio de planejamento é uma tupla  $\Sigma_D = \langle S, O, \gamma \rangle$ , onde  $O$  é o conjunto de todos operadores instanciados a partir de um conjunto  $A$  de modelos de ação.

Conforme foi apresentado na Seção 2.1, a finalidade do planejamento automatizado é definir uma sequência de ações possíveis para atingir um conjunto de estados desejáveis. Entretanto, de acordo com as definições supracitadas, essa sequência é composta de operadores, isto é, instâncias de ações. Logo, a solução para um problema de planejamento clássico é um plano  $\pi = \langle a_1, a_2, \dots, a_n \rangle$ , onde  $a_i \in O$  é uma sequência finita de operadores.

Um problema de planejamento clássico que pode ser caracterizado por um estado inicial e final além das regras de transformação. A Definição 8 constante de Ghallab et al. [1] apresenta o conceito de planejamento automatizado.

**Definição 8.** Um problema de planejamento é uma tupla  $P = \langle \Sigma_D, s_0, g \rangle$ , onde  $\Sigma_D$  é um domínio de planejamento,  $s_0$  é o estado inicial e  $g$  é um conjunto de literais que devem ser satisfeitos nos estados meta.

Portanto, a finalidade do planejamento automatizado é calcular um plano  $\pi$ , tal que  $\gamma(s_{i-1}, a_i) = s_i$ , para  $i = 1, 2, \dots, n$  tal que  $s_n$  satisfaça todos elementos de  $g$ .

Independente da abordagem adotada para representar o ambiente, seja a de espaço de estados ou de planos, o processo de busca da solução pode ser executado a partir do estado inicial em direção aos objetivos ou de forma inversa, caracterizando as técnicas de busca progressivas (*forward*) e regressivas (*backward*), respectivamente apresentadas nos Algoritmos 1 e 2, apresentados em Ghallab et al. [1].

---

**Algoritmo 1** Busca progressiva

---

**Entrada:**  $\Sigma_D, s_0, g$

**Saída:** Plano  $\pi$  ou falha

```

1: Fronteira  $\leftarrow \{(\langle \rangle, s_0)\}$  ▷ Tupla formada pelo plano vazio e estado  $s_0$ 
2: Visitados  $\leftarrow \emptyset$ 
3: while Fronteira  $\neq \emptyset$  do
4:   Selecione um nó  $v = (\pi, s) \in$  Fronteira
5:   Fronteira  $\leftarrow$  Fronteira  $- \{v\}$ 
6:   Visitados  $\leftarrow$  Visitados  $\cup \{v\}$ 
7:   if  $s$  satisfizer  $g$  then
8:     return  $\pi$ 
9:   Novos  $\leftarrow \{(\pi.a, \gamma(s, a)) \mid s \text{ satisfaz } pre(a)\}$ 
10:  Remova elementos de Fronteira, Visitados, Novos
11:  Fronteira  $\leftarrow$  Fronteira  $\cup$  Novos
12: return Sem solução

```

---

Na Linha 9 do Algoritmo 1 é utilizada a operação de concatenação de uma nova ação ao plano. Dessa forma, considerando um plano  $\pi = \langle a_1, a_2, \dots, a_n \rangle$  e um operador  $a_m$ , a operação  $\pi.a$  resulta em  $\pi = \langle a_1, a_2, \dots, a_n, a_m \rangle$ .

O algoritmo regressivo, por explorar o espaço de busca a partir dos objetivos, utiliza a função de regressão  $\gamma^{-1}$  que calcula o estado a partir do qual é possível atingir um dos

---

**Algoritmo 2** Busca regressiva

---

**Entrada:**  $\Sigma_D, s_0, g$ **Saída:** Plano  $\pi$  ou falha

```
1:  $\pi \leftarrow \langle \rangle$ 
2:  $goal \leftarrow g$ 
3: loop
4: if  $s_0$  satisfizer  $goal$  then
5:   return  $\pi$ 
6:  $Ops \leftarrow \{o \in O \mid g \in eff(o)\}$ 
7: if  $Ops = \emptyset$  then
8:   return Sem solução
9: Escolha  $o \in Ops$ 
10:  $goal \leftarrow \gamma^{-1}(goal, o)$ 
11:  $\pi \leftarrow a.\pi$ 
12: return Sem solução
```

---

elementos de  $g$ . Formalmente, a função é apresentada pela Equação 2.2 apresentada em Ghallab et al. [1].

$$\gamma^{-1}(g, a) = pre(a) \cup \{id_v(x_1, x_2, \dots, z_n) \in g \mid id_v(x_1, x_2, \dots, z_n) \notin eff(a)\} \quad (2.2)$$

Para otimizar o processo de exploração do espaço de busca, os algoritmos de planejamento utilizam diversos tipos de funções heurísticas para guiar as escolhas feitas em cada iteração. As funções heurísticas são usadas nos algoritmos especialmente na etapa de remoção e seleção de elementos dos conjuntos, indicadas na Linha 10 do Algoritmo 1 e Linha 9 do Algoritmo 2.

Uma das possibilidades de função heurísticas é baseada na noção de *delete-relaxation*, ou seja, um relaxamento do problema no qual a execução de uma ação nunca remove literais antigos de um estado, mas simplesmente adiciona novos [1]. Nesse contexto, alguns conceitos surgem naturalmente e são apresentados nas Definições 9, 10 e 11.

**Definição 9.** *Um estado relaxado  $\hat{s}$  é um conjunto de literais do tipo ground tal que toda variável de estado do cenário de planejamento está presente em pelo menos um literal.*

**Definição 10.** *Um estado relaxado  $\hat{s}$  satisfaz um conjunto de literais  $g$  se existir um subconjunto  $s \subseteq \hat{s}$  que satisfaz  $g$ .*

**Definição 11.** *Um operador  $a$  é aplicável em um estado relaxado  $\hat{s}$  se  $\hat{s}$  satisfaz  $pre(a)$ . A transição a partir de um estado  $\hat{s}$  aplicando um operador  $a$  é definido pela Equação 2.3:*

$$\gamma^+(\hat{s}, a) = \hat{s} \cup \gamma(s, a) \quad (2.3)$$

Embora a hipótese de ações determinísticas simplifique o processo de planejamento automatizado, ela pode impedir a representação de um problema real. Sendo assim, entre os problemas em aberto na área de planejamento, o uso de modelos probabilísticos

merece destaque. Nesse contexto, os efeitos causados pelas ações devem estar associadas à distribuições de probabilidade [11].

Com a abordagem probabilística, o plano produzido não é uma solução absoluta, mas sim uma sequência de ações com certa chance de atingir os objetivos. Esses modelos, denominados *Markov Decision Process (MDP)*, englobam diversos problemas tais como diagnóstico médico, tratamento de doenças, navegação de robôs, gerenciamento logístico com demanda incerta. O planejamento probabilístico pode tornar-se ainda mais desafiador, uma vez que pode ser difícil ou impossível determinar as distribuições de probabilidades que descrevam o comportamento do ambiente. Nesse caso, o MDP é reclassificado em *Markov Decision Processes with imprecise probabilities (MDP-IPs)*, onde as funções de transição são parametrizadas e sujeitas à restrições [12].

Como o foco desse trabalho é o planejamento clássico, o modelo probabilístico não será explorado de forma mais detalhada.

## 2.2 Linguagens e Domínios de Planejamento

O uso de uma boa linguagem de planejamento é um dos principais fatores para um processo de planejamento eficiente. Pode-se citar como uma linguagem tradicional em planejamento a *Stanford Research Institute Problem Solver (STRIPS)* [13]. A linguagem STRIPS propõe um modelo simples e compacto para especificar os domínios e problemas de planejamento, incluindo  $\Sigma_D$  (Definição 7) e  $P$  (Definição 8), respectivamente. Essa simplicidade implica em algumas restrições na representatividade da linguagem, como por exemplo:

- o estado inicial e o conjunto de pré-condições apenas aceitam literais positivos, isto é, com valor lógico *Verdadeiro*;
- não é permitida a disjunção de literais nos objetivos, pré-condições e efeitos;
- somente literais *ground* são permitidos nos objetivos;
- falta de definição de tipos.

Como consequência, a linguagem STRIPS tem sido evoluída e estendida visando aumentar a capacidade de representação de problemas de planejamento. Nessa direção, *Action Description Language (ADL)* [14] usa um modelo algébrico para definir os estados do mundo e estende STRIPS oferecendo:

- definição de tipos para objetos e parâmetros dos modelos de ações, o que favorece a compreensão dos problemas;
- uso de negação de literais nos objetivos e pré-condições;
- disjunção de fórmulas.

Apesar de ADL ser bem popular, a extensão mais bem sucedida é a *Planning Domain Definition Language (PDDL)* [15], que tem sido a linguagem referência na comunidade (*International Planning Competition - IPC*<sup>1</sup>) além de ser adotada pela maioria

---

<sup>1</sup><http://icaps-conference.org/ipc2008/deterministic/PddlResources.html>

dos planejadores. A linguagem PDDL oferece um conjunto de características, das quais destacam-se:

- modelagem das ações baseada em STRIPS;
- ações hierarquizadas;
- axiomas de domínios que são fórmulas lógicas que estabelecem relações entre os fatos que são satisfeitos em um estado (ao contrário de ações, as quais definem relações entre estados sucessivos).

Da mesma maneira que na linguagem STRIPS, extensões foram realizadas a partir de PDDL, gerando as seguintes versões:

- PDDL2.1 [16] - melhora a expressividade de PDDL adicionando gerenciamento de tempo e capacidades numéricas. Entre suas contribuições, destacam-se:
  - a proposta de uma sintaxe definitiva que permite comparações entre expressões numéricas;
  - a especificação de ações com duração discreta e contínua;
  - a definição de métricas, expressões numéricas ou temporais, para avaliar um problema.
- PDDL2.2 [17] - permite a definição de predicados que não são afetados por nenhuma ação e também de literais que tem seus valores lógicos atribuídos em instantes estabelecidos, independentemente da execução de uma ação;
- PDDL3.0 [18] - estende a capacidade de expressar a qualidade de um plano introduzindo características que permitem definir condições que devem ser satisfeitas por um sequência de estados durante a execução do plano. Além disso, é possível priorizar tais condições;
- PDDL3.1 [19] - enriquece a linguagem com a possibilidade de representar questões de satisfatibilidade, de utilizar variáveis de estado mapeadas em conjunto finito de objetos, definidos por um tipo e de atribuir custos às ações.

As linguagens de planejamento são utilizadas para descrever dois tipos de arquivos. No arquivo de domínio, os modelo de ações, predicados e variáveis de estados são listados. O arquivo de problema expressa o estado inicial, isto é, a percepção do ambiente em um primeiro momento. Os objetivos e os objetos sobre os quais as ações são executadas também são descritos nesse arquivo.

Com relação aos domínios de planejamento automatizado, existe uma coleção de casos de teste que é utilizada pela comunidade científica para a avaliação de desempenho dos planejadores. Esses testes, conhecidos como *domínios do IPC*, são baseados em problemas reais de planejamento que tem sido explorados por competições ao longo dos anos. Novos testes são atualizados a cada vez que competições são realizadas. A primeira delas ocorreu em 1998 e apresentou sete domínios, basicamente relacionados a questões de logística e deslocamento. Maiores detalhes sobre esses e outros domínios do IPC, obtidos de [20].

Entre alguns domínios do IPC, destacam-se:

- *Satellite* - escalonamento de observações de satélites que incluem a coleta e o armazenamento de dados usando os diferentes instrumentos para observar uma coleção de alvos;
- *Rovers* - uso de um conjunto de robôs para percorrer pontos do planeta realizando uma variedade de operações de coleta de amostras e transmissão de dados para a base;
- *Zeno-travel* - relacionado a uma questão de transporte aéreo de pessoas;
- *Elevators* - uso de diferentes tipos de elevadores para deslocamento de pessoas;
- *Depots e Logistics* - execução de atividades logísticas relacionadas a armazenamento, transporte e entrega de produtos.

## 2.3 Sistema Multiagente

O primeiro passo para entender um Sistema Multiagente (SMA) é a caracterização do conceito de agente. Apesar de não existir um consenso universal, a Definição 12, apresentada em Wooldridge [21], sintetiza os termos principais.

**Definição 12.** *Um agente é um sistema de computador que está situado em algum ambiente e que é capaz de executar ações autonomamente nesse ambiente para atingir seus objetivos.*

Uma caracterização que naturalmente surge a partir dessa definição é a noção de agente inteligente, que deve apresentar as seguintes capacidades:

- Reatividade - perceber o ambiente e reagir às mudanças ocorridas nele a fim de satisfazer seus objetivos;
- Proatividade - exibir um comportamento orientado a objetivos, tomando a iniciativa da realização de ações que conduzem aos objetivos;
- Sociabilidade - interagir com outros agentes.

A formalização do agente depende de dois conceitos: ambiente e ações. O ambiente é um conjunto finito de estados  $E = \{e_0, e_1, \dots, e_n\}$ . Os agentes possuem uma coleção de ações disponíveis,  $Ac = \{\alpha_0, \alpha_1, \dots, \alpha_m\}$ , que transformam o ambiente.

Esses dois conceitos se relacionam da seguinte maneira: o ambiente inicialmente está em um estado  $s_0$  e o agente escolhe uma ação  $\alpha_j$  para executar a partir de  $s_0$ ; como consequência dessa ação, o ambiente assumirá um novo estado e então o agente escolhe uma segunda ação; esse ciclo se repete até que uma condição de parada seja atingida. A partir dessa sequência de mudanças, o conceito de *execução*, presente em Wooldridge [21], é apresentado na Definição 13.

**Definição 13.** *Uma execução  $r$  de uma agente em um ambiente é uma sequência de estados do ambiente e ações intercaladas, conforme apresentado na Equação 2.4.*

$$r : e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} \dots e_{u-1} \xrightarrow{\alpha_u} e_u \quad (2.4)$$

As possíveis execuções  $r$  podem ser agrupadas formando os conjuntos:

- $R$  - conjunto de todas sequências finitas de  $r$ , ou seja,  $R = \{r_0, r_1, \dots\}$ ;
- $R^{Ac}$  - subconjuntos de  $R$  que acabam com uma ação;
- $R^E$  - subconjuntos de  $R$  que acabam com um estado de ambiente.

As ações de um agente causam efeitos no ambiente, que por sua vez são representados por uma função de transformação de estado, conforme definida em Wooldridge [21] e apresentada na Equação 2.5. Portanto, uma função  $\tau$  mapeia uma execução  $r$  em um conjunto de partes dos possíveis estados resultantes da execução da ação  $\alpha$  que encerra  $r$ .

$$\tau : R^{Ac} \rightarrow \wp(E) \quad (2.5)$$

Formalmente, o ambiente é uma tupla  $Env = \langle E, e_0, \tau \rangle$ , onde  $E$  é um conjunto de estados de ambiente,  $e_0 \in E$  é um estado inicial, e  $\tau$  é a função de transformação (Equação 2.5).

O ambiente pode apresentar características importantes que o define, conforme [21, 22], tais como:

- Acessível  $\times$  Inacessível - um ambiente acessível, ou totalmente observável, é aquele no qual o agente pode obter informações completas, precisas e atuais sobre os estados do ambiente. Quando essa condição não é atendida, o ambiente é considerado inacessível ou parcialmente observável;
- Determinístico  $\times$  Não-determinístico - se não há incerteza em relação às execuções dos efeitos das ações, o ambiente é dito determinístico;
- Estático  $\times$  Dinâmico - ambientes nos quais as mudanças são causadas apenas por intervenções das ações dos agentes são classificados como estáticos. Quando essas atualizações podem ocorrer como consequência de eventos externos aos agentes, o ambiente é dinâmico;
- Discreto  $\times$  Contínuo - um ambiente é discreto se há um número fixo e finito de ações e percepções sobre ele, caso contrário é denominado contínuo;
- Episódico  $\times$  Sequencial - quando as ações realizadas em um estado (episódio) do ambiente não depende daquelas realizadas anteriormente, o ambiente é episódico. Quando a decisão atual afeta as decisões futuras, o ambiente é sequencial;
- Único agente  $\times$  Múltiplos agentes - classificação de acordo com o número de agentes existentes no ambiente.

Finalmente, agentes podem ser definidos como funções que mapeiam execuções em ações, conforme a Equação 2.6 apresentada em Wooldridge [21].

$$Ag : R^E \rightarrow Ac \quad (2.6)$$

Dessa forma, a relação entre agente e ambiente pode ser caracterizada pelo conjunto das possíveis execuções do agente no ambiente, representados por  $R(Ag, Env)$ .

Um SMA é um conjunto de entidades de software, denominadas agentes (Definição 12), que são capazes de perceber e alterar o estado do ambiente (Definição 13), no qual estão inseridos, por meio de seus sensores e atuadores, respectivamente. Como sistemas inteligentes, os agentes apresentam níveis de autonomia baseados nos processos de raciocínio. A maneira pela qual esse processo é implementado serve para classificar os agentes.

Os agentes mais simples de se implementar são denominados *reativos*, neste caso a escolha da ação é baseada apenas na sua percepção atual do ambiente, ignorando todo o histórico de eventos anteriores do ambiente. Também existem os agentes *orientadas a modelo* quando mantém uma descrição de como o ambiente é modificado, seja por intervenção direta do agente ou por um terceiro. No cenário de planejamento multiagente, essa capacidade é útil quando é necessário lidar com uma observação parcial do ambiente, fato que não satisfaz uma das hipóteses assumidas no cenário de planejamento clássico (Item nº 2 da relação de hipóteses apresentada na Seção 2.1, Página 6). Outra classificação para agentes *orientados a objetivos*, mostrados na Figura 2.2 apresentada e traduzida [22], realizam a seleção das ações baseadas no estado atual e nos objetivos, tornando-as integráveis às técnicas de planejamento automatizado. Entretanto, a noção de objetivos não garante que as ações escolhidas impliquem em um comportamento eficiente. Dessa forma, os agentes *baseados em utilidade* baseiam-se principalmente em fatores como custo e duração de execução para determinar a melhor opção (a hipótese de custo geralmente é desconsiderada no planejamento automatizado clássico). Por fim, um agente com *aprendizagem* é capaz de estender seu conhecimento inicial com o intuito de tornar mais eficiente e racional seu processo deliberativo.

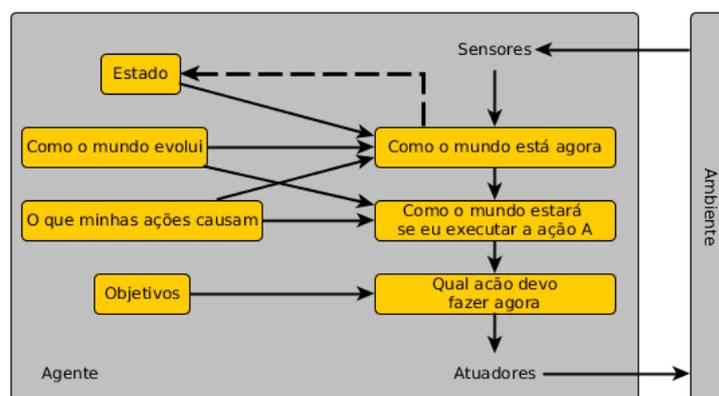


Figura 2.2: Agente orientado a objetivos.

Outro aspecto importante é a capacidade de comunicação entre os agentes, fato esse que permite que eles se relacionem e coordenem suas ações em um ambiente compartilhado seguindo a taxonomia apresentada na Figura 2.3 e traduzida de [23]. Devido a questões de eficiência e robustez e por ser essencialmente distribuído, um SMA deve preferencialmente executar o controle das atividades de forma descentralizada, caracterizando essa coordenação como o processo pelo qual decisões individuais resultam em boas decisões conjuntas para o grupo [24].

Por um lado, se essa coordenação envolver agentes antagônicos caracterizando um cenário de competição, um protocolo de negociação é necessário para definir o comporta-

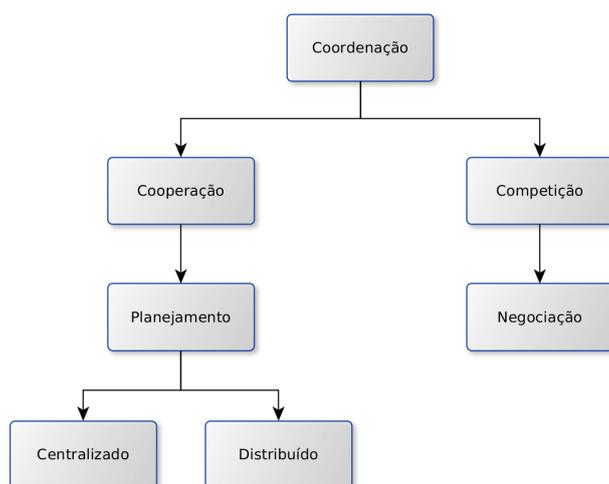


Figura 2.3: Taxonomia de coordenação.

mento adotado pelos agentes. Por outro lado, se existir cumplicidade entre eles em um cenário de cooperação, um protocolo de planejamento deve ser empregado, seja de forma centralizada ou distribuída. A principal motivação para existência desse protocolo é a necessidade da determinação de quais objetivos cada agente tentará satisfazer ou de quais tarefas realizarão, evitando contenção e situações de conflitos no uso dos recursos.

Os protocolos de interação entre agentes incluem diferentes abordagens de coordenação, cooperação, *contract net*, *blackboard systems*, negociação, manutenção de crença e mecanismos de mercado [21, 23]. Esses protocolos regem a forma de conversação entre os agentes, incluindo as trocas de mensagens necessárias para execução de tarefas. Dentre os possíveis protocolos, o *contract net* é um dos protocolos mais utilizados para negociação entre agentes, resultando em um meio eficiente de distribuição de tarefas [23]. As atividades que definem esse protocolo formam um conjunto de quatro etapas sequenciais:

1. o agente reconhece que existe um problema e então inicia a comunicação com os demais elementos;
2. esse agente contratante realiza um anúncio da tarefa que deseja resolver de acordo com o nível de conhecimento que tem dos demais agentes. Quando não possui nenhuma informação sobre suas capacidades, o agente é obrigado a enviar mensagens a todos os nós presentes na rede (*broadcast*). Entretanto, caso o contratante possua algum conhecimento sobre quais são os possíveis candidatos, ele pode realizar uma comunicação limitada (*multicast*). Por fim, se o nó apropriado para a tarefa for conhecido, a comunicação pode ser feita diretamente (*point-to-point*);
3. os agentes que receberam o anúncio realizam uma auto-análise para verificar se possuem os recursos necessários para realizar a tarefa e em caso positivo, cada entidade responde com o custo referente ao pedido;
4. o contratante decide qual agente contratará de acordo com sua estratégia de seleção, por exemplo de minimização de custo, e então enviará uma mensagem ao agente a ser contratado encerrando a execução do protocolo.

Visando implementar esse processo deliberativo, os agentes podem empregar técnicas de planejamento automatizado, caracterizando a área de pesquisa de Planejamento Multiagente (*Multi-Agent Planning - MAP*).

## 2.4 Planejamento Multiagente

De uma maneira simples, planejamento multiagente pode ser entendido como o processo de planejamento na presença de múltiplos agentes [25]. Entretanto, essa definição não torna claro quais papéis os agentes assumem, ou seja, se eles são empregados como executores ou como planejadores. Portanto, é possível que um único elemento efetue o planejamento de forma centralizada empregando diversos agentes ou que vários planejadores, de forma distribuída, calculem um plano para um único agente executor [26].

Quando existe apenas um agente, os conjuntos de ações disponíveis e objetivos são decorrentes de uma única fonte, tornando o problema em um planejamento individualizado. Nesse contexto, algumas condições surgem naturalmente e caracterizam o planejamento de um único agente (*Single-Agent Planning - SAP*) [2]. Em primeiro lugar, todas as capacidades de alterações do estado ambiente, decorrentes dos efeitos causados pelas execuções das ações, estão concentradas em um único agente e portanto não existe a possibilidade de nenhum tipo de divisão de responsabilidade. Entretanto, esse isolamento evita a ocorrência de questões relacionadas à privacidade das informações. Pelo mesmo motivo, não há necessidade de utilização de um protocolo de coordenação que gerencie a interação dos agentes. Por último, a presença de um único agente faz com que todos os objetivos definidos sejam ao mesmo tempo globais e individuais e portanto não há necessidade de utilização de um protocolo de alocação de tarefas. Considerando a definição mais simples, nesse contexto o único agente presente é tanto executor quanto planejador.

Por outro lado, quando o ambiente apresentar mais de uma entidade, o MAP pode ser melhor definido como o problema da busca do plano realizado e destinado a um grupo de agentes [3]. Enquanto no cenário SAP existe exclusivamente a necessidade de um componente de planejamento, nesse segundo caso há necessidade de gerenciar as interdependências entre as atividades, isto é, coordenar a realização dos esforços. A presença desses dois fatores permite aprimorar a definição de planejamento multiagente, conforme apresentado pela Definição 14, proveniente de [2]:

**Definição 14.** *Dada uma descrição do estado inicial, um conjunto de objetivos, um conjunto de (no mínimo dois) agentes, e para cada agente um conjunto de suas capacidades e seus objetivos individuais, encontre um plano para cada agente que satisfaça seus objetivos individuais, de tal forma que esses planos juntos são coordenados e garantam que os objetivos globais também sejam satisfeitos.*

A partir dessa definição, um problema de planejamento multiagente pode ser formalmente definido pela tupla  $\Pi$  apresentada na Definição 15 de [8]:

**Definição 15.**  $\Pi = \langle Ag, V, O, I, G \rangle$ , onde:

- $Ag = \{ag_1, ag_2, \dots, ag_n\}$  é um conjunto finito de agentes;
- $V$  é o conjunto de variáveis de estado, cada qual com seu domínio  $D(v)|v \in V$ ;

- $O$  é o conjunto de operadores, isto é,  $O = \bigcup_{i=1}^n O_{ag_i}$ ;
- $I$  é o estado inicial, isto é,  $I = \bigcup_{i=1}^n I_{ag_i}$ ;
- $G$  é o conjunto de objetivos, isto é,  $G = \bigcup_{i=1}^n G_{ag_i}$ .

As Definições 14 e 15 reforçam a importância de um protocolo de coordenação. Enriquecendo a análise com a taxonomia apresentada pela Figura 2.3 é possível verificar que é através de um processo de planejamento que os agentes efetivamente trabalham de maneira cooperativa.

Portanto, os conceitos de cooperação e competição surgem naturalmente em um cenário de MAP. Nesse sentido, a necessidade de uma interação construtiva entre os agentes pode ser decorrente de alguns fatores. Para facilitar esse entendimento, dois casos de planejamento automatizado clássico serão apresentados e utilizados.

O caso *rovers* consiste em um conjunto de robôs (agentes) que devem percorrer uma região com o objetivo de fotografá-la bem como de coletar amostras de rocha e solo. Nesse contexto, os agentes possuem os mesmos conjuntos de modelos de ações, porém os operadores instanciados podem ser diferentes devido à presença ou ausência de recursos, como equipamentos de imagem e coleta. Logo, o problema de planejamento  $P$  pode ser diferente para cada agente (Definição 8). A Figura 2.4 ilustra o problema *rovers*.

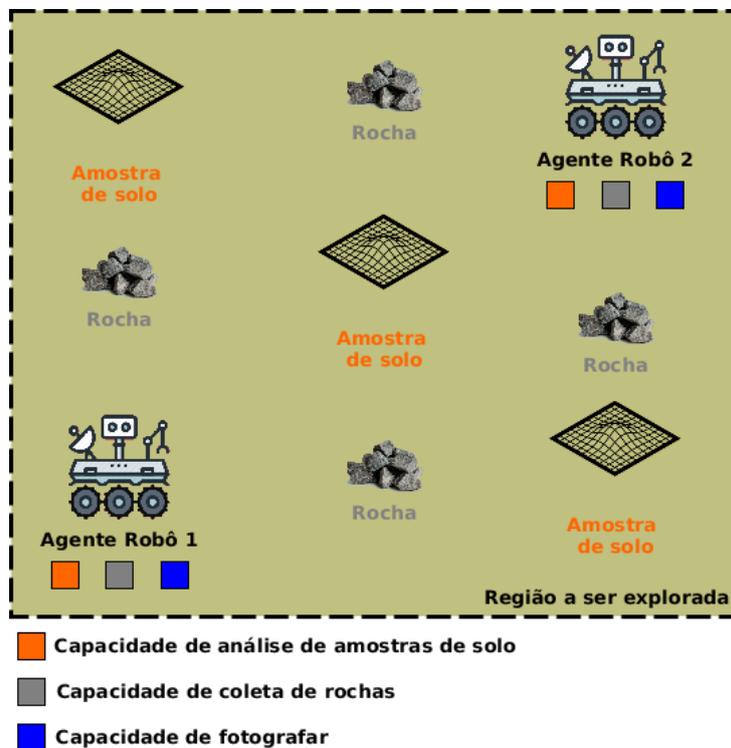


Figura 2.4: Exemplo de problema de planejamento no domínio *rovers*.

O segundo caso, conhecido como *logistics*, retrata a situação de entrega de materiais entre cidades utilizando agentes de dois tipos. O tipo *avião* é capaz de realizar a viagem

entre cidades transportando os objetos que devem ser entregues. Por sua vez, agentes do tipo *caminhão* são limitados ao deslocamento dentro de um única cidade. Nesse contexto, a questão de necessidade de cooperação entre os executores fica clara posto que o transporte de um material, inicialmente na Cidade A para um destino na Cidade B, deve envolver agentes dos dois tipos. O problema é ilustrado na Figura 2.5

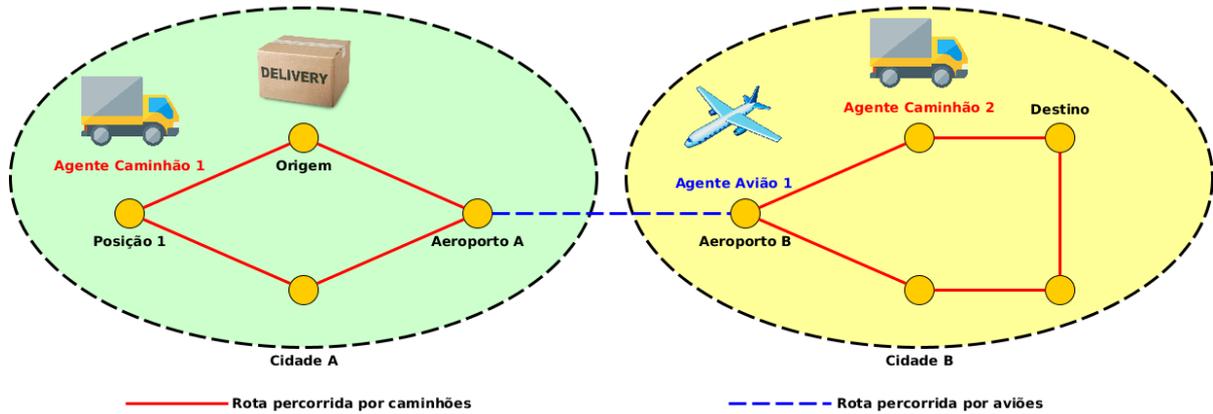


Figura 2.5: Exemplo de problema de planejamento do tipo *logistics*.

Os fatores que implicam na necessidade de cooperação em um planejamento multi-agente é estudado em Zhang et al. [26]. Nesse trabalho, os agentes são classificados como homogêneos ou heterogêneos de acordo com suas variáveis de estado e suas capacidades. No primeiro caso, os agentes possuem conjuntos idênticos de operadores e portanto as variáveis de estado e capacidades também são iguais. Conseqüentemente, qualquer agente é capaz de calcular e executar um plano que satisfaz os objetivos. No caso de agentes heterogêneos, essa condição é resultante de diferenças entre os conjuntos de agentes e operadores. No exemplo *logistics*, os agentes do tipo *caminhão* e *avião* possuem capacidades distintas, isto é, transportes dentro da mesma cidade e viagens entre cidades, respectivamente.

Como a pesquisa de [26] tem foco no papel de executores dos agentes, a análise apresentada é fortemente baseada no conjunto de capacidades (operadores) de cada elemento do sistema. Entretanto, não é apenas nesse trabalho que o conceito de cooperação é observado sob esta dimensão. É comum verificar nos trabalhos correlatos que os problemas de planejamento são classificados conforme [8], da seguinte forma:

- Fracamente acoplados (*loosely-coupled*): os agentes possuem as mesmas capacidades de tal forma que cada agente pode individualmente satisfazer cada objetivo. As atividades podem ser executadas sem a necessidade de cooperação entre os agentes. O caso *rovers* é classificado dessa maneira;
- Fortemente acoplados (*tightly-coupled*): os agente precisam interagir entre si para que os objetivos possam ser satisfeitos. O caso *logistics* é classificado dessa maneira.

Entretanto, como essa classificação baseia-se no conjunto de capacidades dos agentes, algumas condições para a execução do planejamento não são consideradas sob a ótica da cooperação e conseqüentemente da coordenação das atividades. Um exemplo seria um

cenário semelhante àquele descrito na Figura 2.4, porém com os dois agentes possuindo capacidades diferentes ao mesmo tempo que não precisam interagir para satisfazer sub-conjuntos dos objetivos. Em outras palavras, sendo o *Robô 1* capaz de coletar rochas enquanto o *Robô 2* tem a capacidade de fotografar, nenhum deles depende do outro para executar tais tarefas. Um problema nesse cenário seria atribuir objetivos aos agentes negligenciando essas capacidades.

A questão é que quando se deseja avaliar os problemas de planejamento considerando os aspectos referentes tanto ao planejamento quanto à execução do plano, a análise do conjunto de capacidades é necessária mas não é suficiente. Dessa forma, uma extensão da proposta apresentada em Zhang et al. [26], considerando não apenas a homogeneidade ou heterogeneidade dos agentes mas também capacidade de cada um planejar suas atividades de forma individual, configura uma classificação mais abrangente dos problemas. A Figura 2.6 apresenta tal classificação.

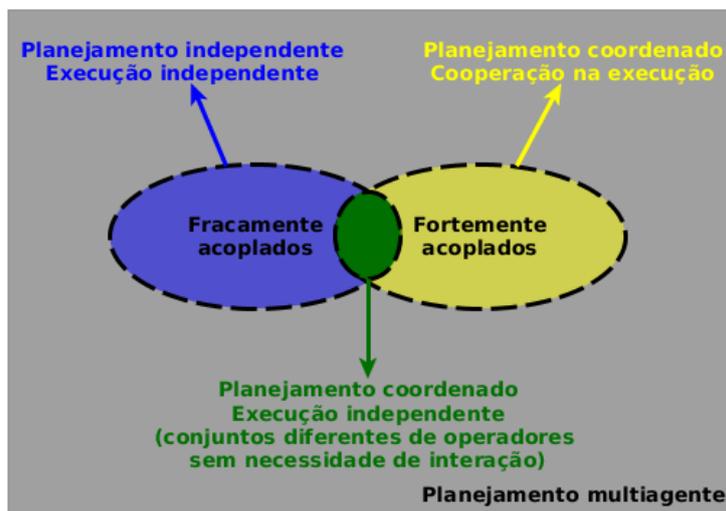


Figura 2.6: Classificação de MAP.

Portanto, a cooperação deve ser buscada tanto na fase de planejamento quanto na de execução dos planos permitindo que:

- agentes homogêneos (fracamente acoplados) possam planejar e executar suas ações de maneira independente, sem a necessidade de interação entre eles;
- agentes heterogêneos (fortemente acoplados) cooperem nas suas tarefas;
- agente com conjuntos de operadores diferentes porém capazes de satisfazer certos objetivos individualmente, coordenem quais metas cada um atenderá durante seus planejamentos e executem os planos de forma independente.

As diversas abordagens de MAP estão situadas dentro de um espaço tri-dimensional formado pelos eixos de dependência de recursos, cooperação e nível de comunicação [3]. Na dimensão que representa a relação entre os recursos, esses podem ser tanto independentes, quando não há qualquer compartilhamento ou dependência entre elas, quanto fortemente relacionados, no caso de ações conjuntas ou divisão de recursos. No eixo que mede a cooperação, os agentes são classificados em função da intenção de otimizar suas utilidades.

Nesse sentido, os agentes podem ser cooperativos quando buscam atingir tanto os objetivos globais quanto os individuais. Na terceira dimensão, o nível de comunicação é avaliado desde um cenário no qual nenhuma mensagem pode ser trocada até aquele que permite comunicações confiáveis. Essa definição é importante pois determina se a coordenação deverá ser feita antes ou durante o processo de solução do problema.

Em geral, as etapas para solução de um problema de MAP são compostas pelas seguintes etapas que, no entanto, não são obrigatórias em sua totalidade [2, 3, 9]:

1. Os objetivos globais são decompostos em conjuntos menores que podem ser feitos por um único agente;
2. Alocação dos objetivos aos agentes seguindo estratégias centralizadas ou distribuídas, o que influencia diretamente nos níveis de autonomia e privacidade. Esses métodos de alocação podem ainda considerar informações sobre as ações como, por exemplo, valores de retorno, custos de execução e até mesmo incertezas quando tratar-se de ambientes não-determinísticos;
3. Coordenação antes do planejamento, isto é, antes da computação dos planos. Nessa fase, as regras e condições de execução são definidas a fim de evitar conflitos durante o planejamento individual, isto é, situações nas quais uma ação pode desfazer um efeito já alcançado por outra ação;
4. As soluções são calculadas e cada agente pode utilizar um algoritmo de planejamento diferente;
5. Os planos individuais encontrados na fase anterior precisam ser integrados para formar um único plano conjunto ou para corrigir eventuais conflitos que podem surgir apesar do esforço da etapa de coordenação.

Em Dunin-Keplicz and Verbrugge [27], as fases necessárias para atingir um estado de cooperação efetiva são definidas como:

1. Reconhecimento de potencial - o trabalho em equipe começa quando o ambiente multiagente reconhece o potencial para a cooperação na busca pela satisfação dos objetivos. Nesse estágio, um conjunto de agentes é selecionado a partir dos recursos disponíveis e as entidades escolhidas devem ser colaborativas e também apresentar capacidades relevantes ao problema;
2. Formação de equipe: os agentes selecionados assumem o compromisso de atingir os objetivos de forma cooperativa e para tanto definem o protocolo de comunicação e interação que será seguido;
3. Planejamento: a etapa da geração do plano é iniciada e pode ser subdividida nas cinco fases descritas em de Weerd et al. [2], de Weerd and Clement [3], Cardoso and Bordini [9]: decomposição, alocação, coordenação, planejamento e integração. A composição dessas fases depende das condições de execução e portanto nem sempre todas serão necessárias ou empregadas nessa etapa [2];
4. Execução: fase de execução que inclui tanto a realização das ações quanto um processo de reconfiguração.

O processo de reconfiguração da fase de execução é justificado para ambientes dinâmicos, os quais podem sofrer mudanças provocadas por eventos diferentes dos efeitos das ações. Esse processo deve permear todas as fases anteriores para fornecer estratégias que evitem que o dinamismo do ambiente inviabilize a execução de um plano já calculado.

Dessa forma, pode existir a necessidade de ativação das fases já executadas, como por exemplo, a de seleção de potenciais entidades com o objetivo de buscar tanto um substituto para um agente impossibilitado de continuar quanto um novo elemento que possua capacidades requeridas pela mudança do ambiente. Essas alterações provavelmente provocarão a reexecução das fases seguintes, seja para ratificar ou retificar os protocolos de comunicação e interação escolhidos, seja para recalcular os planos.

No Capítulo 3 é apresentada uma revisão da literatura na qual são detalhados alguns trabalhos que configuram o estado da arte na área de MAP.

# Capítulo 3

## Trabalhos Correlatos

Existem várias pesquisas relacionadas ao tema de planejamento multiagente, cada uma delas lida com diferentes características do domínio do problema, tais como níveis de acoplamento de recursos, processo de coordenação, tipos de objetivos e privacidade de informações. Nessas abordagens, o processo de planejamento dos agentes é implementado de diversas maneiras, o que torna a análise dessa gama de possibilidades um desafio na área de pesquisa de MAP.

A fim de identificar os principais trabalhos na área de MAP e dessa forma orientar o desenvolvimento do modelo de planejamento multiagente, uma revisão sistemática da literatura foi realizada seguindo o protocolo proposto por Kitchenham [28]. Uma breve introdução sobre revisão sistemática e a forma pela qual ela foi aplicada nessa dissertação são apresentadas no Apêndice A.

Nesse capítulo, os principais trabalhos identificados são detalhados e, em seguida, uma proposta de taxonomia é apresentada a fim de classificar os trabalhos, destacando suas peculiaridades.

### 3.1 Principais Trabalhos

Os trabalhos destacados na revisão sistemática são detalhados a seguir com o objetivo de identificar suas características, contribuições, pontos fracos e principalmente as possibilidades de melhorias. Consequentemente, as decisões tomadas durante o desenvolvimento do modelo proposto e apresentado no Capítulo 4 foram motivadas e justificadas por essas observações.

#### 3.1.1 A Formal Analysis of Required Cooperation in Multi-Agent Planning

Em Zhang et al. [26] é estudada a necessidade de cooperação entre os agentes com o intuito de garantir condições para que objetivos inalcançáveis por um único agente sejam possíveis em um cenário multiagente. A principal questão discutida no trabalho é a análise das condições que definem a necessidade da presença de vários agentes na solução de um problema de planejamento. Esse tipo de verificação prévia, apesar de ser um ponto pouco explorado em outros trabalhos, é de grande importância porque define tipos distintos de uso para MAP, ou seja, situações nas quais vários agentes:

1. são necessários porque não existe um plano com um único agente que satisfaça todos os objetivos;
2. são úteis para melhorar a eficiência da execução, mesmo que um plano com um único agente seja capaz de atingir os objetivos.

Esses dois tipos de uso são importantes também para destacar e diferenciar conceitos próximos. Se por um lado, um plano multiagente é aquele no qual o papel do agente é de executor, por outro lado em um planejamento distribuído, esse papel pode ser de planejador, executor ou ambos.

Além da análise teórica da cooperação requerida entre os agentes, o trabalho apresenta um modelo de planejamento no qual o problema original é transformado para utilizar a quantidade mínima de agentes necessários para atingir os objetivos. Apesar dessa importante contribuição, inicialmente o modelo é capaz de apresentar o limite superior para o número de agentes quando esses são homogêneos, isto é, apresentam as mesmas capacidades.

Os resultados destacam que o modelo proposto é mais rápido que a ferramenta usada (MAP-LAPKT [29]) como comparação apesar de gerar planos com um maior número de ações, consequência direta da transformação do problema. Outra conclusão importante do trabalho é que muitos dos casos de testes usados para avaliar planejadores multiagentes são, na realidade, passíveis de solução por um único agente.

### 3.1.2 FMAP: Distributed Cooperative Multi-Agent Planning

Uma estratégia presente na literatura é a de computação distribuída e interativa dos planos. Em Torreno et al. [8], é apresentado um modelo que integra planejamento e coordenação em um ambiente totalmente distribuído, cujas principais vantagens são as preocupações com a privacidade da informação e a capacidade de poder ser empregado para solucionar problemas em domínios tanto fracamente como fortemente acoplados.

A construção da solução está baseada no conceito de refinamento, pelo qual um plano-base é progressivamente incrementado – adicionando ações – até que o objetivo seja atingido. O processo de refinamento começa com o plano inicial (composto pelo estado inicial  $\alpha_i$  e pelo objetivo  $\alpha_f$  a ser atendido), que é modificado por todos agentes. Esse plano é apresentado na raiz da árvore do espaço de busca que é explorado pelos agentes. A Figura 3.1, obtida de [8], mostra o primeiro nível da árvore gerada para uma determinada tarefa, ilustrando o refinamento realizado pelos agentes *ta1* e *ta2*. Cada nó  $\pi$  da árvore é um plano parcial composto por um operador *drive*.

Para a exploração da árvore, o FMAP faz uso de uma estratégia de busca A\* [22] e uma liderança democrática (*round-robin*). Cada interação no processo de busca é composta pelas etapas:

- Seleção do plano-base - o agente líder escolhe o plano mais promissor e o envia aos outros agentes;
- Geração do plano de refinamento - cada agente expande o plano-base, gerando os nós sucessores;
- Avaliação do plano de refinamento - cada agente avalia seus planos gerados, fazendo uso de uma função heurística;

- Comunicação do plano de refinamento - cada agente transmite para os demais o melhor plano gerado e o resultado da avaliação.

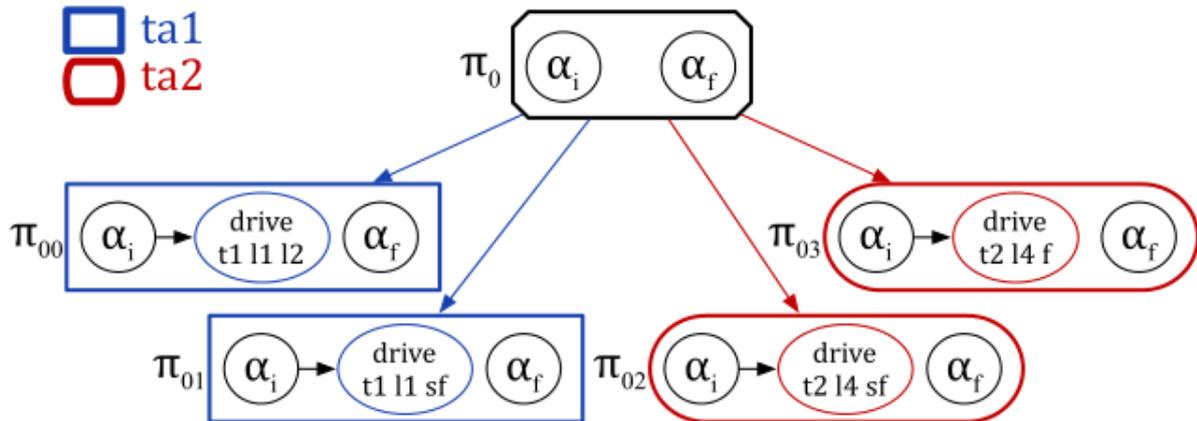


Figura 3.1: Primeiro nível da árvore de busca.

Um plano refinado é considerado solução quando ele apresenta o escalonamento de ações capazes de atingir o objetivo final a partir do estado inicial.

A principal limitação do FMAP é sua escalabilidade, pois quanto maior o número de agentes envolvidos no planejamento, maior é a quantidade de mensagens trocadas entre eles na fase de coordenação. Entre as etapas presentes em cada iteração do método, a única gravemente afetada pelo aumento de agentes é aquela referente à comunicação; as etapas de refinamento e avaliação são praticamente constantes.

Apesar de citar a qualidade de preservar a privacidade, os agentes descritos no modelo FMAP dependem de um conhecimento mútuo entre eles. O Código 3.1 ilustra um trecho de um arquivo de problema utilizado pela solução FMAP [8] no qual o agente *rover0* depende da definição explícita dos agentes *rover1* e *rover2*. Além da própria citação a outros agentes (Linhas 4 e 18), a privacidade também é violada quando objetos que não pertencem ao agente em questão (*rover0*), por exemplo *rover1store* e *rover2store*, são definidos (Linha 6).

Código 3.1: Trecho de arquivo de problema do FMAP.

```

1 (define (problem roverprob4123)
2 (:domain rover)
3 (:objects
4  rover0 rover1 rover2 - rover
5  waypoint0 waypoint1 waypoint2 waypoint3 waypoint4 waypoint5 - waypoint
6  rover0store rover1store rover2store - store
7  camera0 camera1 - camera
8  colour high_res low_res - mode
9  general - lander
10 objective0 objective1 - objective
11 )
12 (:shared-data
13  (communicated_soil_data ?w - waypoint)

```

```

14   (communicated_rock_data ?w - waypoint)
15   (communicated_image_data ?o - objective ?m - mode)
16   (at_soil_sample ?w - waypoint)
17   (at_rock_sample ?w - waypoint) -
18   (either rover1 rover2)
19   )
20   (:init (myRover rover0))

```

No entanto, FMAP é a única abordagem das estudadas nessa revisão da literatura, que apresenta preocupação com técnicas de execução paralela. Os autores afirmam que o procedimento de planejamento é otimizado para aproveitar todo o recurso computacional disponível, através da atribuição do mesmo número de *threads* para cada agente. Infelizmente, a implementação da paralelização não é explicitamente detalhada.

A avaliação dos resultados destaca que o FMAP além de ser independente de domínio, isto é, capaz de resolver problemas fracamente e fortemente acoplados, também apresenta tempos de execução melhores do que as propostas usadas como comparação.

### 3.1.3 MADLA: Planning With Distributed and Local Search

Em Štolba and Komenda [10] é proposto um planejador de domínio independente que combina dois tipos de heurísticas. Em uma primeira fase, o cálculo heurístico baseia-se apenas nas informações de um agente isoladamente, para em seguida, utilizar um compartilhamento de dados para estimar o valor.

De maneira similar ao FMAP [8], esse trabalho apresenta o uso de técnicas de programação paralela através da alocação de uma *thread* por agente e também visa manter a privacidade, compartilhando de maneira seletiva as informações. Entretanto, o planejamento é feito de maneira centralizada diferentemente de FMAP [8].

Uma questão interessante apresentada nessa pesquisa é o fato da tradução da entrada em formato MA-STRIPS para PDDL, o que de certa forma, destaca uma necessidade de desacoplamento entre as atividades de processamento dos arquivos de entrada e de planejamento propriamente dito.

Em relação a avaliação do modelo, o trabalho destaca que a combinação dos tipos de heurísticas aumenta a capacidade de solução de problemas, porém não apresenta métricas como tempo de execução ou tamanho dos planos.

### 3.1.4 MAPJA: Multi-agent planning with joint actions

Um caso especial de planejamento multiagente é quando os agentes devem cooperar para que os objetivos sejam atingidos, uma vez que individualmente eles não podem satisfazê-los. Apesar dessa preocupação ter sido explorada em Zhang et al. [26], trabalho no qual é apresentada uma formulação teórica da necessidade de cooperação, existe um aspecto que é pouco explorado na área de MAP.

Em Chouhan and Niyogi [30], o foco do trabalho é a questão de ações conjuntas (*joint actions*), isto é, agentes executando tarefas sobre o mesmo objeto. Sua principal diferença em relação a outros trabalhos que também são capazes de resolver problemas de planejamento que envolvam domínios fortemente acoplados, nos quais a cooperação é uma exigência, é a forma como os conjuntos de ações são tratados. Enquanto FMAP [8] e MADLA [10] consideram que agentes que possuam o mesmo conjunto de ações

necessariamente apresentam as mesmas capacidades, o trabalho em pauta assume que mesmo com coleções iguais, os agentes podem executar as ações de maneiras distintas. Um exemplo dessas peculiaridades é o fato de dois agentes (robôs) serem capazes de levantar blocos, porém com capacidades diferentes, sendo o primeiro limitado a 10 kg e outro a 20 kg.

Em seu modelo MAPJA, Chouhan and Niyogi [30] explora a transformação do problema original em dois tipos de problemas. Se o novo problema não necessitar de ações conjuntas, o problema é então resolvido por um planejamento multiagente existente e disponível. Porém se tal condição for necessária, o número de agentes é calculado e o planejamento é realizado de forma centralizada. Diferente de outros modelos, esse trabalho não assume nenhuma hipótese sobre quais agentes são necessárias e também não envolve todas entidades disponíveis no processo de planejamento.

Na fase de avaliação, o modelo é comparado com uma abordagem que transforma um problema MAP em vários problemas menores (SAP), os quais são combinados na fase final [6]. Os resultados mostram que o modelo proposto além de apresentar um número menor de nós expandidos durante a busca, também executa o processo em um tempo menor quando comparado com [6]. Os testes consideraram o número de agentes variando entre dois e cinco e nenhuma avaliação da propriedade de escalabilidade do modelo foi apresentada.

### 3.1.5 Multi-Agent Planning by plan Reuse

Um obstáculo que as abordagens devem buscar superar está relacionado com o processo de coordenação entre os agentes. Se eles cooperam sob um controle centralizado, um grande número de mensagens pode ser produzido com o objetivo de garantir a coordenação. Uma possível solução para esse desafio é a transformação do problema de MAP para várias instâncias individuais de planejamento (SAP). Essa estratégia tem sido seguida por trabalhos recentes nos quais os objetivos são divididos e a distribuição deles pelos agentes disponíveis é uma solução em comum, embora diferentes abordagens e técnicas sejam aplicadas [6, 9].

Em Borrajo [5], a abordagem proposta considera agentes com informações públicas e privadas que comunicam-se em uma estrutura com topologia de anel. Nesse trabalho, o processo de planejamento é composto por seis etapas:

1. Um subconjunto dos objetivos globais é delegado para cada agente ao mesmo tempo que seus objetivos privados são preservados. Em relação as estratégias usados nessa etapa de delegação, as possibilidades são as seguintes:
  - atribuir cada objetivo a todos agentes caso esse tenha a capacidade de satisfazê-lo;
  - atribuir todos os objetivos possíveis ao primeiro agente e então removê-los do conjunto. Em seguida, delegar essa nova coleção à segunda entidade, repetindo esse processo para todos agentes até que o conjunto de objetivos esteja vazio;
  - atribuir cada objetivo ao agente que tem melhor condição de satisfazê-lo, ou seja, com um menor custo;

- atribuir um número balanceado de objetivos, ou seja,  $\frac{|Objetivos|}{|agentes|}$  a cada agente respeitando a estratégia anterior.
2. O primeiro agente inicia seu processo de planejamento individualizado e caso não consiga resolver o problema completamente, ele simplesmente repassa um plano vazio ao próximo agente seguindo uma estrutura em anel. Dessa forma, o processo de planejamento pode ser repetido mais de uma vez por cada entidade e o ciclo é mantido até que todos objetivos tenham sido satisfeitos;
  3. Caso o primeiro agente tenha resolvido o seu problema, ele mascara suas informações para manter a privacidade sobre elas uma vez que essa solução é repassada aos demais elementos;
  4. O primeiro agente comunica seu plano ao próximo nó da rede, caracterizando a etapa de comunicação;
  5. O agente que recebeu o plano adiciona suas ações com o intuito de atingir seus objetivos;
  6. Caso o último agente do anel não tenha encontrado a solução, ele repassa o plano parcial ao próximo agente, ou seja, aquele que iniciou a interação. Caso um plano tenha sido encontrado, o processo chega ao fim. Para evitar um ciclo infinito, são adicionadas limites de consumo de tempo e de memória para indicar situações de falha.

Para promover a avaliação da proposta, os resultados obtidos foram comparados e apresentaram melhor desempenho do que [31].

### 3.1.6 Transforming Multi-Agent Planning Into Single-Agent Planning Using Best-cost Strategy

A transformação de um problema MAP em vários problemas SAP é uma estratégia comum e explorada em diversos trabalhos [5, 6, 9, 32]. Essa transformação é realizada através da delegação de subconjuntos de objetivos a diferentes agentes, tornando cada um deles responsável pela solução de problemas menores e possivelmente mais simples de serem resolvidos. Entretanto, o ganho que essa transformação traz é pouco ou raramente discutida na literatura.

Como uma exceção a essa regra, Moreira and Ralha [7] apresentam uma discussão sobre a economia de recursos que pode ser alcançada quando os objetivos são distribuídos entre os agentes disponíveis. O trabalho propõe uma divisão dos objetivos considerando o custo total do plano que, por sua vez, é calculado pela soma das ações que o compõem. Por fim, a configuração que apresentar a menor soma final, isto é, a soma dos custos dos planos de cada agente, é definida como vencedora.

Apesar de realizar avaliação apenas com um domínio fracamente acoplado e com agentes homogêneos em capacidades, da mesma maneira que realizado em Zhang et al. [26], os resultados apresentados justificam a importância de explorar a estratégia de delegação de objetivos. As métricas de desempenho consideradas foram custo total do plano, tempo de planejamento e uso de memória pelo planejador usado, no caso *Graphplan*[33]. As melhoras obtidas foram de 75.66%, 96.64% e 84.34%, respectivamente.

## 3.2 Taxonomia

A análise dos trabalhos listados nessa revisão de literatura, possibilitou a classificação das abordagens de MAP em relação às características presentes ou desejáveis em cada uma delas.

A primeira característica (C1) trata dos arquivos de entrada que definem o domínio e o problema de planejamento. A maioria dos trabalhos utiliza linguagens padrões porém FMAP [8] baseia-se em uma extensão de PDDL, o que ratifica a necessidade de uma preocupação com a flexibilidade do formato de entrada. Uma possível solução para essa questão seria a adoção de uma linguagem pré-definida que seria gerada após um processo de tradução dos arquivos da forma que é realizada em MADLA [10]. Uma vez que as entradas sejam traduzidas, faz-se necessária também a presença de um processador (*parser*) capaz de interpretar os novos arquivos gerados, identificando a declaração dos elementos tais como estado inicial, ações e objetivos, e em seguida instanciando os respectivos objetos.

Uma observação que merece destaque é a inexistência de abordagens que buscam avaliar se um problema pode ser resolvido antes de iniciar o processo de planejamento. Apesar disso, a comunidade científica já apontou a necessidade de pesquisas nessa direção, fato comprovado pela criação do desafio lançado pela *International Planning Competition (IPC)*<sup>1</sup>. Outro destaque é baixa exploração de técnicas de execução paralela nos trabalhos apresentados. Nesse sentido, a segunda característica (C2), desejada pois não foi identificada nessa revisão, é a verificar a possibilidade de solução do problema respeitando as condições impostas. De uma forma geral, um problema pode ser impossível de ser resolvido caso os agentes disponíveis não possuam ações e capacidades que reflitam as necessidades do desafio.

A terceira característica (C3) é uma consequência direta da anterior e está relacionado com a verificação da necessidade de cooperação entre os agentes. Muitas vezes, o domínio descreve uma situação na qual individualmente o problema é impossível de ser resolvido, entretanto de maneira conjunta ele passa a ser possível. Portanto, a análise de cooperação, discutida em Zhang et al. [26], e a presença de ações conjuntas, estudada em MAPJA [30], configuram uma classificação importante para os trabalhos.

Com raras exceções, como por exemplo MAPJA [30] e Zhang et al. [26], os modelos de planejamento utilizam todos os agentes disponíveis. Se por um lado, a alocação de todos os recursos tende a maximizar a possibilidade de encontrar uma solução, por outro lado, isso claramente viola a questão do uso eficiente de recursos, além de influenciar diretamente na escalabilidade do modelo, como mostrado em FMAP [8]. Portanto, a quarta característica (C4) identificada é a seleção dos agentes que serão envolvidos no processo de planejamento.

Uma abordagem comum para resolver problemas MAP é a sua transformação em problemas menores, possivelmente mais simples e capazes de serem resolvidos por um único agente. Nesse sentido, MAPR [5] explora diferentes estratégias de delegação de objetivos entre os agentes, enquanto Moreira and Ralha [7] destacam os ganhos obtidos com essa técnica. Consequentemente, a transformação do problema original é a quinta característica (C5) identificada nessa revisão.

---

<sup>1</sup><http://unsolve-ipc.eng.unimelb.edu.au>

A maneira pela qual o planejamento é conduzido, isto é, a coordenação, é sexta característica (C6) importante para classificar os trabalhos devido ao fato de alguns seguirem abordagens distintas, isto é, centralizada ou distribuída. Se por um lado a primeira opção pode ser mais simples, apesar de apresentar um ponto único de falha, a segunda pode ser mais robusta ao custo do prejuízo da escalabilidade devido a um elevado número de mensagens trocadas.

A questão da manutenção da privacidade das informações entre os agentes é sétima característica (C7) proposta na taxonomia.

Finalmente, a taxonomia proposta é composta pelos termos e valores apresentados na Tabela 3.1. Os termos *não*, de maneira especial, indicam tanto a ausência explícita da característica quanto a falta de citação no trabalho. Em relação às combinações possíveis dos termos, na Figura 3.2 é ressaltado que as características podem existir independente uma da outra, ou seja, a classificação anterior não influencia na seguinte. As características referentes ao teste de solução e análise de cooperação são apresentados justapostos pois fazem parte de uma mesma fase de verificação. Do exposto, cabe destacar que o ordenamento vertical e agrupamento apresentados representam a sequência de classificação de um trabalho de planejamento multiagente.

Tabela 3.1: Taxonomia para trabalhos de MAP.

Característica	Termo	Valores possíveis
C1	Flexibilidade de entrada	Sim ou Não
C2	Teste de solução	Sim ou Não
C3	Análise de cooperação	Sim ou Não
C4	Seleção de agentes	Sim ou Não
C5	Transformação do problema original	Sim ou Não
C6	Coordenação	Centralizada ou Distribuída
C7	Privacidade	Preservada, Flexível ou Ignorada

A taxonomia proposta pode ser usada também como um *framework* para criação de modelos de planejamento multiagente. Sendo assim, o ordenamento passa a identificar as etapas a serem executadas, começando pelo tratamento dos arquivos de entrada (C1) e passando por uma verificação do problema (C2 e C3, preferencialmente em paralelo). A seleção dos agentes (C4) só é viável caso exista a possibilidade de solução e, por sua vez, a transformação do problema (C5) dependerá das entidades escolhidas. Por fim, as características de coordenação (C6) e privacidade (C7) dependerão do algoritmo/modelo proposto.

Para resumir os trabalhos identificados na revisão sistemática da literatura e detalhados nesse capítulo, a Tabela 3.2 sintetiza e classifica-os de acordo com as características da taxonomia proposta.

A principal conclusão que pode ser obtida da análise da classificação dos trabalhos seguindo a taxonomia proposta é que não há preocupação com a flexibilidade de entrada (C1) nem com a verificação do problema (C2 e C3) antes do início do processo de planejamento. Enquanto a seleção de agentes (C4) está presente em dois trabalhos estudados, as três últimas características (C5, C6 e C7) são abordadas, mesmo que de formas distintas, na maioria dos trabalhos, exceção feita apenas em relação à transformação de problemas (C5) para os trabalhos de Zhang et al. [26] e FMAP [8].

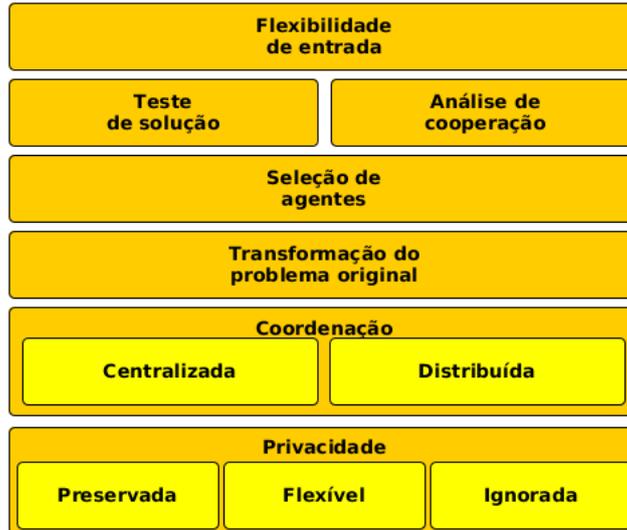


Figura 3.2: Taxonomia para MAP.

Tabela 3.2: Trabalhos classificados de acordo com a taxonomia proposta.

Trabalho	Características						
	C1	C2	C3	C4	C5	C6	C7
Zhang et al. [26]	-	-	✓	✓	-	Centralizada	Ignorada
FMAP (Torreno et al. [8])	-	-	-	-	-	Distribuída	Preservada
MAPJA (Chouhan and Niyogi [30])	-	-	-	✓	✓	Centralizada	Ignorada
MADLA (Štolba and Komenda [10])	✓	-	-	-	✓	Centralizada	Preservada
MAPR (Borrajo [5])	-	-	-	-	✓	Distribuída	Preservada
Moreira and Ralha [7]	-	-	-	-	✓	Centralizada	Ignorada

Portanto, o modelo de planejamento multiagente proposto nessa dissertação e apresentado no Capítulo 4 tem como objetivo preencher as lacunas referentes tanto à verificação do problema a ser executada em uma fase anterior a de planejamento (C2 e C3), quanto à seleção de agentes (C4). Contudo, para respeitar o escopo e o prazo disponíveis para um trabalho de mestrado, o tratamento das entradas visando flexibilizar o uso de linguagens não é abordado nessa dissertação e é sugerido como o prosseguimento da pesquisa.

# Capítulo 4

## Modelo proposto

Um método de coordenação visa facilitar a etapa de planejamento, mas a tentativa de construir uma abordagem independente de domínio esbarra nos problemas de escalabilidade e controle. Em domínios fortemente acoplados os agentes explicitamente necessitam cooperar para atingir os objetivos, uma vez que uma entidade não possui todas as capacidades requeridas para executar a tarefa. Por outro lado, em domínios fracamente acoplados, a coordenação pode ser flexibilizada uma vez que um único agente pode satisfazer sozinho todos ou um subconjunto dos objetivos.

Uma contribuição para a área de planejamento multiagente identificada na revisão de literatura (vide Capítulo 3) é a combinação das características de verificação e transformação do problema. As análises da possibilidade de atingir os objetivos (teste de solução C2) e da necessidade de cooperação entre os agentes (análise de cooperação C3), quando realizadas em uma fase anterior ao processo de exploração do espaço de busca, pode apresentar uma diminuição do tempo de planejamento.

Uma vez que as condições necessárias para a solução do problema tenham sido identificadas e respeitem as restrições impostas, a transformação do MAP depende da seleção dos agentes (C4) e da alocação dos objetivos (C5). Considerando especialmente a atividade de alocação, sua utilização permite explorar técnicas de execução em paralelo, instanciando processos mais simples de planejamento para cada agente, considerando o fato que as plataformas computacionais possuem recursos limitados. Uma vez que as entidades devem analisar um problema menos complexo, com um conjunto menor de ações, restrições e efeitos, o tempo gasto para calcular um plano final ou parcial tende a ser menor. Em outras palavras, sistemas multiagentes fornecem a infraestrutura necessária para execução paralela.

Portanto, a efetividade de um modelo de planejamento multiagente depende do desenvolvimento de métodos capazes de:

- distribuir eficientemente o trabalho entre as entidades disponíveis;
- garantir as condições de cooperação quando necessárias;
- coordenar as atividades utilizando uma quantidade de mensagens que permita a escalabilidade do modelo.

O modelo proposto visa tanto explorar as lacunas identificadas nos trabalhos correlatos quanto configurar uma contribuição e avanço para a área de planejamento multiagente.

A Tabela 4.1, apresenta a comparação das características do modelo com os trabalhos identificados. A flexibilização da entrada para permitir o uso de diversas linguagens para definição de domínio e problema não é tratada nesse trabalho para ajustar o escopo ao tempo disponível.

Tabela 4.1: Contribuição e características do modelo proposto.

Trabalho	Características						
	C1	C2	C3	C4	C5	C6	C7
Zhang et al. [26]	-	-	✓	✓	-	Centralizada	Ignorada
FMAP (Torreno et al. [8])	-	-	-	-	-	Distribuída	Preservada
MAPJA (Chouhan and Niyogi [30])	-	-	-	✓	✓	Centralizada	Ignorada
MADLA (Štolba and Komenda [10])	✓	-	-	-	✓	Centralizada	Preservada
MAPR (Borrajo [5])	-	-	-	-	✓	Distribuída	Preservada
Moreira and Ralha [7]	-	-	-	-	✓	Centralizada	Ignorada
LCMAP	-	✓	✓	✓	✓	Centralizada	<i>Flexível</i>

## 4.1 Visão Geral

O modelo proposto baseia-se na transformação do problema de planejamento multiagente em um conjunto de instâncias mais simples. Conforme apresentado nas Figuras 2.4 e 2.5, os diversos agentes podem ter objetivos alocados conforme suas capacidades. Por exemplo, no domínio *rovers*, caso nenhum agente tenha a capacidade de fotografar, esse objetivo seria impossível de ser satisfeito. Da mesma, a ausência de agente do tipo avião no domínio *logistics* inviabilizaria o transporte intermunicipal.

Embora essa técnica seja explorada em outros trabalhos (MAPR [5],[9], [34]), a inovação que esta proposta apresenta é o fato de verificar a possibilidade de cada agente satisfazer os objetivos antes de dividi-los. Dessa forma, quando for possível, os agentes planejam e executam suas atividades de maneira independente, sob a coordenação de um agente central e responsável por alocar os objetivos e validar os planos calculados. Na impossibilidade da distribuição das tarefas, esse agente coordenador realiza o planejamento de forma centralizada com o intuito de diminuir o número de interações que seriam necessárias caso se assumisse uma estratégia distribuída para solução do problema.

Para facilitar o entendimento das características do modelo, a Figura 4.1 apresenta a interação (a numeração nas conexões entre os elementos indica a ordem cronológica) entre os agentes e as etapas executadas desde a definição do problema de planejamento até a apresentação do plano, quando existir. Os detalhes de cada componente ilustrado nesse instante serão apenas introduzidos para tornar viável uma noção geral da solução proposta e portanto serão detalhados na Seção 4.2.

Como uma das características do modelo é dividir as responsabilidades entre os agentes, sempre que isso for possível, cada agente deve analisar suas capacidades no intuito de descobrir quais objetivos ele é capaz de atingir de forma isolada, sem depender ou interagir com outros agentes. Dessa forma, uma premissa básica que foi definida no desenvolvimento do modelo é que um agente (planejador/executor) não deveria ter conhecimento da existência de outros agentes. Ao final dessa fase de verificação (conexão 2 da Figura 4.1), cada agente tem bem definidos os seguintes itens:

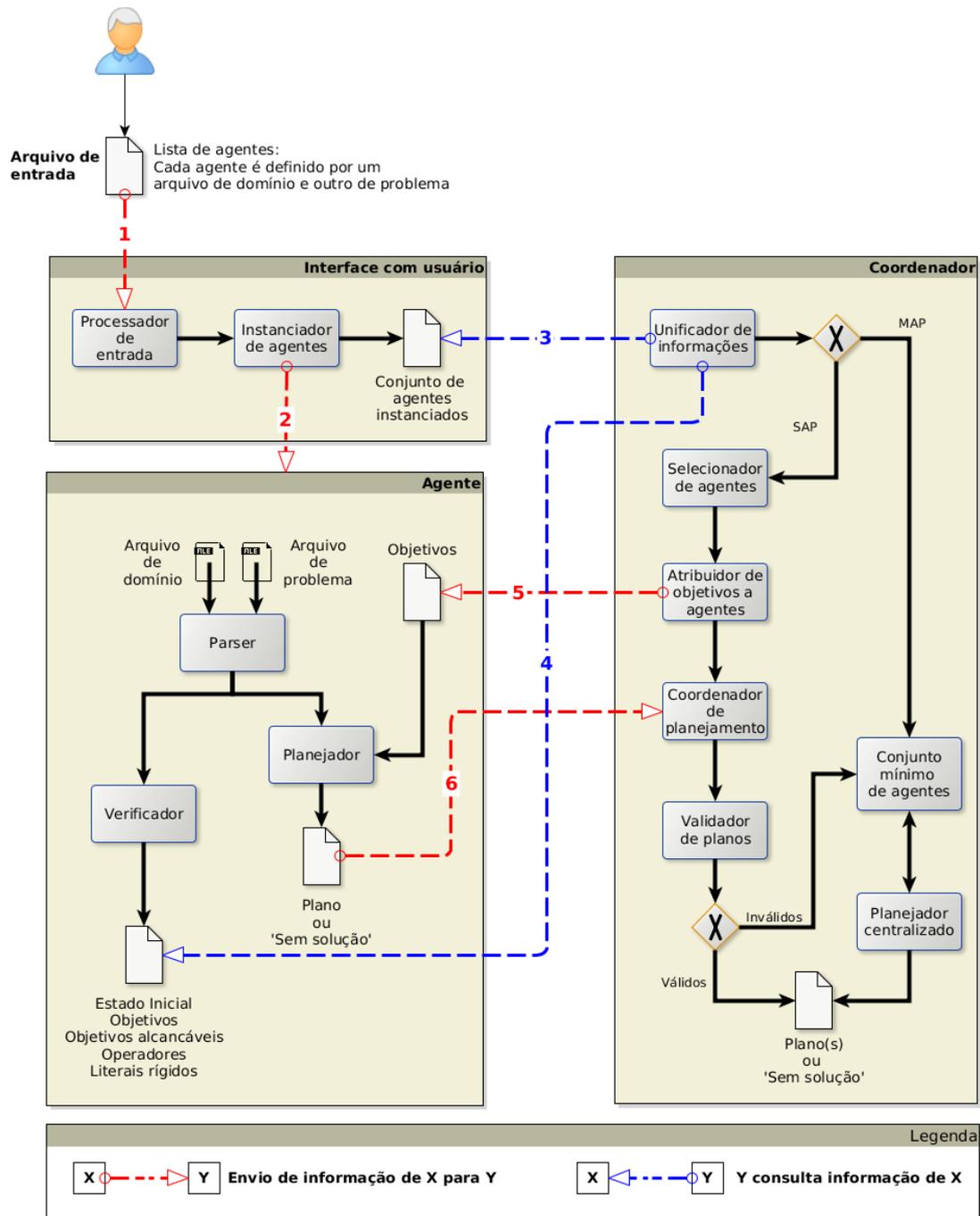


Figura 4.1: Interação entre agentes.

- Estado inicial ( $I_{ag}$  da Tupla II descrita na Definição 15);
- Objetivos ( $G_{ag}$  da Tupla II descrita na Definição 15);
- Objetivos que são alcançáveis individualmente, isto é, sem a necessidade de cooperação com outros agentes na fase de execução;
- Conjunto de operadores ( $O_{ag}$  da Tupla II descrita na Definição 15);
- Predicados e variáveis de estado rígidas (Definição 4).

Em seguida, o coordenador consulta todos os agentes e unifica (conexões 3 e 4 da Figura 4.1) as informações para então formar a Tupla II. A partir dessas informações o coordenador irá selecionar quais agentes serão empregados no processo planejamento.

Com os possíveis planejadores já selecionados, o coordenador inicia o processo de atribuição de objetivos seguindo duas estratégias que são definidas pelo usuário. Na primeira delas, o coordenador adota a técnica de balanceamento de carga, ou seja, cada agente receberá uma quantidade semelhante de objetivos. No modelo proposto, o conceito de carga desconsidera o tamanho<sup>1</sup> do plano gerado para satisfazer os objetivos, assumindo como hipótese que quanto maior é o número de objetivos que o agente deve satisfazer, maior será a quantidade de ações que ele deverá executar. A outra opção é a de economia de recursos na qual o coordenador buscará utilizar o menor número de agentes para atingir os objetivos.

Assim que todos objetivos tenham sido atribuídos aos seus responsáveis (conexão 5 Figura 4.1), cada agente iniciará o processo de planejamento de maneira individual e paralela.

O coordenador então valida os planos (conexão 6 da Figura 4.1), verificando se não existem concorrências de recursos e conseqüentemente os agentes podem executar suas ações de maneira totalmente coordenada, sem nenhum tipo de competição ou cooperação.

O processo de planejamento é finalizado quando o coordenador apresenta o plano final, seja fruto da união dos planos calculados de maneira independente e validados ou da resposta da execução centralizada, ou a mensagem que não existe solução para o problema nas condições definidas.

Por fim, retomando a questão da privacidade das informações, o modelo apresenta uma propriedade flexível. Por um lado, como os agentes planejadores/executores não dependem do conhecimento de outros agentes e caso o problema inicial seja totalmente transformado em instâncias individuais, a privacidade é considerada totalmente preservada, considerando como aceitável e necessário o fato do coordenador ser o único elemento que conhece informações de outros agentes, no caso o conjunto de objetivos alcançáveis. Cabe ressaltar que nessas condições, cada agente planejador desconhece tanto a existência de outros elementos do mesmo tipo quanto o estado do processo de planejamento executado por eles. Por outro lado, caso o planejamento seja realizado de forma centralizada, o coordenador utilizará também o conjunto de operadores dos agentes selecionados, possuindo assim um nível maior de informação do que a condição anterior. Dessa forma, mesmo que os demais agentes não compartilhem esse conhecimento entre si, a privacidade não é considerada preservada, entretanto tão pouco é totalmente ignorada. Por conseguinte, o modelo é considerado flexível.

---

<sup>1</sup>O tamanho de um plano  $\pi = \langle a_1, a_2, \dots, a_n \rangle$  é igual ao número de operadores, ou seja,  $n$ .

No prosseguimento do trabalho o modelo proposto será identificado como *Lightweight Coordination Multiagent Planning (MAP)* por utilizar um processo de coordenação leve, no qual as interações entre os agentes ocorrem em instantes bem definidos e não a cada iteração da fase de planejamento.

## 4.2 Modelo Arquitetural

O modelo proposto é composto por três camadas sendo que os agentes, tanto o coordenador quanto os executores/planejadores, são agrupados na camada central na qual toda coordenação e planejamento são realizados. Como foi apresentado na Seção 4.1, o usuário define as características dos agentes através de um arquivo de entrada que é recebido e processado pela *camada de tratamento da entrada*. Na sequência, os agentes instanciados realizam as verificações de suas capacidades, informam ao coordenador e, se selecionados, informam os planos calculados. Caso as respostas sejam inválidas, um novo planejamento é realizado centralizadamente finalizando as atividades da *camada de coordenação e planejamento*. Por fim, informações como planos, agentes empregados, número de ações e o tempo total de gasto para o planejamento são apresentados ao usuário pela *camada de apresentação dos planos*.

Na Figura 4.2 são ilustradas as três camadas: tratamento de entrada, coordenação e planejamento, apresentação dos planos. Também na Figura 4.2 constam os componentes e as interações entre os agentes. As características dessas partes, bem como dos algoritmos empregados para execução de cada etapa são detalhados na sequência. Toda implementação foi realizada utilizando a linguagem de programação *Python*<sup>2</sup>

No modelo proposto, os agentes podem ser classificados como coordenador ou planejador/executor. No segundo caso, o papel planejador origina-se do fato que o mesmo pode ser incumbido de buscar uma solução para um subconjunto dos objetivos. Já o papel executor advém da necessidade do agente executar uma sequência de ações que ele mesmo calculou ou que foi determinada pelo coordenador.

No modelo proposto, os agentes, com exceção ao coordenador, são caracterizados por três componentes principais:

- arquivo que descreve o domínio;
- arquivo que descreve o problema;
- planejador automatizado.

A descrição dos dois primeiros componentes (arquivo de domínio e de problema) segue o padrão *Planning Domain Definition Language (PDDL2.1)*. Com relação ao terceiro componente, o uso de planejador automatizado permite ao agente deliberar sobre suas ações, garantindo-lhe autonomia e a capacidade de determinar uma sequência de ações que alcancem os objetivos a partir do estado inicial.

A proposta segue algumas premissas do planejamento clássico. O conjunto de estados do ambiente é finito, totalmente observável e estático. Os efeitos das ações são determinísticos e instantâneos. O processo de planejamento é realizado *offline*.

---

<sup>2</sup><https://www.python.org/>

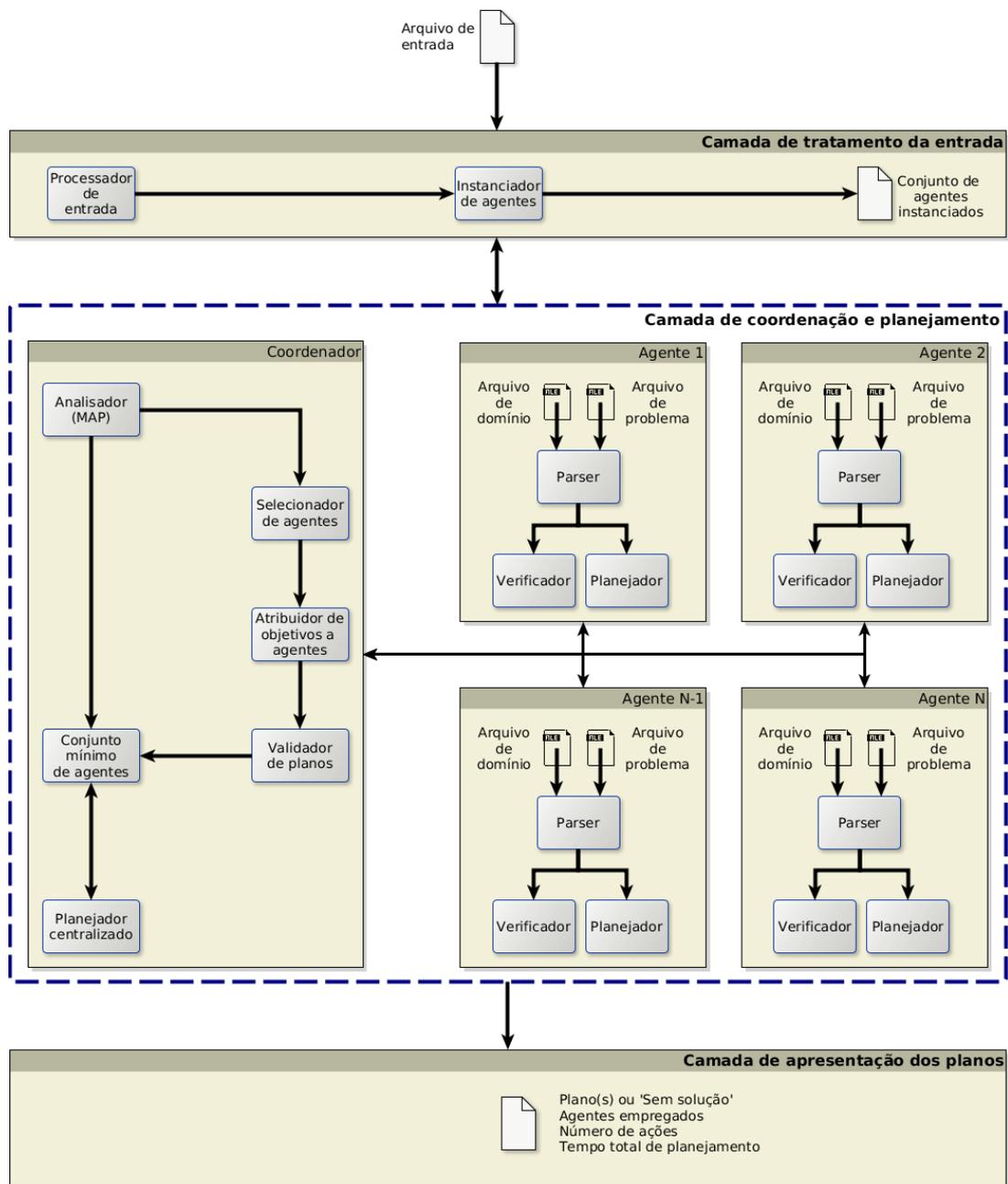


Figura 4.2: Modelo conceitual.

Embora a linguagem PDDL2.1 suporte a disjunção de objetivos, o modelo proposto permite apenas a conjunção deles. Portanto, a ausência de anotações que caracterizem a condição dos objetivos (individual ou global) não causa nenhum prejuízo pois todos eles devem ser satisfeitos simultaneamente por um único estado.

### 4.2.1 Camada de tratamento da entrada

A função dessa camada é receber o arquivo de entrada e a partir dele instanciar os agentes. Os elementos desse arquivo são definidos pelo usuário e o padrão utilizado é o *JavaScript Object Notation (JSON)*<sup>3</sup>, por ser um arquivo texto de simples escrita e leitura e independente de linguagem de programação.

A primeira a ser executada na utilização do modelo é o fornecimento das condições de contorno. Para tanto, a entrada do modelo é definida por um arquivo de entrada no qual são listados os elementos do SMA. Nesse arquivo, cada agente é definido por três elementos:

- *agId*: elemento textual que identifica do agente;
- *domainPath*: arquivo PDDL de domínio;
- *problemPath*: arquivo PDDL de problema;
- *planejador*: planejador automatizado utilizado pelo agente;

Os Códigos 4.1 e 4.2 ilustram exemplos de arquivos de entradas compostos por três agentes que ilustram os problemas de planejamento *rovers* e *logistics*, respectivamente. Ainda, os agentes descritos no arquivo de entrada são instanciados de maneira paralela, visto que nas operações de criação são independentes; portanto, tratá-las de maneira sequencial caracteriza um desperdício de recursos computacionais.

---

<sup>3</sup><http://www.json.org/json-pt.html>

Código 4.1: Arquivo de entrada - logistics. Código 4.2: Arquivo de entrada - logistics.

```

{
  "agentes": [
    {
      "agId": "rover0",
      "domainPath": "domainRovers.pddl",
      "problemPath": "p0.pddl",
      "planner": "ff"
    },
    {
      "agId": "rover1",
      "domainPath": "domainRovers.pddl",
      "problemPath": "p1.pddl",
      "planner": "ff"
    },
    {
      "agId": "rover2",
      "domainPath": "domainRovers.pddl",
      "problemPath": "p2.pddl",
      "planner": "ff"
    }
  ]
}

```

```

{
  "agentes": [
    {
      "agId": "plane1",
      "domainPath": "domainPlane.pddl",
      "problemPath": "plane1.pddl",
      "planner": "ff"
    },
    {
      "agId": "truck1",
      "domainPath": "domainTruck.pddl",
      "problemPath": "truck1.pddl",
      "planner": "ff"
    },
    {
      "agId": "truck2",
      "domainPath": "domainTruck.pddl",
      "problemPath": "truck2.pddl",
      "planner": "ff"
    }
  ]
}

```

Nessa camada, as atividades realizadas são baseadas em componentes responsáveis por processar a entrada e por instanciar os agentes. No primeiro deles, cada objeto, que representa um agente, tem seus atributos identificados através do uso da biblioteca *json*<sup>4</sup> da linguagem Python. A partir desse ponto, os agentes são instanciados em processos independentes e de forma paralela utilizando a biblioteca *multiprocessing*<sup>5</sup> a fim de aproveitar todos processadores disponíveis. Após todos elementos terem sido instanciados e suas capacidades terem sido analisadas, as informações são reagrupadas (*join*) e então a camada disponibiliza o conjunto de agentes instanciados que será utilizado pela próxima camada. A Figura 4.3 ilustra o processo de criação dos agentes de forma paralela

Apesar dos agentes do tipo executor/planejador estarem apresentados na camada de coordenação e planejamento, seus componentes *Parser* e *Verificador* são utilizados durante a fase de instanciação dos agentes, portanto é interessante que eles sejam detalhados ainda nessa seção.

## Parser

Os arquivos de domínio e problema que definem o agente são processados pelo *parser* implementado pela biblioteca *pddlpy*<sup>6</sup>. Essa ferramenta oferece uma interface simples para interagir com problemas de planejamento por possuir métodos para obtenção de informações como: estado inicial, objetivos, conjunto de modelos de ações, conjunto de pré-condições e efeitos dos operadores.

No procedimento de instanciação dos agentes, seus arquivos PDDL são processados e todos possíveis predicados e variáveis de estado são calculados considerando as definições do domínio e do problema. Cabe ressaltar que essa característica aprimora os *parsers*

<sup>4</sup><https://docs.python.org/3/library/json.html>

<sup>5</sup><https://docs.python.org/3.6/library/multiprocessing.html>

<sup>6</sup><https://pypi.python.org/pypi/pddlpy/0.1.9>

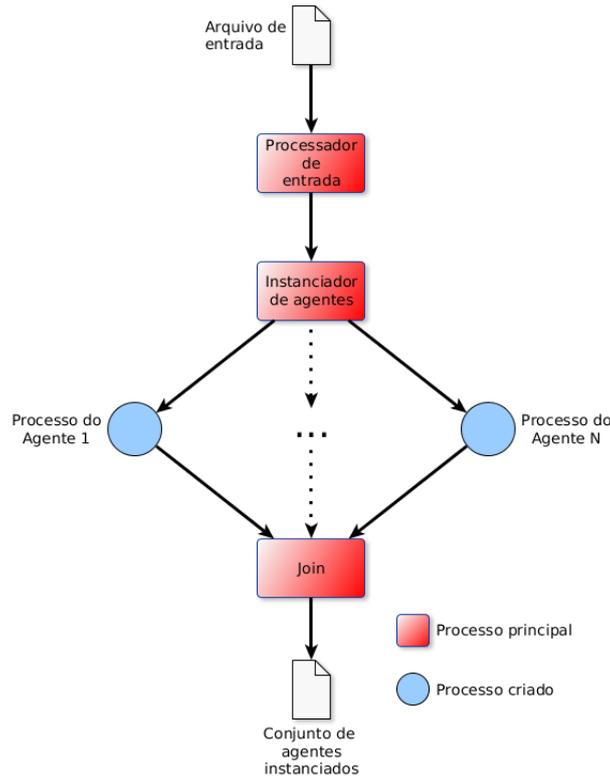


Figura 4.3: Paralelização do processo de instanciação dos agentes.

comuns. Ambos elementos são gerados usando o produto cartesiano dos domínios de cada variável definida.

Apesar de ser facilmente codificado, essa estratégia implica em alguns efeitos colaterais porém nenhum deles prejudica a corretude do modelo. O cálculo do produto cartesiano pode definir estados que não serão atingidos durante o planejamento dependendo das condições de contorno do problema. Esses estados inalcançáveis não causam erros ou soluções incorretas, entretanto, eles demandam uma avaliação extra, afetando o tempo de resposta. O modelo proposto contorna essa questão através do uso de técnicas de execução paralela.

O Algoritmo 3 ilustra as etapas executadas pelo *parser*. Inicialmente, a partir dos arquivos de domínio e problema, o estado inicial, objetivos, objetos e modelos de ações são definidos nas Linhas 1, 2 e 3, respectivamente. O primeiro passo para geração do produto cartesiano é a definição dos objetos e tipos definidos no problema de planejamento. Essas informações são agrupadas, na Linha 4, na forma de um dicionário<sup>7</sup>, cuja chave é o tipo e o valor é um conjunto dos objetos desse mesmo tipo.

O laço da Linha 5 é responsável por iniciar o processo de instanciação dos operadores a partir dos modelos de ação. Para isso, o conjunto de parâmetros ( $def(\alpha)$  da Definição 6) de cada ação existente é consultado e organizado em pares (*variavel, tipo*). Dessa forma, os domínios de cada variável são facilmente calculados a partir do dicionário previamente obtido. Todas as combinações possíveis são geradas a partir do produto cartesiano do

<sup>7</sup>Estrutura de dados que armazena pares (chave, valor) chamados itens.

domínio das variáveis, exemplificado a seguir:

$$D(v_1) = \{A, B, C\}$$

$$D(v_2) = \{D, E\}$$

$$D(v_3) = \{F\}$$

$$D(v_1) \times D(v_2) \times D(v_3) = \{(A, D, F), (A, E, F), (B, D, F), (B, E, F), (C, D, F), (C, E, F)\}$$

Essas combinações são então interpretadas pelo, laço da Linha 9, instanciando os operadores possíveis. A função  $ground(configuracao)$ , descrita na Linha 10, é responsável por substituir todas as referências às variáveis presentes nos conjuntos  $pre(\alpha)$  e  $eff(\alpha)$  pelos respectivos valores presentes na combinação em vigor. Ao final do processamento o conjunto de atributos de cada agente está disponível, isto é, estado inicial, objetivos e operadores.

---

### Algoritmo 3 Parser

---

**Entrada:** Arquivos de domínio e problema

**Saída:** Conjunto de atributos do agente

- 1:  $I_{agId} \leftarrow$  estado inicial do agente  $agId$
  - 2:  $G_{agId} \leftarrow$  objetivos do agente  $agId$
  - 3:  $Acoes_{agId} \leftarrow$  modelos de ações do agente  $agId$
  - 4:  $Obj_{s_{agId}} \leftarrow$  objetos e tipos do agente  $agId$  ▷ dicionário[tipo]={objetos}
  - 5: **for**  $acao \in Acoes$  **do**
  - 6:   **for**  $variavel, tipo \in parametros(acao)$  **do**
  - 7:      $Dominio(variavel) \leftarrow Obj_{s_{agId}}[tipo]$
  - 8:    $configuroes \leftarrow Dominio(v_1) \times \dots \times Dominio(v_n)$  ▷ Produto cartesiano
  - 9:   **for**  $configuracao \in configuracoes$  **do**
  - 10:     $O_{agId} \leftarrow O_{agId} \cup \{ground(configuracao)\}$  ▷ Conjunto de operadores
- 

Em virtude da exploração do paralelismo dos processos de instanciação dos agentes, a geração do conjunto dos operadores por cada agente é facilitado devido ao fato da análise ser realizada a partir dos modelos de ação de um único elemento, ao invés de considerar o conjunto de todos modelos. Essa característica é um exemplo do ganho obtido pela transformação do problema original em instâncias mais simples, no caso, menores.

### Verificador

O centro da fase de verificação é representado pela análise da possibilidade de cada agente satisfazer o conjunto de objetivos. Inicialmente, cada agente avalia sua capacidade de atingir individualmente os objetivos. Para isso, a heurística *delete-relaxation* é aplicada [1].

Basicamente, essa técnica explora o conjunto de ações disponíveis de um único agente, adicionando novos efeitos e nunca removendo os antigos. Por definição, ela retorna um valor que representa o custo para atingir um estado a partir de outro, entretanto nesse modelo o retorno é restrito aos valores *Verdadeiro*, quando é possível obter sucesso, e *Falso*, caso contrário.

O Algoritmo 4 ilustra como tal heurística foi implementada. Apesar de poder ser usado para verificar a possibilidade de atingir um estado a partir de outro, o algoritmo mantém fixa a origem,  $I_{agId}$ . Os operadores aplicáveis no estado inicial são então identificados (Linha 5) e o estado resultante dessa primeira iteração ( $k \leftarrow 1$ ) é igual a união dos estados relaxados gerados pela execução de todos esses operadores (Linha 7). O algoritmo é encerrado quando esse estado for exatamente igual ao anterior (Linha 8), o que significa que nenhum novo efeito foi adicionado, ou quando os objetivos forem satisfeitos, evitando assim uma nova iteração do laço (Linha 4).

---

**Algoritmo 4** Verificação (heurística *delete-relaxation* adaptada de [1])

---

**Entrada:**  $O_{agId}, I_{agId}, G$

**Saída:** *True* ou *False*

```

1:  $\hat{s}_0 \leftarrow I_{agId}$ 
2:  $A_0 \leftarrow \emptyset$ 
3:  $k \leftarrow 1$ 
4: while  $\nexists s \subseteq s_k$  que satisfaça  $G$  do
5:    $A_k \leftarrow \{\text{todos operadores aplicáveis em } \hat{s}_{k-1}\}$ 
6:   for  $a \in A_k$  do
7:      $\hat{s}_k \leftarrow \hat{s}_k \cup \{\gamma^+(\hat{s}_{k-1}, a)\}$ 
8:   if  $\hat{s} = \hat{s}_{k-1}$  then
9:     return False
10:   $k \leftarrow k + 1$ 
11: return True

```

---

Apesar do fato dessa técnica transformar o problema original em uma instância menos restrita, ou mais relaxada, seu uso é capaz de fornecer informações importantes sobre a impossibilidade do agente atingir os objetivos. Em outras palavras, se um objetivo for inalcançável sob condições relaxadas, certamente ele também será em um contexto mais restrito. Portanto, esse teste verifica se é possível que exista uma sequência de ações capaz de atingir um estado que satisfaça todos objetivos.

Um ponto importante a ser destacado é que o uso dessa heurística é destinado à análise da impossibilidade do problema ser solucionado. Nesse sentido, existe uma ligeira diferença entre ser possível e não ser impossível. O primeiro caso só é concretizado quando um plano é apresentado e isso é apenas realizado ao final da fase de planejamento. O foco da fase de verificação é, portanto, verificar se as condições necessárias para a solução estão disponíveis, ou seja, a presença de todas capacidades e objetos dos quais os objetivos dependem.

O Algoritmo 4 é utilizado pelo agente para identificar quais objetivos ele pode satisfazer individualmente. Através dele, o conjunto de metas é verificado de maneira total, isto é, se é possível atingir um estado no qual todos os literais descritos em  $G_{agId}$  estão presentes. Sendo isso impossível, o agente verifica quais átomos ele é capaz de atingir e para isso ele testa um a um. O Algoritmo 5 ilustra esse roteiro de verificação no qual o cálculo da heurística é representado pela função *hdr*.

Outra função importante da fase de verificação é a definição de quais predicados e variáveis de estado apresentam a propriedade de rigidez (Definição 4). Para tanto, o conjuntos de pre-condições e efeitos de cada modelo de ação são analisados com o intuito

---

**Algoritmo 5** Determinação do conjunto de objetivos alcançáveis

---

**Entrada:**  $O_{agId}, I_{agId}, G_{agId}$ **Saída:** Conjunto de objetivos alcançáveis

```
1:  $Objetivos \leftarrow \emptyset$ 
2: if  $hdr(O_{agId}, I_{agId}, G_{agId}) = True$  then
3:    $Objetivos \leftarrow G_{agId}$ 
4: else
5:   for  $g \in G_{agId}$  do
6:     if  $hdr(O_{agId}, I_{agId}, \{g\}) = True$  then
7:        $Objetivos \leftarrow Objetivos \cup \{g\}$ 
8: return  $Objetivos$ 
```

---

de identificar elementos que não são modificados. Da mesma forma que a verificação das capacidades é realizada em um cenário mais relaxado, a determinação dos elementos rígidos é feita de maneira a priorizar o desempenho, porém sem afetar a corretude.

O Algoritmo 6 ilustra como essa verificação é feita. Tal relaxamento é criado dado que a entrada do algoritmo é o conjunto de modelos de ações e não o de operadores. Por definição, o primeiro conjunto é menor posto que é formado por literais do tipo *unground*, cabendo ao segundo item comportar todas as combinações do tipo *ground*. Dessa forma, o laço da Linha 4 considera apenas os elementos do primeiro conjunto, o que garante uma diminuição do espaço de análise. Os laços das Linhas 5-6 e 7-8 recuperam os átomos descritos nas pré-condições e efeitos, respectivamente, porém armazenando apenas os identificadores dos literais. O outro ponto que garante o relaxamento é justamente o fato de avaliar apenas esses identificadores ao invés dos átomos completos, apresentando como saída apenas a diferença entre os conjuntos (Linha 9). Essa operação garante que todos os literais que não estão no conjunto de efeitos são os que permanecessem inalterados.

Por analisar apenas os identificadores, o algoritmo retorna um conjunto de identificadores que possivelmente configurarão literais rígidos após a substituição das variáveis. A corretude é garantida dado que essa informação é verificada na fase de validação dos planos, na qual todos os átomos são do tipo *ground* dado que compõem operadores.

---

**Algoritmo 6** Determinação do conjunto de literais rígidos

---

**Entrada:** Modelos de ação de  $agId$ **Saída:** Conjunto de predicados e variáveis de estado rígidos

```
1:  $Acoes \leftarrow$  modelos de ação
2:  $Precondicoes \leftarrow \emptyset$ 
3:  $Efeitos \leftarrow \emptyset$ 
4: for  $\alpha \in Acoes$  do
5:   for  $p \in pre(\alpha)$  do
6:      $Precondicoes \leftarrow Precondicoes \cup \{id(p)\}$        $\triangleright id(p)$  é o identificador do literal  $p$ 
7:   for  $e \in eff(\alpha)$  do
8:      $Efeitos \leftarrow Efeitos \cup \{id(e)\}$                  $\triangleright id(e)$  é o identificador do literal  $e$ 
9: return  $Precondicoes \setminus Efeitos$                          $\triangleright$  Diferença entre os conjuntos
```

---

Ao final da verificação individual, cada agente conhece seu estado inicial, objetivos,

operadores, predicados e variáveis de estado rígidos, e quais objetivos tem as condições de atingir individualmente.

## 4.2.2 Camada de coordenação e planejamento

A camada de coordenação e planejamento configura o núcleo do modelo pois é nela que as fases de transformação do problema, através das atividades de seleção de agentes e atribuição de objetivos, e de validação dos planos são realizadas. Além delas, a possibilidade de um processamento centralizado ser desempenhado pelo coordenador aumenta a importância dessa camada.

### Analizador (MAP)

A primeira atividade dessa camada é interação do coordenador com os demais agentes com o intuito de unificar as informações formando a Tupla  $\Pi = \langle Ag, V, O, I, G \rangle$ , descrita na Definição 15. Além disso, ele consulta os agente a fim de conhecer os objetivos alcançáveis por cada agente e literais rígidos.

A partir dessas informações, o coordenador analisa o problema de planejamento verificando se é possível transformá-lo em um conjunto de instancias mais simples (SAP). Nesse intuito, é verificado se a união dos conjuntos das metas atingíveis é igual ao conjunto de objetivos, conforme Equação 4.1.

$$\bigcup_{i=1}^n \text{Objetivos}_{ag_i} = G \quad (4.1)$$

Por definição, os conjuntos, união e  $G$ , só podem ser diferentes se existir algum objetivo que não seja alcançável por nenhum agente. Portanto, a comparação de suas cardinalidades ( $\#$ ) é suficiente para verificar se a transformação é possível.

Se  $\# \bigcup_{i=1}^n \text{Objetivos}_{ag_i} = \#G$ , o coordenador iniciará a etapa de seleção de agentes que indicará os candidatos que participarão do processo de atribuição. Caso contrário, ele assumirá uma postura centralizadora, tentando definir o conjunto mínimo de agentes que serão utilizados no planejamento. Esse teste, apesar de simples, é importante pois evita desperdício de tempo buscando candidatos e uma divisão de tarefas que não resultará na satisfação de todos objetivos.

### Selecionador de agentes

A finalidade desse componente é identificar quais agentes são capazes de planejar e executar suas ações individualmente, sem a necessidade de cooperação de outros elementos. Para tanto, o coordenador analisa os conjuntos de objetivos alcançáveis de todos os candidatos. O elementos selecionados são aqueles que apresentam a possibilidade de satisfazer pelo menos uma meta, ou seja, conjuntos com cardinalidade ( $\#$ ) maiores que 0 conforme apresentado na Equação 4.2.

$$\text{candidatos} = \{ag \in Ag \mid \# \text{Objetivos}_{ag} > 0\} \quad (4.2)$$

Essa seleção é baseada em uma análise qualitativa e não quantitativa. Após a fase de verificação, os agentes analisam quais objetivos eles podem alcançar, entretanto o possível custo associado não é computado pois a heurística (vide Algoritmo 4) foi modificada para retornar apenas valores referentes a sucesso e fracasso. Apesar dessa característica impedir que os agentes sejam ordenados, do mais capacitado para o menos, ela não prejudica a transformação do problema posto que todos elementos escolhidos são capazes de atingir algum objetivo.

### Atribuidor de objetivos a agentes

Uma vez que os agentes que podem planejar isoladamente foram determinados, o coordenador inicia o processo de atribuição de objetivos a esses candidatos a fim de evitar que dois ou mais deles trabalhem em prol da mesma meta.

O modelo pode executar a atribuição de duas maneiras distintas, porém antes de iniciar esse processo o coordenador verificar se é possível transformar o problema. Caso um único objetivo não seja atingido de maneira individual por qualquer agente, a fase de atribuição será finalizada e o planejamento centralizado será executado.

Na primeira opção de estratégia, ele alocará o menor número de agentes para a fase de planejamento, portanto, mesmo que todas entidades sejam capazes de atingir as metas, apenas uma delas será empregada. Nesse sentido, a economia dos recursos é o propósito principal e uma seleção gulosa (*greedy*) é usada.

Já na opção de balanceamento de carga, o modelo explorará a técnica de *round-robin* alocando de forma circular cada objetivo a um candidato, até que todos tenham sido atribuídos. Logo, o coordenador busca promover melhores condições para os agentes, assumindo a hipótese que o custo de planejamento é diretamente proporcional ao número de objetivos. O Algoritmo 7 apresenta o processo de atribuição de objetivos aos agentes selecionados.

Inicialmente são criados dicionários para cada candidato, sendo a chave o agente e o valor um conjunto vazio (Linhas 3-4). A laço de atribuição é executado enquanto o número de objetivos já delegados for menor que o total de objetivos e cada iteração, esses valores vão se aproximando (laço da Linha 5). Se a estratégia escolhida for a de economia de recursos, os candidatos são ordenados de forma decrescente da cardinalidade do conjunto de objetivos alcançáveis (Linha 7), o que garante uma escolha gulosa (*greedy*). Em seguida, o primeiro elemento dessa sequência é retirado da condição de candidato (Linha 8 - função *pop()*) e se torna responsável por seus objetivos. Se for desejado o balanceamento de carga, é verificado se existe algum elemento no conjunto *Objetivos* do agente escolhido (Linha 15) e caso exista, um dos itens é retirado e atribuído a esse candidato. O próximo agente é então selecionado (Linha 21) e uma nova iteração é iniciada. Independente da estratégia escolhida, assim que um objetivo for atribuído a um candidato, ele é excluído do conjunto de metas alcançáveis dos demais agentes (Linhas 10-12 e 18-19). Da mesma forma, a coleção de objetivos já designados é atualizada (Linhas 13 e 20). Ao final, os objetivos estarão alocados aos candidatos, separados pelas chaves do dicionário.

### Planejadores

Uma premissa assumida no desenvolvimento desse modelo de planejamento multiagente é que seus componentes sejam modulares e substituíveis. Nesse contexto, cada agente

---

**Algoritmo 7** Atribuição de objetivos

---

**Entrada:**  $Estrategia, Candidatos, G$ **Saída:** Objetivos atribuídos aos candidatos

```
1:  $Atribuidos \leftarrow \emptyset$ 
2:  $agente \leftarrow$  primeiro agente
3: for  $ag \in Ag$  do
4:    $Responsabilidade[ag] \leftarrow \emptyset$  ▷ Criação dos dicionários
5: while  $\#Atribuidos < \#G$  do
6:   if  $Estrategia = Economia$  then ▷ Economia de recursos
7:      $Candidatos \leftarrow sort(Candidatos)$  ▷ Ordem decrescente de  $\#Objetivos_{agId}$ 
8:      $maxAgente \leftarrow Candidatos.pop()$ 
9:      $Responsabilidade[maxAgente] \leftarrow Objetivos_{maxAgente}$ 
10:    for  $goal \in Objetivos_{agente}$  do
11:      for  $ag \in Ag \setminus \{maxAgente\}$  do
12:         $Objetivos_{ag} \leftarrow Objetivos_{ag} \setminus \{goal\}$ 
13:         $Atribuido \leftarrow Atribuidos \cup \{goal\}$ 
14:    else ▷ Balanceamento de carga
15:      if  $\#Objetivos_{agente}$  then
16:         $goal \leftarrow Objetivos_{agente}.pop()$  ▷ Remove o primeiro objetivo
17:         $Responsabilidade[agente] \leftarrow Responsabilidade[agente] \cup \{goal\}$ 
18:        for  $ag \in Ag \setminus \{agente\}$  do
19:           $Objetivos_{ag} \leftarrow Objetivos_{ag} \setminus \{goal\}$ 
20:           $Atribuido \leftarrow Atribuidos \cup \{goal\}$ 
21:         $agente \leftarrow$  próximo agente
22: return  $Responsabilidade$ 
```

---

descrito no arquivo de entrada define, de forma independente, o componente responsável para o planejamento automatizado. Na camada de coordenação e planejamento, esse componente está presente em todos agentes.

Após a fase de atribuição dos objetivos, cada agente inicia seu planejamento utilizando a ferramenta definida. Por se tratar de uma chamada a um programa externo, esse componente deve ser capaz de recuperar e processar a saída produzida, identificando o plano encontrado ou a falha. A única exigência existente em relação à escolha da ferramenta é que ela receba como entrada os arquivos de domínio e problema que definem o agente.

No caso do coordenador, o componente *planejador* possui duas funções extras quando comparado àqueles presentes nos demais agentes. A primeira delas é que antes de invocar a ferramenta propriamente dita, ele precisa criar os arquivos PDDL a partir das informações oriundas dos agentes presentes na equipe selecionada. Nesse intuito, o coordenador constrói um arquivo de domínio, centralizando nele todos modelos de ação existentes, e um arquivo de problema, também centralizado, contendo a união das percepções iniciais. A outra função é a capacidade de interagir com o componente *conjunto mínimo de agentes* e solicitar uma nova equipe sempre que o planejamento utilizando o antigo time resultar em falha. Essa atualização da equipe pode retornar uma coleção de elementos diferentes mas na mesma quantidade ou um conjunto maior.

A última tarefa do componente é reproduzir a partir da informação textual consumida da ferramenta (plano) a sequência de operadores com seus conjuntos de pré-condições e efeitos. Essa transformação é importante porque o *validador de planos* depende da estrutura de dados definida e usada nas diversas fases do modelo, logo uma informação que não respeita tal padrão causará erros.

## Validador de planos

Uma vez que os agente que receberam objetivos tiverem calculado seus planos, esses devem ser validados para determinar se realmente podem ser executados de forma paralela e totalmente independente. A maneira utilizada para essa análise foi baseada em Xu et al. [35], trabalho no qual é apresentado um modelo do comportamento dos agentes baseado nas transições causadas por seus operadores. Em suma, essa análise verifica se dois ou mais planos possuem interseções nos conjuntos de pré-condições e efeitos de seus operadores, o que indica que existe alguma concorrência ou competição entre eles.

O trabalho de Xu et al. [35] formaliza a independência de dois planos de acordo com Definição 16.

**Definição 16.** *Dois planos  $\pi_1$  e  $\pi_2$  são independentes se e somente se*

$$\begin{aligned}
 & (*\delta_{\pi_1} \cup \delta_{\pi_1}^*) \cap (*\delta_{\pi_2} \cup \delta_{\pi_2}^*) = \emptyset, \text{ onde:} \\
 & *\delta_{\pi} = \bigcup_{i=1}^n pre(op_i) | op_i \in \pi = \{op_1, \dots, op_n\}, \\
 & \delta_{\pi}^* = \bigcup_{i=1}^n eff(op_i) | op_i \in \pi = \{op_1, \dots, op_n\}
 \end{aligned}$$

O Algoritmo 8 ilustra o processo de validação dos planos. O laço da Linhas 1-12 responde pela determinação dos conjuntos formados a partir da união de todos literais

presentes nas pré-condições ( $*\delta_\pi$ ) e efeitos ( $\delta_\pi^*$ ) dos operadores dos planos de cada agente. Nesse ponto, apenas átomos cujos identificadores não estão presentes no conjunto de literais rígidos, identificado ao final da fase de verificação, são adicionados. Aqueles que apresentam rigidez são desconsiderados porque não são alterados e portanto são no máximo lidos como pré-condições, como por exemplo para definir um cenário. Em seguida, os agentes planejadores serão combinados dois a dois (Linha 16) para então seus planos serem verificados segundo a interseção dos átomos acessados ou alterados. Cada uma dessas comparações é armazenada para determinar se existe algum par de planos que não pode ser executado paralelamente (Linhas 19 e 20). Existindo uma dupla de planos que são independentes, todos planos que foram recebidos são desconsiderados e então o planejamento centralizado é iniciado.

---

#### Algoritmo 8 Validação de planos

---

**Entrada:** Agentes selecionados, planos

**Saída:** *True* se o planos podem ser executados em paralelo. *False*, caso contrário

```

1: for  $ag \in$  Agentes selecionados do
2:    $plano \leftarrow planos[ag]$ 
3:    $precondicoes \leftarrow \emptyset$ 
4:    $efeitos \leftarrow \emptyset$ 
5:   if  $plano \neq Nulo$  then
6:     for  $op \in plano$  do
7:       for  $p \in pre(op)$  do
8:         if  $id(p) \neq rigido$  then
9:            $precondicoes \leftarrow precondicoes \cup \{p\}$ 
10:      for  $e \in eff(op)$  do
11:        if  $id(e) \neq rigido$  then
12:           $efeitos \leftarrow efeitos \cup \{e\}$ 
13:       $tokens[ag] \leftarrow precondicoes \cup efeitos$ 
14:    else
15:      return False ▷ Um agente não conseguiu encontrar um plano
16:   $duplas \leftarrow combinacoes(agentes, 2)$  ▷ Combinações de agentes dois a dois
17:   $comparacoes \leftarrow []$ 
18:  for  $dupla \in duplas$  do ▷  $dupla = (agente1, agente2)$ 
19:     $comparacoes.append(tokens[dupla[agente1]] \cap tokens[dupla[agente2]])$ 
20:  if  $\exists c \in comparacoes | c \neq \emptyset$  then
21:    return True
22:  else
23:    return False

```

---

### 4.2.3 Conjunto mínimo de agentes

O coordenador pode começar o processo de determinação do conjunto mínimo de agentes seja porque os planos calculados individualmente tenham sido considerados inválidos ou porque a transformação (seleção e atribuição) do problema original não foi efetuada.

Nessa etapa são organizadas equipes, inicialmente com apenas dois agentes, e a possibilidade dos objetivos serem atingidos é analisada usando a heurística *delete-relaxation* (Algoritmo 4). Ao contrário da fase de verificação executada pelos agentes, quando apenas os próprios operadores são utilizados, nessa fase o coordenador avalia as opções de todos membros da equipe.

O Algoritmo 9 apresenta o roteiro de definição da menor equipe com condições de satisfazer o conjunto  $G$ . Inicialmente, um conjunto de dois agentes aleatoriamente escolhidos é definido (Linha 1). O função *HDRCENTRALIZADA* (Linhas 8 - 20) funciona da maneira semelhante a do Algoritmo 4, apenas com a diferença de utilizar um conjunto de operadores oriundos de todos membros da equipe (Linhas 13 e 14), o que aumenta o número de atualizações dos estados relaxados (linha 16). A cada iteração do laço principal (Linhas 2-6), cada equipe é analisada e toda vez que um fracasso for indicado (Linha 5), o conjunto é atualizado através da troca dos agentes ou pelo aumento do tamanho do time. O algoritmo é encerrado quando uma equipe apresentar condições favoráveis (Linhas 3 e 4) ou quando todas combinações forem avaliadas sem sucesso.

---

**Algoritmo 9** Conjunto mínimo de agentes

---

**Entrada:**  $I, G$

**Saída:** *True* ou *False*

```

1: equipe  $\leftarrow$  dois agentes escolhidos aleatoriamente
2: while tamanho(equipe)  $\neq$   $\sharp Ag$  do
3:   if hdrCentralizada( $I, G, \textit{equipe}$ ) = True then
4:     return equipe
5:   else
6:     atualiza(equipe)
7:   return False
8: function HDRCENTRALIZADA( $I, G, \textit{equipe}$ )
9:    $\hat{s}_0 \leftarrow I_{agId}$ 
10:   $A_0 \leftarrow \emptyset$ 
11:   $k \leftarrow 1$ 
12:  while  $\nexists s \subseteq s_k$  que satisfaça  $G$  do
13:    for  $ag \in \textit{equipe}$  do
14:       $A_k \leftarrow \{\text{todos operadores} \in O_{ag} \text{ aplicáveis em } \hat{s}_{k-1}\}$ 
15:      for  $a \in A_k$  do
16:         $\hat{s}_k \leftarrow \hat{s}_k \cup \{\gamma^+(\hat{s}_{k-1}, a)\}$ 
17:      if  $\hat{s} = \hat{s}_{k-1}$  then
18:        return False
19:       $k \leftarrow k + 1$ 
20:  return True

```

---

### Camada de apresentação dos planos

Essa camada é mais simples do modelo por ter como função apenas a apresentação dos planos ao usuário. Os planos calculados por cada agente, oriundos do sucesso da transformação de um problema MAP para vários SAP, são listados separadamente. No caso

do planejamento centralizado, além da sequência de operadores, é indicado o número de agentes utilizados, isto é, o conjunto mínimo que foi necessário para obter sucesso. Em ambos os casos, o tamanho dos planos e o tempo total gasto na execução de todo modelo são apresentados.

Os Códigos 4.3 e 4.4 ilustram exemplos de execuções do modelo e das informações apresentadas.

Código 4.3: Exemplo de saída - logistics.

```

Plano multi-agente:
Agentes empregados: 3
loadTruck ['POS2', 'OBJ21', 'TRUCK2']
loadTruck ['POS2', 'OBJ23', 'TRUCK2']
driveTruck ['APT2', 'POS2', 'TRUCK2', 'CIT2']
unloadTruck ['APT2', 'OBJ21', 'TRUCK2']
unloadTruck ['APT2', 'OBJ23', 'TRUCK2']
loadTruck ['POS1', 'OBJ13', 'TRUCK1']
loadTruck ['POS1', 'OBJ11', 'TRUCK1']
flyPlane ['APT2', 'PLANE1', 'APT1']
loadPlane ['APT2', 'OBJ21', 'PLANE1']
loadPlane ['APT2', 'OBJ23', 'PLANE1']
flyPlane ['APT1', 'PLANE1', 'APT2']
unloadPlane ['APT1', 'OBJ21', 'PLANE1']
unloadPlane ['APT1', 'OBJ23', 'PLANE1']
driveTruck ['APT1', 'POS1', 'TRUCK1', 'CIT1']
loadTruck ['APT1', 'OBJ21', 'TRUCK1']
loadTruck ['APT1', 'OBJ23', 'TRUCK1']
unloadTruck ['APT1', 'OBJ13', 'TRUCK1']
unloadTruck ['APT1', 'OBJ11', 'TRUCK1']
driveTruck ['POS1', 'APT1', 'TRUCK1', 'CIT1']
unloadTruck ['POS1', 'OBJ21', 'TRUCK1']
unloadTruck ['POS1', 'OBJ23', 'TRUCK1']
Total de acoes: 21
Terminado com 0.3196847438812256 segundos

```

#### Código 4.4: Exemplo de saída - rovers.

```
Agente rover1
Goals: {'communicated_rock_data', 'waypoint2'}, ('communicated_rock_data', 'waypoint3'})
Plano:
  navigate ['ROVER1', 'WAYPOINT3', 'WAYPOINT2']
  sample_rock ['ROVER1', 'ROVERSTORE', 'WAYPOINT2']
  communicate_rock_data ['GENERAL', 'WAYPOINT2', 'WAYPOINT3', 'ROVER1', 'WAYPOINT2']
  drop ['ROVER1', 'ROVERSTORE']
  navigate ['ROVER1', 'WAYPOINT2', 'WAYPOINT3']
  sample_rock ['ROVER1', 'ROVERSTORE', 'WAYPOINT3']
  navigate ['ROVER1', 'WAYPOINT3', 'WAYPOINT0']
  communicate_rock_data ['GENERAL', 'WAYPOINT0', 'WAYPOINT3', 'ROVER1', 'WAYPOINT3']
  Total de acões: 8
Agente rover2
Goals: {'communicated_soil_data', 'waypoint1'}, ('communicated_soil_data', 'waypoint4'})
Plano:
  sample_soil ['ROVER2', 'ROVERSTORE', 'WAYPOINT4']
  communicate_soil_data ['GENERAL', 'WAYPOINT4', 'WAYPOINT3', 'ROVER2', 'WAYPOINT4']
  navigate ['ROVER2', 'WAYPOINT4', 'WAYPOINT0']
  drop ['ROVER2', 'ROVERSTORE']
  navigate ['ROVER2', 'WAYPOINT0', 'WAYPOINT1']
  sample_soil ['ROVER2', 'ROVERSTORE', 'WAYPOINT1']
  navigate ['ROVER2', 'WAYPOINT1', 'WAYPOINT0']
  communicate_soil_data ['GENERAL', 'WAYPOINT0', 'WAYPOINT3', 'ROVER2', 'WAYPOINT1']
  Total de acões: 8
Agente rover0
Goals: {'communicated_image_data', 'objective0', 'high_res'}, ('communicated_rock_data', 'waypoint4'})
Plano:
  calibrate ['OBJECTIVE0', 'ROVER0', 'CAMERAO', 'WAYPOINT2']
  navigate ['ROVER0', 'WAYPOINT2', 'WAYPOINT0']
  navigate ['ROVER0', 'WAYPOINT0', 'WAYPOINT4']
  take_image ['HIGH_RES', 'ROVER0', 'CAMERAO', 'OBJECTIVE0', 'WAYPOINT4']
  communicate_image_data ['WAYPOINT4', 'ROVER0', 'OBJECTIVE0', 'WAYPOINT3', 'HIGH_RES', 'GENERAL']
  sample_rock ['ROVER0', 'ROVERSTORE', 'WAYPOINT4']
  communicate_rock_data ['GENERAL', 'WAYPOINT4', 'WAYPOINT3', 'ROVER0', 'WAYPOINT4']
  Total de acões: 7
Estatísticas finais:
Total de agentes: 3
Total de acões: 23
Terminado com 0.7851569652557373 segundos
```

# Capítulo 5

## Experimentos e Análise dos Resultados

A fim de avaliar a eficácia e eficiência do modelo proposto (LCMAP), experimentos foram realizados a partir de casos retirados da *International Planning Competition* (IPC) [20], por esta ser uma fonte largamente explorada nos trabalhos correlatos. Entre as opções disponíveis, seis domínios foram escolhidos, sendo três fracamente e outros três fortemente acoplados.

Os resultados obtidos foram comparados com dois outros modelos de planejamento multiagente: FMAP<sup>1</sup> [8] e MADLA<sup>2</sup> [10]. As razões que motivaram a escolha desses dois sistemas foram:

- ambos disponibilizam implementações executáveis que permitem a realização de testes;
- no caso do FMAP, o trabalho original compara seu desempenho com outras duas ferramentas: MAP-POP [31] e MAPR [5]. Sendo assim, a escolha desse modelo permite a comparação, de forma indireta, com esses dois outros trabalhos.

As configurações de planejamento, ou seja, domínio e problema, foram semelhantes aos exemplos definidos nos casos de testes do FMAP [8], a menos de pequenas diferenças na estrutura da linguagem PDDL que foi estendida (pelo FMAP) para comportar marcações sobre os objetivos e compartilhamento de informações.

Os domínios escolhidos para testes do IPC [20] são:

- *Satellite* - cada agente simboliza um satélite que é individualizado por seus atributos referentes à posição, orientação (direção) e instrumentos disponíveis. Apesar do fato de possuírem qualidades diferentes, a transformação do problema original é favorecida pelo fato dos agentes não necessitarem de cooperação para executar suas ações. O problema consiste em escalonar observações dos satélites que incluem a coleta e o armazenamento de dados usando os diferentes instrumentos para observar uma coleção de alvos;
- *Rovers* - motivado pelas missões de exploração de Marte. O objetivo é usar um conjunto de robôs (agentes) para percorrer pontos do planeta realizando uma variedade de operações de coleta de amostras e transmissão de dados para a base. O

---

<sup>1</sup>Disponível em: <http://users.dsic.upv.es/grupos/grps/tools/map/fmap.html>

<sup>2</sup>Disponível em: <https://github.com/stolba/MADLAPlanner/wiki/MADLA-Planner>

problema inclui restrições de visibilidade dessa base a partir de várias posições e de habilidade de cada agente percorrer caminhos entre pares de pontos. Os robôs são diferenciados pelo conjunto de equipamentos que possuem e utilizam para executar as tarefas. Não há necessidade de cooperação durante a execução e a única interação possível é quando uma amostra é coletada e portanto torna-se indisponível para os demais agentes;

- *Zeno-travel* - relacionado a uma questão de transporte na qual as pessoas devem embarcar em aviões, viajar entre localidades e então desembarcar. Os aviões consomem combustíveis em diferentes taxas de acordo com a velocidade do deslocamento;
- *Depots* - composto por dois tipos de agentes, *depósitos* e *caminhões*, que devem cooperar a fim de satisfazer os objetivos. Esse é o caso mais complexo dos testes e configura um domínio fortemente acoplado com muitas dependências entre os agentes [8];
- *Elevators* - os agentes nesse domínio são elevadores que variam na velocidade, podendo ser rápidos ou lentos, e nos andares que podem acessar. Essa última característica define a necessidade de cooperação para que alguns objetivos sejam satisfeitos. Por exemplo, um elevador pode não ser capaz de transportar um passageiro para um certo andar, então ele irá até um andar intermediário para que o passageiro possa entrar em outro elevador que acessa o destino final [8];
- *Logistics* - os agentes nesse domínio são *aviões* e *caminhões*. A entrega de alguns pacotes envolvem a cooperação de elementos dos dois tipos, dado que a viagem entre cidades só pode ser realizada por aviões e o deslocamento dentro da mesma cidade são de responsabilidade dos caminhões [8].

Todos os experimentos foram realizados em um único computador com um processador *dual-core Intel Core i5-2410M* com quatro *threads* e 6 GB RAM. O ambiente operacional utilizado foi o *Linux Mint 18.1 Serena 64-bit*.

Para os domínios fracamente acoplados foram utilizados de um a cinco agentes. Para os fortemente acoplados, o limites variam entre três e oito de acordo com o domínio. Em relação ao planejador automatizado, todos agentes utilizaram o *Fast-Forward (FF)* [36]. Por disponibilizar duas estratégias diferentes de planejamento, as referências aos resultados do LCMAP tiveram acrescidos rótulos para identificar se foi almejado o balanceamento de carga (LCMAP-BC) ou a economia (LCMAP-E) de agentes.

Nas Tabelas 5.1, 5.2, 5.3, 5.4, 5.5 e 5.6 são apresentados os resultados comparativos entre os modelos usados. As métricas utilizadas foram:

- *Tempo* - intervalo gasto pela aplicação até a apresentação do plano. No caso do LCMAP e MADLA, que são executados por linhas de comando, o tempo compreendido entre o início e o término do processo foi medido através do uso do comando *time*<sup>3</sup>. No caso do FMAP, o processo não é encerrado com o término do planejamento e por isso tal técnica não pode ser aplicada. Entretanto, o tempo gasto por cada agente é apresentado assim que o plano é encontrado, e portanto o maior desses valores foi assumido;

---

<sup>3</sup><https://linux.die.net/man/1/time>

- *Número de ações*: Soma dos tamanhos das sequências calculadas de maneira independente ou do plano centralizado;
- *Agentes empregados*: Quantidade de agentes utilizados durante o planejamento independente ou empregados na execução do plano computado de coordenador.

A primeira avaliação feita a partir dos resultados é baseada na métrica de tempo. Para aprimorar a visualização dos valores, as Figuras 5.1, 5.2, 5.3, 5.4, 5.5 e 5.6 foram construídas com o intuito de facilitar a comparação do desempenho.

É possível verificar que a variação de valores é bem significativa, podendo atingir um aumento de  $1000\times$  entre as simulações de um e cinco agentes (nos domínios *satellite* e *zeno-travel*); por esse motivo, os gráficos são apresentados com escala logarítmica no Eixo Y.

Na quase totalidade dos resultados, o tempo gasto nas execuções caracterizou-se como uma função do número de agentes e objetivos com a propriedade de ser monótona e crescente. A única exceção é feita à configuração com quatro agentes no domínio *elevators* utilizando FMAP, caso no qual a saída apresentou uma diminuição de aproximadamente 50% quando comparada com a anterior. Entretanto, fica claro que qualquer solução é sensivelmente impactada tanto pelo aumento do número de agentes envolvidos quanto pela quantidade de objetivos. Cabe ressaltar que apesar das configurações criarem uma correlação positiva entre essas variáveis, os experimentos do domínio *elevators* mostraram que mesmo mantendo constante o conjunto de metas, a adição de novos membros no sistema implicou em uma resposta mais demorada. De uma forma indireta, essa conclusão mostra que a hipótese assumida na fase de atribuição dos objetivos é válida e portanto justifica a transformação do problema original.

Da análise dos domínios fracamente acoplados, *satellite* (Figura 5.1), *rovers* (Figura 5.2) e *zeno-travel* (Figura 5.3), os resultados obtidos a partir do uso do modelo LCMAP, independente da estratégia escolhida, foram melhores, isto é, menores em 71,43% (10 em 14) dos testes. Esse comportamento ilustra que explorar a ausência da necessidade de cooperação entre os agentes, tanto na fase de planejamento quanto na de execução, é uma alternativa de mitigação dos efeitos do aumento do número de membros no sistema. Na comparação restrita ao FMAP, enquanto essa solução utiliza uma forte interação entre os planejadores a cada iteração do processo do refinamento dos planos, o LCMAP apresentou um desempenho melhor por explorar o desacoplamento total dos agentes. Nos quatro testes nos quais o LCMAP apresentou desempenho inferior ao FMAP (*rovers* com um agente e *zeno-travel* até três agentes), a razão foi o custo (tempo) das execuções das fases de verificação e transformação do problema, atividades que não estão presentes no FMAP. Os resultados apresentados pelo MADLA foram os piores tanto em valores absolutos quanto na abrangência, além disso foi incapaz de resolver cinco problemas. Ainda considerando esse quesito de número de instâncias resolvidas, o LCMAP foi também superior, resolvendo 100% dos casos enquanto o FMAP apresentou um insucesso.

Da análise dos domínios fortemente acoplados, *depots* (Figura 5.4), *elevators* (Figura 5.5) e *logistics* (Figura 5.6), o LCMAP apresentou resultados melhores em 75% (6 em 8) dos testes. Nesses casos, apesar das fases de pré-planejamento, o ganho foi devido o papel centralizador do coordenador que computou os planos sem a necessidade de uma forte interação com os demais agentes. Nas configurações nas quais teve um desempenho pior (*elevators* com quatro e cinco agentes), o LCMAP foi no máximo  $2\times$  mais lento

enquanto que nas vezes que superou os concorrentes chegou a ser aproximadamente  $100\times$  mais rápido (*logistics* com 5 agentes).

Considerando os dois tipos de domínios, o modelo proposto foi superior em 72,72% (16 em 22) dos experimentos e apresentou resultados muito próximos, independente da estratégia adotada na atribuição dos objetivos, isto é, balanceamento de carga ou economia de recursos.

Em relação ao número de agentes empregados, o LCMAP conseguiu apresentar soluções mais econômicas em 36,36% (8 em 22) das configurações. Em uma análise desatenciosa esse desempenho pode parecer ruim porém considerando todas condições de contorno, esse valor pode ser muito melhor. Na realidade, a quantidade de casos que possibilitam o uso de um número menor de agentes deve desconsiderar aqueles com apenas um agente e também aqueles já definidos com o conjunto mínimo<sup>4</sup> (*depots* e *logistics*). Dessa forma, excluindo três e sete ocorrências das respectivas condições, o total de configurações passíveis de economia é na verdade doze. Nessas condições, o desempenho do LCMAP sobe para 66,67% (8 em 12) e 87,5% dos casos de seleções mais conservadoras ocorreram nos domínios fracamente acoplados, fato que já era esperado pela falta de necessidade de cooperação. Essa análise destaca outra vantagem do modelo proposto, que além de ser mais rápido é capaz de ser mais eficiente na alocação dos recursos do que os outros modelos comparados. A estratégia de balanceamento de carga do LCMAP não foi discutida porque, por definição, utiliza o maior número possível de agentes para que cada um receba o menor conjunto possível de objetivos.

Considerando o tamanho do plano final, em nove experimentos o LCMAP conseguiu encontrar soluções que apresentaram um número menor de ações. Essa quantidade final pode ser igual ao plano centralizado ou resultado da soma dos tamanhos do soluções computadas de maneira independente. Apesar da diferença em relação ao segundo plano menor ter variado apenas entre uma e duas ações, essa característica ainda sim deve ser considerada como uma vantagem do modelo proposto.

Por fim, uma ressalva que merece destaque é o fato do LCMAP buscar sempre a transformação do problema, mesmo que a classificação do tipo de domínio induza à necessidade de cooperação, normalmente esperada nos casos fortemente acoplados. No experimento do domínio *elevators* com quatro agentes, as condições definidas foram capazes de permitir que o problema fosse transformado em instâncias menores. Esse fato reforça a afirmação que muitos problemas de planejamento multiagente são passíveis de soluções individuais sob determinadas condições [26].

---

<sup>4</sup>A configuração *elevators* com cinco agentes pode ser resolvida com quatro agentes.

Tabela 5.1: Resultados para *Satellite*.

Modelo	Número de agentes	Objetivos	Tempo [s]	Número de ações	Agentes empregados
LCMAP-BC	1	3	0,166	9	1
	2	5	0,313	14	2
	3	8	0,939	22	3
	4	9	1,78	31	4
	5	13	5,44	38	5
LCMAP-E	1	3	0,179	9	1
	2	5	0,302	11	1
	3	8	0,842	15	1
	4	9	1,76	21	3
	5	13	5,541	30	4
MADLA	1	3	0,975	10	1
	2	5	5,696	41	2
	3	8	28,733	199	3
	4	9	Erro		
	5	13	Erro		
FMAP	1	3	0,208	9	1
	2	5	1,639	11	2
	3	8	3,326	15	1
	4	9	13,115	21	3
	5	13	90,998	31	4

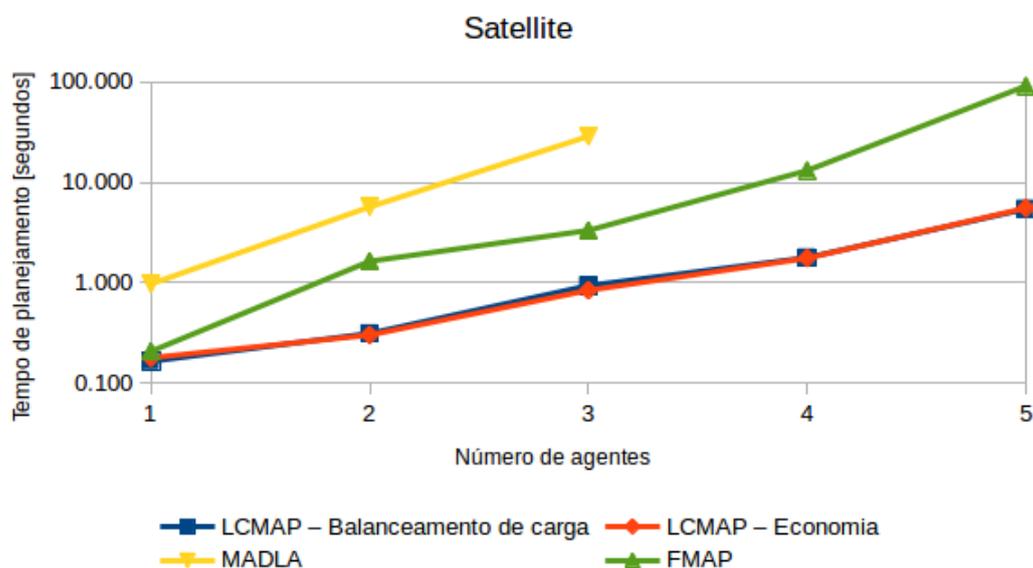


Figura 5.1: Comparação dos tempos obtidos a partir dos modelos - *Satellite*.

Tabela 5.2: Resultados para *Rovers*.

Modelo	Número de agentes	Objetivos	Tempo [s]	Número de ações	Agentes empregados
LCMAP-BC	1	3	0,288	10	1
	2	3	0,344	8	2
	3	6	0,580	26	3
	4	8	1,021	35	4
	5	12	6,322	59	5
LCMAP-E	1	3	0,320	10	1
	2	3	0,364	11	1
	3	6	0,581	25	1
	4	8	1,053	36	2
	5	12	6,333	53	2
MADLA	1	3	0,905	10	1
	2	3	1,673	17	2
	3	6	8,993	36	3
	4	8	18,076	57	3
	5	12	Erro		
FMAP	1	3	0,262	10	1
	2	3	1,048	8	2
	3	6	2,528	19	3
	4	8	20,076	36	3
	5	12	83,610	54	4

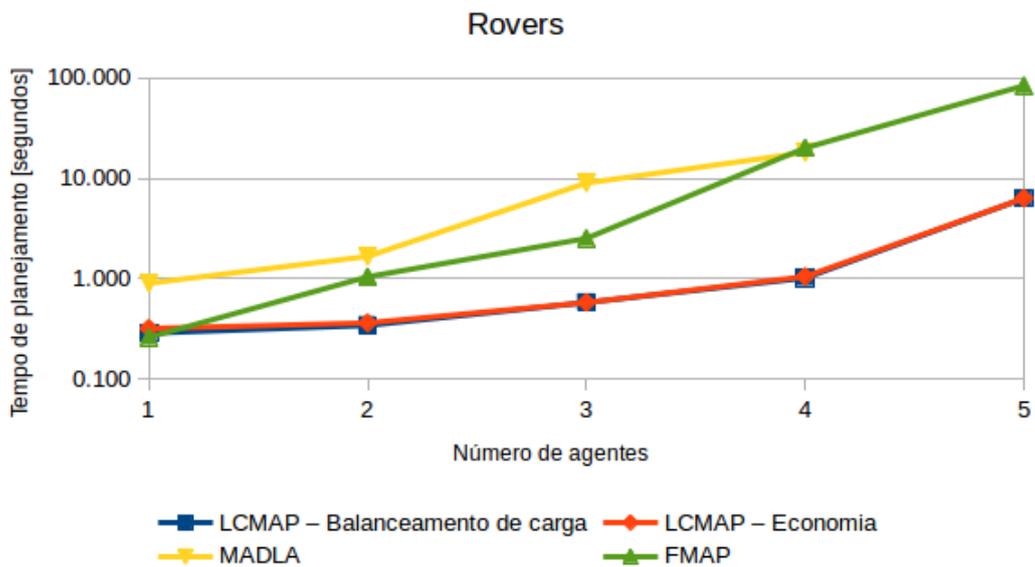


Figura 5.2: Comparação dos tempos obtidos a partir dos modelos - *Rovers*.

Tabela 5.3: Resultados para *Zeno-travel*.

Modelo	Número de agentes	Objetivos	Tempo [s]	Número de ações	Agentes empregados
LCMAP-BC	1	3	0,758	1	1
	2	4	1,160	6	2
	3	7	4,716	11	3
	5	9	35,546	38	5
LCMAP-E	1	3	0,749	1	1
	2	4	1,175	8	1
	3	7	4,714	8	1
	5	9	38,097	38	3
MADLA	1	3	1,045	1	1
	2	4	40,417	322	2
	3	7	Erro		
	5	9	Erro		
FMAP	1	3	0,115	1	1
	2	4	0,581	6	2
	3	7	2,722	12	2
	5	9	86,486	34	3

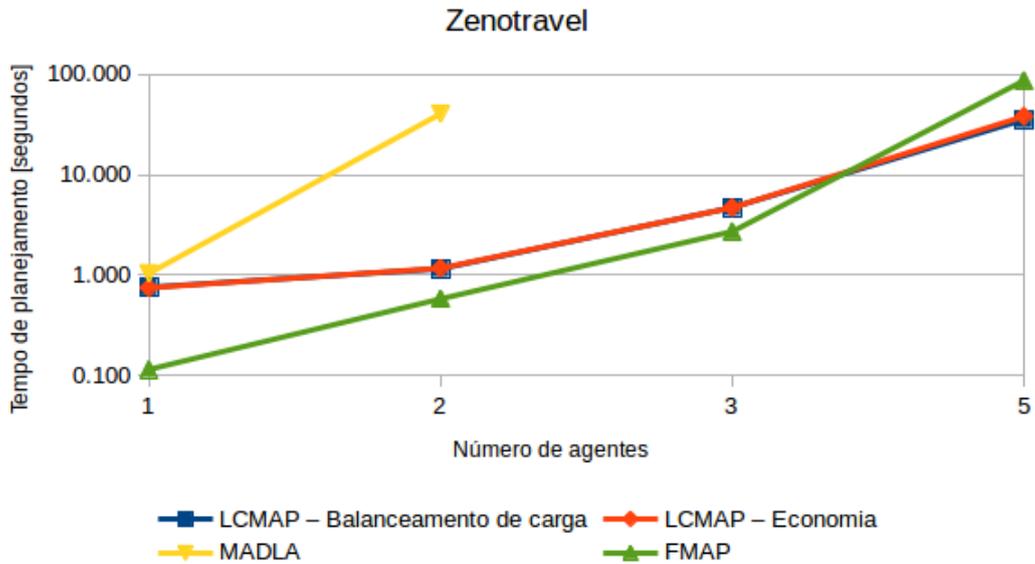


Figura 5.3: Comparação dos tempos obtidos a partir dos modelos - *Zeno-travel*.

Tabela 5.4: Resultados para *Depots*.

Modelo	Número de agentes	Objetivos	Tempo [s]	Número de ações	Agentes empregados
LCMAP-BC	5	2	0,495	12	5
	8	4	28,663	28	8
LCMAP-E	5	2	0,484	12	5
	8	4	30,887	28	8
MADLA	5	2	23,608	30	5
	8	4	Erro		
FMAP	5	2	6,288	14	5
	8	4	Erro		

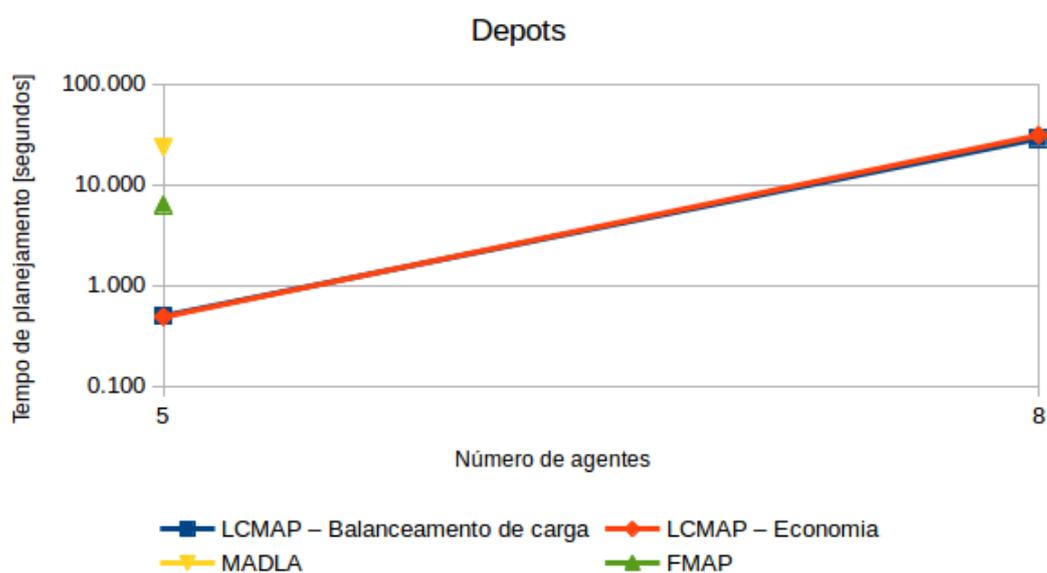


Figura 5.4: Comparação dos tempos obtidos a partir dos modelos - *Depots*.

Tabela 5.5: Resultados para *Elevators*.

Modelo	Número de agentes	Objetivos	Tempo [s]	Número de ações	Agentes empregados
LCMAP-BC	3	3	2,365	14	2
	4	3	2,577	9	3
	5	3	13,763	13	2
LCMAP-E	3	3	2,43	14	2
	4	3	2,606	11	2
	5	3	14,505	13	2
MADLA	3	3	Erro		
	4	3	Erro		
	5	3	Erro		
FMAP	3	3	3,648	15	2
	4	3	1,743	9	2
	5	3	7,124	15	3

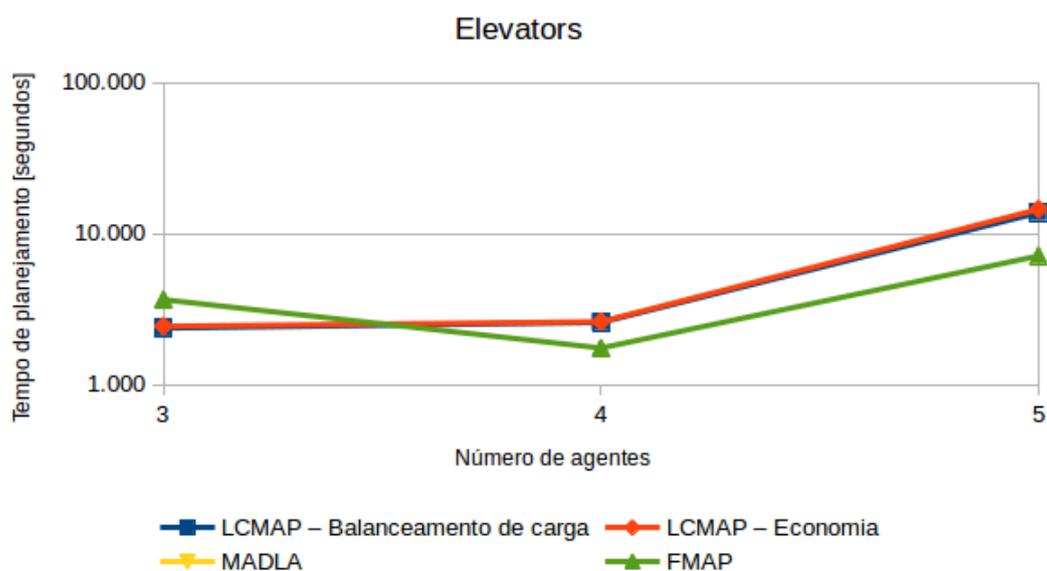


Figura 5.5: Comparação dos tempos obtidos a partir dos modelos - *Elevators*.

Tabela 5.6: Resultados para *Logistics*.

Modelo	Número de agentes	Objetivos	Tempo [s]	Número de ações	Agentes empregados
LCMAP-BC	3	4	0,259	21	3
	4	8	0,372	31	4
	5	10	0,802	46	5
LCMAP-E	3	4	0,262	21	3
	4	8	0,411	31	4
	5	10	0,876	46	5
MADLA	3	4	36,166	651	3
	4	8	Erro		
	5	10	Erro		
FMAP	3	4	4,818	20	3
	4	8	21,941	33	4
	5	10	84,025	47	5

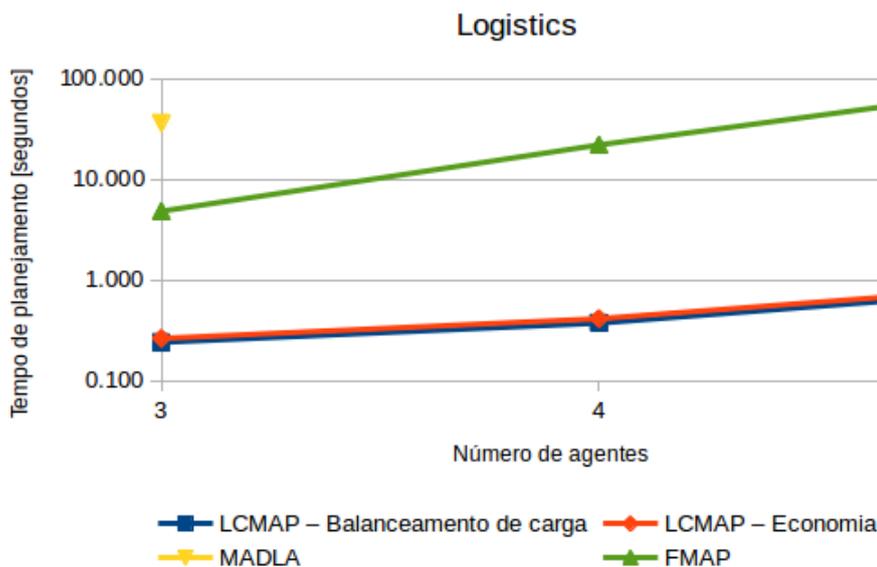


Figura 5.6: Comparação dos tempos obtidos a partir dos modelos - *Logistics*.

## 5.1 Escalabilidade

Dois experimentos adicionais foram preparados para analisar a escalabilidade do LCMAP. Os domínios escolhidos foram *rovers* e *logistics* para avaliar os tipos de problemas fraca e fortemente acoplados, respectivamente. As razões que motivaram essas escolhas foram:

- *rovers* consiste em um conjunto de robôs que podem ser diferenciados por suas capacidades. Sendo assim, novos elementos podem ser criados a partir da configuração com cinco agente experimentada (Tabela 5.2 e Figura 5.2) variando apenas esse conjunto de capacidades. Como os objetivos desse domínio dependem basicamente

de três operações, fotografar, coletar amostras de rochas e do solo, para cada robô (*rover0*, ..., *rover4*) são criados elementos idênticos em relação à posição inicial porém capazes de realizar uma, duas ou três das operações necessárias;

- *logistics* permite a especialização dos agentes a partir de suas restrições de deslocamentos. A configuração avaliada no experimento com cinco agentes (Tabela 5.6 e Figura 5.6) consiste em quatro cidades, cada uma com seu aeroporto, nas quais quatro caminhões e um avião cooperam para satisfazer os objetivos. Então, novos agentes podem ser instanciados e diferenciados pela cidade qual estão presentes. O aumento do número de elementos do sistema consiste nas inserções de novos agentes dos dois tipos com a condição de até quatro aviões (um por aeroporto) e até dezesseis caminhões (quatro por cidade);
- ambos domínios apresentaram elevados valores de tempo para a configuração com cinco agentes, portanto configuram bons pontos de partida para a análise de escalabilidade;

A escolha da configuração com cinco agentes como ponto inicial dessa análise é devido ao fato do conjunto de objetivos ser plenamente satisfeito sob tais condições. Portanto, os agentes adicionados apenas contribuem como novas opções de escolha pelo coordenador. Cabe ressaltar que o número de objetivos foi mantido constante dado que o foco era avaliar o impacto do aumento do sistema. Por fim, para essa análise não serão realizados experimentos com os outros dois modelos dado que o LCMAP demonstrou resultados melhores em relação tanto ao tempo gasto quanto à abrangência.

Para analisar a escalabilidade do modelo uma nova métrica, descrita na Equação 5.1, foi criada para descrever o impacto do crescimento do sistema no tempo de processamento.

$$aumento_i = \frac{\text{Tempo de processamento para } q_i \text{ agentes}}{\text{Tempo de processamento para } q_0 \text{ agentes}}, \text{ com } q = \{5, 6 \dots, 20\} \quad (5.1)$$

A Tabela 5.7 e a Figura 5.7 ilustram os tempos de planejamento gastos pelo LCMAP e as taxas de aumento utilizando as duas estratégias de atribuição disponíveis. A avaliação desses resultados obtidos no contexto do domínio *rovers* mostrou que o LCMAP possui uma boa escalabilidade dado que em 84,37% dos casos (27 de 32), o aumento de tempo (Equação 5.1) foi menor que a taxa de crescimento do número de agentes. Essa propriedade é devido à transformação do problema original em instâncias menores, o que garante aos agentes independência entre si durante o planejamento. Cabe ressaltar que apesar dessa delegação de responsabilidades, um aumento no tempo total era esperado pois o ambiente de testes comporta apenas quatro processos simultâneos (quatro *threads* de execução) e o total de agentes instanciados chegou a ser 5× maior que essa capacidade.

Entretanto, o modelo apresentou um comportamento quase constante no intervalo de [5; 10], o que provavelmente seria conservado nas demais configurações, caso a capacidade de execuções paralelas do computador fosse maior que a utilizada. Em relação aos outros modelos, um fato que merece destaque é que o tempo gasto pelo LCMAP para resolver o problema com vinte agentes é 3× menor que aquele dispendido pelo FMAP para resolver um problema com apenas cinco elementos.

Em relação ao domínio fortemente acoplado, *logistics*, apesar da escalabilidade apresentada não ser comparável ao caso anterior, os resultados ilustrados na Tabela 5.8 e na

Figura 5.8 foram úteis para apontar uma oportunidade de melhoria do modelo. Quando a necessidade de cooperação é percebida ao final da fase de verificação, o coordenador identifica a partir dos conjuntos de objetivos alcançáveis de cada agente que o planejamento centralizado será adotado. Porém antes de iniciá-lo, o coordenador tentará organizar a menor equipe com capacidades de satisfazer todas as metas (vide Algoritmo 9) e nesse instante todas as combinações possíveis de times compostos de 2 a  $n$  agentes serão testadas até que uma apresente possibilidade de sucesso. Entretanto, o processo de avaliação dessas opções não é paralelizado, conseqüentemente o aumento do número de elementos afeta diretamente o tempo gasto nessa fase devido ao fato da quantidade de possibilidades também crescer. Com a paralelização dessa etapa, mais de uma análise pode ser feita simultaneamente, garantindo uma melhoria de desempenho. Ainda em relação ao processo de formação das equipes, é importante ressaltar que a diferença entre as estratégias de balanceamento de carga e economia foi maior que nos outros testes devido ao não-determinismo da escolha da equipe a ser avaliada.

Apesar dessa limitação, o LCMAP apresentou resultados melhores do que os outros modelos mesmo em condições mais complexas. Para a configuração com 16 agentes, os tempos gastos foram menores do que aquele apresentado pelo FMAP com apenas 5 agentes, 61,325 e 75,873 contra 84,025 segundos, respectivamente.

Tabela 5.7: Resultados para *Rovers*.

Número de agentes	Taxa de aumento	LCMAP - BC		LCMAP - E	
		Tempo [s]	Aumento	Tempo [s]	Aumento
5	1,000	6,385	1,000	6,403	1,000
6	1,200	6,660	1,043	6,701	1,047
7	1,400	7,053	1,105	7,159	1,118
8	1,600	7,283	1,141	7,455	1,164
9	1,800	7,422	1,162	7,430	1,160
10	2,000	8,025	1,257	7,968	1,244
11	2,200	9,271	1,452	9,343	1,459
12	2,400	11,205	1,755	11,118	1,736
13	2,600	13,096	2,051	13,100	2,046
14	2,800	14,490	2,269	14,593	2,279
15	3,000	16,823	2,635	16,792	2,623
16	3,200	17,932	2,808	17,768	2,775
17	3,400	20,566	3,221	20,364	3,180
18	3,600	22,688	3,553	23,066	3,602
19	3,800	25,656	4,018	25,514	3,985
20	4,000	27,582	4,320	27,744	4,333

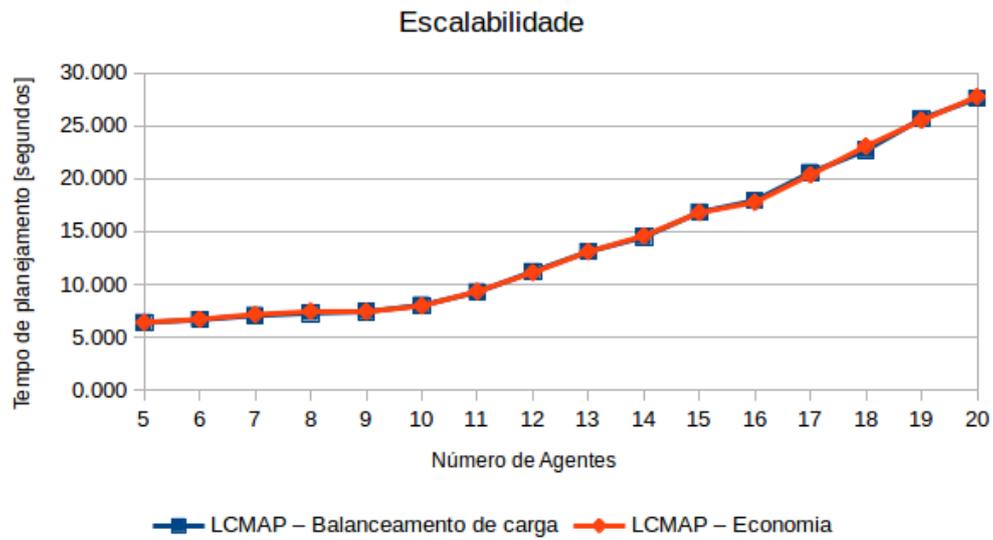


Figura 5.7: Análise de escalabilidade para *rovers*.

Tabela 5.8: Resultados para *Logistics*.

Número de agentes	Taxa de aumento	LCMAP - BC		LCMAP - E	
		Tempo [s]	Aumento	Tempo [s]	Aumento
5	1,000	0,900	1,000	0,895	1,000
6	1,200	1,528	1,698	1,444	1,613
7	1,400	2,353	2,614	2,111	2,359
8	1,600	3,947	4,386	3,919	4,379
9	1,800	6,462	7,180	6,427	7,181
10	2,000	9,988	11,098	9,896	11,057
11	2,200	15,757	17,508	15,110	16,883
12	2,400	23,545	26,161	20,336	22,722
13	2,600	33,871	37,634	25,274	28,239
14	2,800	47,466	52,740	42,372	47,343
15	3,000	54,805	60,894	63,302	70,728
16	3,200	61,325	68,139	75,873	84,774
17	3,400	91,118	101,242	99,851	111,565
18	3,600	126,997	141,108	108,439	121,161
19	3,800	136,374	151,527	131,920	147,397
20	4,000	165,784	184,204	153,758	171,797

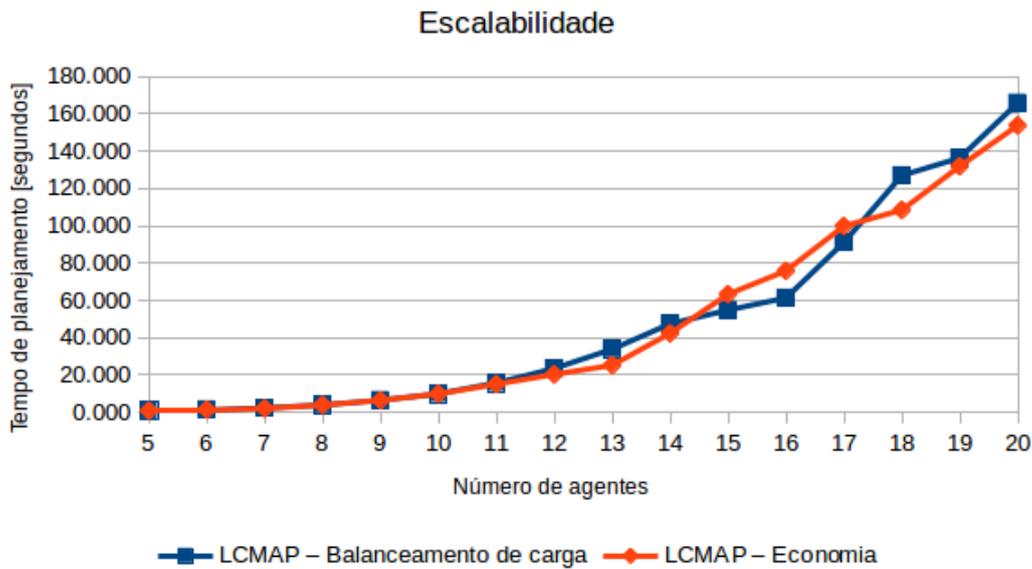


Figura 5.8: Análise de escalabilidade para *logistics*.

## 5.2 Análise dos Resultados

A fim de avaliar a eficácia e eficiência do LCMAP, experimentos foram realizados e os resultados foram comparados com dois modelos com implementações disponíveis, FMAP [8] e MADLA [10]. As análises foram baseadas nas métricas utilizadas: (i) tempo total gasto, (ii) tamanho dos planos, e (iii) número de agentes empregados, e na escalabilidade oferecida pelo modelo.

Do ponto de vista da capacidade de apresentar uma solução, o modelo proposto conseguiu resolver todos os problemas enquanto os demais falharam em pelo menos um caso, o que demonstra sua eficácia. Considerando o tempo gasto, o LCMAP foi mais eficiente em 72,72% dos experimentos e chegou a ser 100× mais rápido. Com relação ao número de agentes empregados, o LCMAP conseguiu propor soluções mais econômicas em 36,36% ou 66,67% dos casos, considerando todos os experimentos ou apenas aquelas passíveis de economia, respectivamente. Em todos os experimentos, a diferença de tempo gasto entre as estratégias de balanceamento de carga (LCMAP-BC) e economia (LCMAP-E) foi pequena porque o processo de atribuição de objetivos (Algoritmo 7) executa quantidades de operações semelhantes, independente da estratégia. Os melhores resultados obtidos nos experimentos dos domínios *satellite*, *rovers* e *logistics* são causados principalmente pelo desacoplamento dos agentes e consequente simplificação do processo de planejamento para os dois primeiros casos, e por uma fase de verificação ser mais simples na condições propostas pelo problema, no último caso.

Em relação à escalabilidade, ou seja, o impacto do aumento do número de agentes no desempenho do LCMAP, o acréscimo no tempo de saída foi menor que a taxa de crescimento do sistema em 84,37% dos experimentos para o domínio fracamente acoplado e só não foi preservado para o caso fortemente acoplado devido a uma limitação do algoritmo de definição do conjunto mínimo de agentes necessários. Outra importante observação é que as Figuras 5.7 e 5.8 comprovam a complexidade exponencial do tempo necessário

para solucionar o problema de planejamento multiagente.

Por fim, outra razão que propiciou que o modelo proposto alcançasse melhores resultados foi o processo de coordenação adotado. Principalmente em relação ao FMAP, a troca mensagens necessária para coordenar as atividades dos agentes a cada iteração do processo de refinamento dos planos configura o maior limitador para a escalabilidade. Por isso, o processo de coordenação mais leve adotado pelo LCMAP, no qual as interações entre os agentes ocorrem em momentos bem definidos, mostrou ser mais eficiente do que as propostas analisadas. Nesse processo, como os agentes interagem apenas com o coordenador para informar quais objetivos podem satisfazer, seus planos calculados ou falhas, o número de troca de mensagem pode ser definido em função do número de elementos presentes no sistema. Portanto, essa coordenação explora ao máximo a independência dos agentes, nos casos fracamente acoplados, e evita um elevado volume de mensagens adotando uma postura centralizadora para domínios fortemente acoplados.

# Capítulo 6

## Conclusões

O modelo de planejamento multiagente proposto nesse trabalho, denominado LCMAP, é um modelo de propósito geral, uma vez que é capaz de resolver tanto problemas fracamente quanto fortemente acoplados. A motivação principal para seu desenvolvimento foi o preenchimento de lacunas identificadas nos trabalhos correlatos, principalmente no tocante à verificação das capacidades dos agentes e à seleção dos agentes envolvidos no processo de planejamento. Portanto, o modelo apresentado apoia-se em fases de verificação e transformação do problema, e na validação de planos.

No modelo proposto, os agentes são caracterizados por arquivos de domínio que definem suas capacidades a partir de modelos de ação, por arquivos de problema que descrevem o conhecimento que cada um tem do ambiente, e por uma ferramenta de planejamento automatizado que garante a autonomia deliberativa.

Uma característica que distingue o LCMAP dos demais trabalhos é o fato de ele explorar as duas dimensões do planejamento multiagente: agentes como executores ou planejadores. Algumas pesquisas propõem soluções para o processo de busca do plano, seja de forma centralizada ou distribuída, e portanto atribuem ao agente o papel de planejador [5, 8, 10]. Em outros casos, o agente é visto como um executor e a análise é feita considerando a execução dos planos, verificando quando uma cooperação entre os agentes é necessária e se os planos são independentes entre si, não importando como o planejamento foi realizado [26, 35]. Nessa perspectiva, o LCMAP considera o agente tanto como planejador quanto executor, delegando conjuntos de objetivos a cada um deles para que deliberem de maneira independente entretanto, garantindo que realizem os planos sem qualquer tipo de competição ou concorrência.

Durante a fase de verificação, os agentes verificam suas capacidades utilizando uma adaptação da heurística *delete-relaxation* para estudar se os efeitos causados pelos operadores, isto é, ações instanciadas, são capazes de produzir um estado no qual todos ou parte dos objetivos são satisfeitos. Essa propriedade é uma importante contribuição para a área de pesquisa porque os trabalhos correlatos, geralmente, assumem como hipótese inicial o fato que qualquer agente é capaz de atender a qualquer objetivo, premissa que facilmente é negada em problemas do mundo real.

Já na fase de transformação, o coordenador busca selecionar os agentes capazes de resolver alguma parte do problema diminuindo assim o número de elementos envolvidos. Em seguida, os objetivos são distribuídos entre os candidatos escolhidos seguindo estratégias que visam o balanceamento de carga ou a economia de recursos. Dessa forma, o

problema original (MAP) é transformado em instâncias menores (SAP) e possivelmente mais simples.

Uma vez que os objetivos foram distribuídos, cada agente selecionado inicia seu planejamento individualmente, e é nesse ponto que o processo de coordenação adotado mostra ser mais eficiente. Ao delegar tarefas aos agentes de tal forma que eles possam deliberar suas ações sem interagir entre si, o coordenador garante que não há necessidade de troca de mensagem entre eles para controlar as atividades. Por outro lado, caso a transformação não seja possível, o coordenador, após de definir a menor equipe necessária, assume a responsabilidade de encontrar um plano utilizando o menor número possível de agentes.

A fase de validação consiste na verificação da independência dos planos, ou seja, se as sequências de ações encontradas de maneira desacoplada pelos agentes planejadores realmente podem ser executados de maneira paralela. Para isso, o coordenador aguarda a resposta de todos os agentes para analisar as interseções dos planos e ter a certeza que não há qualquer tipo de concorrência ou competição.

Essas fases foram importantes para responder à questão de pesquisa definida: *Quais são os efeitos da execução de um pré-planejamento para um planejamento multiagente?*. A execução dessas fases favorecem tanto uma redução do tempo gasto durante o planejamento quanto uma escalabilidade maior do modelo. Dessa forma, é possível enfrentar a complexidade apresentada em problemas de MAP, que se caracteriza por ser exponencial.

Por essas características, o LCMAP apresenta-se como uma contribuição para o estado da arte da área de pesquisa de planejamento multiagente, seja por conseguir resultados melhores nas métricas citadas ou por preencher lacunas presentes em trabalhos correlatos identificados na revisão da literatura.

## Divulgação da Pesquisa

Os resultados obtidos através dessa pesquisa foram publicados conforme segue:

- *Transforming Multi-Agent Planning Into Single-Agent Planning Using Best-cost Strategy*. In Proc 5<sup>th</sup> Brazilian Conference On Intelligent Systems (BRACIS) (Qualis B2) - apresenta uma discussão sobre a economia de recursos que pode ser alcançada quando os objetivos são distribuídos entre os agentes disponíveis. O trabalho propõe uma divisão dos objetivos considerando o custo total do plano que, por sua vez, é calculado pela soma das ações que o compõem.
- *Improving Multi-Agent Planning with Unsolvability and Independent Plan Detection*. In Proc 6<sup>th</sup> Brazilian Conference On Intelligent Systems (BRACIS) (Qualis B2) - apresenta um modelo de planejamento multiagente baseado na alocação dos objetivos considerando as capacidades dos agentes. O modelo foi valido em domínios fracamente acoplados do IPC (*satellite* e *rovers*).
- *Distributing goals in multi-agent planning through agent's capabilities*. International Conference on Automated Planning and Scheduling (ICAPS) (Qualis A2, em avaliação) - apresenta um modelo de planejamento multiagente capaz de resolver problemas de domínios fracamente e fortemente acoplados do IPC mantendo a alocação dos objetivos de acordo com as capacidades dos agentes. A validação mostrou que o modelo apresenta resultados de desempenho que sobrepõe resultados obtidos com outros trabalhos de estado-da-arte (FMAP 2014, MADLA 2015).

## Trabalhos Futuros

Entretanto, algumas oportunidades de melhoria destacam-se como opções de trabalhos futuros. Inicialmente, vislumbra-se sua extensão na direção de modelos de planejamento probabilísticos e dinâmicos, nos quais os efeitos das ações não são determinísticos e o ambiente pode ser alterado por eventos externos alheios às ações dos agentes. Outros domínios de planejamento podem ser avaliados, da mesma forma que outros planejadores podem ser utilizados pelos agentes a fim de comprovar a flexibilidade e modularidade do modelo. Por fim, plataformas computacionais com um maior nível de paralelismo podem ser usadas como ambientes de teste a fim de avaliar os ganhos obtidos pelo modelo quando mais agentes forem instanciados simultaneamente.

# Referências

- [1] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Cambridge University Press, 2016. 1, 5, 6, 7, 8, 9, 10, 11, 42, 43
- [2] Mathijs de Weerd, Adriaan Ter Mors, and Cees Witteveen. Multi-agent planning: An introduction to planning and coordination. In *Handouts of the European Agent Summer*, 2005. 1, 5, 18, 22
- [3] Mathijs de Weerd and Brad Clement. Introduction to planning in multiagent systems. *Multiagent and Grid Systems*, 5(4):345–355, 2009. 1, 18, 21, 22
- [4] Paolo Traverso, Malik Ghallab, and Dana Nau. *Automated Planning: Theory and Practice*. M. Kauffman, 2004. 1, 8
- [5] Daniel Borrajo. Multi-Agent Planning by Plan Reuse. In *12<sup>th</sup> International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1141–1142. IFAAMAS, 2013. 2, 28, 29, 30, 32, 34, 53, 68, 83, 84
- [6] Matthew Crosby, Anders Jonsson, and Michael Rovatsos. A Single-Agent Approach to Multiagent Planning. In *22<sup>nd</sup> European Conference on Artificial Intelligence (ECAI)*, pages 237–242. IOS Press, 2014. 2, 28, 29
- [7] Leonardo H. Moreira and Célia G. Ralha. Transforming Multi-Agent Planning Into Single-Agent Planning Using Best-cost Strategy. In *5<sup>th</sup> Brazilian Conference on Intelligent Systems (BRACIS)*, pages 462–467. IEEE, 2016. 2, 29, 30, 32, 34, 83, 84
- [8] Alejandro Torreno, Eva Onaindia, and Oscar Sapena. FMAP: Distributed Cooperative Multi-Agent Planning. *Applied Intelligence*, 41(2):606–626, 2014. 2, 3, 18, 20, 25, 26, 27, 30, 31, 32, 34, 53, 54, 66, 68, 82, 83, 84
- [9] Rafael C. Cardoso and Rafael H. Bordini. A Distributed Online Multi-Agent Planning System. In *4<sup>th</sup> Workshop on Distributed and Multi-Agent Planning (DMAP)*, pages 15–23. AAAI, 2016. 2, 22, 28, 29, 34
- [10] Michal Štolba and Antonín Komenda. MADLA: planning with distributed and local search. *Competition of Distributed and Multi-Agent Planners (CoDMAP)*, page 21, 2015. 3, 27, 30, 32, 34, 53, 66, 68, 82, 83, 84
- [11] Silvio do Lago Pereira and Leliane Nunes de Barros. A planner agent that tries its best in presence of nondeterminism. *Journal of Applied Logic*, 10(4):419–430, 2012. 12

- [12] Daniel A. M. Moreira, Karina Valdivia Delgado, and Leliane Nunes de Barros. Robust probabilistic planning with ilao. *Applied Intelligence*, pages 1–11, 2016. 12
- [13] Richard E Fikes and Nils J Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189–208, 1971. 12
- [14] Edwin PD Pednault. ADL: Exploring the Middle Ground Between STRIPS and the Situation Calculus. *Kr*, 89:324–332, 1989. 12
- [15] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL-the planning domain definition language. 1998. 12
- [16] Maria Fox and Derek Long. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 2003. 13
- [17] Stefan Edelkamp. Taming numbers and durations in the model checking integrated planning system. *Journal of Artificial Intelligence Research*, 20:195–238, 2003. 13
- [18] Alfonso Gerevini and Derek Long. Plan constraints and preferences in PDDL3. *The Language of the Fifth International Planning Competition. Technical Report, Department of Electronics for Automation, University of Brescia, Italy*, 75, 2005. 13
- [19] Daniel L Kovacs. Complete BNF Description of PDDL3. 2011. 13
- [20] Derek Long and Maria Fox. The 3<sup>rd</sup> International Planning Competition: Results and Analysis. *Journal of Artificial Intelligence Research*, 20(1):1–59, December 2003. ISSN 1076-9757. URL <http://dl.acm.org/citation.cfm?id=1622452.1622453>. 13, 53
- [21] Michael Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons Ltd, 2<sup>nd</sup> edition, 2009. 14, 15, 17
- [22] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 3<sup>rd</sup> edition, 2010. 15, 16, 25
- [23] Gerhard Weiss. *Multiagent systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 2013. 16, 17
- [24] Nikos Vlassis. A concise introduction to multiagent systems and distributed artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 1: 1–71, 2007. 16
- [25] Anders Jonsson and Michael Rovatsos. Scaling Up Multiagent Planning: A Best-Response Approach. In 21<sup>st</sup> *International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI, 2011. 18
- [26] Yu Zhang, Sarath Sreedharan, and Subbarao Kambhampati. A Formal Analysis of Required Cooperation in Multi-Agent Planning. In 26<sup>th</sup> *International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI, 2016. 18, 20, 21, 24, 27, 29, 30, 31, 32, 34, 56, 68, 82, 83, 84

- [27] Barbara Dunin-Keplicz and Rineke Verbrugge. *Teamwork in multi-agent systems: A formal approach*, volume 21. John Wiley & Sons, 2011. 22
- [28] Barbara Kitchenham. Procedures for Performing Systematic Reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004. 24, 77, 78
- [29] Christian Muise, Nir Lipovetzky, and Miquel Ramirez. MAP-LAPKT: Omnipotent multi-agent planning via compilation to classical planning. *Competition of Distributed and Multi-Agent Planners (CoDMAP)*, page 14, 2015. 25
- [30] Satyendra Singh Chouhan and Rajdeep Niyogi. MAPJA: Multi-agent planning with joint actions. *Applied Intelligence*, pages 1–15, 2017. 27, 28, 30, 32, 34, 83, 84
- [31] Alejandro Torreno, Eva Onaindia, and Oscar Sapena. An approach to multi-agent planning with incomplete information. In *20<sup>th</sup> European Conference on Artificial Intelligence (ECAI)*, volume 242, pages 762–767. IOS Press, 2012. 29, 53
- [32] B. Klöpper, S. Honiden, and W. Dangelmaier. Divide and conquer in planning for self-optimizing mechatronic systems - A first application example. In *Computational Intelligence in Control and Automation (CICA)*, pages 108–115. IEEE, 2011. 29
- [33] Felipe Meneguzzi and Michael Luck. Leveraging new plans in AgentSpeak (PL). In *International Workshop on Declarative Agent Languages and Technologies*, pages 111–127. Springer, 2008. 29
- [34] Rafael C. Cardoso and Rafael H. Bordini. Allocating Social Goals Using the Contract Net Protocol in Online Multi-Agent Planning. In *5<sup>th</sup> Brazilian Conference on Intelligent Systems (BRACIS)*, pages 199–204. IEEE, 2016. 34, 83
- [35] Dianxiang Xu, Richard Volz, Thomas Ioerger, and John Yen. Modeling and verifying multi-agent behaviors using predicate/transition nets. In *14<sup>th</sup> International Conference on Software Engineering and Knowledge Engineering*, pages 193–200. ACM, 2002. 48, 68
- [36] Jörg Hoffmann and Bernhard Nebel. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001. 54
- [37] Jörg Hoffmann, Stefan Edelkamp, Sylvie Thiébaux, Roman Englert, Frederico Liporace, and Sebastian Trüg. Engineering benchmarks for planning: the domains used in the deterministic part of ipc-4. *Journal of Artificial Intelligence Research*, 26: 453–541, 2006. 82
- [38] Amar Nath and Rajdeep Niyogi. Validation and Verification of Joint-Actions in Multi-agent Planning. In *International Conference on Information Technology (ICIT)*, pages 187–192. IEEE, 2015. 82, 83
- [39] Iván Palomares, Ronan Killough, Kim Bauters, Weiru Liu, and Jun Hong. A collaborative multiagent framework based on online risk-aware planning and decision-making. In *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*, pages 25–32. IEEE, 2016. 83

- [40] Aditya Undurti and Jonathan P How. A decentralized approach to multi-agent planning in the presence of constraints and uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2534–2539. IEEE, 2011. 83
- [41] Alejandro Torreño, Eva Onaindia, and Óscar Sapena. A flexible coupling approach to multi-agent planning under incomplete information. *Knowledge and Information Systems*, pages 1–38, 2014. 83
- [42] Amar Nath and Rajdeep Niyogi. An extension of FMAP for joint actions. In *Contemporary Computing (IC3), 2015 Eighth International Conference on*, pages 487–492. IEEE, 2015. 83
- [43] Alice Toniolo, Timothy J Norman, and Katia P Sycara. Argumentation Schemes for Collaborative Planning. In *PRIMA*, volume 7047, pages 323–335. Springer, 2011. 83
- [44] Iván Palomares, Kim Bauters, Weiru Liu, and Jun Hong. A Two-Stage Online Approach for Collaborative Multi-agent Planning Under Uncertainty. In *International Conference on Scalable Uncertainty Management*, pages 214–229. Springer, 2016. 83
- [45] Sergio Pajares Ferrando and Eva Onaindia. Defeasible argumentation for multi-agent planning in ambient intelligence applications. In *11<sup>th</sup> International Conference on Autonomous Agents and Multiagent Systems*, pages 509–516. IFAAMAS, 2012. 83
- [46] Benjamin Klöpper, Shinichi Honiden, and Wilhelm Dangelmaier. Divide & conquer in planning for self-optimizing mechatronic systems-A first application example. In *Computational Intelligence in Control and Automation (CICA)*, pages 108–115. IEEE, 2011. 83
- [47] Qing Chang, Zhonghua Wen, and Yulong Hu. Find a multi-agent planning solution in nondeterministic domain. 2012. 83
- [48] Alejandro Torreno, Oscar Sapena, and Eva Onaindia. Global Heuristics for Distributed Cooperative Multi-Agent Planning. In *25<sup>th</sup> International Conference on Automated Planning and Scheduling (ICAPS)*, pages 225–233. AAAI Press, 2015. 83
- [49] Meirav Hadad, Sarit Kraus, Irith Ben-Arroyo Hartman, and Avi Rosenfeld. Group planning with time constraints. *Annals of Mathematics and Artificial Intelligence*, 69(3):243–291, 2013. 83
- [50] Anton Dukeman and Julie A. Adams. *Hybrid mission planning with coalition formation*. PhD thesis, Vanderbilt University, 2017. 83
- [51] Mohamed Elkawkagy and Susanne Biundo. Hybrid multi-agent planning. *Multiagent System Technologies*, pages 16–28, 2011. 83
- [52] Shlomi Maliah, Ronen Brafman, and Guy Shani. Increased privacy with reduced communication in multi-agent planning. 2017. 83

- [53] Geoffrey J Gordon, Pradeep Varakantham, William Yeoh, Hoong Chuin Lau, Ajay S Aravamudhan, and Shih-Fen Cheng. Lagrangian relaxation for large-scale multi-agent planning. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 494–501. IEEE Computer Society, 2012. 83
- [54] Raz Nissim and Ronen I Brafman. Multi-agent a\* for parallel and distributed systems. In *11<sup>th</sup> International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 3, pages 1265–1266. IFAAMAS, 2012. 83
- [55] Pere Pardo, Sergio Pajares, Eva Onaindia, Lluís Godo, and Pilar Dellunde. Multiagent argumentation for cooperative planning in delp-pop. In *10<sup>th</sup> International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 3, pages 971–978. IFAAMAS, 2011. 83
- [56] Chao Wang and Peng Liu. Multi-agent planning for fleet cooperative air defense decision-making. In *9<sup>th</sup> International Conference on Natural Computation (ICNC)*, pages 533–538. IEEE, 2013. 83
- [57] Satyendra Singh Chouhan, Ashutosh Singh, and Rajdeep Niyogi. Multi-agent planning with joint actions. In *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1284–1290. IEEE, 2015. 83
- [58] Satyendra Singh Chouhan and Rajdeep Niyogi. Multi-agent planning with quantitative capability. In *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 602–606. IEEE, 2016. 83
- [59] Oscar Sapena, Alejandro Torreño, and Eva Onaindia. On the construction of joint plans through argumentation schemes. In *10<sup>th</sup> International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 3, pages 1195–1196. IFAAMAS, 2011. 83
- [60] Michal Štolba, Daniel Fišer, and Antonín Komenda. Potential heuristics for multi-agent planning. In *26<sup>th</sup> International Conference on Automated Planning and Scheduling (ICAPS)*, pages 308–316. AAAI Press, 2016. 83
- [61] Cesar Guzman, Pablo Castejon, Eva Onaindia, and Jeremy Frank. Robust Plan Execution in Multi-agent Environments. In *IEEE 26<sup>th</sup> International Conference on Tools with Artificial Intelligence*, pages 384–391. IEEE, 2014. 83
- [62] Joshua D Redding, N Kemal Ure, Jonathan P How, Matthew A Vavrina, and John Vian. Scalable, MDP-based planning for multiple, cooperating agents. In *American Control Conference (ACC)*, pages 6011–6016. IEEE, 2012. 83
- [63] Michal Štolba, Jan Tožička, and Antonín Komenda. Secure Multi-Agent Planning. In *1<sup>st</sup> International Workshop on AI for Privacy and Security*, pages 1–8. ACM, 2016. 83
- [64] Matteo Baldoni, Cristina Baroglio, and Roberto Micalizio. Social Continual Planning in Open Multiagent Systems: A First Study. In *Principles and Practice of Multi-Agent Systems (PRIMA)*, pages 575–584, 2015. 83

- [65] Jan Tozicka, Jan Jakubuv, and Antonin Komenda. Psm-based planners description for codmap 2015 competition. *Competition of Distributed and Multi-Agent Planners (CoDMAP)*, pages 29–32, 2015. 83

# Apêndice A

## Revisão Sistemática

Após a definição do problema a ser investigado em uma pesquisa, uma prática importante é a revisão da literatura. Segundo [28], uma revisão sistemática da literatura é um meio de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma determinada questão de pesquisa, área de tópico ou fenômeno de interesse. Entre os motivos que justificam a importância de sua realização, destacam-se a identificação de trabalhos correlatos, a investigação de lacunas e oportunidades de melhorias e o posicionamento adequado de novas atividades na área de pesquisa. Para facilitar a execução da revisão foi utilizada uma ferramenta disponível em <http://sesra.net>.

Em relação as principais características de uma revisão sistemática, pode-se citar:

- começa pela definição de um protocolo que especifica a questão de pesquisa e os métodos que serão utilizados;
- baseia-se em uma estratégia de busca para encontrar o maior número de trabalhos relacionados possível;
- registra essa estratégia de forma a permitir sua reprodução por outros pesquisadores;
- exige critérios claros para inclusão e exclusão do trabalhos identificados;
- especifica quais informações serão buscadas de cada trabalho e como serão qualificadas;
- é executada em três fases: planejamento, execução e publicação.

Na fase de planejamento é verificada a existência de alguma revisão sistemática relacionado à área de pesquisa e com critérios semelhantes de avaliação. Dessa forma, se houver trabalhos relacionados já publicados, estes deverão ser utilizados. Por outro lado, na ausência deles, um protocolo de revisão deve ser elaborado para permitir a realização da revisão de maneira imparcial, sem vícios e livre de direcionamentos frutos de expectativas individuais dos pesquisadores. De maneira geral, nesse protocolo são definidos a questão de pesquisa, a estratégia de busca incluindo os termos e fontes de buscas, os critérios de seleção dos trabalhos, a avaliação de qualidade dos trabalhos incluídos, o conjunto de informações a ser obtido de cada item selecionado, a síntese dos dados obtidos e um cronograma de execução.

A questão de pesquisa destaca os principais tópicos a serem investigados e sua estrutura considera os seguinte pontos:

- População: quem ou qual será foco da pesquisa;
- Intervenção: aspectos específicos a serem pesquisados;
- Resultados: o que se deseja atingir;
- Contexto: quais são as circunstâncias mais importantes da pesquisa.

Na fase de execução e com o protocolo já definido, a revisão sistemática é realizada em cinco etapas. Primeiramente, a estratégia de busca é definida pela escolha dos termos (palavras-chaves) e fontes de pesquisa (periódicos, conferências etc). De posse dos trabalhos obtidos, na segunda fase é realizada a seleção dos estudos seguindo os critérios de inclusão e exclusão, identificando aqueles potencialmente relevantes à pesquisa. Em seguida, os itens escolhidos são qualificados seguindo um critério definido pelo pesquisador a fim de organizá-los de acordo com importância e relevância. Nessa terceira fase de avaliação de qualidade, o critério a ser seguido é composto, preferencialmente, por um conjunto de perguntas respondidas por sim ou não. Conseqüentemente, um número maior de respostas afirmativas implica em um grau maior de qualidade. Na quarta etapa, dados específicos são obtidos de cada estudo buscando responder as questões de pesquisa. Por fim, os dados são sumarizados em forma de gráficos ou tabelas.

Na fase de publicação, os resultados da revisão sistemática podem ser compartilhados tanto na forma de um trabalho apresentado em periódicos e conferências quanto como parte de uma dissertação ou tese.

A fim de resumir a ordem cronológica das tarefas necessárias para a realização de uma revisão sistemática, a Tabela A.1 apresenta as atividades que foram realizadas na aplicação da técnica nessa dissertação.

Tabela A.1: Fases e atividades de uma revisão sistemática.

Fase	Atividades
Planejamento	Identificar a necessidade de revisão
	Especificar questão de pesquisa
	Desenvolver o protocolo da revisão
Execução	Identificar pesquisa
	Selecionar os estudos primários
	Avaliar a qualidade dos estudos (Proximidade com a questão de pesquisa)
	Extrair e monitorar os dados
	Sintetizar os dados
Publicação	Especificar mecanismo de disseminação
	Formatar o relatório principal
	Validar o relatório

## A.1 Aplicação

Seguindo o roteiro sugerido por [28] e apresentado na Tabela A.1, a primeira atividade realizada foi a identificação da necessidade de uma revisão na área de pesquisa de planejamento multiagente. Nesse sentido, foram realizadas buscas em veículos de publicação

de trabalhos científicos como *ACM Digital Library*, *IEEEExplore Digital Library*, *Springer*, *Google Scholar*.

Cabe ressaltar que visando abranger a maior quantidade possível de trabalhos, os termos utilizados nas buscas foram definidos na língua inglesa, uma vez que grande parte dos trabalhos são publicados nesse idioma.

A questão central da revisão sistemática bem como sua estrutura foram definidas conforme apresentado a seguir:

- Questão de pesquisa: Quais são os efeitos da execução de um pré-planejamento para um planejamento multiagente?
- População: *Multi-agent Systems*, *Automated Planning*, *Multi-agent Planning*;
- Intervenção: modelos, algoritmos e abordagens que tratam de seleção de agentes, validação e verificação em planejamento multiagente;
- Resultados: volume menor de mensagens trocadas entre os agentes e tempo gasto durante o processo de planejamento em comparação com trabalhos que configuram o estado da arte de planejamento multiagente;
- Contexto: *Multi-agent Systems*, *Automated Planning*, *Multi-agent Planning*.

Com a questão já definida, a definição do protocolo de pesquisa foi iniciada pela estratégia de busca. O termo, as fontes e os filtros utilizados nas pesquisas são apresentados na Tabela A.2. O termo “*multi-agent planning*” foi aplicado em qualquer campo, não sendo limitado ao título ou resumo dos trabalhos, incluídos em qualquer parte do texto do trabalho. Além disso, as buscas foram realizadas durante o período de Fevereiro a Junho de 2017.

A condição para inclusão dos resultados das pesquisas foi definida por dois critérios: domínio de aplicação e menção do veículo de publicação. Para ser considerado relevante, o trabalho deveria apresentar modelos ou algoritmos independentes de domínio e aplicáveis em planejamento multiagente. De maneira complementar, apenas os artigos publicados em veículos com índices QUALIS CAPES maiores ou iguais a B2 foram incluídos.

Dessa forma, os trabalhos que não focassem em planejamento multiagente ou que fossem dependentes de domínio de aplicação foram excluídos. Essa condição foi necessária para ignorar trabalhos com foco em cenários restritos como aplicação em tráfego aéreo, manutenção da privacidade das informações dos agentes, composição de *web services* e planejamento de rotas (*path planning*). Portanto, trabalhos com maior abrangência e independente de domínio foram priorizados.

Em relação aos dados a serem obtidos de cada estudo incluído destacam-se o ano e o veículo de publicação, o índice de qualidade (QUALIS CAPES) do veículo, a principal contribuição do trabalho e o tipo de acoplamento, isto é, fraco ou forte.

Na fase de execução, foram realizadas buscas nas fontes de pesquisas de acordo com repositórios e filtros destacados na Tabela A.2. O resultado final da atividade de identificação de pesquisa é resumido na Tabela A.3, com os valores organizados por fonte.

---

<sup>1</sup><http://dl.acm.org/>

<sup>2</sup><http://ieeexplore.ieee.org/Xplore/home.jsp>

<sup>3</sup><https://link.springer.com/>

<sup>4</sup><http://www.aaai.org/Library/ICAPS/icaps-library.php>

Tabela A.2: Protocolo de pesquisa: estratégia de busca.

Termo	<i>“multi-agent planning”</i>
Fontes	<i>ACM Digital Library</i> <sup>1</sup>
	<i>IEEE Xplore Digital Library</i> <sup>2</sup>
	<i>Springer Link</i> <sup>3</sup>
	<i>International Conference on Automated Planning and Scheduling (ICAPS)</i> <sup>4</sup>
	Outras - a partir de referências importantes de outros trabalhos
Filtros	Artigos
	Periódicos
	Publicados entre 2011 e 2017

Tabela A.3: Identificação de pesquisa.

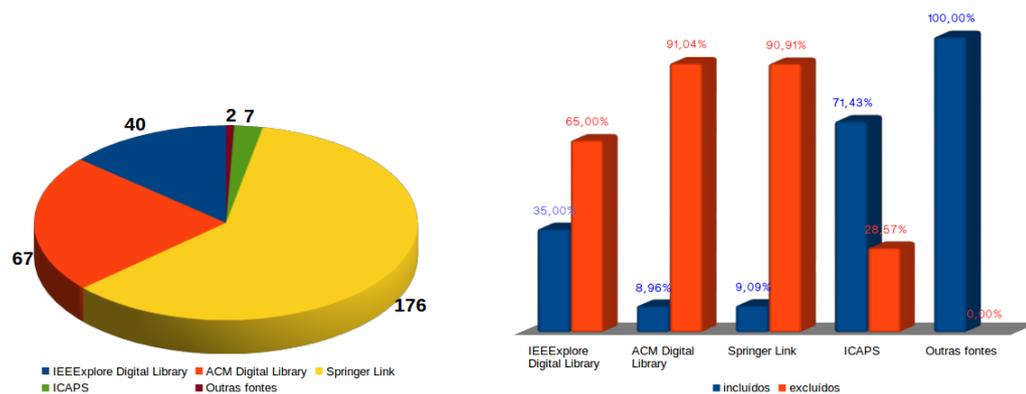
Fonte	Incluídos		Excluídos		Total	
<i>IEEE Xplore Digital Library</i>	14	32,56%	26	10,44%	40	13,7%
<i>ACM Digital Library</i>	6	13,95%	61	24,5%	67	22,94%
<i>Springer Link</i>	16	37,21%	160	64,26%	176	60,27%
ICAPS	5	11,63%	2	0,8%	7	2,4%
Outras fontes	2	4,65%	0	0%	2	0,7%
Geral	43	100%	249	100%	292	100%

Realizando uma análise estatística dos resultados da Tabela A.3 é possível perceber que apesar de fornecer mais de 60% dos trabalhos (176 de um total de 292), a fonte *Springer Link* contribui apenas com 37,21% do total itens incluídos. Por outro lado, a fonte *IEEE Xplore Digital Library* que forneceu 13,7% dos trabalhos, contribui com 32,56% na fase de inclusão. Os maiores índices de aceitação foram obtidos da fonte *ICAPS* e outras fontes, fato que já era esperado pois configuram bases de busca dedicadas à área de planejamento. Os detalhes dessa análise estão apresentados na Figura A.1, na qual são apresentados o total de itens resultantes da busca em cada fonte (Figura A.1(a)), a porcentagem de inclusão e exclusão de cada fonte (Figura A.1(b)) e a contribuição final de cada fonte para o trabalhos incluídos (Figura A.1(c)).

Para a atividade seguinte de avaliação da proximidade dos trabalhos com a questão de pesquisa definida nesta dissertação, apesar do total de trabalhos incluídos ser de 43, a quantidade de trabalhos que foram avaliados de acordo com o critério de qualidade foi de 35. A razão pela diferença foi o fato de oito trabalhos resultantes da busca na base da *Springer Link* estarem parcialmente disponíveis, isto é, apresentavam basicamente o resumo. Os outros trabalhos, mesmo apresentando a mesma limitação, estavam disponíveis em outras plataformas o que permitiu as avaliações.

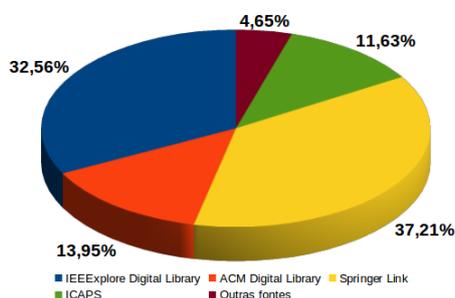
Em relação ao critério de qualidade utilizado para avaliar os trabalhos incluídos, foi elaborado um conjunto de seis perguntas cujas respostas eram sim ou não, listadas na Tabela A.4. A menção final de cada trabalho era calculada através da soma de respostas afirmativas. A Tabela A.5 apresenta os 35 trabalhos incluídos e destaca aqueles cujas menções finais são maiores ou iguais ao valor três. Esse valor foi estipulado a partir da média das menções dos trabalhos que apresentaram pelo menos uma resposta positiva, ou seja,  $\lceil \frac{42}{19} \rceil$ .

Os trabalhos que foram destacados com as maiores proximidades com a questão de



(a) Trabalhos identificados por fonte.

(b) Inclusão e exclusão por fonte.



(c) Contribuição para trabalhos incluídos.

Figura A.1: Análise dos resultados de identificação de pesquisa.

pesquisa tiveram seus dados extraídos, segundo a Tabela A.6 e ordenados pelo índice QUALIS CAPES<sup>1</sup>. Nessa sumarização, a principal motivação foi identificar qual é a principal contribuição do trabalho e se assume algum tipo de restrição quanto ao nível de acoplamento. Assume-se como hipótese que trabalhos que podem ser utilizados independente do tipo de domínio, seja ele fraca ou fortemente acoplado, oferecem uma maior possibilidade de uso.

Tabela A.4: Conjunto de perguntas do critério de avaliação.

Item	Pergunta
Q1	O estudo discute a corretude e/ou completude do algoritmo/modelo proposto?
Q2	O estudo discute a escalabilidade do modelo?
Q3	O estudo apresenta avaliação de resultados e comparação com outros trabalhos?
Q4	Os teste realizados utilizam os domínios do IPC?
Q5	O modelo utiliza um linguagem padrão (PDDL, ADL, STRIPS)?
Q6	Existe alguma implementação executável disponível para reproduzir os resultados?
Q7	Existe alguma seleção de agentes antes do planejamento?
Q8	Algum tipo de verificação do problema é realizada antes do planejamento?

A partir da análise das respostas positivas referentes ao critério de avaliação descrito na Tabela A.4 é possível destacar algumas observações sobre o estado da arte da área

<sup>1</sup>Plataforma Sucupira da CAPES - <https://sucupira.capes.gov.br>

de planejamento multiagente. Discussões sobre corretude e completude dos algoritmos e escalabilidade do modelo são raramente apresentadas nos trabalhos, atingindo coberturas de 11.43% e 17.14%, respectivamente.

Considerando os aspectos de avaliação de desempenho, um em cada cinco propostas apresenta uma comparação com outros trabalhos, o que possibilita a discussão das contribuições, vantagens e desvantagens. Os maiores níveis de cobertura estão relacionados aos casos de testes e à linguagem utilizada, indicando que 25.71% dos trabalhos utilizam *benchmarks* do IPC [37] e que 31.43% utilizam alguma linguagem padrão para descrever o problema de planejamento. Especialmente no aspecto de linguagem, os demais trabalhos ou não explicitavam qual padronização seguiam ou realizam extensões nas linguagens definindo termos próprios para marcar alguns tipos de informações.

Dos itens incluídos, apenas dois trabalhos disponibilizam (na Internet) implementações de seus modelos de forma a permitir a reprodução dos experimentos, execução de novos testes e principalmente comparação de resultados [8], [10].

Por fim, etapas de seleção de agentes ou verificação do problema antes do início do processo de planejamento é uma lacuna não preenchida, tendo sido detectada apenas em dois trabalhos [26], [38].

Os resultados consolidados na atividade de sumarização foram utilizados para definir os principais trabalhos da área de planejamento multiagente. Dessa forma, a fase de publicação da revisão sistemática da literatura foi caracterizada pelo detalhamento dos trabalhos correlatos presentes nessa dissertação.

Portanto, no Capítulo 3 foi dada ênfase aos trabalhos que obtiveram as maiores menções de qualidade. Esse conjunto de propostas forma a base dos estudos que motivaram e justificaram as decisões tomadas para definição do modelo proposta nessa dissertação. No entanto, o fato de uma proposta, referenciada por outros trabalhos, não pertencer a tal conjunto não exclui a possibilidade nem a necessidade de avaliá-la, mesmo que de maneira secundária.

Como resultado paralelo da revisão sistemática foi possível sugerir uma taxonomia de classificação dos trabalhos correlatos, destacando características importantes como exploração de técnicas de execução paralela, avaliação do problema antes do início do planejamento, seleção de agentes e tipo de abordagem de planejamento, isto é, centralizada ou distribuída.

Tabela A.5: Avaliação de qualidade dos trabalhos incluídos.

Item	Trabalho	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Total
1	A Collaborative Multiagent Framework Based on Online Risk-Aware Planning and Decision-Making [39]	N	N	N	N	N	N	N	N	0
2	A decentralized approach to multi-agent planning in the presence of constraints and uncertainty [40]	N	N	N	N	N	N	N	N	0
3	A flexible coupling approach to multi-agent planning under incomplete information [41]	N	S	N	N	N	N	N	N	1
4	A Formal Analysis of Required Cooperation in Multi-Agent Planning [26]	S	N	S	N	S	N	S	S	5
5	Allocating Social Goals Using the Contract Net Protocol in Online Multi-agent Planning [34]	N	N	N	N	N	N	N	N	0
6	An extension of FMAP for joint actions [42]	N	N	N	N	N	N	N	N	0
7	Argumentation Schemes for Collaborative Planning [43]	N	N	N	N	N	N	N	N	0
8	A Two-Stage Online Approach for Collaborative Multi-agent Planning Under Uncertainty [44]	N	N	N	N	N	N	N	N	0
9	Defeasible Argumentation for Multi-agent Planning in Ambient Intelligence Applications[45]	N	N	N	N	S	N	N	N	1
10	Divide & conquer in planning for self-optimizing mechatronic systems - A first application example [46]	N	N	N	N	N	N	N	N	0
11	Find a Multi-agent planning solution in nondeterministic domain [47]	N	N	N	N	N	N	N	N	0
12	FMAP: Distributed cooperative multi-agent planning [8]	S	S	S	S	N	S	N	S	5
13	Global Heuristics for Distributed Cooperative Multi-Agent Planning [48]	N	S	S	N	N	N	N	N	2
14	Group planning with time constraints [49]	N	N	N	N	N	N	N	N	0
15	Hybrid mission planning with coalition formation[50]	N	N	N	S	N	N	N	N	1
16	Hybrid Multi-agent Planning [51]	N	N	N	S	N	N	N	N	1
17	Increased Privacy with Reduced Communication in Multi-Agent Planning[52]	S	N	N	S	N	N	N	N	2
18	Lagrangian Relaxation for Large-Scale Multi-agent Planning [53]	N	S	N	N	N	N	N	N	1
19	MAPJA: Multi-agent planning with joint actions[30]	S	N	S	N	S	N	S	N	4
20	Multi-agent A* for Parallel and Distributed Systems [54]	N	N	N	N	N	N	N	N	0
21	Multiagent Argumentation for Cooperative Planning in DeLP-POP [55]	N	N	N	N	N	N	N	N	0
22	Multi-agent Planning by Plan Reuse [5]	N	N	S	S	S	N	N	N	3
23	Multi-agent planning for fleet cooperative air defense decision-making [56]	N	S	N	N	S	N	N	N	2
24	Multi-agent planning with joint action [57]	N	N	N	N	S	N	N	N	1
25	Multi-agent planning with quantitative capability[58]	N	N	N	S	S	N	N	N	2
26	On the Construction of Joint Plans Through Argumentation Schemes[59]	N	N	N	N	N	N	N	N	0
27	Potential Heuristics for Multi-Agent Planning[60]	N	N	N	S	S	N	N	N	2
28	Robust Plan Execution in Multi-agent Environments[61]	N	N	N	N	N	N	N	N	0
29	Scalable, MDP-based planning for multiple, cooperating agents[62]	N	N	N	N	N	N	N	N	0
30	Secure Multi-Agent Planning[63]	N	N	N	N	N	N	N	N	0
31	Social Continual Planning in Open Multiagent Systems: A First Study [64]	N	N	N	N	N	N	N	N	0
32	Transforming Multi-agent Planning Into Single-Agent Planning Using Best-Cost Strategy[7]	N	S	S	N	S	N	N	N	3
33	Validation and Verification of Joint-Actions in Multi-agent Planning[38]	N	N	N	N	N	N	N	S	1
34	Psm-based planners description for codmap 2015 competition[65]	N	N	N	S	S	N	N	N	2
35	MADLA: planning with distributed and local search[10]	N	N	S	S	S	S	N	N	4

Tabela A.6: Sumarização dos principais trabalhos.

Trabalho	Publicação		Índice QUALIS	Contribuição	Acomplamento
	Ano	Veículo			
MADLA: planning with distributed and local search ([10])	2015	AAAI	A1	Planejador multiagente distribuído ou centralizado	Ambos
Multi-agent Planning by Plan Reuse ([5])	2013	AAMAS	A1	Distribuição de objetivos entre os agentes	Fraco
A Formal Analysis of Required Cooperation in Multi-Agent Planning ([26])	2016	ICAPS	A2	Algoritmo para determinar se é preciso haver cooperação entre agentes para atingir objetivos	Ambos
FMAP: Distributed cooperative multi-agent planning ([8])	2014	<i>Applied Intelligence</i>	B1	Planejador independente de domínio utilizando a técnica de refinamento de planos	Ambos
MAPJA: Multi-agent planning with joint actions ([30])	2017	Applied Intelligence	B1	Planejador multiagente independente de domínio e capaz de resolver ações conjuntas	Ambos
Transforming Multi-agent Planning Into Single-Agent Planning Using Best-Cost Strategy ([7])	2016	BRACIS	B2	Análise do ganho com distribuição de objetivos	Fraco