



Um estudo sobre algoritmos de coordenação de uma frota flexível de veículos  
móveis autónomos em ambientes de logística interna

Ana Filipa Antunes Lage

UMinho | 2023



Universidade do Minho  
Escola de Engenharia

Ana Filipa Antunes Lage

Um estudo sobre algoritmos de coordenação de uma  
frota flexível de veículos móveis autónomos em  
ambientes de logística interna

janeiro de 2023





Universidade do Minho  
Escola de Engenharia

Ana Filipa Antunes Lage

Um estudo sobre algoritmos de coordenação de uma frota flexível de veículos móveis autónomos em ambientes de logística interna

Dissertação de Mestrado  
Mestrado Integrado em Engenharia Eletrónica  
Industrial e Computadores  
Controlo, Automação e Robótica

Trabalho efetuado sob a orientação do  
Professor Doutor Sérgio Paulo Carvalho Monteiro

## DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

### *Licença concedida aos utilizadores deste trabalho*



Atribuição  
CC BY

<https://creativecommons.org/licenses/by/4.0/>

Guimarães, 30 de janeiro de 2023

---

(Ana Filipa Antunes Lage)

## AGRADECIMENTOS

Terminada uma fase importante da minha formação acadêmica, cabe-me deixar umas palavras de agradecimento. Este é o momento de expressar a minha gratidão a todos aqueles que, de uma forma ou de outra, me ajudaram nesta longa jornada.

Quero expressar a minha mais profunda gratidão ao meu orientador, professor doutor Sérgio Monteiro, pela sua orientação e apoio durante todo o processo de elaboração da minha dissertação. Agradeço-lhe também pela sua paciência e disponibilidade, sempre disposto a esclarecer as minhas dúvidas e a oferecer alternativas face às adversidades encontradas. Por fim, agradeço-lhe também pelo seu apoio constante e incentivo durante todo o processo, sem o qual eu não teria conseguido completar esta jornada.

À equipa do MarLab, pela sua colaboração e disponibilidade imediata em ajudar sempre que era preciso.

Aos meus pais e ao meu irmão, pela paciência e compreensão para comigo quando os problemas foram surgindo. Desde o início dos meus estudos, sempre acreditaram em mim e me apoiaram incondicionalmente.

Aos meus amigos, pela compreensão e apoio quando precisei de me afastar para me dedicar à dissertação e pelos “cafezinhos” fundamentais para manter o equilíbrio emocional durante todo este processo.

E por último, mas não menos importante, ao meu namorado, por ser o meu confidente, por me ouvir quando eu precisava de desabafar e por me fazer rir quando eu estava desanimada. Agradeço pelo apoio e incentivo que me deu, por ser uma fonte constante de inspiração e suporte, pela motivação que me dava quando as coisas corriam menos bem e por nunca me deixar desistir.

Esta conquista não seria possível sem todos vocês.

Eternamente grata,

Ana Filipa Lage

## DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração. Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Guimarães, 30 de janeiro de 2023

---

(Ana Filipa Antunes Lage)

## RESUMO

Este projeto de dissertação aborda o problema de coordenação de uma frota de veículos autônomos em ambientes de logística interna, sendo que o ambiente é conhecido à priori com obstáculos estáticos e dinâmicos detetados pelas leituras dos sensores dos robôs. Para que haja coordenação é necessário garantir a existência dos seguintes aspectos: comunicação, atribuição de tarefas, prevenção de colisões, planeamento de caminhos, gestão de frotas, protocolos de segurança e flexibilidade.

Foi implementada uma abordagem centralizada, isto é, os robôs constituintes da frota comunicam com um componente responsável pela tomada de decisões. Cada robô é responsável por verificar a existência ou ausência de colisões na sua rota. Esta verificação tem em consideração apenas um pequeno traçado da rota, ao invés das poses todas até ao ponto de destino. Quando detetado algum conflito, tal é comunicado ao servidor central, sendo este responsável pela tomada de decisões da frota, mandando o robô avançar ou parar consoante o tipo de conflito detetado e a sua prioridade em relação aos restantes robôs da frota. O sistema foi implementado num ambiente dinâmico.

Para o planeamento de caminhos entre dois pontos, e após uma análise a alguns planeadores, optou-se pelo algoritmo de busca Híbrido A\*. Utilizando este algoritmo de busca foi possível encontrar, de forma rápida, caminhos em ambientes de elevada dimensão e complexidade. Sendo também possível a sua utilização em ambientes com múltiplos robôs, permitindo que cada robô planeie a sua rota de forma independente, evitando, simultaneamente, colisões com os restantes robôs. Para além disso, este algoritmo também se mostrou capaz de lidar com ambientes em mudança e incorporar novas informações em tempo real.

**Palavras-chave:** coordenação, planeamento de caminhos, multi-robô, conflitos, prioridade

## ABSTRACT

This dissertation project addresses the problem of coordinating a fleet of autonomous vehicles in intralogistics environments, where the environment is known a priori with static and dynamic obstacles detected by the robot sensor readings. To achieve coordination is required to accomplish the following aspects: communication, task assignment, collision avoidance, path planning, fleet management, safety protocols, and flexibility.

A centralized approach was implemented, that is, the robots constituting the fleet communicate with a component responsible for making decisions. Each robot was responsible for checking for the existence or absence of collisions on its route. This check considers only a short stretch of the route, rather than all the poses up to the destination point. When a conflict is detected, it is communicated to the central server, which is responsible for making fleet decisions, ordering the robot to move forward or stop depending on the type of conflict detected and its priority in relation to the other robots in the fleet. The system was implemented in a dynamic environment.

Regarding the path planning between two points, and after a brief analysis of some planners, the A\* Hybrid search algorithm was chosen. Using this search algorithm, it was possible to quickly find paths in high-dimensional and complex environments. It is also possible to use it in multi-robot environments, allowing each robot to plan its route independently while avoiding collisions with other robots. In addition, this algorithm has also proven capable of handling changing environments and incorporating new information in real time.

**Keywords:** coordination, path planning, multi-robot, conflicts, priority

## ÍNDICE

Agradecimentos.....	iii
Declaração de Integridade .....	iv
Resumo.....	v
Abstract.....	vi
Índice.....	vii
Índice de Figuras.....	xi
Índice de Tabelas .....	xvi
Lista de Abreviações, Siglas e Acrónimos .....	xvii
1. Introdução .....	1
1.1. Enquadramento e Motivação .....	1
1.2. Descrição do Problema .....	1
1.3. Objetivos da Dissertação .....	2
1.4. Estrutura da Dissertação .....	3
2. Estado da Arte .....	5
2.1. Sistemas Multi-robô.....	5
Coordenação.....	6
Composição.....	6
Ambiente Cooperativo e Competitivo.....	7
Com Comunicação e Sem Comunicação .....	7
Reativo e Deliberativo .....	8
Tamanho da Equipa .....	8
2.2. Classificação da Coordenação Multi-Robô .....	8
Coordenação Estática e Dinâmica.....	9
Coordenação Implícita e Explícita .....	10

Coordenação Fraca e Forte .....	10
Coordenação Centralizada.....	11
Coordenação Descentralizada.....	12
Coordenação Híbrida.....	12
2.3. Planeamento Coordenado de Tarefas .....	13
2.4. Planeamento Coordenado de Movimentos .....	19
2.5. <i>Robot Mobile Fulfillment System</i> .....	26
2.6. <i>Áreas de Aplicação</i> .....	27
Vigilância, Busca e Salvamento .....	28
Forrageamento e Pastoreio.....	28
<i>Formação e Exploração</i> .....	29
<i>Manipulação Cooperativa</i> .....	29
<i>Equipa heterogénea</i> .....	29
<i>Ambiente Adversarial</i> .....	30
3. Fundamentos Teóricos .....	31
3.1. Representação do Ambiente .....	31
3.2. Métodos clássicos .....	31
3.2.1. Método <i>Roadmap</i> .....	31
3.2.2. Método de Decomposição de Células.....	33
3.2.3. Método de Campos Potenciais .....	35
3.3. Algoritmos de Busca – Métodos Exatos.....	35
Algoritmo Dijkstra.....	35
Algoritmo Dijkstra Melhorado.....	36
3.4. Algoritmos de Busca – Métodos Heurísticos.....	36
Algoritmo A*.....	36

Algoritmo A* Melhorado.....	38
Híbrido A* .....	38
Algoritmo A* Melhorado no Tempo .....	40
4. Ferramentas .....	42
4.1. Ferramenta de Simulação .....	42
<i>Navigation Toolbox</i> .....	42
<i>Robotics System Toolbox</i> .....	43
4.2. Cenário de Simulação .....	43
5. Proposta de Solução .....	45
5.1. Análise.....	45
5.2. Arquitetura do sistema .....	52
5.3. Planeador global .....	55
5.4. Resolução de Conflitos .....	56
6. Testes e Resultados .....	60
6.1. Planeamento de caminho apenas para um único robô – cenário 2.....	60
6.2. Planeamento de caminhos para múltiplos robôs – cenário 2.....	61
Frota de dois robôs .....	62
Frota de cinco robôs.....	64
Frota de dez robôs .....	66
Robôs de Patrulha.....	70
6.3. Planeamento de caminhos para múltiplos robôs – cenário 3.....	72
Frota de quatro robôs.....	73
Frota de dez robôs .....	75
Frota de dezasseis robôs.....	77
6.4. Discussão de Resultados .....	79

7. Conclusões e Trabalho Futuro .....	82
7.1. Conclusões .....	82
7.2. Trabalho Futuro .....	83
Referências Bibliográficas .....	84
Apêndices. ....	95
A1. Análise dos Planeadores - Cenário 1 .....	95
A2. Análise dos Planeadores - Cenário 2 .....	98

## ÍNDICE DE FIGURAS

Figura 2-1 - Classificação do MRS. ....	6
Figura 2-2 - Classificação ao nível da coordenação do MRS. ....	9
Figura 2-3 - Esquemático da coordenação centralizada.....	11
Figura 2-4 - Esquemático Coordenação Descentralizada Distribuída.....	12
Figura 2-5 - Trajetórias do robô com leilões de item único. ....	15
Figura 2-6 - Trajetórias do robô com leilões combinatórios.....	15
Figura 2-7 - Realocação dinâmica de tarefas quando é introduzida uma nova tarefa.....	16
Figura 2-8 - Reconfiguração dinâmica quando entra um novo AGV na frota. ....	17
Figura 2-9 - Trajetórias sintetizadas para 25 robôs num espaço de trabalho compacto.....	19
Figura 2-10 - Trajetórias sintetizadas para 50 robôs num espaço de trabalho sem obstáculos.....	19
Figura 2-11 – Exemplo de CD para dois veículos. ....	20
Figura 2-12 - Representação de uma interseção num mapa de rotas. ....	21
Figura 2-13 - Tempo total versus número de AGV para a Estratégia de Negociação e para a Estratégia Otimizada.....	22
Figura 2-14 - Ambiente real com uma frota de quatro robôs e navegação coordenada num espaço pequeno e denso.....	23
Figura 2-15 - Comparação das trajetórias dos agentes. ....	25
Figura 2-16 - Cenário com vários obstáculos. ....	25
Figura 2-17 - Processo central de um RMFS.....	26
Figura 2-18 - Processo de decisão de um sistema RMFS de alta densidade. ....	27
Figura 3-1 - Espaço de configuração poligonal com ponto de partida e ponto de destino representados. ....	32
Figura 3-2 - Caminho mais curto representado a tracejado. ....	32
Figura 3-3 - Representação do caminho a negrito utilizando o Diagrama de Voronoi.....	33

Figura 3-4 - Método exato de decomposição por células. ....	34
Figura 3-5 - Método aproximado de decomposição por células. ....	34
Figura 3-6 - Decomposição por células usando quadrees. ....	34
Figura 3-7 - Algoritmo A* .....	40
Figura 3-8 - Algoritmo Híbrido A* .....	40
Figura 4-1 - Mapa do chão de fábrica 1 .....	44
Figura 4-2 - Mapa do chão de fábrica 2.....	44
Figura 4-3 - Cenário com um robô e a respetiva rota definida .....	44
Figura 5-1 - Mapa do cenário 1 com três pares de posições iniciais e finais definidas.....	46
Figura 5-2 - Caminho gerado para todos os planeadores na primeira execução – Cenário 1 .....	47
Figura 5-3 - Caminho gerado para todos os planeadores na décima execução – Cenário 1 .....	47
Figura 5-4 - Caminho gerado para todos os planeadores na vigésima execução – Cenário 1 .....	47
Figura 5-5 - Representação da médio e desvio padrão da métrica distância aos obstáculos – Cenário 1 .....	48
Figura 5-6 - Representação da validade do caminho gerado – Cenário 1.....	48
Figura 5-7 - Representação da médio e desvio padrão da métrica tempo de execução – Cenário 1 ....	48
Figura 5-8 - Representação da média e desvio padrão da métrica tempo de inicialização – Cenário 1	48
Figura 5-9 - Representação da média e desvio padrão da métrica comprimento do caminho – Cenário 1 .....	49
Figura 5-10 - Representação da média e desvio padrão da métrica suavidade do caminho – Cenário 1 .....	49
Figura 5-11 – Mapa do cenário 2 com três pares de posições e orientações iniciais e finais definidas.	49
Figura 5-12 – Representação da validade do caminho gerado – Cenário 2 .....	50
Figura 5-13 – Representação da média e desvio padrão da métrica comprimento do caminho – Cenário 2.....	51

Figura 5-14 – Representação da média e desvio padrão da métrica comprimento do caminho para o par de posições cujo caminho é maior – Cenário 2 .....	51
Figura 5-15 – Caminho gerado para todos os planeadores na primeira execução – Cenário 2.....	52
Figura 5-16 – Caminho gerado para todos os planeadores na décima execução – Cenário 2 .....	52
Figura 5-17 – Caminho gerado para todos os planeadores na vigésima execução – Cenário 2 .....	52
Figura 5-18 - Esquema de arquitetura centralizada.....	53
Figura 5-19 – Cenário 2: Posições de partida ou carregamento de baterias (charging) de cada robô, e posições de descarregamento de carga (unloading) ou estações de trabalho para cada robô .....	54
Figura 5-20 – Cenário 3: Posições de partida ou carregamento de baterias (charging) de cada robô, e posições de descarregamento de carga (unloading) ou estações de trabalho para cada robô .....	55
Figura 5-21 - Exemplo de uma rota calculado pelo planeador Híbrido A* (imagem à esquerda) e da respetiva expansão analítica dos segmentos da zona a vermelho (imagem à direita).....	56
Figura 5-22 - Mapa do cenário 2 com as áreas das estações de trabalho com maior prioridade representadas a vermelho, e com menor prioridade representadas a verde .....	57
Figura 5-23 - Mapa do cenário 3 com as áreas das estações de trabalho com maior prioridade representadas a vermelho, e com menor prioridade representadas a verde .....	58
Figura 6-1 - Planeamento inicial, utilizando o algoritmo Híbrido A*, entre a posição inicial e a posição destino inicialmente definidas .....	60
Figura 6-2 - Mapa fornecido com adição de um obstáculo .....	60
Figura 6-3 - Rota final do robô após vários replaneamentos devido ao obstáculo introduzido e à inflação do mapa.....	61
Figura 6-4 - Planeamento Inicial para o robô 1 .....	63
Figura 6-5 - Planeamento Inicial para o robô 2 .....	63
Figura 6-6 - Caminho percorrido pelos robôs até às respetivas posições de destino.....	63
Figura 6-7 - Planeamento Inicial para o robô 1 .....	64
Figura 6-8 - Planeamento Inicial para o robô 2 .....	64
Figura 6-9 - Planeamento Inicial para o robô 3 .....	64

Figura 6-10 - Planeamento Inicial para o robô 4 .....	64
Figura 6-11 - Planeamento Inicial para o robô 5 .....	64
Figura 6-12 - Caminho percorrido pelos cinco robôs até às respetivas posições de destino .....	65
Figura 6-13 - Planeamento Inicial para o robô 1 .....	66
Figura 6-14 - Planeamento Inicial para o robô 2 .....	66
Figura 6-15 - Planeamento Inicial para o robô 3 .....	66
Figura 6-16 - Planeamento Inicial para o robô 4 .....	66
Figura 6-17 - Planeamento Inicial para o robô 5 .....	66
Figura 6-18 - Planeamento Inicial para o robô 6 .....	66
Figura 6-19 - Planeamento Inicial para o robô 7 .....	66
Figura 6-20 - Planeamento Inicial para o robô 8 .....	66
Figura 6-21 - Planeamento Inicial para o robô 9 .....	67
Figura 6-22 - Planeamento Inicial para o robô 10 .....	67
Figura 6-23 - Caminho percorrido pelos dez robôs até às respetivas posições de destino .....	67
Figura 6-24 - Cenário 2 com corredores ocupados .....	68
Figura 6-25 - Caminho percorrido pelos dez robôs até às respetivas posições de destino, para o novo cenário.....	69
Figura 6-26 - Posições Iniciais dos robôs de patrulha e Pontos Intermédios que definem a rota de cada robô .....	70
Figura 6-27 - Caminho percorrido pelos dez robôs até às respetivas posições de destino, para o novo cenário com dois robôs de patrulha.. .....	71
Figura 6-28 - Mapa do cenário 3, com 16 posições iniciais e finais previamente definidas .....	73
Figura 6-29 - Taxa de Sucesso para uma frota de 4 robôs .....	74
Figura 6-30 – Média do comprimento do caminho inicialmente planeado vs. Distância percorrida, para uma frota de 4 robôs.....	74
Figura 6-31 – Média e Desvio Padrão do número de conflitos para uma frota de 4 robôs.....	75

Figura 6-32 - Taxa de Sucesso para uma frota de 10 robôs .....	76
Figura 6-33 - Média do comprimento do caminho inicialmente planejado vs. Distância percorrida, para uma frota de 10 robôs.....	76
Figura 6-34 - Média e Desvio Padrão do número de conflitos para uma frota de 10 robôs.....	77
Figura 6-35 - Taxa de Sucesso para uma frota de 16 robôs .....	78
Figura 6-36 - Média do comprimento do caminho inicialmente planejado vs. Distância percorrida, para uma frota de 16 robôs.....	78
Figura 6-37 - Média e Desvio Padrão do número de conflitos para uma frota de 16 robôs.....	79

## ÍNDICE DE TABELAS

Tabela 6-1 - Número de conflitos e replaneamentos necessários para cada robô consoante o tamanho do traçado verificado para a coordenação multi-robô .....	62
Tabela 6-2 - Comprimentos iniciais e finais do caminho calculado e do caminho percorrido, para uma frota de dois robôs.....	64
Tabela 6-3 - Comprimentos iniciais e finais do caminho calculado e do caminho percorrido, para uma frota de cinco robôs.....	65
Tabela 6-4 - Comprimentos iniciais e finais do caminho calculado e do caminho percorrido, para uma frota de dez robôs.....	68
Tabela 6-5 - Comprimentos iniciais e finais do caminho calculado e do caminho percorrido, para uma frota de dez robôs, para o novo cenário .....	70
Tabela 6-6 - Comprimentos iniciais e finais do caminho calculado e do caminho percorrido, para uma frota de dez robôs, para o novo cenário com dois robôs de patrulha .....	72

## LISTA DE ABREVIÇÕES, SIGLAS E ACRÓNIMOS

Neste documento são utilizadas diversas abreviaturas e siglas, sistematizadas de seguida:

<b>AGV</b>	<i>Automated Guided Vehicles</i>
<b>AMR</b>	<i>Autonomous Mobile Robots</i>
<b>CAPT</b>	<i>Concurrent Assignment and Planning of Trajectories</i>
<b>CCS</b>	<i>Control Center Server</i>
<b>CD</b>	<i>Coordination Diagram</i>
<b>CDL</b>	<i>Curvature Distance Lookup</i>
<b>CoBa</b>	<i>Constraint-Based approach</i>
<b>DCOP</b>	<i>Distributed Constraint Optimization Problems</i>
<b>Dec-POMDP</b>	<i>Decentralized Partially Observable Markov Decision Process</i>
<b>Dec-POSMDP</b>	<i>Decentralized Partially Observable Semi-Markov Decision Process</i>
<b>DNC</b>	<i>Divide and conquer</i>
<b>GPN</b>	<i>Guided Paths Networks</i>
<b>HRI</b>	<i>Human-Robot Interaction</i>
<b>IA</b>	<i>Instantaneous Assignment</i>
<b>IGSA</b>	<i>Improved Gravitational Search Algorithm</i>
<b>ILC</b>	<i>Intersection Local Coordinator</i>
<b>ILP</b>	<i>Integer Linear Programming</i>
<b>k-WTA</b>	<i>k-winners-take-all</i>
<b>LTL</b>	<i>Linear Temporal Logic</i>
<b>MAS</b>	<i>Multi-Agent Systems</i>
<b>MR</b>	<i>Multi-Robot Tasks</i>
<b>MRMP</b>	<i>Multi-Robot Motion Planning</i>
<b>MRS</b>	<i>Multi-Robot Systems</i>
<b>MRTA</b>	<i>Multi-Robot Task Allocation</i>
<b>MT</b>	<i>Multi-Task Robot</i>
<b>RMFS</b>	<i>Robot Mobile Fulfillment System</i>
<b>SMT</b>	<i>Satisfiability Modulo Theories</i>
<b>SR</b>	<i>Single-Robot Tasks</i>
<b>ST</b>	<i>Single-Task Robot</i>
<b>TA</b>	<i>Time extended assignment</i>
<b>UAV</b>	<i>Unmanned Aerial Vehicle</i>

## 1. INTRODUÇÃO

Neste capítulo é realizado um breve enquadramento e descrição da problemática e dos objetivos deste projeto de dissertação. Depois é apresentada a estrutura da dissertação.

### 1.1. ENQUADRAMENTO E MOTIVAÇÃO

A evolução das tecnologias de inteligência artificial e da automação permitiram o desenvolvimento de robôs cada vez mais sofisticados e capazes de lidar com ambientes complexos e em constante mudança, levando ao surgimento das frotas de veículos móveis autônomos. Estas frotas surgiram como uma forma de melhorar a eficiência e a precisão dos processos em ambientes de logística interna, direcionadas para a realização de tarefas repetitivas, como o carregamento e descarregamento de materiais, o transporte de objetos e matérias-primas e o armazenamento de produtos, otimizando a utilização dos recursos (Sardinha, 2017).

Para garantir a coordenação de uma frota de veículos móveis autônomos é necessário que o sistema detenha determinadas características, ou seja, deve-se garantir algum mecanismo de comunicação, atribuição de tarefas, prevenção de colisões, planeamento de caminhos, gestão de frotas, protocolos de segurança e flexibilidade. O planeamento de percursos é necessário para frotas de veículos móveis em ambientes intralogísticos uma vez que permite aos robôs navegar pelo ambiente de forma eficiente e segura (Yuan et al., 2010). Tal inclui evitar obstáculos e outros robôs, bem como planear o caminho mais eficiente para chegar ao seu destino. A coordenação entre a frota de robôs é também importante para assegurar que os robôs não se encontrem no caminho um do outro e minimizar colisões. Além disso, o planeamento do caminho pode ser utilizado para otimizar a atribuição de tarefas entre a frota de robôs, com o objetivo de melhorar o desempenho global do sistema.

A coordenação de frotas de veículos autônomos para ambientes intralogísticos é um dos campos ativos na área da robótica e está em constante crescimento com a evolução da tecnologia.

### 1.2. DESCRIÇÃO DO PROBLEMA

As frotas autônomas de veículos móveis podem ser definidas como grupos de robôs dotados tanto da mobilidade como de manipulação, sendo adequadas para coordenação em ambientes intralogísticos, tais como fábricas e armazéns.

A coordenação de uma frota de veículos móveis autônomos envolve a gestão das tarefas atribuídas a cada robô, como tarefas de *pick-and-place*, transporte de mercadorias e manipulação de objetos. Em ambientes intralógicos, a coordenação da frota pode ser utilizada para otimizar o fluxo de mercadorias, minimizar o tempo de viagem e o consumo de energia, e evitar colisões (Fragapane et al., 2021).

Há várias formas de coordenar uma frota de veículos móveis, entre as quais a utilização de um controlador central, controle descentralizado ou uma abordagem híbrida que combina tanto o controle centralizado como descentralizado. As abordagens de controle centralizado envolvem a utilização de um único controlador que gere a frota, determinando as tarefas e caminhos de todos os robôs. Os métodos de controle descentralizado, por outro lado, envolvem a distribuição do processo de tomada de decisão entre os robôs, com cada robô a tomar decisões locais com base em informações partilhadas com os robôs vizinhos. As abordagens híbridas são uma extensão dos métodos anteriormente mencionados, nos quais um controlador central é utilizado para gerir tarefas de alto nível, e o controle descentralizado é utilizado para tarefas de baixo nível. Em todos os casos, a coordenação da frota deve ter em conta as limitações do ambiente intralógico, tais como a disposição do armazém e a localização das mercadorias, bem como as capacidades e limitações dos próprios robôs móveis (Fragapane et al., 2021).

Um dos principais desafios nestes ambientes é manter um mapeamento e localização precisos, bem como uma atribuição, planeamento e execução eficiente das tarefas. Além disso, nestes ambientes, o problema de coordenação multi-robô pode ser altamente dinâmico, com robôs em constante movimento, recolhendo e movimentando objetos entre estações de trabalho. Desta forma, o algoritmo de coordenação deve ser capaz de reagir a estas mudanças, e encontrar rapidamente novas soluções.

### 1.3. OBJETIVOS DA DISSERTAÇÃO

Neste projeto de dissertação foi realizado um estudo relativamente aos sistemas multi-robô e ao planeamento de caminhos em ambientes conhecidos, mais concretamente para o caso em que é necessário implementar uma frota de veículos móveis autônomos em ambientes de logística interna (Hernández et al., 2018). Como tal, é necessário implementar uma estratégia capaz de coordenar os movimentos e as tarefas dos veículos da frota.

O primeiro objetivo desta dissertação foi estudar os sistemas multi-robô, nomeadamente como estes se comportam e quais as suas características. Para além disso, também foi realizado um estudo relativamente aos diferentes métodos de planeamento de caminhos existentes. Desta forma, e de acordo com o leque de algoritmos já implementados e disponibilizados pela *MathWorks*, foi necessário efetuar um estudo para determinar qual o algoritmo mais adequado para o problema proposto.

O segundo objetivo foi avaliar o comportamento do sistema implementado, quer de cada robô, isto é, como veículo individual, quer da frota de veículos como um todo. Foi necessário ter em atenção se cada robô efetuava um mapeamento individual ou se existia partilha de informação entre eles, verificar o comportamento individual e global quando eram detetados não só obstáculos estáticos, mas como também quando era detetado um outro robô da frota. Para além disso, foi essencial verificar se os veículos se comportavam consoante o esperado, ou seja, se veículos com menor prioridade cediam passagem aos de maior prioridade e o comportamento dos robôs quando se detetava algum conflito com os robôs de patrulha.

Por fim, realizou-se um estudo sobre o comportamento do sistema quando eram aplicadas diferentes regras de prioridade: (1) cedência de passagem ao veículo que se encontra o mais à direita, (2) atribuição de prioridade ao veículo cujo trajeto a percorrer é maior e, (3) atribuição de prioridade consoante a estação de trabalho alocada a cada robô. As três regras mencionadas foram aplicadas a frotas de quatro, dez e dezasseis robôs e as variáveis média das distâncias percorridas, número total de conflitos detetados e taxa de sucesso da missão foram analisados.

#### 1.4. ESTRUTURA DA DISSERTAÇÃO

Este documento foi dividido em sete capítulos. No segundo capítulo, Estado da Arte, foi realizada uma revisão da literatura acerca do tema desta dissertação. No terceiro capítulo, Fundamentos Teóricos, foram essencialmente abordados os diferentes métodos de busca existentes. No quarto capítulo, Ferramentas, foi abordada a ferramenta de simulação utilizada (MATLAB) e qual o cenário escolhido. No quinto capítulo, Proposta de Solução, foi realizado um estudo a alguns dos planeadores existentes, tendo por base algumas métricas fornecidas pelo MATLAB, sendo também apresentada uma proposta de solução para o problema em questão e qual o planeador global utilizado e a abordagem de resolução de conflitos adotada. No sexto capítulo, Testes e Resultados, foram descritos os testes e resultados obtidos para o planeamento de caminhos para um só robô e para uma frota de robôs de dimensões variáveis, e foi realizado um estudo sobre o comportamento de algumas variáveis,

aplicando três regras de prioridade distintas a frotas de diferentes tamanhos. Por fim, no sétimo capítulo, encontram-se as Conclusões desta dissertação.

## 2. ESTADO DA ARTE

Neste capítulo foi realizada uma revisão da literatura relativa a *Multi-Robot Systems*, a *Robot Mobile Fulfillment Systems* em algumas das áreas onde equipas de veículos autónomos podem ser aplicadas. Em 2.1, foi analisada uma possível classificação que pode ser utilizada para distinguir os diferentes tipos de sistema multi-robô existentes, sendo que pode ser distinguido como coordenado ou não coordenado, heterogéneo ou homogéneo, cooperativo ou competitivo, com ou sem comunicação e reativo ou deliberativo. Em 2.2 encontra-se uma classificação utilizada para a coordenação multi-robô, podendo esta ser estática ou dinâmica. Em 2.3 foi abordado o planeamento coordenado de tarefas e em 2.4 o planeamento coordenado de movimentos. Em 2.5 encontra-se uma breve descrição sobre o que é um *Robot Mobile Fulfillment System*. Por fim, em 2.6 foram abordadas algumas das áreas onde se pode aplicar equipas multi-robô, desde aplicações de vigilância, busca e salvamento até aplicações em ambiente adversarial.

### 2.1. SISTEMAS MULTI-ROBÔ

Nos últimos anos, os *Multi-Robot Systems* (MRS) obtiveram um reconhecimento considerável devido à sua eficiência e aplicabilidade em diferentes tipos de situações da vida quotidiana. Os MRS são definidos como sendo um grupo de robôs sistematizados sob a forma de uma arquitetura multi-agente de modo que possam trabalhar para o mesmo objetivo ou para objetivos diferentes. De modo a realizar uma grande variedade de tarefas de forma robusta, são utilizadas equipas de robôs em vez de recorrer a um único robô altamente especializado. Arai et al. (2002); Gustafson & Gustafson (2006); Parker (2008) apontam que um MRS, que trabalha de forma distribuída para executar tarefas coordenadas, fornece maior robustez e eficiência, sendo que tal não é possível com um único sistema robotizado.

Verma & Ranga (2021) definiram cinco dimensões que permitem classificar um MRS, tal como demonstrado na Figura 2-1.

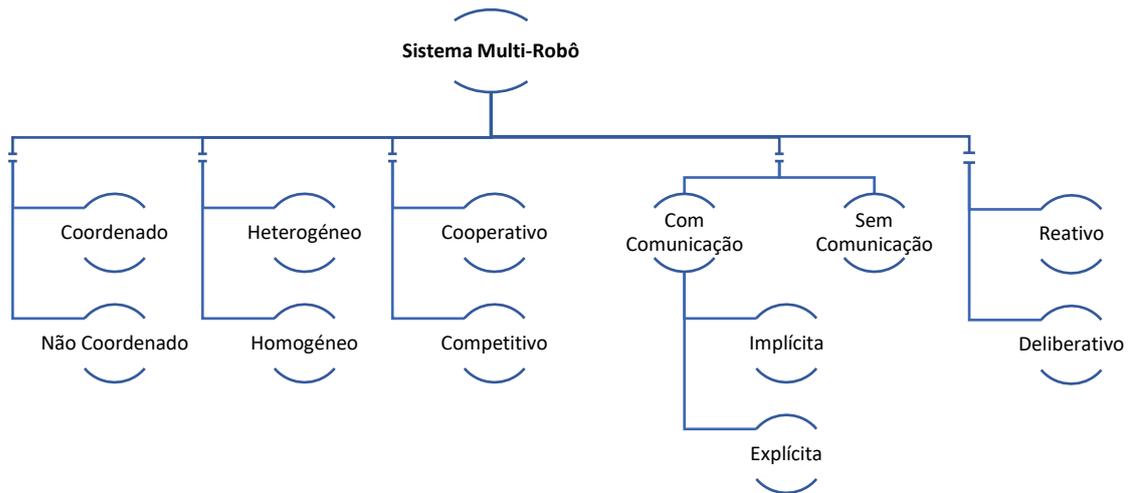


Figura 2-1 - Classificação do MRS. Adaptado de (Verma & Ranga, 2021)

### COORDENAÇÃO

A coordenação pode ser definida como um mecanismo utilizado para alcançar a cooperação, sendo que este mecanismo pode ser simples ou complexo, dependendo do nível de cooperação necessário. Com base na forma como é realizada a tomada de decisão no MRS para tornar o sistema cooperativo, a abordagem de coordenação utilizada pode ser centralizada ou descentralizada.

Porém, o MRS não tem de ser necessariamente coordenado, dado que muitas tarefas podem ser executadas e concluídas com eficiência mesmo sem a existência de coordenação. A vantagem de um MRS não coordenado passa pela baixa complexidade do sistema, com menos risco de defeitos. No entanto, também ocorre uma maior dissipação de recursos devido às intervenções necessárias, consequentes dos robôs que executam tarefas conflituosas. Um MRS coordenado é um sistema mais complexo, todavia pode prevenir ou até mesmo reduzir os problemas de um MRS não coordenado.

### COMPOSIÇÃO

Relativamente à composição de um MRS, este pode ser classificado como homogêneo ou heterogêneo. Num MRS homogêneo todos os robôs são constituídos com o mesmo *hardware* e o mesmo *software*. Por outro lado, num MRS heterogêneo, os robôs que constituem o sistema podem ter diferenças nos procedimentos de controlo de software ou hardware ou de ambos.

A principal vantagem de um MRS heterogêneo é a sua capacidade de fácil adaptação às diferentes situações que surgem num ambiente dinâmico, uma vez que este tipo de sistema possui um melhor alcance, em comparação com o MRS homogêneo, para lidar com tarefas novas e imprevisíveis. Como num sistema homogêneo todos os robôs podem executar tarefas semelhantes com a mesma

eficiência, quando existe uma falha de um robô, a tarefa pode ser alocada a um outro robô, permitindo que o MRS seja robusto. No entanto, a aplicabilidade a diferentes situações e ambientes é fraca pois todos os robôs do sistema possuem o mesmo *hardware* e *software*.

#### AMBIENTE COOPERATIVO E COMPETITIVO

Os ambientes multi-robô demonstram um comportamento coletivo como a sociedade humana (Russell & Norvig, 2010). No comportamento coletivo, os robôs reagem à influência ou estímulo habitual em circunstâncias instáveis, imprevisíveis, espontâneas e não estruturadas (Popenoe, 2000).

O comportamento coletivo é baseado no comportamento cooperativo e competitivo. O comportamento cooperativo refere-se à interação entre robôs a fim de executar uma tarefa paralelamente com o aumento da utilidade global do sistema. Desta forma, todos os robôs do sistema interagem entre si e trabalham para um objetivo comum. Por outro lado, o comportamento competitivo refere-se ao caso em que vários robôs competem entre si com o objetivo de satisfazer os seus próprios interesses.

#### COM COMUNICAÇÃO E SEM COMUNICAÇÃO

Os robôs podem cooperar entre si através de mecanismos de comunicação que lhes permitam partilhar informação entre eles. Por comunicação, entende-se que os robôs possam trocar alguma informação uns sobre os outros ou através da leitura efetuada pelos sensores. Logo, num sistema sem comunicação, não há informação disponível relativamente aos outros robôs. Kahng (1997) classificou a comunicação entre robôs em três categorias com base na forma de interação entre eles: interação através de comunicações explícitas, interação através da deteção e interação através do ambiente. Com base no modo como os robôs sentem ou partilham informação foram propostos dois tipos de comunicação: comunicação direta e indireta (Farinelli et al., 2004).

A taxonomia estabelecida por Verma & Ranga (2021), com base no modo de transmissão de dados, foi a de comunicação implícita e explícita. A comunicação explícita utiliza um *hardware* de comunicação, ou seja, um dispositivo dedicado a sinais que podem ser compreendidos por outros membros do sistema. A troca de informações entre os robôs acontece utilizando o *unicast*, ou seja, comunicação máquina a máquina, e transmitem mensagens intencionais. Em oposição, na comunicação implícita os robôs obtêm informações relativamente aos outros robôs do sistema através do ambiente, podendo ser obtida através da utilização de sensores.

## REATIVO E DELIBERATIVO

Iocchi et al. (2001) apresentaram uma classificação da MRS relativa a arquiteturas reativas e deliberativas. O MRS é considerado deliberativo quando os robôs conseguem lidar com qualquer mudança que possa ocorrer no ambiente, recorrendo de algum tipo de abordagem que se baseie na reestruturação dos comportamentos globais do sistema. Por outro lado, num sistema reativo, cada robô é responsável por lidar com as mudanças no ambiente fornecendo uma solução robusta para reorganizar a sua própria tarefa a fim de completar o seu objetivo inicial. Concluindo-se, assim, que a principal diferença entre um sistema reativo e um sistema deliberativo é a forma como o MRS recupera quando se depara perante uma situação imprevista. Enquanto num sistema deliberativo, o MRS opta por um plano a longo prazo a fim de atingir o objetivo global, num MRS reativo, o plano é fornecido diretamente ao robô em causa para lidar com o problema com o qual se deparou. Para além disso, um MRS que responde rapidamente às alterações ambientais que possam ocorrer, sem afetar os outros membros do mesmo, trata-se de um sistema reativo. Um dos fatores a ter em consideração para definir se um MRS é deliberativo ou não é o facto de se tratar ou não de um sistema coordenado. Caso não seja fortemente coordenado, trata-se de um MRS reativo ou parcialmente reativo; por outro lado, caso seja fortemente coordenado, pode ser um MRS deliberativo ou reativo.

## TAMANHO DA EQUIPA

O número de robôs constituintes do MRS é um parâmetro importante durante a implementação deste tipo de sistema, pois pode afetar o seu desempenho. Existem várias tarefas que não são possíveis de executar com um único robô, como as tarefas que requerem ações em simultâneo. Logo, e de modo a aumentar a eficácia e rapidez do sistema, são utilizados vários robôs no desempenho de tais tarefas. Portanto, para desenvolver uma abordagem de coordenação eficiente, é necessário ter em consideração a dimensão da equipa.

## 2.2. CLASSIFICAÇÃO DA COORDENAÇÃO MULTI-ROBÔ

A coordenação é considerada o mecanismo utilizado para atingir a cooperação. Logo, a coordenação e cooperação no MRS podem ser definidas como a operação ou ação conjunta entre um grupo de robôs (López et al., 2011). Independentemente do objetivo final dos robôs presentes no MRS, estes precisam de se coordenar a fim de controlar as ações de cooperação. Ferber (1999) publicou um artigo relativamente aos *Multi-Agent Systems* (MAS), onde definiu a coordenação como sendo a cooperação onde as ações realizadas por cada agente robótico têm em consideração as ações

executadas pelos outros agentes robóticos de tal forma que todo o processo acaba por ser uma operação coerente e de alto desempenho.

As abordagens de coordenação podem ser classificadas com base em vários parâmetros, como o modo de comunicação, de tomada de decisões, a adaptabilidade e protocolo. Desta forma, Verma & Ranga (2021) apresentou uma possível classificação da coordenação do MRS, tal como apresentado na Figura 2-2.

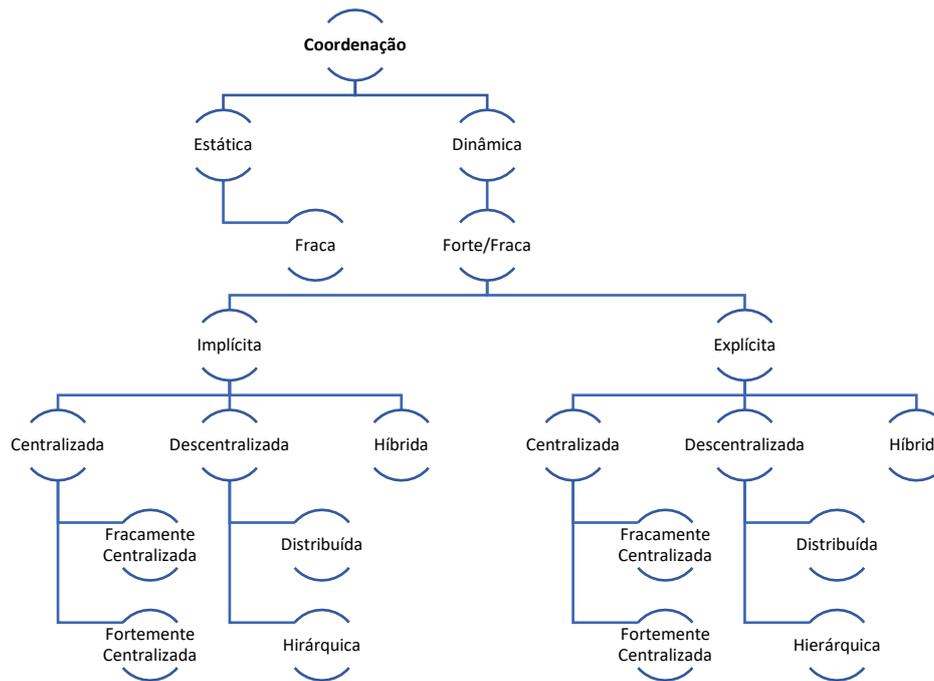


Figura 2-2 - Classificação ao nível da coordenação do MRS. Adaptado de (Verma & Ranga, 2021)

### COORDENAÇÃO ESTÁTICA E DINÂMICA

A coordenação pode ser dividida como sendo uma coordenação estática ou dinâmica (Yan et al., 2013b).

A coordenação estática baseia-se em algumas regras e convenções pré-definidas de forma a evitar colisões. Kato et al. (n.d.), para o caso em que existem dois robôs no MRS, estabeleceram regras de trânsito definidas à priori, de modo a evitar colisões, determinando assim, que um deles se mantinha à esquerda e outro à direita. Logo, a coordenação estática não deve ser utilizada em ambientes dinâmicos, pois não garante um controlo em tempo real e, por conseguinte, evitar colisões e bloqueios.

A coordenação dinâmica é definida como a coordenação que ocorre enquanto a tarefa é executada, e depende do estado atual do sistema, da sua localização e da informação sensorial. Apesar de, para este tipo de coordenação, muitos sistemas tomarem decisões baseando-se no seu estado atual, os

comportamentos podem ser definidos para determinados estados previamente idealizados. Desta forma, é necessário recorrer a algum tipo de comunicação, podendo ela ser uma comunicação implícita ou explícita.

Contudo, o ambiente pode ser demasiado denso ou as tarefas demasiado complexas, sendo difícil utilizar apenas a coordenação estática ou dinâmica. Portanto, dependendo do tipo de tarefa, quais os seus requisitos e qual o ambiente onde o robô se encontra inserido, pode ser benéfico utilizar tanto a coordenação estática como a dinâmica.

#### COORDENAÇÃO IMPLÍCITA E EXPLÍCITA

As abordagens de coordenação implícita utilizam dinâmicas de interação entre robôs e ambiente, que são frequentemente mais eficientes (exploram áreas vastas), demonstrando um elevado potencial, especialmente num MRS de grande escala e constituídos por robôs simples (Colby et al., 2015). De modo que os robôs tenham perceção das mudanças ambientais, são utilizados diferentes sensores ou dispositivos. Por outro lado, as abordagens de coordenação explícitas podem lidar com robôs mais sofisticados, garantindo, também, precisão na troca de informação entre robôs. No entanto, com a adição de robôs no sistema, o desempenho do sistema pode diminuir, podendo, em casos extremos, levar à falha geral do sistema.

Uma vez que a coordenação implícita garante estabilidade, tolerância às falhas e a fiabilidade do MRS, e a fim de alcançar um sistema eficiente, robusto e fiável, podem-se combinar métodos de coordenação implícita e explícita.

#### COORDENAÇÃO FRACA E FORTE

Dependendo de determinados fatores, como o nível de desempenho, qual a tarefa ou o tamanho da equipa, um MRS pode ser definido como não tendo coordenação, tendo uma coordenação fraca ou uma coordenação forte.

Numa coordenação fraca, o sistema não precisa de regras complexas, nem de comunicação direta para alcançar um comportamento coordenado. No entanto, pode precisar de comunicação implícita para efetuar apenas algumas trocas de informação. Como a coordenação fraca do MRS utiliza um método de coordenação que não depende de um protocolo (Ferraresso et al., 2001), então o MRS é mais robusto relativamente a falhas de comunicação.

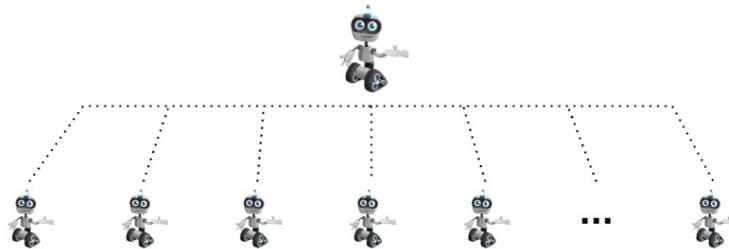
A coordenação forte é definida como sendo um método de coordenação que se baseia num protocolo de coordenação (Iocchi et al., 2001b). Logo, a fim de alcançar um comportamento

coordenado são necessárias implementar regras complexas, protocolos explícitos e comunicação. Neste caso, os protocolos explícitos são as regras que definem o comportamento dos robôs baseando-se na troca de informações entre os robôs.

#### COORDENAÇÃO CENTRALIZADA

A coordenação centralizada é realizada por um único robô coordenador, e que é responsável por tomar decisões relativas à coordenação relativamente a todos os outros robôs (Figura 2-3). Num MRS centralizado, existe um único robô ou servidor, apelidado de líder que é responsável pela alocação das tarefas existentes aos robôs presentes no sistema. No entanto, as abordagens centralizadas não são adequadas para a coordenação de um MRS constituído por uma grande equipa, uma vez que iria requerer uma elevada exigência computacional do líder, e um elevado custo na comunicação entre os robôs.

Nas abordagens de coordenação fortemente centralizadas, o papel do líder é atribuído a um único robô no início da tarefa global, e este continua a ser mesmo até ao fim da missão.



*Figura 2-3 - Esquemático da coordenação centralizada*

Por outro lado, nas abordagens fracamente centralizadas, é permitido que mais que um robô ou servidor seja líder durante o decorrer da missão. O líder não é escolhido previamente, podendo ser selecionado dinamicamente com base em determinados critérios, como o estado atual da tarefa, o ambiente, a comunicação e o nível de bateria.

Quando existem vários líderes, e estes são controlados por um único líder, também se está perante abordagens centralizadas. Pelo contrário, quando existem vários líderes, mas estes trabalham de forma independente uns dos outros, significa que não são controlados por um único líder, tratando-se de uma abordagem hierárquica.

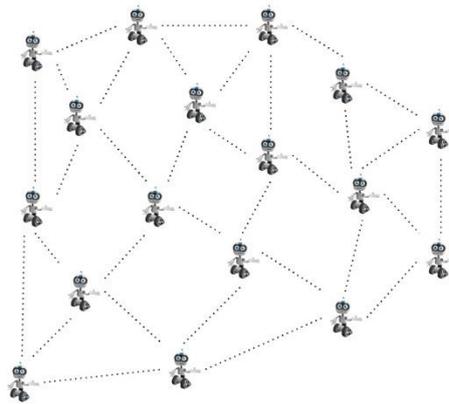
Por fim, pode-se concluir que as abordagens fortemente centralizadas são suscetíveis a falhas, ou seja, não são robustas, pois pode ocorrer algum tipo de defeito que eventualmente afetará o funcionamento do líder ou alguma falha de comunicação que pode levar ao fracasso completo de todo

o processo de coordenação. Já as abordagens fracamente centralizadas são mais robustas do que as fortemente centralizadas, uma vez que podem definir um novo líder caso ocorra alguma falha com o líder atual.

#### COORDENAÇÃO DESCENTRALIZADA

As abordagens descentralizadas podem ser classificadas em dois tipos, as abordagens distribuídas, onde todos os robôs são idênticos no que diz respeito à sua responsabilidade em termos de coordenação, e as abordagens hierárquicas, que são centralizadas localmente.

A coordenação distribuída necessita de um MRS distribuído, onde este sistema é composto por robôs independentes no que toca à tomada de decisão em relação uns aos outros. Neste tipo de sistema todos os robôs são iguais no que diz respeito ao controlo, cada robô toma uma decisão de forma autónoma, uma vez que não existe um único robô de controlo (Figura 2-4). Devido à tomada de decisões de forma independente, as abordagens de coordenação distribuídas garantem uma maior robustez e uma maior complexidade no que diz respeito à coordenação entre robôs.



*Figura 2-4 - Esquemático Coordenação Descentralizada Distribuída*

Na coordenação hierárquica, o MRS é constituído por líderes locais, mas estes não são controlados por um único líder. Este tipo de abordagens são menos robustas do que as distribuídas, no entanto apenas podem ser utilizadas com comunicação local ou com uma comunicação global com menor complexidade e custo.

#### COORDENAÇÃO HÍBRIDA

A coordenação é híbrida quando esta pode ser obtida utilizando tanto abordagens centralizadas como descentralizadas. Neste tipo de coordenação, a coordenação centralizada pode ser alcançada quando se pretende enviar periodicamente a posição do destino a todos os robôs a partir de um servidor central, por exemplo. Já a coordenação descentralizada pode ser alcançada com o

planeamento do movimento para chegar ao destino, sendo este enviado previamente por um servidor central.

### 2.3. PLANEAMENTO COORDENADO DE TAREFAS

Num MRS existem dois problemas fundamentais que devem ser tidos em conta, o problema do planeamento de tarefas e o do planeamento de movimentos. Mesmo que o MRS não seja coordenado, estes problemas podem existir à mesma, mas a diferentes níveis.

De modo a completar qualquer tarefa podem ser necessários um ou mais robôs. Apesar de algumas tarefas poderem ser realizadas recorrendo apenas a um único robô, com a utilização de um sistema multi-robô, a qualidade da tarefa realizada pode ser melhorada. O planeamento de tarefas pode ser dividido em três partes, a decomposição de tarefas (para tarefas complexas), atribuição de tarefas e alocação de tarefas. Esta última, num MRS, pode ser definida como o problema de decisão sobre qual dos robôs é responsável pela execução de uma tarefa, ou parte dela, com o objetivo de a completar e alcançar os objetivos do sistema. Algumas comunicações locais do MRS entre robôs de uma equipa com o ambiente, proporcionam um comportamento de equipa coordenado, também denominado de coordenação implícita. No entanto, na coordenação explícita, as tarefas são explicitamente atribuídas, sendo este problema conhecido como *multi-robot task allocation* (MRTA). Gerkey & Mataric (2004) propuseram três eixos para utilizar na descrição dos problemas MRTA:

- ***Single-task robots (ST) versus multi-task robots (MT)***: robôs de tarefa única, ou ST, significa que cada robô é capaz de executar no máximo uma tarefa de cada vez, enquanto os robôs multitarefas, ou MT, são capazes de executar múltiplas tarefas simultaneamente;
- ***Single-robot tasks (SR) versus multi-robot tasks (MR)***: tarefas de apenas um único robô, ou SR, significa que cada tarefa requer exatamente um único robô para a executar, enquanto tarefas com vários robôs, ou MR, significa que algumas tarefas podem requerer múltiplos robôs na sua execução;
- ***Instantaneous assignment (IA) versus time extended assignment (TA)***: atribuição instantânea, ou IA, significa que a informação disponível relativa aos robôs, às tarefas e ao ambiente permite apenas uma atribuição instantânea de tarefas aos robôs, sem planeamento para atribuições futuras; atribuição prolongada no tempo, ou TA, significa que está disponível mais informação, tal como o conjunto de todas as tarefas que terão de ser atribuídas, ou um modelo como se espera que as tarefas cheguem ao longo do tempo.

Logo, o MRTA é um problema de tomada de decisão dinâmica que muda com o tempo e com o ambiente, daí a alocação estática não ser recomendada para aplicação direta. Desta forma, outra abordagem que pode ser utilizada é a resolução do problema da alocação estática de forma iterativa. Farinelli et al. (n.d.) apresentaram uma solução baseada no modelo *Distributed Constraint Optimization Problems* (DCOP) para o problema da atribuição de tarefas, relevante para o cenário logístico do armazém. A tarefa era transportar objetos desde o cais de carga até ao cais de descarga com o mínimo de interferência. Para tal, recorreram a algoritmos heurísticos, mais concretamente o *binary max-sum* (montante máximo binário), que é um método iterativo de passagem de mensagens. Este modelo baseia-se na comunicação explícita, garante robustez ao sistema, mas sofre de um custo de comunicação elevado. Hu & Wang (2008) apresentaram uma rede neuronal simples capaz de resolver o problema *k-winners-take-all* (k-WTA). Jin et al. (2019) desenvolveram um novo modelo de coordenação para múltiplos robôs não-holonómicos presentes num ambiente competitivo para localização de alvos, possuindo comunicações limitadas. Nesta abordagem, apenas aos vencedores do concurso foi atribuída a tarefa e dada a autorização para avançar em direção ao alvo. Parker (1998) utilizou uma abordagem baseada numa arquitetura totalmente distribuída, com base no comportamento, que incorpora o uso de motivações matematicamente modeladas (tais como impaciência e consentimento) em cada robô para conseguir uma seleção de ação adaptativa.

Berhault et al. (2003) estudaram a coordenação de uma equipa de robôs móveis para visitar um certo número de alvos num terreno parcialmente desconhecido. Foram estudados leilões de item único, isto é, os robôs licitam apenas em alvos únicos, de modo a realizar a tarefa de exploração, no entanto estes não têm em consideração as sinergias entre os alvos. Portanto, foram projetados leilões combinatórios, ou seja, os robôs licitam sobre lotes de alvos para lidar com as dependências dos horários.

Na Figura 2-5 e Figura 2-6 estão representados dois robôs, R1 e R2, e quatro alvos, G1, G2, G3 e G4. Entre os alvos G3 e G4, e relativamente ao robô R1, existe uma forte sinergia positiva, uma vez que estão próximos um do outro e o robô consegue alcançar o segundo alvo com uma viagem de curta distância depois de ter alcançado o primeiro alvo. Por outro lado, e também para o robô R1, existe uma forte sinergia negativa entre os alvos G1 e G3, pois estão em lados opostos, mas, para alcançar o segundo alvo, o robô necessita de realizar uma viagem de maior distância após ter chegado ao primeiro alvo.

Os leilões combinatórios permitem que aos robôs licitadores licitem em pacotes de itens, isto é, o vencedor do leilão ganha todos os itens desse pacote, sendo capaz de incorporar as suas sinergias nas licitações. Tal como mostra a Figura 2-5, se os itens forem leiloados como itens únicos, então o robô R1 irá primeiro ao alvo G3 e depois ao alvo G1, e o robô R2 irá até ao alvo G4 e de seguida ao alvo G2. Desta forma, até chegar ao último alvo, cada robô teve de percorrer uma viagem total de 33 unidades. Enquanto, se os alvos fossem leiloados através de um leilão combinatório, o robô R1 iria leiloar e ganhar o pacote {G3, G4} e o robô R2, o pacote {G2, G1}, acabando por realizar uma viagem total de 17 unidades (Figura 2-6).

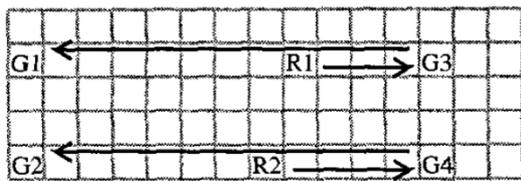


Figura 2-5 - Trajetórias do robô com leilões de item único.  
Retirado de (Berhault et al., 2003)

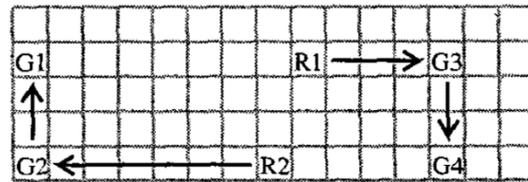


Figura 2-6 - Trajetórias do robô com leilões combinatórios.  
Retirado de (Berhault et al., 2003)

A maioria das abordagens baseadas em leilões e licitações assumem que a comunicação existente é perfeita, e tal não é realista. Portanto, foi apresentada uma abordagem inovadora por Otte et al. (2020) que trata de ambientes de comunicação limitada. Foi proposto o algoritmo M+ (Botelho & Alami, 1999) que se baseia numa abordagem descentralizada e dividida em três camadas: uma componente responsável pela alocação de tarefas, outra pela tolerância a falhas e, por fim, uma componente de execução de tarefas, responsável pela coordenação. Smith (1980) desenvolveu o protocolo de rede contratual (CNP - *Contract Net Protocol*), para solucionar os problemas de comunicação e controlo de nós recorrendo a uma abordagem distribuída. A distribuição de tarefas é afetada por um processo de negociação, onde é inicializada uma discussão entre os nós com as tarefas a serem executadas e com os nós com os robôs com capacidades de executar essas tarefas. MURDOCH, uma variante do protocolo *Contract Net*, e implementado por Gerkey & Mataric (2002), demonstrou ser extremamente reativo a mudanças no ambiente, incluindo falhas abruptas de robôs e introdução de novas tarefas. Trigui et al. (2014) apresentaram dois algoritmos distribuídos baseados no mercado para resolver o problema da atribuição de tarefas, onde N robôs competem por N tarefas, pressupondo que a cada robô só pode ser atribuída uma única tarefa. No primeiro algoritmo, denominado de *Distributed Market-Based*, cada robô concorre a uma tarefa. O segundo algoritmo é uma extensão do anterior, consistindo na troca de tarefas entre robôs, com o objetivo de melhorar a eficiência de toda a tarefa, reduzindo o seu custo. Viguria et al. (2008) apresentaram um outro algoritmo distribuído baseado no mercado denominado de "S+T", que resolve o problema MRTA em

aplicações que requerem a cooperação entre os robôs para completar todas as tarefas. Além disso, este algoritmo permite a adaptação de parâmetros de modo a dar prioridade ao tempo de execução ou ao consumo de energia da missão.

Xia et al. (2020) propuseram uma arquitetura descentralizada de uma frota de robôs e um método de atribuição dinâmica de tarefas a esta mesma frota, apresentando também uma comunicação baseada em licitações. Primeiramente foi desenvolvido um algoritmo dinâmico descentralizado de atribuição ou realocação de tarefas para coordenar tarefas num ambiente de *smart factory*. Através deste algoritmo, os veículos de uma frota podiam licitar entre si para determinar qual a tarefa mais adequada para cada um deles. Esta frota possuía a flexibilidade de reconfigurar dinamicamente a sua distribuição de tarefas em caso de contingências, como avaria de um robô ou adesão de um novo veículo ao sistema. Posteriormente, foi introduzido um sistema de gestão de robôs generalizado e escalável que permitia a colaboração entre os robôs e outras máquinas dentro da *smart factory*.

Na Figura 2-7, após ter sido introduzida uma tarefa no sistema, representada a vermelho, e esta ser alocada a um determinado veículo, procedeu-se ao registo de uma nova tarefa, representada a amarelo. Como esta nova tarefa se encontrava mais próxima do AGV, o veículo abandonou a primeira tarefa que lhe tinha sido alocada e moveu-se em direção à nova que tinha sido publicada.

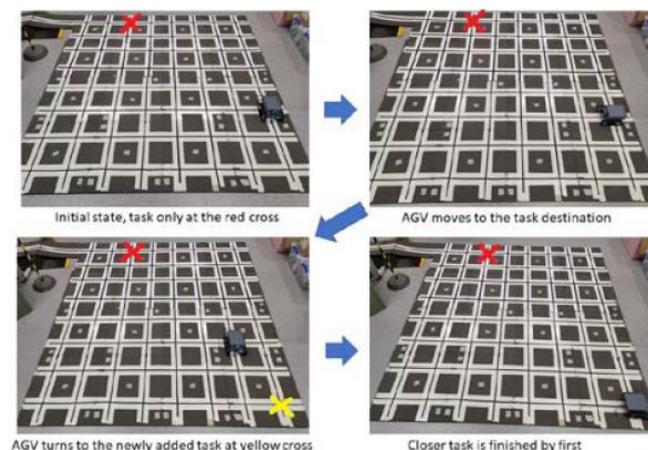


Figura 2-7 - Realocação dinâmica de tarefas quando é introduzida uma nova tarefa. Retirado de (Xia et al., 2020)

Por outro lado, a Figura 2-8 representa a situação de quando existiam duas tarefas (vermelho e amarelo) disponíveis no sistema e um AGV (preto), cuja localização se encontrava mais próxima da tarefa marcada a vermelho. Desta forma, o veículo movia-se em direção a esta tarefa. Posteriormente, foi inserido um novo veículo no sistema, representado a cinzento, numa localização ainda mais próxima da tarefa a vermelho do que o veículo preto. Consequentemente, a tarefa a vermelho foi realocada para o veículo cinzento e o veículo preto dirigia-se à tarefa representada a amarelo.

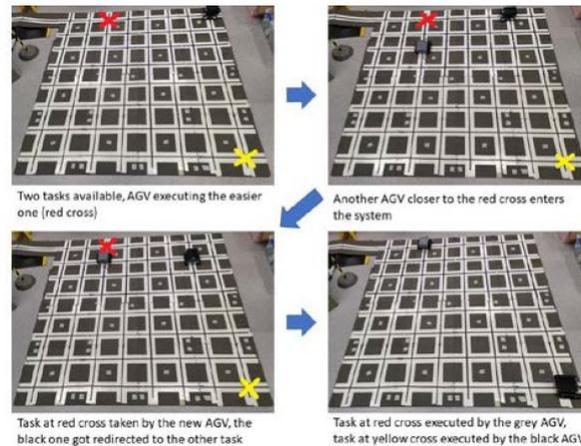


Figura 2-8 - Reconfiguração dinâmica quando entra um novo AGV na frota. Retirado de (Xia et al., 2020)

Kloetzer et al. (2012) forneceram uma solução que se foca na formulação de um estado finito (discreto), onde ocorre uma abstração do ambiente para um gráfico ponderado, e os *clusters* de amostras aparecem aleatoriamente nos nós deste gráfico. Trata-se de uma abordagem centralizada que utiliza uma unidade central para comunicar com todos os robôs. Liu et al. (2013) propuseram uma solução baseada no mercado para a coordenação multi-robô, à qual chamaram de CoBA (*Constraint-based Approach* para MRS), que permite que as tarefas sejam negociadas entre uma equipa de robôs heterogêneos num ambiente complexo que suporta a árvore de tarefas AND/OR e leilão de tarefas direto/indireto. Para executar uma tarefa AND, todas as subtarefas devem ser executadas. Por outro lado, para executar uma tarefa OR, qualquer uma das subtarefas pode ser executada. As restrições de comunicação são utilizadas para permitir que um dado robô possa apresentar uma licitação em nome de um outro robô durante o leilão de tarefas, denominando-se desta ação de oferta indireta. De Lope et al. (2013) propuseram uma abordagem baseada em Modelos de Limiar de Resposta e Algoritmos Probabilísticos de Aprendizagem baseados em Autômatos. Esta abordagem consiste numa solução descentralizada, em que os próprios robôs, de forma autónoma e individual, são responsáveis pela seleção de uma determinada tarefa de modo que todas as tarefas existentes sejam distribuídas e executadas de forma otimizada.

Turpin et al. (2014) propuseram duas soluções para o problema de atribuição simultânea e do planeamento de trajetórias, designado por CAPT (*Concurrent Assignment and Planning of Trajectories*), para uma equipa de robôs. Este problema envolve a abordagem simultânea de dois desafios. O primeiro trata-se de um problema complexo de encontrar uma atribuição adequada de robôs aos locais de destino. O segundo diz respeito à geração de trajetórias sem colisões e parametrizadas no tempo para cada robô. Foram propostas duas soluções algorítmicas para duas variações do problema CAPT.

O primeiro algoritmo, c-CAPT, é uma solução centralizada para o problema da atribuição de tarefas e planeamento de trajetórias que minimizam o custo funcional baseado no quadrado da velocidade ao longo da trajetória e demonstram que as trajetórias obtidas eram globalmente ótimas e seguras. Por outro lado, o segundo algoritmo, d-CAPT, é um algoritmo descentralizado que se baseia na troca de informações entre robôs dentro do raio de comunicação. O d-CAPT fornece resultados sub-ótimos em comparação com o c-CAPT.

Omidshafiei et al. (2017) concentraram-se na resolução de problemas gerais de planeamento multi-robô em espaços contínuos com observabilidade parcial, dada uma descrição de alto nível do domínio. Esta abordagem baseia-se em Processos de Decisão Markov Parcialmente Observáveis Descentralizados (Dec-POMDP), de modo a tirar partido das representações de alto nível que são naturais para problemas multi-robô e facilitar soluções escaláveis para grandes problemas discretos e contínuos. A formulação utilizada, Processo de Decisão Semi-Markov Parcialmente Observável Descentralizado (Dec-POSMDP), utiliza macro ações de tarefas criadas a partir de ações locais de nível inferior que permitem a tomada de decisões assíncronas pelos robôs, algo que é crucial em sistemas multi-robô. Otte et al. (2017) apresentaram três algoritmos de leilão para a atribuição de tarefas multi-robô nos casos em que a comunicação entre os robôs não é fiável. Schillinger et al. (2016) propuseram uma abordagem para identificar subtarefas numa representação automática da especificação da missão e, simultaneamente, atribuir as tarefas e planear a sua execução. Utiliza a Lógica Temporal Linear (LTL) para definir uma missão de alto nível e especificações de tarefas. Saha et al. (2016) apresentaram um algoritmo incremental baseado em teorias do módulo de satisfação, SMT (Barrett & Tinelli, 2018), onde os robôs foram divididos em pequenos grupos, com base num algoritmo de atribuição de prioridades. O algoritmo incremental apresentado considera os grupos de robôs um por um com base na sua prioridade e sintetiza as trajetórias para o grupo de robôs em conjunto. Ao sintetizar as trajetórias para os robôs de um grupo, o algoritmo considera os robôs com maior prioridade como obstáculos dinâmicos, e introduz um atraso mínimo na execução das trajetórias de custo ótimo para evitar a colisão com os robôs de maior prioridade. A escalabilidade do algoritmo foi testada, e demonstrou-se que é possível realizar o planeamento do movimento até vinte e cinco robôs num espaço de trabalho compacto, e até cinquenta robôs num espaço de trabalho livre de obstáculos (Figura 2-9 e Figura 2-10, respetivamente).



Figura 2-9 - Trajetórias sintetizadas para 25 robôs num espaço de trabalho compacto. Retirado de (Saha et al., 2016)

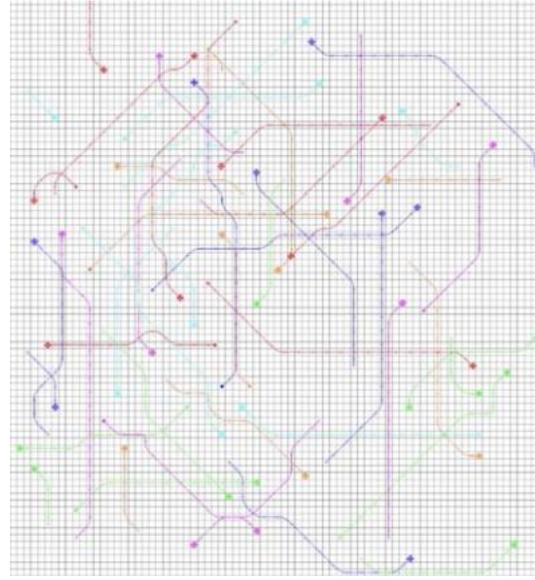


Figura 2-10 - Trajetórias sintetizadas para 50 robôs num espaço de trabalho sem obstáculos. Retirado de (Saha et al., 2016)

#### 2.4. PLANEAMENTO COORDENADO DE MOVIMENTOS

Como referido, o planeamento de movimentos é um dos problemas fundamentais do MRS. Quando os robôs estão a trabalhar no mesmo ambiente, é crucial que estes se coordenem entre si de modo a gerar um caminho eficiente para cada robô, ou seja, um caminho curto, sem bloqueios e sem colisões. Para tal, é necessário considerar o movimento dos outros robôs na fase de planeamento do movimento. O planeamento do movimento de vários robôs (MRMP) tem em consideração os obstáculos estáticos e dinâmicos no ambiente e qualquer interferência que possa ocorrer entre robôs, incluindo o planeamento de caminho e da trajetória. Dado um ambiente, a solução para o problema do planeamento do movimento consiste na geração de um movimento contínuo de um ponto para outro.

A maioria das abordagens existentes relativas ao MRMP são expandidas a partir dos resultados de um sistema composto por apenas um único robô. Yan et al. (2013a) abordaram três das principais abordagens existentes, entre elas, a decomposição de células, campos potenciais e *roadmaps*. Através da identificação de alguns estados e caminhos conhecidos dentro do espaço livre, todas estas abordagens têm por base a representação do problema de planeamento do movimento contínuo num problema de pesquisa gráfica discreta. As abordagens de planeamento de caminhos também podem ser classificadas como acopladas e desacopladas. As abordagens desacopladas dividem o problema em partes, e podem tanto ser centralizadas como descentralizadas. Para além disso, podem planear o caminho de cada robô de forma discreta e só à posteriori é que se preocupam com o problema de evitar colisões, sendo o planeamento para cada robô independente e podendo ser aplicado em tempo

real. Por outro lado, as abordagens acopladas são capazes de encontrar soluções ótimas ou quase ótimas (Standley, 2010; Standley & Korf, 2011; Wagner & Choset, 2015). Consequentemente, quando ocorre um conflito, por exemplo, a complexidade aumenta com o número de robôs, tornando-se algo irrealista para a coordenação de um MRS composto por um elevado número de robôs.

Olmi et al. (2011) propuseram uma abordagem baseada numa unidade centralizada de controlo de tráfego que definia a coordenação da frota de forma incremental durante o movimento dos veículos. Desta forma, tornava-se possível ter em consideração eventos inesperados que podiam causar desvios de alguns veículos das ações planeadas. O algoritmo incremental determinava quais os segmentos reservados para cada veículo, isto é, os segmentos que o robô estava autorizado a seguir, de modo que não ocorressem colisões.

A Figura 2-11 demonstra o modo de funcionamento deste algoritmo. Cada lista de segmentos reservados, de um determinado veículo, não deve conter segmentos que possam colidir com os das listas reservadas dos restantes.

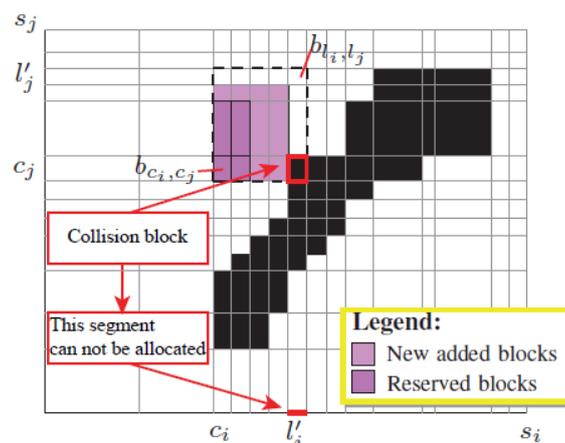


Figura 2-11 – Exemplo de CD para dois veículos. O segmento  $l'_i$  não pode ser adicionado à lista de reservas pois o bloco  $b_{l'_i, c_j}$  é um bloco de colisão. Retirado de (Olmi et al., 2011)

Digani et al. (2014) propuseram uma estratégia combinada para o planeamento de trajetos e criação do mapa de rotas com o propósito de coordenar múltiplos robôs, também referida como estratégia de negociação (Figura 2-12). Concluiu-se que era facilmente possível gerir um elevado número de robôs quando recorrendo a esta abordagem, garantindo também a prevenção de colisões. Duas das principais vantagens desta aplicação eram a capacidade de comunicação entre os veículos que se encontram na sua vizinhança e o conhecimento prévio do ambiente por onde circulava.

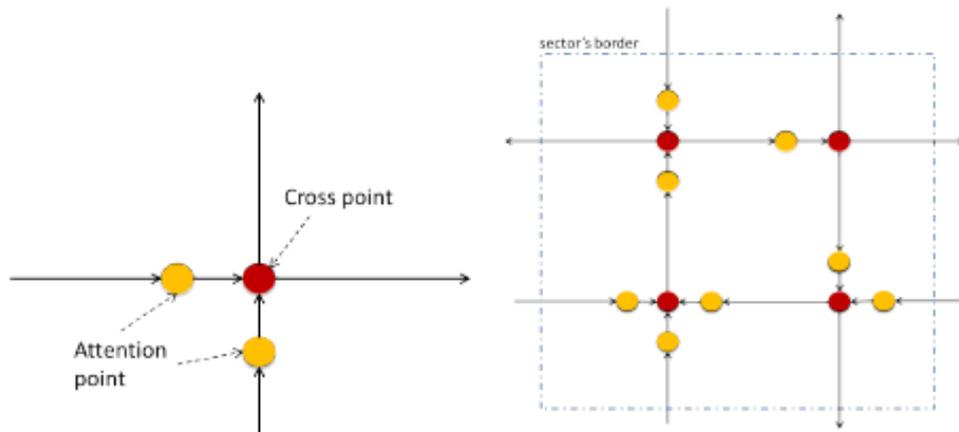


Figura 2-12 - Representação de uma interseção num mapa de rotas. A amarelo estão representados os pontos de atenção, responsáveis por alertar os AGV que estão a chegar a uma interseção. A vermelho estão os pontos de cruzamento, estes resultam da interseção de dois ou mais segmentos. Cada ponto de cruzamento só pode ser alocado a um único AGV em cada negociação. Retirado de (Digani et al., 2014)

Com o objetivo de minimizar a quantidade de tempo que os AGV despendiam para negociar padrões de tráfego complexos dentro de cada setor a fim de evitar colisões, Digani et al. (2015) apresentaram uma nova estratégia de coordenação local baseada numa abordagem centralizada otimizada. A estratégia de coordenação adotada, procurou maximizar o rendimento dos AGV através de cada setor, minimizando o número de interações entre veículos que aqui podiam ocorrer. A estratégia otimizada que foi adotada para o planeamento de percursos entre setores, tinha em vista reduzir ao máximo o tempo total de travessia, tentando evitar por completo as negociações, que eram responsáveis por atrasos reais. Para tal, foi necessário escolher a melhor velocidade para cada AGV, minimizando o tempo total de travessia e evitando simultaneamente conflitos com outros veículos. Apesar de não ter sido demonstrada a existência de uma solução viável em todos os cenários possíveis, os resultados empíricos sugeriram a existência de soluções para o problema de otimização na maioria das condições realistas.

A Figura 2-13 representa o tempo total em função do número de veículos, concluiu-se que a estratégia otimizada proposta superou a estratégia de negociação descentralizada sugerida em Digani et al. (2014).

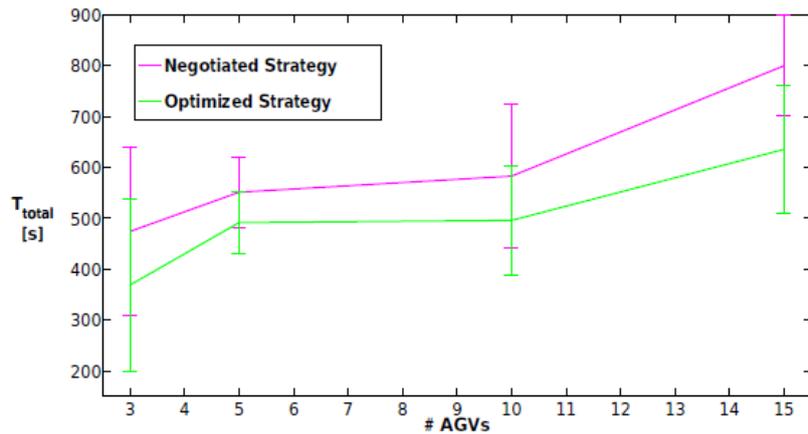


Figura 2-13 - Tempo total versus número de AGV para a Estratégia de Negociação e para a Estratégia Otimizada. Retirado de (Digani et al., 2015)

As frotas de múltiplos robôs precisam de ser coordenadas para assegurar a eficiência do transporte. Para tal, Lutz et al. (2016) propuseram uma abordagem distribuída para robôs autônomos que formavam uma frota capaz de executar tarefas de transporte dentro de um sistema de produção industrial. O sistema, incluindo a frota de robôs, podia reagir e adaptar-se rapidamente a mudanças que podiam ocorrer no fluxo de produção, reconfigurando apenas o *software*. Por esta razão, foram utilizados sistemas baseados na localização partilhada para aumentar a eficiência global do transporte. Foi necessário combinar a previsibilidade e eficiência da navegação baseada em caminhos com a flexibilidade de reagir rapidamente a obstáculos dinâmicos, utilizando a abordagem CDL (Schlegel, 1998). A abordagem CDL utiliza tabelas de pesquisa geradas offline para acelerar o tempo de execução e para permitir contornos de robôs de forma poligonal enquanto considera a cinemática e dinâmica do robô. O princípio orientador utilizado para seguir o caminho era análogo a seguir um ponto virtual no caminho à frente do robô. Caso o robô se deparasse com um obstáculo, as velocidades admissíveis podiam tomar valores de zero (situação em que o robô se encontra parado em frente ao obstáculo). Portanto, foi utilizada uma função objetiva que prefere as velocidades admissíveis que mantinham o robô em movimento enquanto o aproximavam para um ponto intermédio. Desta forma, e em todos os casos, o CDL garantiu um movimento seguro até onde o robô tinha a capacidade de perceber o ambiente envolvente.

A Figura 2-14 representa a navegação de uma frota de quatro robôs que operavam numa área pequena e densa.

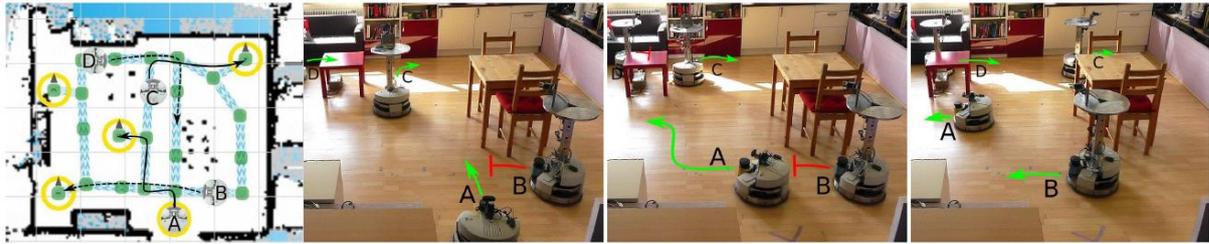


Figura 2-14 - Ambiente real com uma frota de quatro robôs e navegação coordenada num espaço pequeno e denso. A imagem da esquerda mostra o layout da rede de caminhos e as rotas que cada robô vão seguir. Retirado de (Lutz et al., 2016)

Jose & Pratihari (2016) apresentaram alguns métodos heurísticos para a atribuição de tarefas e planeamento de percursos sem colisões para três robôs que trabalham no mesmo espaço de trabalho, cujo tempo total de conclusão da tarefa deveria ser o mínimo possível. Foram utilizados o algoritmo genético para a atribuição de tarefas, e o algoritmo A\* foi utilizado para o planeamento de percursos. Patle et al. (2018) basearam-se no algoritmo *Firefly* (FA) para a navegação de robô móvel em ambiente dinâmico. Este algoritmo consiste na variação do brilho do *Firefly*, variando também a atração de um *Firefly* em direção ao seguinte. O controlador proposto era capaz de gerar caminhos ótimos aptos para evitar obstáculos de forma eficaz num tempo mínimo e com um custo computacional baixo. Das et al. (2021) propuseram um algoritmo de pesquisa gravitacional melhorado (IGSA – *Improved Gravitational Search Algorithm*) para o planeamento de caminhos para vários robôs, a fim de encontrar uma trajetória otimizada sem colisões, num ambiente dinâmico.

Cap et al. (2015) apresentaram uma versão adaptada do planeamento prioritário clássico, ao qual se deu o nome de planeamento prioritário revisto. Foi proposta uma versão descentralizada do planeamento prioritário, tanto do clássico como do revisto. Esta abordagem garante a computação de trajetórias coordenadas através da execução de um protocolo baseado na negociação entre robôs individuais. O planeamento prioritário é normalmente utilizado para o planeamento de múltiplos robôs, onde a cada robô lhe é atribuído um nível de prioridade diferente. O algoritmo progride começando pelo robô de prioridade máxima até ao robô de prioridade mais baixa e, a cada iteração, o robô executa a sua trajetória de modo a não entrar em conflito com os robôs de prioridade mais alta.

Jain et al. (2018) apresentaram uma abordagem descentralizada, baseada em eventos, onde um controlador cooperativo é concebido a fim de alcançar a coordenação entre robôs que seguem um caminho de referência definido à priori. A transmissão de dados necessários é transmitida em tempos de eventos discretos, reduzindo o custo de comunicação. Quando comparando com outras abordagens, a solução proposta apresenta um bom desempenho em termos de escalabilidade e custo de comunicação. Os métodos baseados em eventos decompõem o problema em duas partes; uma que

trata do controlo do movimento do robô individual, ou seja, do caminho a seguir; outra, que se trata do controlador cooperativo, que trabalha de modo a atingir a coordenação entre os robôs.

Kanjanawanishkul (2014) foi proposto um controlador baseado no modelo de controlo preditivo. A abordagem aplicada considera a formação de todo o sistema como um corpo rígido, isto é, uma estrutura virtual. Consoante o movimento de cada robô e da dinâmica da estrutura virtual, as leis de controlo são otimizadas para os robôs líderes e os reais. Luis et al. (2020) apresentaram um modelo distribuído de controlo preditivo capaz de gerar trajetórias em tempo real para múltiplos robôs. Desta forma, as perturbações são tidas em conta, calculando de forma eficaz trajetórias que não colidem com tarefas em transição. Solovey et al. (2016) apresentaram uma abordagem que resulta na combinação da representação implícita de *roadmaps*, para o planeamento de movimentos multi-robô, com um novo algoritmo, para a localização de caminhos. O algoritmo proposto, denominado de RRT discreto (dRRT), é uma adaptação do algoritmo RRT para o caso discreto de um gráfico, permitindo uma exploração rápida do *roadmap*.

Zhou et al. (2017) estudaram o controlo do movimento de MRS onde a cada robô lhe é atribuído um caminho pré-determinado e fechado para executar movimentos persistentes. Foi proposto um algoritmo em tempo real e distribuído para evitar colisões e bloqueios, parando e retomando o movimento dos robôs. No algoritmo, cada robô deve verificar os dois estados seguintes consecutivos para determinar se pode avançar.

Fanti et al. (2018) propuseram uma estratégia de controlo descentralizada para a atribuição de caminhos e coordenação de AGV em Redes de Caminhos Guiados (GPN). A GPN era controlada por zonas, ou seja, era necessário atribuir um conjunto de tarefas, que precisavam de ser executadas em algumas zonas, a um conjunto de AGV, controlando o movimento destes veículos de tal modo a que fosse possível evitar bloqueios e colisões. A estratégia proposta foi dividida em dois problemas. O primeiro dizia respeito à atribuição descentralizada das tarefas e respetivos destinos finais, assim como qual a rota para aqui chegar. O segundo, era relativo ao problema de coordenação descentralizada que era executado pelos AGV para chegar aos destinos anteriormente atribuídos, de modo que evitasse colisões e bloqueios.

Demesure et al. (2018) apresentaram um planeamento e escalonamento de movimentos descentralizados de AGV numa indústria flexível. O problema de escalonamento para cada robô dizia respeito a encontrar qual o melhor recurso sobre o qual seria realizada a operação de transporte do produto. Foi utilizado um planeador de movimento combinado com um escalonador de modo a

selecionar o destino da operação em curso de forma descentralizada e sem parar para sincronizar o escalonamento. A arquitetura global do sistema englobava um supervisor, responsável pela detecção de possíveis colisões entre robôs, sendo possível antecipar os conflitos e proporcionar um escalonamento mais apropriado e com melhores desempenhos globais. Devido à abordagem descentralizada que foi utilizada, as trajetórias do veículo eram atualizadas gradualmente ao longo do tempo em duas etapas. A primeira etapa, responsável por gerar uma presumível trajetória, representava a intenção do robô. Enquanto na segunda etapa, onde se trocava as informações sobre as intenções com robôs vizinhos, planeava a trajetória final.

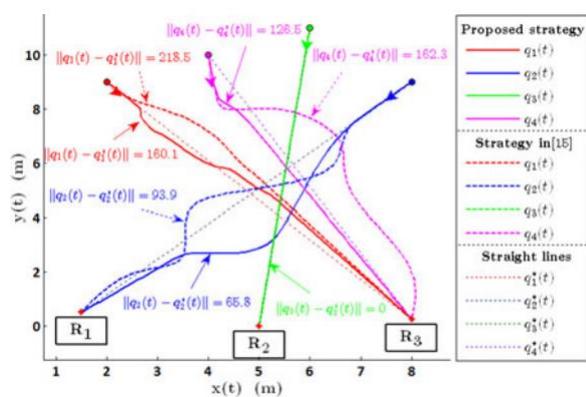


Figura 2-15 - Comparação das trajetórias dos agentes. Retirado de (Demasure et al., 2018)

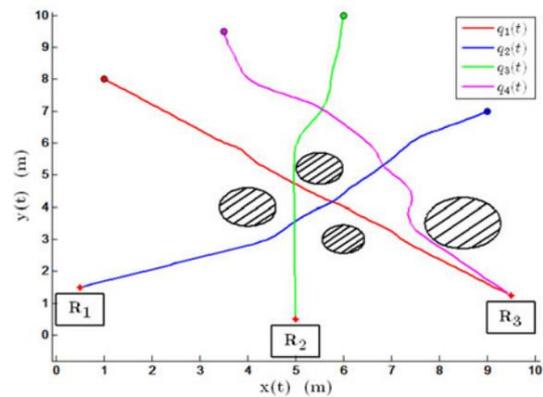


Figura 2-16 - Cenário com vários obstáculos. Retirado de (Demasure et al., 2018)

Wang et al. (2021) propuseram um controlo híbrido composto pelo *Control Center Server* (CCS) e pelo *Intersection Local Coordinator* (ILC). O CCS utilizava a informação da localização do robô para atualizar o fator de congestionamento e, desta forma, evitá-la quando em funcionamento o algoritmo dinâmico de planeamento de trajetórias. O ILC era responsável por evitar conflitos em cruzamentos, utilizando o modelo de prioridade de tráfego para decidir qual a ordem de passagem dos robôs nestas interseções. Ou seja, quando um robô não se encontrava a trabalhar, o CCS atribuía-lhe tarefas e chamava o algoritmo de planeamento de trajetórias para gerar um caminho. Após o robô obter as tarefas e ter-lhe sido atribuído um caminho pelo CCS, verificava se esta trajetória possuía algum desvio. Caso o desvio existisse, o ILC determinava se se encontrava em condições de otimizações dos desvios. Também quando um robô se encontrava prestes a atravessar um cruzamento, necessitava de realizar um pedido de passagem ao ILC. Na existência de robôs em ambos os sentidos do cruzamento, o ILC determinava qual a ordem de passagem, seguindo o modelo de prioridade do tráfego.

Pratissoli et al. (2021) propuseram uma coordenação que tinha conhecimento do tráfego e uma estratégia de planeamento de percursos baseada numa arquitetura multicamadas que foi utilizada para

modelar o ambiente. Para resolver o problema de gestão de tráfego, o ambiente foi dividido em três camadas. A camada superior era dedicada à modelação do tráfego de veículos. A camada intermédia ficou responsável pelo planeamento de percursos dos AGV. E, por fim, a camada inferior que calculava o percurso no *roadmap* e tratava da coordenação dos veículos. Quanto à gestão do movimento de uma frota de AGV, a estratégia adotada era constituída por dois processos. O primeiro era relativo ao planeamento e replaneamento da rota que tinha sido definida para cada veículo, baseando-se em informações de todas as camadas. O segundo processo consistia na coordenação de modo a evitar conflitos e bloqueios, realizada na camada inferior.

### 2.5. ROBOT MOBILE FULFILLMENT SYSTEM

Em 1989 foi concebido o conceito de *Robot Mobile Fulfillment System* – RMFS, (Azadeh et al., 2019), no entanto este apenas foi implementado décadas mais tarde (Giuzzo, 2008) pela Kiva Robots. Este tipo de sistema é caracterizado pela capacidade de levantar e transportar prateleiras móveis, através de robôs móveis autónomos (*Autonomous Mobile Robots - AMR*), até aos pontos de recolha, aumentando assim a produtividade do armazém.

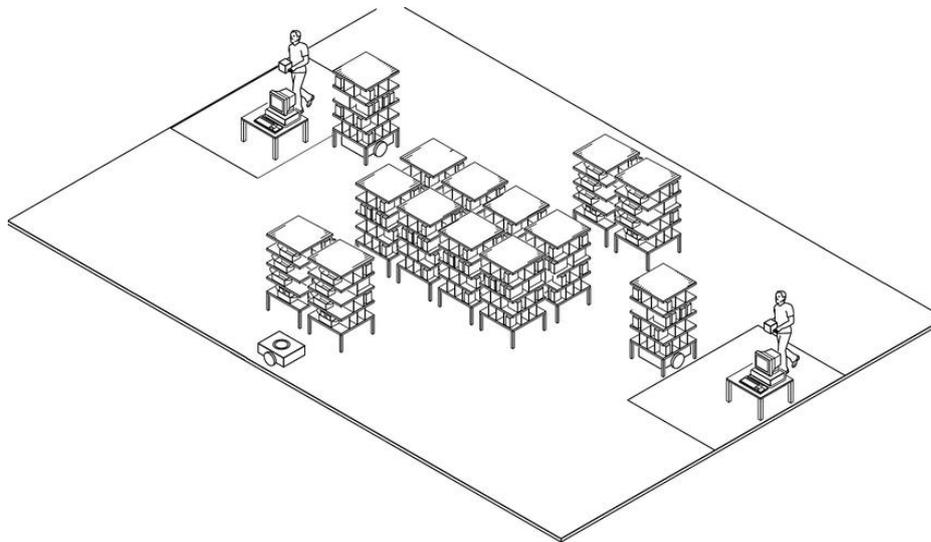


Figura 2-17 - Processo central de um RMFS. Retirado de (Hoffman et al., 2016)

O RMFS é um sistema complexo que envolve decisões a nível estratégico, tático e operacional. Li et al. (2021), centraram-se apenas nas decisões ao nível operacional, ou seja, decisões relativas à execução de tarefas neste tipo de sistemas. Focaram-se nas decisões de despacho de robôs ao nível operacional de forma a avaliar a eficácia e eficiência do RMFS de alta densidade, perante um conjunto de tarefas geradas.

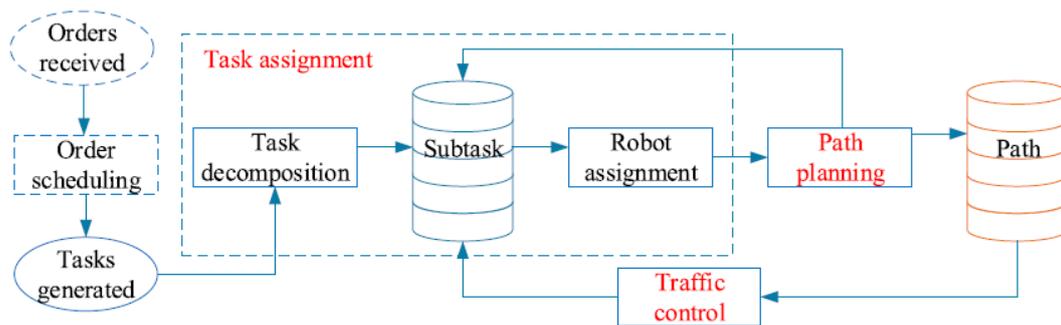


Figura 2-18 - Processo de decisão de um sistema RMFS de alta densidade. Retirado de (Li et al., 2021)

A Figura 2-18 ilustra o processo de decisão do sistema RMFS de alta densidade proposto pelos autores. Este processo foi dividido em módulos: atribuição de tarefas, planeamento de caminhos e controlo do tráfego. Inicialmente, e por se tratar de um sistema de natureza RMFS de alta densidade, as tarefas são decompostas em subtarefas. Posteriormente, é atribuído a cada robô uma subtarefa, seguindo uma *“greedy policy”*. Para cada robô cuja subtarefa já lhe tenha sido atribuída, o módulo de planeamento de caminhos gera um caminho para o veículo, podendo este caminho gerado incluir o caminho de aproximação, o de busca e o de regresso. Por fim, o módulo de controlo de tráfego monitoriza os robôs em tempo real para garantir a prevenção de situações indesejadas. Caso tal aconteça, ocorrerá uma recuperação do sistema através da geração de subtarefas específicas para redirecionar o robô.

De modo a controlar todos os robôs da frota foi utilizado um controlador central. Assim, em vez de os veículos comunicarem entre si, apenas necessitam de comunicar com o controlador central o seu estado atual, a sua localização atual e a próxima localização para a qual o robô deseja ir. O controlador é responsável por avaliar se o robô se pode mover em frente numa zona e comunicará ao mesmo a sua decisão.

Quando o espaço de armazenamento é limitado ou a sua ocupação é elevada, o RMFS de alta densidade é uma solução de armazenamento promissora. No entanto, neste estudo foram impostas muitas regras e políticas para auxiliar na tomada de decisão. Para além disso, as tarefas foram executadas de forma sequencial e não de uma forma otimizada.

## 2.6. ÁREAS DE APLICAÇÃO

Como já foi referido, o trabalho autónomo e coordenado de vários veículos num ambiente de trabalho partilhado permite, entre outros aspetos, melhorar os tempos de resposta, algo que é crucial na maioria das aplicações. Para além disso, o trabalho cooperativo do MRS aumenta a eficiência global

do sistema, pois permite a realização de diferentes tarefas em simultâneo. Adicionalmente, possibilita que o sistema seja configurado de diferentes formas, proporcionando, a este tipo de sistema, competências heterogêneas, tornando o MRS apto para que a cada agente lhe seja atribuída a missão que mais se adapta às suas especificações. Pode-se também afirmar que, devido à sua capacidade de tolerar falhas ou avarias, o MRS apresenta uma robustez bastante elevada.

Devido ao desenvolvimento exponencial que este tipo de sistemas sofreu e às soluções inovadoras que apresenta, tem uma vasta gama de aplicabilidade. O MRS tem sido vastamente utilizado em missões de emergência, como combates a incêndios, missões de busca e salvamento ou monitorização de áreas vastas. Darmanin & Bugeja (2017) atribuíram uma classificação a trabalhos recentes no âmbito dos sistema multi-robô de acordo com os seus domínios de aplicação.

#### VIGILÂNCIA, BUSCA E SALVAMENTO

As aplicações de vigilância, busca e salvamento são do relevo da vida quotidiana e têm atraído uma atenção considerável por parte da comunidade MRS nos últimos anos. Inicialmente, as aplicações de vigilância eram exclusivas ao patrulhamento de áreas interiores (Stancovici et al., 2016), mas com a introdução dos veículos aéreos não tripulados (UAV), os estudos das aplicações neste ramo foram expandidos de modo a incluir a vigilância de áreas exteriores, por exemplo áreas marítimas.

Desastres como o acidente nuclear de Fukushima em 2011, levaram os investigadores a implementar MRS avançados em aplicações da vida real, como busca e salvamento. Mendonca et al. (2016) propuseram um esquema cooperativo para uma equipa multi-robô para a vigilância de sobreviventes de naufrágios no mar. Couceiro et al. (2019) apresentaram um projeto onde um sistema formado por um UAV e um UGV, cuja missão era realizar ações que ajudassem a prevenir incêndios florestais. O UGV tinha como missão eliminar resíduos florestais, enquanto o UAV era responsável por explorar as áreas onde a ação do UGV era necessária. Gregory et al. (2016) abordaram uma aplicação de assistência humanitária e assistência em catástrofes. O objetivo desta investigação visava, simultaneamente, em avaliar os danos causados ao ambiente e localizar as vítimas.

#### FORRAGEAMENTO E PASTOREIO

Os robôs forrageiros são robôs móveis dotados da capacidade de busca e transporte de objetos para um ou mais pontos de recolha. Na robótica, o forrageamento é importante por diversos motivos:

- Representa uma ampla classe de problemas, como a exploração e navegação e identificação, manipulação e transporte de objetos;

- Nos sistemas multi-robô, no que diz respeito ao estudo do problema da cooperação robô-robô;
- Muitas aplicações reais são exemplos do forrageamento de robôs, como, limpeza, colheita, busca e salvamento, remoção de minas terrestres, entre outras.

A tarefa de *flocking*, ou pastoreio, é utilizada de forma a garantir que os membros do MRS possam seguir uma direção e manter-se na linha do rebanho para manter a coesão, mas, ao mesmo tempo, é necessário assegurar uma distância mínima entre eles para evitar colisão (Mendiburu et al., 2016).

#### *FORMAÇÃO E EXPLORAÇÃO*

Nas aplicações de formação, a equipa de robôs deve seguir uma determinada disposição de forma rigorosa, evitando, simultaneamente, obstáculos que ocorram no caminho. O obstáculo deve ser evitado coletivamente, sem que nenhum dos membros abandone a formação durante muito tempo. Por outro lado, nas aplicações de exploração, os robôs da equipa devem distribuir-se num ambiente desconhecido, com o objetivo de explorar a área.

#### *MANIPULAÇÃO COOPERATIVA*

A interação de um operador humano com uma equipa de robôs cooperativos, que manipulam um objeto de forma colaborativa com o Homem, colocou grandes desafios no desenvolvimento do controlo deste tipo de sistemas.

Sieber & Hirche (2019) utilizaram uma equipa robotizada controlada por um operador humano numa tarefa de manipulação cooperativa, recorrendo a uma abordagem de controlo baseada na formação. O operador humano, agarrava o objeto de forma rígida, manipulando-o sem esforço e através do movimento da sua mão foram definidos os pontos intermédios que serviram de guia para os robôs cooperativos.

#### *EQUIPA HETEROGÉNEA*

Uma equipa de robôs móveis heterogéneos diferencia-se pelo facto de ter a capacidade de a equipa lidar com tarefas complexas de um modo mais eficiente e eficaz, explorando as diferentes capacidades dos seus membros. A heterogeneidade pode ser classificada como heterogeneidade Homem-Robô e equipa de robôs heterogéneos.

A interação Homem-Robô (HRI) pode estar presente num sistema onde o operador humano esteja a supervisionar ou a comandar uma equipa de vários robôs, como por exemplo num projeto já referido, a *framework* ALLIANCE proposta por Parker (1998). Neste projeto era incentivado o uso de uma equipa

de robôs heterogêneos, devido à vantagem das próprias diferenças existentes entre eles. Para tal foi necessária uma adaptação do modelo comportamental e dos conjuntos de comportamentos possíveis de entre as capacidades existentes entre os robôs.

Por outro lado, Howard et al. (2006) conceberam um programa para a implementação de um sistema autónomo multi-robô, capazes de explorar e mapear um ambiente interior de grandes dimensões, implementar uma rede de sensores neste ambiente e utilizá-la para localizar intrusos. Foi utilizada uma equipa heterogênea de aproximadamente oitenta robôs, num ambiente interior de seiscentos metros quadrados. Os robôs foram divididos em duas classes distintas, e, conseqüentemente, a missão também foi dividida em duas fases. Na primeira fase, exploração e mapeamento, a subequipa explorou o ambiente e criou um mapa de grelha onde representa a ocupação do ambiente. A exploração era coordenada e os robôs cartográficos eram capazes de se detetarem e identificarem entre si. Na segunda fase, também conhecida como implementação e deteção, o mapa de grelha que representa a ocupação era utilizado para calcular um conjunto de locais de implementação, e os robôs sensoriais simples são mobilizados para esses locais, seguindo os robôs mais capazes. Uma vez distribuídos, os robôs sensoriais colaboram de forma a criar uma rede de sensores distribuída que segue os intrusos com base na sua assinatura acústica.

#### *AMBIENTE ADVERSARIAL*

Em 1997, a RoboCup foi fundada a fim de promover a investigação sobre a robótica e a inteligência artificial. Desta forma, a competição redirecionou algumas investigações para o caso do MRS em ambientes adversos, como competições de futebol ou em campos de batalha. Weigel et al. (2002) propuseram uma abordagem inovadora que acompanha a bola e os jogadores adversários, coordenando, simultaneamente, a equipa de forma estratégica. Foi atribuído a cada robô uma função específica, que requer uma série de competências, garantindo que cada membro da equipa possa adotar um comportamento adequado à adversidade em questão. Por outro lado, o MRS também pode ser implementando em campos de batalha. Zhang & Wang (2007) utilizaram o algoritmo genético para dotar os robôs com o conhecimento de quais eram as regiões de defesa dos seus adversários, aprendendo que não podiam entrar nessas zonas pois poderiam ser “mortos”.

### 3. FUNDAMENTOS TEÓRICOS

Este capítulo está dividido em quatro secções. Inicialmente é feita uma breve definição e distinção do que é a representação do ambiente. Posteriormente, são abordados os métodos utilizados para o planeamento de caminhos que podem ser divididos em três tipos e que são estudados neste capítulo: métodos clássicos e algoritmos de busca recorrendo a métodos exatos ou métodos heurísticos.

#### 3.1. REPRESENTAÇÃO DO AMBIENTE

Para o planeamento de caminhos, o ambiente é tipicamente representado como uma grelha ou um gráfico. Numa representação em grelha, o ambiente é dividido num conjunto de células, onde cada célula representa um local específico. As células podem ser livres ou ocupadas, e o planeador tem como função encontrar um caminho válido através das células livres entre o ponto inicial e o ponto de destino. Numa representação gráfica, o ambiente é representado como um conjunto de nós e arestas, onde cada nó representa um local específico e cada aresta representa um caminho possível entre dois pontos do gráfico. O planeador tem de encontrar um caminho através do gráfico, percorrendo as arestas entre nós consecutivos. Tanto a representação em grelha como a representação gráfica têm as suas próprias vantagens e desvantagens, e a escolha da representação depende dos requisitos específicos do problema de planeamento de trajetórias em questão.

#### 3.2. MÉTODOS CLÁSSICOS

O planeamento de caminhos trata do problema de gerar percursos viáveis entre um ponto inicial e um dado conjunto de pontos de destino, evitando ao mesmo tempo obstáculos e bloqueios. Para tal, é necessário adotar estratégias para resolver este problema. Estas estratégias podem ser classificadas, de forma mais geral, em duas classes de algoritmos: métodos de planeamento combinados e métodos de planeamento baseados na amostragem (Lee et al., 2016; *Planning Algorithms - Steven M. LaValle - Google Livros*, n.d.).

##### 3.2.1. MÉTODO *ROADMAP*

O método *roadmap* é definido como uma abordagem que conecta o espaço livre, do ambiente em que o robô se encontra, através de um gráfico, isto é, a ligação entre os espaços livres é representada por um conjunto de curvas unidimensionais (Choset & Burdick, 2000). Este método pode ser desenvolvido recorrendo a diferentes algoritmos, como os gráficos de visibilidade para os caminhos mais curtos ou o diagrama de *Voronoi* para um caminho com o máximo de espaço livre.

## GRÁFICOS DE VISIBILIDADE

O método dos gráficos de visibilidade é amplamente utilizado para determinar o caminho mais curto, uma vez que se trata de um método bastante prático e intuitivo. O gráfico de visibilidade é definido num espaço de configuração poligonal bidimensional. Os nós do gráfico de visibilidade incluem o ponto de partida, o ponto de destino e todos os vértices dos obstáculos do espaço de configuração. As arestas de um gráfico deste tipo são segmentos de reta que resultam da união entre dois vértices consecutivos, sendo que este segmento de linha entre dois vértices não intersesta nenhum dos obstáculos (Kaluder et al., 2011).

Na Figura 3-1 encontra-se representado o espaço de configuração poligonal, com ponto de partida e ponto de destino definidos. Na Figura 3-2 está apresentado o caminho mais curto resultante da união dos nós do gráfico.

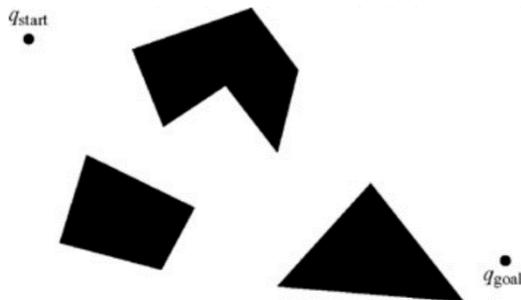


Figura 3-1 - Espaço de configuração poligonal com ponto de partida e ponto de destino representados. Retirado de (Mojtahedzade, 2011)

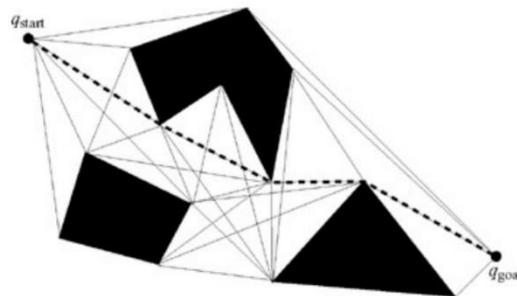


Figura 3-2 - Caminho mais curto representado a tracejado. Retirado de (Mojtahedzade, 2011)

## DIAGRAMA DE VORONOI

O método de *Voronoi* consiste na criação de um conjunto de pontos equidistantes entre os obstáculos presentes no ambiente. Num segundo passo, estes pontos são ligados através de linhas curvas que maximizem o espaço livre entre os obstáculos e esses pontos (Figura 3-3). Desta forma, é possível encontrar um caminho entre o ponto de partida e o ponto de destino. (Siegwart & Nourbakhsh, 2011)

No entanto, um caminho obtido diretamente através deste método não é ideal, uma vez que tem muitas curvas desnecessárias e o comprimento do caminho pode ser demasiado longo em regiões onde os obstáculos se encontram bastante afastados. (Bhattacharya & Gavrilova, 2007)

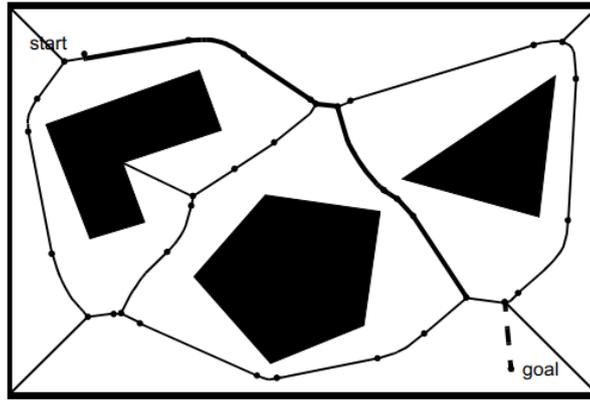


Figura 3-3 - Representação do caminho a negrito utilizando o Diagrama de Voronoi. Retirado de (Siegwart & Nourbakhsh, 2011)

### 3.2.2. MÉTODO DE DECOMPOSIÇÃO DE CÉLULAS

Este método consiste na partição do espaço livre do ambiente num conjunto finito de regiões que podem ser atravessadas em segurança pelo robô. É frequentemente empregue em abordagens de planeamento de alto nível onde o robô deve cumprir determinados requisitos lógicos ou temporais (Fainekos et al., 2009; Guo et al., 2014), em vez de métodos generalizados como gráficos de visibilidade ou diagramas de *Voronoi* que foram desenvolvidos para o planeamento clássico do movimento ponto a ponto.

O método de decomposição de células é tipicamente utilizado no planeamento de percursos geométricos, ignorando a otimização, como o comprimento do caminho, e a complexidade computacional, por exemplo, o tempo para dividir o ambiente num conjunto finito de regiões.

A abordagem padrão utilizada para resolver o problema consiste na decomposição, ou partição, do ambiente livre, ou seja, a área que não se encontra ocupada por obstáculos, num conjunto de polígonos (LaValle, 2006). Os polígonos obtidos devem ter a mesma forma e são denominados de células. Cada célula é representada por um nó no espaço livre e o caminho final é obtido através de um conjunto de nós adjacentes, ligados por arcos, entre o ponto inicial e o ponto final (Lingelbach, 2004). Podem ser utilizados diversos métodos no modo como se faz a decomposição das células, entre eles, o método exato, o método aproximado e métodos que decompõe o ambiente recursivamente através de *quadtrees*, por exemplo.

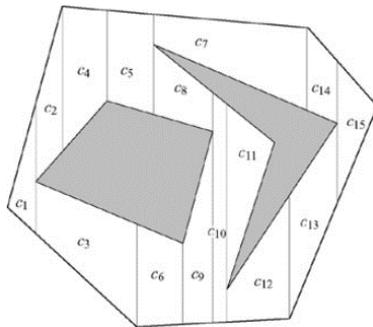


Figura 3-4 - Método exato de decomposição por células. Retirado de (Mojtahedzade, 2011)

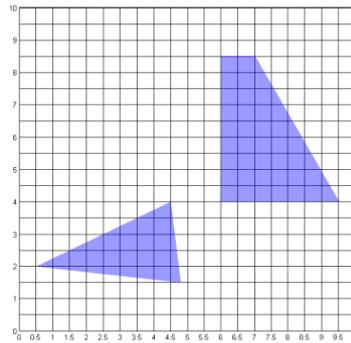


Figura 3-5 - Método aproximado de decomposição por células. Retirado de (Ahmad Abbadi & Vaclav Prenosil, 2015)

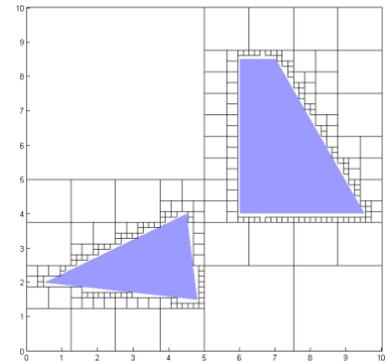


Figura 3-6 - Decomposição por células usando quadrees. Retirado de (Ahmad Abbadi & Vaclav Prenosil, 2015)

### MÉTODO EXATO DE DECOMPOSIÇÃO POR CÉLULAS

O método exato define os obstáculos como sendo poligonais e a partição do espaço livre resulta em células triangulares e trapezoidais. Esta partição é feita recorrendo a linhas verticais que unem os vértices dos obstáculos com o limite externo do espaço de trabalho ou com outros obstáculos, como possível observar na Figura 3-4.

### MÉTODO APROXIMADO DE DECOMPOSIÇÃO POR CÉLULAS

Este método, ao contrário do método exato, decompõe o espaço total do ambiente, no qual o robô se encontra inserido, num conjunto de células com uma determinada resolução e forma previamente definidas, como pode ser observado na Figura 3-5. As células que interseccionam obstáculos são consideradas como proibidas do ponto de vista do robô, e as restantes são definidas como células livres. A resolução das células é definida de acordo com o problema em questão. Quanto mais pequena for o tamanho das células, melhor a precisão. No entanto requer uma elevada capacidade computacional, tanto ao nível de memória como de tempo de processamento. Por outro lado, se for utilizado um tamanho muito grande para as células, a precisão diminui e, mesmo existindo, pode tornar impossível a tarefa de encontrar um caminho entre dois pontos. (Ahmad Abbadi & Vaclav Prenosil, 2015)

### DECOMPOSIÇÃO POR CÉLULAS USANDO QUADTREES

O método de decomposição por células usando *quadrees* utiliza células de tamanho variável que são decompostas sucessivamente e de forma recursiva em quatro células-filhas. Podem ser utilizados dois critérios de paragem desta função recursiva; ou a célula encontra-se numa zona totalmente ocupada ou livre; ou, então, a resolução limite da célula é atingida. Se as quatro células-filhas de uma dada célula estiverem totalmente livres ou ocupadas, então a célula-mãe também pode ser considerada

como livre ou ocupada, respetivamente. Este método de decomposição encontra-se demonstrado na Figura 3-6. (Ahmad Abbadi & Vaclav Prenosil, 2015)

### 3.2.3. MÉTODO DE CAMPOS POTENCIAIS

O método de campos potenciais pode ser aplicado para o planeamento de caminhos de robôs móveis, em tempo real e sem a ocorrência de colisões. Neste método, o robô é considerado como uma direção de partículas que agem perante a influência de um campo potencial, determinado pela soma das forças repulsivas e atrativa existentes (Equação 3-1). O ponto de destino é caracterizado pela sua força potencial atrativa, enquanto os obstáculos geram as forças potenciais repulsivas. Desta forma, o robô navega até ao ponto de destino, evitando os obstáculos, devido ao campo potencial que foi criado.

*Equação 3-1*

$$U(p) = \sum_{i=0}^n U_i(p)$$

Onde  $p$  representa um ponto no mapa,  $U_0$  a função potencial para o ponto de destino, e  $U_i$ , para  $i = [1, n]$ , representa as funções potenciais para cada um dos  $n$  obstáculos. (Khatib, 1985)

No entanto, quando as forças atrativas e repulsivas se anulam umas às outras ocorre um fenómeno denominado de armadilha dos mínimos locais. Este tipo de situações pode ocorrer quando o obstáculo se encontra exatamente entre o robô e o ponto de destino, por exemplo. Uma das soluções utilizadas para resolver este problema foi a criação de um obstáculo imaginário na região dos mínimos locais para afastar o veículo móvel. (Siddiqui, n.d.)

### 3.3. ALGORITMOS DE BUSCA – MÉTODOS EXATOS

#### ALGORITMO DIJKSTRA

O algoritmo Dijkstra é normalmente utilizado quando se pretende encontrar o caminho mais curto, em termos de distância, entre o nó inicial e o nó final. Para tal, numa fase inicial este algoritmo procura qual o caminho mais curto existente entre o nó inicial e os respetivos nós adjacentes. O nó terminal do caminho resultante é definido como sendo um nó intermédio. Desta forma, o algoritmo procura qual o caminho mais curto entre o nó intermédio e os respetivos nós adjacentes. Esta iteração é repetida até que seja obtido qual o caminho mais curto entre o nó de partida e o nó de destino, percorrendo todos os nós intermédios entre estes. Tal acontece, pois, é pressuposto que qualquer subcaminho do

caminho mais curto é o caminho mais curto desde o seu nó inicial até ao nó terminal, obtendo assim o caminho mais curto desde o nó de partida até qualquer nó de destino.

#### ALGORITMO DIJKSTRA MELHORADO

O Algoritmo de Dijkstra inicialmente proposto armazenava apenas um nó intermédio, resultando apenas num único caminho mais curto encontrado. No entanto, poderia existir mais que um caminho mais curto e, por isso, Qing et al. (2017) propuseram uma melhoria do Algoritmo de Dijkstra onde todos os nós intermediários, cuja distância ao nó de origem era a mesma, fossem reservados, procurando novamente a partir de todos os nós intermédios até definir o caminho ao nó de destino. Após múltiplas iterações, todos os caminhos mais curtos com a mesma distância teriam sido encontrados.

Para além disso, o Algoritmo tradicional de Dijkstra apenas utiliza uma busca em quatro ângulos para a expansão do nó num mapa de grelha, limitando o ângulo de direção do AGV até um certo ponto e aumentando o caminho redundante. Logo, Sun et al. (2021) propuseram uma melhoria no algoritmo já existente onde se baseia numa busca em oito ângulos de forma a reduzir a distância até ao ponto de destino, sendo um método mais adequado para ambientes de armazenamento complexos.

### 3.4. ALGORITMOS DE BUSCA – MÉTODOS HEURÍSTICOS

#### ALGORITMO A\*

Devido ao problema generalizado de encontrar um caminho através de um gráfico, Hart et al. (1968) propuseram uma abordagem heurística de modo a diminuir a complexidade temporal dos problemas, obtendo soluções de forma mais rápida.

O algoritmo A\* é um dos algoritmos mais utilizados quando se pretende planear o caminho mais curto entre um nó inicial e um nó final. Trata-se de um algoritmo bastante fiável pois utiliza uma combinação de pesquisa heurística e pesquisa baseada no caminho mais curto.

A fim de expandir o menor número possível de nós na procura de um caminho ótimo, um algoritmo de pesquisa deve tomar uma decisão tão informada quanto possível sobre qual o nó a expandir a seguir. Se expande nós que não podem estar num caminho ótimo, está a desperdiçar esforço. Por outro lado, se continuar a ignorar nós que possam estar num caminho ótimo, por vezes poderá não conseguir encontrar tal caminho e, portanto, não será admissível. Logo, um algoritmo eficiente necessita de ter a capacidade de avaliar de alguma forma os nós disponíveis para determinar qual

deles deve ser o próximo a ser expandido. A função de avaliação é definida de tal modo que o nó disponível com menor valor de  $f$  seja o nó que deve ser expandido a seguir.

O algoritmo  $A^*$  utiliza a função de custo descrita na Equação 3-2.

*Equação 3-2*

$$f(n) = g(n) + h(n)$$

Onde,  $g(n)$  representa o custo exato do movimento entre um nó e um nó adjacente a esse nó e  $h(n)$  representa o custo heurístico estimado entre um nó e o nó de destino. Para definir o valor a atribuir ao custo heurístico,  $h(n)$ , a distância entre o nó atual e o nó final pode ser estimada de diferentes formas. Isto é, normalmente é utilizada uma de três funções:

**DISTÂNCIA DE MANHATTAN** – é normalmente utilizada quando apenas é permitido haver movimento em quatro direções, ou seja, apenas movimentos horizontais e movimentos verticais. Esta distância consiste no número total de células que é necessário percorrer para atingir a célula final, sendo definida de acordo com a Equação 3-3.

*Equação 3-3*

$$h(n) = \Delta x + \Delta y$$

**DISTÂNCIA DIAGONAL** – é normalmente utilizada quando é permitido o movimento em oito direções (movimentos horizontais, verticais e diagonais). A distância diagonal é definida de acordo com a Equação 3-4.

*Equação 3-4*

$$h(n) = \max(\Delta x + \Delta y)$$

**DISTÂNCIA EUCLIDIANA** – é utilizada quando o movimento pode ser realizado em qualquer direção. Trata-se da distância em linha reta entre o ponto atual e o ponto final (Equação 3-5).

*Equação 3-5*

$$h(n) = \sqrt{(\Delta x)^2 + (\Delta y)^2}$$

O algoritmo  $A^*$  utiliza duas listas de nós que é atualizada no decorrer da busca. Uma lista em aberto que contém os nós que ainda não foram verificados e uma lista fechada que é constituída pelos nós que já foram visitados. O algoritmo expande consecutivamente o nó da lista em aberto com o menor

custo e adiciona os nós vizinhos à lista em aberto, inserindo depois o nó analisado na lista com os nós fechados. O algoritmo termina quando o nó final é encontrado.

#### ALGORITMO A\* MELHORADO

Como verificado na Equação 3-2, o algoritmo A\* tem apenas em consideração o custo do movimento entre dois nós e o custo heurístico entre um nó e o nó final, não tendo em conta o número de pontos de inflexão no planeamento do caminho, podendo originar bastantes pontos de inflexão no caminho delineado. Consequentemente, não existe qualquer garantia que o caminho encontrado pelo algoritmo de busca A\* é definitivamente o caminho ótimo. Como tal, o Xu et al. (n.d.) introduziram a função de custo de avaliação do ângulo à função de custo do algoritmo A\* de modo que seja possível encontrar o caminho com o menor ponto de inflexão.

O algoritmo A\* melhorado, devido à adição da componente relativa à função de custo de avaliação do ângulo, tem como prioridade escolher o caminho com o menor ponto de inflexão no processo de busca, a fim de encontrar um caminho ótimo adequado ao movimento do AGV. Para além disso, o algoritmo convencional A\* necessita de muito tempo para avaliar o custo de um nó. Logo, foi utilizada uma estratégia de pesquisa denominada de *jump point* para realizar o pré-processamento dos nós pertencentes à lista em aberto do algoritmo A\*, resultando na redução do processo de avaliação de nós desnecessários e alcançando o objetivo de acelerar a pesquisa.

O pré-processamento dos nós que é necessário efetuar, e que foi anteriormente referido, deve-se reger da seguinte forma: utiliza a função de custo para encontrar o nó intermédio com o menor custo em torno do ponto inicial, sendo esse o próximo nó do planeamento do caminho, e procurar ao longo do nó inicial até ao nó intermédio. O princípio do *jump point* é encontrar o ponto de salto em dada direção, colocá-lo na lista em aberto para adicionar o último ponto de salto à lista fechada. Com o novo ponto de salto como nó atual, a pesquisa é repetida até que o ponto de destino seja encontrado, a pesquisa termina e o caminho é exibido. Por outro lado, caso o ponto alvo não seja encontrado e a lista em aberto esteja vazia, a pesquisa é abortada.

#### HÍBRIDO A\*

O algoritmo de busca híbrido A\* é baseado no algoritmo tradicional A\*, sendo conhecido como um dos planeadores de caminhos mais eficiente que existe perante os veículos autónomos não-holonómicos.

A fim de obter um caminho eficaz, este algoritmo foi dividido em duas fases: a pesquisa avançada e as expansões analíticas (Dolgov et al., 2010). Na pesquisa avançada, tal como o algoritmo tradicional A\*, o espaço de busca é discretizado e definido um mapa de custos, no qual apenas os centros das células são visitados e utilizados para a procura de nós vizinhos. Sendo assim, a função de custo do algoritmo híbrido A\* é composta por  $g(n)$ , que representa o custo desde o nó inicial até ao nó atual ( $n$ ) e pela função heurística  $h_1(n)$ , que utiliza o mapa de custos para calcular o custo da distância do nó atual ( $n$ ) até ao nó destino, no domínio discreto. No entanto, e o elemento diferenciador destes dois algoritmos, o algoritmo híbrido A\* associa a cada célula do mapa de custos um estado contínuo 3D do veículo. Ou seja, a função de custo do algoritmo híbrido é também composta pela função heurística  $h_2(n)$ , cuja função é considerar a cinemática de um robô não-holonómico. A cinemática do modelo do robô é um fator importante pois permite prever os movimentos que dependem da velocidade, ângulo de direção e outros parâmetros relevantes do robô no domínio contínuo.

Por outro lado, as expansões analíticas garantem que o algoritmo atinja a coordenada contínua exata do nó de destino. Nesta fase, foram utilizadas as expansões analíticas baseadas no modelo *Reed-Shepp*. Na primeira fase do algoritmo, foi encontrado um conjunto de nós, onde cada nó-pai possuía um conjunto de nós-filhos. Nesta segunda etapa, para além dos nós-filhos gerados, alguns nós geram um nó-filho adicional através do cálculo de um caminho ideal de *Reed-Shepp* desde o ponto atual até ao ponto de destino, assumindo um ambiente sem obstáculos. Posteriormente, este caminho é examinado, recorrendo ao mapa de custos, de modo a verificar se ocorre alguma colisão e o novo nó-filho só é adicionado à árvore se o caminho estiver livre de colisões. Por conseguinte, é possível auxiliar o algoritmo na escolha do nó sucessor mais adequado. A Equação 3-5 representa a função de custo deste planeador.

*Equação 3-6*

$$f(n) = g(n) + h_1(n) + h_2(n)$$

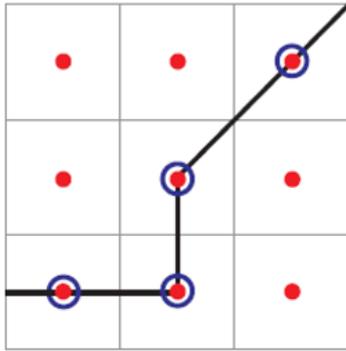


Figura 3-7 - Algoritmo A\*. Os custos são associados aos centros das células e apenas visita os estados que correspondem aos centros das células do grid map. Retirado de (Dolgov et al., 2010)

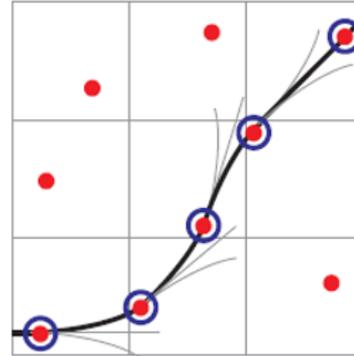


Figura 3-8 - Algoritmo Híbrido A\*. Associa um estado contínuo a cada célula. Retirado de (Dolgov et al., 2010)

Na Figura 3-7 e Figura 3-8 é possível observar a diferença entre as abordagens. Enquanto na primeira existe uma limitação onde apenas os centros das células do *grid map* podem ser visitados, na segunda, é utilizado um modelo cinemático contínuo que expande os nós, tornando as trajetórias produzidas dirigíveis. Foi comprovado que a expansão analítica da árvore de pesquisa, para além da elevada precisão que a caracterizava, também conduzia a ganhos notáveis na velocidade de replaneamento do caminho.

#### ALGORITMO A\* MELHORADO NO TEMPO

Devido ao aumento da competitividade em ambiente industrial, isto é, aumento da produtividade e redução de custos associados ao sistema logístico interno de modo a assegurar um fluxo de material eficiente, era necessário implementar algoritmos mais avançados de coordenação de frotas multi-robô. Estes algoritmos avançados visam assegurar o funcionamento contínuo do robô móvel, garantindo o tempo mínimo de execução de tarefas e a suavidade dos movimentos do veículo. Considerando que um determinado ponto no mapa só pode ser visitado uma vez por apenas um veículo num dado intervalo de tempo e baseando-se num sistema de escalonamento de tarefas e respetiva alocação a diferentes veículos, tendo em consideração os prazos de execução de cada tarefa e o estado do tráfego, Santos et al. (2021) propuseram um algoritmo denominado de *Time Enhanced A\** (TEA\*). Este algoritmo segue uma abordagem que integra tanto o planeamento de rotas como o escalonamento de tarefas e respetivo despacho para cada veículo. Como se trata de um método online, e de acordo com os movimentos do veículo e alterações em torno do mesmo, os caminhos são recalculados continuamente a fim de evitar colisões e garantir o funcionamento contínuo do sistema.

O elemento caracterizador deste algoritmo é a adição de uma componente temporal ao algoritmo tradicional A\* que gera rotas de forma eficiente, considerando que cada veículo móvel conhece as

posições dos outros robôs durante o tempo. A componente que foi acrescentada, o tempo, é representada através de camadas temporais. Cada gráfico temporal representa um conjunto de vértices livres e ocupados. Adicionalmente, este algoritmo determina a rota para cada AGV durante as camadas temporais, garantindo a identificação de colisões futuras e acabando por permitir que estas sejam evitadas com uma antecipação considerável.

Considerando um gráfico  $G$  com um conjunto de vértices  $V = \{0, 1, \dots, NUM_{VERTEXES}\}$  e arestas  $E = \{0, 1, \dots, NUM_{EDGES}\}$  (que são definidas como ligações entre os vértices), com uma representação do tempo  $[0, 1, \dots, k_{max}]$ . Cada veículo pode apenas arrancar e parar num vértice, sendo que cada vértice só pode ser ocupado por um único veículo em cada camada temporal. No processo de busca de caminhos para um único AGV, os vértices vizinhos são avaliados utilizando uma abordagem semelhante à do algoritmo  $A^*$ , adicionando o valor da função de custo para cada borda em cada camada temporal, denotado como  $f(j, k)$ , que resulta da soma de dois termos.

*Equação 3-7*

$$f(j, k) = \alpha g(j, k) + \beta h(j, k) \quad \text{onde } k \in [0, k_{max}], j \in [0, NUM_{VERTEXES}]$$

O primeiro termo da Equação 3-7,  $\alpha g(j, k)$ , representa a distância entre o vértice atual e o vértice de partida, na camada temporal  $k$ . O segundo termo,  $\beta h(j, k)$ , representa um custo heurístico que calcula a distância até ao vértice de destino. Os termos  $\alpha$  e  $\beta$  atribuem pesos diferentes à distância e à função heurística.

Cada vértice, em cada camada temporal, tem diferentes valores de  $g(j, k)$ , dada pela soma da distância entre o vértice atual e o vértice inicial (Equação 3-8), e  $h(j, k)$ , sendo a distância da aresta entre o vértice atual e o seu adjacente (Equação 3-9).

*Equação 3-8*

$$g(j, k) = dis(j, j_0, k) + dis(j, j + 1, k)$$

*Equação 3-9*

$$h(j, k) = dis(j, j_f, k)$$

No entanto, para que este algoritmo seja aplicado, é necessário que o *layout* do chão de fábrica seja modelado como um conjunto de vértices e arestas. Deste modo, cada AGV que navega neste gráfico, desde um nó para outro, o faça através de um conjunto pré-definido de arestas.

## 4. FERRAMENTAS

Este capítulo encontra-se dividido em duas secções. Numa primeira fase, é apresentada a ferramenta de simulação utilizada, assim como duas das *Toolboxes* disponibilizadas pela *MathWorks* importantes para este tema de dissertação. Posteriormente, é apresentado o cenário de simulação utilizado. Nesta secção é apresentada a abordagem utilizada para proceder à representação do sistema, quais as possíveis áreas de *charging* (posições iniciais) e *docking* (posições de destino ou estações de trabalho).

### 4.1. FERRAMENTA DE SIMULAÇÃO

A ferramenta de simulação utilizada foi o MATLAB. Este *software* combina um ambiente de trabalho sintonizado para processos de análise e desenho iterativos com uma linguagem de programação baseada em matriz. Trata-se de uma plataforma de programação concebida mais especificamente para engenheiros e cientistas, de modo a facilitar a análise de dados, desenvolver algoritmos e criar modelos e aplicações. Este *software* disponibiliza um conjunto de *Toolboxes* que pode ser definida como um pacote de funções e/ou classes, onde são fornecidas ferramentas para um determinado tópico, desde processamento de imagem a finanças, ou, mais especificamente e relativo ao tema deste projeto, navegação autónoma ou sistemas robóticos.

#### *NAVIGATION TOOLBOX*

A *Navigation Toolbox* fornece algoritmos e ferramentas de análise para o planeamento do movimento, localização e mapeamento simultâneos (SLAM) e navegação por inércia. Esta caixa de ferramentas inclui planeadores de caminhos baseados na pesquisa e na amostragem, e também um conjunto de métricas que podem ser utilizadas para validar e comparar os caminhos obtidos na fase de planeamento.

O delineamento do modelo da navegação autónoma para qualquer robô ou veículo tem por base o planeamento de caminhos, em conjunto com os sistemas de visão e controlo. Esta *Toolbox*, através do MATLAB e do Simulink, fornece ferramentas para o planeamento de percursos, uma vez que permite:

- Implementar algoritmos de planeamento de caminhos baseados na amostragem (RRT e RRT\*);
- Planear caminho utilizando mapas de grelha de ocupação, como por exemplo o Híbrido A\*;
- Gerar trajetórias locais para o replaneamento dinâmico em espaços interiores e mudança automática de faixa de rodagem;

- Comparar a validade e otimização do caminho utilizando métricas;
- Gerar pontos intermédios de passagem e enviar comandos de controlo para os seguir utilizando o controlador de perseguição puro;
- Implementar o algoritmo de planeamento de caminhos como um nó ROS ou código C/C++ numa plataforma integrada.

### *ROBOTICS SYSTEM TOOLBOX*

A *Robotics System Toolbox* fornece ferramentas e algoritmos para a conceção, simulação, teste e implementação de aplicações de manipuladores e de robôs móveis. Relativamente aos robôs móveis, esta caixa de ferramentas possui algoritmos de cartografia, localização, planeamento de trajetórias, seguimento de trajetórias e controlo de movimentos.

Para além de também ser possível construir cenários de teste e utilizar os exemplos de referência fornecidos para validar aplicações robóticas industriais comuns, possui uma biblioteca de modelos de robôs industriais disponíveis comercialmente que permite a importação, visualização, simulação e utilização nas aplicações pretendidas.

Uma das categorias desta *toolbox* é a *Mobile Robot Algorithm Design* que se concentra na robótica móvel ou em aplicações de veículos terrestres. São fornecidos algoritmos que auxiliam o fluxo de trabalhos da robótica móvel, desde o mapeamento até ao planeamento e controlo de trajetórias. Torna possível a criação de mapas através de grelhas de ocupação, desenvolver algoritmos de planeamento de caminhos para robôs num determinado ambiente e ajustar os controladores para definir um conjunto de pontos intermédios (*wayPoints*), determinados pelo planeador, para o robô seguir entre a sua posição inicial e a posição de destino. A perceção do espaço é realizada com base nos dados recolhidos pelos sensores lidar constituintes do robô em questão.

#### 4.2. CENÁRIO DE SIMULAÇÃO

Para representar o ambiente, foi utilizado o método aproximado de decomposição por células, ou seja, o mapa foi dividido em células de igual dimensão, recorrendo a uma matriz binária. Os espaços ocupados foram definidos com o valor aproximado de 0,999 e os espaços livres como sendo aproximadamente 0,001.

Foram utilizados dois cenários de simulação, baseando-se em espaços interiores onde ocorrem operações de logística. Foram definidos corredores de circulação comuns amplos, tal como é possível

observar na Figura 4-1 e Figura 4-2. Representados a laranja encontram-se os locais definidos como zonas de *docking*. Estas zonas podem, por exemplo, ser estações de trabalho ou locais de carregamento ou descarregamento de carga. Por outro lado, as zonas representadas a verde são os possíveis locais de partida dos robôs.

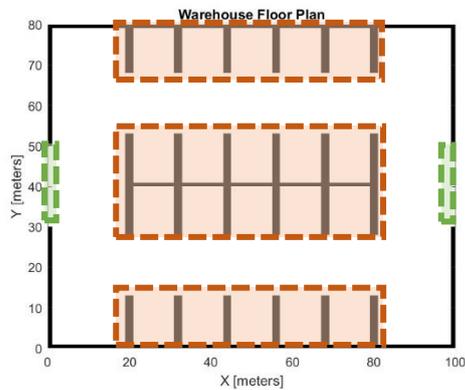


Figura 4-1 - Mapa do chão de fábrica 1

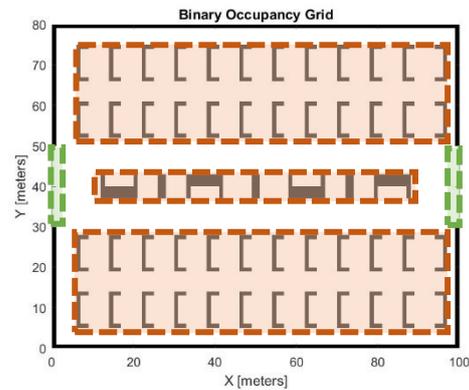


Figura 4-2 - Mapa do chão de fábrica 2

De modo a ser capaz de visualizar a posição atual e as ações do veículo robótico, este foi definido como sendo um polígono, tal como é possível observar na Figura 4-3. O vértice a vermelho representa a direção do movimento do robô. Como o veículo foi representado como sendo apenas um ponto unitário no mapa, as paredes do cenário necessitaram de ser inflacionadas. Desta forma, o planeamento do caminho tem em consideração o raio do robô, garantindo assim, que a rota gerada não causa nenhuma colisão com os obstáculos estáticos.

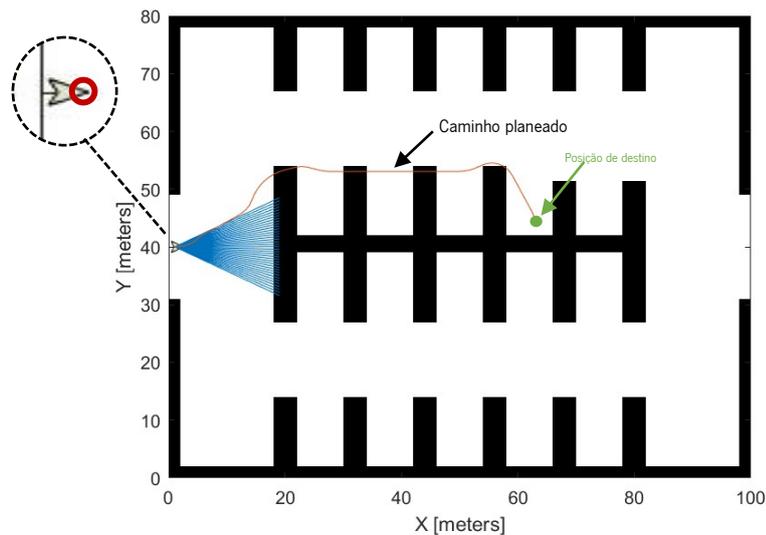


Figura 4-3 - Cenário com um robô e a respetiva rota definida

## 5. PROPOSTA DE SOLUÇÃO

O problema proposto consiste na coordenação de uma frota de robôs, em ambientes industriais, que tanto poderão ser ambientes internos como externos. Estes robôs realizam tarefas de transporte de matéria-prima entre estações de trabalho, sendo necessário encontrar entre elas um caminho otimizado. Para encontrar um caminho otimizado é necessário ter em consideração alguns fatores, como o tempo necessário que cada veículo necessita para completar o trajeto, a distância percorrida pelos robôs, a energia consumida, o tráfego existente, sendo que o objetivo final é alcançar a máxima eficiência e segurança na coordenação das frotas e no planeamento dos caminhos. Quando se está perante áreas de grande movimentação de pessoas e veículos, o tráfego é um dos principais fatores deve ser considerado no planeamento de caminhos otimizados para frotas multi-robô. Este capítulo encontra-se dividido em quatro secções. Inicialmente é realizada uma análise a alguns planeadores. A segunda secção diz respeito ao processo de definição das posições iniciais e finais. A terceira é relativa ao planeador utilizado para encontrar um caminho entre estas duas posições. Por fim, a quarta secção aborda o modo como foram tratados os conflitos entre os robôs da frota.

### 5.1. ANÁLISE

De modo a realizar uma breve análise a alguns dos planeadores disponibilizados pela *MathWorks*, foi utilizado o exemplo *plannerBenchmark* que é utilizado para comparar, através de parâmetros de referência, os diversos planeadores de caminhos 2D. Num determinado ambiente, e utilizando os mesmos pontos de partida e chegada, os planeadores especificados são executados, e são calculadas as seguintes métricas:

- *clearance*, que representa a distância mínima para obstáculos no ambiente;
- *executionTime*, tempo de execução do planeador em questão;
- *initializationTime*, tempo de inicialização do planeador a executar;
- *isPathValid*, se o caminho existe e é livre de colisões;
- *pathLength*, comprimento do caminho gerado;
- *smoothness*, suavidade do caminho para todas as poses.

Enquanto as métricas *executionTime* e *initializationTime* são calculadas durante a execução do planeador, as restantes métricas são calculadas a partir do caminho resultante após a execução do planeador.

O exemplo *Benchmark Path Planners for Differential Drive Robots in Warehouse Map*, presente na *Navigation Toolbox*, utiliza o objeto *plannerBenchmark* para demonstrar qual o melhor planeador de caminhos 2D para um robô de tração diferencial num ambiente de armazém a partir dos planeadores de caminhos disponíveis. Utilizando as métricas acima referidas, foram testados os seguintes planeadores: *Rapidly-exploring Random Tree* (*plannerRRT*), *Rapidly-exploring Random Tree Star* (*plannerRRTStar*), *Bidirectional Rapidly-exploring Random Tree* (*plannerBiRRT*), *Probabilistic Roadmap* (*plannerPRM*) e *Hybrid A\** (*plannerHybridAstar*), cujas posições de partida e destino foram escolhidas aleatoriamente.

### CENÁRIO 1

No cenário 1, o ambiente de simulação utilizado foi o que já estava associado ao exemplo mencionado disponível na *Navigation Toolbox*. Foram definidas duas áreas, a primeira como sendo a área de carregamento, ou também conhecida como posição inicial do robô, e a segunda como área de descarregamento, ou posição destino. Foram definidos três pares de poses aleatórias para proceder aos testes das métricas (Figura 5-1).



Figura 5-1 - Mapa do cenário 1 com três pares de posições iniciais e finais definidas.

Cada planeador foi executado vinte vezes para cada par de posições iniciais e finais, de modo a assegurar que os resultados da métrica são precisos, pois os planeadores baseados na amostragem (*plannerRRT*, *plannerRRTStar*, *plannerBiRRT* e *plannerPRM*) produzem resultados diferentes para o mesmo par de posições, tal como é possível observar entre a Figura 5-2 e Figura 5-4. Nestas figuras estão apenas ilustrados o caso para a primeira, décima e vigésima execução, sendo que os resultados obtidos nas restantes execuções se encontram em anexo (A1. Análise dos Planeadores - Cenário 1).

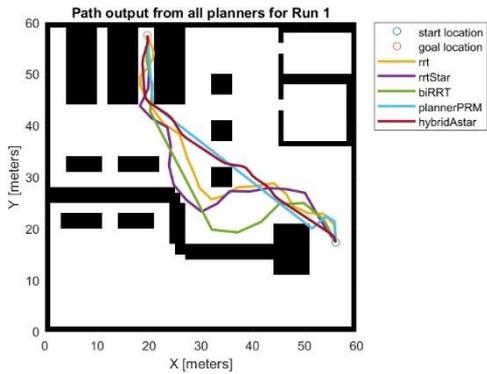


Figura 5-2 - Caminho gerado para todos os planeadores na primeira execução – Cenário 1

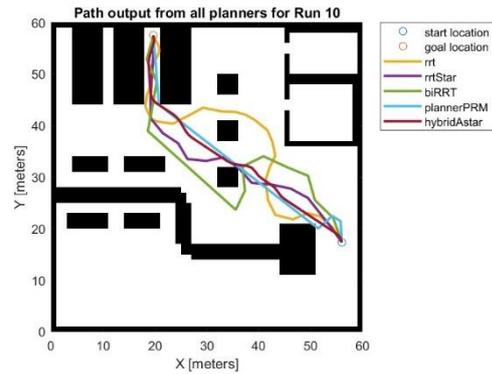


Figura 5-3 - Caminho gerado para todos os planeadores na décima execução – Cenário 1

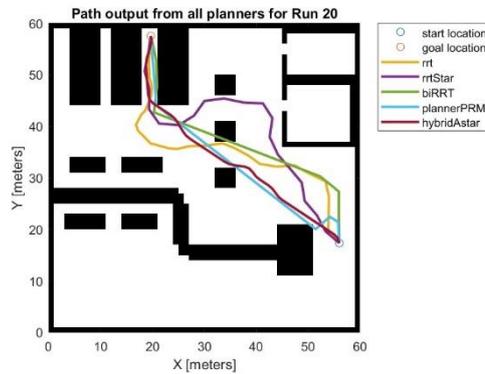


Figura 5-4 - Caminho gerado para todos os planeadores na vigésima execução – Cenário 1

Todos os planeadores foram executados vinte vezes para cada par de posições, ou seja, cada planeador foi executado sessenta vezes.

A métrica de desobstrução (*clearance*) representa a distância mínima do caminho em relação aos obstáculos no ambiente. Através do gráfico da Figura 5-5 conclui-se que o planeador *Probabilistic Roadmap* (PRM) é o que apresenta uma maior distância livre.

Na Figura 5-6 encontra-se representada a métrica que define a taxa de sucesso de cada planeador, expressa em percentagem. Para cada par de posições, todos os planeadores foram capazes de gerar um caminho válido.

A métrica tempo de execução representa o tempo que leva à função *plan* a executar, ou seja, o tempo que leva a encontrar o caminho mais curto e livre de obstáculos entre dois pontos. De acordo com o gráfico da Figura 5-7, o PRM foi o planeador que obteve um tempo de execução menor, seguido pelo BiRRT. Por outro lado, a métrica tempo de inicialização indica o tempo que é necessário para executar a função de inicialização de cada planeador. Desta forma, o tempo de execução total é conseguido através da soma do tempo de execução e o tempo de inicialização de cada planeador. O

planeador PRM foi o que teve um tempo de inicialização maior (Figura 5-8) e o BiRRT o que teve um menor tempo de execução total.

A métrica comprimento do caminho representa o comprimento do caminho gerado, e o planeador Híbrido A\* foi o que obteve um menor comprimento total (Figura 5-9). Por fim, a métrica suavidade é importante para garantir que o veículo siga uma trajetória sem oscilações indesejadas e com elevados raios de curvatura, sendo o BiRRT o planeador que produz uma rota mais suave (Figura 5-10).

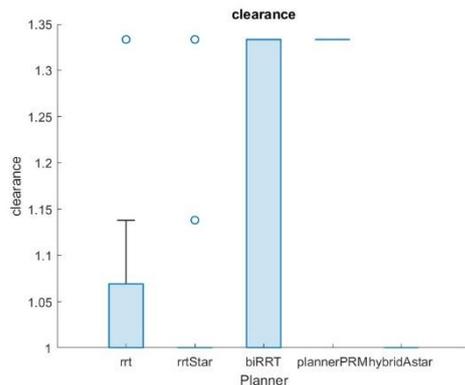


Figura 5-5 - Representação da média e desvio padrão da métrica distância aos obstáculos – Cenário 1

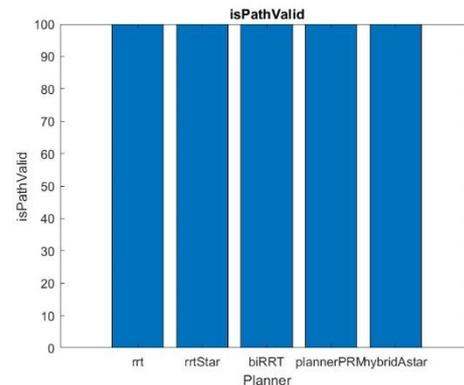


Figura 5-6 - Representação da validade do caminho gerado – Cenário 1

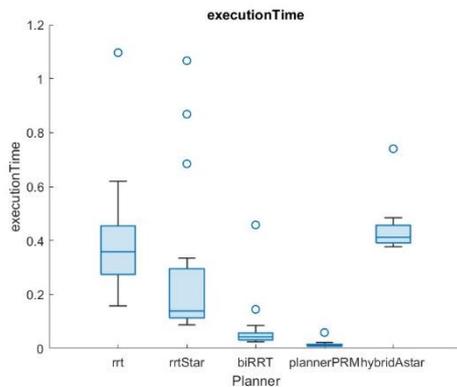


Figura 5-7 - Representação da média e desvio padrão da métrica tempo de execução – Cenário 1

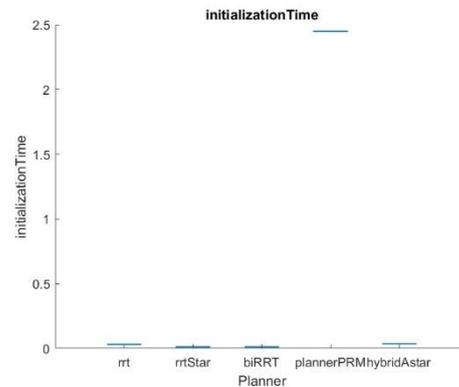


Figura 5-8 - Representação da média e desvio padrão da métrica tempo de inicialização – Cenário 1

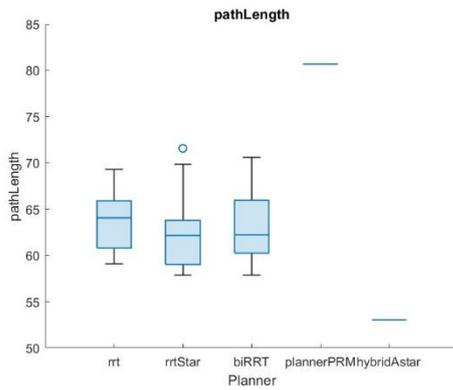


Figura 5-9 - Representação da média e desvio padrão da métrica comprimento do caminho – Cenário 1

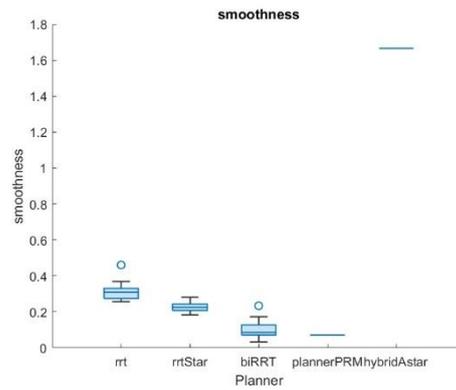


Figura 5-10 - Representação da média e desvio padrão da métrica suavidade do caminho – Cenário 1

## CENÁRIO 2

Num segundo cenário, aquele que foi escolhido para a simulação de coordenação de multi-robô em ambientes intralogísticos (Figura 5-11), foram tidas algumas das métricas anteriormente mencionadas para determinar qual o algoritmo de busca mais adequado a ser utilizado para o problema em questão. Cada planeador foi executado novamente vinte vezes para cada par de posições.

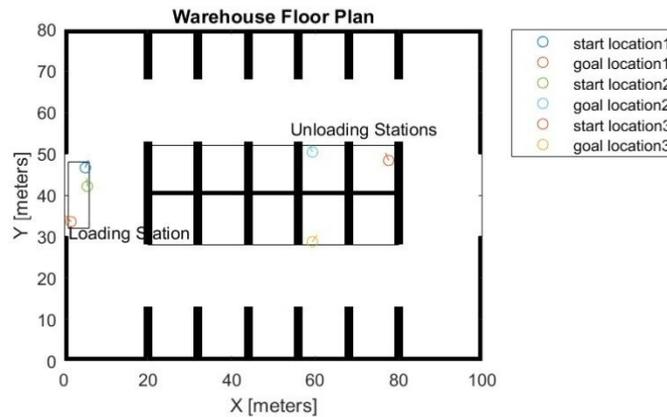


Figura 5-11 – Mapa do cenário 2 com três pares de posições e orientações iniciais e finais definidas.

Através da Figura 5-12, verifica-se que um dos algoritmos, mais concretamente o PRM, não teve sucesso na procura de um caminho num dos três pares de posições, sendo que os restantes planeadores conseguiram determinar uma rota para todos os pares de posições.

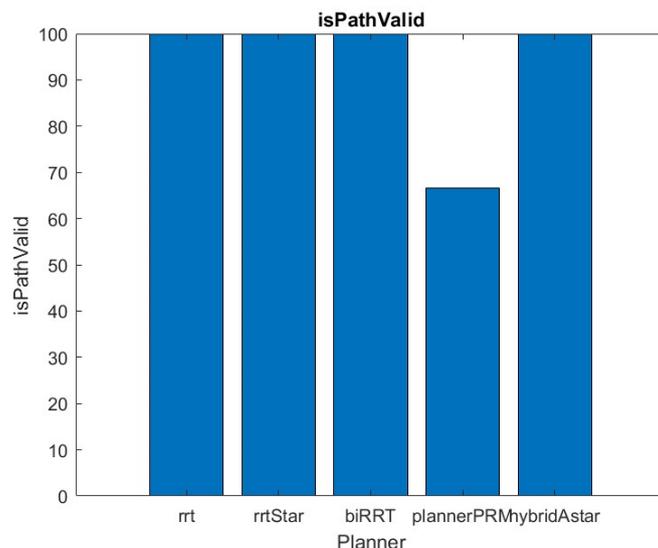


Figura 5-12 – Representação da validade do caminho gerado – Cenário 2, cada planeador executado 20 vezes

Na Figura 5-13 estão representados a média e o desvio padrão do comprimento do caminho consoante os diferentes planeadores, baseando-se nos resultados obtidos após cada um ser executado vinte vezes. Por outro lado, na Figura 5-14 apenas se encontra a média e o desvio padrão do comprimento do caminho para o par de posições cuja distância entre a posição inicial e final é maior. Tal como se pode verificar, não existem métricas correspondentes ao planeador PRM, podendo-se então concluir que o conjunto de posições para os quais não foi possível determinar um caminho utilizando este planeador foi o par de posições número um, representada posição de partida a azul e a posição de destino a vermelho na Figura 5-11 (os círculos a azul representam valores atípicos ou aberrantes (*outlier*) dos comprimentos dos trajetos encontrados). Relativamente ao algoritmo de busca PRM existem alguns problemas associados que podem justificar a falha na busca de um caminho válido entre o par de posições mencionado, tais como a escalabilidade, ou seja, pode requerer um elevado esforço computacional à medida que o número de obstáculos e o tamanho do ambiente aumenta, sendo difícil gerar um caminho para ambientes muito grandes ou complexos; mínimos locais, o algoritmo pode ficar preso numa solução mínima local, o que significa que o planeador encontrou um caminho que não é o mais curto nem o mais eficiente. Tal pode acontecer quando o ambiente não é suficientemente denso ou quando a configuração inicial do robô não é bem escolhida; sensibilidade às condições iniciais, o planeador é sensível às condições iniciais do robô, como a sua localização e orientação de partida, podendo dificultar a tarefa de procura de um caminho se estas não forem bem escolhidas; qualidade da solução, o caminho gerado pelo PRM é, geralmente, subótimo e pode ser mais longo do que o caminho mais curto; tempo computacional, uma vez que pode precisar de um

tempo significativo para gerar um caminho, especialmente para ambientes grandes e complexos; interpolação, este planeador não fornece qualquer meio para uma interpolação suave do caminho entre os nós do caminho, o que pode tornar o movimento do robô brusco; ou manutenção, quando o ambiente sofre alguma alteração é necessário atualizar o mapa de custos, o que pode requerer um elevado custo computacional e exigir um tempo significativo para tal.

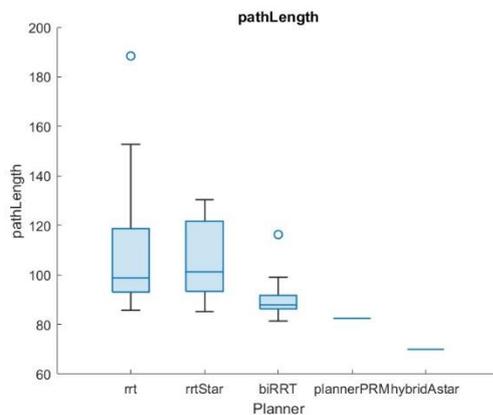


Figura 5-13 – Representação da média e desvio padrão da métrica comprimento do caminho – Cenário 2, cada planeador executado 20 vezes

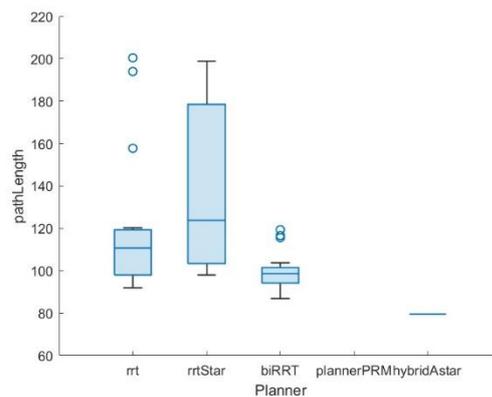


Figura 5-14 – Representação da média e desvio padrão da métrica comprimento do caminho para o par de posições cujo caminho é maior – Cenário 2, cada planeador executado 20 vezes

Entre a Figura 5-15 e a Figura 5-17 são demonstrados os caminhos gerados para o par de posições número um, utilizando os planeador RRT, RRT\*, RRT bidirecional e Híbrido A\*, uma vez que não foi possível encontrar com sucesso nenhum caminho para este par utilizando o planeador PRM. Tal como é possível observar nas figuras Figura 5-15, Figura 5-16 e Figura 5-17, a trajetória gerada pelo Híbrido A\* é semelhante em todas as execuções. Por outro lado, os restantes planeadores, geram caminhos diversificados. Como se pretende que o planeador seja robusto, e tal como verificado nos resultados obtidos, utilizando vinte amostras, o Híbrido A\* é a opção mais adequada para este projeto de dissertação, não só por gerar o caminho mais curto, quanto pela coerência dos resultados obtidos. Os resultados das restantes execuções encontram-se em anexo (A2. Análise dos Planeadores - Cenário 2).

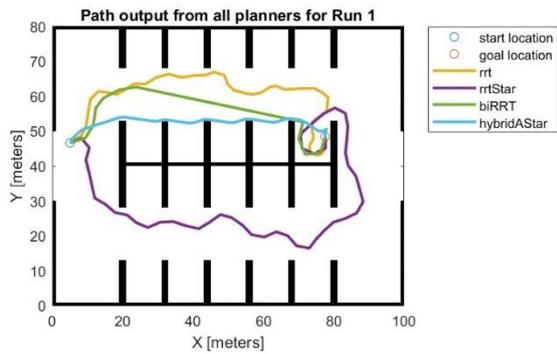


Figura 5-15 – Caminho gerado para todos os planeadores na primeira execução – Cenário 2

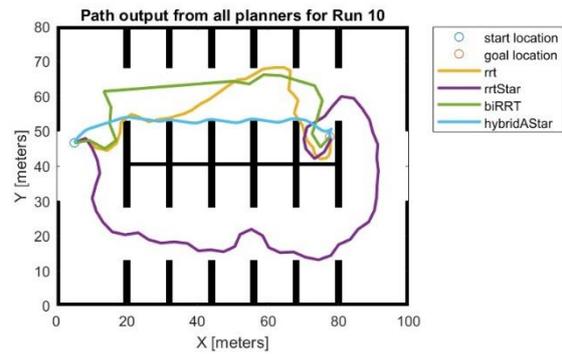


Figura 5-16 – Caminho gerado para todos os planeadores na décima execução – Cenário 2

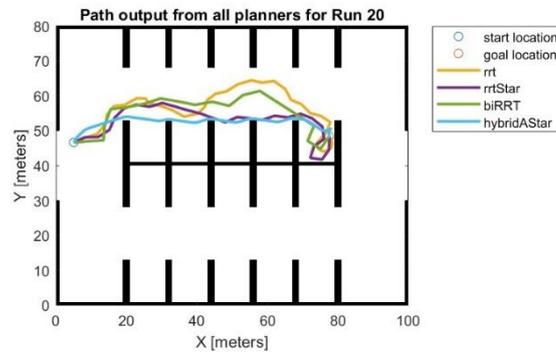


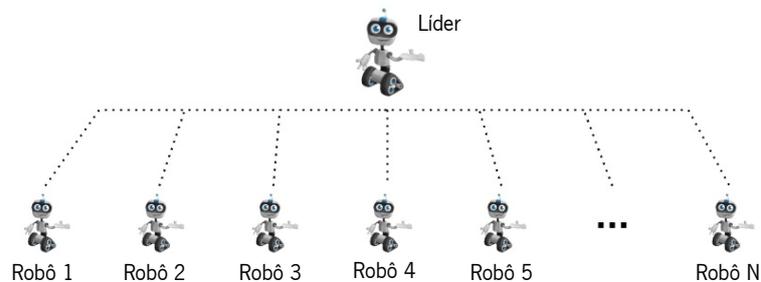
Figura 5-17 – Caminho gerado para todos os planeadores na vigésima execução – Cenário 2

Os testes dos planeadores acima realizados basearam-se no exemplo *Benchmark Path Planners for Differential Drive Robots in Warehouse Map* para determinar qual o algoritmo de busca mais adequado para esta dissertação. Com base no sucesso do planeador para os pares de posições iniciais e finais definidos e na métrica do comprimento do caminho, concluiu-se que o Híbrido A\* é o algoritmo mais adequado para o problema em questão. Para além de ter obtido sempre sucesso no planeamento de caminhos para os diferentes pares de posições, também foi o planeador que apresentou a média do comprimento do caminho calculado menor. Além disso, e tendo em conta o número de amostras utilizado, demonstrou ser um planeador robusto, uma vez que o caminho gerado entre dois pontos era idêntico em todas as execuções efetuadas.

## 5.2. ARQUITETURA DO SISTEMA

O sistema implementado baseia-se numa arquitetura centralizada, ou seja, existe um líder responsável por controlar todos os robôs da frota, tal como demonstrado na Figura 5-18. Este líder pode ser um dos robôs da frota ou um servidor estático. Cada membro da frota comunica com o

servidor central (líder) dando a conhecer a sua rota e o comprimento da mesma, assim como a sua posição atual. Para além disso, as leituras efetuadas pelos sensores de cada robô são comunicadas ao sistema central e transmitidas aos restantes robôs da frota, ou seja, a informação sensorial lida por um determinado veículo é partilhada, pelo líder, com os outros veículos.



*Figura 5-18 - Esquema de arquitetura centralizada*

Posteriormente, e consoante o critério escolhido, são definidos os níveis de prioridade para cada robô de forma a resolver os conflitos que possam surgir (a ser abordado em 5.4). O servidor central é responsável por atribuir uma tarefa, ou posição de destino, a cada veículo da frota. Adicionalmente, o líder tem como função averiguar as posições atuais e futuras de cada robô, e com esta informação verificar se existe algum conflito entre as rotas dos veículos da sua frota. Quando detetado um conflito, o líder informa os robôs em perigo de colisão qual o veículo que tem prioridade e deve prosseguir com a sua missão e quais os veículos que devem parar a marcha e aguardar que seja seguro continuar até à respetiva estação de trabalho.

Nos cenários utilizados foram definidas duas zonas de partida para a frota de robôs e vinte zonas consideradas como estações de trabalho, para o cenário 2 apresentado na Figura 5-19, e cinquenta e duas estações de trabalho, para o cenário 3 apresentado na Figura 5-20, sendo deste modo as posições destino possíveis para atribuir aos robôs. Por um lado, as coordenadas de partida são definidas de forma sequencial, ou seja, os robôs são colocados lado a lado nas zonas de partida. Tal como é possível observar na Figura 5-19 e Figura 5-20 onde estas zonas estão representadas a verde e de forma consecutiva, isto é, ao robô número um está associada a zona de carregamento/partida número um, e assim sucessivamente. Por outro, as zonas de descarregamento, representadas a vermelho na figura, podem ser atribuídas de um entre dois modos: ou são definidas sequencialmente, ou são atribuídas a cada robô de forma aleatória.

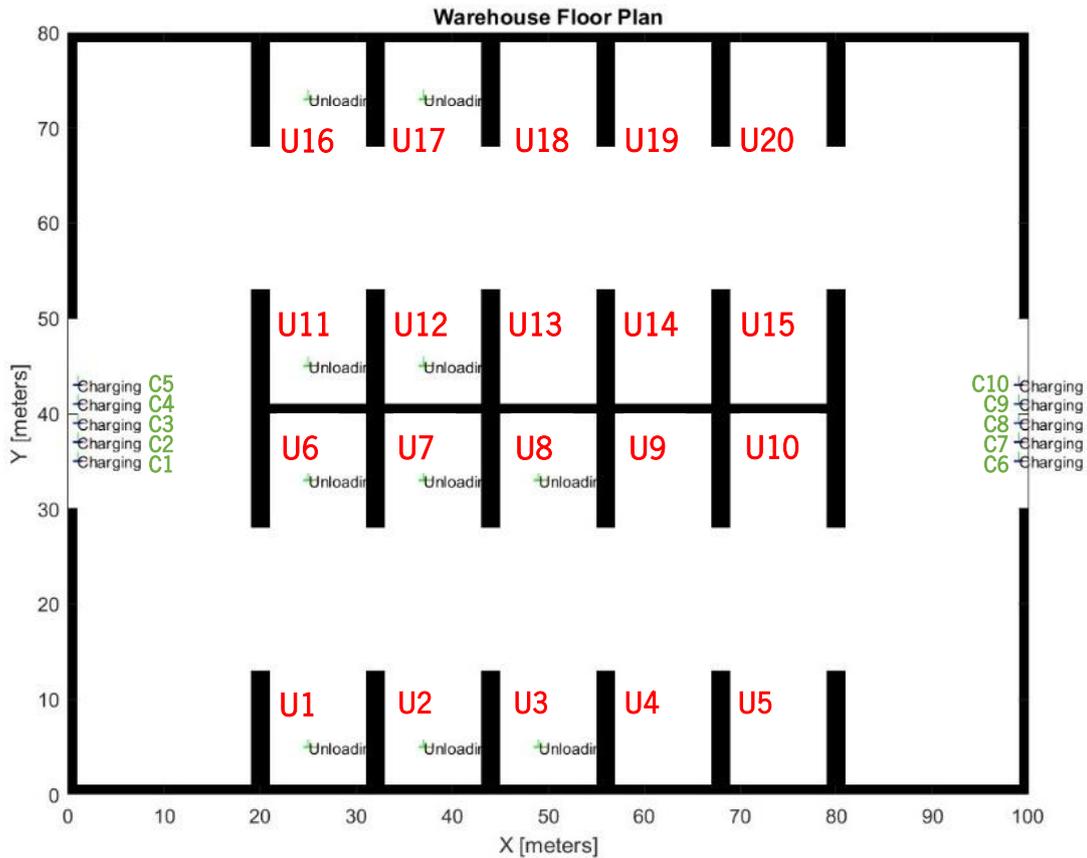


Figura 5-19 – Cenário 2: Posições de partida ou carregamento de baterias (charging) de cada robô, e posições de descarregamento de carga (unloading) ou estações de trabalho para cada robô

De forma a tornar o ambiente mais desafiador para a frota de robôs, foram adicionados dois robôs de patrulha ao cenário acima. Estes robôs de patrulha têm como função circular pelo corredor principal em tarefas de vigilância ou monitorização, mas são geridos independentemente da frota de modo a causar distúrbios nas rotas inicialmente planeadas para a frota, analisando como reage o sistema a tais perturbações. Deste modo, é possível representar, para a frota de robôs e o servidor central, a imprevisibilidade que ocorre em situações reais, tal como a presença de operadores humanos no espaço de trabalho. Sempre que um robô da frota entra em conflito com um robô de patrulha, o primeiro tem de ceder passagem, e só após o caminho estar livre é que o servidor central indica ao robô que pode avançar até à sua posição final.

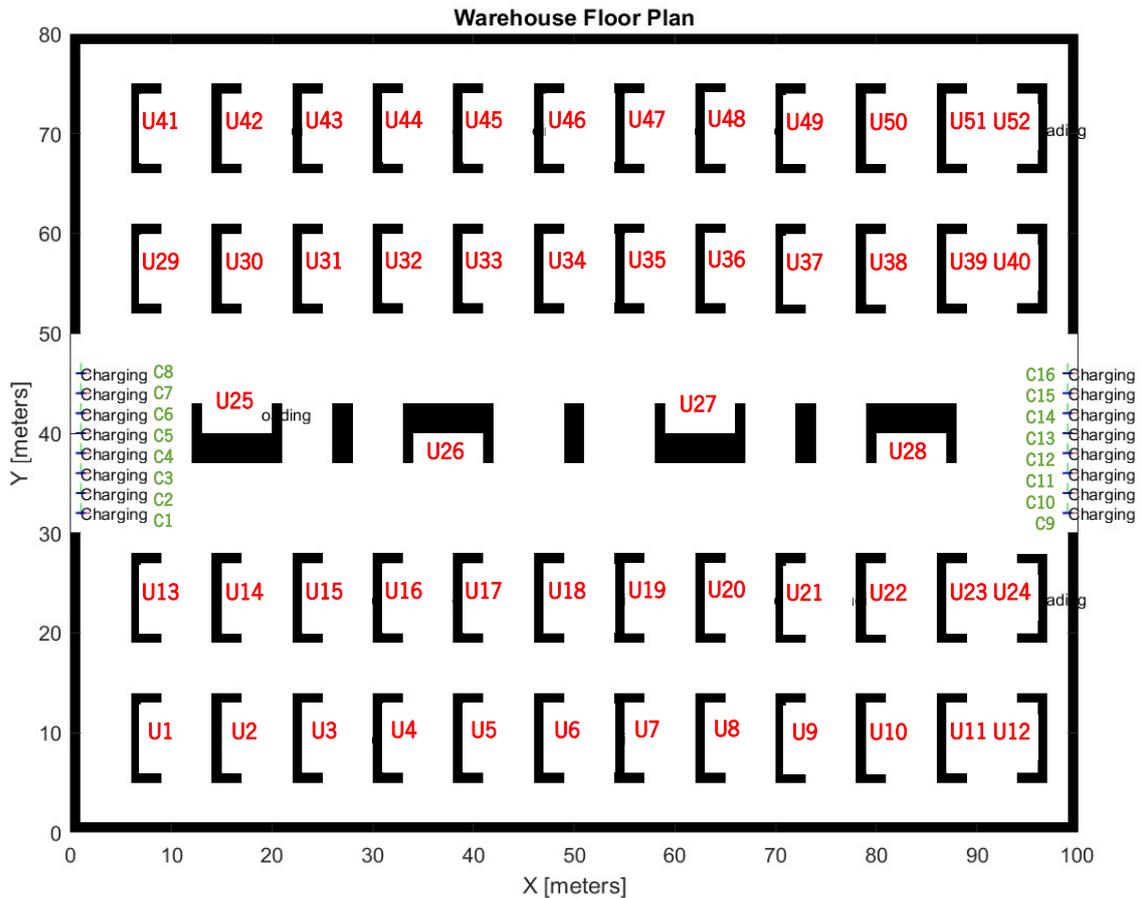


Figura 5-20 – Cenário 3: Posições de partida ou carregamento de baterias (charging) de cada robô, e posições de descarregamento de carga (unloading) ou estações de trabalho para cada robô

### 5.3. PLANEADOR GLOBAL

No capítulo dos Fundamentos Teóricos foram estudados alguns métodos passíveis de serem utilizados na procura de caminhos entre dois pontos. De entre os métodos que foram analisados, e após alguns testes realizados e analisados em 5.1, optou-se pelo algoritmo de busca Híbrido A\*.

Este algoritmo gera um caminho suave, num espaço de duas dimensões, para veículos com restrições não-holonómicas. Este planeador verifica a existência de colisões no mapa através da interpolação dos movimentos primitivos e da expansão analítica a partir de uma distância inicialmente definida.

Como já foi referido acima, o algoritmo Híbrido A\* foi dividido em duas fases. Numa primeira fase, funciona como sendo o algoritmo tradicional A\*, determinando um caminho entre uma dada posição inicial e uma posição final, que resulta num conjunto de pontos intermédios entre estas posições. A segunda fase deste algoritmo, trata das expansões analíticas baseadas no modelo *Reed-Shepp*. Este

modelo tem como função conectar duas poses consecutivas, resultando num segmento de caminho. Através da Figura 5-21, é perceptível que este segmento une as duas poses através de uma sequência de cinco movimentos (movimento reto, em frente ou para trás, curva à esquerda ou à direita na direção máxima ou nenhum movimento).

Por fim, este planeador verifica a existência de colisões no mapa através da interpolação dos movimentos primitivos e da expansão analítica com base numa distância de validação previamente definida. No planeamento de caminhos, a interpolação pode ser utilizada para estimar o custo de alcançar um ponto no ambiente, que não tenha sido explicitamente definido, com base no custo de alcançar os pontos vizinhos. Por outro lado, as expansões analíticas, num ambiente em grelha 2D, por exemplo, podem ser utilizadas para estimar o custo de alcançar um ponto com base na distância entre o ponto e obstáculo mais próximo. Desta forma, tanto a interpolação como as expansões analíticas podem ser utilizadas em combinação com a função heurística utilizada no algoritmo tradicional A\* para melhorar a precisão e eficiência do planeamento da trajetória. Com a utilização destes métodos, o algoritmo Híbrido A\* é capaz de atingir um tempo de convergência mais rápido e um desempenho melhor do que o algoritmo tradicional A\* em determinados tipos de ambientes onde a função heurística pode não fornecer uma estimativa precisa da verdadeira distância ao destino.

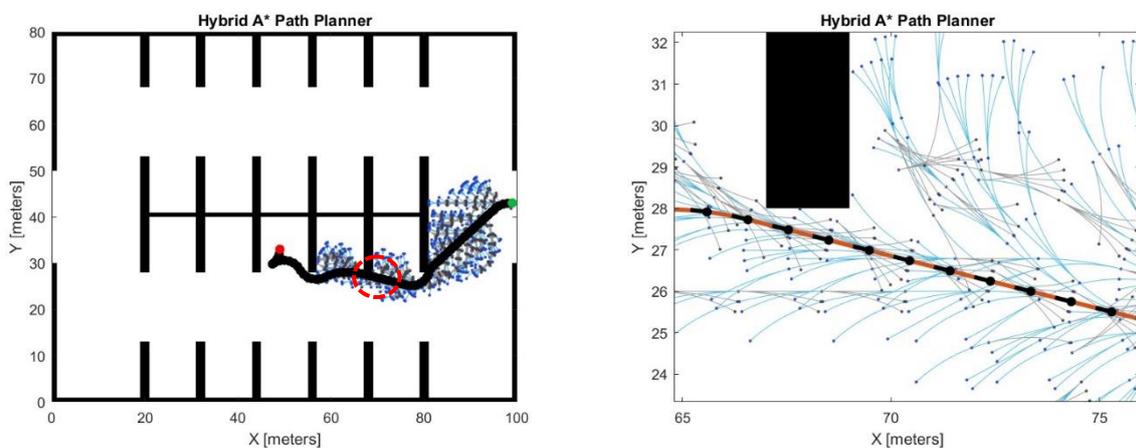


Figura 5-21 - Exemplo de uma rota calculado pelo planeador Híbrido A\* (imagem à esquerda) e da respetiva expansão analítica dos segmentos da zona a vermelho (imagem à direita).

#### 5.4. RESOLUÇÃO DE CONFLITOS

A rota inicialmente planeada pelo algoritmo, para um determinado robô, não tem em consideração as rotas dos restantes veículos. Logo, podem ocorrer inúmeros conflitos de rotas entre os robôs constituintes da frota que não foram tidos em conta no primeiro planeamento. Deste modo, é necessário implementar um sistema que permita a resolução destes conflitos.

Foi implementando um sistema de prioridades de forma a coordenar a frota multi-robô. Numa primeira fase, é implementada uma regra de trânsito, dando prioridade ao veículo que se encontra à direita. Caso o conflito detetado entre os veículos seja uma colisão frontal, então a prioridade é atribuída considerando um dos seguintes fatores:

- Pela posição de destino, ilustrando o caso em que existem tarefas mais prioritárias e urgentes que outras, definindo, assim, determinadas estações de trabalho mais prioritárias que outras. Na Figura 5-22, as estações de trabalho localizadas na área central e representadas a vermelho têm uma prioridade mais elevada em relação às posições de destino localizadas nas áreas laterais e representadas a verde. O mesmo acontece na Figura 5-23, onde as estações de trabalho localizadas nos cantos e representadas a verde têm menor prioridade em relação às estações representadas a vermelho.

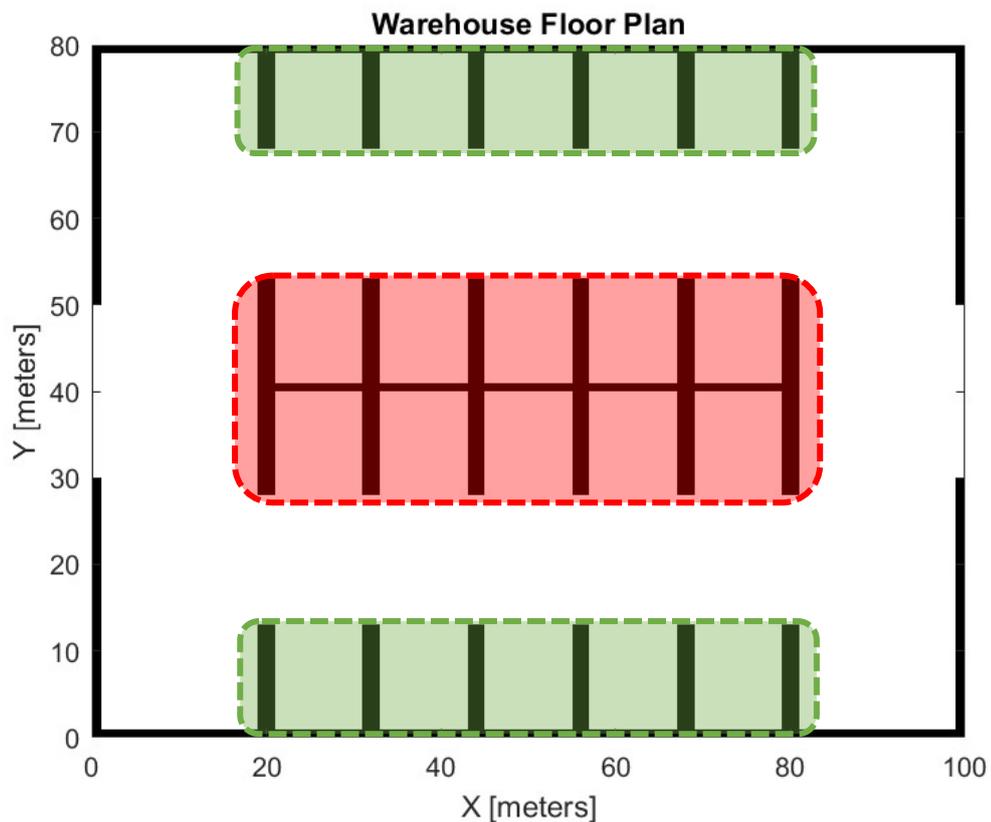


Figura 5-22 - Mapa do cenário 2 com as áreas das estações de trabalho com maior prioridade representadas a vermelho, e com menor prioridade representadas a verde

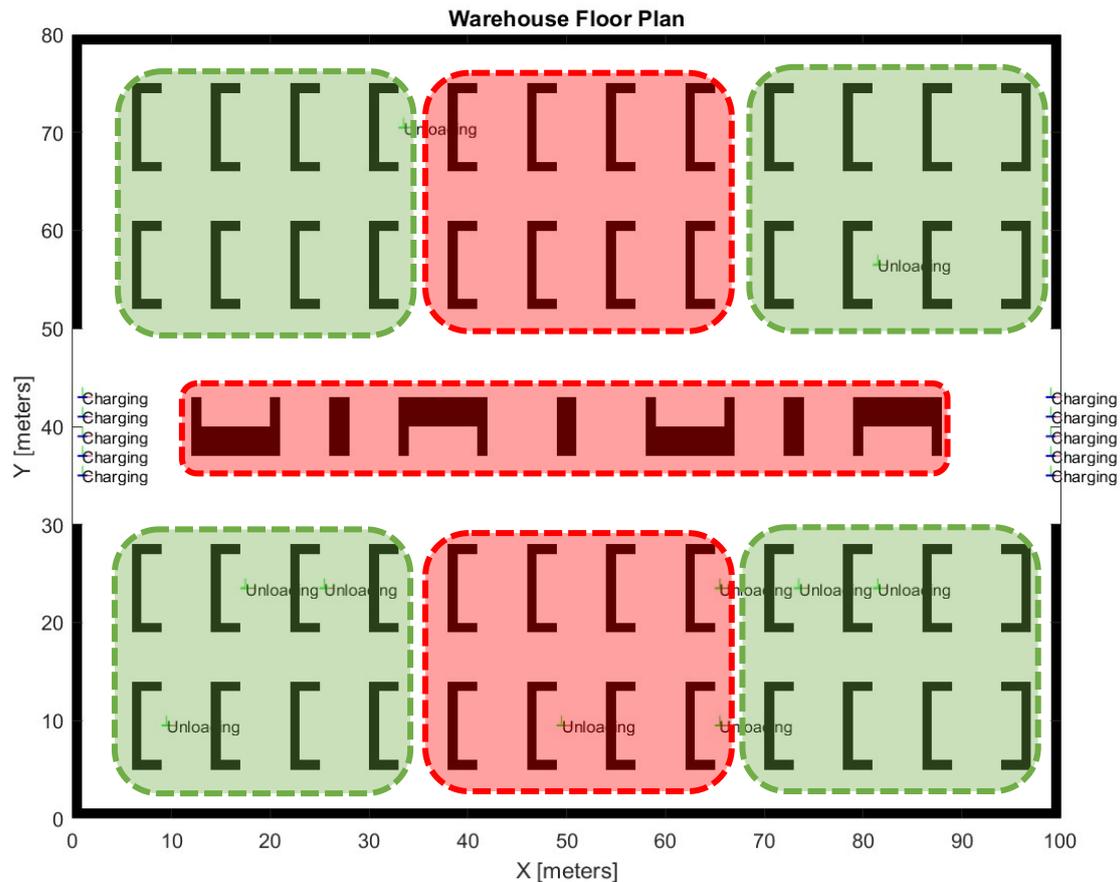


Figura 5-23 - Mapa do cenário 3 com as áreas das estações de trabalho com maior prioridade representadas a vermelho, e com menor prioridade representadas a verde

- Pelo comprimento do caminho, ou seja, a prioridade é atribuída consoante o comprimento da rota de cada robô. O robô cuja distância que tem a percorrer for maior será aquele que, conseqüentemente, também terá maior prioridade. Por sua vez, aquele que tiver de percorrer uma distância menor, terá menor prioridade na ocorrência de conflitos. A distância considerada é a distância total, ou seja, aquela que é calculada sempre que ocorre um planeamento, numa primeira fase, ou um replaneamento, numa fase posterior.
- Por fim, e caso não seja definido qual o modo de prioridade que o sistema se deve reger, ou, pelo menos dois robôs tenham o mesmo nível de prioridade, esta será definida aleatoriamente de modo a solucionar os conflitos que possam surgir.

Quando é detetado um conflito entre dois ou mais veículos, é verificado, pelo sistema central, qual deles tem maior prioridade. Inicialmente, é verificado qual o robô que se encontra mais à direita, dando prioridade ao mesmo para proceder com a sua tarefa e, simultaneamente, os restantes veículos envolvidos no conflito aguardam até que seja seguro prosseguir. No entanto, quando é detetada uma

colisão frontal, tal regra não é passível de ser aplicada. Deste modo, a resolução deste conflito rege-se ou pela distância que falta percorrer entre a posição atual e a posição de destino ou pela tarefa que foi atribuída a cada um dos envolvidos, sendo que existem tarefas (estações de trabalho) mais prioritárias que outras.

Como tal, foi definido um intervalo de poses que é utilizado para verificar o perigo de colisão existente entre as rotas dos robôs da frota. O intervalo de poses considerado é de igual dimensão para todos os robôs, onde são apenas tidas em conta a pose atual e as poses visitadas num futuro próximo, e caso exista alguma célula que seja visitada por mais que um robô no mesmo intervalo de instantes de tempo, é detetada uma colisão, acionando o gestor de conflitos. Como foram considerados intervalos de tempo discreto, a cada instante é visitada apenas uma pose para cada robô, uma vez que todos os robôs navegam à mesma velocidade.

## 6. TESTES E RESULTADOS

Neste capítulo encontram-se apresentados os testes realizados, assim como os respectivos resultados obtidos. Realizaram-se testes para o planeamento de rotas para um único robô e para múltiplos robôs. Foram utilizadas frotas de dois, cinco e dez robôs para o cenário 2 e frotas de quatro, dez e dezasseis robôs para o cenário 3. Por fim, foram analisados os resultados obtidos para os diferentes testes executados.

### 6.1. PLANEAMENTO DE CAMINHO APENAS PARA UM ÚNICO ROBÔ – CENÁRIO 2

Após a escolha do planeador a utilizar, realizaram-se testes para um único robô onde foi possível observar o replaneamento da rota quando introduzido, após a leitura do mapa fornecido, um obstáculo. Na Figura 6-1 encontra-se o planeamento inicial feito para um único robô entre duas posições inicialmente definidas, utilizando o algoritmo Híbrido A\*, e sem inflação do mapa, ou seja, o raio do robô não é considerado neste planeamento inicial. Na Figura 6-2 encontra-se representado o mapa após a introdução de um obstáculo. É de notar que o obstáculo foi introduzido após o primeiro planeamento, logo o robô não sabe da existência do mesmo.

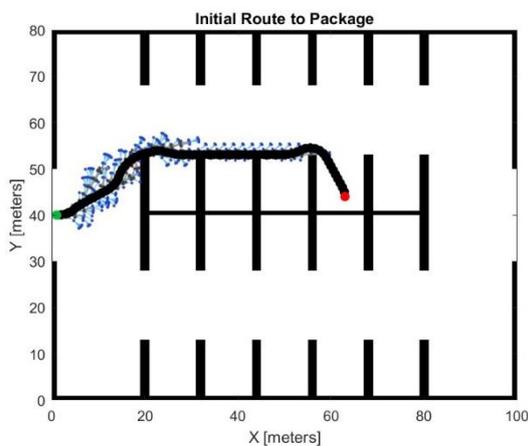


Figura 6-1 - Planeamento inicial, utilizando o algoritmo Híbrido A\*, entre a posição inicial e a posição destino inicialmente definidas

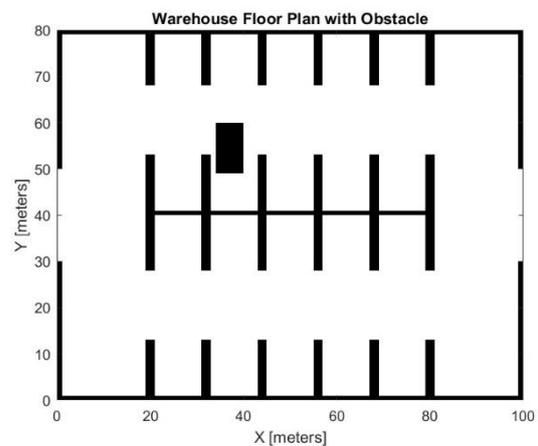


Figura 6-2 - Mapa fornecido com adição de um obstáculo

Na Figura 6-3 estão representados o caminho final percorrido pelo robô, pela linha a laranja contínua, e os trajetos obtidos pelos planeamentos anteriormente executados, a tracejado. Como o primeiro planeamento não teve em consideração o raio do robô, ou seja, a inflação do mapa, nem o obstáculo posteriormente adicionado, foi necessário replanear novas rotas para o robô conforme as leituras obtidas pelo sensor.

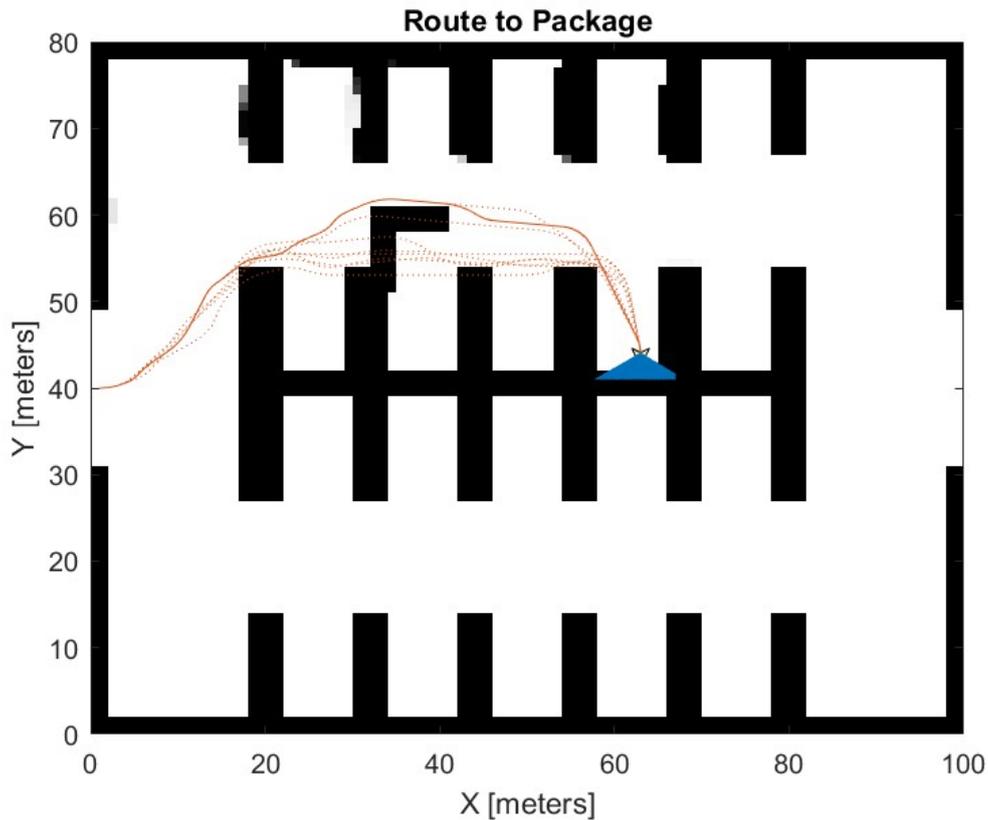


Figura 6-3 - Rota final do robô após vários replaneamentos devido ao obstáculo introduzido e à inflação do mapa

## 6.2. PLANEAMENTO DE CAMINHOS PARA MÚLTIPLOS ROBÔS – CENÁRIO 2

Posteriormente, foram introduzidos mais robôs no cenário de modo a verificar o comportamento do sistema face às múltiplas rotas calculadas e como este reagia às possíveis colisões e conflitos entre robôs.

A abordagem utilizada para garantir que as rotas calculadas para a frota de robôs não entrassem em colisão e conflito umas com as outras, baseou-se num sistema de atribuição de prioridades. No entanto, e de modo a otimizar esta abordagem, era somente verificado um pequeno traçado da rota, uma vez que, apenas tem interesse investigar os conflitos que pudessem ocorrer num futuro próximo. Foram realizadas algumas simulações utilizando diferentes tamanhos para o intervalo de valores empregue para averiguar a existência de conflitos. As medidas de desempenho que foram tidas em consideração para determinar qual o tamanho mais adequado para este intervalo foram o número de conflitos que ocorreram e o número de vezes que a rota necessitou de ser replaneada, utilizando uma frota de cinco robôs. Na Tabela 6-1 estão representados os resultados obtidos para os diferentes tamanhos do traçado que foram simulados. O número de conflitos indica o número de iterações que foi necessário recorrer ao sistema de prioridades para garantir que as rotas de, pelo menos, dois robôs

não cruzassem e não ocorressem colisões entre os veículos em questão. Por outras palavras, também se pode definir como o número de iterações que o veículo em questão teve de ceder prioridade ao robô com prioridade superior à sua. Quanto ao número de replaneamentos, houve sempre pelo menos uma vez que foi necessário replanear a rota inicialmente calculada para cada robô pois esta não teve em consideração o raio do robô. O replaneamento de uma rota para um dado robô é apenas despoletado quando detetado um obstáculo estático a interseção o último caminho planeado do veículo em questão. Analisando os valores obtidos, optou-se pelo tamanho do intervalo de valores ser de trinta.

*Tabela 6-1 - Número de conflitos e replaneamentos necessários para cada robô consoante o tamanho do traçado verificado para a coordenação multi-robô*

**Número de poses futuras consideradas para a verificação de conflitos entre as rotas dos robôs**

	20		30		40		50		60		70		80		100		120	
	Conflitos	Replaneamento																
<b>Robô 1</b>	533	2	487	2	131	2	0	5	151	2	144	2	163	4	154	2	160	2
<b>Robô 2</b>	0	1	0	1	388	1	501	1	388	1	335	1	277	1	354	1	365	1
<b>Robô 3</b>	76	1	0	1	0	1	0	1	0	1	1	1	0	1	2	1	5	1
<b>Robô 4</b>	0	2	65	2	16	2	16	2	16	2	0	1	0	2	510	1	530	1
<b>Robô 5</b>	0	6	0	1	0	1	0	5	0	1	0	6	0	10	0	5	0	6

O veículo move-se entre poses consecutivas, obtidas através da interpolação efetuada no planeamento ou replaneamento do caminho e cuja distância entre elas é de dez centímetros. Foi considerado um sistema de tempo discreto, podendo a velocidade variar entre apenas dois valores: velocidade nula ou velocidade de cruzeiro. O veículo tem velocidade nula quando se encontra parado, que tanto pode ocorrer quando o robô chega à posição de destino, completando a sua tarefa, como enquanto espera que algum conflito seja resolvido. Por outro lado, a velocidade de cruzeiro da frota é constante, uma vez que a distância entre poses consecutivas é igual e utiliza-se um sistema de tempo discreto, ou seja, tempo entre iterações é constante.

#### FROTA DE DOIS ROBÔS

Primeiramente, começou-se por simular uma frota de apenas dois robôs. Na Figura 6-4 e Figura 6-5 estão representados os caminhos obtidos inicialmente para os robôs um e dois, respetivamente, entre

as suas posições iniciais, ilustradas a verde, e finais, ilustradas a vermelho. Estes robôs partiram lado a lado e a sua posição de destino foi definida aleatoriamente.

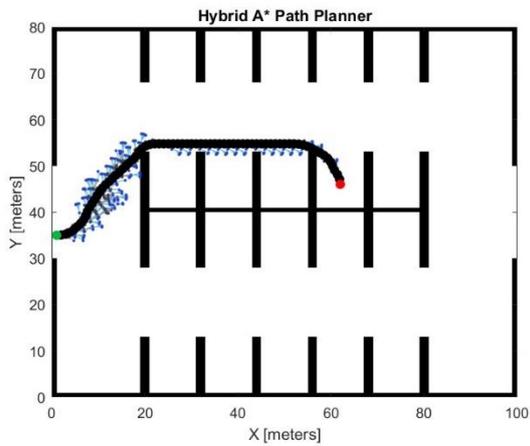


Figura 6-4 - Planeamento Inicial para o robô 1

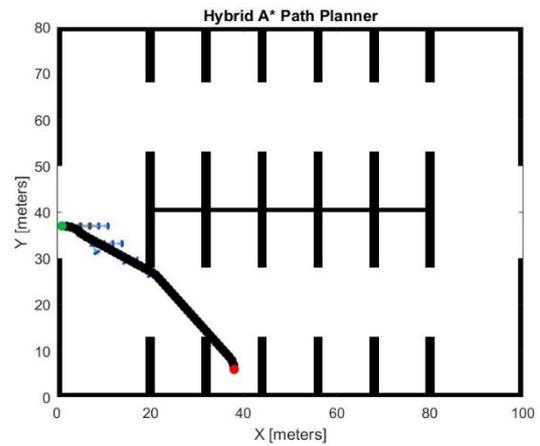


Figura 6-5 - Planeamento Inicial para o robô 2

Na Figura 6-6 encontram-se representados os caminhos efetivamente percorridos pelos robôs. Como é de notar, as rotas inicialmente traçadas sofreram alguns replaneamentos que se encontram representados a tracejado na figura abaixo. Estes replaneamentos devem-se às leituras efetuadas pelos sensores, sendo que estes detetaram a inflação aplicada às paredes do mapa de modo a considerar o raio do robô, e à introdução de um obstáculo que anteriormente não existia.

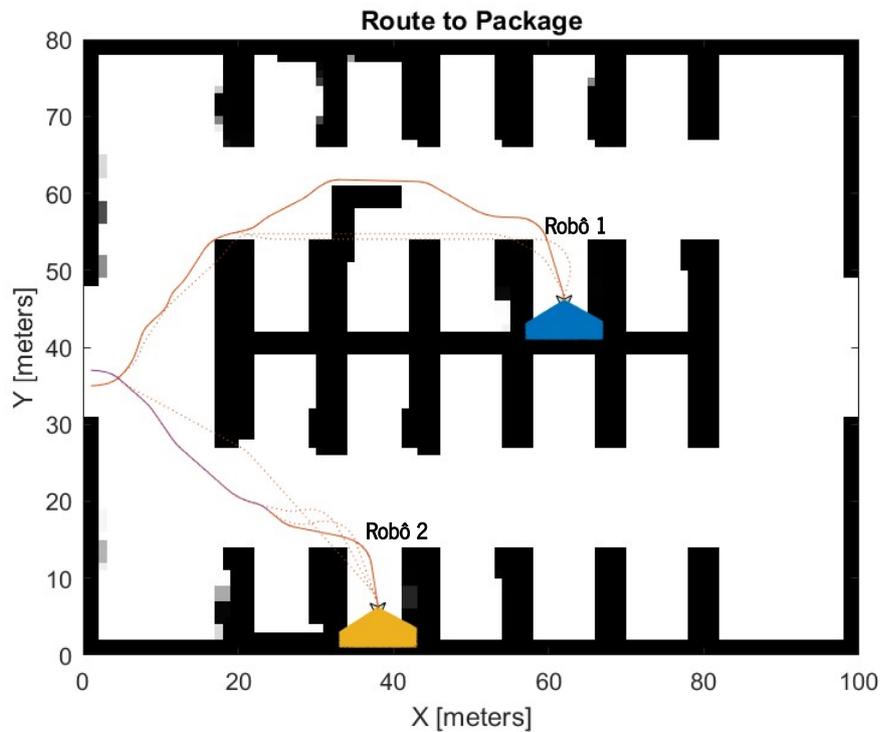


Figura 6-6 - Caminho percorrido pelos robôs até às respetivas posições de destino

Através da Tabela 6-2, verifica-se que em consequência dos replaneamentos ocorridos resultantes das leituras dos sensores, o comprimento do trajeto percorrido foi ligeiramente superior ao do inicialmente planeado. Para o robô um (azul), houve uma diferença maior entre os comprimentos em análise, devido ao obstáculo que foi introduzido no meio do percurso inicialmente traçado onde este obstáculo não existia, levando a um ajuste maior da rota, como se verifica na figura acima.

Tabela 6-2 - Comprimentos iniciais e finais do caminho calculado e do caminho percorrido, para uma frota de dois robôs

	Comprimento Inicial da Rota Calculada (metros)	Comprimento Final do Caminho Efetivamente Percorrido (metros)
<i>Robô 1</i>	74,3901	81,3964
<i>Robô 2</i>	49,6572	51,6341
<i>Média</i>	62,02365	66,51525

### FROTA DE CINCO ROBÔS

Aumentando a frota de robôs para cinco veículos, procedeu-se à simulação do sistema implementado de modo a observar o comportamento. Entre a Figura 6-7 e Figura 6-11 estão representadas as rotas inicialmente obtidas para os pares de posições iniciais e finais de cada robô.

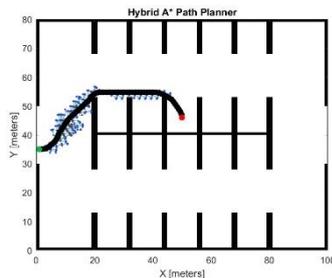


Figura 6-7 - Planeamento Inicial para o robô 1

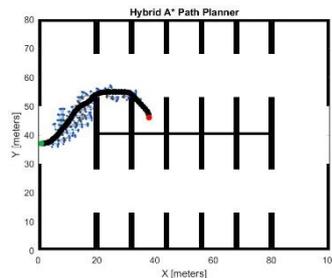


Figura 6-8 - Planeamento Inicial para o robô 2

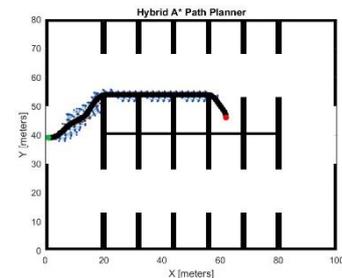


Figura 6-9 - Planeamento Inicial para o robô 3

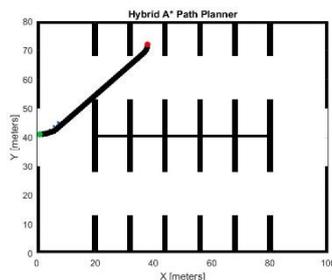


Figura 6-10 - Planeamento Inicial para o robô 4

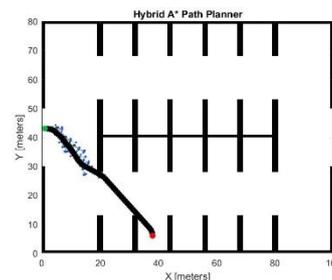


Figura 6-11 - Planeamento Inicial para o robô 5

A posição de destino atribuída ao segundo robô (representado a amarelo na Figura 6-12) torna-se inviável após a leitura dos sensores, uma vez que a entrada para esta estação de trabalho se encontra obstruída no instante em que o robô se aproxima da mesma. A tentativa de replaneamento da rota à

entrada desta área foi falhada, tendo de ser definida uma nova posição de destino para o robô seguir, tendo em conta o obstáculo que foi introduzido no mapa.

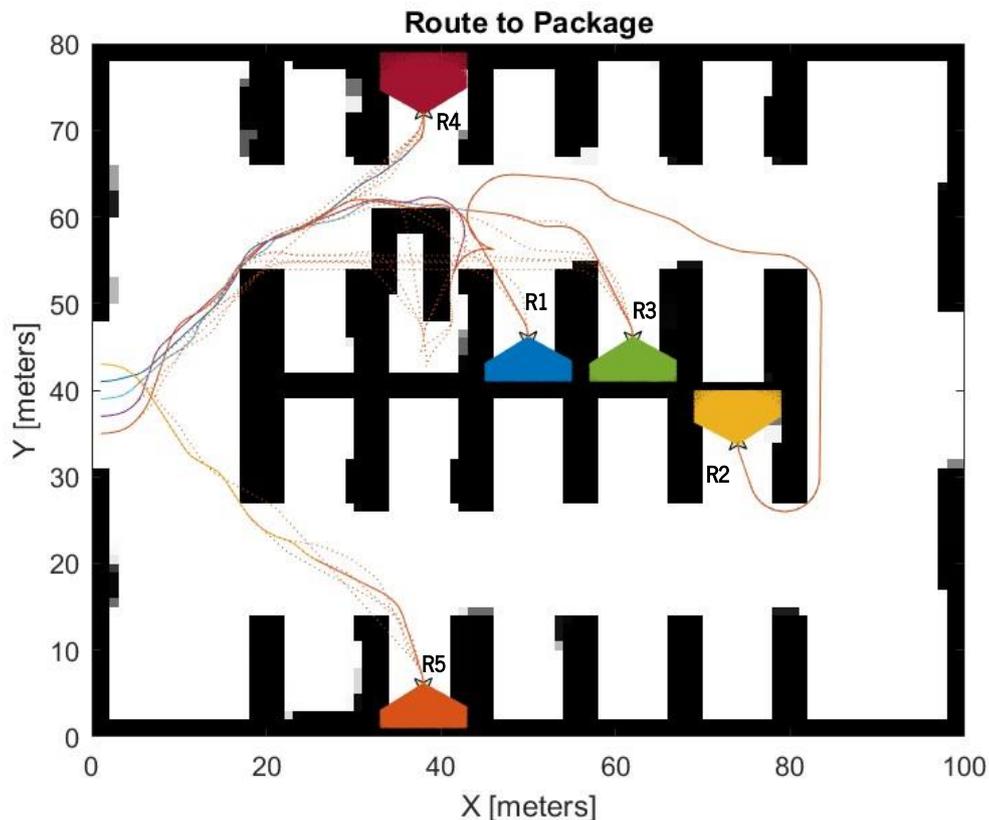


Figura 6-12 - Caminho percorrido pelos cinco robôs até às respetivas posições de destino

Tal como no caso anterior, os comprimentos das rotas efetivamente percorridas não variaram de forma significativa em relação ao comprimento da rota inicialmente planeada, à exceção do robô número dois. Como verificado na figura acima e corroborado pela Tabela 6-3, a distância total percorrida por este robô foi bastante superior à inicialmente calculada devido à reatribuição da posição de destino, pois a que lhe foi inicialmente conferida tornou-se inviável.

Tabela 6-3 - Comprimentos iniciais e finais do caminho calculado e do caminho percorrido, para uma frota de cinco robôs

	Comprimento Inicial da Rota Calculada (metros)	Comprimento Final do Caminho Efetivamente Percorrido (metros)
Robô 1	62,2990	70,9034
Robô 2	48,8684	156,3693
Robô 3	71,1037	77,9757
Robô 4	49,2893	51,8510
Robô 5	53,6384	55,0204
<b>Média</b>	<b>53,6384</b>	<b>70,9034</b>

## FROTA DE DEZ ROBÔS

Aumentando o número de robôs constituintes da frota para dez robôs verificou-se um aumento significativo no número de replaneamentos das rotas da frota em geral. Entre a Figura 6-13 e Figura 6-22 encontram-se representadas os caminhos inicialmente planejados para cada veículo.

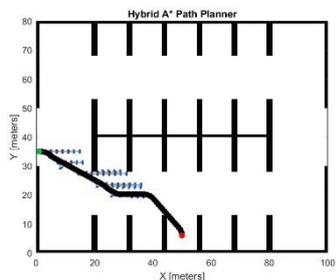


Figura 6-13 - Planejamento Inicial para o robô 1

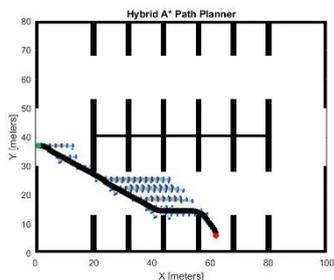


Figura 6-14 - Planejamento Inicial para o robô 2

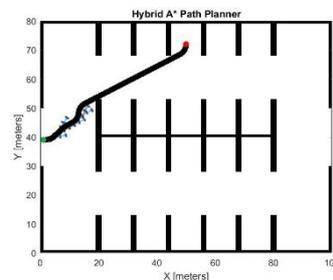


Figura 6-15 - Planejamento Inicial para o robô 3

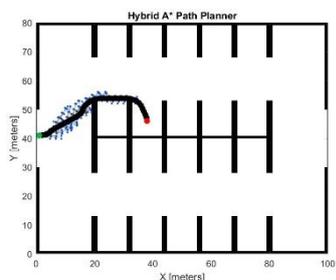


Figura 6-16 - Planejamento Inicial para o robô 4

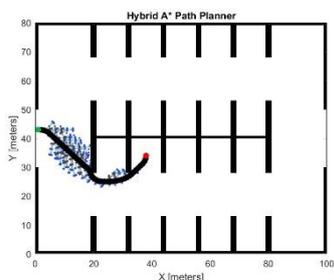


Figura 6-17 - Planejamento Inicial para o robô 5

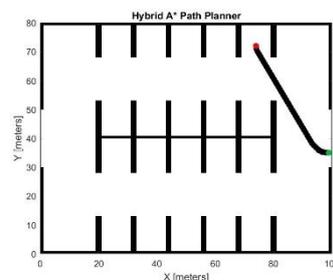


Figura 6-18 - Planejamento Inicial para o robô 6

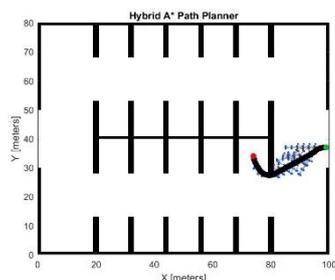


Figura 6-19 - Planejamento Inicial para o robô 7

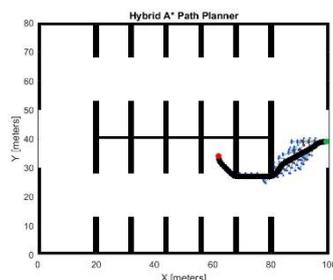


Figura 6-20 - Planejamento Inicial para o robô 8

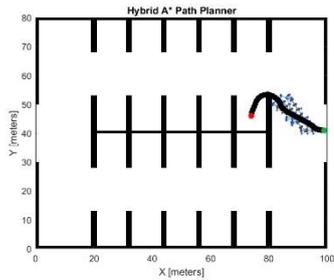


Figura 6-21 - Planeamento Inicial para o robô 9

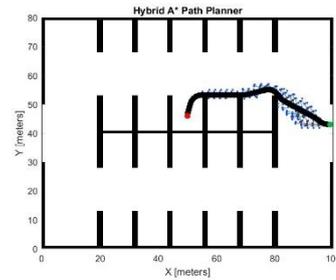


Figura 6-22 - Planeamento Inicial para o robô 10

Na Figura 6-23 estão representados os caminhos finais percorridos por cada robô. Tal como no caso anterior, a estação de trabalho com a entrada obstruída foi atribuída a um dos veículos da frota. Neste caso, foi atribuída ao robô número quatro.

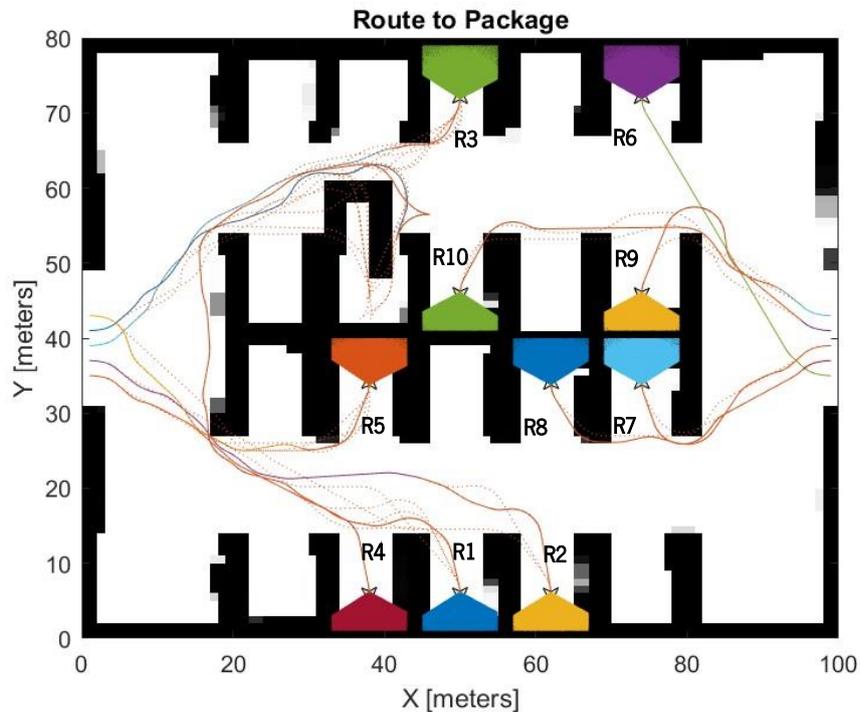


Figura 6-23 - Caminho percorrido pelos dez robôs até às respetivas posições de destino

Similar aos casos anteriores, e recorrendo ao auxílio da Tabela 6-4, a distância que cada robô percorreu não diferiu muito dos comprimentos dos trajetos inicialmente calculados. O caso do robô número quatro, onde se denota uma diferença mais significativa, deveu-se à obstrução detetada pelos sensores à entrada da estação de trabalho que lhe tinha sido atribuída, sendo necessária a atribuição de uma nova posição de destino. A distância percorrida por este veículo resultou da soma das distâncias entre a posição de partida e a primeira posição de destino atribuída com a distância entre esta última e a segunda posição de destino definida.

Tabela 6-4 - Comprimentos iniciais e finais do caminho calculado e do caminho percorrido, para uma frota de dez robôs

	Comprimento Inicial da Rota Calculada (metros)	Comprimento Final do Caminho Efetivamente Percorrido (metros)
Robô 1	56,8822	60,8950
Robô 2	71,3390	75,6461
Robô 3	61,1899	62,8304
Robô 4	45,9703	159,3472
Robô 5	48,2563	50,6291
Robô 6	45,8512	45,7446
Robô 7	31,2498	33,5341
Robô 8	44,5501	46,6401
Robô 9	33,9063	40,3328
Robô 10	57,6264	58,6957
<b>Média</b>	<b>47,1133</b>	<b>54,6624</b>

Utilizando um cenário com mais espaço ocupado nos corredores, e para as mesmas condições, realizaram-se novos testes. Este cenário encontra-se ilustrado na Figura 6-24 e ainda sem a inflação considerada para representar as dimensões do robô.

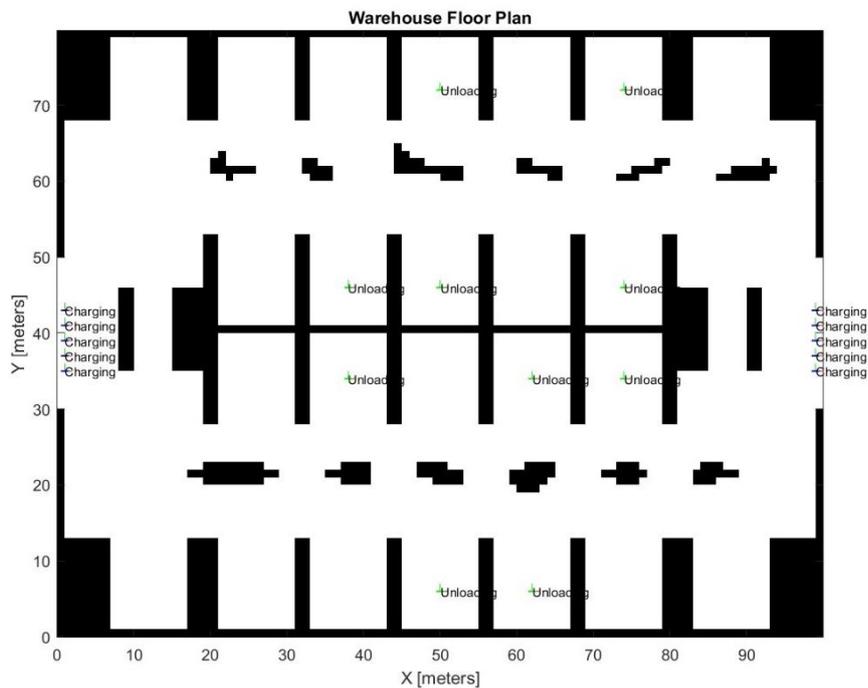
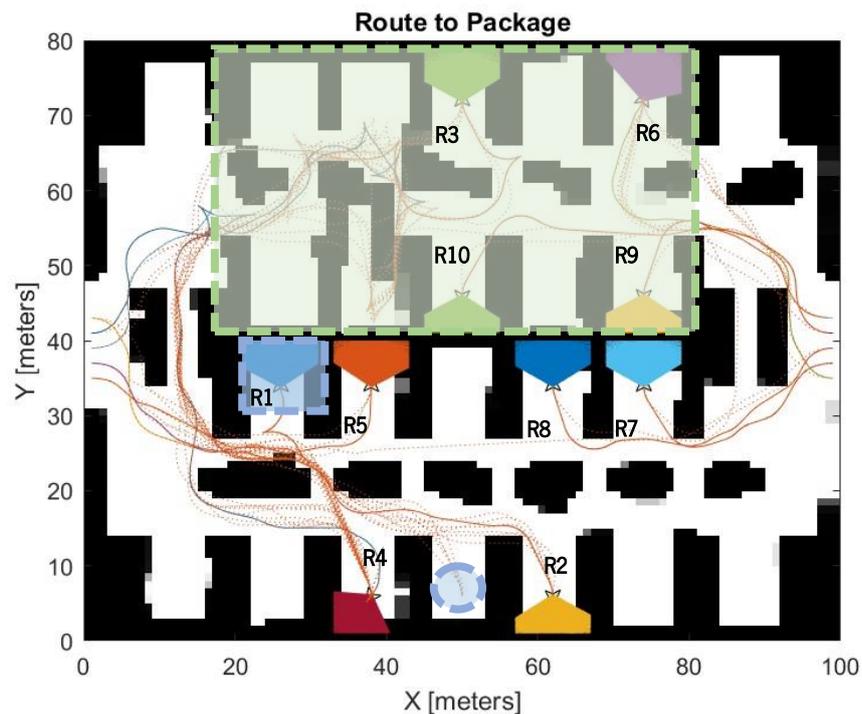


Figura 6-24 - Cenário 2 com corredores ocupados

Foram consideradas as mesmas posições de destino para cada robô que no caso anterior. No caso do robô número quatro, e como visto anteriormente, não foi possível encontrar um caminho válido até à posição de destino após a deteção de uma obstrução à entrada da respetiva estação de trabalho.

Logo, quando este obstáculo é detetado, é definida uma nova posição de trabalho para o respetivo veículo, sendo que foi considerada, novamente, a mesma estação de trabalho que no caso anterior.

Na Figura 6-25 estão representados os caminhos efetivamente percorridos por cada veículo. Ao contrário dos resultados anteriores e apesar de inicialmente ter tido sucesso na busca por um caminho entre a sua posição inicial e final, para esta nova configuração, o robô número um não foi capaz de concluir a sua tarefa, tendo-lhe sido atribuída uma nova estação de trabalho posteriormente.



*Figura 6-25 - Caminho percorrido pelos dez robôs até às respetivas posições de destino, para o novo cenário. Para o robô número um, a sua posição de destino inicial está representada com um círculo azul e a tracejado na figura, e a posição de destino que foi necessário reatribuir encontra-se representada no quadrado azul a tracejado.*

Apesar das médias dos comprimentos das rotas inicialmente traçadas e percorridas serem similares ao caso anterior, houve um aumento significativo no número de conflitos e replaneamentos ocorridos. De notar que neste cenário alguns dos robôs constituintes da frota, mais especificamente os robôs três e quatro, ou seja, aqueles cuja posição inicial se encontrava do lado direito e posição final se encontrava no retângulo verde da figura acima, apresentaram algumas dificuldades na navegação. Tal acontecimento se deve ao difícil acesso que se originou neste local, consequência da adição dos obstáculos no meio dos corredores que anteriormente estavam livres. Estas dificuldades na navegação traduziram-se em inúmeros replaneamentos, sendo que alguns deles originavam, momentaneamente, um trajeto mais longo e exaustivo para o veículo em questão.

Tabela 6-5 - Comprimentos iniciais e finais do caminho calculado e do caminho percorrido, para uma frota de dez robôs, para o novo cenário

	Comprimento Inicial da Rota Calculada (metros)	Comprimento Final do Caminho Efetivamente Percorrido (metros)
Robô 1	58,93858	46,7468
Robô 2	71,47812	74,99346
Robô 3	62,42353	105,3243
Robô 4	46,02249	234,7878
Robô 5	48,21368	54,74129
Robô 6	47,21719	56,95386
Robô 7	30,78469	33,76483
Robô 8	44,09152	47,66475
Robô 9	34,38308	39,35474
Robô 10	58,38115	60,81923
<b>Média</b>	<b>47,71544</b>	<b>55,84757</b>

### ROBÔS DE PATRULHA

Com o objetivo de analisar o comportamento do sistema quando detectado um outro veículo móvel que não faz parte da frota e não tem qualquer tipo de comunicação com o servidor central, foram adicionados dois robôs de patrulha. Na Figura 6-26 encontram-se representadas as posições iniciais de cada robô de patrulha e os pontos intermédios que definem a rota de cada um deles. Os dois veículos de patrulha navegam pela mesma rota, mas começam em posições distintas, seguindo de *via point* em *via point* por um tempo indefinido.

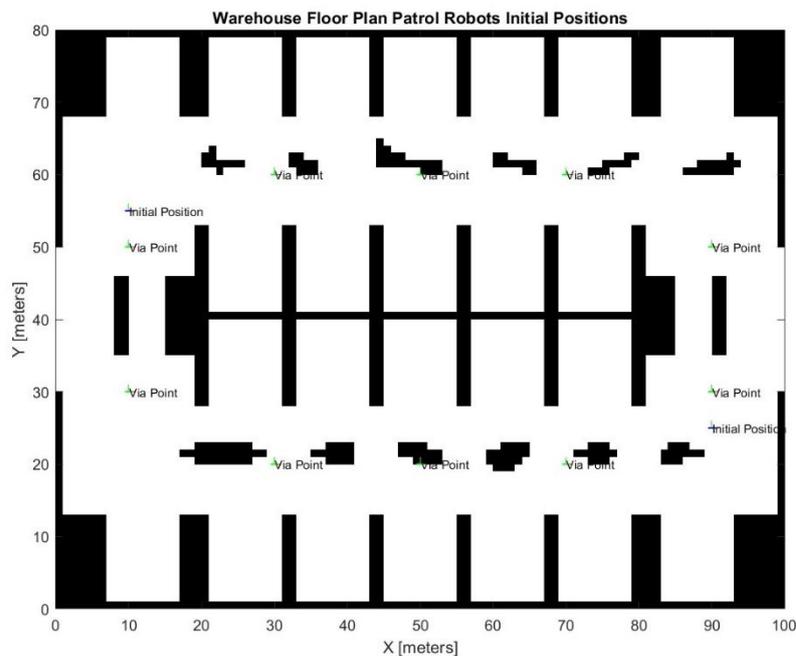


Figura 6-26 - Posições Iniciais dos robôs de patrulha e Pontos Intermédios que definem a rota de cada robô

Neste cenário com os dois robôs de patrulha, e apesar de inicialmente ter sido traçada uma rota válida entre as posições inicial e final, o robô número dez não foi capaz de alcançar o seu destino inicial. Desta forma, foi-lhe atribuída uma nova estação de trabalho representada no quadrado laranja na Figura 6-27, ou posição de destino, algo que não tinha acontecido nos casos anteriores. Ao contrário da última simulação analisada, onde era utilizado o mesmo cenário, mas sem a presença de robôs de patrulha, o robô número um foi capaz de concluir a sua tarefa, alcançando a posição de destino que inicialmente lhe tinha sido atribuída.

Analisando o comportamento da frota de robôs quando partilhando o espaço com robôs de patrulha, observou-se que o sistema se comportava de acordo com o esperado. Quando detetado um veículo de patrulha por um dos elementos da frota, o servidor central mandava o robô em questão parar o movimento até que fosse seguro continuar até à sua estação de trabalho sem ocorrência de colisões.

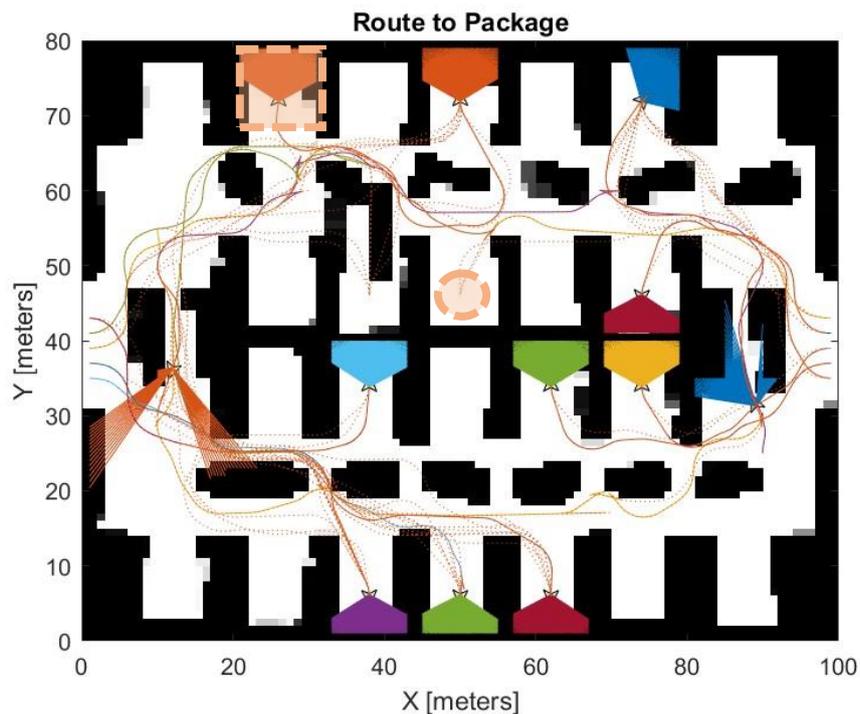


Figura 6-27 - Caminho percorrido pelos dez robôs até às respetivas posições de destino, para o novo cenário com dois robôs de patrulha. Para o robô número dez, a sua posição de destino inicial está representada com um círculo laranja e a tracejado na figura, e a posição de destino que foi necessário reatribuir encontra-se representada no quadrado laranja a tracejado.

Comparativamente ao caso anterior e conforme analisado na Tabela 6-6 a distância média efetivamente percorrida aumentou. Nesta situação o robô número um foi capaz de alcançar a posição de destino inicialmente definida, sendo que percorreu uma distância maior que no caso anterior, onde lhe tinha sido atribuída uma nova estação de trabalho mais perto da sua posição inicial, resultando

numa distância final menor. Por outro lado, enquanto nas condições anteriores o robô número dez realizou o trajeto até à sua posição de destino inicialmente definida, para este caso houve necessidade de uma reatribuição de estação de trabalho, originando numa distância percorrida maior. Os restantes robôs realizaram as mesmas ações que o caso anterior, sendo que apenas os robôs número três e quatro apresentaram maiores discrepâncias no comprimento final do caminho efetivamente percorrido quando comparado com os mesmos comprimentos nas condições da simulação anterior.

*Tabela 6-6 - Comprimentos iniciais e finais do caminho calculado e do caminho percorrido, para uma frota de dez robôs, para o novo cenário com dois robôs de patrulha*

	<b>Comprimento Inicial da Rota Calculada (metros)</b>	<b>Comprimento Final do Caminho Efetivamente Percorrido (metros)</b>
<i>Robô 1</i>	58,9386	64,5517
<i>Robô 2</i>	71,4781	76,4829
<i>Robô 3</i>	62,4235	91,6545
<i>Robô 4</i>	46,0225	160,834
<i>Robô 5</i>	48,2137	54,1158
<i>Robô 6</i>	47,2172	57,4901
<i>Robô 7</i>	30,7847	33,7648
<i>Robô 8</i>	44,0915	47,6638
<i>Robô 9</i>	34,3831	39,3548
<i>Robô 10</i>	58,3812	97,0043
<b>Média</b>	<b>47,71545</b>	<b>61,0209</b>

### 6.3. PLANEAMENTO DE CAMINHOS PARA MÚLTIPLOS ROBÔS – CENÁRIO 3

Numa última fase de testes, e para um cenário mais complexo (Figura 6-28), foram estudados os comportamentos de determinadas métricas (média da distância total percorrida em comparação com a média do comprimento do caminho inicialmente calculado, número total de conflitos e taxa de sucesso) aplicando diferentes regras de prioridade. Foram implementadas três regras de prioridade: cedência de passagem ao veículo que se encontra mais à direita – regra 1, prioridade mais elevada para o veículo com uma distância maior a percorrer – regra 2, e foram definidas que determinadas estações de trabalho eram mais prioritárias que outras – regra 3. Para cada uma das regras foram simuladas frotas de quatro, dez e dezasseis robôs. Para além disso, foi considerado que cada tarefa (navegar entre duas posições) é composta por duas subtarefas, a navegação propriamente dita e o processo de acostagem, que consiste na aproximação à posição final com precisão e de acordo com a orientação estipulada.

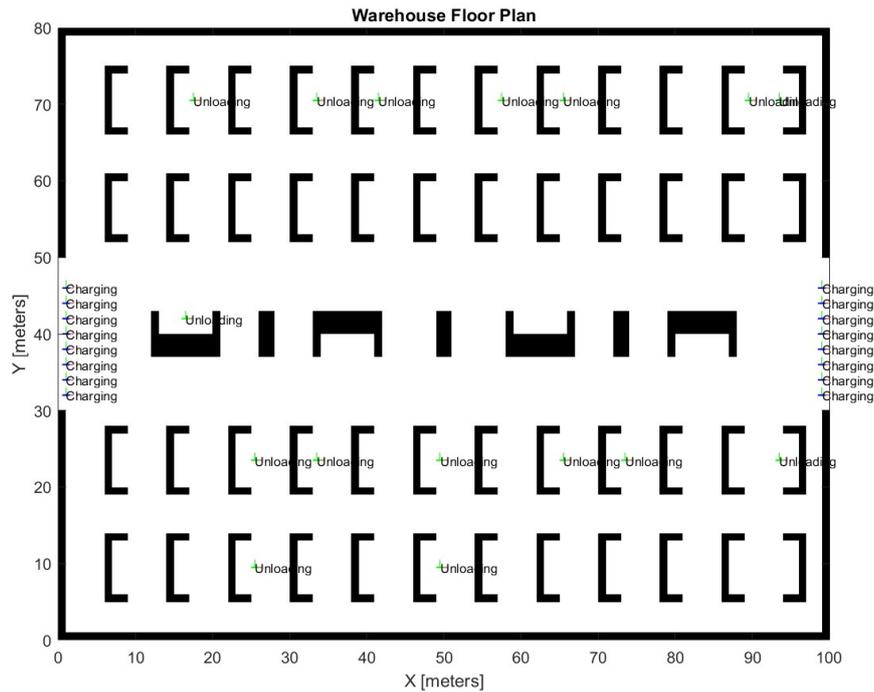


Figura 6-28 - Mapa do cenário 3, com 16 posições iniciais e finais previamente definidas

Foram novamente consideradas as trinta poses para o tamanho do intervalo de poses que deve ser considerado para a averiguação da existência de conflitos entre as rotas de dois ou mais robôs.

#### FROTA DE QUATRO ROBÔS

Para uma frota de quatro robôs, e tal como se verifica na Figura 6-29 **Erro! A origem da referência não foi encontrada.**, quando os níveis de prioridade são determinados pela regra 2 ou pela regra 3, obteve-se uma taxa de sucesso de 100%, ou seja, todos os veículos foram capazes de concluir com sucesso tanto a subtarefa de navegação como a subtarefa de acostagem. Por outro lado, quando é aplicada a regra 1, apenas se obteve uma taxa de sucesso de 87,5%, uma vez que um dos veículos da frota não foi capaz de completar o processo de acostagem com sucesso.

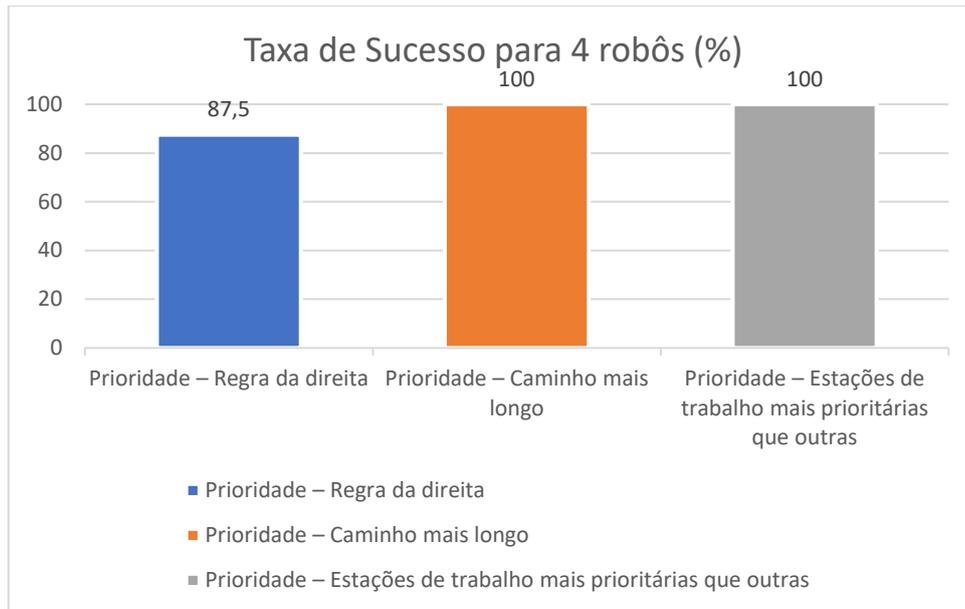


Figura 6-29 - Taxa de Sucesso para uma frota de 4 robôs

Quando comparando o comprimento do caminho inicialmente planejado com as distâncias que foram efetivamente percorridas (Figura 6-30 **Erro! A origem da referência não foi encontrada.**), verificou-se que, quando aplicada a regra 1, existe uma elevada discrepância entre as médias dos comprimentos inicial e final, ou seja, os robôs da frota percorreram mais vinte metros do que aquilo que tinha sido inicialmente planejado. Por outro lado, quando aplicada a regra 3, a distância total percorrida tem uma diferença de cerca de treze metros do que aquela que foi inicialmente calculada.

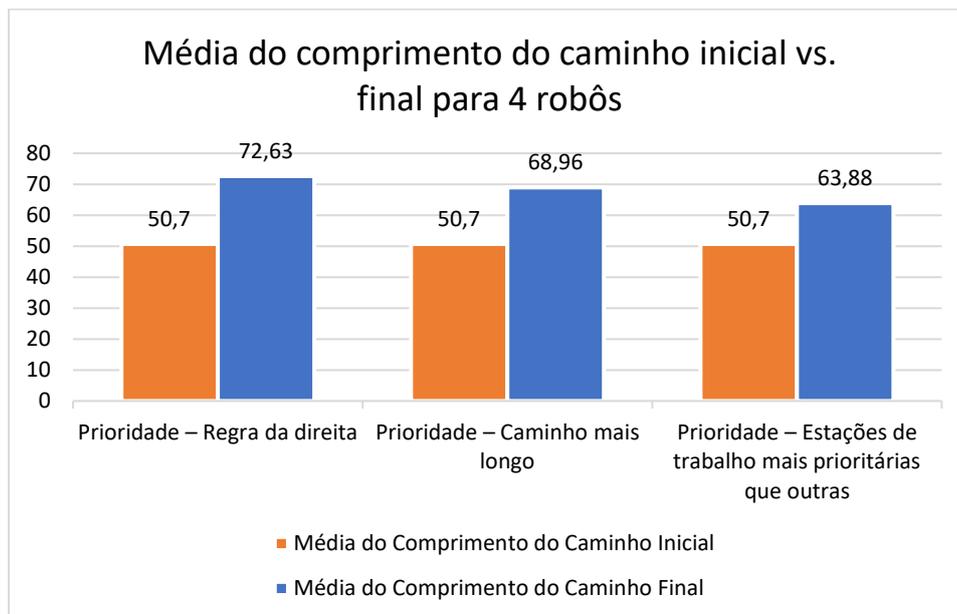


Figura 6-30 – Média do comprimento do caminho inicialmente planejado vs. Distância percorrida, para uma frota de 4 robôs

O número de conflitos totais detetados entre as rotas dos robôs (Figura 6-31 **Erro! A origem da referência não foi encontrada.**) aumentou com a diminuição da média da distância que foi efetivamente percorrida pelos veículos da frota. De notar quando a frota é constituída por apenas quatro veículos o número de conflitos é distribuído simetricamente pelos robôs.

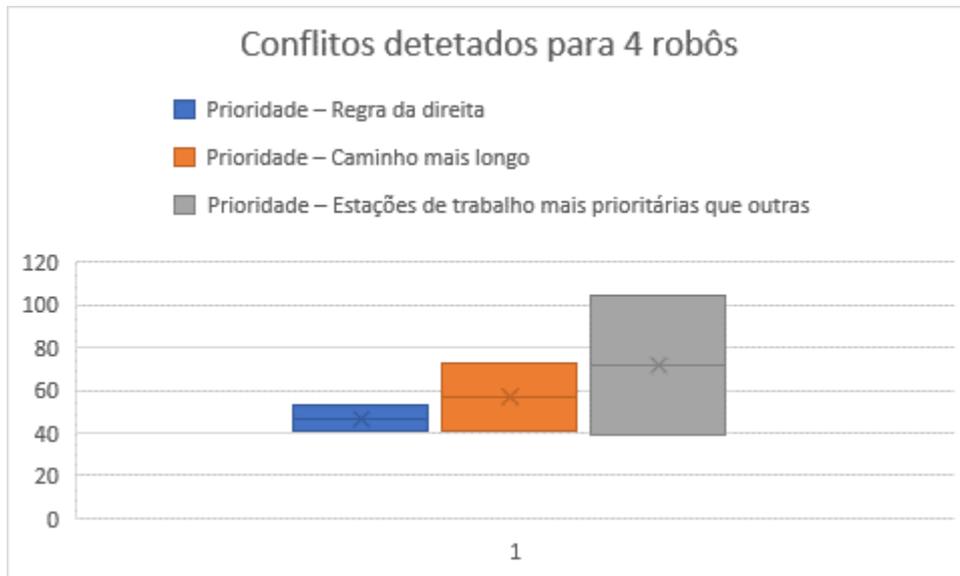


Figura 6-31 – Média e Desvio Padrão do número de conflitos para uma frota de 4 robôs

#### FROTA DE DEZ ROBÔS

Para uma frota de dez robôs, utilizando a regra 1, obteve-se uma taxa de sucesso de 100% (Figura 6-32). Por outro lado, quando utilizando a regra 2 ou 3, a taxa de sucesso foi de 90%, uma vez que, em ambos os casos, dois dos dez veículos da frota tiveram de abortar o processo de acostagem, mas completaram a primeira subtarefa com sucesso.

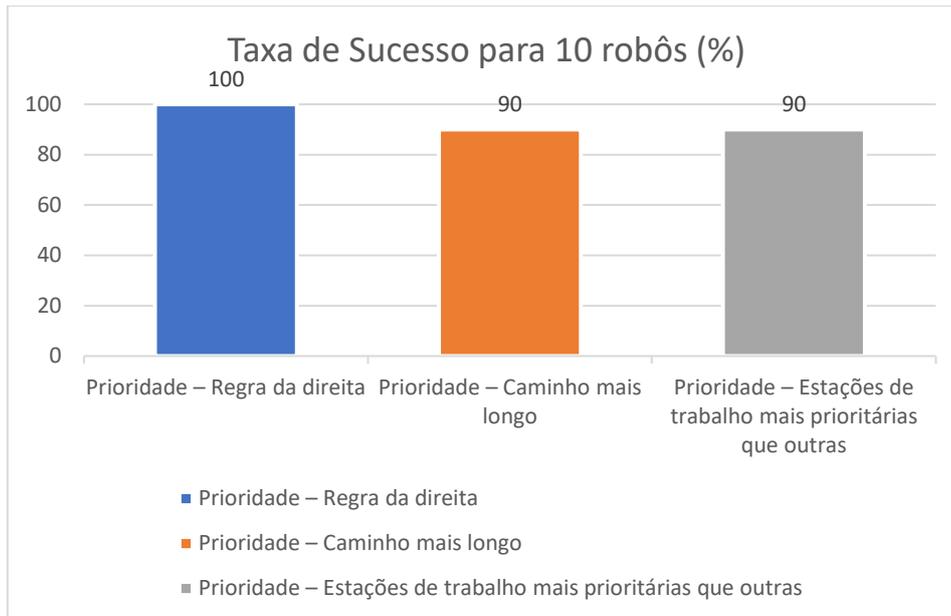


Figura 6-32 - Taxa de Sucesso para uma frota de 10 robôs

Tal como no caso anterior, no pior dos casos existe uma discrepância de cerca de vinte metros entre a média do comprimento do caminho inicialmente calculado e a média da distância percorrida, quando aplicada a regra 3 (Figura 6-33 **Erro! A origem da referência não foi encontrada.**). Quando aplicada a regra 2, a diferença entre as médias dos comprimentos inicial e final é cerca de quinze metros.

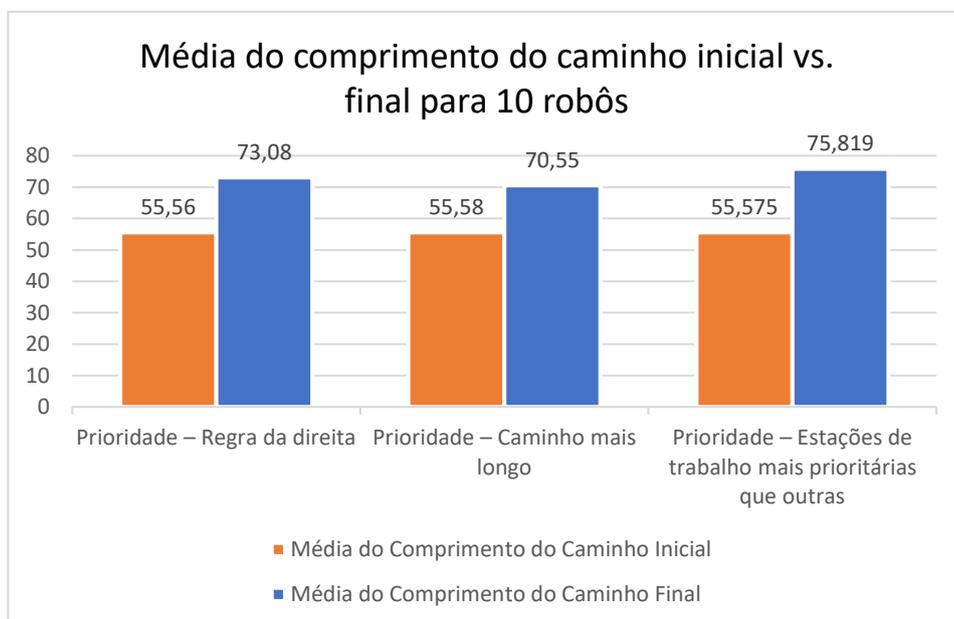


Figura 6-33 - Média do comprimento do caminho inicialmente planeado vs. Distância percorrida, para uma frota de 10 robôs

Através da Figura 6-34 **Erro! A origem da referência não foi encontrada.**, quando aplicada a regra 1, verifica-se que o número de conflitos ocorridos não se encontra igualmente distribuído. Neste caso,

obteve-se uma distribuição assimétrica do número de conflitos pelos robôs da frota, sendo que houve maior tendência para a ocorrência de conflitos quando aplicada a prioridade de cedência de passagem do veículo que se encontra o mais à direita, numa frota de dez robôs. Por outro lado, quando aplicada a regra 3, a tendência é existir um menor número de conflitos entre as rotas dos diversos veículos da frota.

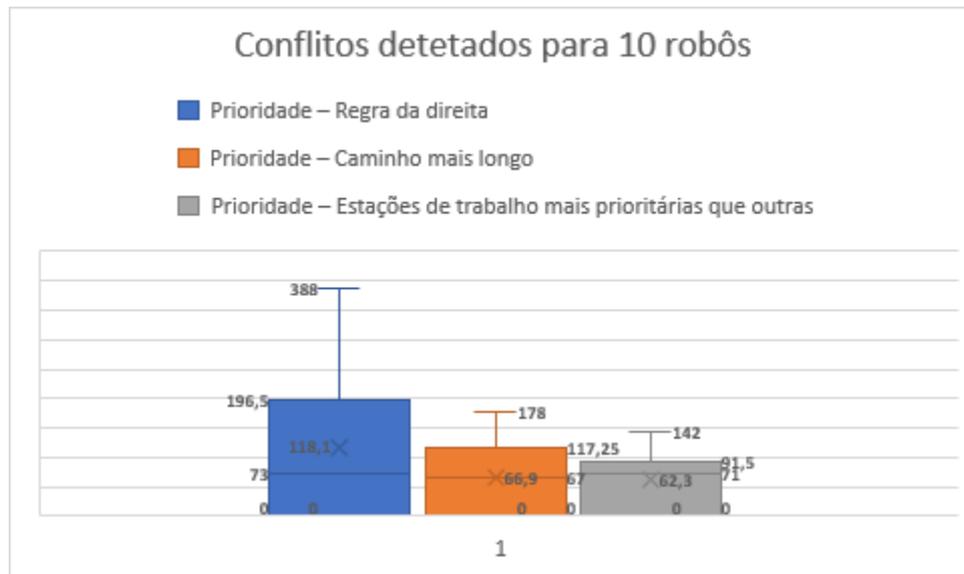


Figura 6-34 - Média e Desvio Padrão do número de conflitos para uma frota de 10 robôs

#### FROTA DE DEZASSEIS ROBÔS

Para uma frota de dezasseis robôs, não foi possível obter uma taxa de sucesso de 100% com nenhuma das três regras de prioridade implementadas (Figura 6-35). A taxa de sucesso mais elevada foi de 93,75% que se obteve utilizando a regra 2, onde dois dos dezasseis veículos da frota não foram capazes de concluir a segunda fase da tarefa com sucesso, obrigando o servidor central a abortar a subtarefa processo de acostagem para estes veículos. Utilizando as regras 1 e 3, verificou-se, em ambos os casos, que um dos veículos não foi capaz de alcançar a posição de destino, abortando a missão na primeira subtarefa (navegação), e outros dois veículos apresentaram dificuldades na conclusão da segunda subtarefa (acostagem). Deste modo, a taxa de sucesso nestes casos foi apenas de 87,5%.

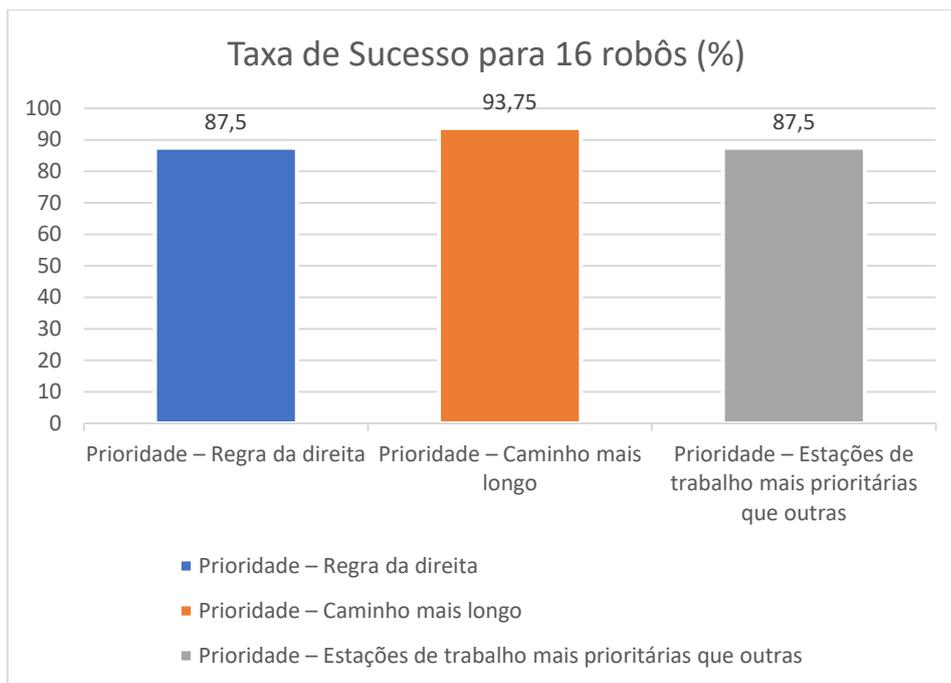


Figura 6-35 - Taxa de Sucesso para uma frota de 16 robôs

Perante uma frota de dezasseis robôs, a discrepância maior entre as médias do comprimento do caminho inicial e final foi cerca de onze metros, quando aplicada a regra 2 (Figura 6-36). A regra 2 foi a única que obteve uma taxa de sucesso de 100% na conclusão da primeira subtarefa (navegação), ou seja, onde todos os robôs da frota foram capazes de navegar entre as posições inicial e final com sucesso, demonstrando apenas dificuldades no processo de acostagem de dois veículos.

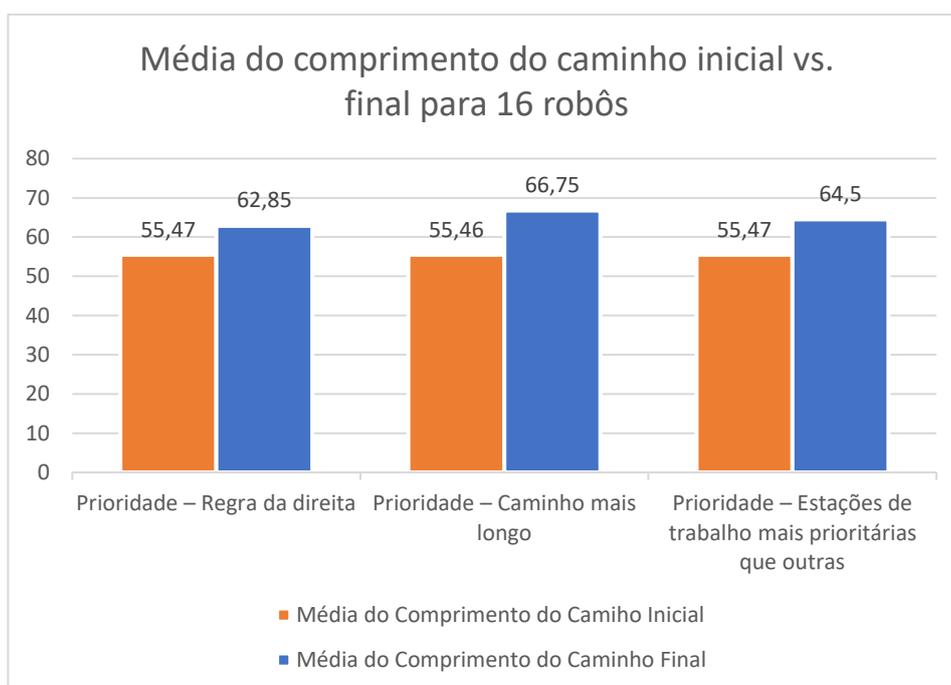


Figura 6-36 - Média do comprimento do caminho inicialmente planeado vs. Distância percorrida, para uma frota de 16 robôs

Através da Figura 6-37, verifica-se que, utilizando a regra 3, obtém-se um número de conflitos menor, em comparação com as restantes regras. Para além disso, é de notar que, em todos os casos, existe uma maior tendência para a ocorrência de conflitos numerosos num grupo mais alargado de veículos.

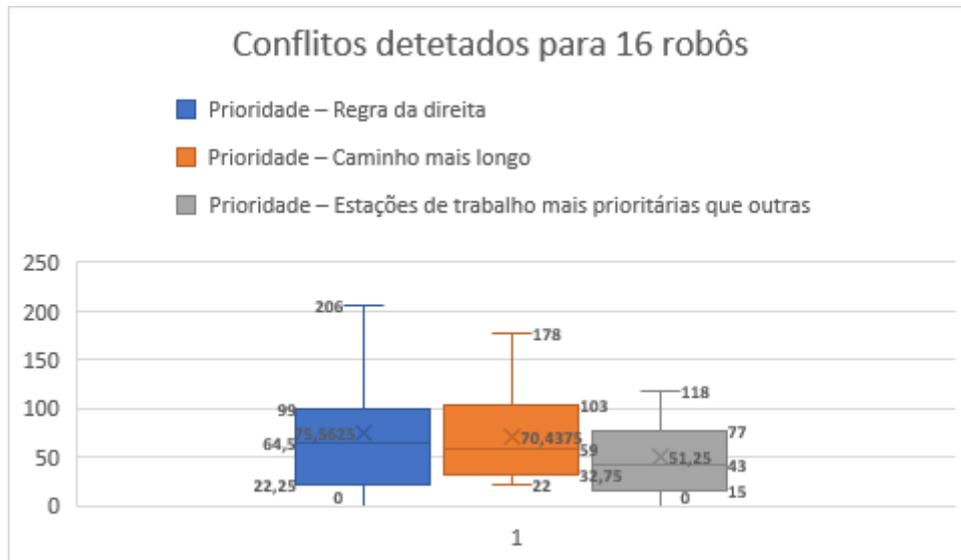


Figura 6-37 - Média e Desvio Padrão do número de conflitos para uma frota de 16 robôs

#### 6.4. DISCUSSÃO DE RESULTADOS

Para analisar a capacidade de replaneamento da rota de cada robô quando o veículo se encontra perante condições adversas, foram realizados novos testes onde se introduziu um obstáculo estático no meio de um corredor. Logo, e utilizando apenas um único robô, foi colocado um obstáculo estático a interseção da rota inicialmente planeada para analisar o comportamento do algoritmo escolhido. Assim que detetado o novo obstáculo, foi encontrado um novo caminho para a posição de destino, evitando com sucesso a obstrução com que o veículo se deparou. Este replaneamento gerou novamente o caminho mais curto entre a posição atual e a posição de destino, face às condições existentes.

Posteriormente, e de forma a validar a utilização deste algoritmo para a coordenação e planeamento de rotas em sistemas multi-robô, foram simuladas frotas de dois, cinco e dez robôs para o cenário 2. Inicialmente, foi estudado o modo como o número de pontos intermédios que são considerados no algoritmo de prevenção de conflitos entre robôs influenciam o comportamento dos mesmos e da coordenação da frota para um sistema de cinco robôs. Variando a dimensão do intervalo de coordenadas analisado, concluiu-se que com o aumento do tamanho do intervalo de valores

considerado para determinar a existência ou ausência de conflitos, aumenta também o número de robôs em conflito para os mesmos pares de posições iniciais e finais. Relativamente ao número de replaneamentos necessários não foi possível retirar conclusões, uma vez que a média desta variável se manteve, de uma forma geral, constante com o aumento do tamanho do intervalo de valores das coordenadas analisadas. Analisando os dados obtidos, foi definido que cada veículo apenas deveria analisar as trinta coordenadas seguintes, que corresponde a cerca de duas células, evitando assim que sejam detetados vários robôs em conflito e sejam requeridos vários replaneamentos da rota.

Estudando os diferentes casos simulados, e tal como seria de esperar, o número de conflitos das rotas dos robôs aumenta com o aumento dos veículos constituintes da frota. Consequentemente, o número de replaneamentos também aumenta, no entanto, a média da distância percorrida, quando nenhuma entrada para as estações de trabalho se encontra obstruída, é, apenas, ligeiramente superior à média da distância inicialmente calculada. Ou seja, o aumento do tamanho da frota não interfere, de forma significativa, na distância que cada veículo percorre.

Numa fase final de testes, foram realizadas duas simulações num cenário similar ao anteriormente utilizado, mas com menos espaço livre nos corredores, sendo que a única diferença entre estas simulações é a introdução de dois robôs de patrulha na última. Verificou-se que um cenário mais realista, ou seja, com corredores mais estreitos e/ou ocupados, colmatou num maior número de conflitos e replaneamentos. Em determinadas ocasiões, mesmo quando a posição de destino inicial era válida, o planeador não foi capaz de encontrar um caminho válido entre a pose atual e a final, obrigando o servidor central a atribuir uma nova tarefa, ou seja, definir uma nova posição de destino. Para além disso, averiguou-se que, sob determinadas circunstâncias, houve dificuldades no replaneamento, resultando, durante um curto período, numa oscilação dos caminhos gerados.

Por último, e utilizando o cenário 3, foram simuladas frotas de quatro, dez e dezasseis veículos a fim de estudar o comportamento de três variáveis quando aplicadas diferentes regras de prioridade: (1) regra de cedência de passagem do veículo que se encontra mais à direita, (2) prioridade cedida ao veículo cujo trajeto a percorrer é maior e, (3) prioridade atribuída consoante a estação de trabalho que foi alocada ao robô, ou seja, existem posições de destino mais prioritárias que outras. Analisando os resultados obtidos para as diferentes dimensões da frota, verificou-se que a regra 2 foi a única que obteve uma taxa de sucesso igual ou superior a 90%. Tal como foi verificado nas simulações efetuadas para o cenário 2, a média das distâncias percorridas não varia de forma significativa com a regra de prioridade adotada. Por outro lado, o número total de conflitos detetados é maior quando utilizada a regra 1 e, de forma geral, é menor quando utilizada a regra 3.

Em síntese, o algoritmo Híbrido A\* permitiu um replaneamento rápido e eficaz da rota quando detetado um obstáculo no meio do trajeto inicialmente planeado, impedindo a passagem do veículo naquela direção. Para além disso, e em conjugação com a estratégia adotada, este algoritmo também obteve resultados de sucesso quando a passagem para uma das posições de destino, inicialmente atribuídas aos veículos, estava interdita, sendo necessário atribuir uma nova tarefa (definir nova posição de destino) para o respetivo robô. No entanto, quando o cenário se torna mais complexo, o sistema já apresenta algumas falhas, tais como adversidades na busca do caminho mais curto entre duas posições ou até na busca de qualquer tipo de caminho válido, sendo obrigado a definir uma nova tarefa. Relativamente às regras de prioridade implementadas, concluiu-se que a regra 2 apresentou, de forma geral, melhores resultados, uma vez que foi a regra que obteve uma taxa de sucesso maior, onde todos os veículos foram capazes de alcançar a posição de destino, ou seja, todos os robôs concluíram o processo de navegação, mesmo que, por vezes, não tenham conseguido completar o processo de acostagem com sucesso num número reduzido de casos. Para além disso, tanto a média das distâncias percorridas como o número total de conflitos encontram-se dentro da gama de valores das médias das distâncias percorridas e do número total de conflitos quando comparado com as restantes regras.

## 7. CONCLUSÕES E TRABALHO FUTURO

Neste capítulo encontra-se uma revisão do trabalho efetuado e encontram-se apresentadas as conclusões obtidas e, assim como, algumas perspectivas que devem ser trabalhadas no futuro.

### 7.1. CONCLUSÕES

Com a aceleração da evolução tecnológica e a globalização dos mercados, as empresas têm sofrido uma pressão crescente para se tornarem mais eficientes e flexíveis, melhorando a sua capacidade de produção e eficiência operacional. Para além disso, existe o fator ambiental que tem ganho uma preocupação cada vez maior, sendo necessário adotar métodos sustentáveis, como a redução no consumo de recursos e na emissão de poluentes através da automação e otimização dos processos industriais. Em ambientes de logística interna, as frotas de veículos móveis autónomos são uma forma de automação que pode ajudar a melhorar a eficiência e flexibilidade dos processos de movimentação de produtos ou matéria-prima dentro de uma fábrica ou armazém. Os robôs constituintes da frota podem ser programados para realizar tarefas específicas, como transportar um dado produto entre duas estações de trabalho, e podem ser controlados remotamente para se adaptarem a possíveis mudanças na frota, como a remoção ou introdução de um novo veículo na frota ou alteração da tarefa que o robô em questão deve realizar, ou, até mesmo, alterações no *layout* do chão de fábrica.

Apresentado um dos problemas atuais presente na indústria, nesta dissertação foram analisadas algumas metodologias existentes para a coordenação de uma frota de veículos móveis autónomos em ambientes intralógicos. Adicionalmente, foram estudados alguns métodos de planeamento de caminhos, sendo capaz de os distinguir entre métodos clássicos, métodos exatos ou métodos heurísticos.

O *software* utilizado foi o MATLAB, tendo sido, também, estudadas as *toolboxes* mais relevantes para esta dissertação e, recorrendo a um dos exemplos disponíveis na *Navigation Toolbox*, foi efetuada uma breve análise aos planeadores, comparando-os através de parâmetros de referência, determinando que o planeador Híbrido A\* seria o mais adequado.

Por fim, foi implementada uma frota de veículos autónomos, baseado numa abordagem centralizada. A este servidor central são comunicadas as posições atuais de cada robô, assim como, as poses futuras para que seja possível a prevenção de colisões. Para além disso, as leituras efetuadas pelos sensores de cada robô são também comunicadas ao servidor central, o mapa é atualizado e o

novo mapa é comunicado aos restantes robôs constituintes da frota. Foram também implementadas três regras de prioridade de modo a estudar qual apresenta melhores resultados para um sistema multi-robô.

Relativamente às três regras de prioridade estudadas, e dando um peso maior à taxa de sucesso obtida para cada uma delas utilizando frotas de diferentes dimensões, conclui-se que a regra que cede prioridade ao veículo cujo caminho a percorrer é maior é a mais adequado para o sistema. Apesar do número de conflitos detetados utilizando esta regra ser ligeiramente superior aos detetados quando aplicada a regra que cede maior prioridade a determinadas estações de trabalho em comparação com outras, a regra 2 foi a única que apresentou uma taxa de sucesso de 100% na subtarefa de navegação.

## 7.2. TRABALHO FUTURO

Para trabalho futuro, e de modo a garantir a coordenação segura e eficiente de uma frota de robôs móveis, é necessário implementar um gestor de tarefas e garantir que os protocolos de segurança são obedecidos.

Adicionalmente, também não foram tidas em consideração nenhuma restrições temporais para a coordenação da frota dos veículos móveis. Um dos objetivos iniciais desta dissertação era a implementação de um sistema que respeitasse os tempos de execução de tarefas previamente estabelecidos. Isto é, a coordenação da frota ter por base o tempo e distância a percorrer. Tal poderia ser alcançado de diferentes maneiras, um dos métodos que poderia ser adotado seria recorrer ao servidor central para definir qual dos veículos tinha maior prioridade com base no tempo e distância que faltava percorrer a cada robô. Uma outra abordagem, seria o ajuste da velocidade do veículo consoante estes parâmetros, ou seja, se faltasse percorrer uma distância de valor considerável e o tempo para completar esta tarefa ser reduzido, aumentar a velocidade do robô.

Para além disso, o comportamento dos robôs foi testado através de simulações, sob condições ideais, ou seja, não foi considerado o ruído dos sensores reais nem nenhum outro fator adverso e imprevisível para o sistema, como o nível de carga das baterias do robô. Logo, futuramente, seria necessário realizar estes testes em robôs reais.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Ahmad Abbadi, & Vaclav Prenosil. (2015). Safe Path Planning Using Cell Decomposition Approximation. *International Conference Distance Learning, Simulation and Communication*.
- Arai, T., Pagello, E., & Parker, L. E. (2002). Editorial: Advances in Multi-Robot Systems. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 18(5), 655–661.
- Azadeh, K., de Koster, R., & Roy, D. (2019). Robotized and automated warehouse systems: Review and recent developments. In *Transportation Science* (Vol. 53, Issue 4, pp. 917–945). INFORMS. <https://doi.org/10.1287/trsc.2018.0873>
- Barrett, C., & Tinelli, C. (2018). Satisfiability modulo theories. *Handbook of Model Checking*, 305–343. [https://doi.org/10.1007/978-3-319-10575-8\\_11](https://doi.org/10.1007/978-3-319-10575-8_11)
- Berhault, M., Huang, H., Keskinocak, P., Koenig, S., Elmaghraby, W., Griffin, P., & Kleywegt, A. (2003). Robot Exploration with Combinatorial Auctions. *IEEE International Conference on Intelligent Robots and Systems*, 2, 1957–1962. <https://doi.org/10.1109/iros.2003.1248932>
- Bhattacharya, P., & Gavrilova, M. L. (2007). Voronoi diagram in optimal path planning. *Proceedings - ISVD 2007 The 4th International Symposium on Voronoi Diagrams in Science and Engineering 2007*, 1(c), 38–47. <https://doi.org/10.1109/ISVD.2007.43>
- Botelho, S. C., & Alami, R. (1999). M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. *Proceedings - IEEE International Conference on Robotics and Automation*, 2, 1234–1239. <https://doi.org/10.1109/robot.1999.772530>
- Cao, Y. U., Fukunaga, A. S., & Kahng, A. B. (1997). Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, 4(1), 7–27. <https://doi.org/10.1023/A:1008855018923>
- Cap, M., Novak, P., Kleiner, A., & Selecky, M. (2015). Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots. *IEEE Transactions on Automation Science and Engineering*, 12(3), 835–849. <https://doi.org/10.1109/TASE.2015.2445780>
- Choset, H., & Burdick, J. (2000). Sensor-based exploration: The hierarchical generalized Voronoi graph. *International Journal of Robotics Research*, 19(2), 96–125. <https://doi.org/10.1177/02783640022066770>

- Colby, M., Chung, J. J., & Tumer, K. (2015). Implicit adaptive multi-robot coordination in dynamic environments. *IEEE International Conference on Intelligent Robots and Systems, 2015-December*, 5168–5173. <https://doi.org/10.1109/IROS.2015.7354105>
- Couceiro, M. S., Portugal, D., Ferreira, J. F., & Rocha, R. P. (2019). SEMFIRE: Towards a new generation of forestry maintenance multi-robot systems. *Proceedings of the 2019 IEEE/SICE International Symposium on System Integration, SII 2019*, 270–276. <https://doi.org/10.1109/SII.2019.8700403>
- Darmanin, R. N., & Bugeja, M. K. (2017). A review on multi-robot systems categorised by application domain. *2017 25th Mediterranean Conference on Control and Automation, MED 2017*, 701–706. <https://doi.org/10.1109/MED.2017.7984200>
- Das, P. K., Behera, H. S., Jena, P. K., & Panigrahi, B. K. (2021). An intelligent multi-robot path planning in a dynamic environment using improved gravitational search algorithm. *International Journal of Automation and Computing*, 18(6), 1032–1044. <https://doi.org/10.1007/s11633-016-1019-x>
- de Lope, J., Maravall, D., & Quiñonez, Y. (2013). Response threshold models and stochastic learning automata for self-coordination of heterogeneous multi-task distribution in multi-robot systems. *Robotics and Autonomous Systems*, 61(7), 714–720. <https://doi.org/10.1016/j.robot.2012.07.008>
- Demesure, G., Defoort, M., Bekrar, A., Trentesaux, D., & Djemai, M. (2018). Decentralized Motion Planning and Scheduling of AGVs in an FMS. *IEEE Transactions on Industrial Informatics*, 14(4), 1744–1752. <https://doi.org/10.1109/TII.2017.2749520>
- Digani, V., Hsieh, M. A., Sabattini, L., & Secchi, C. (2015). A Quadratic Programming approach for coordinating multi-AGV systems. *IEEE International Conference on Automation Science and Engineering*, 600–605. <https://doi.org/10.1109/CoASE.2015.7294144>
- Digani, V., Sabattini, L., Secchi, C., & Fantuzzi, C. (2014). Hierarchical traffic control for partially decentralized coordination of multi AGV systems in industrial environments. *IEEE International Conference on Robotics and Automation*, 6144–6149. <https://doi.org/10.1109/ICRA.2014.6907764>

- Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2010). Path planning for autonomous vehicles in unknown semi-structured environments. *International Journal of Robotics Research*, 29(5), 485–501. <https://doi.org/10.1177/0278364909359210>
- Fainekos, G. E., Girard, A., Kress-Gazit, H., & Pappas, G. J. (2009). Temporal logic motion planning for dynamic robots. *Automatica*, 45(2), 343–352. <https://doi.org/10.1016/j.automatica.2008.08.008>
- Fanti, M. P., Mangini, A. M., Pedroncelli, G., & Ukovich, W. (2018). A decentralized control strategy for the coordination of AGV systems. *Control Engineering Practice*, 70, 86–97. <https://doi.org/10.1016/j.conengprac.2017.10.001>
- Farinelli, A., Boscolo, N., Zanutto, E., & Pagello, E. (2017). Advanced approaches for multi-robot coordination in logistic scenarios. *Robotics and Autonomous Systems*, 90, 34–44.
- Farinelli, A., Iocchi, L., & Nardi, D. (2004). Multirobot systems: A classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(5), 2015–2028. <https://doi.org/10.1109/TSMCB.2004.832155>
- Ferber, J. (1999). *Multi-Agent System: An Introduction to Distributed Artificial Intelligence* (Vol. 1).
- Ferrarezzo, M., Ferrari, C., Pagello, E., Polesel, R., Rosati, R., Speranzon, A., & Zanette, W. (2001). *Collaborative emergent actions between real soccer robots*. 297–302.
- Fragapane, G., de Koster, R., Sgarbossa, F., & Strandhagen, J. O. (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research*, 294(2), 405–426. <https://doi.org/10.1016/j.ejor.2021.01.019>
- Gerkey, B. P., & Mataric, M. J. (2002). Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5), 758–768. <https://doi.org/10.1109/TRA.2002.803462>
- Gerkey, B. P., & Mataric, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9), 939–954. <https://doi.org/10.1177/0278364904045564>
- Giuzzo, E. (2008). Three engineers, hundred of robots, one warehouse. *IEEE Spectr.* 45, 26–34.

- Gregory, J., Fink, J., Stump, E., Twigg, J., Rogers, J., Baran, D., Fung, N., & Young, S. (2016). Application of multi-robot systems to disaster-relief scenarios with limited communication. *Springer Tracts in Advanced Robotics*, 113, 639–653. [https://doi.org/10.1007/978-3-319-27702-8\\_42](https://doi.org/10.1007/978-3-319-27702-8_42)
- Guo, M., Tumova, J., & Dimarogonas, D. v. (2014). *Cooperative Decentralized Multi-agent Control under Local LTL Tasks and Connectivity Constraints*.
- Gustafson, S., & Gustafson, D. A. (2006). Issues in the scaling of multi-robot systems for general problem solving. *Autonomous Robots*, 20(2), 125–136. <https://doi.org/10.1007/s10514-006-5939-0>
- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions of Systems Science and Cybernetics*, 4(2), 100–107.
- Hernández, C., Bermejo-Alonso, J., & Sanz, R. (2018). *A Self-adaptation Framework Based on Functional Knowledge for Augmented Autonomy in Robots*. 157–172.
- Hoffman, A. E., Mountz, M. C., Barbehenn, M. T., & ... (2016). System and method for inventory management using mobile drive units. In *U.S. Patent No. 9,317,034*. Washington, DC: U.S. Patent and Trademark Office. <https://patents.google.com/patent/US9317034B2/en>
- Howard, A., Parker, L. E., & Sukhatme, G. S. (2006). Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *International Journal of Robotics Research*, 25(5–6), 431–447. <https://doi.org/10.1177/0278364906065378>
- Hu, X., & Wang, J. (2008). An Improved Dual Neural Network for Solving a Class of Quadratic Programming Problems and Its k-Winners-Take-All Application. *IEEE Transactions on Neural Networks*, 39(12), 2022–2031. <https://doi.org/10.1016/j.neunet.2012.12.009>
- locchi, L., Nardi, D., & Salerno, M. (2001a). Reactivity and deliberation: A survey on multi-robot systems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2103, 9–32. [https://doi.org/10.1007/3-540-44568-4\\_2](https://doi.org/10.1007/3-540-44568-4_2)
- locchi, L., Nardi, D., & Salerno, M. (2001b). Reactivity and Deliberation: A Survey on Multi-Robot Systems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2103, 9–32. [https://doi.org/10.1007/3-540-44568-4\\_1](https://doi.org/10.1007/3-540-44568-4_1)

- Jain, R. P., Aguiar, A. P., & de Sousa, J. B. (2018). Cooperative path following of robotic vehicles using an event-based control and communication strategy. *IEEE Robotics and Automation Letters*, *3*(3), 1941–1948. <https://doi.org/10.1109/LRA.2018.2808363>
- Jin, L., Li, S., La, H. M., Zhang, X., & Hu, B. (2019). Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach. *Automatica*, *100*, 75–81. <https://doi.org/10.1016/j.automatica.2018.11.001>
- Jose, K., & Pratihari, D. K. (2016). Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robotics and Autonomous Systems*, *80*, 34–42. <https://doi.org/10.1016/j.robot.2016.02.003>
- Kaluder, H., Brezak, M., & Petrović, I. (2011). A visibility graph based method for path planning in dynamic environments. *MIPRO 2011 - 34th International Convention on Information and Communication Technology, Electronics and Microelectronics - Proceedings*, 717–721.
- Kanjanawanishkul, K. (2014). Koordinirano slije.Enje putanje za mobilne robote zasnovano na strategiji virtualne strukture i modelskom prediktivnom upravljanju. *Automatika*, *55*(3), 287–298. <https://doi.org/10.7305/automatika.2014.12.460>
- Kato, S., Nishiyama, S., & Takeno, J. (1992). Coordinating mobile robots by applying traffic rules. *IEEE International Conference on Intelligent Robots and Systems*, *3*, 1535–1541. <https://doi.org/10.1109/IROS.1992.594218>
- Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. *Proceedings - IEEE International Conference on Robotics and Automation*, *5*(1), 500–505. <https://doi.org/10.1109/ROBOT.1985.1087247>
- Kloetzer, M., Burlacu, A., & Panescu, D. (2012). On a class of multi-robot task allocation problems. In *IFAC Proceedings Volumes (IFAC-PapersOnline)* (Vol. 14, Issue PART 1). IFAC. <https://doi.org/10.3182/20120523-3-RO-2023.00327>
- LaValle, S. M. (2006). Planning algorithms. In *Planning Algorithms* (Vol. 9780521862). <https://doi.org/10.1017/CBO9780511546877>
- Lee, S. U., Gonzalez, R., & Iagnemma, K. (2016). Robust sampling-based motion planning for autonomous tracked vehicles in deformable high slip terrain. *Proceedings - IEEE International*

- Conference on Robotics and Automation, 2016-June*, 2569–2574.  
<https://doi.org/10.1109/ICRA.2016.7487413>
- Li, X., Yang, X., Zhang, C., & Qi, M. (2021). A simulation study on the robotic mobile fulfillment system in high-density storage warehouses. *Simulation Modelling Practice and Theory*, 112.  
<https://doi.org/10.1016/j.simpat.2021.102366>
- Lingelbach, F. (2004). Path planning using probabilistic cell decomposition. *Proceedings - IEEE International Conference on Robotics and Automation, 2004(1)*, 467–472.  
<https://doi.org/10.1109/robot.2004.1307193>
- Liu, Y., Yang, J., Zheng, Y., Wu, Z., & Yao, M. (2013). Multi-robot coordination in complex environment with task and communication constraints. *International Journal of Advanced Robotic Systems*, 10.  
<https://doi.org/10.5772/54379>
- López, J., Pérez, D., & Zalama, E. (2011). A framework for building mobile single and multi-robot applications. *Robotics and Autonomous Systems*, 59(3–4), 151–162.  
<https://doi.org/10.1016/J.ROBOT.2011.01.004>
- Luis, C. E., Vukosavljev, M., & Schoellig, A. P. (2020). Predictive Control for Multi-Robot Motion Planning. *Ieee Robotics and Automation Letters*, 5(2), 604–611.
- Lutz, M., Verbeek, C., & Schlegel, C. (2016). Towards a robot fleet for intra-logistic tasks: Combining free robot navigation with multi-robot coordination at bottlenecks. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2016-November*.  
<https://doi.org/10.1109/ETFA.2016.7733602>
- Mendiburu, F. J., Morais, M. R. A., & Lima, A. M. N. (2016). Behavior coordination in multi-robot systems. *2016 IEEE International Conference on Automatica, ICA-ACCA 2016*.  
<https://doi.org/10.1109/ICA-ACCA.2016.7778506>
- Mendonca, R., Marques, M. M., Marques, F., Lourenço, A., Pinto, E., Santana, P., Coito, F., Lobo, V., & Barata, J. (2016). A cooperative multi-robot team for the surveillance of shipwreck survivors at sea. *OCEANS 2016 MTS/IEEE Monterey, OCE 2016*, 2–7.  
<https://doi.org/10.1109/OCEANS.2016.7761074>
- Mojtahedzade, R. (2011). Robot Obstacle Avoidance using the Kinect Master of Science Thesis Robot Obstacle Avoidance using the Kinect. *Thesis Master, May*, 1–54.

- Olmi, R., Secchi, C., & Fantuzzi, C. (2011). An efficient control strategy for the traffic coordination of AGVs. *International Conference on Intelligent Robots and Systems*, 4615–4620. <https://doi.org/10.1109/iros.2011.6094474>
- Omidshafiei, S., Agha-Mohammadi, A. A., Amato, C., Liu, S. Y., How, J. P., & Vian, J. (2017). Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions. *International Journal of Robotics Research*, 36(2), 231–258. <https://doi.org/10.1177/0278364917692864>
- Otte, M., Kuhlman, M. J., & Sofge, D. (2020). Auctions for multi-robot task allocation in communication limited environments. In *Autonomous Robots* (Vol. 44, Issues 3–4). Springer US. <https://doi.org/10.1007/s10514-019-09828-5>
- Otte, M., Kuhlman, M., & Sofge, D. (2017). Multi-robot task allocation with auctions in harsh communication environments. *2017 International Symposium on Multi-Robot and Multi-Agent Systems, MRS 2017, 2018-Janua*, 32–39. <https://doi.org/10.1109/MRS.2017.8250928>
- Parker, L. E. (1998). ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2), 220–240. <https://doi.org/10.1109/70.681242>
- Parker, L. E. (2008). Multiple Mobile Robot Systems. *Springer Handbook of Robotics*, 921–941. [https://doi.org/10.1007/978-3-540-30301-5\\_41](https://doi.org/10.1007/978-3-540-30301-5_41)
- Patle, B. K., Pandey, A., Jagadeesh, A., & Parhi, D. R. (2018). Path planning in uncertain environment by using firefly algorithm. *Defence Technology*, 14(6), 691–701. <https://doi.org/10.1016/j.dt.2018.06.004>
- Planning Algorithms - Steven M. LaValle - Google Livros*. (n.d.). Retrieved October 24, 2022, from [https://books.google.pt/books?hl=pt-PT&lr=&id=PwLBAAAQBAJ&oi=fnd&pg=PT9&dq=S.+M.+LaValle,+Planning+Algorithms.+New+York,+NY,+USA:+Cambridge+University+Press,+2006.&ots=0iEz4tunmp&sig=f8TIY42haDmKM6Xdw\\_08n1FlpkM&redir\\_esc=y#v=onepage&q&f=false](https://books.google.pt/books?hl=pt-PT&lr=&id=PwLBAAAQBAJ&oi=fnd&pg=PT9&dq=S.+M.+LaValle,+Planning+Algorithms.+New+York,+NY,+USA:+Cambridge+University+Press,+2006.&ots=0iEz4tunmp&sig=f8TIY42haDmKM6Xdw_08n1FlpkM&redir_esc=y#v=onepage&q&f=false)
- Popenoe, D. (2000). *Sociology*.
- Prattisoli, F., Battilani, N., Fantuzzi, C., & Sabbatini, L. (2021). *Hierarchical and Flexible Traffic Management of Multi-AGV Systems Applied to Industrial Environments*. 10009–10015. <https://doi.org/10.1109/icra48506.2021.9560828>

- Qing, G., Zheng, Z., & Yue, X. (2017). Path-planning of automated guided vehicle based on improved Dijkstra algorithm. *Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017*, 7138–7143. <https://doi.org/10.1109/CCDC.2017.7978471>
- Russell, S. J., & Norvig, P. (2010). Artificial intelligence. In *2010 The 2nd International Conference on Computer and Automation Engineering, ICCAE 2010* (Vol. 4). <https://doi.org/10.1109/ICCAE.2010.5451578>
- Saha, I., Ramaithitima, R., Kumar, V., Pappas, G. J., & Seshia, S. A. (2016). Implan: Scalable Incremental Motion Planning for Multi-Robot Systems. *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems, ICCPS 2016 - Proceedings*. <https://doi.org/10.1109/ICCPS.2016.7479105>
- Santos, J., Rebelo, P. M., Rocha, L. F., Costa, P., & Veiga, G. (2021). A\* Based routing and scheduling modules for multiple agvs in an industrial scenario. *Robotics*, *10*(2). <https://doi.org/10.3390/robotics10020072>
- Sardinha, F. D. J. (2017). A Gestão de Transportes na Cadeia de Logística. (*Dissertação de Mestrado, Instituto Superior de Gestão, Lisboa*), 87. [https://comum.rcaap.pt/bitstream/10400.26/22181/1/fernando\\_sardinha\\_-\\_ISG\\_final.pdf](https://comum.rcaap.pt/bitstream/10400.26/22181/1/fernando_sardinha_-_ISG_final.pdf)
- Schillinger, P., Burger, M., & V.Diamarogonas, D. (2018). Simultaneous Task Allocation Planning for Temporal Logic Goals in Heterogeneous Multi-Robot Systems. *The International Journal of Robotics Research*, *37*(7), 818–838. <https://doi.org/10.1177/ToBeAssigned>
- Schlegel, C. (1998). Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot. *IEEE International Conference on Intelligent Robots and Systems*, *1*, 594–599. <https://doi.org/10.1109/iros.1998.724683>
- Siddiqui, R. (n.d.). *Path Planning Using Potential Field Algorithm*. Retrieved October 23, 2022, from <https://medium.com/@rymshasiddiqui/path-planning-using-potential-field-algorithm-a30ad12bdb08>
- Sieber, D., & Hirche, S. (2019). Human-Guided Multirobot Cooperative Manipulation. *IEEE Transactions on Control Systems Technology*, *27*(4), 1492–1509. <https://doi.org/10.1109/TCST.2018.2813323>

- Siegwart, R., & Nourbakhsh, I. R. (2011). *Introduction to Autonomous Mobile Robots*. The MIT press.  
<http://home.deib.polimi.it/gini/robot/docs/siegwart.pdf>
- Smith, R. G. (1980). The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, *C-29*(12), 1104–1113.  
<https://doi.org/10.1109/TC.1980.1675516>
- Solovey, K., Salzman, O., & Halperin, D. (2016). Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning. *International Journal of Robotics Research*, *35*(5), 501–513. <https://doi.org/10.1177/0278364915615688>
- Stancovici, A., Micea, M. v., & Cretu, V. (2016). Cooperative positioning system for indoor surveillance applications. *2016 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2016*, *1*. <https://doi.org/10.1109/IPIN.2016.7743584>
- Standley, T. (2010). Finding optimal solutions to cooperative pathfinding problems. *Proceedings of the National Conference on Artificial Intelligence*, *1*, 173–178.
- Standley, T., & Korf, R. (2011). Complete algorithms for cooperative pathfinding problems. *IJCAI International Joint Conference on Artificial Intelligence*, 668–673. <https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-118>
- Sun, Y., Fang, M., & Su, Y. (2021). AGV Path Planning based on Improved Dijkstra Algorithm. *Journal of Physics: Conference Series*, *1746*(1). <https://doi.org/10.1088/1742-6596/1746/1/012052>
- Trigui, S., Koubaa, A., Cheikhrouhou, O., Youssef, H., Bennaceur, H., Sriti, M. F., & Javed, Y. (2014). A distributed market-based algorithm for the multi-robot assignment problem. *Procedia Computer Science*, *32*, 1108–1114. <https://doi.org/10.1016/j.procs.2014.05.540>
- Turpin, M., Michael, N., & Kumar, V. (2014). CAPT: Concurrent assignment and planning of trajectories for multiple robots. *International Journal of Robotics Research*, *33*(1), 98–112.  
<https://doi.org/10.1177/0278364913515307>
- Verma, J. K., & Ranga, V. (2021). Multi-Robot Coordination Analysis, Taxonomy, Challenges and Future Scope. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *102*(1).  
<https://doi.org/10.1007/s10846-021-01378-2>

- Viguria, A., Maza, I., & Ollero, A. (2008). S+T: An algorithm for distributed multirobot task allocation based on services for improving robot cooperation. *Proceedings - IEEE International Conference on Robotics and Automation*, 3163–3168. <https://doi.org/10.1109/ROBOT.2008.4543692>
- Wagner, G., & Choset, H. (2015). Subdimensional expansion for multirobot path planning. *Artificial Intelligence*, 219, 1–24. <https://doi.org/10.1016/j.artint.2014.11.001>
- Wang, S., Liu, C., Ding, N., Zhao, Y., & Gao, X. (2021). A Hybrid Coordination Method for Multi-robot in Restricted Movement Scene. *Proceedings - 5th IEEE International Conference on Smart Internet of Things, SmartIoT 2021*, 205–212. <https://doi.org/10.1109/SmartIoT52359.2021.00040>
- Weigel, T., Gutmann, J. S., Dietl, M., Kleiner, A., & Nebel, B. (2002). CS Freiburg: Coordinating robots for successful soccer playing. *IEEE Transactions on Robotics and Automation*, 18(5), 685–699. <https://doi.org/10.1109/TRA.2002.804041>
- Xia, W., Goh, J., Cortes, C. A., Lu, Y., & Xu, X. (2020). Decentralized coordination of autonomous AGVs for flexible factory automation in the context of Industry 4.0; Decentralized coordination of autonomous AGVs for flexible factory automation in the context of Industry 4.0. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. [https://doi.org/10.0/Linux-x86\\_64](https://doi.org/10.0/Linux-x86_64)
- Xu, B., Institute of Electrical and Electronics Engineers. Beijing Section, & Institute of Electrical and Electronics Engineers. (n.d.). *Proceedings of 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC 2019) : October 11-13, 2019, Chongqing, China*.
- Yan, Z., Jouandeau, N., & Cherif, A. A. (2013a). A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10. <https://doi.org/10.5772/57313>
- Yan, Z., Jouandeau, N., & Cherif, A. A. (2013b). *International Journal of Advanced Robotic Systems A Survey and Analysis of Multi-Robot Coordination Regular Paper*. <https://doi.org/10.5772/57313>
- Yuan, F., Twardon, L., & Hanheide, M. (2010). Dynamic path planning adopting human navigation strategies for a domestic mobile robot. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings, April*, 3275–3281. <https://doi.org/10.1109/IROS.2010.5650307>

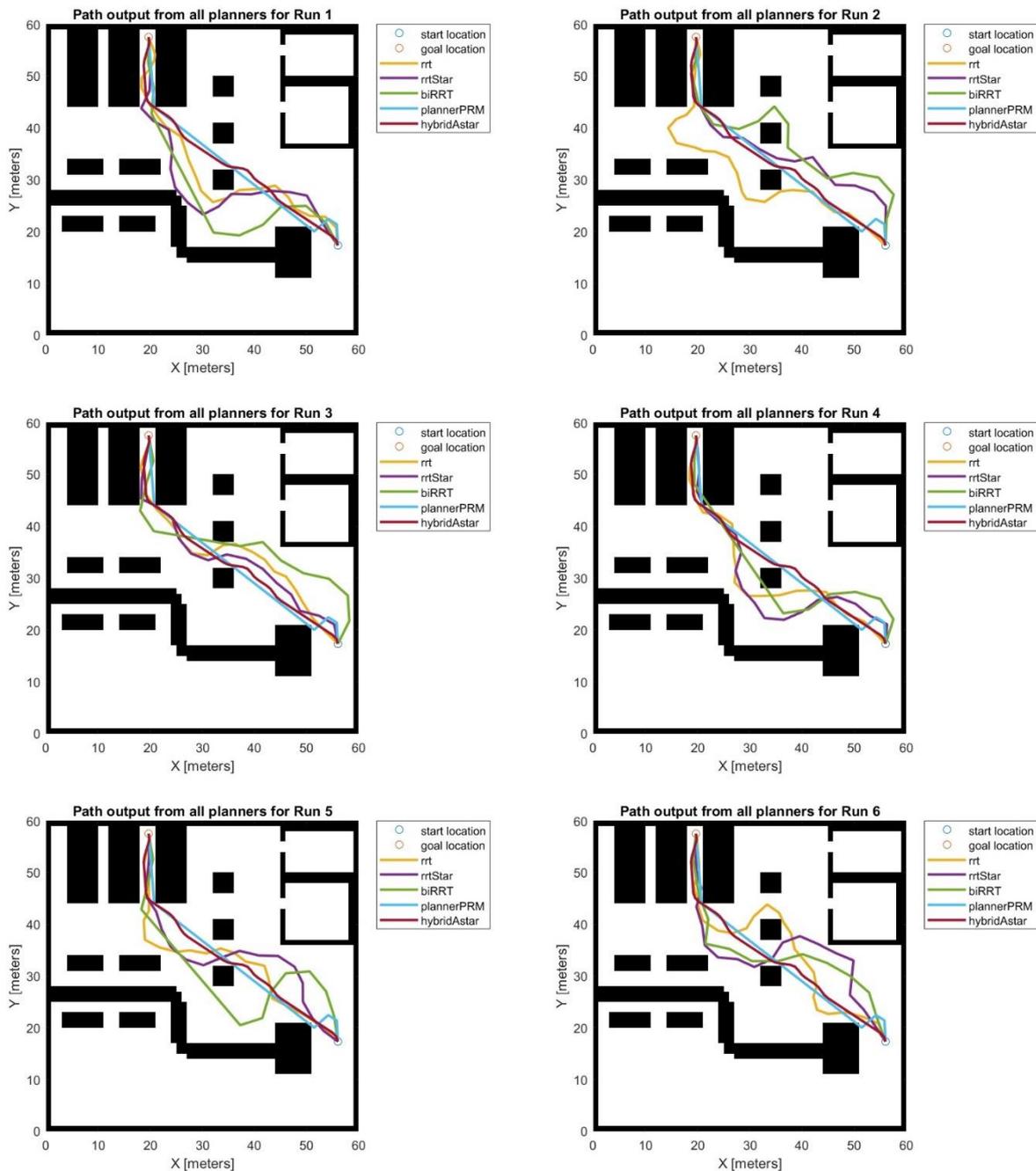
Zhang, D., & Wang, L. (2007). Target topology based task assignment for multiple mobile robots in adversarial environments. *Proceedings of the IEEE Conference on Decision and Control*, 5323–5328. <https://doi.org/10.1109/CDC.2007.4434010>

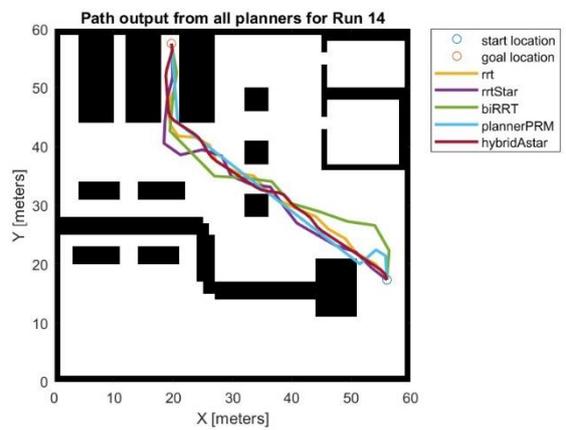
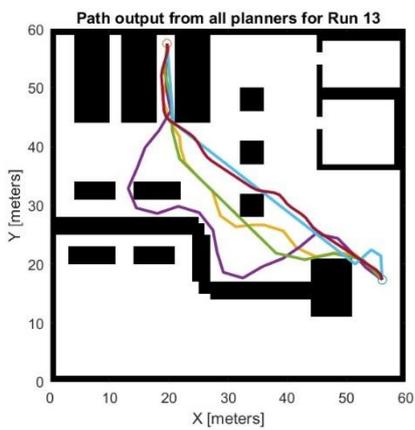
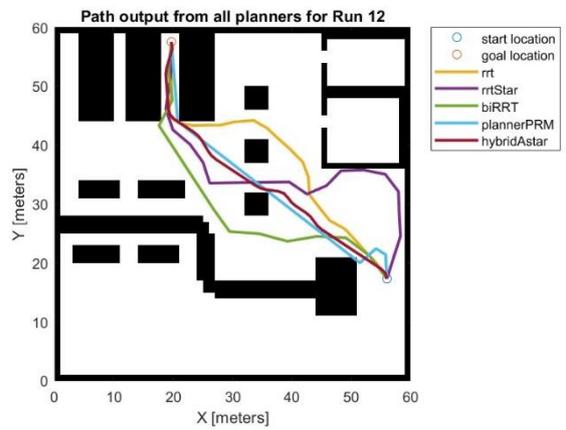
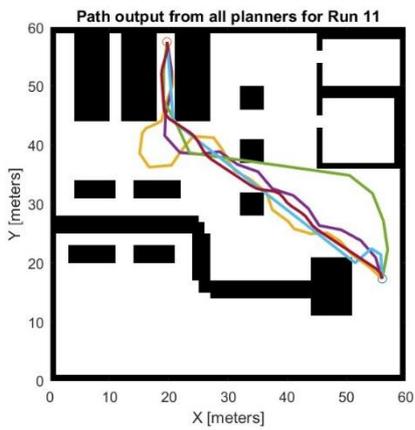
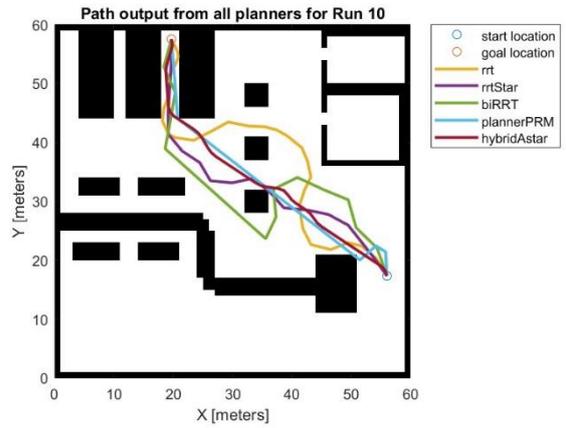
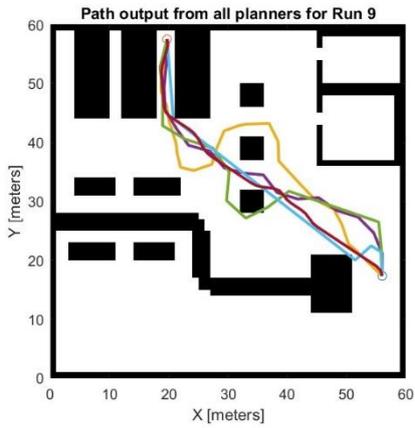
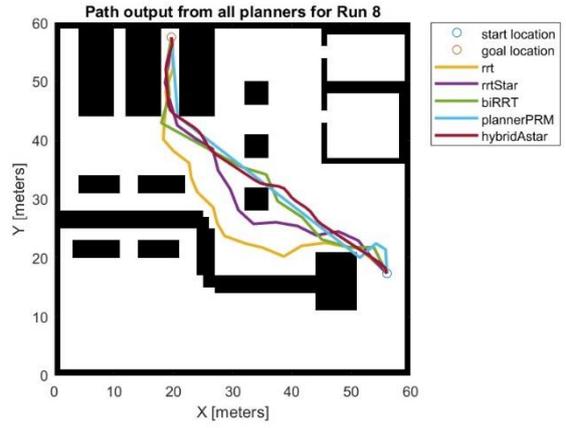
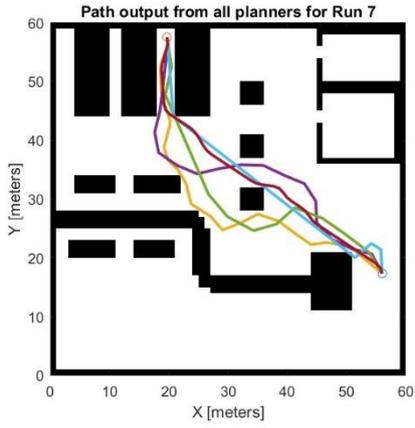
Zhou, Y., Hu, H., Liu, Y., & Ding, Z. (2017). Collision and deadlock avoidance in multirobot systems: A distributed approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7), 1712–1726. <https://doi.org/10.1109/TSMC.2017.2670643>

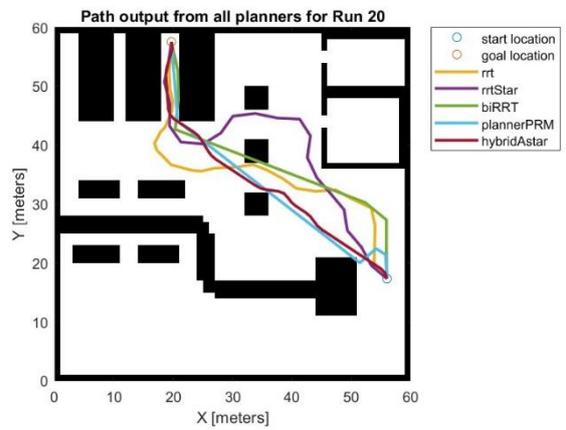
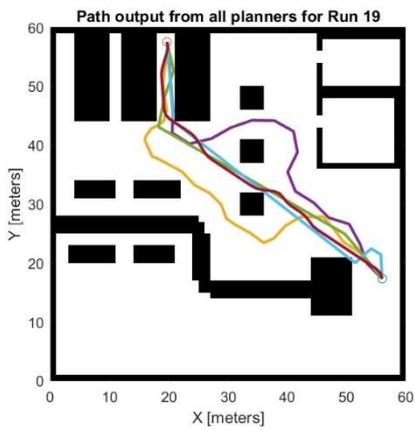
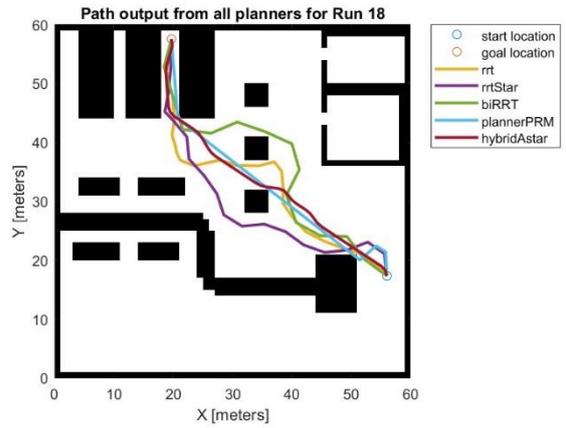
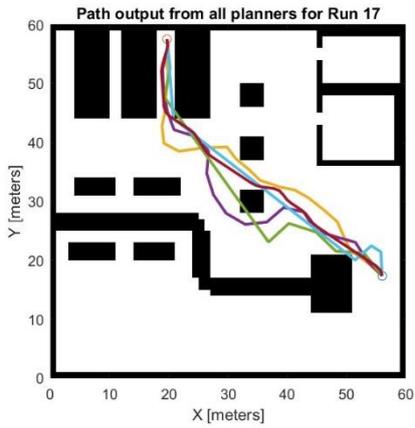
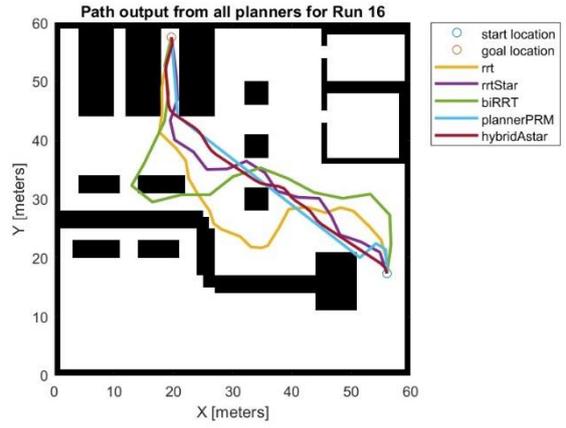
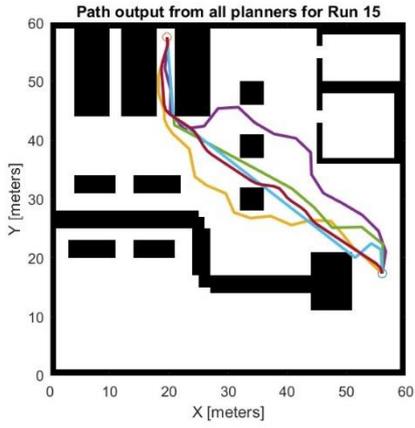
## A. APÊNDICES

### A1. ANÁLISE DOS PLANEADORES - CENÁRIO 1

Nas figuras abaixo encontram-se representados os caminhos encontrados por todos os planeadores para as vinte execuções realizadas ao sistema, para o cenário 1. Os planeadores Híbrido A\* e PRM foram os que demonstraram uma maior robustez, sendo que o caminho encontrado, para cada um dos algoritmos de busca, foi semelhante em todas as execuções.







## A2. ANÁLISE DOS PLANEADORES - CENÁRIO 2

Numa segunda fase de testes, e alterando o cenário para aquele que iria ser a base das restantes simulações, realizaram-se os mesmos testes que em A1. Análise dos Planeadores - Cenário 1. Para este cenário e para o par de posições utilizado, não foi possível encontrar um caminho utilizando o planeador PRM. O algoritmo Híbrido A\* demonstrou ser robusto, uma vez que, em vinte execuções, foi consistente no caminho gerado para o par de posições dadas. Por outro lado, e conforme analisado no caso anterior, os restantes algoritmos, para as vinte execuções realizadas, encontravam sempre caminhos diferentes.

