

**ALOCAÇÃO DE TAREFAS PARA TRANSPORTE
DE OBJETOS COM ROBÔS HETEROGÊNEOS**

RAMON SOARES DE MELO

**ALOCAÇÃO DE TAREFAS PARA TRANSPORTE
DE OBJETOS COM ROBÔS HETEROGÊNEOS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais - Departamento de Ciência da Computação como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: DOUGLAS GUIMARÃES MACHARET
COORIENTADOR: MARIO FERNANDO MONTENEGRO CAMPOS

Belo Horizonte

Agosto de 2015

© 2015, Ramon Soares de Melo.
Todos os direitos reservados.

Melo, Ramon Soares de.

M528a Alocação de Tarefas para Transporte de Objetos com Robôs Heterogêneos / Ramon Soares de Melo. — Belo Horizonte, 2015
xiii, 75 f. : il. ; 29cm

Dissertação (Mestrado) — Universidade Federal de Minas Gerais - Departamento de Ciência da Computação

Orientador: Douglas Guimarães Macharet

Coorientador: Mario Fernando Montenegro Campos

1. Computação – Teses. 2. Robôs – Movimento – Teses. 3. Robótica – Teses. 4. Materiais – Manuseio. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*82.10 (043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Alocação de tarefas para transporte de objetos com robôs heterogêneos

RAMON SOARES DE MELO

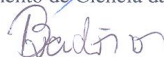
Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:



PROF. DOUGLAS GUIMARÃES MACHARET - Orientador
Departamento de Ciência da Computação - UFMG



PROF. MARIO FERNANDO MONTENEGRO CAMPOS - Coorientador
Departamento de Ciência da Computação - UFMG



PROF. BRUNO VILHENA ADORNO
Departamento de Engenharia Elétrica - UFMG



PROF. PEDRO MITSUO SHIROMA
Departamento de Ciência da Computação - UFSJ



PROF. LUIZ CHAIMOWICZ
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 28 de agosto de 2015.

Resumo

A aplicação de sistemas autônomos pode ser observada em diversos cenários, como em atividades de vigilância, busca e resgate, rastreamento e mapeamento, além destas, o transporte e manipulação de objetos é um tipo de missão que tem ganhado atenção por parte de pesquisadores e grandes empresas, principalmente por acrescentar à atividade características como precisão, agilidade e controle, além de poder garantir melhoras de eficiência geral do sistema.

Neste trabalho é demonstrado um conjunto de técnicas utilizadas para a coordenação de uma equipe de agente robóticos heterogêneos, tendo como missão o transporte de objetos utilizando suas capacidades físicas e computacionais. O processo de transporte e manipulação é tratado em três etapas: (i) planejamento de caminhos para os objetos, na qual um plano de movimentação do objeto é criado baseado em métricas de uso energético e tempo; (ii) planejamento e alocação de tarefas entre os agentes disponíveis para o transporte, realizando a distribuição de tarefas de modo a minimizar o custo total do sistema; (iii) coordenação e execução, na qual os agentes realizam o transporte de forma sincronizada.

Foram executados experimentos quantitativos, demonstrando a efetividade do método avaliado em diversos cenários distintos, além de testes simulados e reais, que ilustram a manipulação dos objetos por agentes robóticos propriamente ditos. Desta maneira, é demonstrado uma metodologia capaz de executar de forma completa a missão de transporte, aprimorando-a através do uso de agentes inteligentes.

Palavras-chave: Transporte Cooperativo, Manipulação de Objetos, Alocação de Tarefas.

Abstract

The application of autonomous systems can be observed in various scenarios, such as surveillance, search and rescue, tracking and mapping, apart from these, transportation and manipulation of objects is a kind of mission that has gained attention from researchers and large companies, mainly by adding to the activity features like precision, speed and control, in addition to ensuring overall efficiency improvements to the system.

This work demonstrated a set of techniques used to coordinate a team of heterogeneous robotic agents, whose mission is to transport objects using their physical and computational skills. The process of transport and manipulate is handled in three stages: (i) object path planning, in which a plan is created based on metrics of energy use and cost of time; (ii) planning and task allocation between the available agents for transportation, making the distribution of tasks in order to minimize the total utility of the system; (iii) coordination and execution, in which agents perform transport synchronously.

Were performed quantitative experiments, demonstrating the effectiveness of the method valued in several different scenarios, as well as simulated and real tests, illustrating the manipulation of objects by robotic agents themselves. Thus, it is shown a methodology capable of performing a full transport mission, improving it through the use of intelligent agents.

Keywords: Cooperative Transport, Object Manipulation, Task Allocation.

Lista de Figuras

1.1	Aplicações de Robôs para Transporte e Manipulação de Objetos	4
1.2	Uma solução do problema de transporte de objetos	6
2.1	Conjunto das principais técnicas não-preênsil utilizadas para o transporte de objetos.	9
2.2	Modelos de disposição de agentes para o transporte de objetos	11
2.3	Planos de movimentação de um objeto a ser transportado	13
2.4	Transporte realizado por um quadrotor que utiliza uma garra para segurar o objeto	16
3.1	Diagrama geral da Metodologia do trabalho	22
3.2	Planejamento e Segmentação do plano de movimentação de um objeto	32
3.3	Planos de movimentação dos agentes	34
3.4	Ambiente de exemplo com 2 objetos e 3 robôs	37
3.5	Exemplo de Coordenação entre 2 Agentes durante o transporte de objetos	42
4.1	Mapa de exemplo do transporte de um objeto por um único agente	44
4.2	Cenários utilizados nos experimentos relacionados à complexidade do ambiente	46
4.3	Gráfico do deslocamento do objeto em ambiente com diferentes complexidades. São exemplificados três amostras onde os agentes são dispostos em posições diferentes.	47
4.4	Gráfico de deslocamento dos agentes em cenários com variados graus de restrição de movimentos. Cada amostra demonstrada, representa cenários com agentes em posições distintas.	47
4.5	Gráfico do tempo médio necessário para o planejamento total do sistema	47
4.6	Mapa utilizado nos testes de deslocamento e tempo de planejamento dos agentes	48

4.7	Deslocamento de um agente sem a aplicação da penalização de mudança de direção	49
4.8	Deslocamento de um agente utilizando a penalização de mudança de direção	50
4.9	Deslocamento total dos agentes com a aplicação da penalidade de mudança de direção.	50
4.10	Tempo de planejamento dos agentes sem a aplicação da penalidade de mudança na direção.	51
4.11	Tempo de planejamento dos agentes com a aplicação da penalidade de mudança na direção.	51
4.12	Exemplos de planos de movimentação de um objeto com diferentes pesos na função de utilidade	53
4.13	Resultado do transporte de 3 objetos por um agente	54
4.14	Mapa utilizado nos testes sequencial e oportunista	55
4.15	Deslocamento do robô mediante nos transportes sequencial e oportunista .	55
4.16	Tempo de planejamento do agente nos transporte sequencial e oportunista	56
4.17	Ambientes de trabalho utilizados durante os experimentos simulados. . . .	57
4.18	Exemplo de transporte por dois agentes em um ambiente simulado.	58
4.19	Exemplo de transporte utilizando um agente terrestre e um aéreo.	59
4.20	Plataforma robótica ePuck utilizada durante os experimentos reais.	63
4.21	Configuração do ambiente de trabalho no experimento real.	63
4.22	Exemplo de transporte de um objeto utilizando três agentes reais.	64

Lista de Tabelas

2.1	Tabela demonstrativa das pesquisas no campo do transporte de objetos elencando as principais áreas de atuação.	21
3.1	Descrição dos elementos utilizados nos diagramas de exemplificação.	25
4.1	Tabela com as variações nos pesos da função de utilidade	52
4.2	Dados referente às características dos tipos de agentes	53

Lista de Algoritmos

3.1	AStar	28
3.2	ExpansionAlgorithm	29
3.3	SegmentPointSetCreation	31
3.4	SegmentSetCreation	31
3.5	TaskAllocationProcess	35
3.6	CoordinationProcess	39
3.7	ResponseTokenProcess	40
4.1	AerialVelocityController(goal_pose, last_cmd, vertical_lim, P, I)	60
4.2	AerialVelocityController(goal_pose, last_cmd, angular_lim, P, I)	61

Lista de Símbolos

- \mathcal{W} Ambiente de trabalho utilizado para realização do transporte de objetos.
- c Célula que descreve uma posição do ambiente de trabalho \mathcal{W} .
- \mathcal{O} Conjunto de objetos a serem transportados.
- \mathcal{B} Conjunto de obstáculos dispostos em \mathcal{W} .
- \mathcal{R} Conjunto de robôs disponíveis para executar o transporte.
- Θ Função de utilidade utilizada na comparação de planos.
- Υ Função que descreve a velocidade de movimentação de robô r_i em \mathcal{W} .
- Ψ Função que descreve a energia gasta por um agente r_i durante sua movimentação.
- Θ_p Valor de utilidade de um plano.
- \mathcal{M} Conjunto de ações que objetos e agentes conseguem realizar em \mathcal{W} .
- a_i Representa uma ação do conjunto \mathcal{M} .
- \mathcal{P} Conjunto $\mathcal{P} = \{S_o, \dots, S_d\}$, representa um plano de movimentação.
- \mathcal{P}_ε Conjunto de planos \mathcal{P} utilizados durante a missão de transporte.
- S_i Estrutura que descreve informações sobre uma célula c , além de dados relacionados a um plano \mathcal{P} .
- \mathcal{P}_g Grafo utilizado na busca e criação de um plano \mathcal{P} de movimentação.
- \mathcal{F} Lista de células c que estão na fronteira da exploração do algoritmo de planejamento de caminhos.
- \mathcal{S} Conjunto de segmentos gerados a partir de um plano de movimentação.
- \mathcal{T}_M Conjunto de tipos de planos gerados para os agentes.

Sumário

Resumo	v
Abstract	vi
Lista de Figuras	vii
Lista de Tabelas	ix
Lista de Símbolos	xi
1 Introdução	1
1.1 Contextualização	1
1.2 Motivação	2
1.3 Problema	5
1.4 Contribuições	6
1.5 Organização do Trabalho	7
2 Trabalhos Relacionados	8
2.1 Manipulação de Objetos	8
2.1.1 Manipulação Não-preênsil	8
2.1.2 Manipulação Preênsil	12
2.2 Alocação de Tarefas e Coordenação	17
2.3 Abordagem Proposta	19
2.4 Comparativo	20
3 Metodologia	22
3.1 Fundamentos e Formalização	23
3.1.1 Notação gráfica	25
3.2 Função de Utilidade	26
3.3 Planejamento de Caminhos - Objeto	27

3.3.1	Planejamento	27
3.3.2	Segmentação	30
3.4	Planejamento de Caminhos - Agentes	32
3.5	Alocação de Tarefas	33
3.6	Coordenação	38
4	Experimentos	43
4.1	Estudo Quantitativo	44
4.1.1	Complexidade do Ambiente	45
4.1.2	Deslocamento e Tempo de Planejamento dos Agentes	48
4.1.3	Função de Utilidade	52
4.1.4	Sequenciamento do Transporte	54
4.2	Ambiente Simulado	56
4.2.1	Ambiente Real	62
5	Conclusão e Trabalhos Futuros	65
5.1	Conclusão	65
5.2	Trabalhos Futuros	67
	Referências Bibliográficas	69

Capítulo 1

Introdução

1.1 Contextualização

A Robótica é um campo de pesquisa multi-disciplinar, que envolve desde mecânica e eletrônica a algoritmos de inteligência artificial, visão computacional, aprendizado de máquina, teoria de grafos, dentre outras, oriunda de áreas como a engenharia mecânica, elétrica e ciência da computação. Tem como foco o estudo e desenvolvimento de máquinas capazes de auxiliar seres humanos em diversos cenários. Estas máquinas podem possuir um certo nível de inteligência, podendo tomar decisões de modo a completar as atividades que foram empregados.

No princípio, os robôs foram utilizados principalmente no âmbito industrial, no qual exerciam o ofício de manipulação e construção de itens em linhas de montagem (veículos, empacotamento), onde se encontravam fixos ao ambiente de trabalho, possuindo limitada área de atuação, com grande acurácia de operação (Siegwart et al. [2011]). Porém, com o passar dos anos, os mesmos deixaram de ser somente manipuladores de base fixa e passaram a poder mover-se pelo ambiente, atingindo maiores áreas e ganhando outras aplicações. Estes foram adotados em funções como as de vigilância e reconhecimento de regiões (Tanner [2007]; Sujit & Saripalli [2013]), busca e resgate no caso de acidentes ou catástrofes (Casper & Murphy [2003]; Murphy [2004]), mapeamento (Tokekar et al. [2013]), detecção e rastreamento de alvos (Grocholsky et al. [2006]) e transporte de objetos (Michael et al. [2011]; Fink et al. [2008]).

Esta pluralidade de tarefas foi desenvolvida e melhorada com o uso de sistemas autônomos, agregando características como precisão, eficiência e eficácia em inúmeros aspectos (uso energético, diminuição do tempo de execução, custo financeiro, entre outros).

Uma das categorias nas quais o uso deste tipo de agente podem ser classificados

diz respeito à quantidade de robôs empregados na tarefa, sendo estes de dois tipos: (i) sistemas de único robô (*Single Robot System* [SRS]) – no qual um robô é responsável por toda a missão e deve possuir todos os recursos necessários para cumpri-la, e (ii) sistemas multi-robôs (*Multi Robot System* [MRS]) – onde um grupo de agentes, reunindo suas capacidades, exerce a função requerida em conjunto, de forma coordenada, aproveitando os recursos da equipe.

Comparativamente, um MRS apresenta algumas vantagens em relação à utilização de um SRS, como uma maior robustez, menor taxa de falhas, além de diminuir em muitos casos o tempo necessário para realização da tarefa. Uma vantagem notável do uso de times está na sua capacidade de realizar atividades complexas, como as descritas anteriormente, sem a necessidade de ser formado por agentes especializados para a tarefa, por exemplo, usando robôs mais simples para transportar um objeto ao invés de um único com um manipulador de alta tecnologia.

Outra particularidade somente encontrada em um MRS, é que este pode atuar em ambientes nocivos ao ser humano, assim, poupando-o de qualquer acidente ou perigo que poderia ocorrer. Exemplo este, é a utilização de grupos de busca e mapeamento aplicados em acidentes nucleares, como o ocorrido em Fukushima¹, utilizados para estudar o cenário após o desastre ocorrido, e fornecer informações à equipe humana a fim reduzir as consequências do mesmo. Outros ambientes, como a inspeção de vulcões, locais com incêndio/fumaça, as profundezas do oceano são territórios indicados para adoção de robôs.

Ao comparar o desempenho de algumas atividades sendo realizadas por agentes humanos e por agentes autônomos, é perceptível que a aplicação de um grupo de robôs pode agregar à tarefa qualidades (rapidez, acurácia, eficiência) que não poderiam ser alcançadas ou que seriam de grande custo para um time humano.

1.2 Motivação

O transporte e manipulação de objetos, apesar de transparecer ser uma tarefa simples, é de fundamental importância em diversas outras atividades, e requer o estudo de áreas como o planejamento de caminhos e a coordenação de agentes, quando é utilizado um MRS, por exemplo.

Dentro do campo de pesquisa sobre manipulação usando robôs, além da classificação dos agentes entre os tipos fixos ou móveis (Siegwart et al. [2011]), os mesmos podem ser também agrupados considerando as estratégias usadas para mover e/ou por-

¹<https://goo.gl/TJRYh9>

tar o objeto, são elas: (i) não-preênsil (*non-prehensile*) – na qual o agente manipulador utiliza de ações como arremessar, rolar e/ou empurrar para completar o transporte do objeto, e (ii) preênsil (*prehensile*) – na qual o objeto a ser transportado é acolhido ou segurado pelo manipulador, que o leva até a posição final desejada e o deposita (Lynch [1996]; Murray et al. [1994]).

A aplicabilidade de robôs capazes de manipular e transportar objetos tem crescido, principalmente buscando melhorar o rendimento e ganhos nas tarefas solicitadas. Exemplos práticos deste uso podem ser vistos em cenários como: retirada de sedimentos em uma mina (Murphy et al. [2009]), realização de cirurgias (Lehman et al. [2008]), transporte de objetos (Michael et al. [2011]), manipulação de mercadorias em um armazém (Guizzo [2008]), e na construção de bens e estruturas (Lindsey et al. [2012]; dos Santos et al. [2013, 2014]; Augugliaro et al. [2014]).

A Figura 1.1 demonstra aplicações reais de transporte de objetos utilizando o modelo MRS, considerando o uso de técnicas preênsil e não-preênsil. Na Figura 1.1a um grupo de agentes terrestres utiliza seu corpo para empurrar objetos por um ambiente (Fink et al. [2008]); a Figura 1.1b demonstra o sistema de transporte de mercadoria *Kiva* (Guizzo [2008]), utilizado por grandes empresas como *Amazon*², em seus armazéns. Utilizando um time de quadrotores, é possível construir estruturas baseado em segmentos pré-existentes (Lindsey et al. [2012]) ou com uso de blocos (Augugliaro et al. [2014]), visto na figuras 1.1c e 1.1d, respectivamente.

Em casos específicos, a realização da manipulação de objetos pode necessitar recursos que são indisponíveis ou que acarretariam em um aumento do custo total de execução se fosse realizado somente por um agente robótico, seja por que o mesmo não possui força mecânica suficiente ou pela tarefa necessitar de monitoramento externo durante, por exemplo.

Essencialmente, a utilização de uma equipe de agentes proporciona algumas vantagens, como: a diminuição no tempo necessário para realização de uma atividade, seja pela divisão das tarefas entre a equipe, seja pela execução de certas funções de forma conjunta; além de dispor de mais recursos (força física ou diversidade de capacidades), necessário para conclusão de algumas tarefas. Baseado nestas propriedades, o uso de um time robótico tem se mostrado uma opção interessante em diversos contextos.

Apesar das evidentes qualidades de um MRS, sua aplicação requer a adoção de técnicas mais complexas, principalmente no que se relaciona ao controle, coordenação e comunicação dos agentes, essencial para que a tarefa seja bem executada.

Questões fundamentais da robótica ainda se aplicam, porém em uma escala de

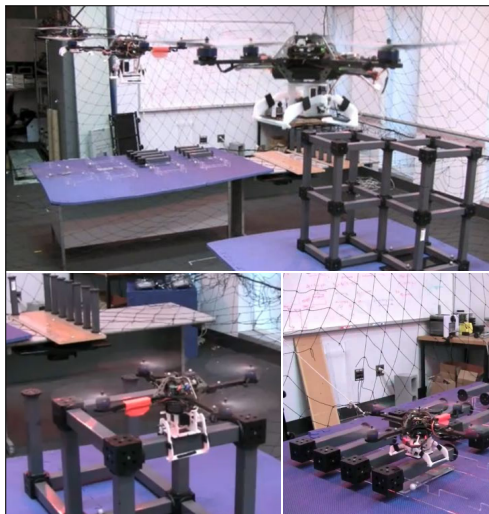
²<http://www.amazon.com/>



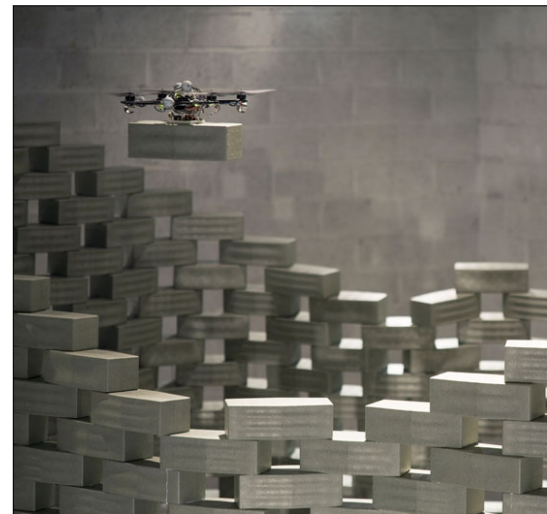
(a) Fink et al. [2008]



(b) Guizzo [2008]



(c) Lindsey et al. [2012]



(d) Augugliaro et al. [2014]

Figura 1.1: Aplicações de sistemas robóticos desempenhando tarefas de transporte e manipulação de objetos. (a) – um grupo de robôs utiliza seu corpo para mover objetos, (b) – sistema de transporte *Kiva*, (c) – equipe de robôs usa segmentos para criar estruturas 3D, (d) exemplo de construção usando blocos e quadrotores.

maior complexidade. Dentre esses problemas, destacam-se a localização, o planejamento de caminhos, a coordenação dos agentes e a própria forma em que a missão será realizada.

1.3 Problema

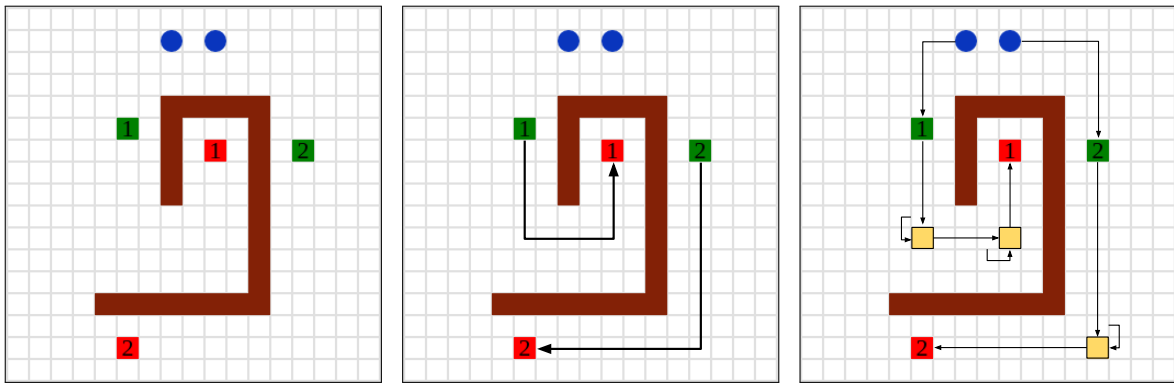
Este trabalho tem por objetivo apresentar uma metodologia para questões decorrentes do uso de uma equipe de agente robóticos, tendo como missão o transporte de objetos utilizando suas capacidades físicas e computacionais. O problema específico tratado é descrito a seguir:

Dado um grupo de agentes heterogêneos, executar a ação de transporte de um conjunto de objetos dispostos em um ambiente de trabalho discretizado, movendo-os de suas posições iniciais para suas respectivas posições finais previamente estipuladas, considerando os obstáculos presentes neste ambiente.

Para resolução do problema descrito, este trabalho apresenta uma proposta de solução mediante a criação de dois subproblemas, proporcionando uma melhor análise do mesmo, descritos a seguir:

Problema 1 Planejamento de Caminhos do Objeto: Considerando o ambiente de trabalho no qual deve ocorrer o transporte, executar o planejamento de uma sequência de posições para cada objeto partindo de sua posição inicial com a finalidade de movê-lo até uma posição final desejada. Este plano deve evadir demais objetos e obstáculos no ambiente, bem como considerar as capacidades dos agentes disponíveis. Na Figura 1.2b é demonstrado o resultado da solução deste problema, um caminho entre a pose inicial e a final.

Problema 2 Alocação de Tarefas e Coordenação de Agentes: Considerando que o plano de manipulação dos objetos foi criado, este sub-problema se concentra na criação de um plano de execução dos agentes, contemplando a movimentação e as ações que os mesmos devem realizar para que o transporte seja completado. Uma vez que o plano de execução tenha sido criado, os agentes devem ser coordenados de forma que suas tarefas individuais estejam em sinergia, propiciando a realização da tarefa. A Figura 1.2c demonstra um exemplo de coordenação para o transporte de um objeto, baseado no trajeto já determinado e dividindo as tarefas de acordo com os recursos de cada agente.



(a) Ambiente de trabalho para o transporte. (b) Solução do Problema 1 - Planejamento para os objetos. (c) Solução do Problema 2 - Planejamento e execução pelos agentes.

Figura 1.2: Solução do problema de transporte de objetos. Um mapa descrevendo o ambiente serve como ponto de entrada para o sistema (a), que na primeira etapa cria os planos de movimentação para os objetos (b), e posteriormente, os utiliza para criar planos para os agentes e coordená-los na execução (c).

1.4 Contribuições

Este trabalho apresenta técnicas para controle e coordenação de um conjunto de agentes robóticos incumbidos da realização das tarefas de manipulação e transporte de um conjunto de objetos.

O principal diferencial do trabalho apresentado é tratar o plano de movimentação do objeto a ser transportado como base para as demais etapas do processo, considerando que a maioria dos trabalhos que propõem soluções neste campo de pesquisa focam na movimentação dos agentes.

São apresentadas propostas nas diversas áreas envolvidas no processo de transporte: no planejamento de caminhos, no qual uma função de utilidade, que pondera dimensões como tempo de travessia e energia, é utilizada para encontrar o melhor plano, além de demonstrar estudos sobre a penalização das mudanças de direção dos objetos; a etapa de alocação de tarefas entre os agentes é realizada de forma descentralizada e coordenada; e finalmente, para a execução da missão é demonstrado um algoritmo baseado em estratégias de redes de computadores, para sincronizar e sistematizar os agentes durante a resolução da tarefa.

Parte desta metodologia, que se concentra no planejamento e transporte dos objetos juntamente com a técnica de penalização utilizando um agente, foi publicada na Conferência LARS 2015 – *12th Latin American Robotics Symposium* e SBR 2015 – 3º Simpósio Brasileiro de Robótica, sob o título *Multi-object Transportation using*

a *Mobile Robot* (Melo et al. [2015]), demonstrando que este tema é relevante para a comunidade pesquisadora de robótica.

1.5 Organização do Trabalho

O presente trabalho está organizado da seguinte forma, o Capítulo 2 apresenta o estado da arte nas diversas áreas que compreendem o mapeamento, planejamento e coordenação de robôs. No Capítulo 3 é apresentada a metodologia utilizada durante a criação e experimentação das soluções trabalhadas, seguido dos experimentos e validações realizadas no Capítulo 4. Por fim, concluímos no Capítulo 5 este trabalho com assertivas acerca da capacidade de aplicação da técnica apresentada bem como de sua importância perante outros trabalhos.

Capítulo 2

Trabalhos Relacionados

Ao considerar que o presente trabalho demonstra uma metodologia para a coordenação de uma equipe de agentes robóticos para o transporte de objetos, será apresentado neste capítulo uma coleção de trabalhos que expõem técnicas e estratégias que tratam de problemas similares, e que serviram de inspiração para o procedimento aqui explicitado.

O capítulo está dividido em duas sessões, as quais mais representam as sub-tarefas tratadas neste trabalho, são elas: (i) técnicas para a manipulação e transporte de objetos, (ii) mecanismos para controle e alocação de tarefas entre os agentes de uma equipe.

2.1 Manipulação de Objetos

A manipulação de objetos é um tema de bastante interesse dentro do campo de pesquisa da robótica, ao considerar que existem maneiras diferentes de realizar a mesma, cada uma com características próprias, agregando à tarefas qualidades e graus de complexidade diferentes. Como definido no Capítulo 1, as técnicas relacionadas ao manejo de objetos serão divididas em duas categorias: (i) não-preênsil e (ii) preênsil.

2.1.1 Manipulação Não-preênsil

As técnicas não-preênsil são aplicadas de forma geral em casos no qual o objeto a ser transportado é maior e/ou pesado, de modo que não possa ser erguido facilmente. Possui uma vantagem em relação às técnicas preênsil, no fato do agente manipulador não necessitar de um contato contínuo com o item a ser transportado.

Para a realização do transporte utilizando uma das técnicas citadas a seguir, em sua maioria, o agente deve aplicar uma força sobre o objeto, principalmente utili-

zando seu próprio corpo, de modo a empurrar o mesmo em uma determinada direção, nomeadas como técnicas de *PUSH*.

Tais técnicas foram desenvolvidas para a resolução do chamado problema *box-pushing* definido por Mataric et al. [1995], descrito como: dado um ambiente formado por itens poligonais rígidos, encontrar um caminho contínuo livre de obstáculos levando um objeto de uma configuração de origem para outra configuração desejada utilizando somente ações de *PUSH* para mover o mesmo. Como demonstrado por Reif [1979], este é um problema *PSACE-hard*, o que implica em ser *NP-hard*.

O estado da arte no âmbito da manipulação não-preênsil compreende quatro tipos de táticas de movimentação dos objetos: (i) Force Closure (FeC) – na qual os agentes impõem forças sobre o objeto de forma a segurá-lo fortemente, até que seja atingido um equilíbrio entre as mesmas, (ii) Form Closure (FmC) – nesta técnica os agentes circundam o objeto sem a necessidade de aplicar forças sobre o mesmo, (iii) Object Closure (OC) – similarmente à FmC, os agentes são arranjados próximos ao objeto, de forma que o objeto fica sempre preso, e (iv) Conditional Closure (CC) – onde os agentes aplicam forças sobre o objeto somente em um determinado lado, condicionados pela necessidade do sistema (Eoh et al. [2011]). Estas técnicas estão ilustradas na Figura 2.1.

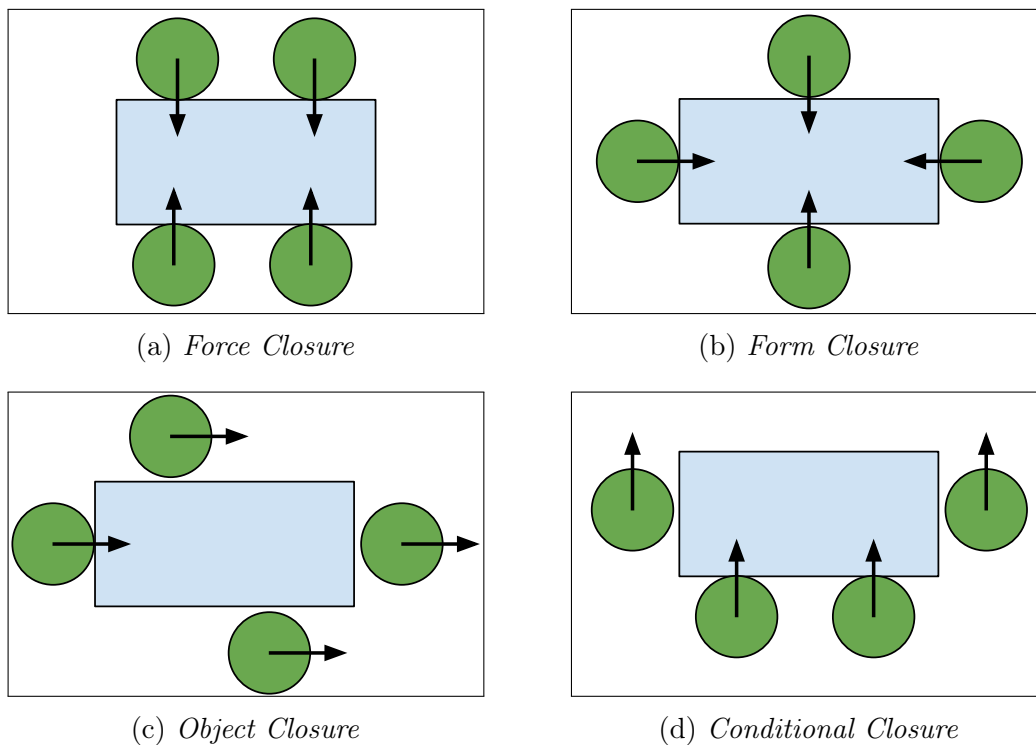


Figura 2.1: Conjunto das principais técnicas não-preênsil utilizadas para o transporte de objetos.

Diversos trabalhos aplicam estas técnicas de diferentes maneiras e com modificações específicas para cada caso, além de estudar particularidades das estratégias a fim de melhorá-las.

Intuitivamente, para realização do transporte de objetos, é necessário que os agentes se aproximem do objeto e depois realizem sua manipulação. Alguns trabalhos descrevem esse processo de diferentes maneiras, propondo soluções interessantes. O trabalho apresentado por Emery & Balch [2001] apresenta um controlador para robôs não holonômicos, que utiliza campos vetoriais para realizar o desvio de obstáculos. Para realizar o transporte, são criados três comportamentos para os agentes: *Search* – quando o robô está procurando pelo objeto a ser transportado, *Acquire* – na qual o agente se posiciona em um local específico de acordo com a posição do objeto e do destino do mesmo, e *Deliver* – quando o agente empurra o objeto levando-o para sua pose final. As tarefas executadas são ditas como de baixa cooperação, ou seja, cada agente executa uma tarefa sozinho com fim de contribuir para objetivo final de transporte. Técnicas similares são demonstradas em Song & Kumar [2002]; Wang & Kumar [2002]; Pereira et al. [2004]; Fink et al. [2008], nas quais ilustram um controle descentralizado, baseado em campos potenciais artificiais para navegação e desvio de obstáculos, sempre buscando utilizar os agentes para se aproximar (*approach*) e circundar (*surround*) o objeto em questão principalmente pelo uso de OC, depois são aplicadas forças que atraem os agentes para a configuração final e enfim transportam (*transport*) o item.

A fim de melhorar a formação e disposição na qual os agentes devem permanecer durante o transporte, Magariyama et al. [2014] apresenta a técnica chamada *Feasible Dynamic Caging Zone*, na qual possibilita que agentes durante o transporte por OC possam desviar de obstáculos ou mover o objeto para evitar uma colisão em ambientes dinâmicos, aprimorando a tarefa de forma geral.

Com uma modificação dos modelos CC e OC, Eoh et al. [2011] apresenta um sistema intitulado *pusher-puller formation* onde dois tipos de agentes, um capaz de empurrar e outra de puxar o objeto a ser transportado, trabalham em conjunto. São definidas três diferentes técnicas para realização da tarefa: (i) *straight line* – os agentes são posicionados atrás ou em frente ao objeto, formando uma linha reta, empurrando ou puxando o objeto respectivamente, (ii) *symmetrical formation* – similar ao modelo anterior, porém os agentes sempre estão juntos do objeto, aplicando forças diretamente no mesmo, e (iii) *pusher-puller* – que é o meio termo entre as estratégias anteriores, um dos agentes empurra o objeto enquanto o outro robô puxa-o. Tais modelos são exemplificados na Figura 2.2.

O trabalho de Neumann et al. [2014] também realiza um estudo sobre a formação e posicionamento dos agentes apresentando a técnica *cluster space control*, que descreve

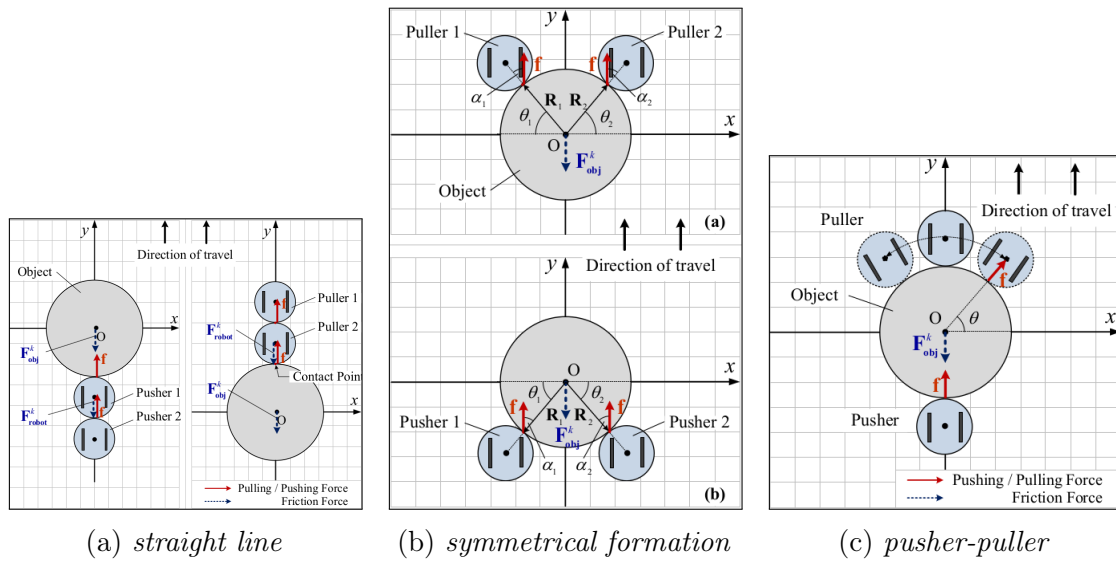


Figura 2.2: Modelos de disposição de agentes para o transporte de objetos apresentados por Eoh et al. [2011].

uma metodologia de modo a controlar a posição relativa entre robôs e objeto. Além disso, trata os agentes de transporte como atuadores que influenciam a movimentação do objeto, e estuda como a força aplicada sobre o mesmo determina sua trajetória e seu comportamento, salientando que comumente, trabalhos focam somente no estudo cinemático do transporte. Em consenso a este trabalho, Behrens et al. [2010] discute um modelo matemático que descreve o controle da magnitude e direção da força aplicada por um agente no objeto, resultando em seu transporte.

Outro método que estuda a influência do comportamento do agente perante o objeto é demonstrado por Meriçli et al. [2013], que faz uso de algoritmos de aprendizado de máquina para criar um planejador capaz de prever a movimentação de objetos munidos de baixo atrito com o ambiente, ou seja, que uma vez empurrados, deslizam por um período de tempo. A técnica envolve o treinamento do sistema com várias experiências de transporte, variando intensidade e tempo de duração da ação de *PUSH*. Baseado neste conhecimento, monta uma sequência de ações-reações esperadas para os objetos afim de transportá-los. O sistema é capaz de desviar de obstáculos e recalcular o plano se as reações encontradas não condizem com o plano previamente traçado.

Gerkey & Mataric [2002] demonstra um modelo onde existem dois tipos de agentes, o chamado *pusher*, que são capazes empurrar o objeto, enquanto o agente do tipo *watcher* observa e coordena durante o desempenho do transporte. Este é um exemplo do paradigma de coordenação *leader-follower* no qual o líder não faz parte diretamente da realização da missão, e faz uso do sistema de alocação de tarefas MURDOC (Gerkey & Mataric [2001]).

No trabalho de da Costa [2012] é demonstrado um sistema para transporte cooperativo no qual alguns agentes da equipe utilizam técnicas CC. A estratégia apresentada é capaz de estimar características dos objetos (tamanho e peso) a serem transportados por meio do uso de matrizes de ocupação, além de utilizar de métodos bio-inspirados e baseados em leilão para recrutamento e alocação de tarefas dentre os robôs da equipe.

Também guiado por comportamentos encontrados na natureza, Maghsoud et al. [2014] demonstra sua técnica de transporte de objetos embasado no sistema de imunidade humana, nomeada de *artificial immune system (AIS)*. Neste esquema, um time de agentes heterogêneos, possuindo diferentes capacidades de transporte, são utilizados para mover objetos com características distintas em um ambiente dinâmico. O processo de alocação dos objetos ocorre criando uma correspondência entre as capacidades do agentes e as necessidades impostas pelos itens. Tais itens são adicionados de forma arbitrária no sistema, o que o torna um exemplo de alocação *online* de tarefas.

No trabalho de Ghosh et al. [2012], o transporte de objetos é resolvido através da solução do problema denominado *Multi-objective Particle Swarm Optimization*, que consiste em um algoritmo de otimização baseado no melhoramento iterativo de um conjunto de soluções guiado por uma heurística construída com base no problema a ser resolvido. O sistema apresentado busca minimizar as dimensões de tempo e energia utilizadas durante o processo, que são essencialmente dimensões conflitantes, uma vez que com maior gasto energético, ganha-se tempo, e o oposto também é válido.

Ao considerar a unidade básica do transporte por ações *PUSH*, Parra-González et al. [2008] apresenta um estudo sobre os pontos de contato entre os agentes e os objetos necessários para realização do transporte. Esta pesquisa foi aprimorada e estendida sendo apresentada por Parra-Gonzalez & Ramirez-Torres [2011], que descreve a utilização do algoritmo de planejamento *Wavefront* (LaValle [2006]) para discretizar o ambiente de trabalho e criar planos de movimentação os objetos, planos estes criados com base em três diferentes heurísticas (Figura 2.3), com diferentes objetivos: (i) busca diminuir a quantidade de mudanças de direção do objeto durante o transporte, (ii) valoriza certas orientações/rotações do objeto, pois em determinadas poses, a manipulação pelos agentes é facilitada e (iii) baseia-se em grafos, permitindo a avaliação de mais de um plano simultaneamente, onde os nós do grafo representam uma mudança na direção do transporte, e o melhor plano é aquele com a menor quantidade de nós. Em Parra-Gonzalez & Ramirez-Torres [2012], os autores demonstram a utilização de planos previamente criados para os objetos, sendo explorados para o planejamento de caminhos dos agentes sempre procurando diminuir a quantidade de manobras realizadas.

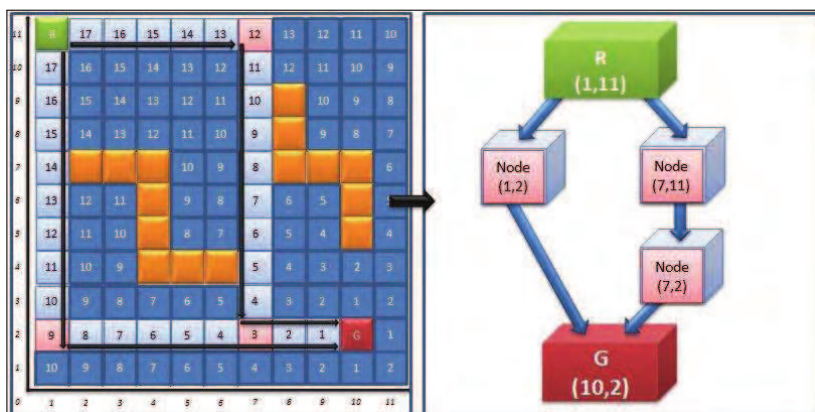


Figura 2.3: Planos de movimentação de um objeto a ser transportado apresentados no trabalho de Parra-Gonzalez & Ramirez-Torres [2011]. Os autores apresentam os resultados de três heurísticas distintas para o planejamento.

2.1.2 Manipulação Preênsil

Neste tipo de manuseio, os agentes fazem uso de manipuladores como garras ou braços robóticos para auxiliar na movimentação dos objetos. Desta maneira, o item sendo transportado é preso, segurado e/ou erguido pelo agente, sendo conhecido como método *GRASP*.

Antes de tornar a manipulação de objetos uma tarefa para agente móveis, alguns estudos foram realizados com manipuladores de base fixa. No trabalho de Montemayor & Wen [2005], é considerado um método para controle da movimentação e da quantidade de força exercida sobre um objeto durante o transporte. O sistema apresentado contempla uma coordenação descentralizada entre os agentes, feita mediante somente a leitura dos sensores presentes no manipulador. Este trabalho é uma extensão do trabalho de Wen & Kreuz-Delgado [1992], que fora inicialmente concebido de forma centralizada. Outro exemplo é o apresentado por Nagai et al. [2011], que demonstra como realizar o transporte de objetos utilizando um grua com 3 graus de liberdade realizando um planejamento de caminhos *online* baseado em um mapa do ambiente, porém faz uso de sensores ultrassônicos para detectar mudanças no cenário durante o transporte.

Uma estratégia para deslocar o objeto em uma determinada direção é exposta por Ahmadabadi & Nakano [2001], que cria o conceito de *constrain-move*, na qual alguns agentes seguram e impedem a movimentação do objeto em certas direções, enquanto outros agentes o empurram para o destino. Esta mesma estratégia é utilizada para rotacionar o objeto, porém todos os robôs cooperam movendo o item. Ambas ações são realizadas de forma descentralizada, onde os agentes percebem o estado do objeto

usando seus sensores.

Técnicas de planejamento baseadas em campos potenciais são largamente utilizadas para o transporte. Tanner et al. [2003] apresenta um método de controle centralizado que utiliza os chamados campos potenciais artificiais dipolares para criar um plano de movimentação para agentes não holonômicos possuindo manipuladores articulados, garantindo o desvio de obstáculos. Outro caso retratado por Wada & Torii [2013] utiliza robôs holonômicos e omnidirecionais, baseados em um sistema de rodas *caster* ativas, os quais são capazes de erguer o objeto e transportá-lo através de portas e corredores.

Similar aos casos no qual o sistema de transporte faz uso de ações *PUSH*, alguns trabalhos com *GRASP* utilizam métodos de coordenação *leader-follower*, como o exposto por Machado et al. [2013], em que dois agentes são capazes de desviar de obstáculos tanto estáticos quanto dinâmicos levando um objeto longo. O robô líder transporta o item, direcionando-o para sua pose final, enquanto o dito robô *helper* deve alinhar o corpo do objeto para que não colida. Sugar & Kumar [2002] faz algo similar, porém o objeto transportado pode ser flexível, e possui uma equipe com mais agentes. A manipulação do item ocorre com o envio da trajetória seguida pelo líder, e os demais devem segui-la, mantendo uma formação em relação ao objeto. É interessante salientar que no caso apresentado, o líder não é fixo, e pode ser realocado durante o trajeto.

O problema de transporte de múltiplos objetos também é tratado por Inoue et al. [2008, 2011], onde os robôs podem ser classificados em dois tipos: (i) única função (*single-function*) – capazes de transportar somente um objeto por vez, e (ii) multi função (*multi-function*) – aqueles que conseguem transportar mais de um objeto. Este trabalho estuda como ordenar e planejar uma sequência de transportes utilizando estes recursos.

Outra estratégia apresentada por alguns trabalhos para manipuladores móveis terrestres, é a exercitação do planejamento e controle mediante o uso de dois tipos de comando, um global e outro local. Essencialmente o planejador global deve ser capaz de criar um plano de execução do transporte considerando as limitações que o ambiente impõe, bem como as capacidades de manipulação dos agentes. O controlador local toma como base a sequência de ações a serem realizadas e monitora o desempenho dos agentes durante a tarefa, tomando as devidas decisões se algo não ocorrer como esperado, seja requisitando o replanejamento ou cancelando o transporte (Yamashita et al. [2003]; Hekmatfar et al. [2014]).

No trabalho de Adorno [2011] é apresentado um modelo de controle para dois braços robóticos, estendendo o trabalho apresentado em Adorno et al. [2010], agora

considerando o movimento do corpo inteiro do agente para realização da manipulação de itens com ambos manipuladores. Neste caso foi apresentado o controle cinemático de um robô com uma base móvel não holonômica, possuindo um torso e dois braços.

Um passo natural para o transporte cooperativo é a utilização de veículos aéreos para realização deste tipo de tarefa. Esta evolução garante diversas vantagens, como a extensão do ambiente de trabalho, que normalmente se limitava à movimentação em um plano (2D), e agora pode ser exercido em três dimensões. Porém também implica no aumento da complexidade do planejamento e execução da missão, onde os agentes não estão fixos, sofrendo mais facilmente de perturbações do ambiente, além de possuírem limitada capacidade de transporte, no que se relaciona ao peso e forma do objeto a ser manipulado.

Quando os agentes não são capazes de segurar o objeto através de uma garra, cabos podem ser utilizados para que o item possa ser erguido, porém esta tática acrescenta dificuldades para o controle da carga. Este problema é estudado por Michael et al. [2011] que discute um controlador para um conjunto de três agentes capaz de mensurar o equilíbrio estático do objeto, e assim descrever os comandos de movimentação dos robôs para atingir uma posição específica do objeto. Similarmente, Jiang & Kumar [2013] apresenta um algoritmo analítico que faz uso da técnica de eliminação dialítica (*dialytic elimination*), a fim de gerar um conjunto de soluções finitas para o sistema de cinemática inversa para posicionamento do grupo (agentes e objeto). É criado o chamado *tension workspace*, que limita a busca por soluções e faz o processo convergir.

Uma aplicação bastante interessante quando são utilizados agentes ágeis, como são os robôs aéreos, seja no formato de *quadrotors* ou helicópteros, é o transporte e entrega de pequenos objetos, como pacotes de remédios, comunicadores, sensores e afins, em casos de missões de busca e resgate. Casos como este podem ajudar vítimas de uma catástrofe a receber ajuda de forma mais rápida quando comparados a times usuais de humanos. O trabalho de Bernard et al. [2011] demonstra este cenário, e utiliza *drones* para entregar de forma precisa vários itens deste tipo, Maza et al. [2010] coordenam uma equipe de robôs para entregar em conjunto uma carga em um local previamente especificado.

Com a adição de um manipulador articulado ao robô aéreo, este conjunto pode ser tratado de duas maneiras, uma que considera as partes como separadas, e devem ser controladas independentemente, outra que julga o sistema como único, ou seja, o controle da pose do *end-effector* do manipulador deve ponderar não somente as articulações do braço, mas também a liberdade garantida por um agente que se move em um espaço 3D. Um modelo matemático pode ser derivado do sistema completo, o

que proporciona o controle direto da posição onde a ação de *GRASP* deve ser realizado. Este modelo pode ser aplicada em um ambiente dinâmico e considerar por exemplo erros mecânicos do aparato (Orsag et al. [2013]; Kim et al. [2013]). Seguindo um modelo similar, Arleo et al. [2013] discute um controlador hierárquico que planeja a cinemática inversa de todo o composto e o monitora durante o transporte.

No trabalho de Baizid et al. [2014] são demonstradas duas técnicas para o transporte de objetos usando *drones*, ambas baseadas no método *Null-Space-based Behavioral (NSB)*, que permite várias tarefas serem executadas de forma concorrente. No cenário apresentado, dois quadrotoros transportam uma barra, desviando dos obstáculos do ambiente. O primeiro método apresentado, maneja todo o conjunto para evitar colisões, já no segundo, somente o agente mais próximo ao obstáculo tem sua trajetória afetada.

Na maioria dos trabalhos, durante a etapa de captura, na qual o robô aéreo segura o objeto, ambos encontram-se em uma posição estática, ou seja, pousados sobre o solo, facilitando o manejo. Porém, existem casos onde isso não é possível, e o robô deve pegar o objeto sem pousar. Este é o problema descrito por Pounds et al. [2011], que cria um controlador *PID* capaz de manter o agente em *hovering*¹, enquanto um objeto é pego, e posteriormente o estabiliza para transporte (Figura 2.4). O caso demonstrado por Thomas et al. [2014] é uma evolução deste cenário, no qual o robô aéreo não somente permanece no ar, mas também em movimento. O trabalho apresenta um manipulador inspirado nas ações de captura e transporte realizada por falcões, que é capaz de planejar a trajetória do *end-effector* e manter o deslocamento do agente durante o processo de *GRASP*.

Outra aplicação interessante, uma vez que agentes aéreos são capazes de transportar objeto, é a criação de estruturas de forma autônoma. Uma maneira demonstrada em alguns trabalhos é a utilização de elementos pré-fabricados que possam ser conectados facilmente, como é o caso do chamado *Special Cubic Structures (SCS)*, peças como barras e juntas, possuem ímãs e são facilmente encaixadas (Lindsey et al. [2011, 2012]). Partindo do mesmo conceito, os trabalhos de dos Santos et al. [2013, 2014] descrevem um *framework* que cria um esquema adaptativo a partir de algoritmos de aprendizado de máquina (treinamento por reforço). Este *framework* criar planos de alto nível em um simulador e posteriormente acompanha a execução da construção em robôs reais.

Outro exemplo de construção usando agente autônomos é o nomeado *Aerial Robotic Construction (ARC)*, o qual define um estudo sobre a mudança nos paradigmas de design e fabricação de itens. Na pesquisa são expostos algumas vantagens deste

¹permanecer parado no ar



Figura 2.4: Transporte realizado por um quadrotor que utiliza uma garra para segurar o objeto demonstrado por Pounds et al. [2011].

sistema: (i) a construção das estruturas não necessita de armações de apoio para ser realizado, (ii) o modelo digital pode ser complexo, e os agentes são guiados explicitamente por este, e (iii) o sistema é escalável no sentido de não estar preso ao ambiente, além de poder escalar um grande equipe capaz de atuar simultaneamente durante o processo (Willmann et al. [2012]; Augugliaro et al. [2014]).

2.2 Alocação de Tarefas e Coordenação

Ao considerar que diversas atividades, como é o caso do transporte cooperativo, possuem um conjunto de subtarefas que devem ser realizadas para completar a tarefa principal, a etapa de alocação de tarefas compreende o estudo e organização deste conjunto, de modo a selecionar os melhores agentes para realização de cada subtarefa específica. Após esta etapa, surge a necessidade de coordenar os agentes para que realizem suas atividades de forma sincronizada e cooperativa.

Os trabalhos citados a seguir demonstram e estudam algumas técnicas empregadas principalmente em ambas estas etapas, discutindo métodos distintos, com suas próprias vantagens e desvantagens.

Neste sentido, Shiroma & Campos [2009] apresenta em seu trabalho o *framework* CoMutaR (*Coalition formation based on Multitasking Robots*), que descreve um método para alocação e coordenação de agentes multi-tarefas para realização de diversas tarefas. É baseado na suposição de que uma dada tarefa pode ser dividida em partes atômicas, denominadas ações; cada ação possui restrições e recursos necessários para ser ativada. Uma coleção de ações ativáveis resulta na realização da tarefa.

Outra proposta é apresentada por Carvalho et al. [2013], que pretende distribuir uma equipe de agentes para exploração de uma extensa região aplicando técnicas de coloração de grafos para selecionar qual robô deve ir para qual região. Kulatunga et al. [2006] apresenta uma técnica que busca planejar caminhos e alocar tarefas simultaneamente entre agente aéreos usando como base comportamentos inspirados em formigas.

É apresentado por Tiganas et al. [2013] uma técnica na qual agentes devem coletar e transportar amostras de uma região para outra, os mesmo são capazes de identificar e modificar seu comportamento mediante a necessidade do sistema, coordenando-se para coletar mais em uma determinada fonte ou entrar em modo de espera, por exemplo. Também apresentando uma estratégia para ambiente dinâmicos, Wang et al. [2009] descreve a utilização de grafos para a coordenação e mudança de classes de agentes dentro do simulador de futebol *RoboCup Soccer*.

Estas técnicas, além de suas variações e expansões são aplicadas em diferentes contextos, como no trabalho de Sujit & Saripalli [2013], que apresenta uma avaliação empírica de algumas estratégias de coordenação entre robôs heterogêneos com a missão de explorar uma determinada área a fim de coletar dados sobre a água para casos como o rastreamento de óleo, medida de gradientes de temperatura ou proliferação de algas nocivas, apresentando três táticas de coordenação: (i) periódica – que ocorre em intervalos pré-determinados de tempo, (ii) distância – que busca percorrer uma área com um plano de menor tamanho e (iii) probabilística – onde a coordenação ocorre em tempos adaptativos. Também buscando a cooperação entre agentes de diferentes tipos, Grocholsky et al. [2006] demonstra uma técnica de localização de alvos em uma região, usando um princípio descentralizado de controle implementando comportamentos diferentes para tipos de agentes.

Demonstrando como aplicar a técnica *AlphaBeta*, Goldsmith & Robinett III [1998] descreve seu uso na exploração de ambientes, onde não é possível uma supervisão direta e o sistema deve ser robusto o suficiente para funcionar sem intervenção. O grande diferencial desta técnica de coordenação está na diferença complementar de comportamentos entre as duas equipes, enquanto uma explora o ambiente, motivados por achar novas regiões não exploradas, a outra equipe segue os caminhos já gerados, economiza energia, gerando uma melhor leitura do ambiente. Neste mesmo sentido, Tanner [2007] também apresenta uma coordenação *AlphaBeta*, mas agora para robôs terrestres e aéreos, para a tarefa de monitoramento de uma área a fim de detectar e rastrear intrusos.

Especificamente em relação ao controle e coordenação de uma equipe de agentes, é possível listar uma série de modelos que compreendem esta etapa, com diferentes

aplicações e características.

Na técnica comportamental (*Behavior Based*), a organização dos agente no sistema é realizada através da associação de comportamentos aos mesmos, definidos como um conjunto de ações que são mapeadas diretamente dos sensores para os atuadores do robô. Estes comportamentos podem ser dispostos como uma máquina de estados finitos, promovendo uma mudança de atitude dos robôs mediante necessidade. No trabalho de Brooks [1986], é explanado a criação de um comportamento mais complexo baseado na disposição hierárquica de procedimentos mais simples. Deste modo, a ação executada será aquela resultante da junção de todas as saídas depois de passar pelos níveis do comportamento. Uma vantagem notável é a possibilidade da atuação de diversas condutas concorrentemente. Um aditivo a tal possibilidade seria a priorização de certos comportamentos, como por exemplo, o desvio de obstáculos. Em contraste a esta vantagem, a adição de mais níveis de controle aumenta a complexidade da tomada de decisão, que pode tornar o sistema intratável.

Outros formatos de coordenação são aqueles biologicamente inspirados (*Biologically Inspired*), na qual o controle dos agentes é baseada em traços biológicos, na tentativa de implementar mecanismos que foram desenvolvidos e testados na natureza no âmbito de sistema robóticos. Observando animais de diversos tipos, como insetos, é possível extrair características comportamentais para ações como exploração ou navegação (Kube & Bonabeau [2000]).

Uma das técnicas bastante utilizada, como visto em alguns trabalhos de transporte cooperativo, é a coordenação *AlphaBeta*, que pode ser aplicada como uma camada superior às demais técnicas de coordenação, pois, apesar de influenciar diretamente no comportamento do grupo, é tratado como um aditivo, e não como um substitutivo. Este método consiste na divisão da equipe de agentes em duas sub-equipes com objetivos complementares, seja pelo caso de um time preparar o ambiente para o outro ou servir como uma equipe sensorial, por exemplo. As duas equipes podem trabalhar simultaneamente, trocando informações pertinentes em tempo real, ou podem funcionar em modo comutado (*switched*), ou seja, enquanto um atua, o outro espera pelos resultados para só então entrar em ação (Goldsmith & Robinett III [1998]).

Existem também dois tipos de controle que geralmente são aplicados em uma camada acima dos demais citados anteriormente. O controle centralizado é baseado na coordenação e tomada de decisão considerando um líder, que deve processar e distribuir as tarefas dentre todos os membros da equipe. Os demais componentes do sistema recebem ordem e retornam relatórios de suas ações. Esta estratégia se destaca na velocidade em que todos os agentes da equipe tomam ciência de suas atribuições, sem necessidade de acordos ou negociações. A grande penalidade se encontra na parte

comandante, que, se houver falhas, todo o sistema pode sofrer uma falha (Farinelli et al. [2004]; Miyata et al. [2002]).

De forma contrária, no controle descentralizado, todos os integrantes do grupo participam das diversas etapas de que o sistema pode necessitar, como o sistema de coordenação e alocação de tarefas. A sincronização entre os agentes é feita mediante a troca de informações referentes às decisões individuais e posterior comunicação com os demais agentes. Neste caso, uma maior tolerância a falhas é agregada ao sistema, pois, mesmo que um agente pare de funcionar, os demais agentes podem continuar a execução da tarefa. Porém, neste tipo de controle, os agentes devem entrar em consenso sobre a responsabilidade das tarefas, o que pode acarretar um acréscimo no tempo total das atividades (Farinelli et al. [2004]).

2.3 Abordagem Proposta

Neste trabalho, será apresentada uma série de técnicas que propõem metodologias para o controle e coordenação de agentes para realização do transporte e manipulação de objetos utilizando uma equipe heterogênea de robôs.

Em contraste com a maioria dos trabalhos listados, a estratégia apresentada se baseia na trajetória criada para cada objeto a ser transportado, plano este que considera os recursos disponibilizados pelos robôs dentro do sistema. Comumente, o transporte de objetos é realizado baseado nos trajetos que os agentes devem realizar, uma vez que estes, utilizando táticas preênsil por exemplo, se deslocam e por consequência, transportam o objeto.

A etapa de mapeamento não é contemplada pela técnica, considerando que existe conhecimento dos estados e localidades de cada obstáculo, objetos e agentes dentro do ambiente de trabalho. O mapa criado como base para a tarefa compreende o método de segmentação celular em forma de matriz para dispor e descrever estas informações espacialmente.

Baseado na pré-existência do mapa, o foco desta pesquisa se encontra principalmente em como alocar tarefas e coordenar os agentes, de modo que dimensões como a distância percorrida, energia gasta e o tempo total da atividade sejam minimizados considerando as necessidades do sistema. Para tal, foi desenvolvida uma função de utilidade que considera tais dimensões, bem como a disparidade entre os tipos dos agentes, compreendendo um ambiente com robôs heterogêneos.

O controle e coordenação dos agentes será realizada de forma descentralizada, garantindo vantagens em tempo de planejamento, pois as atividades são separadas

dentre a equipe, sendo realizadas de forma simultânea.

2.4 Comparativo

A Tabela 2.1 apresenta um demonstrativo dos trabalhos relacionados à área de transporte cooperativo, salientando as áreas em que mais tiveram atuação dentro da pesquisa, além de elencar os campos que o presente trabalho possui contribuições.

Tabela 2.1: Tabela demonstrativa das pesquisas no campo do transporte de objetos elencando as principais áreas de atuação.

	Foco de Transporte		Tipo de Transporte			Coordenação	
	Objeto	Agentes	Preênsil	Não-Preênsil		Centralizada	Descentralizada
				Preênsil	Não-Preênsil		
Pereira et al. [2004]		x		x			x
Fink et al. [2008]		x		x			x
Behrens et al. [2010]		x	x		x		
Eoh et al. [2011]	x	x		x	x		
Inoue et al. [2011]	x			x			x
Nagai et al. [2011]	x			x	x		
da Costa [2012]	x	x		x			x
Parra-Gonzalez & Ramirez-Torres [2012]		x	x				x
Arleo et al. [2013]	x			x	x		
Jiang & Kumar [2013]	x		x		x		
Machado et al. [2013]	x			x	x		
Meriçli et al. [2013]		x	x		x		
Orsag et al. [2013]	x			x	x		
Kim et al. [2013]	x			x	x		
Wada & Torii [2013]	x		x				x
Hekmatfar et al. [2014]	x		x		x		
Magariyama et al. [2014]		x		x			x
Maghsoud et al. [2014]	x	x		x			x
Neumann et al. [2014]		x		x			x
Este Trabalho	x	x	x	x			x

Capítulo 3

Metodologia

Este capítulo descreve a técnica desenvolvida para a coordenação de uma equipe de agentes autônomos responsáveis pela manipulação e transporte de um conjunto de objetos. Para uma melhor apresentação da técnica, a mesma será fragmentada nas seguintes etapas: (i) formalização do problema e apresentação das notações utilizadas no decorrer do processo, (ii) descrição do algoritmo de planejamento de caminhos dos objetos, (iii) detalhamento sobre o planejamento e alocação de tarefas entre os agentes, (iv) discussão do método de coordenação e execução da tarefa de transporte. O diagrama apresentado na Figura 3.1 ilustra uma visão geral da metodologia apresentada neste trabalho, destacando onde as soluções para os problemas 1 e 2 serão utilizadas.

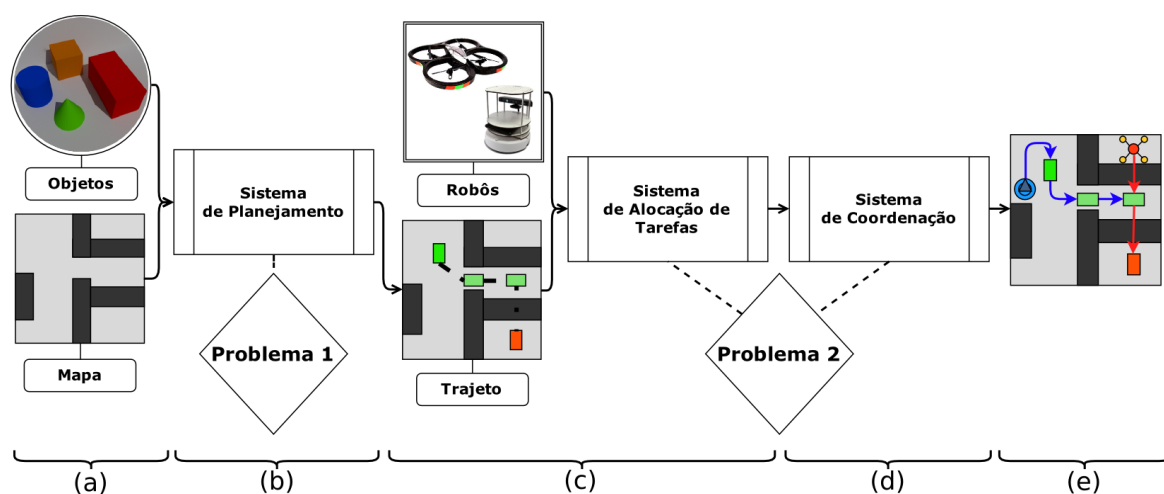


Figura 3.1: Diagrama da organização do sistema apresentado, ilustrando as etapas desenvolvidas durante o processo de transporte de objetos por um time de agentes. (a) descrição do ambiente de trabalho, (b) planejamento de caminhos dos objetos, (c) planejamento de caminhos dos agentes, (d) alocação de tarefas e (e) coordenação e execução.

3.1 Fundamentos e Formalização

Os problemas tratados por este trabalho compreendem todas as etapas necessárias para a manipulação de objetos em um determinado ambiente mediante o uso de uma equipe de agentes robóticos. Esta problemática pode ser formalmente definida como:

Dado um ambiente conhecido em um espaço euclidiano \mathbb{R}^3 , é definido por $\mathcal{W} \subset \mathbb{R}^3$ um poliedro convexo como o ambiente de trabalho. Este espaço é discretizado em uma matriz, possuindo células c de tamanho unitário.

Inseridos em \mathcal{W} , são descritos os conjuntos: (i) conjunto $\mathcal{O} = \{o_1, o_2, \dots, o_y\}$ contendo y objetos que devem ser transportados para seus respectivos destinos, (ii) conjunto $\mathcal{B} = \{b_1, b_2, \dots, b_x\}$, no qual são representados demais objetos no ambiente de trabalho considerados como obstáculos durante a movimentação tanto dos agentes quanto dos objetos, e (iii) conjunto $\mathcal{R} = \{r_1, r_2, \dots, r_k\}$ possuindo k robôs heterogêneos utilizados no processo de manipulação dos objetos em \mathcal{O} .

A solução aqui descrita possui algumas limitações, casos que não foram tratados ou fogem do escopo da técnica, desta maneira, as seguintes suposições foram adotadas durante o desenvolvimento:

1. Os agentes tem ciência da sua própria localização, dos objetos e dos obstáculos no ambiente;
2. Não existirão dentro do ambiente de trabalho \mathcal{W} demais corpos capazes de se mover sozinhos além dos agentes;
3. Existe um canal de comunicação entre os agentes, e o mesmo não possui falhas;
4. Todos os objetos a serem transportados são iguais e de dimensões conhecidas.

Estas suposições foram necessárias para tornar o problema tratável no sentido de focar os esforços de pesquisa e implementação especificamente na resolução da questão principal de coordenação dos agentes.

O conjunto \mathcal{R} dos robôs utilizados será a união de dois conjuntos menores formados por tipos diferentes de agentes. Estes tipos são definidos como: (i) robô terrestre (*ground*) – agente capaz de se movimentar utilizando um sistema diferencial de locomoção, e não possuindo nenhum artefato para manipulação do objeto, que utiliza seu próprio corpo para aplicar a técnica de transporte não-preênsil *PUSH*, (ii) robô

aéreo (*aerial*) – agente capaz de voar utilizando qualquer configuração que o permita pairar sobre um determinado ponto no espaço, como os modelos em formato de helicóptero, tricóptero ou quadricóptero, para transporte o mesmo utilizará a estratégia préensil *GRASP* para segurar o objeto e transportá-lo. Ambos os tipos são definidos no conjunto $\mathcal{T} = \{\text{ground, aerial}\}$.

O ambiente de trabalho \mathcal{W} , que em essência é um ambiente contínuo, será discretizado espacialmente através da criação de uma matriz de 3 dimensões. As células desta matriz possuem tamanho unitário, e serão tratadas como a unidade básica de deslocamento e localização, ou seja, qualquer item no ambiente (robôs, objetos e obstáculos) ocupa no máximo 1 célula por vez, e uma determinada célula só pode ser ocupada por 1 item em um determinado tempo. Como será descrito a seguir, todos os planos de movimentação dentro do ambiente compreenderão o deslocamento entre uma série de células, e o tamanho de um determinado plano será a sumarização de quantas células são percorridas durante sua execução. Cada célula c é representada por uma tupla com valores (x, y, z) .

Considerando que o objetivo do trabalho apresentado é estudar e criar planos de movimentação que, uma vez executados, culminem no transporte de todos os objetos do ambiente, para uma melhor apresentação desta solução é definida uma nomenclatura relacionada ao sistema de planejamento:

\mathcal{M} : Conjunto de ações $\mathcal{M} = \{\text{left, right, forward, backward, up, down}\}$ que podem ser desempenhadas pelos agentes. Cada ação é representada pelo símbolo a_i , e algumas destas só podem ser executados por um tipo específico de agente, por exemplo, somente o agente de tipo aéreo pode executar as ações $\langle \text{up, down} \rangle$.

S_i : Estrutura que descreve informações sobre os passos de um plano \mathcal{P} . Contém uma referência para uma célula $c \in \mathcal{W}$, além de outros de dados que são utilizados durante a etapa de planejamento, como o tipo de agente que executará a ação a_i e um valor de utilidade (descrito na seção 3.2). Esta estrutura pode possuir duas notações especiais: (i) S_o – quando é o primeiro elemento do plano, e (ii) S_d – quando é o último elemento.

\mathcal{P} : Plano $\mathcal{P} = \{S_o, \dots, S_d\}$ que descreve uma sequência de estruturas S_i . Somente através desta sequência é possível registrar quais ações são necessárias para realização da movimentação. Por exemplo, para realizar a movimentação da célula $c_1 (5, 5, 1)$ para $c_2 (5, 6, 1)$, deve ser executada a ação $a_3 \langle \text{forward} \rangle$.

\mathcal{P}_ε : Conjunto de planos \mathcal{P} , nomeado plano de execução, que descreve todos as trajetórias a serem executadas pelos agentes para realizar o transporte de todos os objetos em \mathcal{O} .

Objetivamente, durante as etapas de planejamento, é desejável encontrar os melhores planos, isto é, aqueles com a melhor combinação de ações, capazes de movimentar um determinado objeto de sua posição atual até um ponto de destino. Para tal, foi desenvolvido uma função de utilidade (descrita na seção 3.2) utilizada para quantificar um determinado plano, utilizada para comparar distintos planos quanto à sua qualidade. Desta maneira, o conjunto de planos \mathcal{P}_ε , é aquele contendo os planos necessários para o transporte de todos os objetos minimizando o custo total das ações.

3.1.1 Notação gráfica

Como o sistema desenvolvido compreende um ambiente de trabalho no qual diversos componentes estão disposto espacialmente no mesmo, para um melhor entendimento do processo desenvolvido, alguns diagramas foram desenvolvidos, demonstrando os casos estudados nesta metodologia, possuindo uma série de elementos gráficos. A seguir é listado todos os elementos utilizados:








Figura	Descrição
	Figura representativa de um robô terrestre;
	Figura representativa de um robô aéreo;
	Figura representativa da posição inicial de um objeto;
	Figura representativa da posição final de um objeto;
	Figura representativa de um obstáculo;
	Figura representativa de uma célula no ambiente de trabalho;
	Figura representativa de um plano;

Tabela 3.1: Descrição dos elementos utilizados nos diagramas de exemplificação.

3.2 Função de Utilidade

Durante o processo de planejamento de caminhos do objeto, de modo a considerar a heterogeneidade dos agentes, são analisadas características relacionadas ao tipo de robô que pode executar o plano, de modo a encontrar aquele que melhor satisfaça as necessidades do sistema.

Para tal, são avaliados dois parâmetros para cada tipo agente, estes descrevem a utilidade da execução de um determinada ação a_i dentro de um plano \mathcal{P} , de modo a quantificar esta ação em um aspecto passível de ser comparado. As características a seguir são utilizadas na avaliação:

- **Tempo de deslocamento:** Tempo necessário para movimentação entre células c . Este aspecto está relacionado à velocidade média que um determinado robô consegue realizar o deslocamento. Descrito pela função Υ ;
- **Custo de deslocamento:** Custo para movimentação entre células c . Para cada tipo de robô utilizado no transporte, é calculado o gasto médio de energia que o mesmo tem durante o deslocamento. Descrito pela função Ψ .

Formalmente, a Equação 3.1 descreve a função de utilidade Θ , que avalia um determinado tipo de agente $t_i \in \mathcal{T} = \{ground, aerial\}$, dando-lhe uma pontuação baseado em seus atributos. O valor gerado desta análise é utilizado para guiar a criação do plano de movimentação do objeto (descrito na Seção 3.3).

$$\Theta(t_i, a_i) = \alpha \cdot \Upsilon(t_i) + (1 - \alpha) \cdot \Psi(t_i). \quad (3.1)$$

As constantes definidas por $\langle \alpha \rangle = \{\alpha \in \mathbb{R} | 0 \leq \alpha \leq 1\}$ são utilizadas como fatores de ponderação entre as características avaliadas de forma a refletir a priorização de certas atividades, como executar no menor tempo possível ou com a menor energia gasta, por exemplo. Este modelo é flexível pois pode ser reconfigurado para atender a diversas necessidades, além de adicionar ao planejamento particularidades de cada agente. O valor resultante desta função é utilizado como parte da heurística utilizada na busca pelo melhor plano de movimentação.

Para determinar o valor de utilidade de um determinado plano \mathcal{P} , os valores de utilidade de cada S_i que o constitui são somados como demonstrado na Equação 3.2:

$$\Theta_p(\mathcal{P}) = \sum_{S_i \in \mathcal{P}} \Theta(k(S_i)). \quad (3.2)$$

Considerando que a função de utilidade está baseada no tipo t_i de um agente que realizará uma determinada ação a_i , é definida a equação 3.3, que extrai tais informações diretamente de uma estrutura S_i , como definida a seguir:

$$k : S_i \mapsto t_i, a_i. \quad (3.3)$$

A função de utilidade Θ juntamente com a equação 3.2, é utilizada para determinar, dentre os diversos planos de movimentação que podem ser criados, qual o melhor plano, ou seja, aquele que possuir o menor custo total para realização das ações. Neste sentido, a metodologia apresentada visa encontrar o conjunto $\mathcal{P}_{\mathcal{E}}$, aquele que minimize o custo total de todo o sistema. A equação 3.4 ilustra este objetivo.

$$\arg \min_{\mathcal{P}_{\mathcal{E}}} \sum_{\mathcal{P} \in \mathcal{P}_{\mathcal{E}}} \Theta_p(\mathcal{P}). \quad (3.4)$$

3.3 Planejamento de Caminhos - Objeto

A primeira etapa da técnica apresentada é a criação de planos de movimentação para cada objeto $o_i \in \mathcal{O}$, este processo é dividido em dois estágios: (i) planejamento de caminhos e (ii) segmentação do plano. Para realização do planejamento, assume-se que o objeto a ser transportado é capaz de movimentar-se dentro da área de trabalho \mathcal{W} . Os movimentos disponíveis para o objeto nesta fase estão listados no conjunto \mathcal{M} .

3.3.1 Planejamento

O algoritmo implementado é uma variação do Algoritmo A* (Hart et al. [1968]), como é descrito no Algoritmo 3.1. Inicialmente é criado um grafo direcionado vazio denominado $\mathcal{P}_{\mathcal{G}}$, que guardará referências para todas as estruturas S_i que foram exploradas enquanto a busca pelo plano é executada. Além disso, seguindo a mesma estratégia do algoritmo, é criada uma lista nomeada \mathcal{F} , que guarda as referências para todos os nós folhas do grafo $\mathcal{P}_{\mathcal{G}}$, ou seja, todas as estruturas S_i que ainda não foram exploradas e se encontram na borda da área de busca.

Uma etapa necessária durante a execução deste algoritmo é a seleção do melhor vértice a ser expandido (linha 8 do Algoritmo 3.1), para tal, é considerada como heurística de seleção dois valores: (i) a distância remanescente estimada entre a célula c referenciada por S_i e a célula de destino do plano – utiliza-se a distância de Manhattan; e (ii) o valor de utilidade calculado para S_i , considerando o tipo de agente que pode realizar a ação. Estes valores são somados, resultando no valor de utilidade da estrutura

Algorithm 3.1 AStar($S_o, S_d, \mathcal{M}, \mathcal{W}$)

```

1:  $\mathcal{P}_G \leftarrow \emptyset$ 
2:  $\mathcal{F} \leftarrow \{S_o\}$ 
3:
4:  $S_o[utility] \leftarrow \Theta(S_o) + \text{distance}(S_o, S_d)$ 
5:
6: while  $\mathcal{F} \neq \emptyset$  do
7:   // vertex selection step
8:    $S_j \leftarrow \min(\mathcal{F})$ 
9:
10:  if  $S_j = S_d$  then
11:    return reconstruct_path( $S_j$ )
12:  end if
13:
14:   $\mathcal{F} \leftarrow \mathcal{F} \setminus \{S_j\}$ 
15:   $\mathcal{P}_G \leftarrow \mathcal{P}_G \cup \{S_j\}$ 
16:
17:  // expansion step
18:  expansion_algorithm( $S_j, S_d, \mathcal{M}, \mathcal{F}, \mathcal{P}_G, \mathcal{W}$ )
19: end while
20: return false

```

S_i para cada tipo de agente. Para garantir que o plano siga por um caminho sempre tentando minimizar o valor de custo total, aquela que possuir a menor pontuação, independente de tipo, será escolhida para expansão. Esta operação ocorre considerando os elementos do conjunto \mathcal{F} .

O Algoritmo 3.2 descreve o processo de expansão de um determinado vértice do grafo \mathcal{P}_G (linha 19 do Algoritmo 3.1). As demais etapas do algoritmo A*, como a construção do caminho desde a origem até o destino do plano, são similares à implementação original do mesmo. O passo de expansão garante que todas as ações possíveis de serem realizadas a partir de uma determinada posição em \mathcal{W} possam ser posteriormente avaliadas pela função de seleção.

Durante a etapa de expansão, um vértice do grafo de busca é modificado de forma a verificar quais ações podem ser realizadas a partir do mesmo. Baseado na referência para a célula c da estrutura S , conseguida através da função *get_position()*, o algoritmo cria novas referências de células aplicando todas as ações $a_i \in \mathcal{M}$ (função *apply_action()*). Uma vez com esta posição, a função *is_executable* considera se a ação é possível de ser executada, no caso, se a posição S_r , na qual o agente deve estar para realizar a ação, não colide com outros obstáculos, robôs ou objetos. Possuindo estes dados, é criado um novo vértice S' , que guarda informações sobre a ação utilizada para

Algorithm 3.2 ExpansionAlgorithm($S, S_d, \mathcal{M}, \mathcal{F}, \mathcal{P}_G, \mathcal{W}$)

```

1:  $c \leftarrow \text{get\_position}(S)$ 
2: for all  $a_i \in \mathcal{M}$  do
3:
4:    $c' \leftarrow \text{apply\_action}(a_i, c)$ 
5:    $S_r \leftarrow \text{is\_executable}(a_i, c')$ 
6:    $S' \leftarrow \text{create\_struct}(S, c', a_i)$ 
7:    $S'[\text{utility}] \leftarrow \Theta(S') + \text{distance}(S', S_d)$ 
8:
9:   if  $S' \in \mathcal{P}_G$  or  $\text{is\_colliding}(\mathcal{W}, S')$  or not  $S_r$  then
10:      $\mathcal{P}_G \leftarrow \mathcal{P}_G \cup \{S'\}$ 
11:     continue
12:   end if
13:
14:    $\mathcal{F} \leftarrow \mathcal{F} \cup \{S'\}$ 
15: end for

```

chegar ao mesmo, além do parentesco com a estrutura S (função $\text{create_struct}()$).

Um processo importante que é realizado durante a criação do novo vértice é a execução do cálculo de utilidade do mesmo, que ocorre similar ao representado na linha 4 do Algoritmo 3.1. Além deste cálculo, é aplicada uma penalização no custo sempre que a ação que resultou a criação do vértice for diferente da ação do vértice pai. Esta penalização proporciona a criação de planos com a menor quantidade de mudanças de direção possível, bastante importante, principalmente no transporte terrestre, pois o agente não necessita realizar mais deslocamentos para empurrar o objeto para outras direções.

Dispondo destes resultados, o algoritmo testa se a nova posição assinalada já foi estudada anteriormente (se está presente em \mathcal{P}_G), se a mesma colide com algum obstáculo do ambiente (função $\text{is_colliding}()$) e se não é possível de ser executada. Se algum destes testes for positivo, esta nova estrutura S' é considerada como inválida e adicionada diretamente ao grafo de vértices já estudados, caso contrário, o mesmo será adicionado ao conjunto \mathcal{F} , para uma posterior expansão.

O algoritmo de exploração termina em dois casos, (i) quando o conjunto \mathcal{F} está vazio, indicando que não existe um caminho até o estado de destino S_d , ou (ii) quando o estado selecionado para expansão é o estado final, neste caso, é reconstruído o caminho desde o destino, utilizando as informações de parentes de vértices, até a posição de origem. O resultado do algoritmo de exploração é um plano \mathcal{P} , que pode ser visto como um sub-grafo de \mathcal{P}_G . A Figura 3.2 mostra um planejamento de exemplo para um objeto dentro de um certo ambiente. Uma vez que o plano seja criado, o passo de

segmentação é executado.

Como o algoritmo implementado é uma variação do Algoritmo A*, o mesmo possui uma complexidade similar, relacionado a dois fatores: (i) a organização do conjunto \mathcal{F} para escolha do vértice a ser expandido, que no pior caso possui complexidade $\mathcal{O}(n^k)$, na qual n é o número de vértices explorados durante a busca pelo caminho, e k é a quantidade de possíveis ações que podem ser tomadas durante o processo, no caso $k = |\mathcal{M}| = 6$. (ii) o tempo necessário para computar o valor da heurística de utilidade, que neste caso é $\mathcal{O}(1)$.

3.3.2 Segmentação

A segmentação tem sua importância no transporte terrestre pois, considerando que os robôs terrestres só possuem a capacidade de empurrar os objetos, sempre que há uma mudança na direção do movimento, o robô deve realizar a ação de *PUSH* por outro lado objeto, e neste momento, existe a possibilidade de que um agente diferente seja mais apto, com uma melhor utilidade para realizar a ação, que o robô que está atualmente realizando a tarefa. Em relação à mudança de tipo de movimento, passando de *PUSH* para *GRASP*, a segmentação do plano consegue destacar esta diferença, e facilitar o tratamento posteriormente.

O processo de segmentação executa a divisão do plano \mathcal{P} criado pelo algoritmo de busca, gerando o conjunto \mathcal{S} como resultado da operação. Este conjunto será utilizado durante a etapa de planejamento dos robôs e posteriormente para alocação de tarefas entre os agentes.

Para realizar a segmentação do plano, são executados dois algoritmos: (i) algoritmo de criação dos pontos de segmentação (Algoritmo 3.3), no qual é criado um conjunto auxiliar, nomeado \mathcal{S}_p , contendo uma lista de tuplas, cada uma destas assinalando uma mudança de direção dentro do plano, e (ii) algoritmo de criação dos segmentos (Algoritmo 3.4), que utiliza a lista de referências em \mathcal{S}_p para criar uma lista de segmentos do plano.

O processo de criação dos pontos de segmentação (Algoritmo 3.3), percorre par a par os elementos de um determinado plano \mathcal{P} , primeiramente observando se ambas as estruturas representam posições aéreas (função *is_aerial()*), se o teste for positivo, então não há necessidade de criar um ponto de segmentação, pois o transporte aéreo, diferente do terrestre, não pode ser interrompido durante sua execução, independente de quantas vezes troque de direção; após o teste, é requisitado das estruturas as ações utilizadas para a sua criação (função *get_action()*), pois sempre que estas ações forem diferentes, uma mudança de direção ocorreu, ocasionando na criação de um novo

Algorithm 3.3 SegmentPointSetCreation($\mathcal{P}, \mathcal{S}_p$)

```

1: for all  $S_i, S_{i+1} \in \mathcal{P}$  do
2:   if  $\text{is\_aerial}(S_i)$  and  $\text{is\_aerial}(S_{i+1})$  then
3:     continue
4:   end if
5:    $a_i \leftarrow \text{get\_action}(S_i)$ 
6:    $a_{i+1} \leftarrow \text{get\_action}(S_{i+1})$ 
7:   if  $a_i <> a_{i+1}$  then
8:      $sp_i \leftarrow \text{create\_segment\_point}(S_i, S_{i+1})$ 
9:      $\mathcal{S}_p \leftarrow \mathcal{S}_p \cup \{sp_i\}$ 
10:  end if
11: end for
12: return  $\mathcal{S}_p$ 

```

Algorithm 3.4 SegmentSetCreation($\mathcal{S}_p, \mathcal{S}, S_o, S_d$)

```

1: for all  $sp_i \in \mathcal{S}_p$  do
2:    $S_0 \leftarrow sp_i[0]$ 
3:    $S_1 \leftarrow sp_i[1]$ 
4:   if  $\mathcal{S} = \emptyset$  then
5:      $s_i \leftarrow \text{create\_segment}(S_o, S_0)$ 
6:   else
7:      $s_i \leftarrow \text{create\_segment}(S_{temp}, S_0)$ 
8:   end if
9:    $S_{temp} \leftarrow S_1$ 
10:   $\mathcal{S} \leftarrow \mathcal{S} \cup \{s_i\}$ 
11: end for
12:  $s_i \leftarrow \text{create\_segment}(S_{temp}, S_d)$ 
13:  $\mathcal{S} \leftarrow \mathcal{S} \cup \{s_i\}$ 
14: return  $\mathcal{S}$ 

```

ponto de segmentação (função $\text{create_segment_point}()$), posteriormente adicionado ao conjunto \mathcal{S}_p . Cada ponto de segmentação é uma tupla com dois valores, referenciando ambas estruturas (S_i, S_{i+1}).

Em continuação, é executado o Algoritmo 3.4, que toma como base o conjunto auxiliar \mathcal{S}_p para criar os segmentos. O algoritmo itera sobre todos os pontos de segmentação, criando segmentos do plano \mathcal{P} (função $\text{create_segment}()$). Após este processo, o conjunto $\mathcal{S} = \{s_1, s_2, \dots, s_e\}$ é criado, possuindo e segmentos, cada um destes é um sub-plano do plano original \mathcal{P} e denominam trajetos do plano que: (i) se for terrestre, não possui nenhuma mudança de direção, e (ii) se for aéreo, se estende desde o passo no qual é assinalada a decolagem, até o passo onde um pouso é descrito. Similar ao plano \mathcal{P} , cada segmento s_i no conjunto \mathcal{S} também possui referências para os estados

S_o e S_d , com a mesma definição. Na Figura 3.2 podem ser observados os segmentos que foram criados para o plano de um objeto de teste.

A criação dos pontos de segmentação demonstrada no Algoritmo 3.3, como é baseado exclusivamente na quantidade de passo do plano, possuindo uma complexidade de $\mathcal{O}(n)$, onde n é o tamanho do plano. Já o Algoritmo 3.4 possui similar complexidade de $\mathcal{O}(m)$, porém m é a quantidade de elementos do conjunto \mathcal{S}_p .

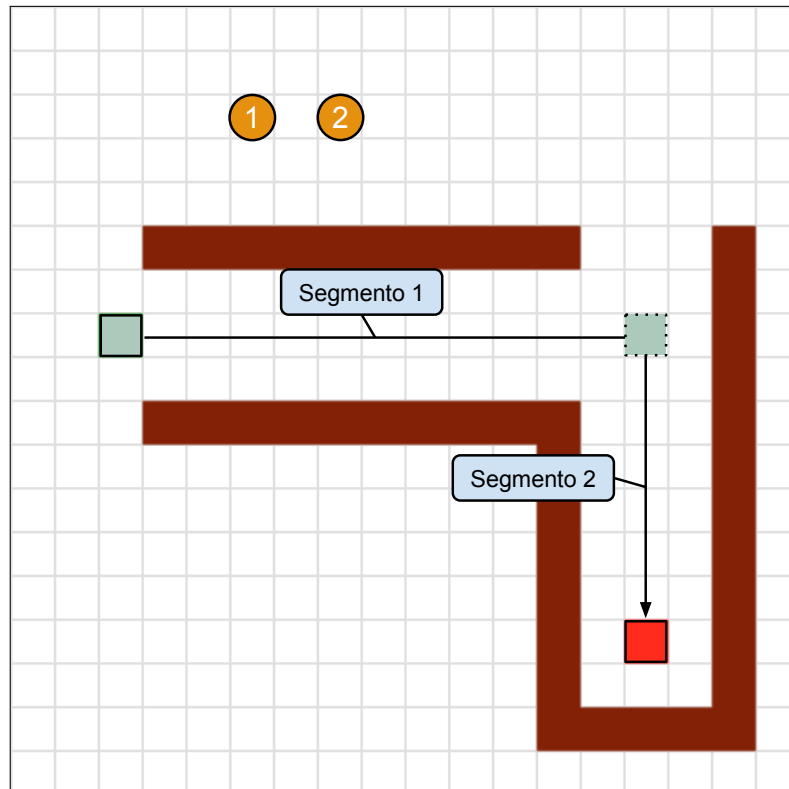


Figura 3.2: Etapa de Planejamento de caminhos e Segmentação do plano de um objeto a ser transportado em um ambiente de teste. O exemplo ilustra o plano de um objeto, a criação de dois segmentos, pois há somente uma troca de direção em seu plano. Como não existem agente aéreos no ambiente, somente um trajeto terrestre foi planejado.

3.4 Planejamento de Caminhos - Agentes

Esta etapa visa a criação de planos de movimentação para os agentes, de modo a provê-los do melhor trajeto para se locomover dentro do ambiente. Planos estes utilizados nos casos onde o agente deve se aproximar do objeto e posteriormente, indicar a rota de transporte. Dentro do sistema, estes planos serão utilizados tanto na fase de alocação de tarefas quanto na execução do transporte.

O planejamento da movimentação de cada robô $r_i \in \mathcal{R}$ ocorre de maneira iterativa, de modo que os planos são gerados no decorrer da fase de alocação de tarefas (Seção 3.5). Cada plano possui um valor de utilidade mediante a função Θ_p descrita na Equação 3.2 que quantifica sua qualidade em relação aos demais planos. Estes valores serão utilizados para escolher o melhor agente para realizar um determinado segmento do transporte.

Para criar um plano para um determinado agente, é utilizado um algoritmo similar ao utilizado para criação dos planos para os objetos (algoritmos 3.1 e 3.2), com as seguintes modificações: (i) o teste de colisão usando a função $is_colliding()$ usa o corpo do robô como referência, e (ii) a função $is_executable()$ não é executada, considerando que este teste só se aplica ao objeto. Este algoritmo é adotado na criação de dois tipos de planos, listados no conjunto $\mathcal{T}_M = \{\text{preparação, transporte}\}$ e descritos a seguir:

- **preparação:** usado pelos agentes para se aproximar do objeto a ser transportado. O plano inicia-se na posição atual do robô até a posição indicada por S_r , que se encontra na estrutura S_o do primeiro segmento ainda não realizado de um objeto alvo.
- **transporte:** executado quando o robô está realmente realizando o transporte. Parte da posição inicial do segmento (S_o) até o estado final (S_d).

A Figura 3.3 ilustra todos os possíveis planos que os agentes podem executar a fim de realizar a tarefa de transporte, e seus respectivos tipos. Os planos partem da posição do agente I enumerados por agente (note que apesar do diagrama demarcar uma célula adjacente, a célula utilizada para o planejamento é a qual o agente se encontra no momento), indo até os estados S_r descritos nas estruturas S_o ou S_d de cada segmentos. Para uma melhor diferenciação, os estados S_r foram enumerados.

3.5 Alocação de Tarefas

Na etapa de alocação de tarefas, o principal objetivo é determinar uma sequência de movimentos, que uma vez realizados, culminem no transporte de todos os objetos para suas respectivas posições finais, simultaneamente minimizando o custo total dos planos executados pelos agentes.

A ordem na qual os objetos serão transportados tem direta influência na distância percorridas pelos agentes, desta forma, a equipe de agentes deve analisar o estado do ambiente de trabalho (suas próprias posições e as posições dos objetos) e escolher a

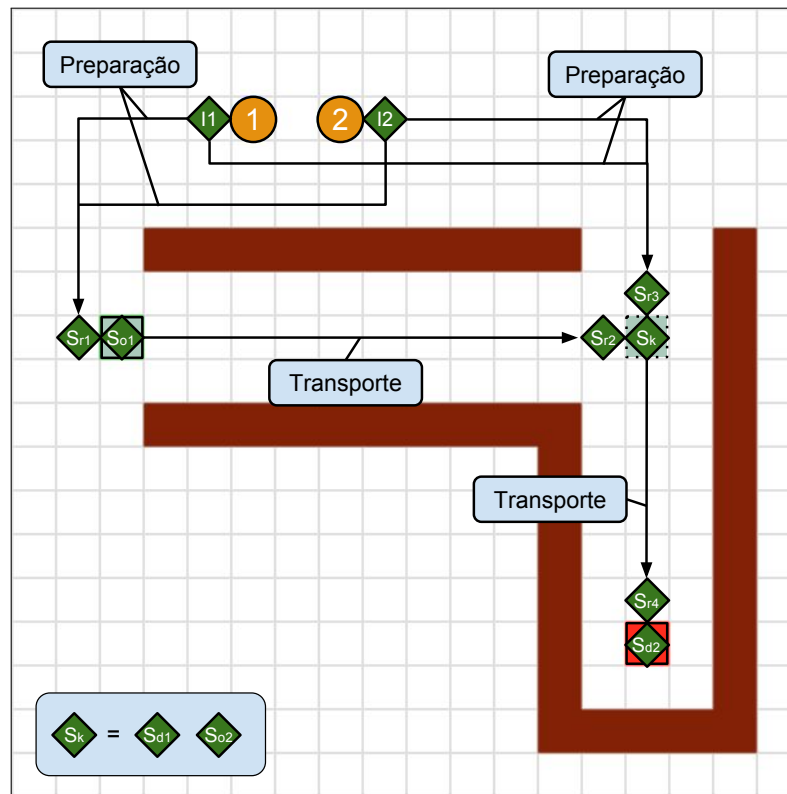


Figura 3.3: Exemplo de planos gerados para os agentes que podem ser utilizados para realizar o transporte do objeto. Cada agente cria seus próprios planos baseado na quantidade e configuração de cada segmento do trajeto do objeto a ser transportado. A quantidade de planos criados cresce à medida que existem mais agentes, objetos, ou mesmo quando existem mais segmentos em um determinado plano.

melhor sequência para execução, por exemplo, realizar o transporte visando os objetos mais próximos de cada agente ao invés dos mais distantes.

transportar os objetos mais próximos seja melhor que outros mais distantes.

O foco da alocação de tarefas no âmbito do transporte cooperativo de objetos é o de encontrar a melhor seleção de tarefas, ao considerar os agentes e objetos dispostos no ambiente. Desta maneira, é apresentado um algoritmo distribuído que realiza a alocação dos segmentos entre os agentes.

A fase de planejamento de caminhos dos objetos (Seção 3.3) ocorre de forma centralizada, de modo que não mudará no decorrer da execução das tarefas. Um membro da equipe é elegido de forma arbitrária e realiza esta etapa, disponibilizando para os demais agentes os planos resultantes através de uma comunicação em *broadcast*, de forma a sincronizar todos os agentes para o início da fase de alocação. Munidos destas informações, toda a equipe de agentes participa da alocação de tarefas, executando o Algoritmo 3.5.

Algorithm 3.5 TaskAllocationProcess(\mathcal{O})

```

1: loop
2:    $\mathcal{W}_{\mathcal{O}} \leftarrow \text{build\_work\_list}(\mathcal{O})$ 
3:   if  $\mathcal{W}_{\mathcal{O}} = \emptyset$  then
4:     break
5:   end if
6:
7:   utility_memory  $\leftarrow \{\}$ 
8:   for all  $o_i \in \mathcal{W}_{\mathcal{O}}$  do
9:      $s_i \leftarrow \text{get\_first\_segment}(o_i)$ 
10:     $\mathcal{P}_m \leftarrow \text{create\_move\_plan}(s_i)$ 
11:     $\mathcal{P}_t \leftarrow \text{create\_transport\_plan}(s_i)$ 
12:    total_utility  $\leftarrow \Theta_p(\mathcal{P}_m) + \Theta_p(\mathcal{P}_t)$ 
13:    utility_memory  $\leftarrow \text{utility\_memory} \cup \{o_i: \text{total\_utility}\}$ 
14:   end for
15:
16:   share_memory(utility_memory)
17:   wait_until_receive_all_memories()
18:
19:   utility_memory  $\leftarrow \text{utility\_memory} \cup \text{get\_team\_memory}()$ 
20:   allocation  $\leftarrow \text{execute\_hungarian\_method}(\text{utility\_memory})$ 
21:   update_own_tasks(allocation)
22:
23:   share_ready()
24:   wait_until_reveice_all_ready()
25: end loop

```

O algoritmo demonstrado é executado por cada agente, e se torna sincronizado dentre a equipe mediante o uso das funções de espera (*wait_until_receive_all_memories()* e *wait_until_reveice_all_ready()*). Internamente, para decidir qual agente é o mais apto para qual tarefa, é executado o Algoritmo de Otimização Combinatória Húngaro (Munkres [1957]).

Inicialmente o algoritmo cria um subconjunto de objetos, com todos aqueles que ainda não foram totalmente transportados (função *build_work_list()*); quando esta lista estiver vazia, não existem mais objetos fora da sua posição final. Considerando que cada agente deve planejar seus próprios planos para cada objeto, a variável *utility_memory* é criada como um dicionário, que mapeia um determinado objeto para uma determinada utilidade associada aos planos do agente. Nas linhas 10 e 11 são criados os planos de preparação e transporte, que posteriormente são utilizados para o cálculo da utilidade total (*total_utility*), e por fim, este valor é associado ao objeto na linha 13.

Uma vez que o agente possua os valores de utilidade para cada objeto da lista de trabalho (\mathcal{W}_o), o mesmo realiza o envio desses dados para todos os demais agentes e aguarda até receber as informações do restante da equipe.

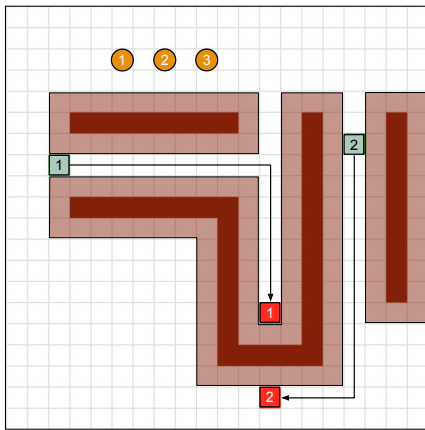
Assim que toda a equipe possuir os dados compartilhados, as utilidades dos outros agentes é associada às utilidades internas, criando assim a chamada tabela de utilidade, que descreve as utilidades associadas aos objetos por cada agente (linha 19). A próxima etapa é a execução do algoritmo de alocação Húngaro, que recebe esta tabela e retorna uma lista com as alocações.

O Algoritmo Húngaro utiliza como base para alocação a chamada tabela de utilidade, que descreve a utilidade associada entre agentes e tarefas (segmentos). Tal tabela é representada pela variável *utility_memory*, que guarda a utilidade do agente executar aquela determinada tarefa. Uma vez com a tabela totalmente preenchida, o algoritmo busca pela combinação par a par entre agentes e tarefas, na qual a soma total das utilidades seja a mínima possível (função *execute_hungarian_method()*). Desta maneira, sempre os agentes mais aptos serão alocados para os objetos certos. Um fato importante a ser mencionado é que este algoritmo sempre aloca tarefas para todos os agentes, ou seja, quando existem mais segmentos (tarefas) que agentes, todos receberão alguma tarefa para realizar.

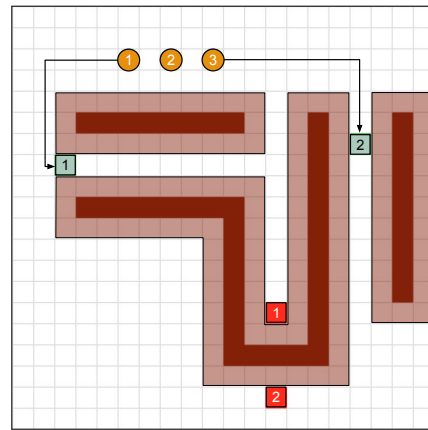
Utilizando a função *update_own_tasks()* o agente guarda internamente quais tarefas foram alocadas para si na variável τ , observando os pares de alocação na variável *allocation*. Por fim, o agente avisa aos demais que está pronto para as demais etapas de alocação e espera que os demais informem o mesmo.

O Algoritmo 3.5 de alocação de tarefas possui uma série de sub-algoritmos que influenciam seu comportamento assintótico. De forma principal, o número de objetos a serem transportado implica em uma complexidade $\mathcal{O}(n * m)$, com n objetos e cada um possuindo em média m segmentos, pois os agentes devem alocar todos os segmentos de todos os objetos. Outro ponto importante, está na utilização do Algoritmo de Húngaro, o qual possui complexidade de $\mathcal{O}(n^3)$, no qual n é o valor máximo entre a quantidade de objetos e agentes, ou seja, se em um exemplo existem 3 robôs e 2 objetos, n será 3. Este algoritmo, apesar de possuir implementações de alta performance, pode se tornar uma etapa que consuma bastante tempo, quando o valor de n é consideravelmente grande.

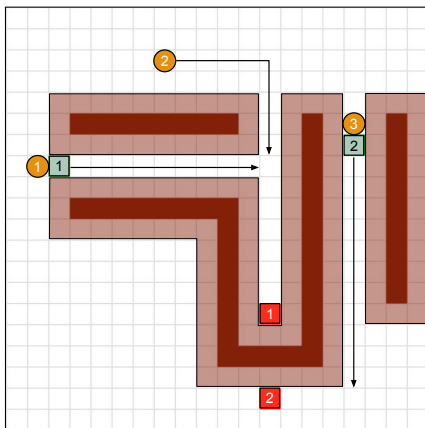
A Figura 3.4 ilustra a execução da tarefa de transporte de objetos usando uma equipe de três agentes terrestres para manipular 2 objetos em um ambiente de exemplo. As etapas descritas na figura demonstram que o algoritmo de alocação considera principalmente a distância percorrida pelos agentes, pois no caso de uma equipe homogênea, somente esta dimensão é diferenciada entre os planos.



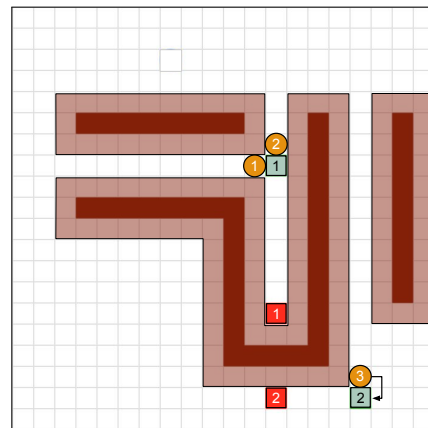
(a) A etapa de planejamento para os objetos é executada.



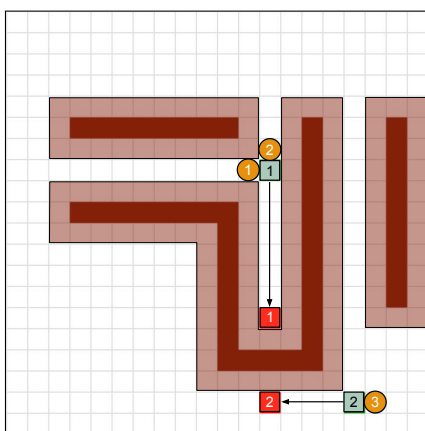
(b) Pela proximidade dos objetos, os Agentes 1 e 3 são designados para os Objetos 1 e 3 respectivamente. Agente 2 não é alocado neste momento.



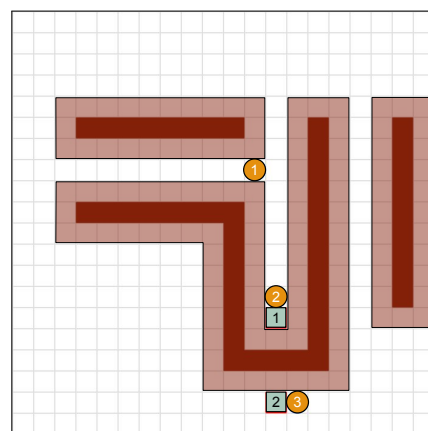
(c) Plano de transporte é executado por cada agente em seu respectivo objeto.



(d) O agente 2 é alocado para o objeto 1, pois os demais agentes estão impossibilitados. Agente 3 continua responsável pelo objeto 2.



(e) Agentes se aproximam dos objetos a serem transportados.



(f) É realizada a movimentação final dos objetos para seus locais de destino.

Figura 3.4: Ambiente de Exemplo com 2 Objetos e 3 Robôs. É demonstrada a execução de todas as etapas da tarefa de transporte por uma equipe de agente autônomos.

Como mencionado anteriormente, toda a etapa de planejamento e alocação das tarefas é executada antes que os agentes iniciem o processo de transporte, desta maneira cada agente já está ciente das tarefas que deve realizar.

3.6 Coordenação

Coordenação é a etapa de controle e ordenação de um processo para realização de uma tarefa ou atividade. Para realizar o transporte de objetos por um equipe de agentes, os mesmos devem ser gerenciados de modo a executar suas tarefas na ordem correta e de forma coordenada.

Durante a execução do transporte, um objeto a ser transportado não pode ser manipulado por mais de um agente ao mesmo tempo, de modo que somente um agente esteja autorizado a realizar sua tarefa em uma dada fase do transporte. O controle de autorização é feito mediante a troca de recursos nomeados *token*, que são compartilhados pelos agentes. Esta técnica é baseada no protocolo de comunicação de redes de computadores *Token Ring*, no qual uma estrutura especial chamada *token* circula pela rede, e somente o sistema que possuir tal informação é capaz de enviar dados pela rede. No sistema implementado, existem várias estruturas *token*, cada uma possui uma referência para um determinado objeto, que define a permissão para o agente.

Como mencionado na Seção 3.5, as tarefas a serem realizadas pelos agentes são os segmentos de cada objeto, de modo que para executar o segmento s_{i+1} , o segmento antecessor s_i deve ter sido completado anteriormente. Esta precedência é válida e necessária em qualquer momento durante o transporte, pois um segmento futuro inevitavelmente não pode ser realizado, considerando que o objeto não estará na posição S_o demarcada pelo mesmo. Os segmentos são uma sequência de ações que devem ocorrer na forma ordenada que foram planejados e de nenhuma outra maneira.

O Algoritmo 3.6 é executado em cada agente do sistema e propõe o método de coordenação da equipe mediante a troca das estruturas *token*, de modo a autorizar ou não algum agente a realizar suas tarefas para um determinado objeto. Similar ao planejamento dos objetos, inicialmente um agente arbitrário possui todos os *tokens* de todos os objetos, e mediante a execução do algoritmo os mesmos são enviados e recebidos entre a equipe.

Após a etapa de alocação de tarefas, cada agente possui sua própria lista de tarefas τ que deve completar. No processo de coordenação, o agente executa ordenadamente suas tarefas utilizando a lista como uma fila, ou seja, as primeiras tarefas adicionadas também serão as primeiras a serem executadas. A função *get_first_task()* retorna a

Algorithm 3.6 CoordinationProcess(τ)

```

1: loop
2:   task  $\leftarrow$  get_first_task( $\tau$ )
3:   if task = NULL then
4:     break
5:   end if
6:
7:   if own_token(task) then
8:     token  $\leftarrow$  get_token(task)
9:   else
10:    token  $\leftarrow$  request_token(task)
11:    save_token(token)
12:   end if
13:
14:   if not token = NULL then
15:     execute_prepare_plan(task)
16:     execute_transport_plan(task)
17:     drop_task(task)
18:   end if
19:
20:   share_ready()
21:   wait_until_reveice_all_ready()
22: end loop

```

primeira tarefa seguindo este modelo de fila, ou retorna vazio (*null*) se o agente não possuir mais tarefas a serem realizadas.

Uma vez com a tarefa selecionada, o agente testa se já possui o *token* referente ao objeto a ser transportado (função *own_token()*), se possuir tal *token*, o agente necessita recuperar de sua lista de *tokens*, caso contrário, realiza uma requisição para os demais agentes através da função *request_token()*, que será melhor explanado a seguir, pois um algoritmo de resposta deve ser executado nos agentes que recebem a requisição. Quando a requisição for realizada com sucesso, o *token* é transferido entre os agentes, e o robô o salva internamente (função *save_token()*).

Mediante o *token* de autorização, o agente é capaz de realizar ambos os planos relacionados à tarefa, de preparação (*execute_prepare_plan()*) e de transporte (*execute_transport_plan()*). Após a execução destes passos, o agente pode descartar a tarefas e marcá-la como realizada (função *drop_task()*).

Neste momento, em que a tarefa atual foi realizada, o robô avisa aos demais agentes de seu estado (função *share_ready()*) e entra em modo de espera, até que toda a equipe informe o mesmo (função *wait_until_reveice_all_ready()*).

Durante a execução das tarefas, os agentes realizam diversas trocas das estruturas *token*, requisitando autorização para realizar suas próximas tarefas. Sempre que um agente realiza o pedido para a equipe, o Algoritmo 3.7 é executado de modo a processar o pedido.

O algoritmo recebe como parâmetro a tarefa vinda do outro agente (*foreign_task*), e realiza uma comparação com as suas próprias tarefas. Se o agente no primeiro momento não possuir o *token* referente ao objeto da tarefa externa, não há necessidade de realizar demais testes, e retorna vazio após o teste da função *own_token()*.

No caso em que o agente requisitado possuir o *token*, o algoritmo é continuado realizando algumas validações na lista de tarefas τ do mesmo. Para cada tarefa da lista, dois dados são comparados: (i) se ambas as tarefas (do agente e a tarefa externa) se referem ao mesmo objeto (função *get_object()*), e (ii) se o teste anterior for verdadeiro, é testado se a tarefa possui um índice menor de segmento, ou seja, se a tarefa deve ser realizada antes da tarefa externa, neste caso, o agente não pode autorizar a realização da tarefa e retorna vazio (*null*). Na situação em que todas as tarefas do agente não atenderem esses testes, o mesmo disponibiliza seu *token* para o agente requerente (função *get_token()*).

Algorithm 3.7 ResponseTokenProcess(*foreign_task*)

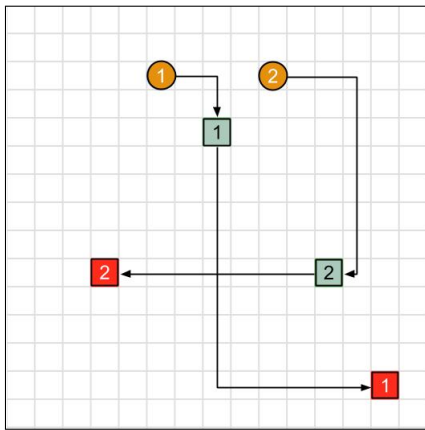
```

1:  $\tau \leftarrow \text{get\_tasks}()$ 
2:  $o_f \leftarrow \text{get\_object}(\text{foreign\_task})$ 
3:  $s_f \leftarrow \text{get\_segment}(\text{foreign\_task})$ 
4: if not own_token(task) then
5:   return NULL
6: end if
7:
8: for all task  $\in \tau$  do
9:    $o_i \leftarrow \text{get\_object}(\text{task})$ 
10:   $s_i \leftarrow \text{get\_segment}(\text{task})$ 
11:  if  $o_f = o_i$  and  $s_i < s_f$  then
12:    return NULL
13:  end if
14: end for
15: return get_token( $o_f$ )

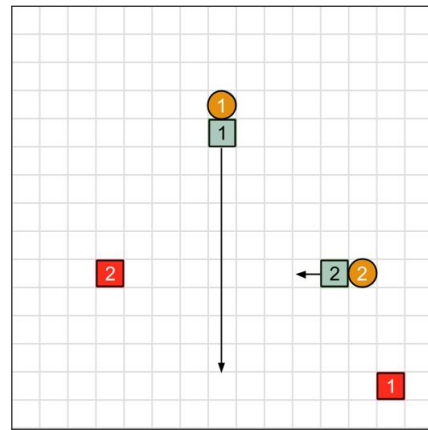
```

Ao considerar que mais de um agente pode mover-se ao mesmo tempo, e as rotas de movimentação podem se interceptar, durante a execução dos trajetos, os robôs mantêm uma comunicação com os demais membros da equipe, informando sua posição atual e o plano que está executando. Os agentes realizam testes de intersecção entre os planos que estão executando, e se for detectada uma eminente colisão, um sistema

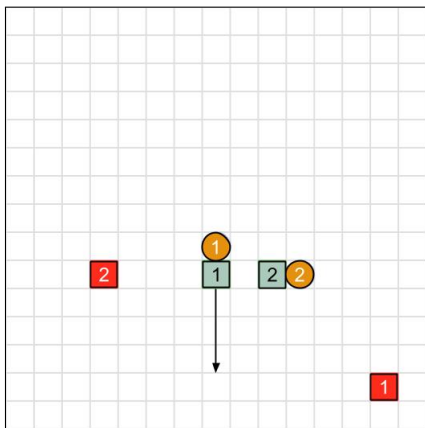
de prioridade é aplicado seguindo uma regra de maior distância, ou seja, mediante o tamanho do plano a ser executado (quantidade de passos a serem executados), o agente que possuir a menor tarefa deve esperar até que o outro robô envolvido no processo prossiga com seu plano. Esta estratégia é exemplificada na Figura 3.5.



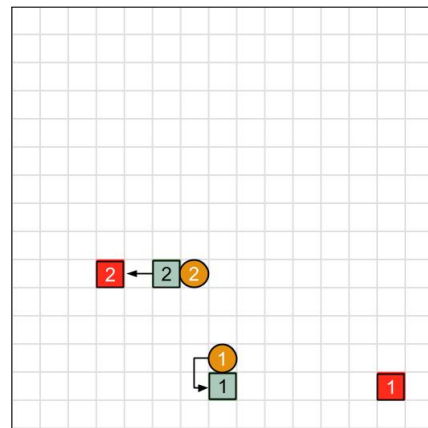
(a) Ambiente de Exemplo: 2 Objetos, 2 Robôs. É executado o planejamento para os objetos e a alocação. É demonstrado os planos de preparação dos robôs.



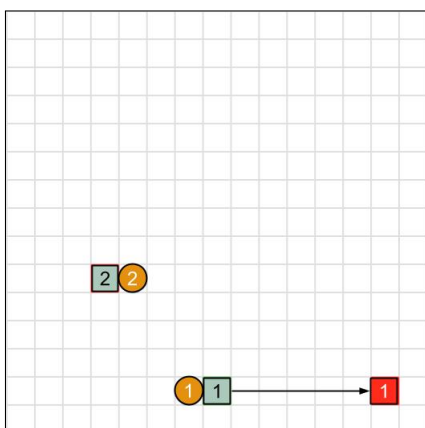
(b) Durante o transporte, é verificado que o plano do objeto 1 é maior, portanto tem prioridade sobre o objeto 2.



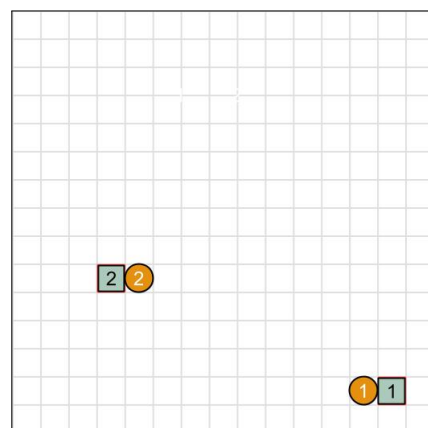
(c) O Agente 2 executa parte do plano total, e espera a passagem do outro robô.



(d) Após o Agente 1 executar seu plano, o Agente 2 pode continuar o transporte do Objeto 2.



(e) Agente 2 finaliza o transporte. Agente 2 realiza o movimento de preparação.



(f) O transporte de todos os objetos é concluído.

Figura 3.5: Exemplo de Coordenação entre 2 Agentes durante o transporte de objetos demonstrando a técnica aplicada quando há uma intersecção entre os planos de movimentação.

Capítulo 4

Experimentos

Neste capítulo serão demonstrados os resultados obtidos com a utilização da metodologia apresentada. Esta experimentação tem por objetivo expor a efetividade da técnica apresentada através de sua aplicação em diversos cenários. Os experimentos realizados estão divididos em três categorias: (i) um estudo quantitativo, que demonstra um conjunto de métricas e comparações entre casos distintos, (ii) ambiente simulado no qual são utilizados agentes virtuais para realização do transporte, e (iii) ambiente real, utilizando plataformas robóticas reais para transportar um objeto.

A metodologia apresentada foi totalmente implementada utilizando a linguagem de programação *Python*¹, fazendo uso de algumas bibliotecas, tais como: *networkx*, *shapely*, *descartes*, *numpy*, *sympy*, *matplotlib*, dentre outras. O computador utilizado durante os experimentos e simulações possui como configuração de *hardware*: processador Intel Core i7-4500U CPU 1.80GHz*4, 8Gb de memória RAM, 128Gb SSD, Sistema Operacional Ubuntu 14.04, 64Bit.

¹<https://www.python.org/>

4.1 Estudo Quantitativo

Nesta seção são discutidos os experimentos realizados com a utilização da técnica apresentada no Capítulo 3, com a finalidade de avaliar as características que influenciam a tarefa de transporte cooperativo, realizando diversas comparações sobre as possíveis configurações do sistema, como a complexidade do ambiente de trabalho, quantidade de objeto e robôs, e como estes dados refletem na mudança de comportamento do método. Uma análise detalhada é apresentada em cada uma das próximas subseções, nas quais o elemento em estudo é sempre variável, enquanto demais características permanecem fixas, proporcionando um melhor entendimento de como uma certa propriedade influencia todo o sistema.

Inicialmente é apresentado um resultado para ilustração dos mapas e experimentos demonstrados a seguir. Como mencionado anteriormente, o ambiente de trabalho do transporte é um espaço discretizado, formado por uma matriz de três dimensões, possuindo células de tamanho unitário. Desta maneira, todos os experimentos que se referem ao deslocamento dentro do espaço, discutem a distância percorrida em unidades, ou seja, a quantidade de células a serem percorridas para realizar a movimentação de um ponto A para outro ponto B no mapa. A Figura 4.1 demonstra um mapa de exemplo no qual um agente deve transportar um objeto. O trajeto vermelho é o plano criado para o objeto, partindo do ponto verde até a posição assinalada pelo ponto vermelho. O plano de movimentação do agente é demarcada por uma linha em azul, indicando todas as células percorridas pelo mesmo durante a execução da tarefa, que por exemplo, no primeiro plano de movimentação o agente percorreu um plano de tamanho 15.

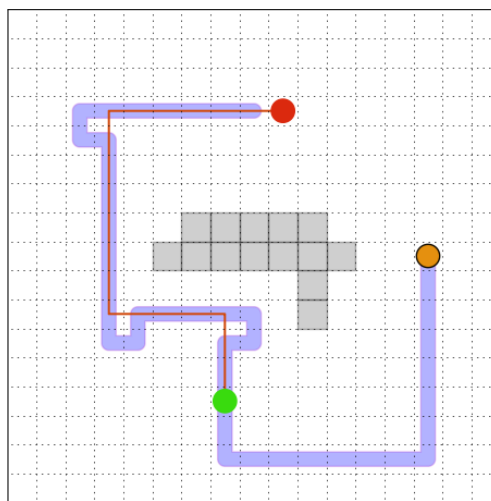


Figura 4.1: Mapa de exemplo do transporte de um objeto por um único agente, destacando os trajetos realizados tanto pelo objeto quanto pelo agente.

4.1.1 Complexidade do Ambiente

Ao considerar que o transporte dos objetos sofre interferência direta do ambiente em que está sendo realizado, neste estudo, serão utilizados diferentes mapas, com variáveis níveis de complexidade e quantidade de obstáculos, apresentando como o sistema se comporta mediante estas diferentes circunstâncias.

A Figura 4.2 apresenta exemplos de mapas utilizados nos experimentos desta seção. Nestes cenários, a quantidade de objetos e robôs foi mantida fixa durante os experimentos (3 robôs e 3 objetos), havendo somente o incremento da complexidade do ambiente, possuindo mais obstáculos, com diferentes configurações.

A principal intenção deste experimento foi demonstrar que o sistema, além de ser capaz de tratar uma diversidade de mapas, não sofre uma grande influência da disposição dos obstáculos no ambiente, apesar de, inevitavelmente, um ambiente com grande restrição de movimentação contribuir em um aumento na distância percorrida pelos agentes.

Os experimentos realizados seguiram as seguintes regras: (i) cada cenário específico foi executado 10 vezes para obtenção dos dados relativos ao tempo médio de planejamento, e (ii) foram realizadas 3 variações aleatórias do posicionamento dos objetos e agentes dentro do ambiente para fins demonstrativos.

Os resultados dos experimentos realizados nestes ambientes podem ser vistos nos gráficos das figuras 4.3, 4.4, que demonstram o deslocamento total dos objetos e agentes, respectivamente. Como pode ser observado, o tamanho do plano de deslocamento dos objetos não segue diretamente um padrão relacionado a complexidade do ambiente, e principalmente à disposição dos mesmos, podendo haver um acréscimo ou decréscimo mediante a relação entre o ponto inicial e final. O mesmo ocorre com o deslocamento dos robôs, que seguem um modelo similar ao plano dos objetos, quando os planos de transporte são reduzidos, a movimentação dos agentes também é reduzida.

Em contra-partida, o gráfico de tempo de planejamento, demonstrado na Figura 4.5, aponta para o aumento na demanda de tempo necessário durante o planejamento dos planos. Isso era esperado, porque em um ambiente mais restrito, o objeto geralmente deve realizar uma quantidade maior de mudanças de direção durante o transporte, influenciando a etapa de planejamento dos agentes, que devem criar mais planos para cada segmento. Desta maneira, pode-se concluir que o ambiente certamente gera grande influência sobre o comportamento geral do sistema.

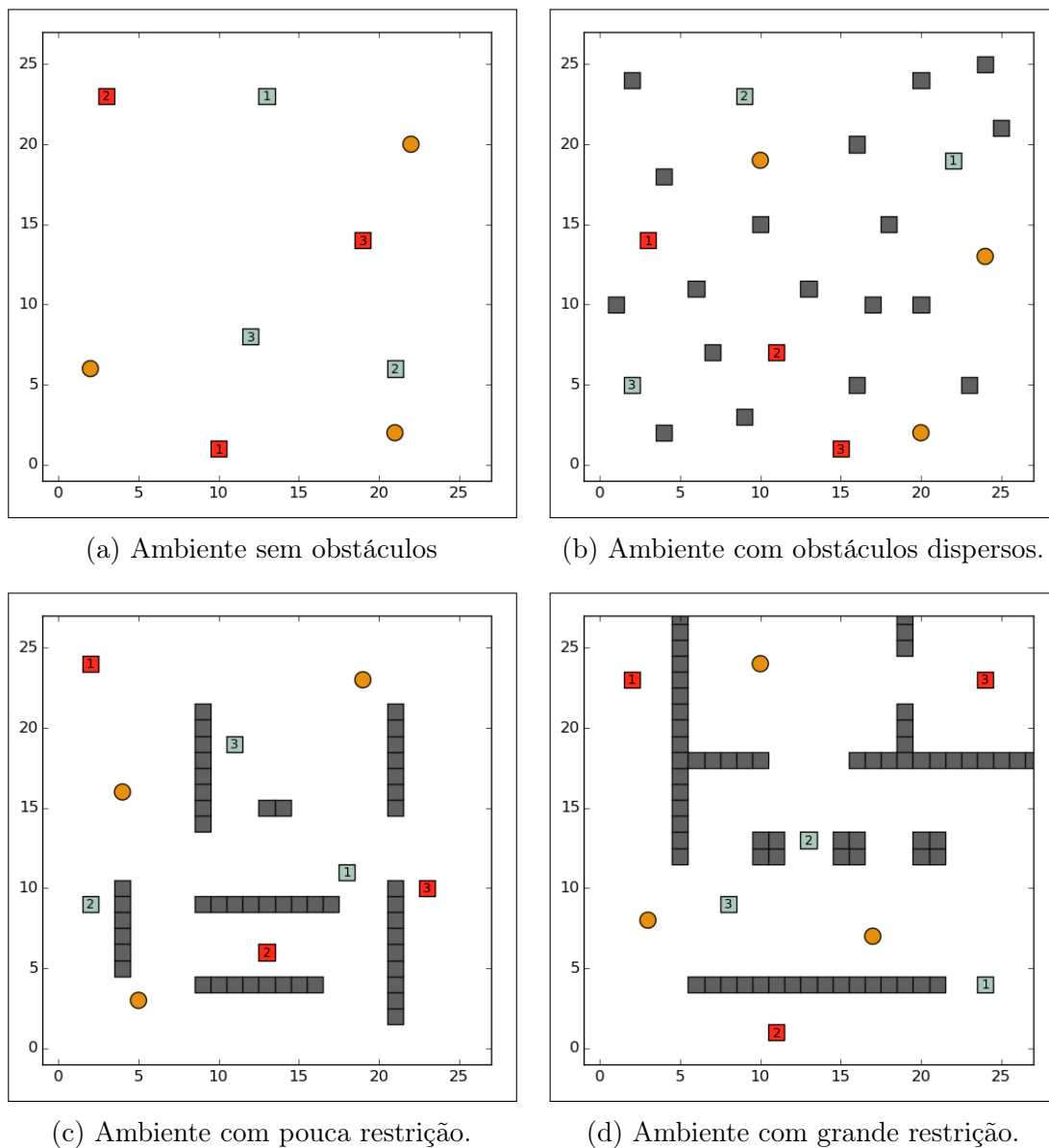


Figura 4.2: Cenários utilizados nos experimentos relacionados à complexidade do ambiente. Os mapas foram criados buscando descrever cenários que possuam diferentes classes de mobilidade, partindo desde o exemplo onde não existem obstáculos, até outros que restringem drasticamente a movimentação tanto de objetos quanto agentes. Cada objeto é demarcado por um identificador numerado, descrevendo o ponto inicial (caixa azul) até seu respectivo destino (caixa vermelha). Os agentes são representados por círculos laranjas.

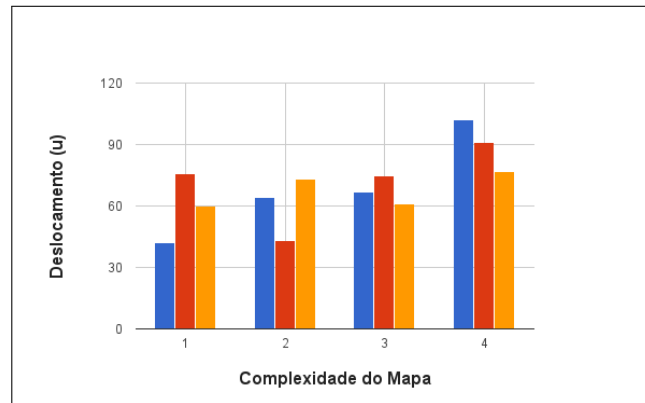


Figura 4.3: Gráfico do deslocamento do objeto em ambiente com diferentes complexidades. São exemplificados três amostras onde os agentes são dispostos em posições diferentes.

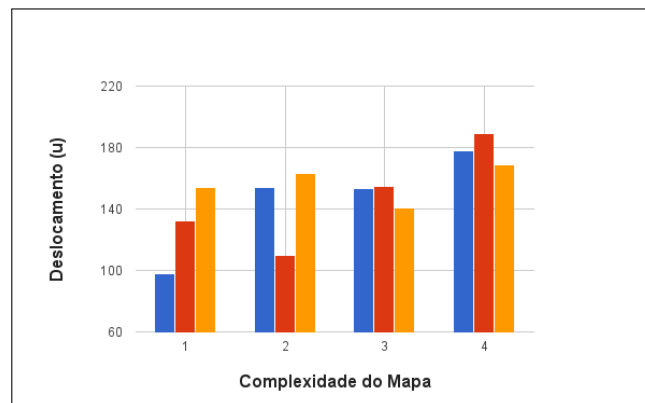


Figura 4.4: Gráfico de deslocamento dos agentes em cenários com variados graus de restrição de movimentos. Cada amostra demonstrada, representa cenários com agentes em posições distintas.

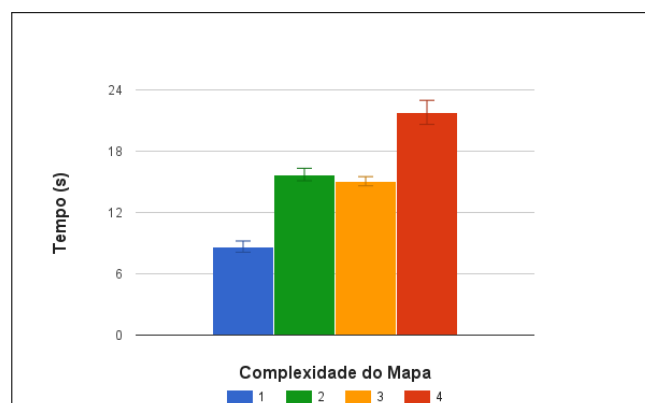


Figura 4.5: Gráfico do tempo médio necessário para o planejamento de todos os passos a serem executados durante o transporte em cenários com distintos níveis de complexidade.

4.1.2 Deslocamento e Tempo de Planejamento dos Agentes

Para realizar um estudo específico do planejamento dos agentes, é utilizado um ambiente fixo de complexidade média, como o demonstrado na Figura 4.6. Será apresentado um conjunto de experimentos expondo a relação entre a quantidade de robôs e a quantidade de objetos a serem transportados. Quanto maior a equipe, existe uma maior possibilidade de haver agentes mais aptos para o transporte, porém, como será demonstrado a seguir, em alguns casos, uma equipe maior prejudica a execução total.

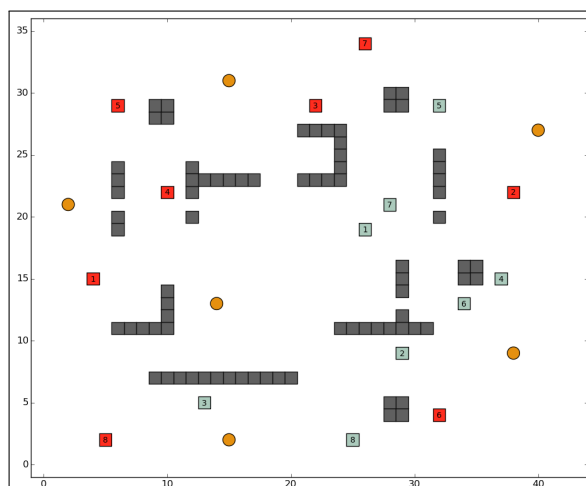


Figura 4.6: Mapa utilizado nos testes de deslocamento e tempo de planejamento dos agentes. Neste caso específico, é exibida uma equipe com 6 agentes tendo de transportar 8 objetos dispersos no ambiente.

Esta análise visa demonstrar como o sistema se beneficia do tamanho da equipe além de ilustrar o seu comportamento quando há um certo volume de objetos. Os experimentos realizados contemplam equipes de 1 até 6 agentes, e o conjunto de objetos variando desde 1 até 8 itens.

Como discutido no Capítulo 3, o método aqui proposto aplica uma penalização nos planos criados para os objetos e agentes, sempre que existe uma mudança de direção, criando planos mais retilíneos, ou seja, planos que não alternam excessivamente a direção durante o transporte. Esta penalização se mostrou importante no decorrer da manipulação, pois diminui a distância percorrida pelos agentes, além de minimizar o custo em tempo necessário para o planejamento.

Para exemplificar visualmente esta diferença, que será discutida a seguir, são apresentados dois gráficos na Figura 4.7, que ilustram somente o deslocamento de um agente para transportar 3 objetos, nos casos sem e com a penalização da mudança de direção. Nos mapas existem 3 áreas destacadas, onde é possível perceber que no exemplo sem a penalidade (Figura 4.7a), o agente realiza uma série de mudanças de

direção que não são realizadas no caso em que há a penalidade (Figura 4.7b). Desta maneira, o agente tende a se deslocar menos durante toda a tarefa de transporte, além de conseguir completar a mesma tarefa em um menor tempo.

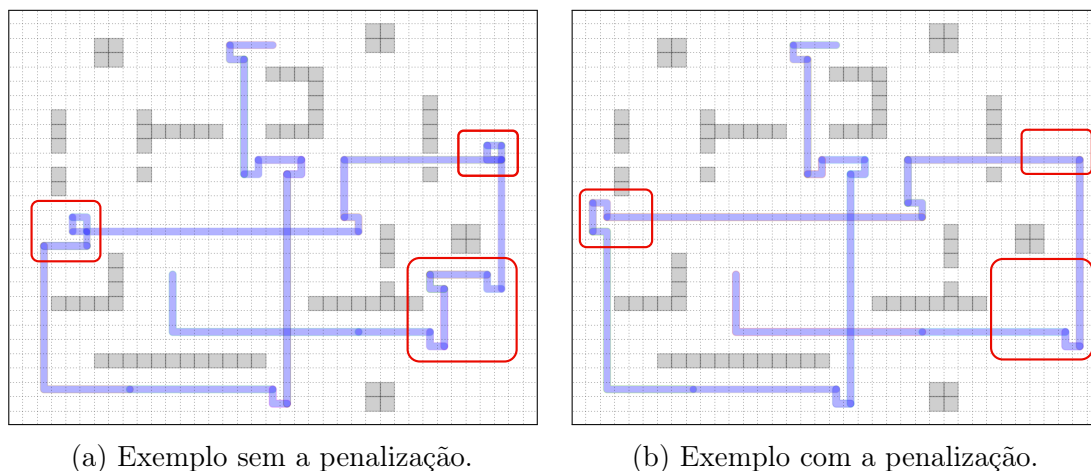
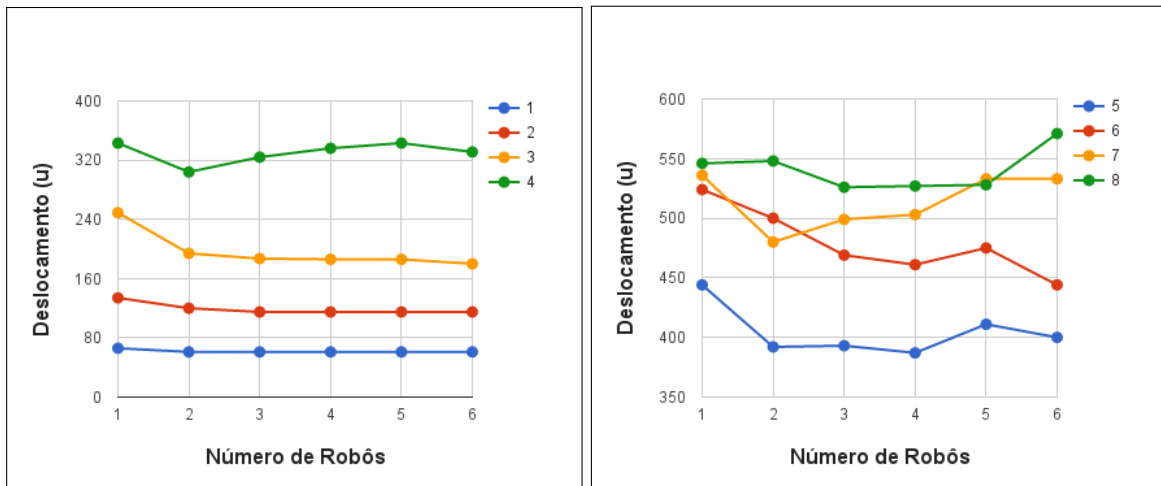


Figura 4.7: Mapas de deslocamento de um agente demonstrando o transporte de um conjunto de objetos quando é não aplicada a penalização de mudança de direção e o caso onde a penalidade foi utilizada.

Os gráficos a seguir representam o resultado de vários testes realizados, exibindo tanto dados referentes ao deslocamento total dos agentes, quanto ao tempo médio necessário para o planejamento das ações. Foram executados testes tanto aplicando a penalização da mudança de direção quanto sem a mesma. Para uma melhor visualização, foram criados gráficos separando as quantidade de objetos, de 1 a 4 e de 5 a 8, pois a visualização de todos os dados em um só gráfico não permitia uma análise direta devido à escala aplicada nos eixos do gráfico. Cada gráfico exibe em seu eixo horizontal a quantidade de agentes utilizados no experimento, além de representar certas quantidades de objetos sendo transportados por curvas no gráfico.

Os gráficos das figuras 4.8 e 4.9 demonstram o somatório do deslocamento de todos os agentes, respectivamente os casos onde não foi aplicada a penalização e quando a mesma foi utilizada. O mesmo pode ser visto nas figuras 4.10 e 4.11, porém exibindo os dados de tempo de planejamento, no cenário sem e com a penalização.

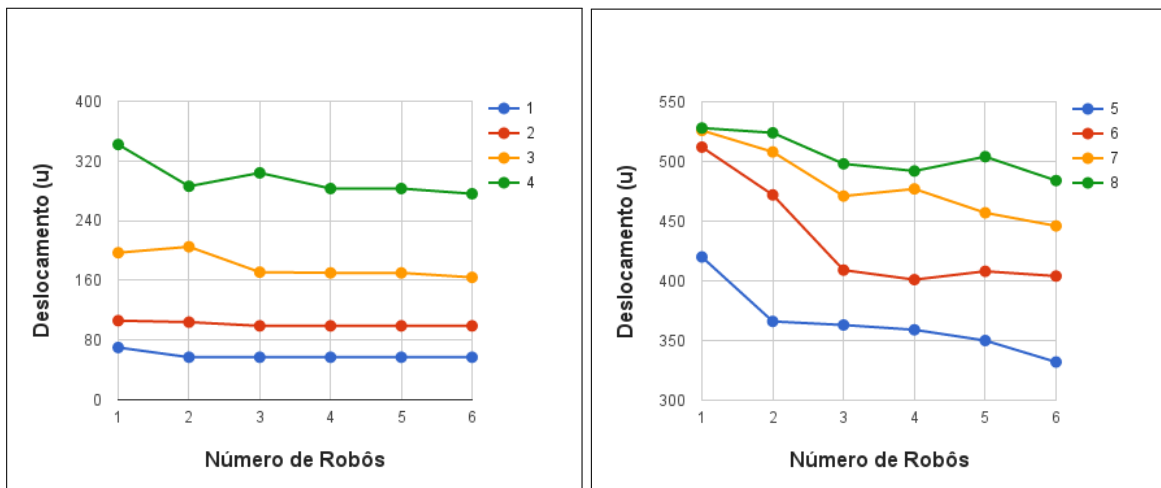
Ao analisar os gráficos de deslocamento, é possível notar que nos casos em que não há a penalização da mudança de direção, o sistema não se beneficia tanto da quantidade de agente, que contrariamente ao esperado, aumenta a distância percorrida quando a equipe aumenta de tamanho. Comportamento este que não é observado quando é aplicada a penalização, em que a tendência do sistema é necessitar de uma menor quantidade total de deslocamento para realizar o transporte dos objetos. A diminuição nos trajetos era esperada, considerando que uma maior quantidade de agentes garante



(a) 1 a 4 objetos.

(b) 5 a 8 objetos.

Figura 4.8: Deslocamento total dos agentes quando não foi aplicada a penalidade de mudança de direção.



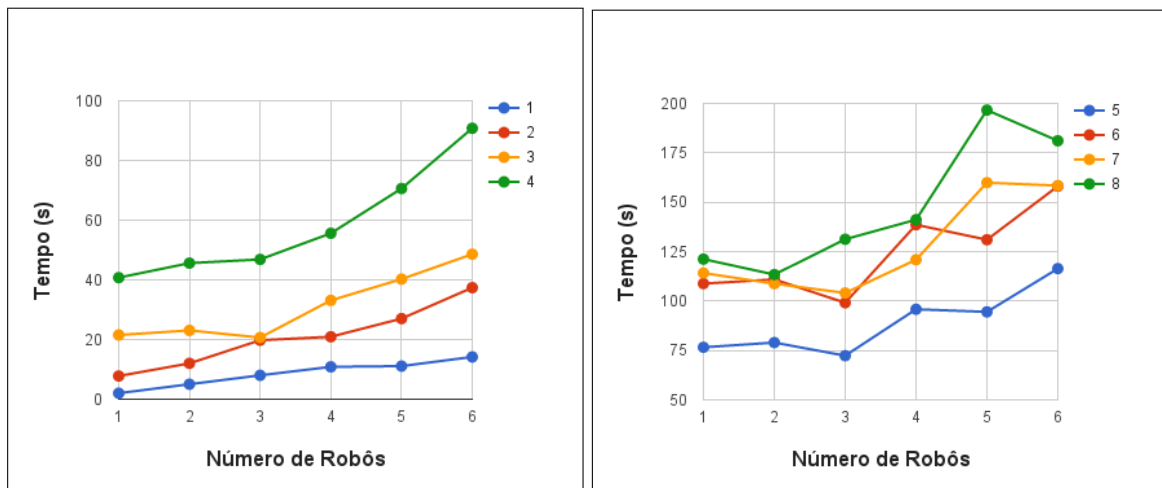
(a) 1 a 4 objetos.

(b) 5 a 8 objetos.

Figura 4.9: Deslocamento total dos agentes com a aplicação da penalidade de mudança de direção.

uma maior diversidade de opções para transportar um determinado objeto, criando mais chances de existirem robôs com planos melhores, contribuindo para minimizar a movimentação geral.

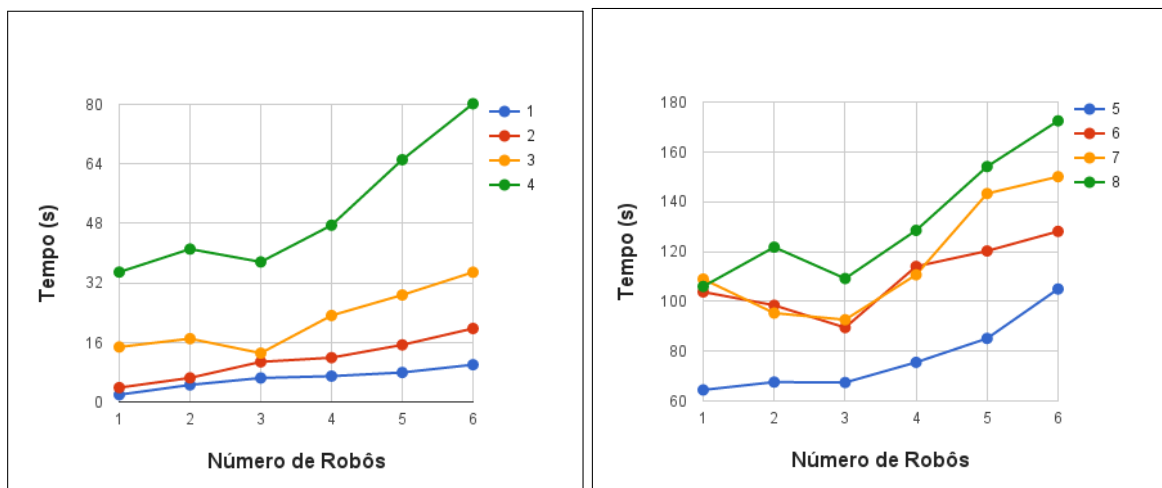
Apesar da diminuição na movimentação dos agentes com a penalização, em alguns casos, é notado que o aumento da quantidade de agentes, mesmo que sutil, faz com que os agentes andem mais que nos casos com uma menor equipe, como na situação do transporte de 8 objetos por 4 e 5 agentes. Isso é explicado pelo fato do algoritmo de alocação de tarefas (Seção 3.5) sempre alocar tarefas para todos os agentes, mesmo que



(a) 1 a 4 objetos.

(b) 5 a 8 objetos.

Figura 4.10: Tempo de planejamento dos agentes sem a aplicação da penalidade de mudança na direção.



(a) 1 a 4 objetos.

(b) 5 a 8 objetos.

Figura 4.11: Tempo de planejamento dos agentes com a aplicação da penalidade de mudança na direção.

a melhor configuração fosse deixar um determinado robô sem nenhuma tarefa, afetando assim o deslocamento total dos agentes. Mesmo com este acréscimo, deve ser lembrado que com mais agentes, executando mais tarefas ao mesmo tempo, o tempo de execução total da tarefa reduzirá.

Em relação ao tempo de planejamento, um efeito contrário é observado. Seja com o aumento no número de objetos ou no número de agentes, o sistema tende a consumir mais tempo para planejar todas as rotas necessárias para o transporte. O aumento no número de objetos faz com que os agentes tenham mais opções de planos a

Tabela 4.1: Tabela com as variações nos pesos utilizados para realização de testes aplicados na função de utilidade durante a fase de planejamento dos objetos.

Experimento	Tempo	Energia
1	1	0
2	0	1
3	0.5	0.5

executar, já o aumento no número de agentes aumenta a quantidade de possíveis robôs para executar o transporte de um objeto, além de uma necessidade maior de tempo para que toda a equipe entre em acordo sobre quais tarefas executar. Mesmo com este comportamento, observa-se nos gráficos onde há a aplicação da penalidade, o tempo necessário para o planejamento é sempre menor ou bastante aproximado dos casos em que não é aplicada, demonstrando como esta estratégia contribui para o melhoramento da performance do sistema.

Estes fatores são interessantes de serem estudados pois ocasionam um *trade-off* entre a quantidade de robôs a ser utilizada, o tempo de planejamento e o ganho em deslocamento total. Encontrar a melhor combinação destes valores não foi objetivo deste trabalho, mas sim demonstrar que cada uma destas variáveis influencia diretamente na performance total do sistema.

4.1.3 Função de Utilidade

Durante o desenvolvimento do método, chegou-se à conclusão que seria possível criar um sistema no qual fosse possível configurar os algoritmos de planejamento de modo a atender diferentes demandas, criando planos que fossem ou mais rápidos de serem executados, ou mesmo com um menor custo energético para realização.

Esta configuração é possível através de um artifício na heurística utilizada no algoritmo modificado A* utilizado nas etapas de criação dos planos de movimentação. A função heurística descrita na Seção 3.2 exerce sua influência para modificar o plano.

Neste sentido, foram realizados experimentos em um mapa simples, como o demonstrado a seguir na Figura 4.12, modificando as constantes de ponderação em cada dimensão analisada, tempo e energia. A Tabela 4.1 expõe os pesos utilizado em cada experimento.

Um fato importante de ser mencionado é que estes experimentos, apesar de serem reflexo da influência da função de utilidade, só possuem sentido levando em consideração as características dos agentes utilizados durante o planejamento. Estes parâmetros são utilizados durante os cálculos para decidir a utilidade de certa ação de movimento

Tabela 4.2: Dados referente às características dos tipos de agentes utilizados durante os experimentos da função de utilidade.

Tipo	Energia	Tempo
Terrestre	1	8
Aéreo	12	1

realizada por cada tipo de agente, e assim, decidir o melhor caminho que atenda às configurações previamente estabelecidas pelas constantes de ponderação.

Os experimentos ilustrados nos mapas da Figura 4.12, utilizam dois tipos de agentes, um terrestre e outro aéreo. Para averiguar as características dos mesmo, foram realizados dois tipos de testes: (i) a velocidade média de deslocamento entre dois pontos fixos do ambiente, considerando somente a movimentação em linha reta, e (ii) o tempo médio de duração da bateria em uso contínuo. Uma vez realizados estas provas, a tabela 4.2 apresenta os valores estimados para cada tipo de agente. Estes dados são demonstrados de forma unitária, ou seja, o agente com melhor performance em uma determinada característica possui valor 1, e o outro agente possui um valor determinado em quantas vezes o mesmo é pior naquela propriedade. Estes dados não são fixos ou finais, pois estão diretamente relacionados ao tipos de agentes utilizados durante os testes, podendo haver uma grande diferença se outras plataformas robótica fossem utilizadas durante as provas.

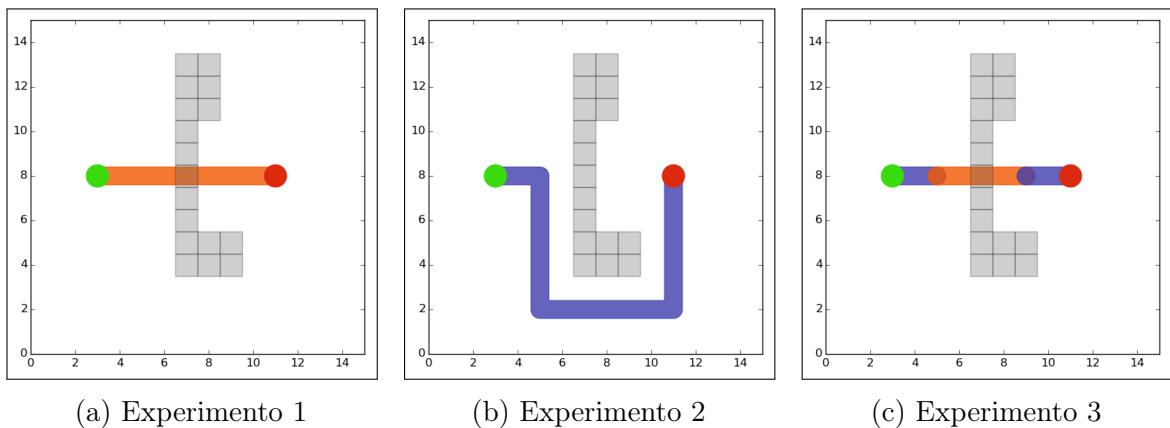


Figura 4.12: Exemplos de planos de movimentação de um objeto mediante a troca das constantes de ponderação na função de utilidade utilizada como heurística do planejador. O caminho demarcado em laranja representa um trajeto aéreo, enquanto o lilás é terrestre.

Mediante os dados expostos, é possível observar como a mudança nos pesos da função de utilidade é capaz de atuar sobre os planos criados, tornando o método apresentado flexível, passível de ser configurado para atender a diferentes necessidades.

4.1.4 Sequenciamento do Transporte

Um importante fator durante a ação de transporte de objetos é a escolha da sequência na qual estes objetos serão manipulados, pois, uma ordenação mal determinada pode levar os agentes a transcorrer grandes distâncias para realizar o deslocamento.

Neste experimento serão realizados experimentos comparando dois tipos de transporte: (i) sequencial – na qual os objetos são manipulados em uma ordem pré-definida, não avaliando qualquer custo excessivo associado ao deslocamento dos agentes, e (ii) oportunista – onde é avaliado o melhor objeto a ser transportado em um determinado momento baseado na localização do agente, procurando sempre minimizar a distância percorrida.

De forma demonstrativa, são exibidos na Figura 4.13 dois gráficos com o deslocamento de um agente realizando o transporte de 3 objetos, onde são destacadas as áreas de maior discrepância entre os resultados do tipo de transporte. O mapa utilizado durante este experimento é similar ao da Figura 4.14. Como pode ser observado, durante o transporte sequencial o agente realiza planos mais longos, pois esta métrica (tamanho do plano) não é levada em consideração durante a escolha do objeto, em comparação ao transporte oportunista, que sempre visa o objeto com melhor custo, este apresenta um comportamento com menor distância total percorrida nos planos.

O mapa apresentado na Figura 4.14 ilustra o ambiente no qual os demais experimentos foram realizados, nos quais também um agente foi utilizado, o que proporciona uma melhor visualização do seu deslocamento referente à sequência do transporte.

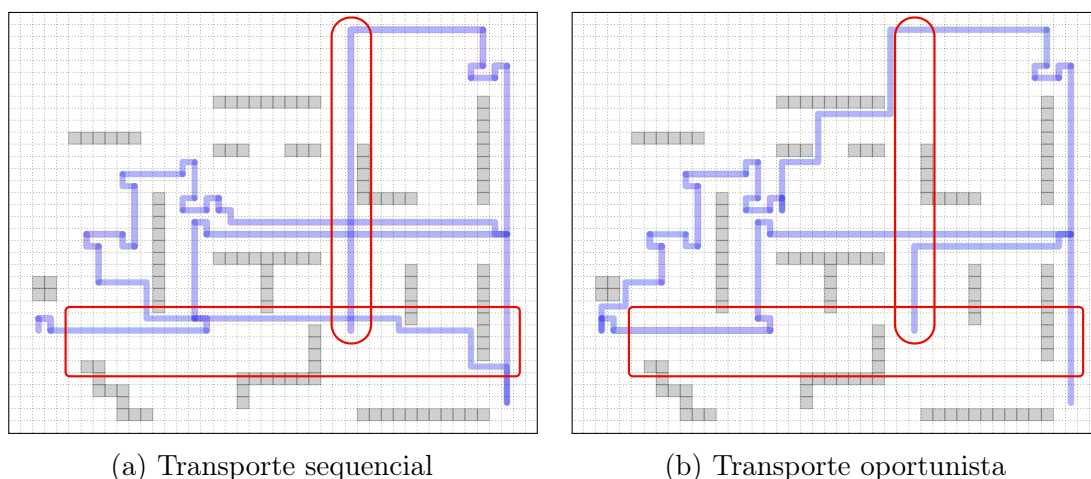


Figura 4.13: Resultado do transporte de 5 objetos por um agente utilizando o método sequencial e o oportunista.

A Figura 4.15 demonstra um gráfico comparativo entre as distâncias percorridas pelo agente quando o número de objetos a ser transportado aumenta. Mesmo

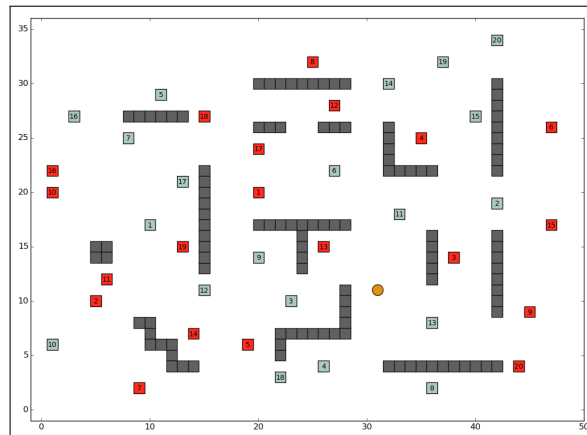


Figura 4.14: Mapa utilizado para avaliar a diferenciação entre os tipos de atitudes no transporte, sequencial e oportunista.

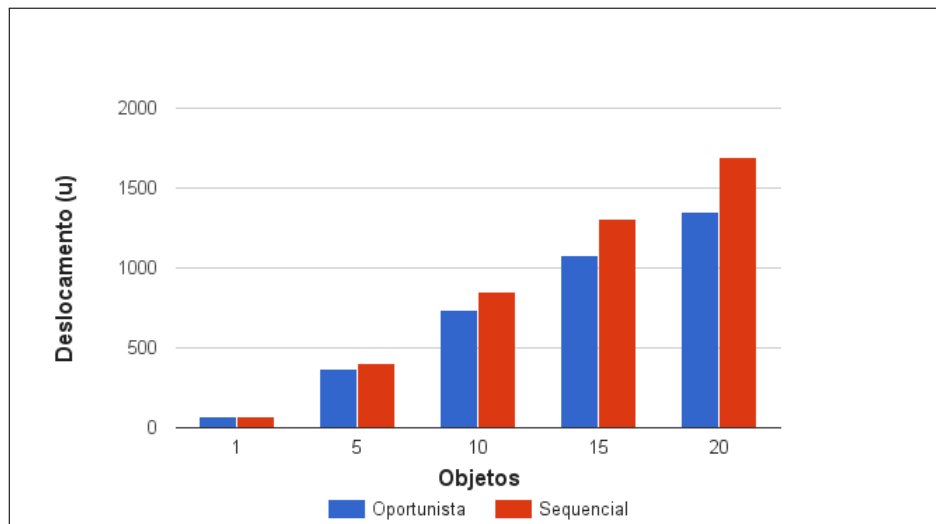


Figura 4.15: Deslocamento do robô mediante a adição de objetos utilizando dois tipos de transporte, sequencial e oportunista.

que os planos se tornem mais longos com a adição de objetos, que é um fato esperado, é possível perceber que o método oportunista faz um melhor uso da posição do robô, proporcionando uma menor travessia total e assim economizando tanto tempo como energia, pois estes custos estão diretamente relacionados ao tamanho do plano executado.

Apesar da vantagem apresentada, como é demonstrado no gráfico da Figura 4.16, o tempo necessário de planejamento no caso oportunista apresenta um comportamento de crescimento muito superior à técnica sequencial. Este desempenho é justificado ao considerar que o agente deve decidir qual o melhor objeto a transportar, e para tal, deve considerar planos de transporte para todos os objetos ainda não transportados,

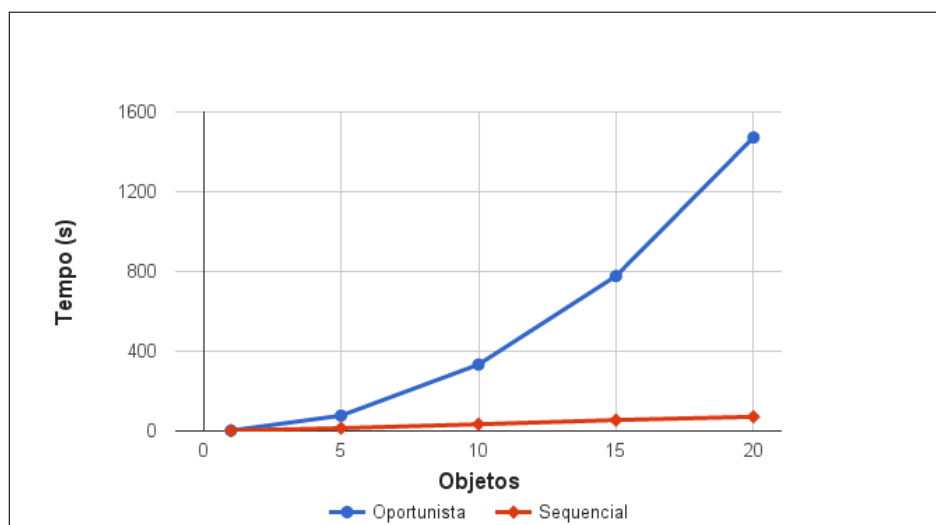


Figura 4.16: Tempo de planejamento do agente nas duas técnicas de transporte, sequencial e oportunista.

gerando uma maior demanda de tempo nesta etapa, diferente da técnica sequencial, na qual somente o objeto atualmente sendo manipulado é tido como base pra realização do planejamento.

Porém, é interessante ressaltar que, apesar do grande gasto de tempo durante o planejamento, este não afeta drasticamente o gasto energético do agente, pois o mesmo não está em movimentação, o que justifica o uso da técnica oportunista, considerando que o ganho proporcionado por um menor plano é superior ao ganho quando o planejamento é executado rapidamente.

4.2 Ambiente Simulado

Para realização dos testes simulados, foi utilizado o *Robot Operating System (ROS)*², que é um conjunto bibliotecas para desenvolvimento de plataformas autônomas largamente utilizado tanto em pesquisas quanto em aplicações comerciais. Especificamente a versão *Indigo* foi utilizada na implementação. Este sistema abstrai as camadas de envio e recebimento de dados, provendo acesso aos sensores e atuadores de diversas plataformas robóticas.

Os algoritmos desenvolvidos foram testados no ambiente de simulação *Gazebo*³. O mesmo já possui integração direta com o *ROS*, capaz de simular a dinâmica entre corpos virtuais utilizando diferentes motores de física (*ODE*, *Bullet*, *Simbody*, *DART*),

²www.ros.org

³<http://gazebosim.org/>

além de implementar uma diversidade de sensores (câmeras RGB-D, laser, torque) e possuir vários modelos pré-existentes para criação de ambientes realísticos, tanto de robôs, mas também inclui ferramentas, casas, veículos, dentre outros.

Para uma demonstração da técnica apresentada por este trabalho, serão documentados dois experimentos simulados, visando exemplificar o processo de coordenação entre diferentes agentes. Serão apresentados dois casos do transporte de um objeto: (i) dois agentes terrestres e (ii) um agente terrestre e outro aéreo.

A Figura 4.17 demonstra os ambientes de trabalho construídos para a simulação. Neles é possível observar a disposição do objeto a ser transportado, bem como dos agentes e obstáculos.

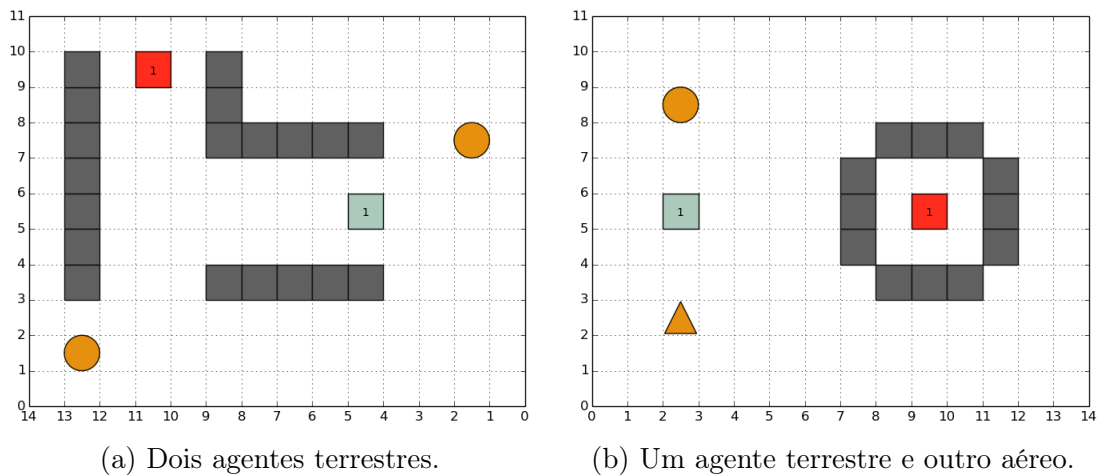


Figura 4.17: Ambientes de trabalho utilizados durante os experimentos simulados.

A Figura 4.18 ilustra o transporte de um objeto por uma equipe de dois agentes terrestres. Os agentes representados são da plataforma *iRobot Create*, robô diferencial com dimensões de 33cm de diâmetro e 12cm de altura, capazes de executar somente a ação de *PUSH* no objeto. Cada sub-figura ilustra um determinado estado do sistema com seu respectivo tempo de ocorrência.

Neste exemplo de execução, o agente mais próximo do objeto será alocado para realizar o primeiro segmento do transporte do objeto através do corredor formado pelos obstáculos (figuras 4.18a a 4.18d). Considerando que para continuar a execução do transporte, o primeiro agente deve andar uma grande distância, o segundo agente assume o objeto para realização do segundo segmento, levando-o até a posição final, demarcada por uma caixa verde semitransparente (figuras 4.18e a 4.18h).

No outro ambiente simulado apresentado na Figura 4.19, é demonstrado o transporte do objeto em questão utilizando agentes de tipos diferentes, um terrestre, da mesma plataforma utilizada anteriormente (*iRobot Create*), e um aéreo, que é repre-

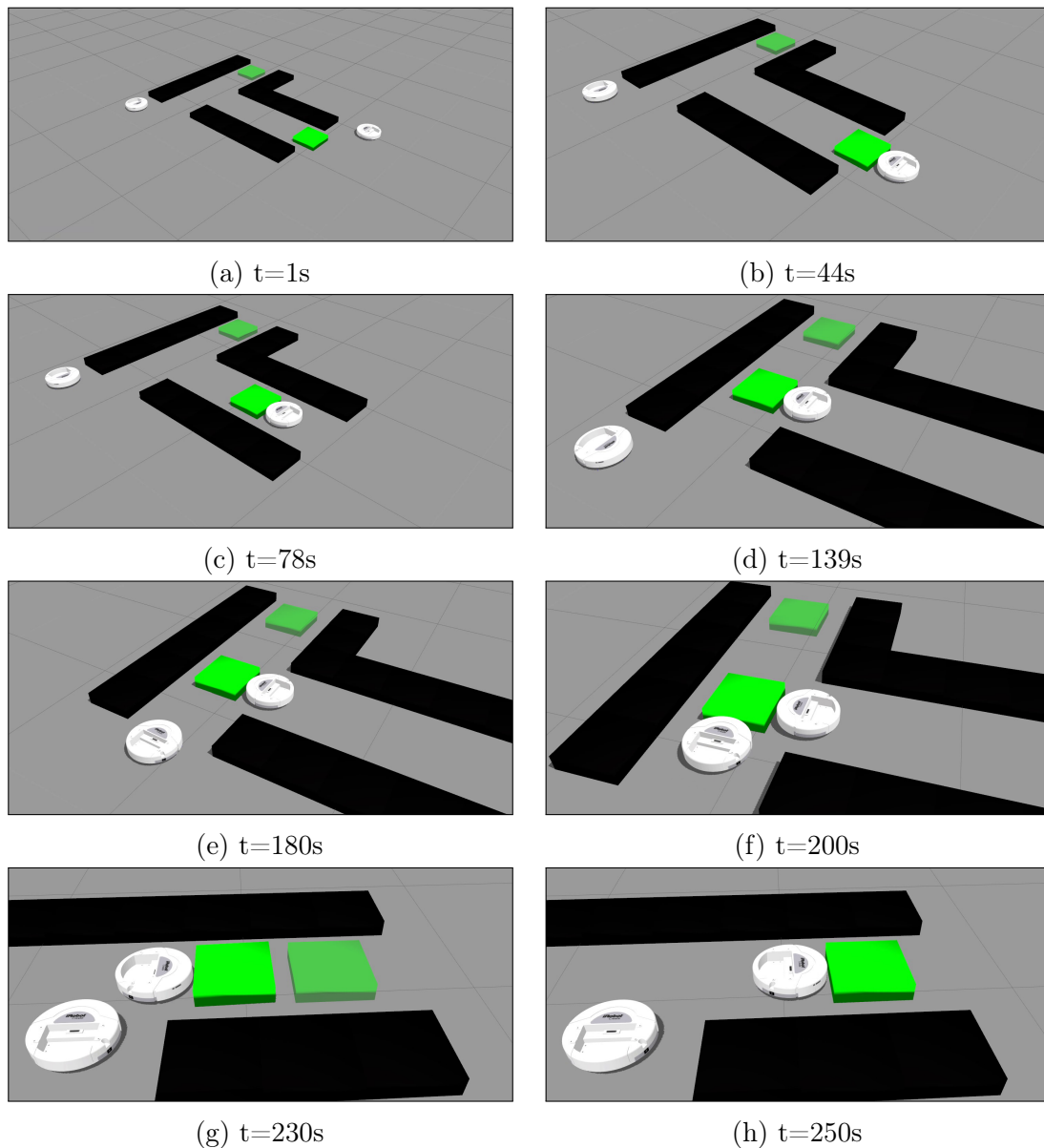


Figura 4.18: Exemplo de transporte por dois agentes em um ambiente simulado.

sentado por um robô genérico no modelo *quadrotor*, este tipo de agente foi escolhido por sua capacidade de pairar, sendo capaz de pegar o objeto ainda no ar utilizando a ação de *GRASP*.

Diferente do ambiente anterior, neste novo cenário o objeto não tem acesso direto ao seu destino por vias terrestres, sendo necessário o uso do agente aéreo. Em um primeiro momento, visto nas figuras 4.19a até 4.19e, o agente terrestre transporta o objeto até o local mais próximo possível de sua posição final evitando o objeto colida com os obstáculos, em seguida (figuras 4.19f a 4.19j) o agente aéreo é acionado, indo até a posição atual do objeto, realiza a ação de *GRASP* e o leva até a pose final designada.

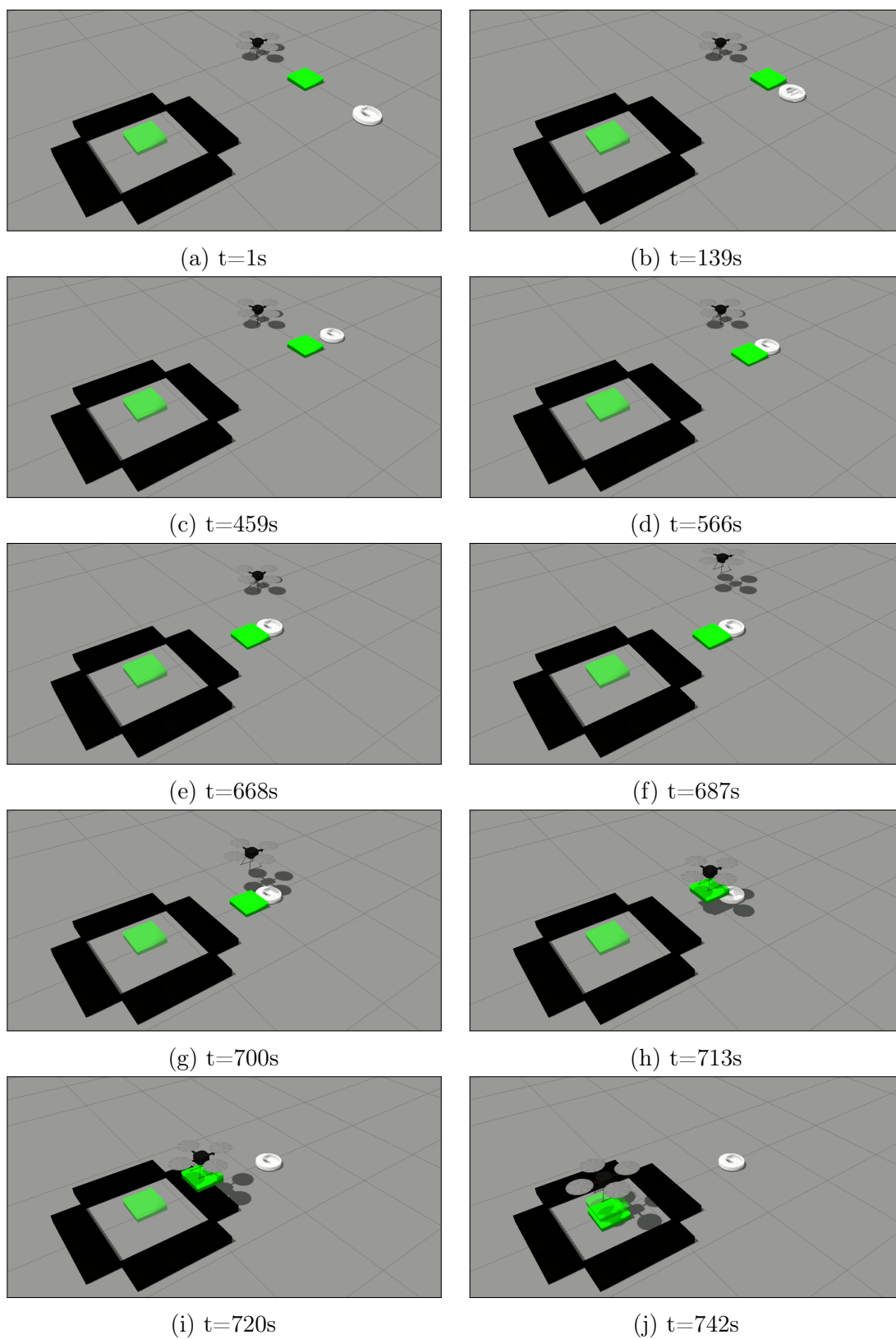


Figura 4.19: Exemplo de transporte utilizando um agente terrestre e um aéreo.

Mediante estes exemplos, é possível ver todas as etapas anteriormente discutidas, havendo o planejamento para o objeto, os agentes realizam seus próprios planejamento e se coordenam de modo a transportar o objeto da melhor maneira possível, respeitando as restrições impostas pelo ambiente.

O controle dos agentes durante a movimentação no ambiente de trabalho foi realizado utilizando um conjunto de algoritmos separados em dois níveis: (i) controlador de navegação, que mantém um registro de todas as posições no ambiente de trabalho que o agente deve percorrer para completar o plano previamente criado, além de gerenciar o envio de novos pontos de destino sempre que detectar a chegada do agente a uma posição previamente enviada; (ii) controlador de velocidades, este recebe um ponto de destino do controlador de navegação, e baseado na posição atual do agente, aplica um controlador PI (proporcional-integrativo) para realizar a movimentação do agente para a posição requerida.

Para ambos os agentes, o algoritmo do controlador de navegação utilizado é o mesmo, pois é necessário somente o repasse dos pontos de destino, porém o controlador de baixo nível (velocidades) é diferenciado, considerando que os agentes possuem maneiras de movimentação distintas. Os Algoritmos 4.1 e 4.2 demonstram sucintamente estes controladores, respectivamente do agente aéreo e terrestre.

Algorithm 4.1 AerialVelocityController(goal_pose, last_cmd, vertical_lim, P, I)

```

1: current_pose ← get_current_pose()
2: current_orientation ← get_orientation(current_pose)
3:
4: diff_x ← goal_pose[x] - current_pose[x]
5: diff_y ← goal_pose[y] - current_pose[y]
6: diff_z ← goal_pose[z] - current_pose[z]
7:
8: angular_vel ← P * current_orientation + I * last_cmd[angle]
9:
10: if abs(diff_z) > vertical_lim then
11:   linear_vel_x ← 0
12:   linear_vel_y ← 0
13: else
14:   linear_vel_x ← P * diff_x + I * last_cmd[x]
15:   linear_vel_y ← P * diff_y + I * last_cmd[y]
16: end if
17: linear_vel_z ← P * diff_z + I * last_output[z]
18: return [linear_vel_x, linear_vel_y, linear_vel_z, angular_vel]

```

O Algoritmo 4.1, demonstra o controle para o agente aéreo. Em um primeiro momento, o mesmo guarda referências para sua posição e orientação atual (referente

Algorithm 4.2 AerialVelocityController(goal_pose, last_cmd, angular_lim, P, I)

```

1: current_pose ← get_current_pose()
2: current_orientation ← get_orientation(current_pose)
3:
4: diff_orientation ← angle_between_poses(current_pose, goal_pose)
5: diff_distance ← distance_between_poses(current_pose, goal_pose)
6:
7: if abs(diff_orientation) >= angular_lim then
8:   linear_vel ← 0
9: else
10:  linear_vel ← P * diff_distance + I * last_cmd[linear]
11: end if
12: angular_vel ← P * diff_orientation + I * last_cmd[angular]
13: return [linear_vel, angular_vel]

```

ao sistema de coordenada do mundo), e calcula a diferença entre sua atual localidade até o ponto de destino. Considerando que não há necessidade de rotação do agente, o mesmo deve manter uma orientação fixa próximo ao valor 0, assim é calculada a velocidade angular necessária para correção de possíveis variações (linha 8). Nas demais linhas, são calculadas as velocidades lineares de movimentação em três dimensões (x, y, z), observando para um detalhe na linha 10, que faz uso da constante *vertical_lim* para delimitar uma distância mínima entre a altitude do agente e a altitude de destino, fazendo com o mesmo se movimente no plano (x, y) somente quando este limiar for atingido. Sem este detalhe de implementação, o agente movimentava-se nos três eixos simultaneamente, ocasionando em colisões desnecessárias com o ambiente.

Para o agente terrestre, foi utilizado o Algoritmo 4.2, que funciona de forma similar ao demonstrado anteriormente, porém de forma mais simples. Baseado na posição e orientação atual, o agente calcula a diferença de ângulo até o ponto de destino, aplicando a função arctan, além da distância entre os mesmos. Análogo ao limiar de altitude, o agente terrestre utiliza a constante *angular_lim* para delimitar uma diferença de ângulo mínimo para que o robô possa se mover. Durante os testes, foi utilizado um valor reduzido para tal constante, ocasionando em uma ausência de curvas por parte do agente, de modo que o mesmo primeiramente realiza um alinhamento em direção à posição de destino e só então realiza a movimentação.

Em ambos os algoritmos, os valores passados através variável *last_cmd*, são todos aqueles calculados durante a última iteração do mesmo, controlado externamente pelo algoritmo de navegação.

4.2.1 Ambiente Real

Os experimentos reais foram realizados utilizando somente agentes terrestres, em um espaço de área aproximada de 2m x 2m situado no laboratório de Visão Computacional e Robótica (VeRLab)⁴. O mesmo controlador anteriormente descrito foi utilizado, com pequenos ajustes nos valores das constantes para atender às variações do ambiente real.

A metodologia apresentada considera que a localização de todos os agentes, objetos e obstáculos no ambiente é conhecida durante a missão de transporte, para tanto, a localização de tais itens foi realizada utilizando marcadores fiduciais que identificam unicamente cada um e são reconhecidos por uma câmera posicionada acima da área de trabalho, sendo capaz de detectar a posição de cada marcador e informar ao sistema estas informações em tempo real.

A plataforma robótica utilizada nos experimentos é do modelo *ePuck*⁵, robô diferencial, com 70mm de diâmetro e 150g, controlado através de comunicação *bluetooth*. Esta robô é capaz de se movimentar, além de possuir alguns sensores, como uma câmera, um conjunto de infra-vermelhos para detecção de obstáculos, além de *encoders* para as rodas. A Figura 4.20 exemplifica o agente utilizado durante o experimentos.

O ambiente de trabalho montado para realização deste experimento consistia na utilização de três robôs *ePucks*, um objeto a ser transportado, e um conjunto arbitrário de obstáculos dispostos no ambiente, como é demonstrado na Figura 4.21. Como apresentado na figura, os agentes foram colocados em uma região à parte do ambiente, passíveis de serem utilizados durante o transporte. O objeto a ser transportado é considerado possuindo peso e resistência desprezíveis, por ser constituído de uma espuma de baixa densidade e apresentar baixa taxa atrito com a superfície na qual o teste está sendo realizado, além de possuir a mesma dimensão (largura e comprimento) que os agentes. Os agentes tem por objetivo levar o objeto de sua posição inicial, assinala em verde, para a posição final, destacada em vermelho.

A Figura 4.22 demonstra a execução da missão de transporte pelos três agentes, considerando que as etapas de planejamento de caminhos do objetos (destacada pelo caminho na cor branca na Figura 4.21, possuindo 3 segmentos) e dos agentes, além da alocação de tarefas já foram realizadas enquanto os agentes permaneciam em suas respectivas posições iniciais. Como pode ser observado, o agente 1 é selecionado para iniciar o transporte, enquanto os demais agentes se movimentam para esperar sua vez. Assim que o agente transporta objeto através do primeiro segmento (S1) na Figura 4.22e o agente 2 torna-se o responsável pelo transporte através do segmento S2, até

⁴<http://www.verlab.dcc.ufmg.br/>

⁵<http://www.e-puck.org/>

que a missão é repassada para o agente 3 como demonstrado na Figura 4.22g, que termina a movimentação do objeto até a posição final passando pelo segmento S3.

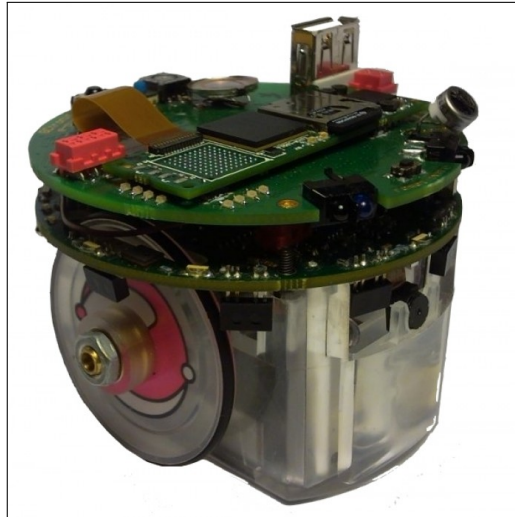


Figura 4.20: Plataforma robótica ePuck utilizada durante os experimentos reais.

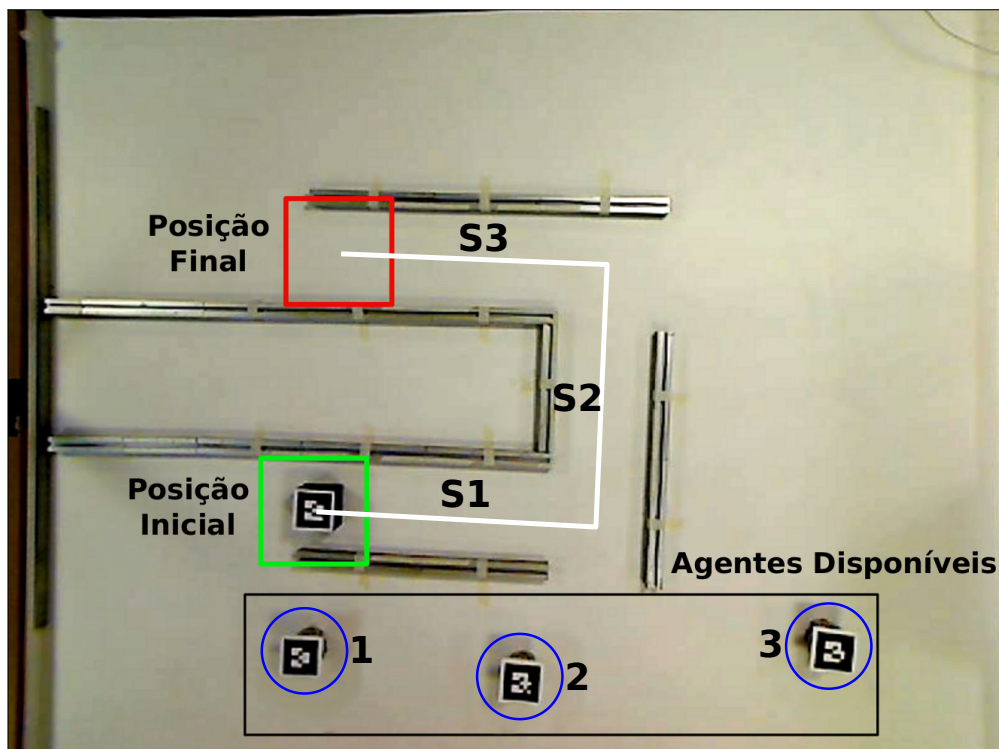


Figura 4.21: Configuração do ambiente de trabalho no experimento real.

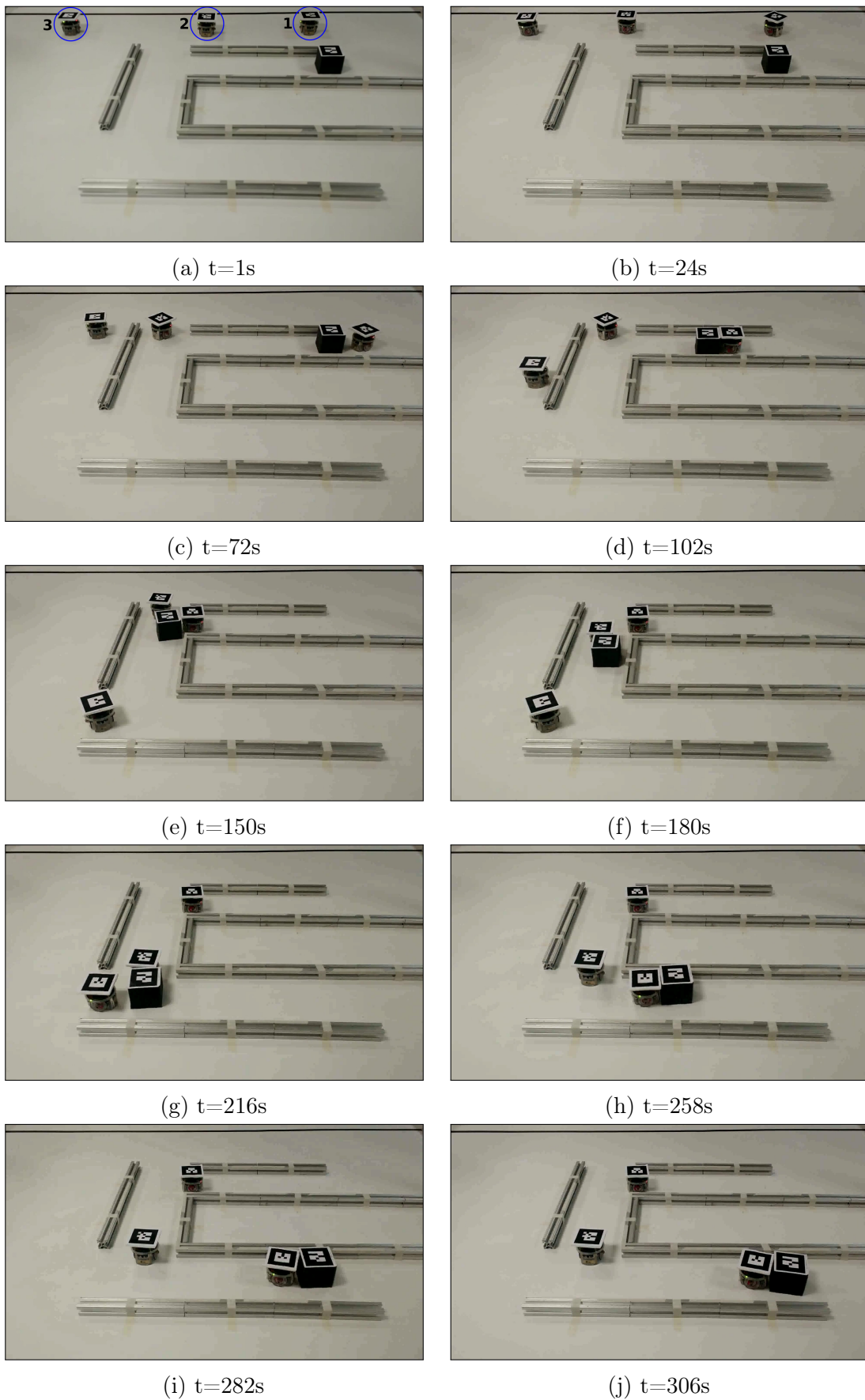


Figura 4.22: Exemplo de transporte de um objeto utilizando três agentes reais.

Capítulo 5

Conclusão e Trabalhos Futuros

5.1 Conclusão

Neste trabalho foi apresentada uma metodologia para o transporte e manipulação de objetos utilizando uma equipe de agentes heterogêneos, demonstrando um conjunto de técnicas capaz de tratar o problema de forma completa, desde o estudo de como movimentar os objetos, até a execução propriamente dita. Este se difere de outros trabalhos na literatura por tratar o planejamento do objeto como ponto de partida para os demais estágios durante a execução da tarefa.

A metodologia trata do problema de transporte mediante o estudo e resolução de dois sub-problemas: (i) descrever uma sequência de posições válidas para os objetos, de modo a garantir sua chegada à posição final, e (ii) como utilizar os recursos robóticos disponíveis para realizar a manipulação dos objetos. Afim de propor uma resolução para estes problemas, a técnica apresentada é dividida em três etapas principais: (i) planejamento de caminhos para os objetos – na qual um plano de movimentação para os objetos a serem transportados é construído considerando os agentes disponíveis, (ii) planejamento e alocação de tarefas – processo no qual a equipe de agentes planeja as trajetórias que devem ser executadas para transportar os objetos, além de entrarem em acordo sobre quais robôs são mais aptos para tais ações, e (iii) coordenação e execução – fase final em que os agentes executam suas respectivas tarefas de forma sincronizada.

Para considerar a heterogeneidade dos agentes utilizados para realizar o transporte dos objetos, os algoritmos de planejamento desenvolvidos neste trabalho fazem uso de uma função heurística que considera certas características destes agentes. Esta função, chamada de função de utilidade, pondera duas dimensões: (i) tempo de travessia – que é o tempo médio de movimentação de um agente, e (ii) energia gasta – tida como a estimativa de energia utilizada pelo agente para se mover. Com a utilização

desta função como heurística, o sistema se torna flexível no sentido de que pode ser configurado para atender certas demandas, como o transporte no menor tempo possível ou poupando o máximo de energia, bastando a troca dos pesos que influenciam cada característica. Como demonstrado nos experimentos realizados, mediante a mudança dos pesos, planos diferentes são criados para atender à configuração criada.

Ao considerar o planejamento do objeto como base estratégica utilizada nas demais etapas da técnica, certas questões foram estudadas durante o processo de desenvolvimento do sistema. Por meio de diversos experimentos, foi demonstrado como a complexidade do ambiente de trabalho tem grande influência nos planos de transporte criados, pois o conjunto de obstáculos molda como o objeto deve se mover, de modo a evadi-los. Outro ponto de influência para os planos é a disponibilidade de agentes para realização dos mesmos, no qual a ausência de certos tipos ou mesmo a quantidade de agentes presentes tornavam o processo mais rápido e/ou com menor custo total. O estudo realizado considerando a penalização ou não da mudança de direção durante o transporte, demonstra que, apesar do aumento na distância percorrida pelo objeto, esta configuração proporciona um menor tempo e deslocamento por parte do agentes, provando ser artifício interessante de ser utilizado.

O planejamento de caminhos dos agentes e posterior alocação de tarefas foram tratadas como etapas conjuntas, pois todo plano criado por um agente descreve como o mesmo pretende ajudar na performance total da missão, e se a escolha destes planos não for executada de forma precisa, todo o sistema terá seu rendimento prejudicado. O algoritmo de alocação de tarefas desenvolvido funciona de forma descentralizada, fazendo uso de procedimentos de sincronização entre os agentes para atingir o consenso em relação às tarefas a serem executadas. Internamente, é utilizado o Algoritmo Húngaro para alocação das tarefas entre os agentes, sendo adaptado para o uso em um sistema distribuído.

A etapa de execução e coordenação dos agentes para o transporte foi idealizada mediante a influência de algoritmos de redes de computadores, onde os robôs fazem uso dos chamados *tokens* para assinalar a sua permissão de transporte de um determinado objeto. Mediante o uso destas estratégias, foi possível a sincronização entre a equipe, onde cada membro só executaria sua tarefa se possuísse a respectiva autorização para o mesmo. Além desta estratégia, foi desenvolvido um tipo de acordo entre agentes que tivessem planos de transporte cruzados, no qual o agente que possuir um menor plano de travessia, possui também menor prioridade durante o processo.

Diversos desafios foram encontrados durante o desenvolvimento da técnica aqui apresentada, envolvendo principalmente o planejamento de caminhos dos objetos, pois as decisões aplicadas neste módulo regem todas as demais etapas. Questões como

a definição das estratégias de desvio de obstáculos ou mesmo a aplicação de algoritmos de alocação de tarefas foram pontos que demandaram grande estudo para serem desempenhadas.

5.2 Trabalhos Futuros

Este trabalho não é em sua totalidade um trabalho fechado, sendo passível de melhorias e refinamentos. Uma importante adição ao sistema seria um módulo de exploração e mapeamento do ambiente de trabalho, proporcionando ao método um meio automático de descobrimento dos objetos a serem transportados, retirando a necessidade da criação manual do mapa para cada caso de estudo. Este módulo poderia ser implementado utilizando por exemplo um robô aéreo, que fazendo uso de uma câmera apontada para baixo, seria capaz de ter uma visão global do mapa e assim auxiliar os demais agentes terrestres durante a tarefa de transporte, promovendo uma nova forma de coordenação entre a equipe.

Outra suposição que poderia ser desconsiderada está no fato da localização dos agentes e objetos ser conhecida durante todo o processo, não havendo qualquer tipo de incerteza neste parâmetro. Ao considerar que o sistema poderia ser utilizado em ambiente reais e de grandes dimensões, este pressuposto não condiz com tal realidade. Um passo neste sentido seria a complementação dos agentes com sensores capazes de detectar os objetos em conjunto com algoritmos de localização própria, de modo que cada agente, uma vez que encontrasse um determinado objeto, informasse aos demais sua localização e a estimativa de posição do item a ser transportado. Similar ao caso do mapeamento, um agente aéreo poderia contribuir nesta etapa, ficando à parte do sistema de transporte, informando ao time suas respectivas localizações.

A colaboração entre os agentes também deve ser melhor explorada, por exemplo, no caso do transporte de objetos que demandam mais de um agente para a sua realização. Dessa forma, a etapa de alocação de tarefas deve ser modificada para associar possíveis segmentos a mais de um robô simultaneamente. Uma possível colaboração entre agentes robóticos e humanos também pode ser foco de futura investigação.

Além destas melhorias, o sistema apresentado também não é capaz de lidar com a falha de um determinado agente durante a execução da tarefa, o que torna a tarefa de transporte impossibilitada de ser completada. A adição de métodos para tratamento deste tipo de situação seria um meio de tornar o método ainda mais robusto, considerando que, uma vez que um determinado agente pare de assinalar o seu pleno funcionamento, os demais integrantes da equipe deveriam recalculer seus planos de

modo a realocar as tarefas e por fim concluir a missão de transporte fazendo uso dos robôs restantes.

Referências Bibliográficas

- Adorno, B. V. (2011). *Two-arm manipulation: From manipulators to enhanced human-robot collaboration*. Tese de doutorado, Université Montpellier II-Sciences et Techniques du Languedoc.
- Adorno, B. V.; Fraisse, P. & Druon, S. (2010). Dual position control strategies using the cooperative dual task-space framework. Em *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 3955--3960. IEEE.
- Ahmadabadi, M. & Nakano, E. (2001). A "constrain and move" approach to distributed object manipulation. *IEEE Transactions on Robotics and Automation*, 17(2):157--172.
- Arleo, G.; Caccavale, F.; Muscio, G. & Pierri, F. (2013). Control of quadrotor aerial vehicles equipped with a robotic arm. Em *Control & Automation (MED), 2013 21st Mediterranean Conference on*, pp. 1174--1180. IEEE.
- Augugliaro, F.; Lupashin, S.; Hamer, M.; Male, C.; Hehn, M.; Mueller, M. W.; Willmann, J. S.; Gramazio, F.; Kohler, M. & D'Andrea, R. (2014). The flight assembled architecture installation: Cooperative construction with flying machines. *Control Systems, IEEE*, 34(4):46--64.
- Baizid, K.; Caccavale, F.; Chiaverini, S.; Giglio, G. & Pierri, F. (2014). Safety in coordinated control of multiple unmanned aerial vehicle manipulator systems: Case of obstacle avoidance. Em *Control and Automation (MED), 2014 22nd Mediterranean Conference of*, pp. 1299--1304. IEEE.
- Behrens, M.; Huang, S. & Dissanayake, G. (2010). Models for pushing objects with a mobile robot using single point contact. pp. 2964--2969.
- Bernard, M.; Kondak, K.; Maza, I. & Ollero, A. (2011). Autonomous transportation and deployment with aerial robots for search and rescue missions. *Journal of Field Robotics*, 28(6):914--931.

- Brooks, R. A. (1986). A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14--23.
- Carvalho, F. F.; Cavalcante, R. C.; Vieira, M. A.; Chaimowicz, L. & Campos, M. F. (2013). A multi-robot exploration approach based on distributed graph coloring. Em *Robotics Symposium and Competition (LARS/LARC), 2013 Latin American*, pp. 142--147. IEEE.
- Casper, J. & Murphy, R. (2003). Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 33(3):367--385.
- da Costa, E. D. S. (2012). Exploração e transporte cooperativo de objetos em ambientes desconhecidos. Mestrado, ICEx - Instituto de Ciências Exatas, UFMG - Universidade Federal de Minas Gerais.
- dos Santos, B. S. R.; Givigi, S. N. & Nascimento, C. L. (2013). Autonomous construction of structures in a dynamic environment using reinforcement learning. Em *Systems Conference (SysCon), 2013 IEEE International*, pp. 452--459. IEEE.
- dos Santos, B. S. R.; Givigi, S. N. & Nascimento, C. L. (2014). Autonomous construction of multiple structures using learning automata: Description and experimental validation.
- Emery, R. & Balch, T. (2001). Behavior-based control of a non-holonomic robot in pushing tasks. Em *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pp. 2381 – 2388 vol.3. ISSN 1050-4729.
- Eoh, G.; Jeon, J. D.; Choi, J. S. & Lee, B. H. (2011). Multi-robot cooperative formation for overweight object transportation. Em *System Integration (SII), 2011 IEEE/SICE International Symposium on*, pp. 726--731. IEEE.
- Farinelli, A.; Iocchi, L. & Nardi, D. (2004). Multirobot systems: a classification focused on coordination. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(5):2015--2028.
- Fink, J.; Ani Hsieh, M. & Kumar, V. (2008). Multi-robot manipulation via caging in environments with obstacles. pp. 1471--1476.
- Gerkey, B. & Mataric, M. (2002). Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination. volume 1, pp. 464--469.

- Gerkey, B. P. & Mataric, M. J. (2001). Principled communication for dynamic multi-robot task allocation. Em *Experimental Robotics VII*, pp. 353--362. Springer.
- Ghosh, A.; Konar, A. & Janarthanan, R. (2012). Multi-robot cooperative box-pushing problem using multi-objective particle swarm optimization technique. Em *Information and Communication Technologies (WICT), 2012 World Congress on*, pp. 272--277. IEEE.
- Goldsmith, S. Y. & Robinett III, R. (1998). Collective search by mobile robots using alpha-beta coordination. Em *Collective robotics*, pp. 135--146. Springer.
- Grocholsky, B.; Keller, J.; Kumar, V. & Pappas, G. (2006). Cooperative air and ground surveillance. *IEEE Robotics and Automation Magazine*, 13(3):16--26.
- Guizzo, E. (2008). Three engineers, hundreds of robots, one warehouse. *IEEE spectrum*, 7(45):26--34.
- Hart, P. E.; Nilsson, N. J. & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100--107.
- Hekmatfar, T.; Masehian, E. & Mousavi, S. J. (2014). Cooperative object transportation by multiple mobile manipulators through a hierarchical planning architecture. Em *Robotics and Mechatronics (ICRoM), 2014 Second RSI/ISM International Conference on*, pp. 503--508. IEEE.
- Inoue, R.; Fujii, N.; Takano, R. & Ota, J. (2008). Rearrangement of multiple objects by a robot group having a multi-task function. pp. 2013--2018.
- Inoue, R.; Fujii, N.; Takano, R. & Ota, J. (2011). Realization of a multiple object rearrangement task with two multi-task functional robots. *Advanced Robotics*, 25(11):1365--1383.
- Jiang, Q. & Kumar, V. (2013). The inverse kinematics of cooperative transport with multiple aerial robots. *Robotics, IEEE Transactions on*, 29(1):136--145.
- Kim, S.; Choi, S. & Kim, H. (2013). Aerial manipulation using a quadrotor with a two dof robotic arm. pp. 4990--4995.
- Kube, C. R. & Bonabeau, E. (2000). Cooperative transport by ants and robots. *Robotics and autonomous systems*, 30(1):85--101.

- Kulatunga, A.; Liu, D.; Dissanayake, G. & Siyambalapitiya, S. (2006). Ant colony optimization based simultaneous task allocation and path planning of autonomous vehicles. Em *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, pp. 1--6. IEEE. task allocation.
- LaValle, S. M. (2006). *Planning algorithms*. Cambridge university press.
- Lehman, A.; Berg, K.; Dumpert, J.; Wood, N.; Visty, A.; Rentschler, M.; Platt, S.; Farritor, S. & Oleynikov, D. (2008). Surgery with cooperative robots. *Computer Aided Surgery*, 13(2):95--105.
- Lindsey, Q.; Mellinger, D. & Kumar, V. (2011). Construction of cubic structures with quadrotor teams. *Proc. Robotics: Science & Systems VII*.
- Lindsey, Q.; Mellinger, D. & Kumar, V. (2012). Construction with quadrotor teams. *Autonomous Robots*, 33(3):323--336.
- Lynch, K. M. (1996). *Nonprehensile robotic manipulation: controllability and planning*. Carnegie Mellon University.
- Machado, T.; Malheiro, T.; Monteiro, S.; Bicho, E. & Erhagen, W. (2013). Transportation of long objects in unknown cluttered environments by a team of robots: A dynamical systems approach. Em *Industrial Electronics (ISIE), 2013 IEEE International Symposium on*, pp. 1--6. IEEE.
- Magariyama, S.; Matsumoto, H.; Ruiz, L. & Wang, Z. (2014). Estimation of a feasible dynamic caging zone for multirobot object transportation. Em *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on*, pp. 508--514. IEEE.
- Maghsoud, P.; De Silva, C. W. & Khan, M. T. (2014). Autonomous and cooperative multirobot system for multi-object transportation. Em *Computer Science & Education (ICCSE), 2014 9th International Conference on*, pp. 211--217. IEEE.
- Matarić, M. J.; Nilsson, M. & Simsarin, K. T. (1995). Cooperative multi-robot box-pushing. Em *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pp. 556--561. IEEE.
- Maza, I.; Kondak, K.; Bernard, M. & Ollero, A. (2010). Multi-uav cooperation and control for load transportation and deployment. *Journal of Intelligent and Robotic Systems*, 57(1-4):417--449.

- Melo, R. S.; Macharet, D. G. & Campos, M. F. M. (2015). Multi-object transportation using a mobile robot. *LARS 2015 12th Latin American Robotics Symposium*.
- Meriçli, T.; Veloso, M. & Akin, H. (2013). Achievable push-manipulation for complex passive mobile objects using past experience. volume 1, pp. 71–78.
- Michael, N.; Fink, J. & Kumar, V. (2011). Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86.
- Miyata, N.; Ota, J.; Arai, T. & Asama, H. (2002). Cooperative transport by multiple mobile robots in unknown static environments associated with real-time task assignment. *IEEE Transactions on Robotics and Automation*, 18(5):769–780.
- Montemayor, G. & Wen, J. (2005). Decentralized collaborative load transport by multiple robots. volume 2005, pp. 372–377.
- Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.
- Murphy, R. (2004). Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 34(2):138–153.
- Murphy, R.; Kravitz, J.; Stover, S. & Shoureshi, R. (2009). Mobile robots in mine rescue and recovery. *IEEE Robotics and Automation Magazine*, 16(2):91–103.
- Murray, R. M.; Li, Z.; Sastry, S. S. & Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*. CRC press.
- Nagai, S.; Kaneshige, A. & Ueki, S. (2011). Three-dimensional obstacle avoidance on-line path-planning method for autonomous mobile overhead crane. Em *Mechatronics and Automation (ICMA), 2011 International Conference on*, pp. 1497–1502. IEEE.
- Neumann, M. A.; Chin, M. H. & Kitts, C. A. (2014). Object manipulation through explicit force control using cooperative mobile multi-robot systems. Em *Proceedings of the World Congress on Engineering and Computer Science*, volume 1.
- Orsag, M.; Korpela, C. & Oh, P. (2013). Modeling and control of mm-uav: Mobile manipulating unmanned aerial vehicle. *Journal of Intelligent & Robotic Systems*, 69(1-4):227–240.
- Parra-González, E. F.; Ramírez-Torres, G. & Toscano-Pulido, G. (2008). Motion planning for cooperative multi-robot box-pushing problem. Em *Advances in Artificial Intelligence—IBERAMIA 2008*, pp. 382–391. Springer.

- Parra-Gonzalez, E. F. & Ramirez-Torres, J. G. (2011). Object path planner for the box pushing problem. *Multi-Robot Systems, Trends and Development*, T. Yasuda, Editor, pp. 291--306.
- Parra-Gonzalez, E. F. & Ramirez-Torres, J. G. (2012). Grasp quality for the object transportation by communities of mobile robots. Em *Electrical Engineering, Computing Science and Automatic Control (CCE), 2012 9th International Conference on*, pp. 1--6. IEEE.
- Pereira, G.; Campos, M. & Kumar, V. (2004). Decentralized algorithms for multi-robot manipulation via caging. *International Journal of Robotics Research*, 23(7-8):783--795.
- Pounds, P. E.; Bersak, D. R. & Dollar, A. M. (2011). Grasping from the air: Hovering capture and load stability. Em *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 2491--2498. IEEE.
- Reif, J. H. (1979). Complexity of the mover's problem and generalizations extended abstract. Em *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*, pp. 421--427.
- Shiroma, P. & Campos, M. (2009). Comutar: A framework for multi-robot coordination and task allocation. pp. 4817--4824.
- Siegwart, R.; Nourbakhsh, I. R. & Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*. MIT Press, Cambridge, MA, USA, 2^a edição. ISBN 0262015358.
- Song, P. & Kumar, V. (2002). A potential field based approach to multi-robot manipulation. Em *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pp. 1217--1222. IEEE.
- Sugar, T. & Kumar, V. (2002). Control of cooperating mobile manipulators. *IEEE Transactions on Robotics and Automation*, 18(1):94--103.
- Sujit, P. & Saripalli, S. (2013). An empirical evaluation of co-ordination strategies for an auv and uav. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 70(1-4):373--384.
- Tanner, H. (2007). Switched uav-ugv cooperation scheme for target detection. pp. 3457--3462.

- Tanner, H.; Loizou, S. & Kyriakopoulos, K. (2003). Nonholonomic navigation and control of cooperating mobile manipulators. *IEEE Transactions on Robotics and Automation*, 19(1):53–64.
- Thomas, J.; Loianno, G.; Polin, J.; Sreenath, K. & Kumar, V. (2014). Toward autonomous avian-inspired grasping for micro aerial vehicles. *Bioinspiration & biomimetics*, 9(2):025010.
- Tiganas, V.; Kloetzer, M. & Burlacu, A. (2013). Multi-robot based implementation for a sample gathering problem. Em *System Theory, Control and Computing (ICSTCC), 2013 17th International Conference*, pp. 545–550. IEEE.
- Tokekar, P.; Vander Hook, J.; Mulla, D. & Isler, V. (2013). Sensor planning for a symbiotic uav and ugv system for precision agriculture. Em *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 5321–5326. ISSN 2153-0858.
- Wada, M. & Torii, R. (2013). Cooperative transportation of a single object by omnidirectional robots using potential method. Em *Advanced Robotics (ICAR), 2013 16th International Conference on*, pp. 1–6. IEEE.
- Wang, J.; Wang, T.; Wang, X. & Meng, X. (2009). Multi-robot decision making based on coordination graphs. Em *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*, pp. 2393–2398. IEEE.
- Wang, Z. & Kumar, V. (2002). Object closure and manipulation by multiple cooperating mobile robots. Em *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pp. 394–399. IEEE.
- Wen, J. T. & Kreutz-Delgado, K. (1992). Motion and force control of multiple robotic manipulators. *Automatica*, 28(4):729–743.
- Willmann, J.; Augugliaro, F.; Cadalbert, T.; D'Andrea, R.; Gramazio, F. & Kohler, M. (2012). Aerial robotic construction towards a new field of architectural research. *International journal of architectural computing*, 10(3):439–460.
- Yamashita, A.; Arai, T.; Ota, J. & Asama, H. (2003). Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Transactions on Robotics and Automation*, 19(2):223–237.