



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

*Danielli Araújo Lima*

---

**AUTÔMATOS CELULARES E SISTEMAS  
BIO-INSPIRADOS APLICADOS AO CONTROLE  
INTELIGENTE DE ROBÔS**

---

Uberlândia - Minas Gerais

2017





UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

*Danielli Araújo Lima*

**AUTÔMATOS CELULARES E SISTEMAS BIO-INSPIRADOS  
APLICADOS AO CONTROLE INTELIGENTE DE ROBÔS**

Tese de Doutorado apresentada à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como parte dos requisitos exigidos para obtenção do título de Doutora em Ciência da Computação.

Área de concentração: Inteligência Artificial.

Orientadora:

Profa. Dra. Gina Maira Barbosa de Oliveira

Uberlândia, Minas Gerais  
2017

Dados Internacionais de Catalogação na Publicação (CIP)  
Sistema de Bibliotecas da UFU, MG, Brasil.

---

L732a      Lima, Danielli Araújo, 1989-  
2017      Autômatos celulares e sistemas bio-inspirados aplicados ao controle  
inteligente de robôs [recurso eletrônico] / Danielli Araújo Lima. - 2017.

Orientadora: Gina Maira Barbosa de Oliveira.  
Tese (Doutorado) - Universidade Federal de Uberlândia, Programa  
de Pós-Graduação em Ciência da Computação.  
Modo de acesso: Internet.  
Disponível em: <http://doi.org/10.14393/ufu.te.2020.3011>  
Inclui bibliografia.  
Inclui ilustrações.

1. Computação. I. Oliveira, Gina Maira Barbosa de, 1967-, (Orient.).  
II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em  
Ciência da Computação. III. Título.

CDU: 681.3

---

UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da dissertação intitulada “**Autômatos celulares e sistemas bio-inspirados aplicados ao controle inteligente de robôs**” por **Danielli Araújo Lima** como parte dos requisitos exigidos para a obtenção do título de **Doutora em Ciência da Computação**.

Uberlândia, 10 de Abril de 2017

Orientadora:

---

Profa. Dra. Gina Maira Barbosa de Oliveira  
Universidade Federal de Uberlândia

Banca Examinadora:

---

Prof. Dr. Jefferson Rodrigo de Souza  
Universidade Federal de Uberlândia

---

Prof. Dr. Stéphane Julia  
Universidade Federal de Uberlândia

---

Profa. Dra. Roseli Aparecida Francelin Romero  
Universidade de São Paulo

---

Profa. Dra. Patrícia Amâncio Vargas  
Heriot Watt University



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Data: Abril de 2017

Autor: **Danielli Araújo Lima**  
Título: **Autômatos celulares e sistemas bio-inspirados aplicados ao controle inteligente de robôs**  
Faculdade: **Faculdade de Computação**  
Grau: **Doutorado**

Fica garantido à Universidade Federal de Uberlândia o direito de circulação e impressão de cópias deste documento para propósitos exclusivamente acadêmicos, desde que a autora seja devidamente informada.

---

Autora

A AUTORA RESERVA PARA SI QUALQUER OUTRO DIREITO DE PUBLICAÇÃO DESTE DOCUMENTO, NÃO PODENDO O MESMO SER IMPRESSO OU REPRODUZIDO, SEJA NA TOTALIDADE OU EM PARTES, SEM A PERMISSÃO ESCRITA DA AUTORA.



---

# Dedicatória

---

*Aos meus pais pelo amor incondicional, compreensão, exemplo de vida e apoio irrestrito em todos os momentos de minha vida e ao meu irmão pelo carinho e incentivo.*





---

# Agradecimentos

---

Agradeço...

Em especial a minha orientadora e professora Gina, por nossas discussões, pela paciência e amizade em todos esses anos e, principalmente, por estimular meu senso de independência como pesquisadora e cientista e me dar a liberdade de trabalhar com um tema tão interessante, acreditando sempre no meu potencial. Confiança depositada desde a graduação que, com certeza, me trouxe até aqui.

Aos meus pais e familiares que compreenderam o isolamento necessário para a dedicação a este trabalho. Em especial ao meu irmão que teve a paciência de ouvir algumas das minhas inquietações durante a elaboração desta tese de doutorado durante todos esses anos.

Aos meus amigos do Laboratório de Computação Bio-inspirada (LCBio) e do Grupo de Pesquisa Robótica Bio-Inspirada (Robo-Bio), que contribuíram para a realização deste trabalho a partir do compartilhamento de ideias, e também pela amizade de cada um de vocês.

A todos os meus amigos que me acompanharam durante esses anos, e que realmente admiram o meu trabalho. Gostaria também de agradecer especialmente ao Erisvaldo Fialho pela nossa longa amizade e boas risadas, e também pela paciência e presteza ao longo desses anos, uma pessoa que nunca mediu esforços para me ajudar.

À todos os professores que participaram da minha formação. Aos professores da Fa-

culdade de Computação (FACOM), meu muito obrigada pelo conhecimento transmitido durante todos esses anos. Em especial aos professores que eu tive a oportunidade de conviver no período de doutorado.

Aos professores membros da banca, Dra. Roseli Romero, Dr. Stephane Julia, Dra. Patrícia Vargas, Dr. Jefferson Souza, que aceitaram participar da discussão deste trabalho e pelas valiosas contribuições concedidas.

Ao Instituto Federal do Triângulo Mineiro (IFTM), que concedeu a liberação para a realização do doutorado.

À Faculdade de Computação (FACOM) da Universidade Federal de Uberlândia (UFU), pela estrutura oferecida necessária ao cumprimento deste trabalho.

À Heriot-Watt University (HWU), especialmente na figura da professora Patrícia Vargas pela dedicação de tempo ao processo de mobilidade internacional para experimentos no Robotics Laboratory.

Ao CNPQ e à CAPES pela concessão das bolsas de estudos para os programas de mobilidade internacional - Ciência sem Fronteiras (CsF) e Programa de Doutorado Sanduíche no Exterior (PDSE), respectivamente.

À FAPEMIG, a FACOM e a PROAP-UFU pelo apoio financeiro para a participação e apresentação de artigos científicos em congressos internacionais.

A todos que contribuíram de alguma maneira para a concretização desta tese de doutorado.

*“A imaginação é mais importante que a ciência, porque a ciência é limitada, ao passo que a imaginação abrange o mundo inteiro.”*  
*(Albert Einstein)*



---

# Resumo

---

Em diversas situações, o volume de tarefas a serem cumpridas não pode ser realizado por um único robô. Assim, um campo que tem despertado crescente interesse é a investigação do comportamento de enxame de robôs de busca. Estratégias de cooperação e controle desse enxame devem ser consideradas para um desempenho eficiente do time de robôs. Existem várias técnicas clássicas em inteligência artificial que são capazes de resolver este problema. Neste trabalho um conjunto de técnicas bio-inspiradas, que engloba um modelo baseado em autômatos celulares com memória e feromônio invertido, foi considerado inicialmente para coordenar um time de robôs na tarefa de forrageamento para ambientes previamente conhecidos. Os robôs do time compartilham o mesmo ambiente, comunicando-se através do feromônio invertido, que é depositado por todos os agentes a cada passo de tempo, resultando em forças de repulsão e maior cobertura do ambiente. Por outro lado, o processo de retorno para o ninho é baseado no comportamento social observado no processo de evacuação de pedestres, resultando em forças de atração. Todos os movimentos deste processo são de primeira escolha e a resolução de conflitos proporciona uma característica não-determinista ao modelo. Posteriormente, o modelo base foi adaptado para a aplicação nas tarefas de coleta seletiva e busca e resgate. Os resultados das simulações foram apresentados em diferentes condições de ambiente. Além disso, parâmetros como quantidade e disposição da comida, posição dos ninhos e largura, constantes relacionadas ao feromônio, e tamanho da memória foram analisados nos experimentos. Em seguida, o modelo base proposto neste trabalho para tarefa de forrageamento, foi implementado usando os robôs e-Puck no ambiente de simulação Webots, com as devidas adaptações. Por fim, uma análise teórica do modelo investigado foi analisado através da teoria dos grafos e das filas. O método proposto neste trabalho mostrou-se eficiente e passível de ser implementado num alto nível de paralelismo e distribuição. Assim, o modelo torna-se interessante para a aplicação em outras tarefas robóticas, especialmente em problemas que envolvam busca multi-objetiva paralela.

**Palavras chave:** autômato celular, robótica de enxames, computação bio-inspirada, busca multi-objetiva, dinâmica inteligente, sistemas complexos.



---

# Abstract

---

In several situations, the volume of tasks to be accomplished can not be performed by a single robot. Thus, a field that has attracted growing interest is the behavior investigation of the search swarm robots. Cooperation and control strategies of this swarm should be considered for an efficient performance of the robot team. There are several classic techniques in artificial intelligence that are able to solve this problem. In this work a set of bio-inspired techniques, which includes a model based on cellular automata with memory and inverted pheromone, was initially considered to coordinate a team of robots in the task of foraging to previously known environments. The team's robots share the same environment, communicating through the inverted pheromone, which is deposited by all agents at each step of time, resulting in repulsive forces and increasing environmental coverage. On the other hand, the return process to the nest is based on the social behavior observed in the process of pedestrian evacuation, resulting in forces of attraction. All movements in this process are first choice and conflict resolution provides a non-deterministic characteristic to the model. Subsequently, the base model was adapted for the application in the tasks of selective collection and search and rescue. The results of the simulations were presented under different environment conditions. In addition, parameters such as amount and arrangement of food, nest position and width, pheromone-related constants, and memory size were analyzed in the experiments. Then, the base model proposed in this work for foraging task, was implemented using the e-Puck robots in the simulation environment Webots, with the appropriate adaptations. Finally, a theoretical analysis of the investigated model was analyzed through the graphs and queuing theory. The method proposed in this work proved to be efficient and capable of being implemented at a high level of parallelism and distribution. Thus, the model becomes interesting for the application in other robotic tasks, especially in problems that involve parallel multi-objective search.

**Keywords:** cellular automata, swarm robotics, bio-inspired computing, multi-objective search, intelligent dynamics, complex systems.





---

# Sumário

---

<b>Lista de Figuras</b>	<b>xxi</b>
<b>Lista de Tabelas</b>	<b>xxxii</b>
<b>Lista de Abreviaturas e Siglas</b>	<b>xxxiii</b>
<b>Lista de Símbolos</b>	<b>xxxv</b>
<b>1 Introdução</b>	<b>37</b>
1.1 Motivação . . . . .	41
1.2 Questões de pesquisa . . . . .	41
1.3 Hipótese . . . . .	42
1.4 Objetivos . . . . .	42
1.5 Justificativa . . . . .	43
1.6 Caracterização da Contribuição . . . . .	43
1.7 Organização do trabalho . . . . .	44
<b>I Fundamentação Teórica</b>	<b>45</b>
<b>2 Autômatos celulares</b>	<b>47</b>
2.1 Descrição básica e conceito . . . . .	48
2.1.1 Dimensão e tipos de vizinhança . . . . .	49
2.1.2 Geometria . . . . .	52
2.1.3 Regras de transição . . . . .	53
2.1.4 Modo de atualização das células . . . . .	54

2.1.5	Variações do autômato celular padrão . . . . .	55
2.2	Modelagem de sistemas complexos via autômatos celulares . . . . .	56
2.2.1	Aplicação em sistemas naturais: Formação de padrões naturais e fractais . . . . .	56
2.2.2	Aplicação em sistema físico-ambiental: propagação de incêndios florestais . . . . .	57
2.3	Considerações em relação ao método proposto . . . . .	59
<b>3</b>	<b>Dinâmica coletiva</b>	<b>61</b>
3.1	Inteligência de enxames e dinâmica coletiva . . . . .	62
3.2	Algoritmos de busca . . . . .	63
3.2.1	Algoritmo de Busca Tabu . . . . .	65
3.2.2	Sistemas bio-inspirados . . . . .	66
3.3	Modelos para simulação de dinâmica coletiva . . . . .	68
3.3.1	Principais abordagens . . . . .	68
3.4	Modelos de dinâmica coletiva baseados em ACs . . . . .	74
3.4.1	Modelos de primeira escolha . . . . .	75
3.4.2	Modelos estocásticos . . . . .	83
3.5	Comentários em relação aos modelos propostos . . . . .	91
<b>4</b>	<b>Robótica de enxames</b>	<b>93</b>
4.1	Robótica de enxame e cooperativa . . . . .	94
4.2	Domínios de aplicação na robótica . . . . .	97
4.3	Abordagens para o controle de trajetórias de robôs . . . . .	98
4.4	Trabalhos anteriores em robótica de enxame . . . . .	100
4.5	Trabalhos de robótica baseados em ACs . . . . .	107
4.6	Visão geral dos trabalhos de robótica . . . . .	111
4.7	Arquitetura investigada . . . . .	111
4.8	Ambientes de simulação . . . . .	114
4.9	Implementação da Movimentação do Robô . . . . .	114
4.9.1	Movimentação por Odometria . . . . .	115
4.9.2	Movimentação por GPS . . . . .	115
4.10	Considerações finais . . . . .	116
<b>II</b>	<b>Proposta e Resultados</b>	<b>117</b>
<b>5</b>	<b>Modelos para controle de robôs</b>	<b>119</b>
5.1	Modelo de forrageamento . . . . .	119
5.1.1	Busca no forrageamento . . . . .	122

5.1.2	Captura no forrageamento . . . . .	124
5.1.3	Regresso no forrageamento . . . . .	125
5.1.4	Depósito no forrageamento . . . . .	128
5.1.5	Resolução de conflitos e movimentos do robô . . . . .	128
5.1.6	Variações no modelo de forrageamento . . . . .	129
5.1.7	Comparação entre o modelo proposto e modelos da literatura . . . . .	134
5.2	Modelo de coleta seletiva de objetos . . . . .	136
5.2.1	Busca na coleta seletiva . . . . .	141
5.2.2	Identificação na coleta seletiva . . . . .	143
5.2.3	Captura na coleta seletiva . . . . .	143
5.2.4	Regresso na coleta seletiva . . . . .	144
5.2.5	Depósito na coleta seletiva . . . . .	145
5.3	Modelo de busca e resgate . . . . .	145
5.3.1	Modelagem de propagação do fogo . . . . .	146
5.3.2	Busca na tarefa de busca e resgate . . . . .	149
5.3.3	Identificação na tarefa de busca e resgate . . . . .	150
5.3.4	Captura na tarefa de busca e resgate . . . . .	150
5.3.5	Regresso na tarefa de busca e resgate . . . . .	151
5.3.6	Depósito na tarefa de busca e resgate . . . . .	152
5.3.7	Resolução de conflitos na tarefa de busca e resgate . . . . .	152
5.4	Implementação do modelo no Webots . . . . .	152
5.4.1	Modelo de comunicação para sincronização do time de robôs . . . . .	153
5.4.2	Resolução de conflitos . . . . .	154
5.4.3	Criação de obstáculos virtuais . . . . .	154
5.4.4	Utilização da odometria, GPS e rotação dos robôs . . . . .	155
5.4.5	Processamento de imagens para identificação de objetos e mundo inteligente . . . . .	156
5.4.6	Melhorias na tarefa de forrageamento do Webots . . . . .	157
5.5	Comentários finais sobre os modelos propostos . . . . .	158
<b>6</b>	<b>Análise da formal da proposta</b>	<b>159</b>
6.1	Contextualização sobre cobertura de ambientes . . . . .	159
6.2	Modelagem da busca no algoritmo CAA através da teoria dos grafos . . . . .	162
6.3	Análise de complexidade do algoritmo CAA . . . . .	165
6.4	Razão de aproximação do da busca do modelo CAA segundo o problema da árvore de cobertura . . . . .	167
6.5	Análise do sistema baseada em teoria das filas . . . . .	169
6.6	Análise da complexidade dos métodos SCAAM e FSCAM . . . . .	178

<b>7</b>	<b>Experimentos e análise de resultados</b>	<b>179</b>
7.1	Considerações iniciais . . . . .	179
7.2	Tarefa de forrageamento modelo CAAM . . . . .	180
7.2.1	Resultados de refinamento do modelo CAAM . . . . .	181
7.2.2	Variação do tamanho da fila . . . . .	183
7.2.3	Explorando a configuração do ambiente de forrageamento . . . . .	184
7.2.4	Exploração e cobertura do ambiente . . . . .	193
7.2.5	Desempenho do time e distribuição dos robôs nos ninhos . . . . .	195
7.2.6	Análise visual do efeito de arqueamento . . . . .	196
7.3	Tarefas de forrageamento: modelos PCAAM, D-CAAM e HCAAM . . . . .	198
7.4	Tarefa de coleta seletiva (SCAAM) . . . . .	209
7.4.1	Time homogêneo . . . . .	209
7.4.2	Time heterogêneo . . . . .	213
7.4.3	Cobertura do ambiente para os modelos homogêneo e heterogêneo . . . . .	214
7.5	Tarefa de busca e resgate (FSCAAM) . . . . .	217
7.5.1	Testes de análise estrutural . . . . .	217
7.5.2	Testes de configuração do modelo . . . . .	220
7.5.3	Testes de análise de cobertura do ambiente . . . . .	227
7.6	Experimentos no ambiente Webots (RCAAM) . . . . .	230
7.6.1	Implementação do estado <i>homing</i> . . . . .	230
7.6.2	Modelo com odometria sem manutenção do ângulo de rotação corrente	233
7.6.3	Modelo com odometria e manutenção do ângulo de rotação corrente	235
7.6.4	Modelo com GPS e sem correção de giro . . . . .	236
7.6.5	Modelo com GPS e com correção de giro . . . . .	238
7.6.6	Comparação das abordagens implementadas no Webots . . . . .	240
7.6.7	Melhorias da tarefa de forrageamento no Webots . . . . .	243
7.7	Discussões finais sobre os resultados . . . . .	246
<b>8</b>	<b>Conclusão e trabalhos futuros</b>	<b>249</b>
8.1	Comentários sobre a relevância . . . . .	254
8.2	Respostas às questões de pesquisa . . . . .	256
8.3	Contribuição bibliográfica . . . . .	257
8.4	Trabalhos futuros . . . . .	258
	<b>Referências Bibliográficas</b>	<b>261</b>

---

## Lista de Figuras

---

2.1	(a) Regra de transição de raio 1. (b) Evolução do AC por $T = 2$ passos de tempo (Castro and Lima, 2013a). . . . .	49
2.2	(a) Evolução de um AC unidimensional de raio 1 por $T = 29$ passos. (b) Evolução do AC representado na forma de uma imagem pixelizada em preto e branco (Castro and Lima, 2013a; Silva et al., 2016). . . . .	50
2.3	(a) Vizinhança de Von Neumann. (b) Vizinhança de Moore. (c) Vizinhança arbitrária. (d) Vizinhança aleatória (Castro and Lima, 2013a). . . . .	51
2.4	Evolução do AC Jogo da Vida por $T = 4$ passos de tempo (Castro and Lima, 2013a). . . . .	52
2.5	(a) Vizinhança de Von Neumann 3D. (b) Vizinhança de Moore 3D (Castro and Lima, 2013a). . . . .	52
2.6	Geometrias possíveis das células de um AC: (a) triangular, (b) quadrangular e (c) hexagonal (Castro and Lima, 2013a). . . . .	53
2.7	(a) Estado das células no instante $t$ . (b) Aplicação da regra de transição probabilística no instante $t + 1$ (Castro and Lima, 2013a). . . . .	54
2.8	Exemplificação de uma regra com evolução temporal por $T = 3$ instantes de tempo (Castro and Lima, 2013a). . . . .	54
2.9	(a) Célula vivas no instante $t$ . (b) Aplicação da regra de transição probabilística no instante $t + 1$ (Castro and Lima, 2013a). . . . .	55
2.10	Padrão triangular: (a) exemplo de concha marinha <i>Conus textile</i> com padrão triangular; (b) a representação computacional do padrão natural da concha (Dilão, 1993). . . . .	57

2.11	Autômatos Celulares aplicados em simulação de incêndios florestais: resultado gráfico da implementação do modelo proposto por (Lima and Lima, 2014). . . . .	58
3.1	(a) Estrutura predial 3D. (b) Sobreposição da malha de pedestres na malha de campo de piso (Castro and Lima, 2013a). . . . .	75
3.2	Campo de piso apresentado no modelo de (Varas et al., 2007). Campo de piso para um ambiente com $18 \times 14$ células. A porta de saída está na parede esquerda (células com um valor igual a 1); células correspondentes às paredes é atribuído um valor de 500 (Varas et al., 2007). . . . .	78
3.3	Movimentos possíveis para os pedestres no modelo de (Varas et al., 2007). . . . .	79
3.4	Tipos de arqueamento observados no modelo de (Varas et al., 2007) em ambiente com 50 indivíduos após 30 passos de iteração. (a) Campo de piso proposto por (Varas et al., 2007). (b) Campo de piso sem a inclusão de valor de diagonal. (c) Campo de piso utilizando métrica de Manhattan. . . . .	79
3.5	Exemplo de evacuação no modelo de Alizadeh (2011) onde a maioria dos pedestres escolheram a porta A para a saída. . . . .	81
3.6	Casos da força física de repulsão (Wei-Guo et al., 2006). (a) Caso de repulsão entre quatro agentes em movimento para uma mesma célula. (b) Caso de repulsão entre um agente em movimento e um agente estático. (c) Caso de repulsão entre um agente em movimento e um obstáculo. . . . .	82
3.7	Casos da força física de fricção (Wei-Guo et al., 2006). (a) Caso de fricção entre dois agentes em movimento. (b) Caso de fricção entre um agente em movimento e um agente estático. (c) Caso de fricção entre um agente em movimento e um obstáculo. . . . .	83
3.8	Movimentos possíveis para um pedestre no modelo de (Schadschneider, 2001a) e matriz de preferências. . . . .	84
3.9	Resolvendo conflitos de acordo com as probabilidades relativas para o caso de dois indivíduos com matrizes de preferência $M^{(1)}$ e $M^{(2)}$ (Schadschneider, 2001a). . . . .	85
3.10	Pessoas saindo de uma sala com única porta. Foram apresentados três estágios da dinâmica durante a evacuação (Schadschneider, 2001a). . . . .	87
3.11	Interações entre robôs através da cor projetado no campo de piso (Nishinari et al., 2006). . . . .	90
3.12	Evacuação de pedestres num ambiente em chamas no modelo de (Zheng et al., 2011) . . . . .	91
4.1	(a) Robô equipado com uma pinça para segurar objetos. Os experimentos foram realizados variando-se com 1 a 5 robôs do mesmo tipo (Beckers et al., 1994). (b) Robô Khepera equipado com uma pinça. . . . .	103

4.2	Comportamento do robô no modelo de (Liu et al., 2007). . . . .	104
4.3	Diagrama de um controlador do robô para o cenário de forrageamento simplificado (Lerman et al., 2005). . . . .	105
4.4	Diagrama de um controlador do robô para o cenário de forrageamento simplificado (Winfield, 2009). . . . .	106
4.5	Máquina de estados para o forrageamento com comunicação (Winfield, 2009). . . . .	106
4.6	Disposição espacial dos sensores de proximidade em torno do robô e-Puck (EPFL, 2013). . . . .	112
5.1	Modelo representado em uma máquina de estado finito (Winfield, 2009). . . . .	120
5.2	Construção do ambiente de simulação com obstáculos e um único ninho para a execução da tarefa de forrageamento pelo enxame de robôs (em verde) e um conjunto de alimentos (em laranja) que deve ser procurado e depositado no ninho (Castro and Lima, 2013a). . . . .	120
5.3	Construção do ambiente de simulação com obstáculos e dois ninhos para a execução da tarefa de forrageamento pelo enxame de robôs (vermelho, roxo, verde ou laranja) e um conjunto de alimentos (em azul) que deve ser procurado e depositado no ninho. . . . .	121
5.4	Exemplo de memória com política de fila para o robô da arquitetura e-Puck proposta para o processo de busca (EPFL, 2013). . . . .	124
5.5	Exemplo de formação de filas pelos robôs e criação de zonas de bloqueio. . . . .	125
5.6	Movimentos possíveis para os robôs adaptado do modelo de (Varas et al., 2007). . . . .	129
5.7	Exemplo do cálculo probabilista híbrido onde para células mais distantes do ninho uma escolha mais randomizada é realizada e à medida que os agentes chegam perto dos ninhos os agentes tendem a realizar movimentos mais elitistas, se aproximando mais do modelo determinístico de primeira escolha. . . . .	133
5.8	Exemplo do cálculo probabilista híbrido com cálculo de densidade onde para células mais próximas ao ninho um cálculo de densidade da quantidade de outros agentes é calculada e repassada ao agente. Com esse cálculo é possível encontrar o melhor melhor ninho para se depositar o objeto coletado pelo agente. . . . .	134
5.9	Os robôs em vermelho estão carregando um objeto enquanto que os robôs em verde acabaram de depositar um objeto, os símbolos (★) representam os focos de comida. . . . .	136
5.10	Autômato finito que representa o controlador da tarefa de coleta seletiva para um time homogêneo. . . . .	137

5.11	Autômato finito que representa o controlador da tarefa de coleta seletiva para um time heterogêneo. . . . .	137
5.12	Construção do ambiente de simulação com obstáculos e dois ninhos para a execução da coleta seletiva pelo time de robôs e um conjunto de alimentos (em azul). . . . .	139
5.13	Construção do ambiente de simulação com obstáculos e dois ninhos para a execução da coleta seletiva pelo enxame de robôs e um conjunto de alimentos (em azul). . . . .	140
5.14	Autômato finito que representa o controlador da tarefa de busca e resgate.	145
5.15	Construção do ambiente de simulação com obstáculos e dois ninhos para a execução da tarefa de forrageamento pelo enxame de robôs (vermelho, roxo, verde ou rosa) e um conjunto de alimentos (em azul) que deve ser procurado e depositado no ninho. Os losangos laranja, amarelo e cinza claro estão apresentados os estados que a chama do incêndio dentro do ambiente de simulação modelado através de um autômato celular pode alcançar. . .	147
5.16	Função da secante hiperbólica. . . . .	148
5.17	Função da tangente hiperbólica com $a = 35$ , $b = 15$ e $c = 2$ . . . . .	150
5.18	Diagrama que representa a sincronização de movimento do time de robôs (Lima et al., 2017b). . . . .	153
5.19	Deteção de conflitos do tipo “X” (Lima et al., 2017b). . . . .	154
5.20	Exemplo da estrutura de obstáculos virtuais proposta em (Marchese, 2011).	155
5.21	Ambiente inteligente simulação do mundo com ■ obstáculos e um ninho A, o enxame robô é representado por ■ e unidades de alimentação são representados pela ■, o sensor de comida é representado pela ↑, e cada unidade de alimentação tem apenas um 90° sensor de posição vertical (Castro and Lima, 2013a). . . . .	157
6.1	Redução do problema. (a) Representação real do ambiente com obstáculos e comida. (b) Representação do ambiente através de um grafo não-orientado. (c) Grafo não-orientado com exclusão dos obstáculos. (d) Grafo final com as células de menor valor que representam todas as vizinhanças possíveis de raio 1 do grafo anterior. . . . .	165
6.2	Redução do problema. (a) Grafo com a representação do grafo de cobertura de vértices mínima. (b) Representação do grafo da clique maximal (Sipser, 2006). . . . .	166
6.3	Redução do problema. (a) Grafo com a representação da instância inicial do problema. (b) Árvore para a resolução do problema. . . . .	168
6.4	Redução do problema. (a) Grafo com a representação da instância inicial do problema. (b) Árvore para a resolução do problema. . . . .	169



6.5	Número total de iterações usando $N = \{1, 30\}$ robôs para coletar todos os objetos. . . . .	171
6.6	Número total de iterações usando $N = \{1, 30\}$ robôs para todos os objetos a deposição. . . . .	172
6.7	Número total de iterações usando $N = \{3, 6, 9, 12, 15\}$ robôs para encontrar todos os objetos. . . . .	172
6.9	Mapas de temperatura para cada uma das abordagens de $\beta, \delta'$ utilizando-se $N = \{3, 6, 9, 12, 15, 18, 21\}$ robôs. . . . .	175
6.8	Mapas de passos por célula para cada uma das abordagens de $\beta, \delta'$ utilizando-se $N = \{3, 6, 9, 12, 15, 18, 21\}$ robôs. . . . .	177
7.1	Campo de piso para uma sala com $20 \times 30$ células. A porta de saída está na parede esquerda (células com um valor igual a 1); células correspondentes a paredes é atribuído um valor de 500 (Varas et al., 2007). . . . .	180
7.2	Perspectivas do campo de piso vista através de gráficos de temperatura bidimensional e tridimensional. . . . .	180
7.3	Número médio de execuções para executar o forrageamento com $N = 6$ robôs, $F = 7$ unidades de comida e $\delta = 1$ . $E_1 = \{\alpha = 0, r_v = 0\}$ , $E_2 = \{\alpha = 0, r_v = 1\}$ , $E_3 = \{\alpha = 1, r_v = 0\}$ e $E_4 = \{\alpha = 1, r_v = 1\}$ . . . . .	182
7.4	Gráficos para determinar as diferenças de médias entre o tamanho das filas para dois tipos de simulações referentes à combinação $C_4$ e $C_{16}$ . . . . .	184
7.5	Boxplots para determinar as diferenças entre diferentes números de robôs e iterações no ambiente. . . . .	185
7.6	Efeito de sobrecarga dos ninhos (clogging effect), à medida que aumentamos a quantidade de robôs $N = \{3, 9, 12, 18\}$ . Robôs vermelhos não carregam alimento (estrelas em azul), robôs em verde carregam alimento. . . . .	186
7.7	Regressão linear para robôs entre iterações. . . . .	187
7.8	Boxplots para determinar as diferenças entre um número diferente de comida e iterações no ambiente. . . . .	188
7.9	Regressão linear para variação de unidades de comida durante as iterações. . . . .	188
7.10	Variação de largura de ninhos no campo com $20 \times 30$ células. . . . .	189
7.11	Gráficos para determinar as diferenças entre a largura das portas e o número de iterações. . . . .	190
7.12	Variação de ninhos no campo de piso com $20 \times 30$ células. . . . .	190
7.13	Gráficos para determinar as diferenças entre diferentes posições de pares de portas e número de iterações. . . . .	191
7.14	Diferentes disposições iniciais de comida ( $F = 9$ ) em um piso $20 \times 30$ . . . . .	192
7.15	Resultados da variação do número de robôs, tempo e variação da comida no campo de piso. . . . .	193

7.16	Graficos dos experimentos com a variação $C_4$ , constante de $\delta' = 0.25$ , constante de declínio $\beta = 0.2$ e $\alpha = 5$ e raio de visão $r = 0$ . . . . .	194
7.17	Resultados que mostram a relação entre o desempenho de cada robô em cada tempo. . . . .	196
7.18	Relação entre o fluxo nos ninhos em cada tempo e o número de iterações. . . . .	196
7.19	Relação entre o fluxo nos ninhos e o desempenho de cada robô e o número de iterações. . . . .	197
7.20	Modelo de Varas simulado em uma sala grande com dimensão de $20 \times 30$ e $N = 50$ robôs alocados e tempo total de evacuação foi de $T = 37$ passos de tempo. Os tempos observados na figura são $t = \{1, 5, 10, 15, 30\}$ . . . . .	197
7.21	Modelo de homing simulado em uma sala grande com dimensão de $20 \times 30$ , $\alpha = 5$ e 50 robôs alocados e tempo total de evacuação foi de $T = 34$ passos de tempo. Os tempos observados na figura são $t = \{1, 5, 10, 15, 30\}$ . . . . .	198
7.22	Número total de iterações (em média) usando $N = \{3, 6, 9, 12, 15\}$ . (a) Linha vermelha: modelo determinístico. (B) Linha Azul: PCAAM. . . . .	199
7.23	Iterações $T$ comparação da equipe desempenho $N = 9$ e uso ninho. (a) A linha azul representa o modelo determinista. (b) A linha vermelha representa o modelo PCAAM. . . . .	200
7.24	Arqueamento análise de formação visual e linha, que <span style="color: red;">■</span> representa robôs que buscam, <span style="color: green;">■</span> representa robôs no estado homing <span style="color: blue;">■</span> representa unidades alimentares. (a) Determinístico $N = 9$ . (b) PCAAM $N = 9$ . (c) Determinístico $N = 12$ . (d) PCAAM $N = 12$ . (e) Determinístico $N = 15$ . (f) PCAAM $N = 15$ . . . . .	201
7.25	Número médio total de iterações usando $N = \{3, 6, 9, 12, 15\}$ robôs com diferentes políticas de variação de memória (QP). . . . .	202
7.26	Cada ponto $\{QP_0, QP_1, QP_2, QP_3\}$ representa a média de $5 \times 10^3$ para cada abordagem descrita na Tabela 7.1. . . . .	203
7.27	Número total de iterações (em média) usando Homing com HCAAM e $N = \{3, 6, 9, 12, 15\}$ Robôs de um e variando $d_k^A = \{3, 7, 10, 14, 18\}$ usando $c_{max} = 1000$ . . . . .	203
7.28	Número total de iterações (em média) usando Homing com DCAAM e com $N = \{3, 6, 9, 12, 15\}$ robôs e variando $d_k^A = \{3, 7, 10, 14, 18\}$ com $c_{max} = 1000$ . . . . .	204
7.29	Boxplots para $T$ passos em $N = \{3, 6, 9, 12, 15\}$ para $5 \times 10^3$ pontos: o HCAAM políticas homing e DCAAM, e $c_{max} = \{100, 1000\}$ . . . . .	205
7.30	Gráfico que contrasta todas as abordagens implementadas neste trabalho em termos do desempenho da equipe variando o número de robôs $N = \{6, 9, 12, 15, 18, 21\}$ . . . . .	206
7.31	Variação de métodos implementados comparando a distribuição de ninhos sobre as iterações $T$ usando $N = 15$ robôs. . . . .	208

- 7.32 Iterações  $T$  comparação da equipe desempenho  $N = 9$  e uso ninho. (A) A linha azul representa a abordagem elitista. (B) A linha vermelha representa a abordagem HCAAM. (C) A linha verde representa a abordagem DCAAM. 208
- 7.33 Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de coleta seletiva utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs com variações das constantes para  $\delta, \delta'$  e  $\beta$ . . . . . 211
- 7.34 Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15, 18, 21\}$  robôs, variações nos *searching* e *homing*. . . . . 212
- 7.35 Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs, time heterogêneo com um campo de busca (*HO*), time heterogêneo com dois campos de busca (*HT*). . . . . 213
- 7.36 Mapas de temperatura que apresentam a cobertura do ambiente em relação ao número de passos do time de robôs utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs. A coluna da esquerda representa uma cobertura do time homogêneo, enquanto que a da esquerda representa um time de robôs heterogêneo. . . . 215
- 7.37 Mapas de temperatura que apresentam a cobertura do ambiente em relação à quantidade de feromônio do time de robôs utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs. A coluna da esquerda representa uma cobertura do time homogêneo, as colunas central e da direita representam a matriz de feromônio do para cada um dos tipos de robôs. . . . . 216
- 7.38 Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs, com  $b = 10$  e  $c = 2$ , para as variações da tangente hiperbólica com  $V_1 = 5, V_2 = 10, V_3 = 25, V_4 = 50$  e  $V_5 = 100$ . . . . . 218
- 7.39 Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs, com  $b = 20$  e  $c = 2$ , para as variações da tangente hiperbólica com  $V_1 = 5, V_2 = 10, V_3 = 25, V_4 = 50$  e  $V_5 = 100$ . . . . . 218
- 7.40 Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs, com as seguintes funções para propagação do fogo: tangente hiperbólica  $a = 25, b = 10$  e  $c = 2$  e a secant hiperbólica com  $a = 1, b = 10$  e  $c = 2$ . . . . . 219
- 7.41 Bean plot de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs para as posições de início de foco de incêndio nas posições: noroeste (NW), nordeste (NE), sudeste (SE), sudoeste (SW) e central (CE). . . . . 220

- 7.42 Análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs para as seguintes variações de cosseno hiperbólico:  $a = 0.01, 0.05, 0.1, 0.2$ . . . . . 221
- 7.43 Análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs para as seguintes variações de políticas de fila:  $Q_0 = M_0L_0$ ,  $Q_1 = M_0L_1$ ,  $Q_2 = M_1L_0$  e  $Q_3 = M_1L_1$ ,  $M_i$  representa esvaziar completamente, se  $i = 0$  ou remover, se  $i = 1$ .  $L_i$  representa esvaziar completamente, se  $i = 0$  ou remover, se  $i = 1$ . . . . . 222
- 7.44 Análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = 12$  para as seguintes variações de atualização de  $\delta$ ,  $\delta'$  e  $\beta$ :  $CV_1 = \{\delta = 1, \delta' = 0.25, \beta = 0.01\}$ ,  $CV_2 = \{\delta = 1, \delta' = 0.25, \beta = \cosh\}$ ,  $CV_3 = \{\delta = 1, \delta' = \cosh, \beta = \cosh\}$ ,  $CV_4 = \{\delta = \cosh, \delta' = \cosh, \beta = \cosh\}$  . . . . . 224
- 7.45 Análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = 12$  para o cálculo de densidades utilizando-se as seguintes distâncias para a mudança de cálculo probabilista e visão de ninho  $d_t^A = \{3, 7, 10, 14, 18\}$ . . . . . 225
- 7.46 Análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  para as seguintes combinações:  $D_i$  determinístico,  $P_i$  probabilista elitista,  $V_i$  *homing* com visão de ninho,  $A_i$  *searching* e *homing* com processos estocásticos e  $E_i$  *searching* elitista e *homing* com processos estocásticos. . . . . 226
- 7.47 Análise de variação do tempo  $T$  para a conclusão da tarefa de busca utilizando-se  $N = 9$  para as seguintes combinações: probabilista elitista e estocástico com densidade. . . . . 227
- 7.48 Análise de variação do tempo  $T$  para a conclusão da tarefa de depósito utilizando-se  $N = 9$  para as seguintes combinações: probabilista elitista e estocástico com densidade. . . . . 227
- 7.49 Mapas de temperatura que apresentam a cobertura do ambiente em relação ao número de passos do time de robôs utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs. A coluna da esquerda representa uma cobertura do usando homing probabilista elitista e a coluna da direita representa uma cobertura do usando homing com cálculo de densidades. . . . . 228
- 7.50 Mapas de temperatura que apresentam a cobertura do ambiente em relação à quantidade de feromônio do time de robôs utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs. A coluna da esquerda representa uma cobertura do usando homing probabilista elitista e a coluna da direita representa uma cobertura do usando homing com cálculo de densidades. . . . . 229

7.51	Campo de piso utilizado na simulação do robô e-Puck no software Webots com os valores calculados a partir do algoritmo proposto em (Varas et al., 2007). . . . .	231
7.52	Simulação do robô e-Puck no software Webots $N = 1$ . . . . .	231
7.53	Simulação de 3 robôs do tipo e-Puck no software Webots ( $N = 3$ ). . . . .	232
7.54	Experimentos relativos à tarefa de forrageamento para 6 robôs utilizando-se odometria para localização do robô sem gravação de rotação. . . . .	234
7.55	Experimentos relativos à tarefa de forrageamento para 6 robôs utilizando-se odometria para localização do robô com gravação de rotação. . . . .	237
7.56	Experimentos relativos à tarefa de forrageamento para 6 robôs sem rotação e com utilização de GPS para correção do movimento. . . . .	239
7.57	Experimentos relativos à tarefa de forrageamento para 6 robôs para contrastar a melhoria imposta ao empregar a correção da guinada. . . . .	240
7.58	Experimentos relativos à tarefa de forrageamento para 6 robôs utilizando-se rotação e GPS para correção do movimento. . . . .	241
7.59	Gráfico de comparação do tempo de execução em segundos (s) em relação ao instante de tempo discreto ( $t$ ) do algoritmo da tarefa de forrageamento para cada uma das abordagens implementadas no Webots. . . . .	242
7.60	Instantâneos da simulação RPCAAM em Webots usando e-Pucks (comida é representado em azul): (a) inicial $t = 0$ , (b) $t = 15$ and (c) final $t = 57$ momentos. . . . .	243
7.61	Snapshots Webots (a) desempenho elitista na $t = \{0,6,15,35,59,109\}$ , and (b) performance do modelo RHCAAQ em $t = \{0,7,15,30,40,55\}$ . . . . .	245
7.62	Iterações $T$ , $N = 6$ robôs e uso dos ninhos no Webots. (a) O azul representa a abordagem elitista. (b) O vermelho representa a abordagem RDCAAM. . . . .	246



---

## Lista de Tabelas

---

3.1	Abordagens para modelar evacuação coletiva (Zheng et al., 2009). . . . .	72
3.2	Principais trabalhos de dinâmica coletiva baseados em autômatos celulares. . . . .	76
4.1	Principais trabalhos de robótica coletiva investigados. . . . .	113
6.1	Os dados coletados para verificar a formação de linha perto dos ninhos e a cobertura do ambiente. . . . .	174
7.1	Diferentes combinações de memória durante o processo de pesquisa. . . . .	201
7.2	Distribuição dos ninhos de acordo com cada uma das abordagens investigadas. . . . .	207





---

## Lista de Abreviaturas e Siglas

---

AC	Autômato Celular
AG	Algoritmo Genético
CBI	Computação Bio-Inspirada
IAS	Inverted Ant System
IA	Inteligência Artificial
IC	Inteligência Coletiva
LG	Lattice Gas Automaton
CAA	Celular Automata Ant
CAAQ	Cellular Automata Ant Queue
CAAM	Celular Automata Ant Memory
DCAAM	Density Celular Automata Ant Memory
PCAAM	Probabilistic Celular Automata Ant Memory
HCAAM	Hybrid Celular Automata Ant Memory
SCAAM	Stochastic Celular Automata Ant Memory
FSCAAM	Fire Stochastic Celular Automata Ant Memory
RCAAM	Robot Celular Automata Ant Memory
RCAAQ	Robot Celular Automata Ant Queue
RPCAAM	Robot Celular Automata Ant Memory
RHCAAM	Robot Hybrid Celular Automata Ant Memory
RDCAAM	Robot Density Celular Automata Ant Memory
GPS	Global Processing Sensor
PSO	Particle Swarm Optimization
RNA	Rede Neural Artificial
BT	Busca Tabu

SA	Simulated Annealing
GRASP	Greedy Randomized Adaptive Search Procedure
VNS	Variable Neighborhood Search
IR	Sensor Infra-vermelho
RUR	Rossum's Universal Robots
PIC	Interface Controladora de Periféricos
EPFL	École Polytechnique Fédérale de Lausanne
IEEE	Institute of Electrical and Electronics Engineers

---

## Lista de Símbolos

---

$T$	tempo total de iterações
$N$	quantidade de robôs
$F$	quantidade de comida (objetos)
$Q$	tamanho da memória finita do tipo fila
$t$	tempo de iterações
$t_1$	tempo de iterações do estado queimando do tipo “1”
$t_2$	tempo de iterações do estado queimando do tipo “2”
$r$	raio do autômato celular
$r_v$	raio do campo de visão
$r_p$	raio do depósito de feromônio
$m$	tamanho da vizinhança do AC
$\alpha$	coeficiente para efeito dinâmico
$\delta$	taxa de difusão do feromônio célula central
$\delta'$	taxa de difusão do feromônio células vizinhas
$\beta$	taxa de declínio do feromônio nas células
$\theta_{\max}$	taxa de máxima do feromônio
$\tau_{\max}$	célula com maior número de visitas
$\phi$	quantidade de células no estado queimando
$\varepsilon$	quantidade de células no estado morto
$c_{\max}$	valor de normalização máximo
$c_{\min}$	valor de normalização mínimo



---

## Introdução

---

Um campo que tem despertado crescente interesse nos últimos anos é a robótica, pois é uma área responsável por movimentar a economia e a indústria mundial, devido à sua grande aplicabilidade (Chung and Slotine, 2009). Os robôs são responsáveis por diferentes tarefas em diversos setores industriais, e podem ser empregados em atividades de indústria química, telemedicina, soldagem e até mesmo na manipulação de componentes eletrônicos (Russell and Norvig, 2005). Os robôs podem realizar todas estas tarefas durante horas, com um bom nível de precisão e repetição constante, além das capacidades humanas (Bhattacharya et al., 2012). Contudo, em muitas situações, a maioria desses robôs industriais realizam suas tarefas de uma posição fixa (Saigol et al., 2015) e o conjunto de tarefas necessárias para a realização de um trabalho torna-se difícil ou de alto custo para ser realizado por um único robô de maneira sequencial (Russell and Norvig, 2005). Assim, a robótica tem procurado compensar uma de suas grandes desvantagens que é a falta de mobilidade no ambiente. Essa nova abordagem da robótica é denominada robótica móvel (Ducatelle et al., 2010). Adicionalmente, é interessante o emprego de um conjunto de robôs de baixo custo para a realização de determinadas tarefas (Calvo et al., 2011a). Ou seja, um enxame de robôs móveis de baixo custo que realiza o trabalho de maneira paralela e distribuída, maximizando o desempenho esperado. Esses enxames de robôs podem operar com maior robustez a falhas, possibilitando ao setor da indústria aumento de produtividade com redução de custos com recursos humanos. Assim, esse campo de investigação tem despertado o interesse de diversos pesquisadores nos últimos anos (Bhattacharya et al., 2012), (Purcaru et al., 2013), (Saigol et al., 2015). Em especial, com o advento e popularização de microprocessadores e microntroladores, este campo se

consolidou de maneira definitiva (Calvo et al., 2011a), portanto, as pesquisas nessa área têm se mantido intensas (Ducatelle et al., 2010).

Na robótica móvel objetiva-se fazer com que os robôs realizem comportamentos básicos de movimentação em ambientes complexos, tais como, desvio de obstáculos, exploração e busca (Couceiro et al., 2014a). Para tal, esses robôs devem manter estratégias de cooperação entre si para completarem a tarefa de forma efetiva (Ducatelle et al., 2010). Existem várias técnicas clássicas em inteligência artificial (IA) que são capazes de resolver este problema. Neste trabalho um conjunto de técnicas baseadas em algoritmos bio-inspirados híbridos podem ser elaborados para coordenar a navegação de robôs na aplicação dos principais problemas na área de robótica, visto que a mobilidade é uma capacidade associada aos seres vivos (Gordon, 2014). Dentre as principais tarefas investigadas em robótica móvel podemos citar: planejamento de caminhos, vigilância, forrageamento, coleta e transporte de objetos, entre outros. Dentre as aplicações dessas tarefas podemos citar: transportar equipamentos em indústrias, controlar veículos autônomos, robótica aérea e aeroespacial, competições esportivas entre robôs, coletar de lixo ou outras tarefas que são inóspitas ao trabalho humano (Chung and Slotine, 2009).

Em robótica, a meta é dotar aos agentes um comportamento que resulte na execução da tarefa de forma eficiente (McLurkin and Smith, 2007). A tarefa de forrageamento (Ducatelle et al., 2010), (Couceiro et al., 2014a) é uma aplicação potencial para o desenvolvimento de modelos que descrevem o comportamento individual de cada robô e que é capaz de emergir um comportamento global desejado pelo enxame, que é o cumprimento da tarefa. Outra tarefa que é relevante neste contexto é a vigília, onde os agentes têm por objetivo fazer uma cobertura do ambiente e prover a segurança necessária ao mesmo. Além disso, tarefas centradas em busca e resgate (Das et al., 2003) e coleta seletiva (Vargas et al., 2012) têm tido muita atenção pelos segmentos produtivos, dada a relevância que essas tarefas assumem na sociedade contemporânea.

A partir da análise de estudos anteriores sobre evacuação de pessoas foi possível traçar um paralelo entre a evacuação de pedestres em ambientes com saída limitada e tarefas de forrageamento envolvendo enxames de robôs. Na robótica, o objetivo é alcançar um determinado recurso e pode-se estabelecer um paralelo com a busca pela saída no problema da evacuação de pedestres (Varas et al., 2007). Além disso, os conflitos devem ser controlados entre múltiplos robôs até a meta, uma situação similar aos conflitos entre pessoas e obstáculos que podem mudar a trajetória inicial. Entretanto, no caso da dinâmica de pedestres, o objetivo é encontrar modelos que imitem o comportamento natural exibido pela multidão (Helbing et al., 2000) durante a evacuação.

Neste trabalho, os modelos de evacuação baseados em autômatos celulares (ACs) serviram de inspiração para a elaboração de modelos de navegação de robôs no forrageamento. Os ACs são sistemas dinâmicos discretos no tempo, no espaço e nas suas variáveis. Além disso, já foi provado que os ACs são modelos matemáticos que possuem computabili-

dade universal (Wolfram, 2002a). Sabe-se também que os ACs possuem alto nível de paralelismo, quando implementados em um hardware paralelo (Oliveira, 2003). Existem diversas aplicações no uso de autômatos celulares, tais como, a modelagem de fenômenos naturais, físicos ou biológicos, como alternativa à modelagem por equações diferenciais. Entre os problemas que já foram investigados nessa temática, podemos citar, a evolução biológica dos seres vivos (Powathil et al., 2012), a dinâmica das reações químicas (Sarkar and Abbasi, 2006), espalhamento geográfico (Ortigoza, 2015), os estudos entre força e movimento de corpos da física (Puliafito, 2007) e até mesmo no fluxo de mercados (Huang et al., 2008).

Todas as técnicas inspiradas em sociedades biológicas do tipo enxame vêm sendo reunidas numa área específica denominada Inteligência Coletiva (IC). Dentre as técnicas de IC mais estudadas, as que mais se destacam são as baseadas enxames de partículas e colônias de formigas (Dorigo et al., 1996). O *Particle Swarm Optimization* (PSO) é inspirado pelo comportamento social e cooperativo exibido por várias espécies (bandos de pássaros ou cardumes de peixes) e surge como sendo um algoritmo bastante promissor para a resolução dos mais diversificados problemas de otimização (Kennedy, 2011).

O Ant Colony Optimization (ACO) é uma técnica baseada no comportamento biológico das formigas utilizada na busca e depósito de alimentos. Proposto por (Dorigo et al., 1996) baseia-se na utilização de colônia de formigas artificiais como heurística para otimização de problemas. O algoritmo fundamenta-se na utilização de comunicação indireta global, denominada estigmergia (Calvo et al., 2015). Dessa forma, o enxame de formigas torna-se apto a encontrar o menor caminho entre a comida e o ninho, a partir da força de interação denominada sinergia (Dorigo et al., 2008). As variáveis impostas ao ambiente, tais como, evaporação e distância, fazem com que as formigas se movam para o menor caminho. Em consequência das inúmeras facilidades de implementação computacional e as características presentes nas colônias de formigas artificiais, diversos trabalhos propõem o uso do ACO para a resolução de diferentes problemas, tais como, roteamento de veículos, atribuição de frequência, agendamento, coloração de grafos e robótica (Dorigo et al., 2008). Dentre os trabalhos de robótica baseados em ACO podemos destacar a auto-organização cooperativa (Ducatelle et al., 2010), o planejamento de caminhos (Ioannidis et al., 2011b), (Akbarimajd and Hassanzadeh, 2012).

O algoritmo Inverted Ant System (IAS) (Calvo et al., 2015), (Falleiros et al., 2015) é baseado no ACO com feromônio invertido. No IAS o objetivo do feromônio é criar campos de repulsão, diferentemente do ACO que cria campos de atração (Ducatelle et al., 2010). O algoritmo IAS foi estudado para o problema de vigilância utilizando robôs em ambiente conhecido, de tal forma que os robôs eram capazes de percorrer o ambiente de maneira contínua e repetitiva, fazendo assim, uma cobertura completa do ambiente. A utilização de feromônio repulsivo, uma estratégia similar a (Calvo et al., 2011a), também foi abordada neste trabalho.

Dentre os trabalhos publicados mais recentemente sobre a modelagem de evacuação de pedestres, modelos baseados em Autômatos Celulares (ACs) podem ser ressaltados (Chen et al., 2015), (Feliciani and Nishinari, 2016). Outros trabalhos sobre evacuação usam abordagens baseadas em ACs e PSO (Eberhart and Kennedy, 1995b) para modelagem de evacuação de pedestres (Zheng et al., 2012) e outros usam a combinação de ACs e ACO (Zong et al., 2014). Nestes modelos, os traços do feromônio fornecem um tipo de informação global e distribuída que os agentes usam para tomar decisões e interagir com outros. Outra técnica estudada neste trabalho é a busca Tabu, proposta por (Glover, 1989) que é um método de busca local, que tem como objetivo explorar o espaço de soluções movendo-se de uma solução para outra que seja seu melhor vizinho, a partir de uma estrutura de memória para armazenar as soluções geradas (ou características dessas). Essas características possibilitam a Busca Tabu escapar de ótimos locais (Glover, 1990). Esse algoritmo tem sucesso quando empregado com métodos de otimização por colônias de formigas, evitando a redundância de caminhos previamente percorridos.

Neste trabalho, modelos de dinâmica coletiva baseados em ACs também serviram de inspiração para a criação de um novo modelo para o desenvolvimento de sistemas que envolvem robôs móveis de busca em ambientes complexos com múltiplos recursos. Esse modelo híbrido é baseado em ACs com memória (Tabu) e feromônio invertido (que causam repulsão entre as formigas ao invés de atração), sendo voltado à navegação de robôs móveis e cooperativos. O comportamento autônomo do enxame permite que o comportamento de cada robô se torne mais efetivo e faz com que as tarefas sejam executadas de forma mais eficiente pelo enxame como um todo. Características básicas dos modelos de dinâmica de pedestres propostos em (Varas et al., 2007), (Alizadeh, 2011) foram adotados com algumas importantes adaptações para o contexto das tarefas para a robótica de enxames. Além disso, trabalhos anteriores sobre dinâmica de robôs móveis e cooperativos (Lerman et al., 2005) baseados em ACs com memória (Glover, 1989) e modelos baseados em colônia de formigas (IAS) (Calvo et al., 2012) também foram considerados, resultando em um modelo efetivo na determinação da trajetória dos robôs. O novo modelo de navegação de sistemas multi-robôs foi avaliado na realização de três tarefas distintas. Inicialmente a tarefa de forrageamento foi utilizada para avaliar e refinar as diversas características do modelo. Em seguida essa tarefa foi simulada no Webots com a arquitetura e-Puck, mostrando ser um modelo útil na aplicação em robótica. Posteriormente, as tarefas de coleta seletiva (Vargas et al., 2012) e busca e resgate (Zheng et al., 2011) foram investigadas a partir da adaptação do modelo de forrageamento. Assim, é possível corroborar que o modelo base híbrido proposto nesta tese é flexível a novos parâmetros.



## 1.1 Motivação

A modelagem baseada na dinâmica autônoma de enxames naturais permite que o comportamento dos robôs seja mais inteligente, permitindo que as tarefas por eles executadas sejam finalizadas com menor tempo. As características principais dos modelos de dinâmica coletiva de agentes bidimensionais proposto por (Varas et al., 2007) foram preservadas no modelo de navegação. Além disso, trabalhos sobre dinâmica enxame de robôs de busca móveis e cooperativos baseada em ACs também foram considerados para obtenção de uma menor complexidade em trajetórias robóticas. Para a avaliação do modelo sugerido, testes em diferentes situações foram realizados, permitindo a investigação de sua aplicabilidade.

Esses algoritmos utilizados de forma integrada que combinam: autômatos celulares, busca Tabu, forças sociais observada na evacuação inteligente de pedestres, formigas utilizando feromônio invertido, e robótica do enxame, não foram encontrados na literatura combinados em um único algoritmo para resolver um problema em robótica, tornando o modelo proposto uma abordagem híbrida para cenários de cooperação. Sabe-se que muitos sistemas artificiais tendem a imitar comportamentos que são bem sucedidos na natureza para resolver tarefas específicas de forma cooperativa, tornando este modelo uma abordagem interessante que combina não só um comportamento natural de êxito, mas muitas abordagens coletivas de sucesso.

A composição de estratégias sociais se encaixa em robótica para resolver tarefas de busca multi-objetiva utilizando-se um ambiente com multi-agentes que trabalham cooperativamente através de interações locais e a partir da emergência de um comportamento global do sistema é possível a realização de tarefas complexas. A primeira tarefa investigada nesta tese foi o forrageamento. Esse modelo base foi adaptado para a resolução de outras tarefas clássicas, tais como, a busca e resgate, a coleta seletiva e até mesmo a vigilância. Assim, esse modelo pode ser considerado flexível à entrada de novos parâmetros e outras tarefas de sistemas multi-agentes. Para mostrar a eficiência e flexibilidade, foram realizados testes em software (num ambiente de simulação real), e sabe-se que essas simulações fornecem o tempo para que todos os robôs atinjam um determinado recurso ou meta.

## 1.2 Questões de pesquisa

As questões de pesquisa que orientam esta tese de doutorado, e que posteriormente serão validadas através de experimentos ou análises matemáticas e apresentadas de forma mais simplificada nas contribuições sobre relevância, são:

1. A utilização da dinâmica de pedestres auxilia na tarefa de forrageamento em robótica?

2. Um algoritmo baseado no ferômonio invertido de formigas resolve a busca dos agentes de forma completa?
3. A utilização de uma técnica híbrida maximiza o desempenho da busca com multiagentes?
4. A partir da formulação do modelo base para a tarefa de forrageamento é possível adaptá-lo para outros problemas de busca em robótica, tais como, vigília, coleta seletiva e busca e resgate?
5. Um algoritmo aproximativo híbrido resolve o problema da cobertura de vértices na busca da tarefa de forrageamento?

### 1.3 Hipótese

A partir da combinação de técnicas bio-inspiradas e comportamentos sociais, autômatos celulares com memória e técnicas tradicionais de robótica coletiva é possível modelar um algoritmo híbrido para resolver coordenar um time de robôs capaz de realizar tarefas de cobertura e busca na área de robótica para ambientes conhecidos. Assim, evita-se o uso do algoritmo de cobertura de vértices não-polinomial (NP) que resolveria a cobertura do ambiente pelos robôs de forma ótima, mas com tempo de execução alto para a busca dessa solução.

### 1.4 Objetivos

Este trabalho tem foco teórico e prático e tem por objetivo geral a criação de um algoritmo híbrido baseado em técnicas bio-inspiradas para o controle de um time de robôs que execute com eficiência a tarefa de forrageamento. Os objetivos específicos são: (i) implementar um novo modelo para a dinâmica de enxame de agentes, a partir da literatura correlacionada, posteriormente, (ii) investigar a aplicabilidade dos modelos de dinâmica populacional baseados em ACs em robótica e adaptá-los para controle de enxame de robôs móveis e cooperativos que disponham de um único recurso, sendo que, as tarefas que vislumbramos como aplicação são: o forrageamento, coleta seletiva e a busca e resgate, (iii) fazer uma análise sob o ponto de vista teórico do sistema de ferômonio invertido para a cobertura do ambiente para descobrir a quantidade de robôs próxima do ótimo que resolve o problema de busca, e (iv) implementar o modelo em um ambiente de simulação robótica e analisar a partir de simulações aspectos de complexidade matemático-computacional de autômatos celulares para serem aplicados à dinâmica inteligente de enxame de robôs móveis cooperativos, mais especificamente à tarefa de busca em ambientes complexos e a comparação desta nova estratégia de busca com trabalhos precursores baseados em ACs.

## 1.5 Justificativa

Os ACs são vistos, não como substitutos dos modelos matemáticos tradicionais, mas como sistemas com poder de representatividade equivalente. Conhecendo o potencial de paralelismo dos ACs, os mesmos tornam-se uma opção interessante para a modelagem discreta de problemas complexos que não podem ser resolvidos em tempo polinomial, como a cobertura de vértices, trazendo uma solução aproximada. Neste caso, os ACs incorporados à modelos de dinâmica de pedestres juntamente com outros sistemas bio-inspirados, e em seguida a sua aplicação em modelagem de sistemas de robôs de busca móveis e cooperativos será o principal objetivo deste trabalho. A partir do modelo proposto, espera-se que o mesmo possa ser utilizado em algumas das principais tarefas investigadas neste campo de pesquisa. Além disso, espera-se que o mesmo sirva como um boa alternativa de algoritmo de aproximação discreto e paralelizável para resolver problemas de otimização não-polinomiais difíceis na teoria dos grafos, como no caso da tarefa de cobertura. Os resultados obtidos através da simulação via autômatos celulares podem confirmar hipóteses para uma posterior formulação de um modelo matemático discreto baseado em grafos para a robótica.

## 1.6 Caracterização da Contribuição

Com a realização deste trabalho foi possível traçar uma revisão da literatura de maneira detalhada buscando encontrar os principais modelos que norteiam a temática de modelagem de dinâmica inteligente de enxames com ACs, tanto para a modelagem de dinâmica populacional quanto para a robótica. A partir dessa análise correlata, foi possível construir um controlador de enxames de robôs para a arquitetura e-Puck e contrastar com modelos precursores de dinâmica populacional. Uma memória auxiliar foi inserida no modelo a fim de aumentar o desempenho do time de robôs. Essa memória tornou o modelo muito eficiente, aumentando o espalhamento do time e por sua vez, melhorando o tempo de busca dos objetos. A primeira tarefa robótica implementada foi o forrageamento, muitos testes de desempenho foram realizados para investigar a performance do modelo em relação à modelos precursores. A partir dessa análise dois outros modelos de tarefas foram implementados para esta tese: a coleta seletiva e a busca e resgate, além do trabalho de vigília, não detalhado neste documento. Adicionalmente, foi possível através de uma análise baseada em teoria dos grafos definir o número ótimo de robôs que realizam a tarefa de busca a partir de uma configuração específica de ambiente. Por fim, uma análise do modelo baseada em teoria das filas buscou entender o gargalo da busca competitiva pelos ninhos, o que gera um atraso no depósito e que é inerente ao ambiente multi-agentes investigado.

## 1.7 Organização do trabalho

Este trabalho está dividido em três partes e em oito capítulos com o objetivo de transmitir as informações necessárias ao entendimento, contextualização e definição do sistema de navegação de robôs proposto, além de apresentar resultados de alguns dos testes efetuados.

A primeira parte relaciona-se aos principais conceitos de fundamentação teórica e alguns trabalhos correlatos presentes na literatura para o entendimento do modelo proposto neste trabalho.

No Capítulo 2 são apresentados os principais conceitos envolvendo autômatos celulares. Dentre esses conceitos, são abordados algumas variações de autômatos celulares padrão que servirá para o entendimento do sistema aqui proposto.

O Capítulo 3 resume os principais modelos de evacuação de pedestres e apresenta uma compilação dos modelos já estudados. Também define os principais conceitos sobre comportamentos sociais já abordados na literatura.

Os principais conceitos de robótica coletiva são mostrados no Capítulo 4. Algumas referências dos principais modelos utilizados e softwares para a simulação também são apresentadas com o intuito de contextualizar as ideias prévias utilizadas neste trabalho.

A segunda parte refere-se às investigações e propostas desenvolvidas neste trabalho e está organizada da seguinte maneira.

No Capítulo 5 o modelo proposto é apresentado bem como todas as tarefas que foram investigadas. Por fim, as principais adaptações do modelo proposto são apresentadas para a implementação do mesmo no ambiente de simulação.

No Capítulo 6 uma análise sob o ponto de vista teórico é apresentada através da Teoria dos Grafos e das Filas para verificar se a técnica abordada neste trabalho realiza a tarefa projetada para o time de robôs de maneira eficiente.

O Capítulo 7 apresenta os resultados dos experimentos realizados com o modelo proposto com o propósito de avaliar sua robustez e refinar o conjunto de parâmetros a serem utilizados. Adicionalmente, uma análise dos resultados é realizada tanto para as simulações estatísticas quanto para as que foram feitas no ambiente Webots para a arquitetura com os robôs e-Puck.

As conclusões e considerações finais são relatadas no Capítulo 8, as principais contribuições bibliográficas e também algumas investigações não contempladas neste trabalho que são sugeridas como continuidade do mesmo.

Parte I

Fundamentação Teórica



---

## Autômatos celulares

---

Autômatos celulares (ACs), do inglês *Cellular Automata*, podem ser entendidos como um conjunto de unidades processadoras simples (células) que interagem entre si, com conectividade local, e ao longo do tempo apresentam um comportamento global. O estudo dos ACs atraiu o interesse de pesquisadores nos últimos anos, devido à sua capacidade de gerar padrões complexos, que podem ser observados na natureza, a partir da aplicação de regras simples (Lima, 2012).

Os ACs foram estudados inicialmente por Ulam e Von Neumann para projetar mecanismos artificiais de auto-reprodução (von Neumann and Burks, 1966) no começo dos anos 50. Naquele momento, von Neumann tentava criar um sistema que se replicasse. Ulam, que estava estudando a formação de cristais por meio de um modelo de uma malha e células, sugeriu-lhe usar esse tipo de modelo. O autômato celular que ele utilizou para demonstrar sua hipótese permitia a uma célula ter 29 estados (von Neumann and Burks, 1966). Mais tarde, Jonh Conway investigou a existência de um AC simples com computabilidade universal (Wolfram, 2002b). No entanto, as suas regras nunca foram implementadas num computador por serem muito complexas. O resultado mais simplificado da pesquisa foi o Jogo da Vida (*Game of Life*), um AC bidimensional onde as células no estado 1 são ditas células “vivas” e as células no estado 0 são ditas células “mortas” (Berlekamp et al., 1992). Os ACs se tornaram mais populares depois do lançamento do livro *A New Kind of Science* (Wolfram, 2002b). Um dos resultados mais importantes das pesquisas de (Wolfram, 2002b) foi o estudo de comportamento de dinâmico dos ACs.

Existem diversas aplicações no uso de autômatos celulares, dentre elas, podemos citar a modelagem de fenômenos naturais, físicos ou biológicos que seriam muito difíceis de

serem modelados pelas equações diferenciais (Lima, 2011), que são as mais utilizadas nesse tipo de tarefa. Alguns dos principais trabalhos conhecidos nessa linha de pesquisa foram: formação de padrões em conchas e peles de animais (Wolfram, 1986), (Markus and Kusch, 1995), (Wolfram, 2002b), escoamento de fluidos (Frisch et al., 1986), propagação de incêndios em florestas (Hernandez et al., 2007), espalhamento de fluidos (Lima et al., 2016b), infecções provocadas por vírus (Santos et al., 2001), evacuação emergencial de pessoas (Varas et al., 2007), formação de cristais (Wolfram, 2002b). Outras tarefas não ligadas à modelagem de fenômenos naturais também têm sido investigadas, tais como, aplicação em criptografia (Silva et al., 2016), na combinação de ACs com o uso de memória de estado para a aplicação em clusterização (Moere and Clayden, 2005), (Alonso-Sanz, 2009), e na tarefa de escalonamento paralelo (Kucharska et al., 2016).

No presente trabalho, o foco está na aplicação de ACs como método para a dinâmica de enxames de robôs. As evoluções espaço-temporais mesmo utilizando-se regras simples podem dar origem a um comportamento dinâmico diversificado e complexo. Essa característica foi aproveitada na proposição de diversos métodos baseados em ACs. Os ACs bidimensionais estocásticos foram utilizados para a proposição de um modelo de busca e coleta de objetos especialmente adequado às tarefas de forrageamento, seleção e coleta de objetos e busca e resgate em ambiente com incêndio. O novo modelo proposto neste trabalho é fortemente baseado nos modelos precursores de dinâmica populacional propostos em (Varas et al., 2007), (Alizadeh, 2011) e algumas adaptações pertinentes a este contexto foram elaboradas a partir do trabalho de (Calvo et al., 2011a).

## 2.1 Descrição básica e conceito

Um AC é composto por um reticulado de dimensão  $d$  dividido em células, sendo que, cada célula contém um símbolo ou estado. As células modificam seus estados a cada passo de iteração de acordo com uma função de transição. Podemos aplicar a função de transição por  $T$  passos de tempo para obter a evolução espaço-temporal do reticulado do AC. Essa função de transição também é chamada de regra de transição. A regra de transição indica o novo símbolo a ser escrito na célula do reticulado de acordo com seu estado atual e dos estados de suas vizinhas (regra local). Em sua definição mais usual, a atualização dos estados se dá de forma síncrona, isto é, a cada passo de tempo todas as  $N$  células do reticulado são atualizadas (Lima, 2012).

A implantação computacional de um autômato celular segue uma classificação caracterizada pelos seguintes fatores: a dimensão, a geometria, o formato das células, os tipos de regra e o tipo de vizinhança para a interação entre as células.



### 2.1.1 Dimensão e tipos de vizinhança

A dimensão espacial implica na forma como as células estão distribuídas pelo reticulado. Para a configuração espacial de um AC, o reticulado deve possuir geometria regular, tendo todas as células o mesmo tamanho. A dimensão refere-se ao seu aspecto perceptível entre angulações e coordenadas simétricas (Silva et al., 2016). Um AC pode ser unidimensional (1D), bidimensional (2D), tridimensional (3D) ou  $n$ -dimensional ( $nD$ ), desde de que  $n$  seja inteiro positivo,  $n \geq 1$  (Castro and Lima, 2013b).

#### Autômatos celulares unidimensionais

A dimensão dos ACs mais estudada é a unidimensional, na qual um novo estado depende de  $m$  células vizinhas, tal que  $m = (2r + 1)$  e  $r$  é o raio do AC. O reticulado de tamanho  $N$  pode ser representado, em termos computacionais, por um vetor.

A Figura 2.1 (b) apresenta um reticulado de 6 células ( $N = 6$ ) sendo que o estado inicial de cada célula é apresentado em  $t = 0$ . A Figura 2.1 (a) mostra uma regra binária de raio 1 na qual a vizinhança de cada célula é formada por três elementos: a própria célula e suas duas vizinhas imediatas (à esquerda e à direita). Como esse AC é binário (2 estados possíveis), existem 8 diferentes vizinhanças, da 000 a 111. A regra em si é dada pelos 8 bits de saída associados a cada vizinhança possível:  $\{01111000\}$ . Na Figura 2.1 (b) vemos a atualização do reticulado por 2 passos de tempo a partir de sua configuração inicial  $\{101110\}$  em  $t = 0$ . A cada passo, cada célula do reticulado é atualizada identificando-se sua vizinhança e seu novo estado é dado pelo bit de saída correspondente na regra de transição. Por exemplo, na Figura 2.1 (b) a vizinhança da quinta célula é destacada: 110. De acordo com o bit de saída regra de transição, o estado da quinta célula deve ser atualizado para 0 ( $110 \rightarrow 0$ ) no próximo instante de tempo. O reticulado é submetido a condições periódicas de contorno, sendo que a primeira célula é vizinha imediata da última, e vice-versa. Aplicando-se esse procedimento para todas as células do reticulado de forma síncrona tem-se a nova configuração do reticulado em  $t = 1$ . Se a regra for aplicada novamente em todas as células, obtém-se a configuração apresentada em  $t = 2$  (Lima, 2011).

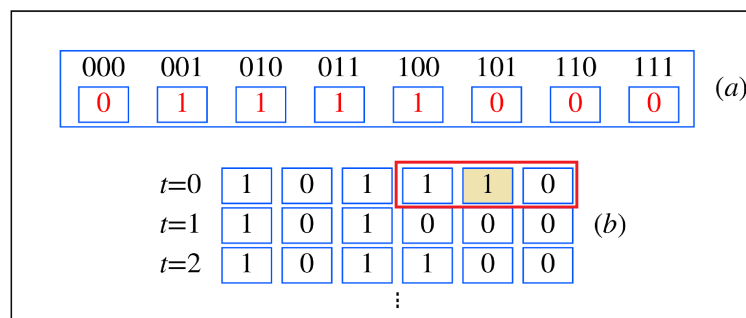


Figura 2.1: (a) Regra de transição de raio 1. (b) Evolução do AC por  $T = 2$  passos de tempo (Castro and Lima, 2013a).

Outra forma de apresentar a evolução espaço-temporal de um reticulado de um AC unidimensional binário é aquela em que as células no estado 0 são representadas por pixels brancos, enquanto que as células no estado 1 são representadas por pixels pretos, conforme é mostrado na Figura 2.2. Nessa figura, a regra da Figura 2.1 (a) foi aplicada em um reticulado de 16 células, partindo da configuração inicial  $\{1011110011010110\}$ . Observe na figura que a evolução é feita a partir da aplicação da regra no reticulado inicial, sempre inicializando essa evolução de cima para baixo. Este tipo de visualização da evolução espaço-temporal das células é largamente utilizado, pois facilita a identificação de padrões associados à dinâmica dos ACs.

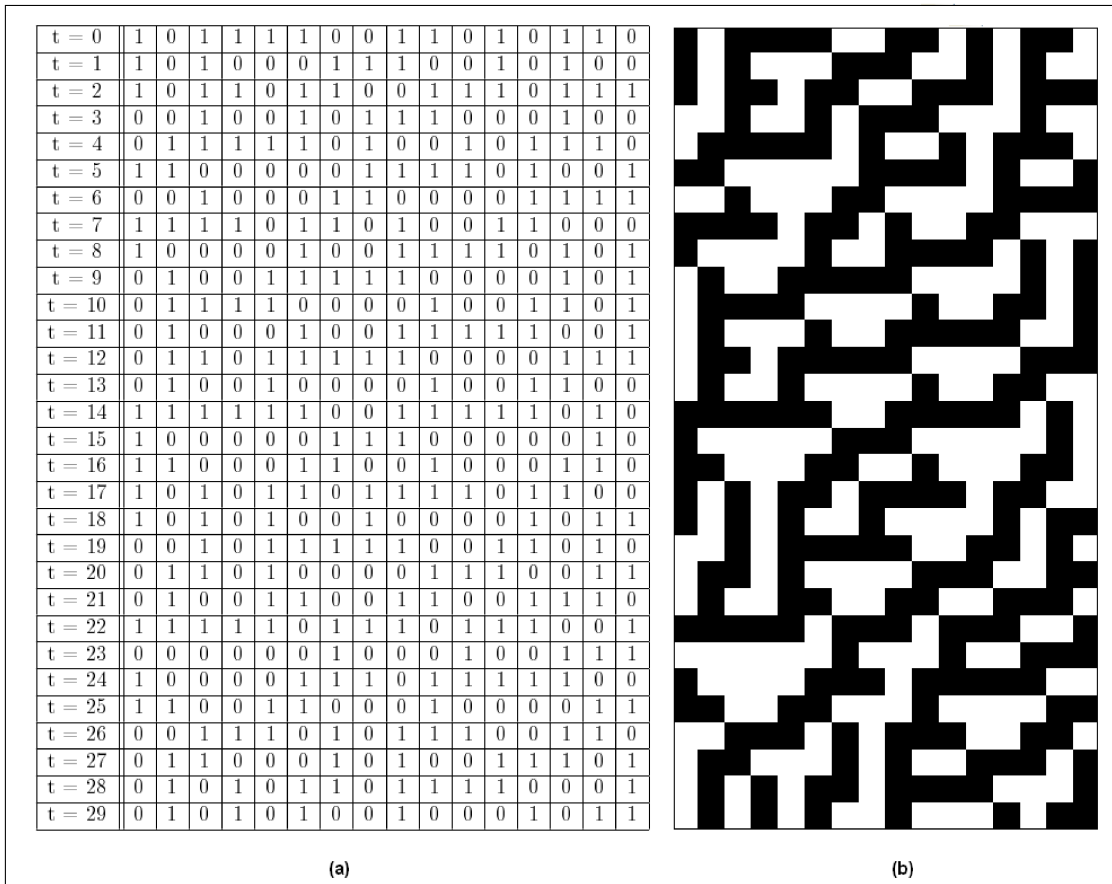


Figura 2.2: (a) Evolução de um AC unidimensional de raio 1 por  $T = 29$  passos. (b) Evolução do AC representado na forma de uma imagem pixelizada em preto e branco (Castro and Lima, 2013a; Silva et al., 2016).

Um AC unidimensional é definido formalmente por seu espaço celular e por sua regra de transição. O espaço celular é um conjunto de células idênticas e justapostas em um arranjo unidimensional (1D). Uma regra de transição fornece o próximo estado de cada célula. O conjunto de estados possíveis é denotado por  $\Sigma$  e o número de elementos desse conjunto é denotado por  $k$ . Cada célula é denotada por um índice  $i$  e seu estado, em um determinado tempo  $t$ , é denotado por  $S_i^t$ , tal que  $S_i^t \in \Sigma$ . O estado da célula  $i$  juntamente com o estado das outras às quais ela está conectada é denominado vizinhança da célula  $i$ , que é denotado por  $\eta_i^t$ .  $\Phi(\eta_i^t)$  é a regra de transição para uma dada célula  $i$ . Seja  $r$

tamanho do raio do AC 1-dimensional,  $\eta_i^t$  tem tamanho  $m = 2r + 1$ , assim, temos que  $\Phi: \Sigma_m \rightarrow \Sigma$  (Mitchell et al., 1997).

O tamanho de uma regra é dado por  $n = k^m$  e o espaço de regras é dado por  $k^n$ , que representa o conjunto de todas as regras possíveis que podem ser geradas para um determinado tamanho de raio. Quanto maior for o raio de um AC maior será a quantidade de regras distintas que poderão ser geradas (cardinalidade do espaço de regras). Por exemplo, em ACs unidimensionais binários de raio 2, o tamanho da vizinhança é  $m = 5$  e as regras possuem  $n = 32$  bits e o espaço de regras tem cardinalidade  $2^{32}$  bits.

Os autômatos celulares mais estudados, são os ACs elementares, que possuem raio 1 e alfabeto  $\Sigma \in \{0, 1\}$ . Dessa forma, a regra é definida por 8 bits e existem 256 regras elementares possíveis.

### Autômatos celulares bidimensionais

Os ACs fundamentados em duas dimensões (2D ou bidimensionais) são amplamente conhecidos na literatura (Wolfram, 2002b). O modelo mais conhecido é o jogo da vida (*Life*) que foi proposto por (Conway, 1970). Os reticulados bidimensionais são representados no plano e em termos de estruturas computacionais em uma matriz. Os ACs bidimensionais também são muito usados para a representação de padrões gráficos, já que sua disposição assemelha-se a uma imagem pixelizada. Essa dimensão contempla variadas formas para a consideração da vizinhança de uma célula, sendo que as duas principais estão apresentadas na Figura 2.3 (a) e (b), tal que o símbolo  $\odot$  representa a célula central e as demais células, demarcadas com o símbolo  $\blacksquare$ , adjacentes à central são denominadas células vizinhas. Outro caso é a vizinhança arbitrária e a aleatória apresentadas, respectivamente, na Figura 2.3 (c) e (d).

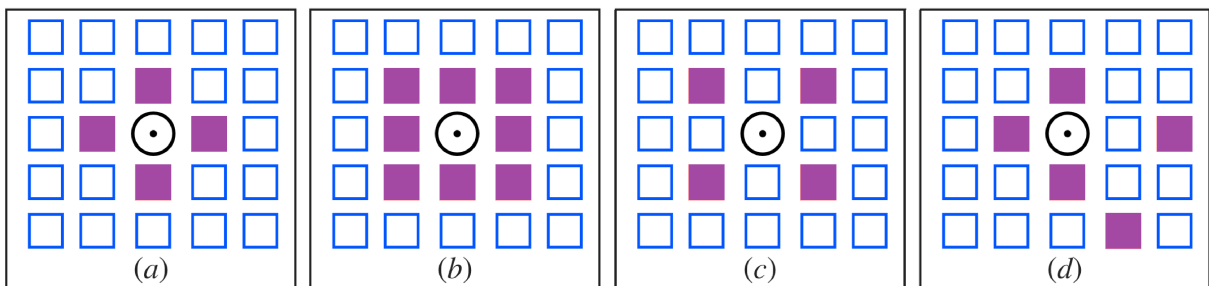


Figura 2.3: (a) Vizinhança de Von Neumann. (b) Vizinhança de Moore. (c) Vizinhança arbitrária. (d) Vizinhança aleatória (Castro and Lima, 2013a).

A vizinhança é um fator considerável na definição de um AC, pois durante a decorrência das interações, uma célula varia seu estado não só com base em seu próprio símbolo, mas muitas vezes é influenciada por sua circunvizinhança. Dentre as aplicações de ACs bidimensionais com regras determinísticas, o modelo mais conhecido é o jogo matemático proposto por (Conway, 1970) denominado *Game of Life* é apresentado na Figura 2.4 (Castro and Lima, 2013a). A figura apresenta a evolução de um AC bidimensional por

4 passos de tempo a partir da configuração inicial mostrada no instante de tempo  $t = 0$ , utilizando a regra do jogo da vida. A regra de transição para a evolução do jogo da vida considera a vizinhança de Moore e tem o seguinte enunciado: “O estado da célula central será 1 (células pretas) no próximo instante de tempo se: o estado atual da célula é 1 e duas ou três células da vizinhança for 1 ou se a célula central é 0 e duas ou três células vizinhas forem 1. Caso contrário, a célula central assumirá valor 0 (células brancas)”.

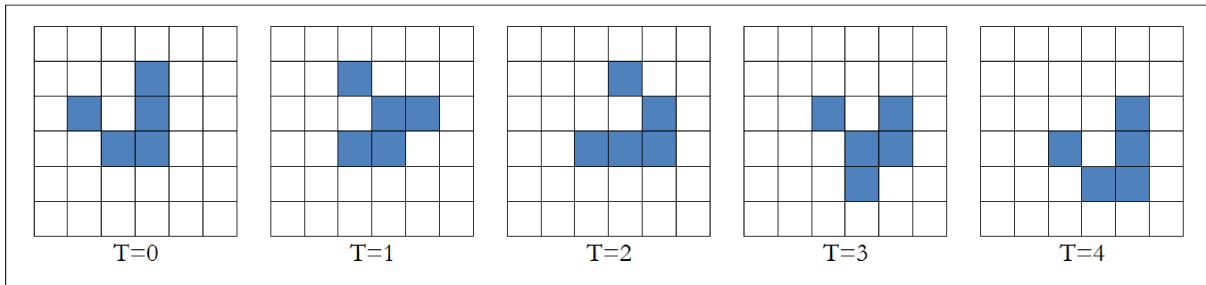


Figura 2.4: Evolução do AC Jogo da Vida por  $T = 4$  passos de tempo (Castro and Lima, 2013a).

### Autômatos celulares tridimensionais

A dimensão de um autômato pode ser estendida também para a terceira dimensão (Lima and Oliveira, 2015). No entanto trabalhos relacionados a ACs 3D são menos comuns, visto sua complexidade de manipulação. As vizinhanças de Moore e de Von Neumann podem ser estabelecidas nesta dimensão seguindo a mesma ideia vista em 2D, conforme nota-se na Figura 2.5.

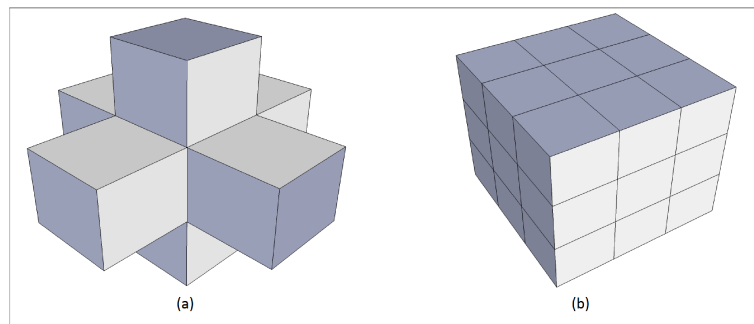


Figura 2.5: (a) Vizinhança de Von Neumann 3D. (b) Vizinhança de Moore 3D (Castro and Lima, 2013a).

### 2.1.2 Geometria

A geometria aplica-se à forma de cada célula e alguns dos exemplos podem ser identificados na Figura 2.6. Aqui são apresentadas três formas (triangular, quadrangular e hexagonal), pois matematicamente as figuras não devem exceder a soma de  $360^\circ$  na soma dos ângulos externos dos encaixes das células, para que não haja sobreposição de unidades (Silva et al., 2016). A geometria quadrangular é a mais utilizada e conhecida na literatura.

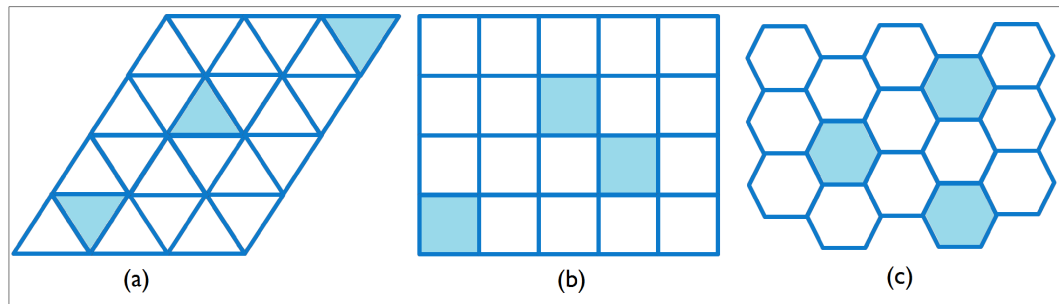


Figura 2.6: Geometrias possíveis das células de um AC: (a) triangular, (b) quadrangular e (c) hexagonal (Castro and Lima, 2013a).

### 2.1.3 Regras de transição

A evolução de um autômato celular é desencadeada de acordo com seu estado inicial. O estado inicial pode ser obtido com uma formatação pré-estabelecida ou probabilística. Mas pequenas alterações na evolução espaço-temporal de uma simulação podem gerar grandes efeitos nas interações que as sucederão, causando variações de resultado (Castro and Lima, 2013a).

As regras ou função de transição são plausíveis na regência do estado ou símbolo de uma célula e das células vizinhas. Uma célula no instante  $t$  deve alterar seu estado, assumindo outro valor em  $t + 1$ , de acordo com uma regra de transição. O modelo de atualização de estados pode ser classificado seguindo seu aspecto determinístico, probabilístico ou temporal (Silva et al., 2016). Em um AC, uma única função de transição pode fazer todo o controle da evolução espaço-temporal do reticulado (Castro and Lima, 2013a). Contudo, podem haver casos em que diferentes regras atuam na atualização dos estados.

#### Regras determinísticas

O modelo de regras determinísticas assume que é possível alterar o estado de uma célula quando se conhece os estados das células vizinhas. Dessa forma, a célula em observação ou célula alvo sofre influência do ambiente que a cerca, e a regra por sua vez rege sua alteração (Castro and Lima, 2013a). Esse modelo é chamado de determinístico porque pode-se dizer exatamente o próximo estado dessa célula a partir dos estados de sua vizinhança (Lima, 2012). O AC apresentado na Figura 2.1 apresenta uma regra do tipo determinística, pois a partir de uma vizinhança qualquer é possível saber exatamente o estado da célula central no próximo instante de tempo (Silva et al., 2016).

#### Regras probabilísticas

O modelo de regras probabilísticas assume que é possível alterar o estado de uma célula  $x$  a partir de uma função de probabilidade ( $P(x)$ ). Nesse caso, as regras de atualização são probabilísticas e, o estado de uma célula em um determinado instante depende probabilis-

ticamente dos estados de suas células vizinhas no passo de tempo imediatamente anterior (Lima et al., 2016a). A Figura 2.7 exemplifica uma regra probabilística onde uma célula central  $x$  tem seu estado definido por um número sorteado  $y$  a partir da configuração da vizinhança no instante  $t$  (Castro and Lima, 2013a). O enunciado da regra é definido por: se a célula central  $x$  está no estado  $\blacksquare$  e as  $m$  vizinhas estão no estado  $\square$ , ela tem uma probabilidade não nula  $P(x)$  de no estado de tempo  $t + 1$  alterar seu estado para  $\square$ , caso o sorteio de  $y$  esteja definido  $y \leq 25$ . Por outro lado, se o sorteio tiver entre  $y > 25$ , o estado da célula central permanece o mesmo  $\blacksquare$ .

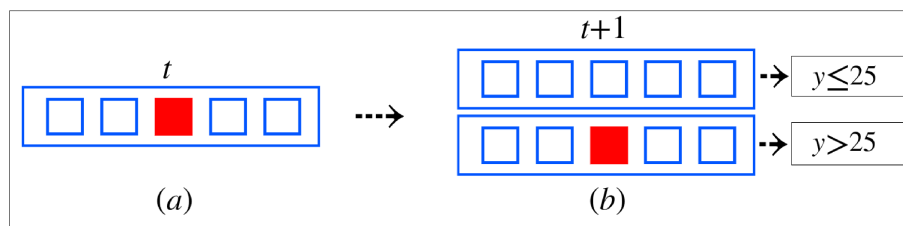


Figura 2.7: (a) Estado das células no instante  $t$ . (b) Aplicação da regra de transição probabilística no instante  $t + 1$  (Castro and Lima, 2013a).

### Regras temporais

O modelo de regras temporais assume que é possível alterar o estado de uma célula central a partir de uma função temporal, ou seja, após  $t$  iterações de tempo transcorridas é possível alterar o estado da célula sem observar a sua vizinhança. Neste caso, a alteração da célula central não depende das vizinhas, e sim de uma função temporal previamente definida (Castro and Lima, 2013a). Por exemplo, “uma célula no estado  $\blacksquare$  pode permanecer neste estado por  $t = 2$  instantes de tempo e então ela vai para o estado  $\odot$ ”, conforme está ilustrado na Figura 2.8.

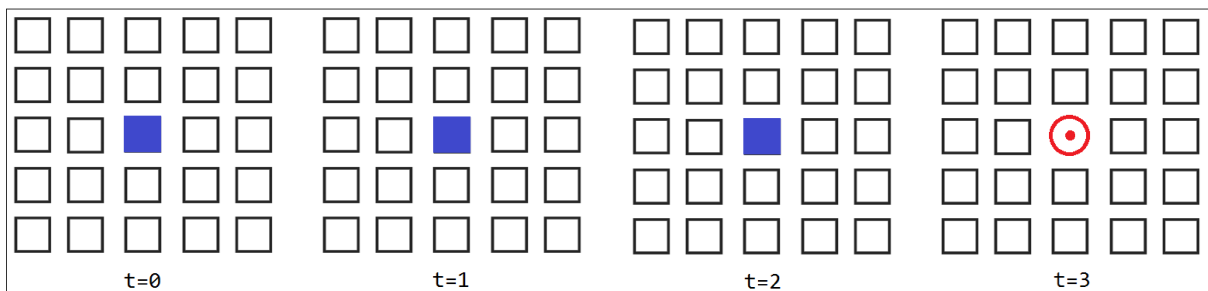


Figura 2.8: Exemplificação de uma regra com evolução temporal por  $T = 3$  instantes de tempo (Castro and Lima, 2013a).

#### 2.1.4 Modo de atualização das células

A atualização das células do reticulado também pode ser efetuada de diferentes maneiras. Existem dois tipos de atualização de células: síncrona e sequencial (Castro and

Lima, 2013a). A atualização síncrona das células ocorre quando todas as células do reticulado são atualizadas ao mesmo tempo (de forma paralela), sendo esta a mais usual. A atualização pode ser realizada também de forma sequencial, na qual a atualização de uma célula depende da atualização prévia do estado da sua célula antecessora no reticulado unidimensional (Silva et al., 2016).

### 2.1.5 Variações do autômato celular padrão

A construção dos modelos de ACs possuem algumas variações que se diferencia do modelo padrão apresentado. Por exemplo, a condição de contorno determina como as células da vizinhança nos extremos de um reticulado serão preenchidas (Silva et al., 2016). A vizinhança das células do extremo do reticulado pode ser estabelecida de diferentes formas, sendo a mais conhecida chamada de condição de contorno periódico que foi empregado na Figura 2.1. Nessa figura, temos que a primeira célula do reticulado é vizinha da última e a última célula é vizinha da primeira (Castro and Lima, 2013a). Outra condição de contorno que pode ser adotada na atualização das células é chamada de condição de contorno nula, na qual a última e a primeira células do reticulado possuem células no estado zero como vizinhas (Lima, 2010). No modelo proposto nesta tese, as células da borda do reticulado representam as paredes, e nesse caso, temos uma condição de contorno não-periódica.

Em muitos cenários de aplicação de um AC com regras diferentes das regras padrão se faz necessária (Castro and Lima, 2013a). Um exemplo de um AC que não é padrão pode ser definido quando a alteração da célula central provoca também a alteração das células vizinhas (Lima, 2010). No exemplo apresentado na Figura 2.9 uma regra probabilística com deslocamento é apresentada. Neste caso, tem-se que a célula central no instante  $t$  que está no estado ■ pode ir para o estado □, caso a variável  $y$  que representa a probabilidade  $P(x)$  for  $y \leq 25$  e  $y \geq 75$ . Observe que o comportamento da célula vizinha também foi alterado. Neste caso, quando o valor sorteado  $y \leq 25$ , existe um deslocamento à esquerda, e quando o valor sorteado  $y \geq 75$ , o deslocamento é à direita.

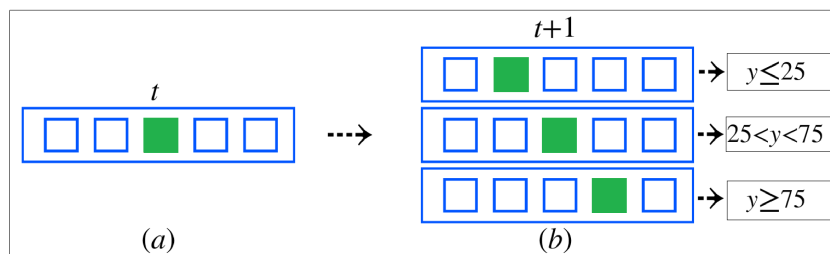


Figura 2.9: (a) Célula vivas no instante  $t$ . (b) Aplicação da regra de transição probabilística no instante  $t + 1$  (Castro and Lima, 2013a).

Caso a variável  $y$  que representa a probabilidade  $P(x)$  for  $25 < y < 75$ , tem-se que o valor da célula central permanece no estado ■ e não há deslocamento (Castro and Lima, 2013a). Várias aplicações encontram-se presentes neste contexto, dentre elas: simulação

de mecânica dos fluidos (Dilão, 1993), de ecossistemas Farina and Dennunzio (2008), de evacuação de pedestres (Lima and Lima, 2014).

Para a evolução espaço-temporal do um AC é usual que todas as células sejam atualizadas de acordo com a mesma função de transição (Castro and Lima, 2013a). Quando isso ocorre, dizemos que o autômato celular é homogêneo. Por outro lado, quando atualizamos as células de um reticulado com dois ou mais tipos de regras dizemos que o AC é heterogêneo ou híbrido. Por exemplo, em um reticulado de tamanho  $N$ , podemos definir  $N$  regras de transição diferentes, uma para cada célula do reticulado. Na arquitetura de navegação de robôs proposto aqui, mais de uma regra podem ser aplicadas ao longo do reticulado para a dinâmica dos robôs.

## 2.2 Modelagem de sistemas complexos via autômatos celulares

Dentre os estudos com uso de ACs, destaca-se a modelagem de fenômenos naturais, ambientais, físicos e biológicos que seriam muito difíceis de serem modelados pelas equações diferenciais. Diferentemente das equações, a simulação computacional garante que um resultado possa ser obtido não apenas por meio dados numéricos, mas também, na observação de padrões gráficos analisados visualmente, em qualquer dimensão em que o problema esteja sendo abordado. Esse método é largamente empregado em ACs cuja evolução é calculada numericamente na grade de células (Wolfram, 2002b).

As regras de transição utilizadas em ACs são simples o que possibilita sua implementação computacional com uso de variáveis e parâmetros de diferentes tipos e naturezas diversas, segundo o que o modelo a ser implementado necessita. O *Game of Life* proposto por Conway (1970) é apresentado na Seção 2.1.1 pode ser visto como um exemplo de modelagem de sistemas biológicos. Dois exemplos em modelagem baseada em ACs são apresentados nas próximas seções.

### 2.2.1 Aplicação em sistemas naturais: Formação de padrões naturais e fractais

Muitos sistemas naturais, incluindo a formação dos flocos de neve, parecem exibir padrões de um objeto fractal. Segundo Wolfram (2002b), é provável que em muitos casos esse padrão fractal seja formado por um processo análogo à evolução de um autômato celular. Estudos mostram evolução de ACs com uma formação semelhante ao padrão de pigmentação encontrado nas conchas de determinados moluscos (Wolfram, 1986), (Markus and Kusch, 1995) e (Wolfram, 2002b).

Os padrões triangulares são formados com base em uma regra de transição que impõe



que o valor de cada célula seja a soma de seus dois vizinhos mais próximos, dados a cada passo de tempo em função do estado anterior. Com a evolução ocorrendo progressivamente o padrão geométrico é gerado contendo “auto-similaridade”. A Figura 2.10, aborda o modelo físico-natural comparado com o modelo computacional.

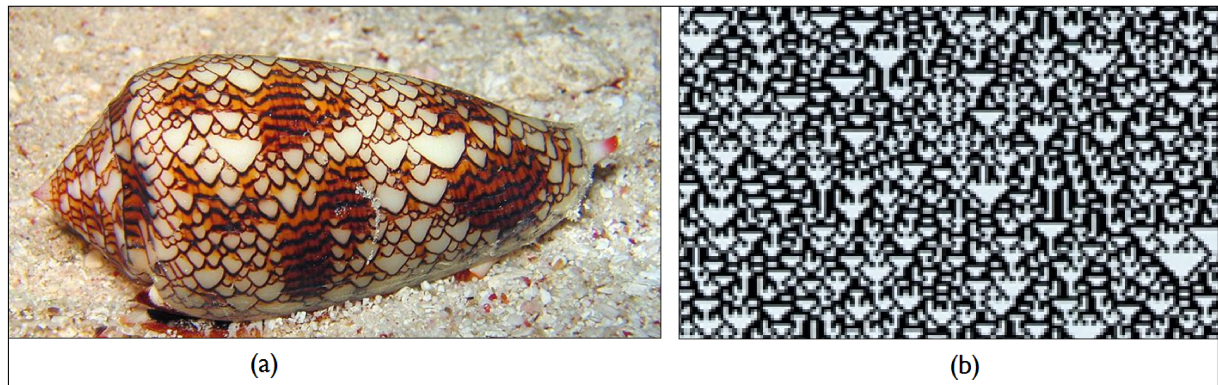


Figura 2.10: Padrão triangular: (a) exemplo de concha marinha *Conus textile* com padrão triangular; (b) a representação computacional do padrão natural da concha (Dilão, 1993).

### 2.2.2 Aplicação em sistema físico-ambiental: propagação de incêndios florestais

A variação climática e vários outros fatores aumentam a probabilidade da propagação de incêndios florestais. Além do clima, a temperatura, a umidade relativa a precipitação, o vento e o direcionamento do fogo em incêndios também contribuem para a progressão do espalhamento das chamas (Lima and Lima, 2014). A atuação do vento em incêndios é decisivo, sendo este responsável por transportar as massas de umidade, tornando, o ambiente mais suscetível a queimadas. Se o incêndio já estiver em processo de queima, este fenômeno natural contribui para o alastrar a queimada, pois fornece o comburente oxigênio e influência na direção de propagação do incêndio.

Assim, uma implementação de AC no modelo físico-ambiental torna-se interessante para tentar diminuir os impactos causados por um processo de queimada. Além disso, os ACs trazem a simplicidade de implementação computacional. Alguns estudos enfatizam como o uso de ACs podem ser usados para a compreensão de como aspectos naturais interferem em um incêndio florestal. Dentre os trabalhos estudados, a propagação de incêndios florestais foi estudada por (Dilão, 1993) (Clarke et al., 1994), (Hernandez et al., 2007) e (Lima and Lima, 2014) e foram idealizados a partir do trabalho matemático de (Rothermel, 1972; Rothermel et al., 1980).

Em (Dilão, 1993) o autor considerou que o reticulado dos ACs era bidimensional (2D), com células que possuíam formato quadrangular e cada célula do reticulado representava uma árvore. Nesse modelo, Dilão (1993) a homogeneidade das árvores é considerada,

assim como a ação que o vento acarreta nas queimadas. O estado associado ao verde corresponde a uma árvore viva, ao vermelho uma árvore sendo queimada e ao preto uma árvore já queimada. Só existe a probabilidade de uma árvore começar a queimar, ou seja, passar para o estado vermelho, se ela estiver no estado verde (viva) e se pelo menos uma das células vizinhas estiverem no estado vermelho (queimando) com uma probabilidade  $P(i)^t$ , essa é a regra de atualização dos estados. As células mantêm-se neste estado vermelho por um determinado intervalo  $t$  de tempo, ao contrário das que apresentam o estado preto, mortas, que vão permanecer neste estado indefinidamente.

A Figura 2.11, aborda o efeito visual gerado pela execução do modelo (Lima and Lima, 2014) nos passos de tempo 6, 20 e 38 respectivamente. As árvores são representadas por 3 estados: verde (árvores intactas), preto (árvores mortas) e demais cores (árvores queimando).

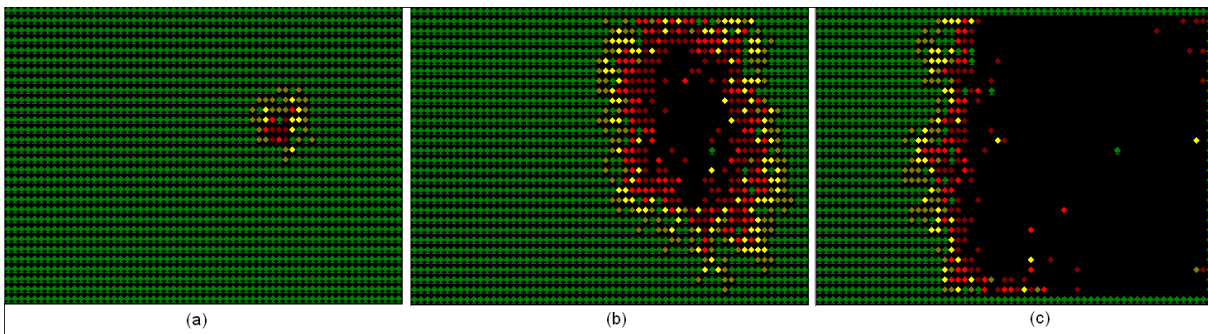


Figura 2.11: Autômatos Celulares aplicados em simulação de incêndios florestais: resultado gráfico da implementação do modelo proposto por (Lima and Lima, 2014).

A simulação de incêndios considera fatores como: tipo de vegetação local, tempo de queima em função do tipo de bioma predominante e direção do vento; altura do relevo, força da combustão. Tais parâmetros tornam o AC o mais próximo possível de um incêndio florestal real (Lima and Lima, 2014), podendo servir como base para a criação de um modelo que atenda às necessidades dos grandes centros de controle. De tal forma que esses modelos matemáticos e computacionais podem ser parte de um sistema real que combinam um sistema de informação geográfica (GIS) e imagens de satélite com um modelo de propagação de incêndio florestal.

Um modelo de propagação de incêndios em ambientes fechados será apresentado em uma das tarefas investigadas no presente trabalho a partir da utilização de regras temporais e probabilísticas. Iremos adotar alguns destes parâmetros para a modelagem do incêndio que compõe o cenário que servirá como base para a implementação da tarefa de busca e resgate pelo grupo de robôs no Capítulo 5.

## 2.3 Considerações em relação ao método proposto

Nesse capítulo foram apresentadas as principais definições sobre autômatos celulares e suas aplicações. Em relação às definições apresentadas, o método de navegação para sistemas multi-robôs discutido no Capítulo 5 apresenta as seguintes características: (i) a dimensão do reticulado empregado que é bidimensional (2D) com células quadradas, (ii) o comportamento dinâmico das regras nos modelos iniciais é determinístico para a escolha a priori do movimento e não-determinístico para a resolução de conflitos no modelo principal, (iii) a modelagem da navegação de robôs apresenta cinco estados principais para cada célula - ocupado por robô, ocupada por objeto, ninho, livre, obstáculo, (iv) em modelos posteriores, a escolha do movimento dos robôs torna-se estocástica a partir da sua vizinhança, (v) o AC utilizado é heterogêneo ou híbrido empregando várias regras de atualização, (vi) o reticulado possui condição de contorno não-periódica e (vii) as células devem ser atualizadas de forma síncrona, (viii) a modelagem do fogo utilizada na tarefa de busca e resgate apresenta um AC com regras estocásticas e temporais, (ix) a modelagem do espalhamento fogo na tarefa robótica apresenta três estados principais para cada célula - livre, em chamas ou em cinzas.



---

# Dinâmica coletiva

---

Pierre Lévy ajudou a estabelecer o conceito da inteligência coletiva através da realização de alguns debates relacionados às conexões sociais que se tornam viáveis pela utilização da inteligência compartilhada que surge, por sua vez, da colaboração de muitos indivíduos em suas diversidades (Lévy, 1997). É uma inteligência distribuída por toda parte, também chamado de inteligência simbiótica (Crosby et al., 2006). O conceito é tanto usado na área da sociologia, como no mundo dos negócios, na comunicação de massa, assim como na ciência da computação. Além disso, é necessário compreender que cada agente possui conhecimento em suas particularidades, ou seja, ter conhecimento não se trata de ser inteligente de fato e sim possuir experiências vividas ao longo da vida e que podem ser compartilhadas. Assim, através de todo conhecimento interligado, todos os agentes de um determinado sistema passam a estar interligados também. Neste contexto, a inteligência coletiva pode ser entendida no sentido em que todos, em suas individualidades, possuem conhecimentos e uma inteligência compartilhada. Ou seja, a inteligência coletiva é resultante de uma mobilização efetiva das competências capaz de explorar ao máximo todo o aprendizado existente nas populações (Woolley et al., 2010).

Estes comportamentos coletivos são amplamente estudados por diversos pesquisadores, por exemplo, quando se deseja observar o comportamento de uma multidão em alguma situação de risco (Castro and Lima, 2013a), quando envolve busca e resgate ou evacuação emergencial Yang et al. (2005), ou até mesmo quando se deseja levantar padrões e analisar comportamentos durante a dinâmica em uma situação de risco (Zheng et al., 2010). Nesse capítulo, a ênfase será nos estudos de modelos para simular a dinâmica populacional. Entretanto, primeiramente, apresentaremos conceitos de duas áreas que estudam técnicas

relacionadas a esses modelos: inteligência de enxames e sistemas bio-inspirados.

### 3.1 Inteligência de enxames e dinâmica coletiva

O campo de pesquisa voltado para o comportamento coletivo em sistemas auto-organizados e descentralizados é chamado inteligência de enxames (Bosque et al., 2009). Neste novo campo, a ideia de um controlador sofisticado que governa o comportamento global do sistema tem sido substituído pelo princípio da inteligência de enxames, que é baseado em várias entidades simples que cooperam entre si de maneira local com o objetivo de exibir um comportamento universal desejado. A inspiração para o projeto destes sistemas vem do comportamento coletivo de insetos sociais como formigas, abelhas, cupins, vespas, bem como o comportamento de outras sociedades de animais, como bando de pássaros ou cardume de peixes (Dorigo et al., 2008) e também foram observados em dinâmica de multidões (Castro and Lima, 2013a). Mesmo que cada membro destas sociedades seja um indivíduo simples, eles são capazes de realizar tarefas complexas em cooperação.

Por exemplo, formigas, cupins e vespas são capazes de construir ninhos sofisticados em cooperação (Gordon, 2014), sem que nenhum dos integrantes tenha um plano global de como proceder. Outro exemplo é o comportamento de exploração que formigas ou abelhas exibem quando procuram por comida. Enquanto as formigas utilizam uma estratégia de comunicação indireta através de trilhas de feromônio, a fim de encontrar caminhos mais curtos entre os seus ninhos e as fontes de alimento, as colônias de abelhas são mais eficientes na exploração de ricas fontes de comida, realizando sua comunicação por meio de uma dança específica (Blum and Merkle, 2008).

Outro fenômeno observado que apresenta este mesmo comportamento em direção ao seu objetivo é o fenômeno complexo de dinâmica de multidões (Helbing et al., 2000). Neste tipo de problema, cada indivíduo realiza movimentos individuais e a partir dessa dinâmica um comportamento global coletivo é exibido. A modelagem desse comportamento passou a ser estudado através de autômatos celulares onde cada indivíduo representa uma célula do AC e o comportamento global emerge a partir dessas interações ao longo do tempo (Varas et al., 2007).

Essas ideias recaem nos conceitos de autômatos celulares que também são definidos como unidades simples que coletivamente apresentam um comportamento global, com poder de realizarem tarefas específicas, tais como, sincronização e paridade (Oliveira and Vaiano, 2005) e (Carvalho et al., 2008). Os ACs são capazes de modelar sistemas complexos, a partir de interações locais entre seus indivíduos, essas interações guiam e controlam todo o sistema de busca e de dinâmica enxames para um determinado objetivo, como por exemplo, em dinâmica populacional.

Diversos pesquisadores têm aplicado esses princípios para novas abordagens, por exem-

plo, na otimização e no controle de robôs (Lima et al., 2016a). Entre as propriedades que caracterizam os sistemas resultantes estão a robustez a falhas e a flexibilidade para a execução de diferentes tarefas.

Na otimização, há aplicações de sucesso como a otimização por colônia de formigas (ACO) e otimização por enxame de partículas (PSO) (Blum and Merkle, 2008). A ACO é uma técnica utilizada para procurar caminhos em grafos, baseando-se no comportamento observado em formigas ao saírem de seus ninhos para encontrarem alimento. Já a PSO é aplicada em diversos tipos de problemas, como o problema do caixeiro viajante, o treinamento de redes neurais, o controle de tensão e potência reativa (Bosque et al., 2009).

## 3.2 Algoritmos de busca

Um aspecto importante a ser considerado em busca são as propriedades dos ambientes onde a busca está sendo realizada (Russell and Norvig, 2005), ou seja, o espaço de busca. O ambiente pode ser completamente observável, isso significa que os sensores do agente dão acesso ao estado completo do ambiente em cada instante e todos os aspectos relevantes do ambiente são acessíveis. Ambiente parcialmente observável, significa que o agente não tem percepção do todo e ocorre devido a ruídos a sensores imprecisos ou porque partes do estado estão simplesmente ausentes dos dados do sensor. Se o ambiente é determinístico, significa que o próximo estado do ambiente é completamente determinado pelo estado atual e pela ação executada pelo agente. Já se o ambiente é determinístico, exceto pelas ações de outros agentes, dizemos que o ambiente é estratégico. E o ambiente é estocástico (não determinístico) quando existem mudanças no ambiente, seja pelo posicionamento diferente de outros agentes no ambiente, seja pelas variáveis que compõem o ambiente, tais como, feromônio ou obstáculos dinâmicos. Em relação à percepção do ambiente podemos dizer que o ambiente é episódico quando a experiência do agente pode ser dividida em episódios (percepção e execução de uma única ação) e a escolha da ação em cada episódio só depende do próprio episódio (Russell and Norvig, 2005). O ambiente é dito sequencial se a decisão atual pode afetar todas as decisões futuras. Em relação às mudanças do ambiente, podemos classificá-lo como estático se ambiente não muda enquanto o agente pensa. O ambiente é semidinâmico se ele não muda com a passagem do tempo, mas o nível de desempenho do agente se altera ou o ambiente é dinâmico se o ambiente se altera enquanto o agente está deliberando. O ambiente pode ser classificado como discreto se existe um número limitado e claramente definido de percepções e ações ou contínuo, quando os estados que o agente pode deliberar mudam continuamente. O ambiente ser monoagente quando um único agente atua sobre o ambiente ou multiagente, quando mais de um agente atua sobre o ambiente. Esses agentes podem trabalhar de maneira competitiva - lutando para ganharem o objetivo, cooperativa - quando eles se ajudam para

realizarem a tarefa, ou híbrida - quando se ajudam para realizar a tarefa, mas competem por um nicho de espaço no ambiente em determinadas situações (Russell and Norvig, 2005).

Na resolução de problemas os agentes decidem o que fazer encontrando sequências de ações que os levam para estados desejáveis Russell and Norvig (2005). Existem diversos algoritmos de busca para a resolução de problemas, incluindo-se, algoritmos de busca: cega, heurística, competitiva, com informações parciais, e de exploração. Um algoritmo de busca toma um problema como entrada e retorna a solução para o problema, geralmente após resolver um número possível de soluções. Normalmente existem quatro propriedades principais na elaboração de algoritmos de busca: (a) fazer o algoritmo de busca ter um tempo de execução sempre aceitável, (b) complexidade de espaço aceitável em relação à quantidade de memória gasta, (c) ser a solução ótima ou provavelmente boa para o problema em todos os casos e (d) o algoritmo deve ser completo, ou seja, ele deve oferecer a solução quando a mesma existir Russell and Norvig (2005). Se um algoritmo de busca não cumpre uma dessas propriedades, por exemplo, (i) ele pode ser ou um algoritmo que encontra boas soluções a maioria das vezes, mas não tem garantias de que sempre a encontrará ou (ii) pode ser um algoritmo que tem processamento rápido, mas não tem provas de que será rápido para todas as situações; então um algoritmo de aproximação (heurístico) é utilizado. Ou seja, é necessário utilizar informação e intuição a respeito da instância do problema e da sua estrutura para resolvê-lo de forma rápida Russell and Norvig (2005). A grande desvantagem das heurísticas reside na dificuldade de escapar de ótimos locais, o que deu origem à outra metodologia, chamada de metaheurística. As metaheurísticas, que são heurísticas genéricas mais sofisticadas, onde uma heurística mais simples é gerenciada por um procedimento que visa explorar inteligentemente a instância do problema e o seu espaço de soluções (Russell and Norvig, 2005).

As metaheurísticas são baseadas na solução do problema observando-se comportamentos da natureza a partir de uma solução única do problema que é melhorada a partir da vizinhança. Dentre esses algoritmos podemos citar, *Simulated Annealing* (SA), Busca Tabu (*Tabu Search*), *Greedy Randomized Adaptive Search Procedure* (GRASP), *Variable Neighborhood Search* (VNS) (Russell and Norvig, 2005). Outros algoritmos baseiam-se em princípios biológicos de populações de seres vivos de sucesso, gerando não uma solução simultânea, mas uma população de soluções. Nesse caso, as metaheurísticas mais conhecidas correspondem às técnicas de otimização por colônia de formigas (ACO), Otimização por Enxame de Partículas (PSO), Algoritmo *Shuffled frog-leaping*, Coleta de Alimentos por Bactérias, Colônia de Abelhas e os Algoritmos Genéticos (AGs). Outros algoritmos baseiam-se na observação de fenômenos da natureza e a solução é guiada a partir da melhoria da vizinhança (Russell and Norvig, 2005). Neste trabalho os algoritmos de busca que são de fundamental importância para a concepção dos modelos de navegação de robôs, que serão descritos no Capítulo 5, foram a Busca Tabu (BT) (Glover, 1989) e a



Otimização por Colônia de Formigas (ACO) (Dorigo et al., 1996).

### 3.2.1 Algoritmo de Busca Tabu

A Busca Tabu (*Tabu Search*) é uma meta-heurística e um procedimento adaptativo auxiliar que guia um algoritmo de busca local na exploração contínua dentro de um espaço de busca (Glover, 1989). A partir de uma solução inicial, essa abordagem tenta avançar para uma outra solução (melhor que a anterior) na sua vizinhança até que se satisfaça um determinado critério de parada. O algoritmo de Busca Tabu foi proposto por (Glover, 1989), na década de 1980 e é uma técnica muito semelhante à do arrefecimento simulado. A Busca Tabu (BT) é construída de forma a evitar o retorno a um ótimo local previamente visitado. Esta característica faz com que o método seja capaz de superar a otimalidade local e atingir um resultado ótimo ou próximo ao ótimo global (Glover, 1990).

O algoritmo da Busca Tabu usa uma vizinhança de busca local para mover iterativamente para uma solução potencial na célula  $x$  para uma solução  $x'$  na vizinhança dessa célula central  $x$  até que um critério de parada seja satisfeito (Glover, 1989, 1990). Procedimentos de busca local muitas vezes ficam presos em áreas pobres de valores baixos. Para evitar isso, e explorar regiões que ainda não foram exploradas, a Busca Tabu explora a vizinhança de cada solução à medida que o algoritmo evolui. As soluções admitidas para a nova vizinhança no próximo instante de tempo  $t$  é denotada por  $\eta_x^t$  e são determinadas através do uso de estruturas de memória. Usando essas estruturas de memória a busca prossegue iterativamente por  $T$  passos de tempo movendo-se a partir da solução atual para uma solução melhorada  $x'$  em  $\eta_x^t$  (Glover, 1989, 1990).

Essa estrutura de memória forma uma Lista Tabu ( $Q$ ), ou seja, um conjunto de soluções banidas que são usadas para definir a próxima vizinhança  $\eta_x^t$  a cada instante de tempo  $t$  que serão exploradas na buscas. Em definição, a Lista Tabu é um conjunto de soluções de curto prazo que tem sido visitadas num passado mais recente (menores que  $T$  iterações passadas, onde  $T$  é o número de soluções prévias que foram guardadas). Geralmente, a Lista Tabu consiste em soluções que mudaram com o processo de uma solução para outra (Glover, 1989, 1990).

Assim, busca Tabu gera várias soluções e move-se para uma solução mais satisfatória atendendo, para tanto, a algum critério predeterminado de restrições, com o intuito de prevenir ciclos e encorajar maior movimentação através do espaço. Pode ser mantida lista de tabus das soluções parciais ou completas. É proibido mover-se para uma solução que contenha elementos presentes nesta lista, que por sua vez é atualizada assim que uma determinada solução transcorre todo o espaço (Glover, 1989, 1990). Essa técnica propicia maior espalhamento da busca no espaço de soluções, criando uma cobertura mais efetiva desse espaço. Neste trabalho, essa característica da Busca Tabu foi acoplada ao modelo de navegação multi-robôs de forma a prover uma melhor cobertura do ambiente, em respeito

à busca realizada pelos agentes.

### 3.2.2 Sistemas bio-inspirados

Os sistemas bio-inspirados utilizam-se da natureza como fonte de inspiração ou metáfora para desenvolvimento de novas técnicas computacionais utilizadas para resolver problemas complexos. Métodos inspirados na teoria da evolução de Darwin foram propostos pela primeira vez em 1958 (Holland et al., 1982) e são capazes de gerar resultados competitivos em relação àqueles encontrados por humanos. A partir disso, diversas áreas do conhecimento passaram a utilizar essa abordagem para resolver problemas. Dentre esses algoritmos mais amplamente discutidos destacam-se os Algoritmos Genéticos (AGs) (Holland et al., 1982), Otimização por Colônia de Formigas (ACOs) (Dorigo et al., 1996), Otimização por Enxames de Partículas (PSO) (Kennedy, 2011).

#### Colônias de Formigas

O algoritmo da otimização por colônia de formigas (ACO, do inglês Ant Colony Optimization), introduzido por Marco Dorigo (Dorigo et al., 1996) é uma heurística baseada em probabilidade, criada para solução de problemas computacionais que envolvem busca de caminhos em grafos. Este algoritmo foi inspirado na observação do comportamento das formigas ao saírem de sua colônia para encontrar comida (Castro Silva, 2005).

As formigas caminham arbitrariamente até que, encontrada comida, elas retornam à colônia deixando um rastro de feromônio (taxa de difusão do feromônio  $\delta$ ). Se outras formigas encontram um desses rastros, elas tendem a não seguir mais caminhos aleatórios. Em vez disso, seguem a trilha encontrada, retornando por ela e inclusive reforçando o feromônio depositado se acharem mais unidades de alimento (Lima et al., 2017a).

Entretanto, com o passar do tempo  $t$  as trilhas de feromônio começam a evaporar (taxa de declínio  $\beta$ ), reduzindo, assim, sua força atrativa. O declínio do feromônio faz com que o algoritmo de busca tenha maior convergência para uma solução local ótima. Ou seja, se a evaporação não ocorresse, todas as trilhas escolhidas pelas primeiras formigas seriam muito atrativas para as outras e, neste caso, a exploração do espaço da solução seria restrita (Dorigo et al., 1996). Quanto mais formigas passarem por um caminho predeterminado, mais tempo será necessário para o feromônio da trilha evaporar. Analogamente, elas caminham mais rapidamente sobre uma trilha mais curta, o que determina um aumento da intensidade de feromônio inserido no ambiente antes da sua evaporação.

Para aumentar a exploração do ambiente, o uso do feromônio foi estudado por (Calvo et al., 2011b) e ele abordou a utilização do mesmo não como um mecanismo de atração entre os agentes, mas de repulsão entre os mesmos. Ele denominou esse sistema de Inverted Ant System (IAS), que é baseado na ideia de repulsão entre os agentes através da força do feromônio invertido (Calvo et al., 2012). Ele aplicou esta técnica na tarefa de robótica

conhecida como vigília e o objetivo era que o menor número de agentes robóticos fossem espalhados pelo maior número de cômodos possíveis, de tal forma que todos os cômodos fossem visitados pelo menor número de robôs e também respeitando-se um tempo  $T$  para a realização da tarefa. Diferentes técnicas de atualização de feromônio foram investigadas no trabalho (Calvo et al., 2011a). Essa estratégia é muito eficiente, especialmente quando o objetivo do enxame é realizar uma cobertura do espaço e assim desviar de pontos previamente já visitados (Lima et al., 2016a).

### Enxames de Partículas

O método do enxame de partículas (em inglês: *Particle Swarm Optimization* ou PSO) é um ramo da inteligência artificial também classificado por alguns autores como um ramo da computação evolucionária, que otimiza um problema iterativamente ao tentar melhorar a solução candidata com respeito a uma dada medida de qualidade. O método de PSO, criado por James Kennedy e Russell Eberhart Eberhart and Kennedy (1995a), encontra-se, atualmente, entre as meta-heurísticas de algoritmos de otimização, baseadas em padrões da natureza (como a representação do movimento de cada indivíduo dentro de um bando de pássaros ou de um cardume de peixes), mais populares nesta área e surge como sendo o algoritmo mais promissor para a resolução dos mais diversificados problemas de otimização, quer na área das ciências, quer das engenharias. Desde a sua criação, já foram desenvolvidas muitas variantes para a resolução de problemas práticos relativos à otimização, tal como em robótica, (Couceiro et al., 2014b).

O PSO é inspirado pelo comportamento social e cooperativo exibido por várias espécies por forma a realizar as suas necessidades no espaço de busca. Adicionalmente, o PSO é uma meta-heurística, pois realiza poucas ou nenhuma premissa sobre o problema que está a ser otimizado e pode procurar soluções candidatas em espaços de grandes dimensões Eberhart and Kennedy (1995a). No entanto, o algoritmo não garante que uma solução ideal seja encontrada. Em termos gerais, o algoritmo guia-se por experiência pessoal (**Pbest**), experiência geral (**Gbest**) e o movimento das partículas atual para decidir as posições seguintes no espaço de busca. O PSO resolve um problema criando uma população de soluções candidatas, também conhecidas como partículas e movendo estas partículas em torno do espaço de busca, de acordo com fórmulas matemáticas simples sobre a posição e velocidade da partícula Eberhart and Kennedy (1995a). O movimento de cada partícula é influenciado pela sua posição do local mais conhecido. Mas, também é guiado em direção às posições mais conhecidas do espaço de busca, que são atualizadas como posições melhores quando encontradas por outras partículas. Isto é o esperado, quando o intuito é mover o enxame em direção da melhor solução Eberhart and Kennedy (1995a).

### 3.3 Modelos para simulação de dinâmica coletiva

A todo tempo é possível observar processos dinâmicos na natureza, seja em fenômenos físicos ou sociais, tais como a movimentação dos agentes em uma colônia de formigas ou de cardumes de peixes. Na sociedade humana podemos observar essa dinâmica a todo instante. Em várias situações os seres humanos aglomeram-se para a realização de uma determinada tarefa, seja para estudo, trabalho ou lazer. Quando a tarefa é cumprida a busca pela saída é sempre conflituosa, principalmente se a aglomeração de pessoas é grande. Além disso, os conflitos entre os pedestres e obstáculos são observados o tempo todo até a saída (meta). Assim, muitos pesquisadores investigam a dinâmica de pedestres, cujo objetivo é encontrar modelos que imitem o comportamento natural exibido pela multidão (Helbing et al., 2000) durante a evacuação.

#### 3.3.1 Principais abordagens

Existem diversos modelos que foram desenvolvidos para simular o processo de dinâmica coletiva de pessoas em **situações** normais ou de emergência (Helbing et al., 2000), (Wu et al., 2015), (Feliciani and Nishinari, 2016) em busca de uma saída. Um grande número de modelos para simulação com pedestres foi desenvolvido ao longo dos anos em uma variedade de disciplinas (computação gráfica, robótica, transportes, etc). Esses modelos podem ser classificados em dois subconjuntos, que se referem à maneira como os agentes são gerenciados (**escala**): modelos macroscópios e microscópios. Modelos macroscópios focam nos sistemas, ou seja, quando uma multidão de pedestres é considerada como um todo. Por outro lado, os modelos microscópios estudam o comportamento individual de decisão dos pedestres e suas interações com outros pedestres na multidão. Modelos macroscópios incluem modelos de regressão, modelos de escolha de rota, modelos de filas e modelos de gases cinéticos (Zheng et al., 2009). Modelos microscópios incluem forças sociais (sistemas de partículas) (Reynolds, 1987), (Liu et al., 2009), baseados em regras (Helbing et al., 2000), (Guo et al., 2011) e em modelos de autômatos celulares (Schadschneider, 2001a), (Kirchner and Schadschneider, 2002) e (Varas et al., 2007). Modelos de pesquisa em computação gráfica, também podem ser considerados modelos microscópios se exploram formas virtuais para a criação de seres humanos que se comportam de maneira autônoma.

As estratégias mais comuns para a simulação de dinâmica coletiva em modelos microscópios são os modelos baseados em forças sociais (Reynolds, 1987), (Reynolds, 1999) e modelos baseados em autômatos celulares (Wolfram, 1983). A principal diferença entre eles é a forma com que eles tratam o ambiente que pode ser contínuo (forças sociais) ou discreto (autômatos celulares). Outra abordagem bastante estudada são os modelos baseados em regras que utilizam o ambiente de computação gráfica para simular seres humanos virtuais (Funge et al., 1999), (Brogan and Hodgins, 2002) e (Huang and Liu,

2012).

Uma das abordagens macroscópicas mais usuais são os modelos de regressão, que usam relações estatísticas estabelecidas entre as variáveis de fluxo para prever as operações de fluxo de pedestres em situações específicas. As características desse fluxo dependem da infraestrutura (escadas, corredores, etc) (Milazzo et al., 1998). Por outro lado, de escolha de rota descrevem a escolha do pedestre baseado nos conceitos de utilidade (Hoogendoorn and Bovy, 2003). Isso significa que o pedestre escolhe seu destino para maximizar algum parâmetro como exemplo (conforto, tempo de fuga, etc). Outra abordagem macroscópica comum utiliza modelos de fila que empregam a cadeia Markoviana para descrever como os pedestres movem-se de um nó da rede até outro. Nós são geralmente salas, e seus links são portas ou portões. Modelos de cadeia markoviana são definidos como sendo um conjunto de estados com suas respectivas transições de probabilidade (Lovas, 1994). A cada passo de exploração, um estado sucessor é selecionado por outro de acordo com a distribuição de transição, ou identificando o sucessor mais provável. Finalmente, modelos de gases cinéticos, usam a analogia de dinâmicas de fluidos ou gases para descrever, utilizando equações diferenciais, como a densidade e velocidade mudam ao longo do tempo (Henderson, 1971).

Outra característica importante que diferencia os diferentes modelos macroscópicos e microscópicos é em relação à manipulação de **indivíduos ou grupos**. Em alguns casos os pedestres são idealmente considerados como indivíduos homogêneos. Em outros, os pedestres são vistos como indivíduos heterogêneos (grupos) pela diferença de características (por exemplo, sexo, idade, psicologia). Também deve ser considerado o **tempo e espaço (SAT)** dos modelos que estão sendo investigados, neste caso, podemos ter abordagens discretas em tempo e espaço e em outros casos, são contínuas. Além disso, **fenômenos típicos** podem ser avaliados, ou seja, diferentes comportamentos podem ser reproduzidos nas simulações de fluxo de pedestres (Zheng et al., 2009).

No trabalho de (Zheng et al., 2009) os autores apresentaram sete abordagens metodológicas (autômato celular, lattice gas, forças sociais, dinâmica dos fluidos, baseada em agentes, teoria dos jogos e experimentos com animais), discutindo quais as vantagens e as desvantagens (Carneiro, 2012) de cada uma delas no processo de dinâmica coletiva que são descritas a seguir:

- Autômatos celulares (AC) têm sido aplicados para descrever a dinâmica de pedestres durante as evacuações. Os principais trabalhos da área são: (Alizadeh, 2011; Burstedde et al., 2001; Carneiro, 2012; Daoliang et al., 2006; Fang et al., 2010; Fu et al., 2012; Georgoudas et al., 2006; Guo et al., 2011; Huang and Liu, 2012; Ji et al., 2013; Jian et al., 2005; Kirchner et al., 2003a,b; Kirchner and Schadschneider, 2002; Liu et al., 2009; Nishinari et al., 2006; Pereira et al., 2011; Perez et al., 2002; Schadschneider, 2001a; Schadschneider et al., 2011; Schultz et al., 2007; Varas et al., 2007; Wei-Guo et al., 2006; Weifeng et al., 2003; Weng et al., 2006; Yamamoto et al.,

2007; Yang et al., 2003, 2002, 2005; Yu and Song, 2007; Yuan and Tan, 2007; Zarita and Aik, 2012). Mais recentemente, (Castro and Lima, 2013a; Fu et al., 2015; Guo et al., 2016; Guo, 2014; Hou et al., 2014).

- Lattice gas (LG) é caracterizado como um caso especial de autômatos celulares. Nesse tipo de modelo, cada pedestre é considerado como uma partícula ativa na grade. Essa abordagem é utilizada com frequência no estudo das características de uma multidão de pedestres, por meio de probabilidade e estatística. Os trabalhos mais citados na literatura são: (Tajima and Nagatani, 2001), (Tajima et al., 2001), (Tajima and Nagatani, 2002), (Itoh and Nagatani, 2002), (Helbing et al., 2003), (Takimoto and Nagatani, 2003), (Nagai et al., 2004), (Isobe et al., 2004), (Nagai et al., 2005), (Nagai et al., 2006), (Song et al., 2006).
- Os modelos de forças sociais (SF) têm sido utilizados na simulação de situações de pânico, entre outras. Nesses modelos o movimento dos pedestres é determinado pela aplicação de forças que resultam nos seguintes efeitos: os pedestres desejam chegar a um determinado destino; eles mantêm certa distância dos outros pedestres e das fronteiras dos obstáculos no ambiente (paredes, por exemplo); e ainda, podem ser atraídos por outros pedestres (amigos, por exemplo) ou por objetos (vitrines, por exemplo) (Carneiro, 2012). Os principais trabalhos correlacionados são: (Helbing et al., 2000; Helbing and Molnar, 1995; Lin et al., 2006; Parisi and Dorso, 2005; Seyfried et al., 2006; Zheng et al., 2002).
- Nos modelos de dinâmica dos fluidos (FD) as multidões de pedestres são descritas com propriedades de fluidos. Esses modelos descrevem como densidade e velocidade mudam com o tempo através do uso de equações diferenciais parciais. Como exemplo, modelos de dinâmica dos fluidos podem ser utilizados com a finalidade de tentar compreender a mecânica do movimento coletivo. Os principais trabalhos encontrados na literatura são: (Colombo and Rosini, 2005; Hughes, 2002, 2003).
- Modelos baseados em agentes (AB) são modelos computacionais que possuem a característica de construir as estruturas sociais de modo “bottom up”, simulando indivíduos como agentes virtuais, e criando organizações emergentes das regras que governam as interações entre os agentes. Eles podem ser utilizados para simular o comportamento de cada agente pertencente ao sistema. Principais trabalhos correlatos: (Bandini et al., 2006; Braun et al., 2005; Henein and White, 2006; Pan et al., 2007; Pelechano et al., 2007; Toyama et al., 2006; ZARBOUTIS and MARMARAS, 2004).
- A abordagem de teoria dos jogos (GT) pode ser adotada para modelar situações em que o processo de decisão interativo dos evacuados é racional (Carneiro, 2012). Modelos de teoria dos jogos podem ser utilizados, por exemplo, para simular o comportamento competitivo em uma saída de emergência. Um dos trabalhos que utilizou esta abordagem está apresentado em (Lo et al., 2006).

- Abordagens baseadas em experimentos com animais (AE) surgem como uma alternativa para a simulação do processo de dinâmica coletiva em situações de pânico genuíno, visto que há dificuldades em realizar esse tipo de experimento com seres humanos, por envolver preocupações éticas e até mesmo legais. Os principais trabalhos investigados utilizaram para experimentos formigas (Altshuler et al., 2005) e camundongos (Saloma et al., 2003).

As abordagens acima visam simular realisticamente a dinâmica coletiva de pedestres e, por isso, são utilizadas no desenvolvimento de diversos modelos (Carneiro, 2012). Uma revisão das principais abordagens de dinâmica coletiva em simulação de evacuação pode ser vista em (Zheng et al., 2009). Apresentamos na Tabela 3.1 um resumo das principais características dessas abordagens extraído desse trabalho. Os modelos baseados em lattice gas e autômatos celulares destacam-se na modelagem de dinâmica coletiva. Esses modelos são discretos nas variáveis de espaço, tempo e estado, o que os torna modelos idealmente adequados para simulações de computador em grande escala. Além disso, modelos de autômatos celulares têm grande poder expressivo para representar comportamentos complexos a partir de um conjunto de regras simples (Wolfram, 1983), bem como comportamentos individuais e coletivos observados na dinâmica de pedestres (Varas et al., 2007). Neste trabalho, será utilizada uma abordagem baseada em autômatos celulares para descrever a dinâmica de multi-agentes no contexto da robótica.

O comportamento de um indivíduo pode afetar a escolha de rota, o tempo e a eficiência do processo de busca em modelos coletivos. Caso a população esteja procurando a melhor trajetória de fuga (evacuação) o processo torna-se ainda mais complexo à medida em que as relações sociais são consideradas (Yang et al., 2002).

Simular os fatores humanos em situação de normalidade já não é uma tarefa simples, e simular esses fatores em situações de emergência torna essa tarefa ainda mais árdua. Isso porque muitos fenômenos e leis gerados pelas interações entre pessoas, construção e ambiente, são obtidos através de modelagem, devido à escassez de dados de uma evacuação real (Yang et al., 2005). Essa dificuldade pode ser a grande responsável pelo desenvolvimento relativamente lento de pesquisas que levam em consideração o comportamento humano. No entanto, essa dificuldade também pode representar um desafio para muitos pesquisadores e atrair suas atenções para esse campo de pesquisa (Carneiro, 2012). Dentre os possíveis comportamentos que podem emergir na simulação de dinâmica de pedestres pode-se destacar os seguintes:

1. Fenômeno de ir com a multidão: também conhecido como comportamento de pastoreio, esse fenômeno é caracterizado pelo fato de alguém desistir de seu próprio raciocínio para seguir as ações da maioria a sua volta. Esse fenômeno é frequentemente observado em um processo de busca pela saída real, quando os agentes são seres humanos. Durante um processo de busca pela saída os agentes podem perder

Tabela 3.1: Abordagens para modelar evacuação coletiva (Zheng et al., 2009).

Abordagem	Indivíduos/ Grupos	Escala	SAT	Situações	Fenômenos Típicos
AC	Homogêneo, Heterogêneo	Microscópio	Discreto	Normal, Emergencial	Comportamento parentes, Arqueamento, Compressão, Efeito de obstrução, fricção e obstáculos Comportamento egresso e repulsão,, Encontrar caminho, Bi-direção do movimento, Troca de ritmo, Atraso, Seguindo o comportamento rotineiro, Entupimento, Formação de pista, Comportamento de massa e rebanho, Inércia e os efeitos do grupo, Mais rápido é mais lento.
LG	Homogêneo	Microscópio	Discreto	Normal, Emergencial	Fluxo livre, Obstrução, Prendendo o fluxo, Contracorrente de pessoas, Caminhar e rastreamento, Comportamento de pessoas cegas, Congestionamento, filas, Comportamento de rebanho, Arqueamento, entupimento.
SF	Homogêneo	Microscópio	Discreto, Contínuo	Normal, Emergencial	Formação de pista, Mudança oscilatória, Entupimento, arqueamento, Mais rápido é mais lento, Comportamento de massa, Multidão impaciente, Obstrução de saída.
FD	Homogêneo	Macroscópio	Contínuo	Normal, Emergencial	Comportamento de fluidos, Compressão de fluidos.
AB	Heterogêneo	Microscópio	Discreto Contínuo	Normal, Emergencial	Comportamento baseado em regras, Comportamento decisão e risco, Enfileiramento, Empilhamento, Pistas Propagação de pânico, Impaciência, Comportamento de rebanho e de filas, Comportamentos de desvio de obstáculos, Comportamento competitivo e colaborativo.
GT	Homogêneo	Microscópios	Discreto	Emergencial	Seleção de saída.
AE	Homogêneo	Microscópios	Não aplica	Emergencial	Filas auto-organizadas, Comportamento sem escala para pânico, Propagação do pânico, Comportamento de rebanho.

a calma e a noção da localização de saídas seguras (principalmente em caso de incêndio em que a visibilidade é baixa) e tendem a seguir outras na esperança de que elas conheçam a localização das saídas. As pessoas que têm uma tendência severa de ir com a multidão podem ser facilmente afetadas por outras pessoas ao seu redor (Burstedde et al., 2001), (Kirchner and Schadschneider, 2002) e (Yang et al., 2005).

2. Fenômeno de reunião: durante esse fenômeno as pessoas apresentam a característica de prevenção (ou seja, sentem que podem se proteger, psicologicamente, negando situações desagradáveis), e estão dispostas a permanecer próximas à multidão em uma emergência; o que pode resultar em recolhimento dos ocupantes no interior do edifício (Carneiro, 2012). Além disso, os ocupantes de um edifício frequentemente evacuam em grupos e reunir-se pode aliviar as suas ansiedades (Yang et al., 2005).



O efeito das pessoas se ajudarem mutuamente em casos de emergência é conhecido como efeitos de grupo (Zheng et al., 2009);

3. Retrocesso: pode ser entendido como o fato de retornar ao local de perigo para tentar salvar pertences, em busca de parentes ou até mesmo para informar ou ajudar outras pessoas (Yang et al., 2005).
4. Comportamento de parentesco: comportamento muito popular durante a processo de busca pela saída. É caracterizado pelo fato de que familiares tendem a se reunir durante uma processo de busca pela saída de emergência ou até mesmo a retroceder ao local de perigo para salvar um membro da família (Yang et al., 2002) e (Yang et al., 2005).
5. Arqueamento: Fenômeno reproduzido na saída. É causado pelo fato de todos os pedestres se moverem em direção à saída, formando um gargalo no fluxo de pedestres (Wei-Guo et al., 2006).
6. Mais rápido é mais lento: caracterizado pelo fato de que quanto mais rápido os agentes desejam se mover mais lenta pode se tornar o processo de busca pela saída (Zheng et al., 2009).
7. Efeito não aventureiro: a maioria das pessoas costuma usar saídas familiares ao evacuar um ambiente (Zheng et al., 2009).
8. Pânico: caracterizado como uma reação psicológica na qual o indivíduo fica em um estado anormal de ansiedade ou estado comportamental de aceitação ao estímulo de desastre (Carneiro, 2012). A reação psicológica causada no desastre ambiental pode fazer a capacidade adaptativa e a capacidade de resposta ao ambiente diminuírem em pouco tempo. No pior caso, os indivíduos irão perder seus sentidos e afetar a fuga (Liu et al., 2009), (Varas et al., 2007).

É importante salientar que esses são apenas alguns fenômenos e comportamentos que podem ser observados durante o processo de busca por uma saída. Visto que o foco deste trabalho não é investigar a psicologia humana, esses fenômenos servem apenas para mostrar que a psicologia e o comportamento humano desempenham um papel importante no processo inteligente de busca pela saída e que tais comportamentos podem emergir durante uma simulação mesmo que o foco do trabalho não seja estudá-los. Além disso, algum desses comportamentos podem ser importados para os sistemas de dinâmica de robôs de busca tornando o processo mais inteligente e eficiente se algum desses fenômenos for considerado.

### 3.4 Modelos de dinâmica coletiva baseados em ACs

Os modelos de dinâmica coletiva, onde os indivíduos são pedestres, foram inspirados nos trabalhos baseados em ACs para modelar o fluxo de tráfego para veículos (Barlovic et al., 1998; Esser and Schreckenberg, 1997). Até aquele momento, os estudos eram voltados para abordagens unidimensionais, sendo que o objetivo era criar um modelo que pudesse calcular ou se aproximar do estado estacionário. Além disso, nos trabalhos envolvendo o fluxo de carros, tem-se que uma partícula pode mover-se para mais de uma célula no mesmo instante de tempo (velocidade máxima  $v_{max} > 1$ ) e o procedimento de atualização mais adequado é o paralelo (síncrono) (Carneiro, 2012). A maior dificuldade de modelagem de dinâmica de pedestres era a natureza bidimensional inerente a esse sistema.

Posteriormente, vários modelos de natureza contínua foram investigados para tratar a dinâmica coletiva, como exemplo, modelo de forças sociais. Nesses modelos, os pedestres são tratados como partículas e forças sociais são introduzidas entre os indivíduos, sendo que, as equações de movimento se assemelham à mecânica Newtoniana. Por outro lado, os modelos discretos baseados em ACs que tratam este problema possuem uma natureza mais generalizada (Varas et al., 2007), uma vez que modelam forças em um nível microscópico.

O uso de autômatos celulares para a dinâmica de pedestres foi proposto em (Fukui and Ishibashi, 1999) e são considerados generalizações dos modelos Biham-Middleton-Levine (Ishibashi and Fukui, 1994) para o tráfego de veículos. Muitos trabalhos focam na maneira que o aumento da densidade de pessoas interfere no congestionamento e a maioria dos autores investigam  $v_{max} = 1$ , ou seja, a cada passo de tempo um pedestre pode se deslocar pela distância de uma célula. Caso  $v_{max} > 1$ , a implementação torna-se mais complexa em um grid bidimensional, além disso, pode reduzir a eficiência do modelo (Schadschneider, 2001a).

Os modelos de dinâmica coletiva para o processo de busca pela saída baseados em ACs pesquisados na literatura, em sua grande maioria, consistem de duas etapas principais: (i) definição de um campo de piso, geralmente estático, que pode ser ou não combinado com um piso dinâmico, e (ii) regras de movimentação dos pedestres. A primeira etapa refere-se à divisão do ambiente em uma malha uniforme (reticulado) apresentado na Figura 3.1 (a) e à transformação do ambiente 3D em um reticulado bidimensional, apresentado na Figura 3.1 (b).

O campo de piso consiste de um reticulado onde as saídas representam um valor mínimo. Além disso, os obstáculos são associados a células com estados no valor máximo e as demais células do reticulado possuem um valor intermediário entre o valor máximo e mínimo, de tal maneira que à medida que a célula se aproxima de uma saída ela possui um valor próximo ao valor mínimo. Posteriormente, cada agente (indivíduo), ocupa uma célula particular do reticulado (Figura 3.1 (b)) e move-se ao longo dessas posições de

acordo com uma regra de transição. As variáveis de cada célula podem ser atualizadas simultaneamente, a partir dos valores das variáveis da vizinhança no passo de tempo anterior, de acordo com a regra de transição. Diversos autores apresentaram abordagens de como utilizar ACs para simular dinâmica coletiva no processo de busca pela saída, descrevendo modos de controle do trânsito dos agentes e como fatores externos e internos afetam a busca pela saída.

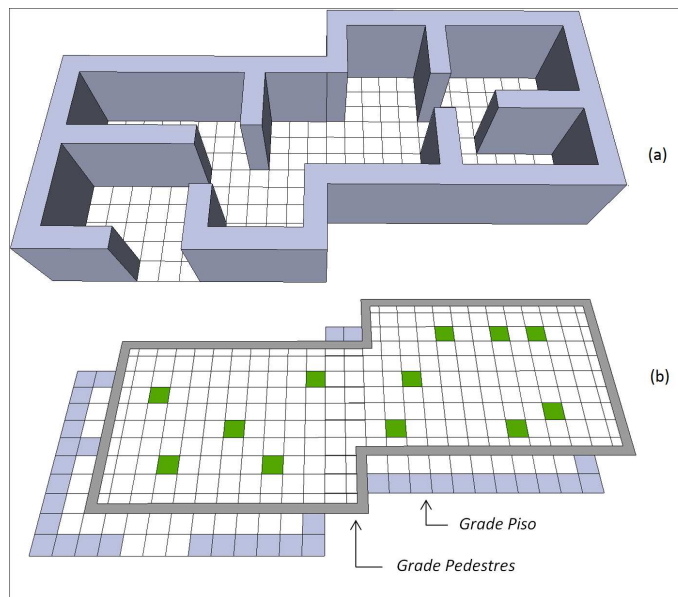


Figura 3.1: (a) Estrutura predial 3D. (b) Sobreposição da malha de pedestres na malha de campo de piso (Castro and Lima, 2013a).

A Tabela 3.2 apresenta uma visão geral dos modelos de dinâmica coletiva baseados em autômatos celulares investigados com base na literatura correlata. Os fatores analisados foram os mesmos abordados no trabalho de (Zheng et al., 2009). Os modelos assinalados com (a) representam modelos que utilizam apenas o campo de piso estático (b) os modelos que combinam a utilização do campo de piso estático e dinâmico simultaneamente. Os modelos assinalados com (1) representam os modelos de primeira escolha, com (2) modelos de transição probabilística e (3) modelos híbridos. Alguns dos modelos de primeira escolha e estocásticos utilizados para a formulação do modelo proposto neste trabalho serão detalhados nas seções a seguir.

### 3.4.1 Modelos de primeira escolha

Um modelo com regra de movimentação de primeira escolha é caracterizado pela escolha do próximo movimento realizado de maneira determinista. A escolha é realizada através de alguma regra direta, como exemplo, pode-se escolher para o próximo movimento a célula adjacente de menor valor (Castro and Lima, 2013a). Posteriormente, se existir um caso onde dois ou mais pedestres escolheram uma mesma célula para realizarem o movimento, deve-se resolver este conflito de maneira probabilística.

Tabela 3.2: Principais trabalhos de dinâmica coletiva baseados em autômatos celulares.

Referência	Abordagem	Indivíduos/ Grupos	Escala	Situações	Fenômenos Típicos
(Schadschneider, 2001a)	CA, b, 2	Homogêneo	Microscópio	Emergencial	Efeito de parentesco, Efeito de auto-organização.
(Perez et al., 2002)	CA, b, 2	Homogêneo	Microscópio	Emergencial	Arqueamento.
(Kirchner, 2002)	CA, b, 2	Homogêneo	Microscópio	Emergencial	Comportamento de rebanho.
(Weng et al., 2006)	CA, b, 2	Homogêneo	Microscópio	Normal	Formação de pista, Congestionamento.
(Kirchner et al., 2003b)	CA, b, 2	Homogêneo	Microscópio	Emergencial	Efeito de obstrução, Efeito de fricção.
(Kirchner et al., 2003a)	CA, b, 2	Homogêneo	Microscópio	Emergencial	Comportamento competitivo.
(Weifeng et al., 2003)	CA, a, 2	Homogêneo	Microscópio	Normal	Congestionamento, Movimento bi-direcional.
(Nishinari et al., 2004)	CA, b, 2	Homogêneo	Microscópio	Emergencial	Efeito de filas. Arqueamento.
(Jian et al., 2005)	CA, a, 2	Homogêneo	Microscópio	Normal	Efeito de afastamento, Efeito de retrocesso.
(Yang et al., 2005)	CA e SF, b, 2	Homogêneo	Microscópio	Emergencial	Comportamento de parentesco.
(Nishinari et al., 2006)	CA, b, 2	Homogêneo	Microscópio	Emergencial	Comportamento de seguir.
(Wei-Guo et al., 2006)	CA e SF, a, 1	Homogêneo	Microscópio	Emergencial	Arqueamento, Efeito de obstrução, Mais rápido é mais lento.
(Schultz et al., 2007)	CA, a, 2	Heterogêneo	Microscópio	Emergencial	Efeito de fricção, Efeito de repulsão, Encontrar caminho.
(Varas et al., 2007)	CA, a, 1	Homogêneo	Microscópio	Emergencial	Efeito de obstáculos.
(Yuan and Tan, 2007)	CA, b, 2	Homogêneo	Microscópio	Emergencial	Comportamento rotineiro, Comportamento de grupo, Efeito de inércia.
(Yamamoto et al., 2007)	CA e LG, b, 3	Homogêneo	Microscópio	Normal	Efeito de obstrução.
(Li-jun et al., 2009)	CA, b, 2	Homogêneo	Microscópio	Normal	Congestionamento.
(Alizadeh, 2011)	CA, b, 1	Heterogêneo	Microscópio	Emergencial	Efeito de obstrução, Arqueamento.
(Zheng et al., 2011)	CA, a, 2	Heterogêneo	Microscópio	Emergencial	Efeito de pânico Efeito do fogo.
(Pereira et al., 2011)	CA, b, 2	Homogêneo	Microscópio	Emergencial	Efeito de lotação.
(Carneiro, 2012)	CA, a, 1	Homogêneo	Microscópio	Emergencial	Efeito de inércia.
(Castro and Lima, 2013a)	CA, a, 1	Homogêneo	Microscópio	Emergencial	Congestionamento. Formação de filas.
(Hou et al., 2014)	CA, b, 1	Homogêneo	Microscópio	Emergencial	Efeito de liderança.
(Fu et al., 2015)	CA, b, 1	Heterogêneo	Microscópio	Emergencial	Efeito de diferentes possibilidades de caminhar
(Guo et al., 2016)	CA, a, 1	Homogêneo	Microscópio	Emergencial	Efeito de Densidade Efeitos de padrões.

## Modelo de Varas

Um modelo de movimentação de indivíduos sobre malhas baseados em ACs foi proposto em (Varas et al., 2007). Uma característica importante desse modelo é que ele é composto de duas etapas. Na primeira etapa, um campo de piso (malha) é demarcado para representar a planta de uma estrutura. Esse campo de piso é um reticulado onde cada célula representa um valor constante que indica a distância dessa célula até a saída/porta. Valores menores correspondem a células mais próximas à saída, e indica ao pedestre o caminho para a saída. Para esse modelo cada célula possui tamanho de  $0,4 \times 0,4 \text{ m}^2$ , que corresponde a superfície típico ocupado por uma pessoa numa situação

densa. A velocidade média dos pedestres é em torno de  $1\text{ m/s}$ , movendo-se  $0,4\text{ m}$  por passo de tempo, o que significa que,  $\Delta t \approx 0,4\text{ s}$  (Burstedde et al., 2001). Nesse modelo, os pedestres sempre preferem mover-se para uma célula de menor valor do que a célula atual. Inicialmente, o ambiente é dividido em uma malha retangular. A saída é atribuída com o valor “1”. Em seguida, todas as células adjacentes ao anterior (de uma segunda camada de células) é atribuído um valor, de acordo com as seguintes regras:

1. Se uma célula tem valor  $N$ , as células adjacentes nas direções vertical ou horizontal é atribuído um valor  $N + 1$ .
2. Para células diagonais será atribuído um valor  $N + \lambda$ , com  $\lambda > 1$ . Esta é uma simples tentativa de representar o fato de que a distância entre duas células adjacentes diagonalmente é maior do que na horizontal ou vertical. No trabalho de (Varas et al., 2007) foi escolhido  $\lambda = \frac{3}{2}$ .
3. Se houver conflitos na atribuição de um valor de uma célula, porque é adjacente à células com diferentes campos de piso, o valor mínimo é atribuído à célula em conflito.

A terceira camada de células é calculada, sendo esta adjacente a todas as células da segunda camada, sendo que, não é mais considerado as células da primeira camada. O processo é repetido até que todas as células sejam avaliadas. Paredes também são considerados como células do ambiente. Para estas células são dados valores muito elevados. Isto assegura que os indivíduos nunca tentarão ocupar uma destas células.

De acordo com (Varas et al., 2007), o resultado é semelhante à métrica de Manhattan, no sentido de que o campo de piso, em cada célula, representa o comprimento mínimo do caminho a partir de uma porta de saída para a célula. A diferença reside no fato de que no trabalho é permitido movimentos diagonais e, assim, foi necessário considerar movimentos diagonais ao calcular a distância entre a porta de saída e uma célula. A Figura 3.2 mostra o campo de piso obtido através da aplicação deste conjunto de regras a uma sala com uma porta de saída na parede à esquerda.

A segunda etapa apresenta um conjunto de regras de transição que permitem o deslocamento dos pedestres sobre essa malha e a interação dos mesmos. Nesse caso, o algoritmo que simula a evacuação contempla a solução de conflitos entre a própria multidão ou entre pessoas e os obstáculos.

Em cada iteração, cada pessoa deve decidir para onde ir. No modelo de (Varas et al., 2007), ele sempre decide se mudar para a célula mais próxima de saída. Essa atualização é realizada para a célula vazia adjacente com o campo de piso mais baixo, e isto é realizado para todos os pedestres de maneira síncrona. Assim, o modelo em princípio é determinista. No entanto, o autor apresenta três maneiras de torná-lo não-determinístico:

500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500
500	7.5	8.0	8.5	9.0	9.5	10	11	12	13	14	15	16	17	18	19	20	21	22	500	500
500	6.5	7.0	7.5	8.0	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	500	500
500	5.5	6	6.5	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	500	500
500	4.5	5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	500	500
500	3.5	4	5	500	8	9	10	11	12	13	14	15	16	17	18	19	20	21	500	500
500	2.5	3.5	4.5	500	9	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	500	500
1	2	3	4	500	10	10.5	11	12	13	14	15	16	17	18	19	20	21	22	500	500
1	2	3	4	500	11	11.5	12	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	22.5	500	500
500	2.5	3.5	4.5	500	10.5	11	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	22.5	500	500
500	3.5	4	5	500	9.5	10	11	12	13	14	15	16	17	18	19	20	21	22	500	500
500	4.5	5	5.5	500	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	500	500
500	5.5	6	6.5	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	500	500
500	6.5	7.0	7.5	8.0	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	500	500
500	7.5	8.0	8.5	9.0	9.5	10	11	12	13	14	15	16	17	18	19	20	21	22	500	500
500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500

Figura 3.2: Campo de piso apresentado no modelo de (Varas et al., 2007). Campo de piso para um ambiente com  $18 \times 14$  células. A porta de saída está na parede esquerda (células com um valor igual a 1); células correspondentes às paredes é atribuído um valor de 500 (Varas et al., 2007).

1. Se duas ou mais células vizinhas têm o menor valor do campo de piso, um número aleatório é utilizado para determinar a célula à qual a pessoa tem a intenção de se mover.
2. Se dois indivíduos decidem passar para a mesma célula, este conflito é decidido por um número aleatório. O vencedor se move, o perdedor não. Isto, aliado ao fato de que um indivíduo não pode mover-se para uma célula ocupada, são as únicas interações entre os indivíduos.
3. Uma certa quantidade de “pânico” é introduzida, dada por uma probabilidade (5% neste modelo) do indivíduo permanecer em sua posição, mesmo que ele possa se mover.

Segundo (Varas et al., 2007), quando um pedestre deseja movimentar-se para uma célula já ocupada, um conflito deve ser resolvido, conforme exemplificado na Figura 3.3. Na figura, os números correspondem ao valor do campo de piso em uma célula, os arcos representam as intenções de movimento, e os símbolos representam os pedestres em várias situações. Círculo preenchido: o pedestre movimentar-se para uma célula adjacente para minimizar o seu campo de piso (movimento padrão). Círculo aberto: dois pedestres tentam se movimentar para a mesma célula, um valor aleatório decide quem se movimenta. Triângulo preenchido: duas células adjacentes com o mesmo campo de piso, os pedestres decidem randomicamente para onde mover. Triângulo aberto: pedestres em pânico, não se movimentam.

Para a avaliação dos resultados obtidos neste trabalho considerou-se o tipo de arqueamento formado ao alterar alguns dos parâmetros, conforme é apresentado na Figura 3.4. O arqueamento é a curvatura que é observada dos pedestres frente a saída. Outra avaliação realizada foi a comparação do tempo de evacuação variando a quantidade de

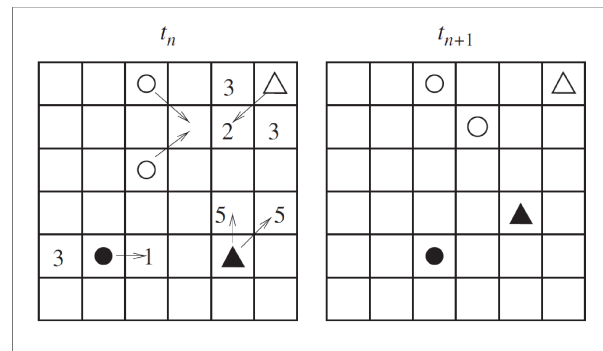


Figura 3.3: Movimentos possíveis para os pedestres no modelo de (Varas et al., 2007).

peçoas dentro de um ambiente com e sem obstáculos e a largura da saídas. Outro fator considerado foi a melhor posição para fixar a saída de um determinada sala. Para este experimento, foi utilizada uma sala de aula com 50 indivíduos previamente alocados em suas posições e posteriormente mudou-se as saídas e anotou-se qual o tempo que os pedestres gastaram para evacuar o ambiente em cada uma delas (Varas et al., 2007). A melhor saída para o ambiente é aquela cujo menor tempo foi observado.

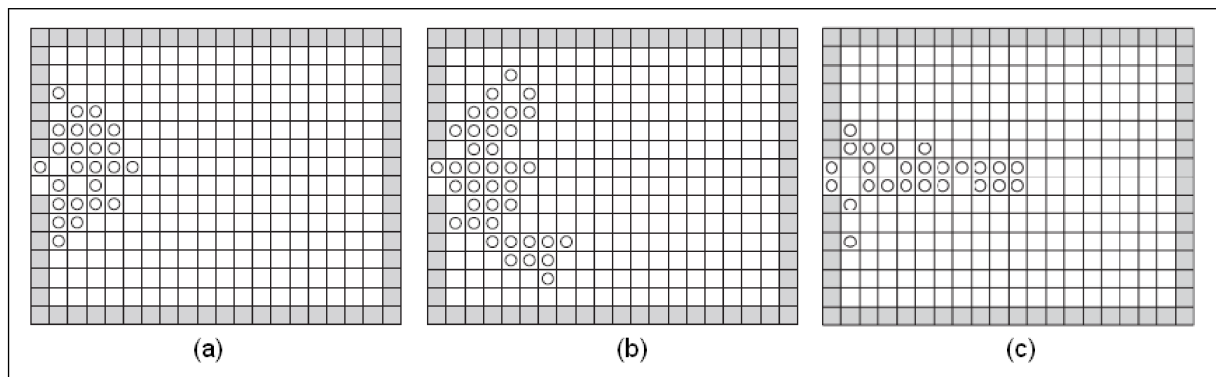


Figura 3.4: Tipos de arqueamento observados no modelo de (Varas et al., 2007) em ambiente com 50 indivíduos após 30 passos de iteração. (a) Campo de piso proposto por (Varas et al., 2007). (b) Campo de piso sem a inclusão de valor de diagonal. (c) Campo de piso utilizando métrica de Manhattan.

### Modelo de Alizadeh

Em (Alizadeh, 2011) também foi proposto um modelo para simular a evacuação em salas com obstáculos. O objetivo foi analisar os efeitos dos obstáculos e da distribuição da multidão no processo de evacuação. Para isso, o autor utilizou o campo de piso estático definido por (Varas et al., 2007) e propôs um novo modelo dinâmico de autômatos celulares que leva em consideração a distância das células à saída, em relação às métricas que podem ser aplicadas. O modelo também considera o número de pedestres que estão mais próximos de uma determinada célula, almejando uma determinada saída, em um determinado passo de tempo. O modelo provê a utilização de dois campos de piso no processo de simulação, um campo de piso estático obtido através do cálculo proposto em (Varas et al., 2007)

e um campo de piso dinâmico obtido através da combinação do campo de piso estático com um atributo que fornece o número de pedestres que estão mais próximos de uma determinada célula, visando uma determinada saída. O ambiente é dividido em uma malha retangular. A saída é atribuída com o valor “1”. Em seguida o campo de piso estático é calculado, para todas as células é atribuído um valor  $W_{static}^A(x)$ , de acordo com o cálculo do campo de (Varas et al., 2007). Posteriormente o campo de piso dinâmico é calculado para determinar o peso  $W_i^A(x)$  e para isso são considerados dois parâmetros:

1.  $W_{static}^A(x)$ : a distância de  $x$  até a saída  $A$ , com a respectiva métrica aplicada.
2.  $T_i^A(x)$ : O número de pessoas que estão mais próximas de  $x$  que de  $A$  no  $i$ -ésimo passo de tempo.
3. Assim, tem-se que,  $W_i^A(x) = W_{static}^A(x) + \alpha \cdot T_i^A(x)$ .

Sendo que,  $d_A$  é dada pela largura da porta de saída e  $\alpha$  é o coeficiente para evitar a multidão, as demais variáveis são dadas por:

1.  $T_i^A(x) = \frac{P_i^A(x) + \frac{1}{2}E_i^A(x)}{d_A}$
2.  $V_i = \{y \mid y \text{ é ocupado por uma pessoa no instante de tempo } i\}$ .
3.  $P_i^A(x) = \{y \mid W_{static}^A(y) < W_{static}^A(x) \text{ e } y \in V_i\}$ .
4.  $E_i^A(x) = \{y \mid W_{static}^A(y) = W_{static}^A(x) \text{ e } y \in V_i\}$ .

Por fim, o peso de cada célula é calculado da seguinte maneira:  $W_i(x) = \min(\{W_i^A(x) \mid A \text{ é uma saída}\})$ .

De acordo com os autores, se  $\alpha$  vale 0, então o algoritmo se comporta de maneira idêntica ao algoritmo de (Varas et al., 2007). À medida que  $\alpha$  aumenta, os pedestres tendem a mover-se para as saídas menos congestionadas. Para a realização do movimento, cada pedestre escolhe a menor célula vizinha. Em caso de conflitos, utilizar o algoritmo de resolução de conflitos proposto em (Varas et al., 2007). De acordo com os experimentos propostos em (Alizadeh, 2011), esse modelo evita a formação de filas longas próximo às saídas, o que diminui o tempo de evacuação total. Além disso, posições ajustadas das saídas proporcionam uma evacuação mais adequada, conforme Figura 3.5. Se o número de pedestres for distribuído uniformemente para cada uma das saídas no primeiro passo de tempo  $t = 1$ , o tempo de evacuação diminui. Por outro lado, se vários pedestres selecionam uma porta específica, o tempo de evacuação, no primeiro passo de tempo  $t = 1$ , o tempo  $T$  de evacuação ainda continua baixo. Em comparação ao modelo estático de (Varas et al., 2007), ao aumentar o tamanho da porta, o tempo total de evacuação  $T$  é reduzido.



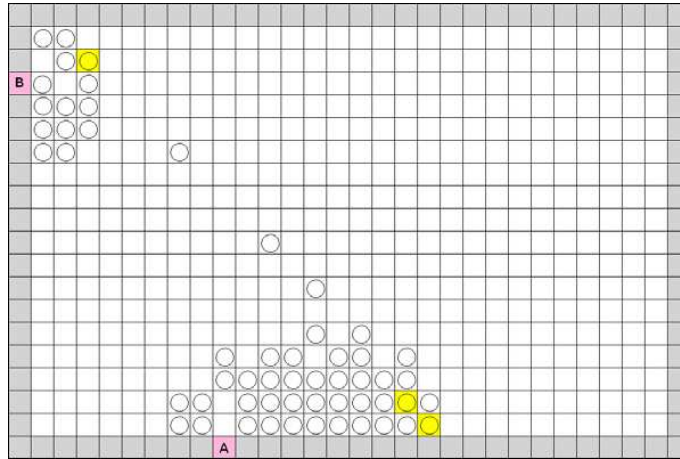


Figura 3.5: Exemplo de evacuação no modelo de Alizadeh (2011) onde a maioria dos pedestres escolheram a porta A para a saída.

### Modelo de Wei-Guo

No trabalho de (Wei-Guo et al., 2006) os autores abordaram os efeitos das forças sociais (SF) físicas, fricção e repulsão, em seus trabalhos. A repulsão indica que o efeito que ocorre quando vários pedestres desejam se movimentarem para uma mesma célula, ou quando um pedestre movimenta-se para uma célula adjacente a um pedestre paralisado ou adjacente a uma célula de obstáculo Figura 3.6. O fenômeno de fricção corresponde a movimento do pedestre passar entre outro pedestre movimentando ou entre um pedestre parado e uma parede Figura 3.7. Todas essas forças foram aplicadas para verificar o poder de atração dos ocupantes do reticulado.

Para este modelo, foi considerado que o tamanho de cada célula é de  $40 \times 40$  cm. Um campo de piso estático  $S_i = \|x_i - y_{exit}\| + \|y_i - x_{exit}\|$ . Os pedestres movem-se ao longo do reticulado para as saídas guiados pela força do campo de piso. Neste caso, a força do campo de piso possui atração inversamente proporcional ao valor do campo de piso, uma vez que as saídas neste modelo possuem valor zero (Wei-Guo et al., 2006).

No primeiro instante, todos os pedestres tendem a mover-se para uma célula de menor valor na sua vizinhança, o que é chamado pelos autores de primeira escolha. Em cada passo de iteração, primeiramente será calculado todas as primeiras escolhas dos pedestres, posteriormente, as suas primeiras escolhas são modificadas, considerando as influências da repulsão e fricção (Wei-Guo et al., 2006).

Primeiramente, o fenômeno físico de repulsão será apresentado (Wei-Guo et al., 2006). Os passos para resolver esse tipo de interação entre os pedestres está apresentado a seguir.

1. Se o conflito for entre pedestres, a probabilidade de movimento de cada pedestre  $i$  é dada por  $p_i = \frac{(1-r_1)}{m}$ , tal que  $(i = 1, 2, \dots, m)$  e  $r_1$  é a probabilidade de cada pedestre escolher a célula desejada. Os pedestres que perderem ficam parados, o vencedor move-se.
2. Se conflito for entre um pedestre estático e um pedestre em movimento, a probabi-

lidade é dada por  $p = 1 - r_2$ , tal que  $r_2$  representa a probabilidade de repulsão de um pedestre estático e um em movimento.

3. Se conflito for entre um pedestre parado e um obstáculo, a probabilidade é dada por  $p = 1 - r_w$ , tal que  $r_w$  representa a probabilidade de repulsão de um pedestre em movimento e um obstáculo.
4. Os valores de  $r$  são calculados através da seguinte função sigmoidal:

$$r = \frac{1 - e^{-\alpha V}}{1 + e^{-\alpha V}}$$

sendo que,  $V$  representa a velocidade relativa e  $\alpha \in [0, \infty]$  representa o grau de dureza, o que corresponde a limite de resistência dos indivíduos à colisão física, bem como a rigidez do indivíduo ou parede.

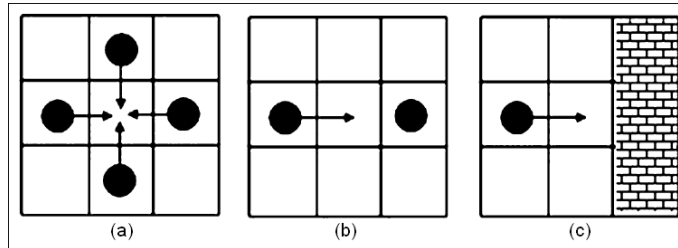


Figura 3.6: Casos da força física de repulsão (Wei-Guo et al., 2006). (a) Caso de repulsão entre quatro agentes em movimento para uma mesma célula. (b) Caso de repulsão entre um agente em movimento e um agente estático. (c) Caso de repulsão entre um agente em movimento e um obstáculo.

Neste modelo foi considerado para repulsão entre dois ou mais pedestres em movimento,  $V = 2v$ , para repulsão de um pedestre em movimento com parede ou pedestre estático,  $V = v$ , sendo que  $v$  denota a velocidade desejada. A probabilidade de fricção é usada para descrever os efeitos do fenômeno de fricção (Wei-Guo et al., 2006). A interação dos pedestres com esse fenômeno está apresentada a seguir:

1. Se o atrito for entre pedestres em movimento, a probabilidade de movimento de cada pedestre  $i$  é dada por  $p_i = 1 - f_1$ , tal que  $f_1$  é a probabilidade de fricção de dois pedestres em movimento.
2. Se conflito for entre um pedestre estático e um pedestre em movimento, a probabilidade é dada por  $p = 1 - f_2$ , tal que  $r_2$  representa a probabilidade de fricção de um pedestre estático e um em movimento.
3. Se conflito for entre um pedestre parado e um obstáculo, a probabilidade é dada por  $p = 1 - f_w$ , tal que  $r_w$  representa a probabilidade de fricção de um pedestre em movimento e um obstáculo.

4. Os valores de  $f$  representam a probabilidade de fricção, no qual é determinado pela distância relativa, velocidade relativa e o coeficiente de fricção dos objetos envolvidos. Assim,  $f$  pode ser descrita como:

$$f = \theta \cdot V$$

sendo que  $\theta \in [0, 1]$  é um coeficiente do raio refletido pela repulsão na fricção.

5. Se  $V$  for muito grande usar:  $f_n = \theta \cdot r_n$

Para este trabalho foram avaliados os efeitos de arqueamento, que aumenta diretamente proporcional com o aumento da velocidade. O congestionamento perto das saídas, que é aumentado quando a velocidade é aumentada, pois influencia o aumento de forças físicas de repulsão e fricção. Efeito de fluxo irregular, que mostra qual é o tempo de saída entre dois pedestres (Wei-Guo et al., 2006). Também foram avaliados os efeitos com diferentes larguras de portas, e mostra que o tempo de evacuação é inversamente proporcional à largura das portas de saída.

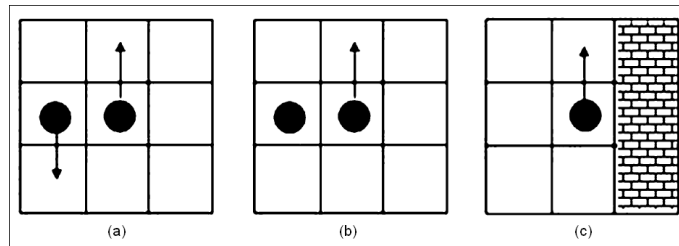


Figura 3.7: Casos da força física de fricção (Wei-Guo et al., 2006). (a) Caso de fricção entre dois agentes em movimento. (b) Caso de fricção entre um agente em movimento e um agente estático. (c) Caso de fricção entre um agente em movimento e um obstáculo.

### 3.4.2 Modelos estocásticos

Um modelo com regra de movimentação estocástica (ou probabilística) é caracterizado pela escolha do próximo movimento realizado de maneira não-determinista (Schadschneider, 2001a). Isto é, a escolha é realizada através de alguma função de probabilidade, como exemplo, pode-se escolher para o próximo movimento uma célula adjacente qualquer, sendo que as células com menor peso possuem mais chances de serem escolhidas. Essa escolha é denominada de movimento estocástico. Posteriormente, se existir um caso onde dois ou mais pedestres escolheram uma mesma célula para realizarem o movimento, deve-se resolver este conflito de maneira probabilística.

#### Modelo de Schadschneider

O trabalho de (Schadschneider, 2001a) para a simulação de dinâmica coletiva baseada em ACs, onde os indivíduos são pedestres, foi inspirado nos trabalhos precursores de fluxo de tráfego para veículos (Schadschneider and Schreckenberg, 1993).

Nesse trabalho, foi utilizado o princípio de que as partículas possuem um certo nível de inteligência para que seja alcançado o efeito de coletividade. Para isto, foi utilizado o conceito de campo de piso (FFM) (Schadschneider, 2001a). Este campo de piso possui células as quais são carregadas e modificadas de acordo com a ocupação do ambiente, sendo que este campo sofre algumas alterações (campo de piso dinâmico). Os valores do campo de piso são alterados de acordo com o coeficiente de difusão e declínio.

No modelo de (Schadschneider, 2001a) utilizou-se um campo de piso bidimensional, que pode ser fechado periodicamente em uma ou em ambas as direções. Cada célula pode estar livre ou ocupada por exatamente uma partícula. A atualização é realizada de forma síncrona para todas as células. As interações entre os pedestres são repulsivas para pequenas distâncias e atrativas para longas distâncias. Foi introduzido no modelo uma escala de tempo na dinâmica que pode ser identificadas como o tempo de reação  $t_{reac}$  ou  $\Delta t$ . Um pedestre pode mover-se com uma velocidade constante, isto é, 40 *cm* por passo de tempo (**tamanho da célula**). Empiricamente, a velocidade média de um pedestre é aproximadamente 1,3 *m/s* (Weidmann, 1992). Isso significa que está de acordo com as regras microscópicas, ou seja, é de aproximadamente  $\Delta t \approx 0,3$  *s*, da ordem do tempo de reação.

A cada partícula é dada uma direção de preferência. A partir desta direção, uma **matriz de preferências**  $3 \times 3$  é construída e contém as probabilidades de um movimento da partícula. O elemento central descreve a probabilidade de a partícula ficar parado ( $M_{0,0}$ ) e as 8 restantes (vizinhança de Moore) correspondem a uma mudança para a célula vizinha (quando raio do AC  $r = 1$ ). As probabilidades podem ser relacionadas com a velocidade e a longitudinal e desvios padrões transversais (Schadschneider, 2001a). Por exemplo, a matriz pode ser calculada a partir de um modelo de roteamento que designa rotas para cada um dos pedestres. No caso mais simples, é permitido que o pedestre se mova em uma única direção da matriz de preferência, sendo esta apenas um elemento e todos os outros são zero, conforme apresentado na Figura 3.8.

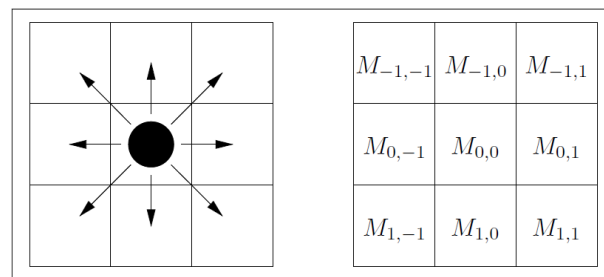


Figura 3.8: Movimentos possíveis para um pedestre no modelo de (Schadschneider, 2001a) e matriz de preferências.

Em cada etapa de atualização, para cada partícula um movimento desejado é escolhido a priori de acordo com as probabilidades. Isto é feito em paralelo para todas as partículas. Em cada iteração a posição de cada pedestre é atualizada de acordo com as probabilidades

de transição que será detalhada a seguir.

1. Se a célula alvo está ocupada, então o indivíduo não se move .
2. Se ela não está ocupada e não tem outro indivíduo que tenha como alvo a mesma célula, então o movimento é executado.
3. Se mais de um indivíduo tem a mesma célula-alvo, então um deles é escolhido de acordo com as probabilidades relativas com que cada indivíduo escolheu seu alvo, conforme é apresentado na Figura 3.9.

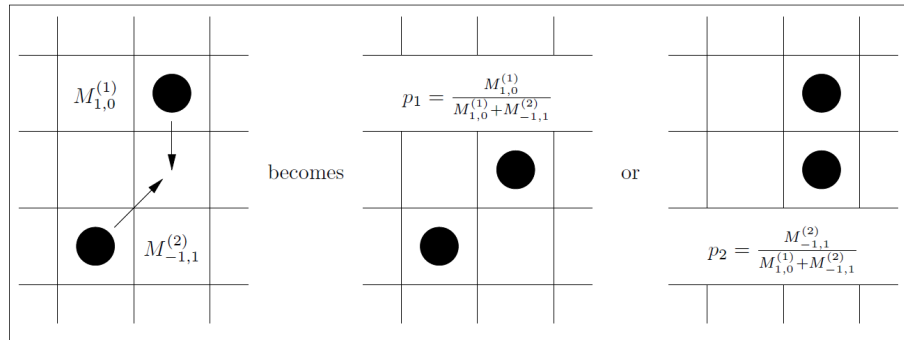


Figura 3.9: Resolvendo conflitos de acordo com as probabilidades relativas para o caso de dois indivíduos com matrizes de preferência  $M^{(1)}$  e  $M^{(2)}$  (Schadschneider, 2001a).

Para evitar as interações de maneira explícita entre os indivíduos, um campo de piso foi introduzido, permitindo que as interações entre o ambiente e os pedestres aconteçam de maneira unificada e simples. O campo de piso modifica a probabilidade de transição à medida que os movimentos/pegadas tendem a ser maiores em uma determinada região.

O **campo de piso** pode ser entendido como um segundo grid de células que se sobrepõe ao grid de pedestres. Este grid pode ser discreto ou contínuo, e ainda, estático ou dinâmico.

O **campo de piso estático**  $S$  não considera a quantidade de vezes que uma determinada célula é visitada. Ou seja, ele não muda ao longo das iterações. Este campo de piso é utilizado para modelar regiões de espaço que são mais atrativas que outras, como saídas de emergência ou janelas (Schadschneider, 2001a).

Por outro lado, o **campo de piso dinâmico**  $D$  é modificado pela presença de pedestres e tem sua própria dinâmica, difusão e declínio. Cada pedestre deixa um traço na célula, ou seja, o peso da célula é incrementado e aquela região/célula torna-se mais atrativa para outros pedestres (Schadschneider, 2001a). Os passos de transição do campo de piso dinâmico é dado a seguir.

1. Se um pedestre deixa uma célula  $(x, y)$  o campo de piso dinâmico dessa célula é aumentado em  $\delta D_{xy}$ . O incremento  $\delta D_{xy}$  é um parâmetro do modelo e pode ser discreto ou contínuo.
2. Para modelar a difusão uma certa quantidade do campo  $\delta D_{xy}$  é distribuída para as células vizinhas.

3. A força do campo em cada célula é diminuída por uma constante  $\omega$  para simular o decaimento do campo.

A trilha apresentada no passo 1 é criada pelo movimento dos pedestres, o passo 2 é necessário porque os pedestres não seguem exatamente o mesmo caminho dos outros, e o passo 3 implica que a trilha virtual tem um tempo de vida finito, ou seja, “evapora” com o passar do tempo.

No geral, a probabilidade de transição  $p_{ij}$  na direção  $(i, j)$ , conforme apresentado na Figura 3.8 é dado pela equação abaixo.

$$p_{ij} = NM_{ij}D_{ij}S_{ij}(1 - n_{ij})$$

Essa probabilidade de transição  $p_{ij}$  na direção  $(i, j)$  depende agora de quatro contribuições de acordo com as seguintes regras de probabilidade:

1. A matriz de preferências  $M_{ij}$  que contém informação sobre a rota do pedestre.
2. O valor  $D_{ij}$ , do campo de piso dinâmico da célula  $(i, j)$ .
3. O valor  $S_{ij}$  para o campo de piso estático.
4. A ocupação  $n_{ij}$  da célula  $(i, j)$ . Um movimento na direção  $(i, j)$  apenas é permitido se a célula está vazia ( $n_{ij} = 0$ ). Se a célula estiver ocupada ( $n_{ij} = 1$ ), o movimento é proibido.

Sendo que  $n_{ij}$  representa a ocupação de uma determinada célula, e  $n_{ij} = 1$  para uma célula ocupada e  $n_{ij} = 0$  para uma célula vazia.  $N$  é um fator de normalização para assegurar que  $\sum_{(ij)} p_{ij} = 1$ , onde a soma é superior às nove possíveis células alvo (Schadschneider, 2001a). A regra de atualização do modelo completo está apresentada a seguir e depende dos conceitos previamente descritos: tamanho da célula, matriz de preferências, colisão, campo de piso dinâmico, campo de piso estático e probabilidade de transição.

1. O campo de piso dinâmico  $D$  é modificado com as regras de difusão e declínio.
2. Para cada pedestre, a probabilidade de transição para mover-se para uma célula não ocupada  $(i, j)$  é determinada pela matriz de preferências e pelos campos de piso estático e dinâmico,  $p_{ij} \propto M_{ij}D_{ij}S_{ij}$ .
3. Cada pedestre escolhe uma célula alvo baseado nas probabilidades da matriz de transição  $P = (p_{ij})$ .
4. Os conflitos são resolvidos quando dois ou mais pedestres tendem a mover-se para a mesma célula alvo.
5. Os pedestres que estão permitidos para mover-se, executam seus movimentos.

6. Os pedestres alteram a célula  $(x, y)$  do campo de piso dinâmico  $D_{xy}$  que eles ocupavam antes do movimento.

No trabalho de (Schadschneider, 2001a) foram avaliadas duas situações típicas: evacuação de uma sala grande (Figura 3.10) e a formação de pistas em um corredor longo. Para a simulação foram usados diferentes variantes do modelo básico para validar o potencial das diferentes abordagens. Foi constatado que a partir do momento que a força de interação é diminuída, melhores tempos na evacuação são atingidos.

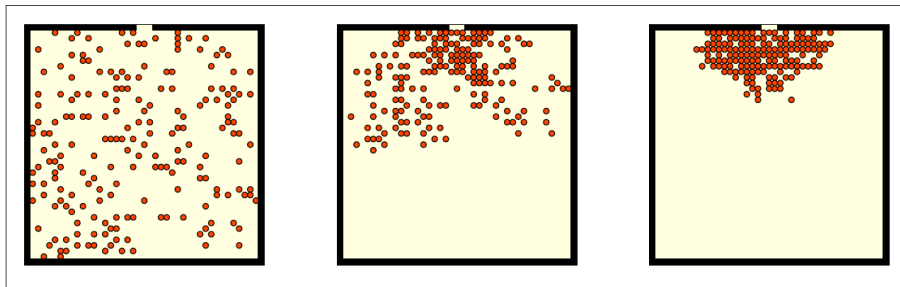


Figura 3.10: Pessoas saindo de uma sala com única porta. Foram apresentados três estágios da dinâmica durante a evacuação (Schadschneider, 2001a).

Schadschneider e colaboradores (Schadschneider, 2001b) e (Burstedde et al., 2001) realizaram como experimentos a simulação da dinâmica em três cenários. O primeiro foi a evacuação de uma sala grande, os autores não informaram a dimensão da sala. Neste trabalho de (Burstedde et al., 2001), verificou-se o comportamento de 100 pessoas posicionadas aleatoriamente. Assumiu-se que as pessoas não tinham informações de onde era a saída, tal como em uma situação de emergência, fumaça ou luzes. O campo de piso estático foi modelado de maneira a ter um valor máximo próximo a saída e ir decrescendo até atingir o valor 0 na parede oposta a saída. Neste caso o comportamento dominante verificado foi que o pedestre tende a seguir seus antecessores a fim de encontrar a saída.

Um outro cenário avaliado em (Burstedde et al., 2001) foi a evacuação de um auditório, uma das características desses ambientes é que eles são salas grandes com diversas fileiras de bancos que servem de obstáculos horizontais para a saída de pedestres. No cenário avaliado, as pessoas deveriam sair por uma porta frontal e o auditório estava completamente ocupado e apenas o efeito do campo de piso estático foi avaliado. Neste cenário, foram observado engarrafamentos no corredor central e um menor fluxo nos dois corredores laterais, caso típico que também ocorre na realidade.

A última simulação que foi realizada verificou-se a otimização de tempos de evacuação. Neste cenário, a estrutura do ambiente foi modificada diversas vezes comparações entre diversos ambientes foram efetuados, como por exemplo, aumentar a largura da saída, aumentar a quantidade de saídas e mudar a localização da saída (Schadschneider, 2001a). Assim, foi avaliada qual a estrutura predial que apresenta maior segurança para os ocupantes, isto é, a simulação que resulta no menor tempo de evacuação.

Para o modelo de (Schadschneider, 2001a), e a partir das simulações apresentadas em (Schadschneider, 2001a), (Schadschneider, 2001b), (Burstedde et al., 2001), (Kirchner and Schadschneider, 2002) constatou-se que é possível reproduzir os efeitos de auto-organização conhecidos por ocorrerem em dinâmica de pedestres. Logo, este é um bom modelo para aplicações realistas (Schadschneider, 2001a). Além disso, cenários complexos podem ser simulados, tais como cenários com mais de uma espécie ou comportamento de pedestre. Neste caso, é possível adaptar para que cada tipo de pedestre interaja com o seu próprio campo de piso. Outra característica interessante é a possibilidade de cada pedestre seguir uma rota estabelecida por um algoritmo de roteamento que determina a cada passo de interação a matriz de preferência. Através dos exemplos de simulação, constatou-se que o modelo é capaz de simular complexos padrões de evacuação, assim como contribuir para análise nos riscos de um processo de evacuação, especialmente em se tratando de uma situação emergencial.

### Modelo de Nishinari

O modelo de campo de piso foi investigado e estendido em (Nishinari et al., 2004). Os autores apresentaram um método para calcular a intensidade do campo de piso estático em uma geometria arbitrária de salas. Eles propuseram a inserção de um parâmetro chamado “potencial de parede” - caracterizado como um potencial repulsivo inversamente proporcional à distância das paredes - na tentativa de simular o fato de que as pessoas evitam andar muito próximas a paredes e/ou obstáculos. Além desse parâmetro, (Nishinari et al., 2004) consideraram um efeito que eles chamaram de “efeito de contração”. Esse efeito nasceu da observação de que as pessoas tendem a andar para o centro da saída a fim de evitar as paredes. Assim, os autores passaram a considerar apenas a largura efetiva das saídas no cálculo do campo de piso estático, isto é, apenas células de saída próximas do centro da porta foram consideradas. Além disso, (Nishinari et al., 2004) investigaram o efeito da posição dos obstáculos no tempo total de evacuação. Eles chegaram à conclusão de que a posição dos obstáculos em uma sala afeta a evacuação através do campo de piso estático.

Em (Nishinari et al., 2006) foi proposto um modelo unidimensional para partículas auto-dirigidas, interagindo formigas, ACs e pedestres (Cellular Ants). O objetivo do trabalho era comparar o efeito do feromônio nas trilhas de formigas com o efeito criado pelo algoritmo de campo de piso dinâmico (FFM), que simula o feromônio em quimiotaxia (Schadschneider, 2001a). Nos modelos de trilhas de formigas (ATM) as formigas caminham unidirecionalmente por uma trilha. O feromônio deixado por elas ao longo do caminho faz com que outras formigas sintam o seu cheiro e sigam até o objetivo. No modelo proposto em (Nishinari et al., 2006) as formigas não movem-se para trás, uma vez que o modelo é unidimensional e tem sua probabilidade aumentada de pular para a próxima célula caso sintam presença de feromônio. A seguir são apresentados os passos



do modelo para a construção do campo de piso.

1. A trilha unidimensional é dividida em  $L$  células, cada uma delas pode acomodar uma única formiga por passo de tempo e recebem um índice  $i = \{1, 2, \dots, L\}$ .
2. Para cada célula é associado um valor  $S_i$  e um valor  $\sigma_i$ .
  - (a)  $S_i$  representa se a célula está ocupada por uma formiga,  $S_i = 1$  se está ocupada,  $S_i = 0$ , caso contrário.
  - (b)  $\sigma_i$  representa se a célula está tem feromônio,  $\sigma_i = 1$  se tem feromônio,  $\sigma_i = 0$ , caso contrário.

O sistema é atualizado a cada passo de tempo em dois estágios. O primeiro estágio apresentado a seguir mostra como as formigas podem mover-se.

1. Uma partícula que está em uma célula  $i$  vazia,  $S_i(t) = 1$  e  $S_{i+1}(t) = 0$ , pode mover-se para frente com uma probabilidade  $Q > q$  (feromônio é mais atrativo):
  - (a)  $Q$ , se  $\sigma_{i+1}(t) = 1$ ,
  - (b)  $q$ , se  $\sigma_{i+1}(t) = 0$

O segundo estágio, mostrado a seguir, apresenta o método para evaporação do feromônio. Em cada passo as regras dinâmicas do AC são aplicadas de forma síncrona a todas as células do reticulado.

1. Para cada célula  $i$  ocupada por uma partícula antes do primeiro estágio, tem-se que um feromônio será criado  $\sigma_i(t+1) = 1$ , se  $S_i(t+1) = 1$
2. Para cada célula  $i$  não ocupada por uma partícula antes do primeiro estágio, tem-se que um feromônio será evaporado com uma probabilidade  $f$  por passo de tempo, isto é, se  $S_i(t+1) = 0$  e  $\sigma_i(t) = 1$ , então:
  - (a)  $\sigma_i(t+1) = 0$  com probabilidade  $f$ ,
  - (b)  $\sigma_i(t+1) = 1$  com probabilidade  $1 - f$ .

Esse trabalho tinha como objetivo criar um modelo para o comportamento das formigas e contrastar com o modelo de campo de piso dinâmico, proposto em (Schadschneider, 2001b). Para isto, além do autor validar o modelo via software, o autor também experimentou o sistema com um conjunto de robôs, conforme apresentado na Figura 3.11. A comunicação do sistema de robôs era feita através de feromônio. Foi demonstrado o modelo de ACs baseado em colônias de formigas (Cellular Ants) mostra uma dependência não-monotônico incomum com a média da velocidade das formigas na sua densidade. Mostrou-se também que este comportamento anômalo é claramente observado em um experimento de vários robôs (Nishinari et al., 2004). Em seguida, a relação entre o modelo e

a fuga da formiga é discutido com o modelo de campo de piso para estudar a dinâmica de evacuação de pedestres. Este último é considerado como uma generalização bidimensional do modelo de fuga da formiga, onde a quimiotaxia, neste caso é considerado o feromônio, é substituído por pegadas. Mostra-se a partir de simulações que pequenas perturbações para os pedestres, às vezes, podem evitar o congestionamento e, conseqüentemente, permitir evacuação segura (Nishinari et al., 2004).

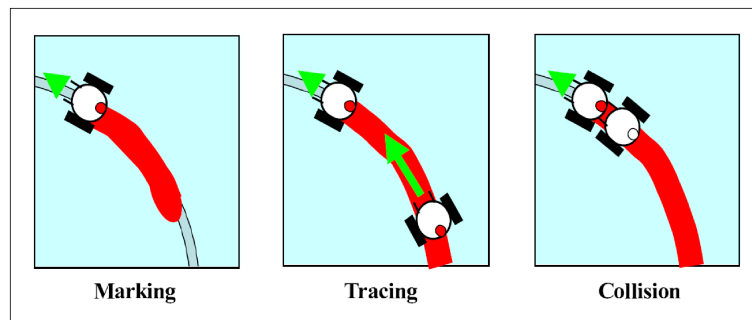


Figura 3.11: Interações entre robôs através da cor projetado no campo de piso (Nishinari et al., 2006).

### Modelo de Zheng

O modelo de (Zheng et al., 2011) investiga a dinâmica durante a evacuação de pedestres durante a influência de espalhamento de fogo. O modelo segue uma noção de escolha de movimento probabilística, baseada em uma função de probabilidade similar à utilizada em (Schadschneider, 2001b). No entanto, no trabalho de (Zheng et al., 2011), um novo campo de piso é proposto considerando a introdução do fogo e é adicionado à função de probabilidade para a escolha do novo movimento. Assim, a intensidade do campo de piso é uma força dos campos de piso, estático, dinâmico e do fogo.

Cada pedestre move-se na sua vizinhança de Von Neumann. As simulações apresentadas no experimento foram realizadas para observar o comportamento dos pedestres durante o processo de evacuação. A influência dos parâmetros de força do peso e taxa de espalhamento do campo de piso fogo foram analisadas em detalhes. Os resultados demonstraram que a taxa de evacuação dos pedestres está relacionada com o posicionamento inicial do foco de incêndio. Além disso, alguns experimentos foram realizados considerando-se o fator de pânico dos pedestres. O foco de incêndio e sua propagação ocorrem radialmente ao redor do foco de incêndio, ou seja, nenhuma regra mais complexa ou baseada em autômatos celulares (Lima and Lima, 2014) foi utilizada para realizar o espalhamento do fogo. A Figura 3.12 representa a evacuação do modelo de (Zheng et al., 2011), onde as células em vermelho representam o fogo, as células azul escuro representam os pedestres mais lentos e as células em azul claro representam os pedestres mais rápidos.

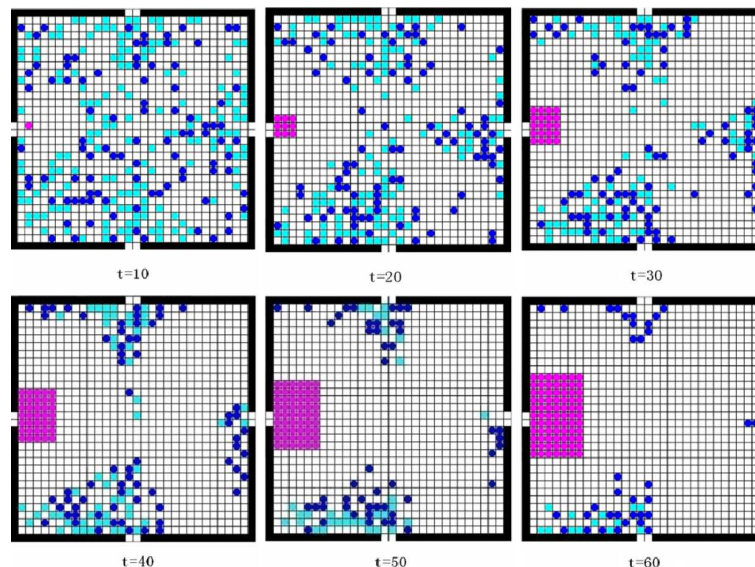


Figura 3.12: Evacuação de pedestres num ambiente em chamas no modelo de (Zheng et al., 2011)

### 3.5 Comentários em relação aos modelos propostos

O Capítulo 5 apresenta um novo modelo proposto para o controle de robôs. Esse modelo se inspirou em diversas estratégias propostas nos modelos anteriores baseados em ACs para a dinâmica coletiva e também nos principais conceitos sobre sistemas bio-inspirados apresentados. O modelo inicial proposto pode ser mais fortemente associado aos modelos de dinâmica de pedestres em (Varas et al., 2007) e (Alizadeh, 2011). O campo de piso estático é baseado no trabalho proposto em (Varas et al., 2007), mas poderia ser alterado pelas métricas de Manhattan ou Euclidiana (Schadschneider, 2001b). A inserção do campo de piso dinâmico associado ao campo de piso estático foi baseado no trabalho de (Alizadeh, 2011). No entanto, na proposição inicial, uma alteração do piso dinâmico foi adotada. Inicialmente os conceitos de Busca Tabu (Glover, 1989) e colônias de formigas (ACO), proposto em (Dorigo et al., 1996) foram apresentados neste trabalho, pois são parte do motor de busca das tarefas desenvolvidas. Essa alteração permite que o arqueamento investigado em (Alizadeh, 2011), (Wei-Guo et al., 2006) e (Perez et al., 2002) seja mais suave para o processo de *homing*. A resolução de conflitos adotada neste trabalho é não-determinística e considera que cada robô deve ocupar uma única célula do campo e não sobrepor os obstáculos, essa resolução foi inspirada do trabalho (Varas et al., 2007). No entanto, neste trabalho não foi adotado o parâmetro de pânico, uma estratégia previamente adotada em (Castro and Lima, 2013a). Posteriormente, o trabalho de (Zheng et al., 2011) também foi levado em consideração na tarefa de busca e resgate, pois o campo de piso de fogo adotado neste trabalho sugeriu uma modelagem do campo de fogo baseada em autômatos celulares (Lima and Lima, 2014). Adicionalmente, o trabalho de (Guo et al., 2016) foi utilizado quando se trata do cálculo de densidades perto dos ninhos. A inspiração de se utilizar dinâmica de pedestres e robótica surgiu inicialmente no trabalho

de (Nishinari et al., 2006). O trabalho de (Fu et al., 2015) serviu de inspiração para a criação da tarefa de coleta seletiva, já que utiliza de times heterogêneos de pedestres. Em relação aos fenômenos típicos observados, o efeito de arqueamento (Perez et al., 2002), efeito de repulsão (Wei-Guo et al., 2006) e o efeito de fogo (Zheng et al., 2011), foram analisados.

---

# Robótica de enxames

---

Robótica é um ramo educacional e tecnológico que compreende processadores, robôs e computação. Ela trata de sistemas compostos por partes mecânicas automáticas e controladas por circuitos integrados, tornando-os sistemas mecânicos motorizados, controlados manualmente ou automaticamente por circuitos elétricos. O termo Robô foi pela primeira vez usado pelo Checo Karel Capek (1890-1938) numa Peça de Teatro: RUR - *Rossum's Universal Robots* (Čapek, 2004), cujo livro foi lançado no Brasil pela editora Hedra com o título “A Fábrica de Robôs”. Inicialmente Capek estava decidido a chamar as criaturas automatizadas da sua peça de labori, em clara referência ao latim labor, trabalho, mas acabou os chamando de roboti (plural), por sugestão de seu irmão. A palavra robô, derivada deste robot (singular) e roboti (plural) tem como raiz a palavra Checa robot, a qual significa “trabalho forçado, servidão” e tem como uma de suas derivações a palavra rabu, que significa “escravo” (Russell and Norvig, 2005). O autor Isaac Asimov, muito conhecido por seus romances de ficção científica e também por suas dezenas de livros publicados, popularizou o uso do termo robótica na sua ficção “*I, Robot*” (Eu, Robô), de 1950 (Asimov and de Naveira, 1987). A ideia de se construir robôs começou a tomar força no início do século XX com a necessidade de aumentar a produtividade e melhorar a qualidade dos produtos. É nesta época que o robô industrial encontrou suas primeiras aplicações, o pai da robótica industrial foi George Devol, devido ao grande número de patentes registradas na área (Devol and Martin, 1975). Devido aos inúmeros recursos que os sistemas de microcomputadores nos oferece, a robótica atravessa uma época de contínuo crescimento que permitirá, em um curto espaço de tempo, o desenvolvimento de robôs inteligentes fazendo assim a ficção do homem antigo se tornar a realidade do homem atual (Dowling,

1999).

Qualquer sistema é caracterizado por variáveis de entrada e saída, existindo um relacionamento de causa e efeito entre elas, de modo a formar um todo organizado. Controle é o processo de forçar uma variável de saída de um sistema para estar em conformidade com um valor desejado, denominado valor de referência. O controle pode ser realizado nas formas manual, semi-automática e automática (Somló et al., 1997). O sistema de controle de robôs é geralmente muito complexo (Zieliński and Winiarski, 2010), sendo que as técnicas tradicionais incluem uma série de equações matemáticas para a descrição do modelo. Programar um sistema de controle para um robô, geralmente, exige conhecimentos não disponíveis durante o processo de desenvolvimento, principalmente se o ambiente no qual o robô irá atuar contenha ruídos, não seja estruturado, ou seja desconhecido (Arkin, 1998).

Neste contexto, pode-se definir um modelo de ambiente mais fácil de trabalhar, ou seja, um modelo onde o robô, o mundo e suas interações são feitas de forma mais simples. Existem diversas abordagens para a resolução de problemas na área de robótica. Se o modelo de mundo for bem definido, a técnica de linearização pode ser usada degradando o desempenho. Por outro lado, a aprendizagem por reforço pode ser utilizada, ao visitar estados corretos e a definição dessa função de recompensa pode ser difícil. A aplicação de técnicas bio-inspiradas tem sido uma das vertentes mais recentes em robótica (Zieliński and Winiarski, 2010). Essas técnicas baseiam-se na inspiração de sociedades biológicas formadas para propósitos gerais. Muitas das vezes o time de robôs operam de forma coletiva para o cumprimento de um determinado objetivo e por isso a associação entre este tema e a robótica torna-se interessante. Pelo fato dos robôs se comportarem como unidades processadoras simples, similar aos indivíduos de grandes populações, tais como, formigas abelhas e peixes, sistemas bio-inspirados tornam adequados a estes cenários (Calvo et al., 2011a).

## 4.1 Robótica de enxame e cooperativa

Robótica de enxame (*swarm robotics*) representa uma nova abordagem para a coordenação de um grande número de robôs, cuja principal inspiração provém da observação dos insetos sociais, tais como, formigas, vespas e abelhas (Calvo et al., 2011a). Enxames têm a característica de coordenar os comportamentos para realizar tarefas que estão além das capacidades de um único indivíduo. Por exemplo, formigas podem transportar grandes presas para o seu ninho (Gordon, 2014), enquanto cupins podem construir grandes montes de lama dentro do qual um nível desejado de temperatura e umidade é mantida. O surgimento de tal comportamento sincronizado no nível do sistema é bastante atrativo para os pesquisadores que trabalham com sistemas multi-robô, uma vez que emergem comportamentos globais, apesar dos indivíduos do sistema serem relativamente incapazes

zes, sem uma coordenação centralizada e através da simplicidade de interações (Ducatelle et al., 2010).

O termo inteligência de enxame foi originalmente concebido em (Beni and Wang, 1993) para designar uma classe de sistemas robóticos celulares. Posteriormente, o termo mudou-se para cobrir uma ampla gama de estudos de otimização para o desenvolvimento de estudos de insetos sociais, perdendo seu contexto em robótica no mesmo período. Recentemente, o termo robótica do enxame começou a ser usado como a aplicação de inteligência de enxame para sistemas físicos. Em (Beni, 2005), o autor aborda as perspectivas do termo enxame e faz uma revisão da inteligência de enxames, otimização de enxames e robótica de enxames e justifica quando o termo enxames deve ser utilizado. O autor acredita que é muito difícil colocar diversos robôs juntos para fazerem algo realmente significativo, assim, é por isso que o termo robótica coletiva deve ser utilizado.

A robótica cooperativa é o campo dedicado ao estudo de técnicas que permitem que robôs agrupados em equipes cooperem entre si e com seres humanos para realizar uma dada tarefa. Conforme (Marcolino and Chaimowicz, 2008), ela é capaz de aumentar a robustez e eficiência da execução de uma tarefa. O uso de times de robôs apresenta várias vantagens sobre abordagens baseadas em um único robô. Primeiramente, dependendo do tipo de tarefa, múltiplos robôs podem executá-la em menor tempo mediante uma divisão da tarefa. Além disso, grupos de robôs simples e com custo unitário mais baixo que trabalham de maneira cooperativa (Ducatelle et al., 2010) podem substituir um robô especializado cujo custo unitário é mais elevado. Adicionalmente, é possível aumentar a robustez em determinadas tarefas pelo uso de robôs que apresentam capacidade de redundância e são reconfiguráveis dinamicamente em caso de falhas (Bosque et al., 2009).

De acordo com (Lerman et al., 2005) os modelos de robótica cooperativa podem ser divididos em duas classes: microscópica e macroscópica. As descrições microscópicas tratam o robô com uma unidade fundamental do modelo. Esses modelos descrevem as interações entre os robôs e entre eles e o ambiente. Simular um sistema composto de muitos agentes dão aos pesquisadores um entendimento do comportamento global de um sistema. Exemplos desses modelos microscópicos podem ser vistos em trabalhos que tem como principal tarefa o levantamento coletivo de objetos e agregação de objetos (Lerman et al., 2005). Além de computar a trajetória exata e a informação sensorial de robôs individuais, as interações com outros robôs e com o ambiente são modelados como uma série de eventos estocásticos, com probabilidades determinadas por simples considerações geométricas e experimentos sistemáticos com um ou dois robôs reais. Simular vários eventos estocásticos em paralelo ao mesmo tempo, um para cada robô, permite aos pesquisadores estudar o comportamento coletivo do enxame. Por outro lado, um modelo macroscópico descreve o comportamento coletivo do enxame de robôs diretamente (Calvo et al., 2011a). É eficiente porque usa poucas variáveis. Esses modelos têm sido aplicados em uma gama de problemas que envolvem enxames na física, química, biologia e ciências sociais. Nestas

aplicações o comportamento microscópico de um indivíduo é mais complexo. As descrições macroscópicas geralmente têm uma simples fórmula que é analiticamente tratável (Martinoli et al., 1999).

Na comunidade de robótica, a maioria das abordagens para planejar o movimento de grupos de robôs é dividida em dois tipos: centralizada e descentralizada (Marcolino and Chaimowicz, 2008). As abordagens centralizadas são aquelas que assumem a existência de uma entidade central, ou seja, é uma entidade capaz de controlar cada robô do grupo. Embora, em geral, seja possível garantir a execução da tarefa, este tipo de técnica não é escalável para grandes grupos de robôs por razão de limitações computacionais. Já nas abordagens descentralizadas, é possível garantir escalabilidade, uma vez que os robôs tomam suas decisões baseados em interações locais, seja com seu ambiente ou com outros robôs que cooperam para a realização da tarefa.

Conforme (Şahin and Winfield, 2008), a operação de um sistema de enxame robótico deve apresentar três propriedades funcionais que são observadas em enxames naturais e também são desejáveis naqueles sistemas robóticos: robustez, flexibilidade e escalabilidade.

O sistema de enxame robótico deve ser capaz de operar mesmo com perturbações do ambiente ou com o mau funcionamento de seus indivíduos. Alguns fatores podem ser observados na robustez das operações dos insetos sociais: enxames são sistemas redundantes (a perda de um indivíduo pode ser imediatamente compensada por outro); a coordenação é descentralizada (Calvo et al., 2011a) e, conseqüentemente, é improvável que o enxame interrompa sua operação com a destruição de alguns membros; os indivíduos que compõem o enxame são relativamente simples (capacidades limitadas de comunicação, sensoriamento e atuação), tornando-os menos propensos a falhas; por fim, o sensoriamento é distribuído, e por isso o sistema é robusto à perturbações locais no ambiente.

Os indivíduos de um enxame devem ser capazes de coordenarem seus comportamentos para realizar tarefas de natureza distinta. Por exemplo, os membros de uma colônia de formigas podem encontrar o caminho mais curto para uma fonte de comida ou transportar uma presa grande por meio da utilização de diferentes estratégias de coordenação (Şahin and Winfield, 2008).

O enxame deve ser capaz de operar para uma grande faixa de tamanhos de grupos e suportar um grande número de indivíduos sem que isso comprometa de forma considerável o desempenho. Ou seja, os mecanismos e estratégias de coordenação a serem desenvolvidas para sistemas de enxames de robôs devem garantir o funcionamento para diferentes tamanhos do grupo (Şahin and Winfield, 2008).

Além das características citadas anteriormente, a robótica de enxame se inspira no comportamento emergente observado em insetos sociais, o que se denomina inteligência de enxame. Nestas técnicas, estuda-se como um grande número de agentes relativamente simples pode ser coordenado de modo a possibilitar um comportamento coletivo universal



desejado a partir das interações locais entre os agentes e o ambiente (Bosque et al., 2009). Um componente decisivo é a comunicação entre os membros de um grupo que constrói um sistema de *feedback* constante. O comportamento de enxame envolve a mudança constante dos indivíduos em cooperação mútua, assim como o comportamento do grupo como um todo. Diferentemente dos sistemas robóticos distribuídos em geral, a robótica de enxame enfatiza um grande número de robôs, e promove a escalabilidade, por exemplo utilizando apenas a comunicação local. De acordo com (Craig, 1989), esta comunicação local pode ser feita através de sistemas de rede sem fio, como frequências de ondas de rádio ou infravermelho. Novos estudos são necessários para se encontrar metodologias que permitam o design e a previsão confiável do comportamento de enxame quando apenas as características dos membros individuais forem conhecidas.

Tanto a miniaturização quanto o custo são fatores determinantes na robótica de enxames. Estes são os gargalos na produção de grandes grupos de robôs, dessa forma, a simplicidade dos indivíduos deve ser enfatizada. Isto poderia motivar uma abordagem que busca um comportamento ordenado a nível coletivo, ao invés de se ater ao individual. Segundo (Liu et al., 2007) muitas aplicações de mundo real podem beneficiar-se do uso de sistemas multirrobóticos. Porém devido a complexidade dos sistemas multirrobóticos em conjunto com a necessidade de evolução da tecnologia atual, poucos desses sistemas foram efetivamente implementados em aplicações do mundo real.

## 4.2 Domínios de aplicação na robótica

Vários domínios de aplicação podem ser investigados na área de enxames de robôs. A manipulação e transporte cooperativo de objetos é um exemplo de aplicação de enxames de robôs (Fujisawa et al., 2013). Neste problema, considera-se que os robôs pertencentes ao grupo não são capazes de manipular ou transportar objetos individualmente, devendo fazer isto com base na coordenação de seus movimentos e controle de formação.

Outro exemplo usual de aplicação de enxames robóticos requer a utilização de múltiplos robôs para detectar e monitorar múltiplos alvos (Kubo et al., 2014). O objetivo é maximizar a quantidade de tempo em que os alvos são observados por pelo menos um dos robôs da equipe. Este domínio possui relação com atividades de segurança, vigilância e reconhecimento de problemas e também com redes de sensores.

As tarefas de exploração e pastoreio também são investigadas em enxames de robôs. Em tarefas de exploração, neste caso, o objetivo do enxame de robôs é coletar o máximo de informações úteis possíveis sobre uma determinada área (Masár, 2013). Em tarefas de pastoreio o objetivo é mover o grupo de robôs pelo ambiente mantendo uma formação específica. A grande questão sobre estes domínios é a utilização da cooperação entre os robôs para a localização e fusão das informações percebidas do ambiente. Uma outra aplicação bastante estudada é o futebol de robôs, oferece um ambiente dinâmico repleto

de incertezas para a pesquisa e desenvolvimento de sistemas multi-agente e cooperação multirrobótica (Bonabeau et al., 1999).

Neste trabalho, estaremos concentrados na tarefa de **fORAGEAMENTO**, sendo este um domínio de aplicação relevante na área de robótica, que consiste na tarefa de múltiplos robôs capturarem objetos espalhados em um determinado local. Associadas à tarefa de forrageamento temos diversas aplicações, tais como, a limpeza de resíduos tóxicos, a limpeza de minas e robôs de serviço, e a remoção de minas explosivas. Os principais desafios destes domínios são definir técnicas eficazes para analisar e cobrir todo o ambiente e evitar que haja interferência entre os robôs durante a execução destas tarefas (Winfield, 2009).

Além do forrageamento, outra tarefa que foi investigada e modelada neste trabalho foi a **COLETA SELETIVA** de objetos. Nesta tarefa os robôs homogêneos ou não podem coletar objetos e identificá-los, e eles podem agrupá-los em locais distintos fazendo uma limpeza na arena (Ferri et al., 2011). Para essa tarefa, podemos utilizar um time de robôs homogêneo ou heterogêneo, sendo que nesse último caso os robôs do time podem ser diferenciados de acordo com o objeto a ser coletado.

Por fim, outra tarefa investigada neste trabalho foi a **BUSCA E RESGATE**, onde os robôs dotados de comportamento autônomo e interações coletivas poderiam facilmente atingir locais de difícil acesso para bombeiros em ambientes complexos com obstáculos durante operações de salvamento, encontrando, dessa maneira, vítimas em operações de buscas (Murphy et al., 2008).

### 4.3 Abordagens para o controle de trajetórias de robôs

Existem diversas formas de fazer com que um robô realize sua trajetória e cumpra sua tarefa de maneira efetiva. Como um robô pode apresentar diversos graus de liberdade (Calvo et al., 2011a), uma abordagem adequada à arquitetura utilizada deve ser empregada, fazendo com que o time de robôs cumpra o seu objetivo.

Uma destas abordagens é denominada **MAPA DE ROTAS** (*roadmap*). Nessa abordagem, o planejamento de caminhos consiste em definir as posições inicial e final no mapa de rotas e buscar um caminho entre eles (Barraquand and Latombe, 1991). Uma característica dos métodos baseados no mapa de rotas é que geralmente eles têm a implementação simples, porém não fornecem uma boa representação do ambiente. Como exemplos de métodos desta abordagem, temos os grafos de visibilidade (Barraquand and Latombe, 1991), diagramas de Voronoi (Hwang and Ahuja, 1992), silhueta e redes de sub-meta (também denominadas redes de caminho livre) (Hwang and Ahuja, 1992).

A abordagem por **DECOMPOSIÇÃO DE CÉLULAS** consiste em dividir o espaço livre do robô em várias regiões, idênticas ou não, chamadas células e determinar entre estas células uma relação de adjacência. Uma destas células conterá o robô e outra conterá a meta. As

células podem ser definidas de acordo com as bordas dos obstáculos existentes no ambiente. Este caso chama-se decomposição por dependência do objeto. A união das células livres define exatamente o espaço livre (Hwang and Ahuja, 1992). Conseqüentemente, caso sejam usadas técnicas de busca e de computação numérica exata, este método é completo, ou seja, sempre encontra um caminho livre, caso exista (Barraquand and Latombe, 1991)]. Na decomposição independente dos objetos, o ambiente é dividido em células idênticas e cada célula pode conter ou não um obstáculo. Assim, as bordas dos obstáculos podem não estar perfeitamente dentro da célula, estes pequenos erros podem ser diminuídos diminuindo o tamanho das células. Esta abordagem para o planejamento de caminhos é a mais utilizada nos trabalhos baseados em autômatos celulares (ACs). Dentro desta abordagem, temos um método de planejamento de caminhos conhecido como frente de onda (*wavefront*) (Barraquand and Latombe, 1991; Ferreira et al., 2014). Neste método, após uma decomposição independente dos objetos, define-se uma célula meta e uma célula como inicial. Depois, adiciona-se a célula meta em uma lista de células que serão visitadas. Esta meta começará com valor de distância até a meta valendo 0. A título de comparação essa técnica é similar ao algoritmo para criação do campo de piso estático proposto em (Varas et al., 2007) para a dinâmica de pedestres, apresentado no capítulo anterior.

A abordagem por **campo potencial** consiste em construir uma função potencial que contém seu valor mínimo na meta e um valor máximo nos obstáculos. Essa técnica é similar à decomposição de células. No entanto, esta abordagem é mais utilizada na resolução de problemas que dependam de um ambiente mais contínuo. Em todos os outros locais do espaço, a função decresce em direção à meta, assim é possível encontrar um caminho partindo de qualquer ponto no espaço até a meta, escapando dos locais onde existem obstáculos, devido ao alto valor de sua função potencial naqueles pontos (Ferreira et al., 2014; Hwang and Ahuja, 1992). Esta denominação surgiu da metáfora que o robô é uma partícula que se movimenta sob a influência de um campo potencial produzido pela meta e os obstáculos do ambiente. Assim, a meta gera um campo potencial atrativo e os obstáculos geram um campo potencial repulsivo. Embora seja a mais empregada em modelagem contínua, uma versão discreta dessa abordagem foi empregada em modelos baseados em ACs (Ferreira et al., 2014).

A abordagem por **programação matemática** desenvolve um sistema de navegação que, através de um conjunto de inequações que representa o desvio de obstáculos, cria um modelo de otimização matemática que encontra uma curva entre as configurações inicial e final minimizando determinada quantidade. Uma vez que esta otimização é não linear e existem muitas restrições de desigualdade, um método numérico é utilizado para encontrar a solução ótima (Ferreira et al., 2014; Hwang and Ahuja, 1992).

## 4.4 Trabalhos anteriores em robótica de enxame

Diversas tarefas têm sido investigadas no escopo da robótica de enxame. Dentre elas, podemos destacar a vigília, a identificação de objetos, o forrageamento, a coleta de objetos, a busca pelo melhor caminho, a navegação com controle de formação, busca e resgate e o levantamento cooperativo de objetos. Nesta seção revisaremos alguns trabalhos nas tarefas de vigília, forrageamento, coleta seletiva de objetos e busca e resgate por estarem relacionadas às tarefas investigadas neste trabalho.

A vigília é uma tarefa bastante explorada, seja com um conjunto de câmeras fixas, seja com um time de robôs. Ela consiste no espalhamento do time de robôs para que se tenha a maior cobertura de espaços com a menor quantidade de robôs possível. Um exemplo de trabalho nesta área é o trabalho de (Basilico and Amigoni, 2009). Neste trabalho, os autores usaram uma estratégia mais teórica baseada no *Multi-Criteria Decision Making* (MCDM) para definir formas de exploração do ambiente pelos robôs. Os robôs adotados para a criação do modelo, embora não tenha sido realizado nenhum experimento com robôs reais, possuem diferentes controladores. O controlador central utilizado foi o AORJRF (controlador utilizado na disputa de torneios robóticos entre universidades) e o robô adotado foi o Pioneer 3AT (Ferreira et al., 2014). Para resolver o problema, os robôs percebem a área com o intuito de realizarem a integração de um mapa do ambiente e tomaram a decisão para onde devem ir no próximo instante de tempo. Ou seja, o conhecimento é fornecido através de mapas parciais construídos pelos próprios robôs e decisões de curto prazo são feitas através da quantidade de alternativas e da observação de candidatos nas fronteiras entre áreas já exploradas e não exploradas. Dessa maneira, o robô seleciona a melhor opção de caminho dentre as apresentadas (Ferreira et al., 2014). Os resultados fornecidos pelos autores foram realizados no simulador USARSim e foram feitas análises na comparação dos diversos controladores para robôs existentes para a solução do mesmo problema, tais como, AORJRF, WS e DIST. Como cenário dos testes, foram considerados times de 1 ou 2 robôs, colocados em um dos dois mapas apresentados: (a) mapa com corredores e muitas salas e (b) mapa com espaços mais abertos (Basilico and Amigoni, 2009). Foi avaliado a maior área coberta para cada configuração em relação ao número de robôs utilizados. Também foi avaliado a maior área coberta em relação à quantidade de minutos gastos (evolução ao longo do tempo) e como resultado foi gerado um mapa de áreas exploradas após a simulação (mapa de rastros dos robôs).

O trabalho de (Calvo et al., 2011a) também investiga o problema da vigília com robôs autônomos para ambientes conhecidos. Neste caso, a abordagem consiste em um sistema cooperativo e distribuído, baseado no algoritmo das formigas invertido, ou seja, o algoritmo utiliza o feromônio para repulsão dos agentes, diferentemente do algoritmo clássico de colônia de formigas (Dorigo et al., 2008) na qual o feromônio provoca convergência dos agentes para uma determinada área. Os autores trabalharam em diferentes abordagens

e denominou o modelo criado de sistema IAS-SS (*Inverse Ant System-Based Surveillance System*). Os robôs que foram considerados para a implementação do trabalho foi o Pioneer 2Dx equipados com um laser denominado SICK LMS 200. Para resolver o problema, os autores utilizaram um sistema de vigília baseado no algoritmo das formigas invertido. O sistema consiste em um grupo de agentes artificiais que se movem independentemente e tomam decisões baseadas nos estímulos recebidos do ambiente. O sistema é um exemplo de AS (*Ant System*), ou seja, enquanto os robôs navegam, eles depositam um feromônio no ambiente. A cada passo de tempo cada robô recebe um estímulo e ajusta sua direção de navegação, sendo esta a única ação do robô. Cada robô tem também autonomia para desviar dos obstáculos. Duas estratégias são usadas para guiar a direção dos robôs (Calvo et al., 2011a). A primeira considera todos os estímulos de feromônio coletados que o sensor detecta no limite de sua fila (*Stochastic Sampling - SS*). A segunda determina o ajuste da direção do ângulo baseado somente nos estímulos associados com menos quantidade de feromônio (*Best Ranked Stochastic Sampling - BRSS*). Os sistemas baseados em ACO depositam feromônio somente onde os agentes estão localizados. No entanto, no trabalho foi considerada uma área maior para o espalhamento dessa substância: (a) *Forward Pheromone - FP* é considerado depósito num ângulo de  $180^\circ$  à frente do agente; (b) *Backward Pheromone - BP* é considerado um depósito de feromônio na iteração anterior num ângulo de  $180^\circ$ ; e (c) *Arround Pheromone - AP* é considerado um ângulo de  $360^\circ$  em relação ao agente para o depósito de feromônio. O feromônio tem uma taxa de evaporação com o passar do tempo e o sistema IAS-SS também contempla essa característica. Para isso, o sistema utiliza uma função para a taxa de decaimento do feromônio em relação ao tempo de iterações decorrido. Os autores utilizaram o simulador *Player/Stage* e o cenário de simulação considera um ambiente de  $20 \times 30$  metros, com sete salas e três robôs (Calvo et al., 2011a). Cada um dos três tipos de depósito de feromônio foi avaliado e combinado com os dois tipos de decisão que podem ser realizadas pelo robô. As taxas de evaporação de feromônio também foram avaliadas e foi mostrado através de gráficos o tempo que cada sala foi vigiada por cada robô e qual o robô foi responsável pela patrulha.

A tarefa de forrageamento foi amplamente investigada em diversos trabalhos de robótica coletiva. Essa tarefa consiste em fazer com que um time de robôs autônomos (agentes artificiais ou naturais) colem objetos (comida) distribuídos pelo ambiente e os robôs devem: localizar, coletar e depositar estes objetos em uma área específica denominada ninho (*home*) ou simplesmente agrupá-los em clusters. Um dos primeiros trabalhos que investigou esta tarefa foi o trabalho de (Calvo et al., 2011a), no qual robôs organizam a comida espalhada pelo ambiente em *clusters*. Neste caso, o autor utiliza de 1 a 5 robôs para a realização dos experimentos e o objetivo dos robôs é formar o menor número de *clusters* possível. O comportamento de cada robô, que é autônomo e independente, é regido por um AF (autômato finito) simples para investigar a capacidade de realizar a tarefa através da estigmergia. O conceito de estigmergia, mecanismo emergente de coordenação indireta,

presente nos insetos de comportamento social, como vespas e abelhas. O comportamento do time de robôs pode ser modelado através das seguintes regras (Calvo et al., 2011a).

1. O robô fica no estado “buscar” até encontrar um obstáculo ou encontrar um objeto (*puck*).
2. Se o robô encontrar um obstáculo, desvia aleatoriamente e volta para o passo 1.
3. Se o robô encontrar um objeto e se ainda puder pegar com a pinça (que cabe 3 objetos), então o robô pega o objeto e volta para o passo 1 novamente.
4. Se a pinça estiver cheia, então o robô deposita o objeto perto do objeto que acabou de encontrar.

Cada robô utilizado para os experimentos possui uma pinça para a captura de objetos (até 3 objetos), um sensor IR (*infra-red sensor* ou sensor infra-vermelho) para evitar obstáculos, e um microinterruptor que indica a quantidade de objetos na pinça. Ou seja, são sensores para a detecção de densidade com um mecanismo de limiar (ver Figura 4.1 (a)). Robôs reais foram utilizados nos experimentos e o tempo que o time de robôs levou para organizar todos os objetos em um único cluster em contraste com a quantidade de robôs utilizados (Calvo et al., 2011a). Também foi investigado o tamanho do maior cluster foi analisado formado comparado com a quantidade de robôs gastos e a eficiência, medida em unidades de tempo e a quantidade de robôs gastos para formar o maior cluster. Foi constatado que o tamanho do enxame é um fator crítico. O tempo médio para realizar as tarefas diminui para um, dois e três robôs, respectivamente, depois aumenta novamente para grupos de quatro e cinco agentes, devido a um aumento exponencial do número de interações entre robôs que são lentos e pode eventualmente resultar na destruição dos aglomerados existentes. Os resultados foram comparados com os relatados por (Deneubourg et al., 1990).

Outro trabalho que utiliza da estratégia de robótica coletiva na tarefa de forrageamento foi investigado em (Martinoli et al., 1999). Nesse trabalho, foi investigado um modelo bio-inspirado que utiliza da interação entre robôs e entre eles e seu ambiente. Os robôs devem coletar objetos com o intuito de formar clusters. Foi mostrado que os robôs são capazes de se comunicar uns com os outros e planejar suas tarefas continuamente. Para os experimentos foi utilizado o simulador Webots e o robô Khepera (ver Figura 4.1 (b)) que mostra que o modelo é válido para várias organizações, e foi contrastado com o modelo de (Beckers et al., 1994). Ao invés de simular os robôs movendo-se em um ambiente, o modelo representa uma atividade de agrupamento como uma sequência de eventos probabilísticos ocorrendo durante a variação do tamanho do cluster. Foi mostrado que, para qualquer plataforma, a evolução do tamanho do cluster é perfeitamente acompanhada, tanto quantitativamente quanto qualitativamente.

Outro trabalho que investiga o forrageamento foi proposto em (Sugawara and Sano, 1997). Diferentemente dos trabalhos anteriores, os robôs além de colecionarem os ob-

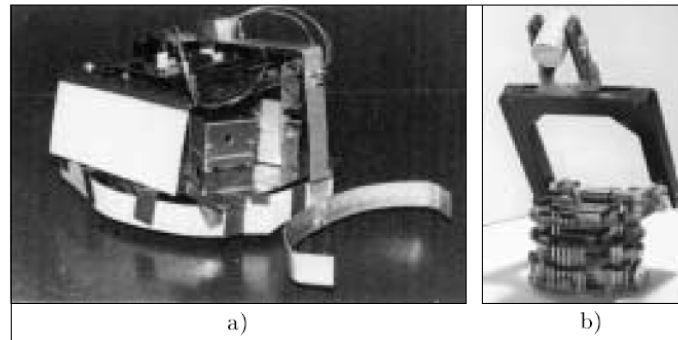


Figura 4.1: (a) Robô equipado com uma pinça para segurar objetos. Os experimentos foram realizados variando-se com 1 a 5 robôs do mesmo tipo (Beckers et al., 1994). (b) Robô Khepera equipado com uma pinça.

jetos, os robôs têm que colocá-los em um ninho previamente estabelecido. Os autores investigaram o comportamento coletivo de uma maneira mais quantitativa que os trabalhos anteriores. Cada robô utilizado nos experimentos, possuem dimensão de  $15 \times 9 \times 6$  cm, possui dois braços fixos, sensores infra-vermelhos (IR) e *photo* que permitem desviar de paredes ou outros robôs. O robô tem uma luz que permite avisar outros robôs. Os sensores IR são utilizados para levar o robô ao centro do campo, enquanto que o sensor *photo* é capaz de avisar sobre a presença de outros robôs. O comportamento de cada robô dentro do sistema é modelado por um AF que é demonstrado a seguir:

1. O robô fica no estado “buscar” até encontrar um obstáculo ou encontrar um objeto (*puck*).
2. Se o robô encontrar um *puck*, então o robô avisa aos outros robôs informando sua localização através de uma luz. Esse tempo pode variar de acordo com o experimento (1 a 30 segundos).
3. Os robôs sem *puck* se dirigem ao local do robô com *puck*.
4. O robô com *puck* deposita o objeto encontrado no ninho (*home*) e volta ao passo 1.

Os autores apresentaram resultados com simulação e analisaram se a mudança dos objetos (homogênea, ocupando entre 1% e 25% do ambiente) afeta no comportamento coletivo dos robôs. Também foram observadas as trajetórias de cada robô e a alteração do tempo de interação que foi modificado para cada experimento (1 a 30 segundos). Os autores também apresentaram os resultados com robôs reais e neste caso foi analisado a variação do tempo de interação e o quanto afeta o tempo para a realização da tarefa, a mudança da disposição dos objetos e a variação na quantidade de robôs.

Em (Liu et al., 2007) foi investigado o problema do forrageamento onde os robôs devem buscar por objetos espalhados pelo ambiente e depositar em um ninho que possui iluminação. O artigo investiga a efetividade do número de estratégias baseado em diferentes combinações de trocas de mensagens e demonstra a divisão do trabalho adaptativa e emergente com sucesso. Cada robô utilizado no trabalho tem dimensão de  $26 \times 26 \times$

26 cm (denominados *linux bots*), possui 3 sensores de intensidade de luz, 3 sensores IR de proximidade, 1 câmera, 1 sensor de cores e uma pinça. O robô vê a comida com a câmera, segura os objetos com a pinça e consegue perceber a casa (ninho) através dos sensores de luz. Ele sabe onde está localizado através dos sensores de cor. Cada robô é controlado por um AF de nove estados que está representado na Figura 4.2 e a mudança de estado é controlada pelo tempo ou por verificações entre os tempos. Os autores realizaram experimentos no simulador *Player/Stage* e quatro grupos de robôs foram montados para os experimentos combinando uma ou mais das seguintes características: dicas internas (verifica se o resgate de comida foi realizado com sucesso), dicas de ambientes (colisões com outros robôs enquanto procura por comida), e dicas sociais (sucesso do time na busca por comida). Assim, os autores avaliaram a energia do enxame ( $10^4$  unidades) pelo tempo em segundos e constatou-se que foi praticamente linear para todos os grupos. Foi analisado também o número de alimentos existentes versus o número de alimentos coletados por cada grupo de experimentos. Outra análise foi em relação à quantidade de energia obtida por cada grupo (quando todas as características analisadas foram usadas simultaneamente o ganho é maior). Também foi analisado a alteração de probabilidades e a relação com o equilíbrio da quantidade de robôs usados no experimento e a alteração de estágios de probabilidade e avaliação e adaptação de cada grupo de robô na quantidade de energia do enxame. Os autores (Liu et al., 2007) não apresentaram experimentos com robôs reais.

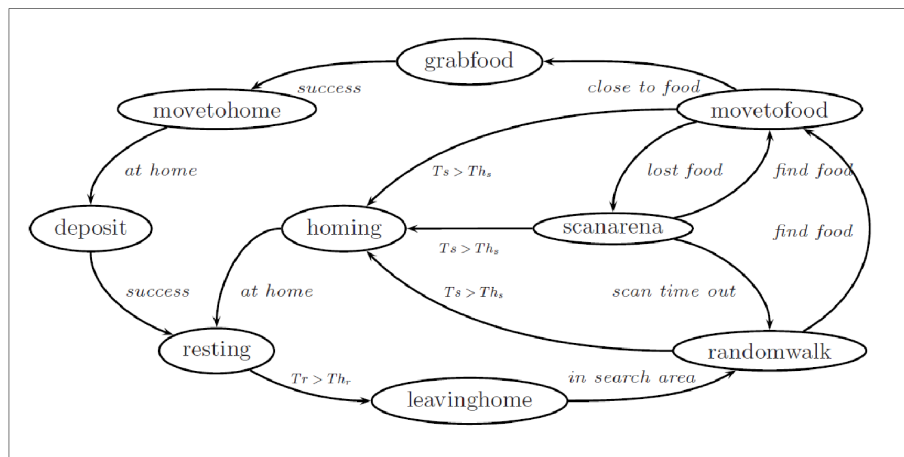


Figura 4.2: Comportamento do robô no modelo de (Liu et al., 2007).

O trabalho de (Lerman et al., 2005) revisa alguns modelos usados para a modelagem macroscópica no forrageamento e analisa o comportamento coletivo de sistemas de robótica coletiva para o cumprimento da tarefa de coletar objetos e para o levantamento de bastões. Os autores apresentaram que embora o comportamento de um robô em um sistema coletivo seja caracterizado por um componente estocástico, o comportamento coletivo é estatisticamente previsível e tem pouca probabilidade associada (Lerman et al., 2005). Além disso, foi mostrado que a classe dos modelos matemáticos que descrevem o



comportamento coletivo pode ser gerado usando os controles individuais de robôs. Foi mostrado que a modelagem macroscópica pode ajudar em resultados de coleta na distribuição dos experimentos, tais como, levantamento colaborativo de bastão, forrageamento e agregação. Os modelos foram comparados com métodos de probabilidade e também em simulações baseadas em sensores com robôs reais. Dependendo da abordagem, da métrica usada e da complexidade do modelo, foi mostrado que é possível alcançar quantitativamente previsões corretas. De acordo com os autores, o comportamento de robôs individuais em enxames tem várias influências complexas. Robôs são influenciados por forças externas, que muitas das vezes não podem ser antecipadas, como fricção, força da bateria, sons ou iluminação. Mesmo com todas essas forças conhecidas, os robôs continuam sendo submetidos a alguns eventos aleatórios. Outro aspecto observado foi o desvio de obstáculos. A Figura 4.3 demonstra um controlador para um cenário de forrageamento simplificado. Cada caixa representa o estado de um robô, ou seja, a ação que está sendo executada. Para cumprir a tarefa de pegar e levar o objeto para casa o robô vai transitar pelos três estados. As transições entre os estados são disparadas por estímulos externos (Lerman et al., 2005). Assim, o robô apresentado pode ser descrito por um processo Markoviano, ou seja, por uma máquina de estados finitos (MEF).

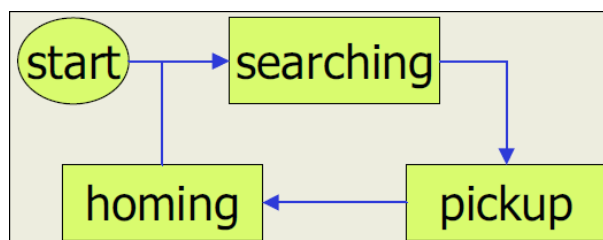


Figura 4.3: Diagrama de um controlador do robô para o cenário de forrageamento simplificado (Lerman et al., 2005).

Posteriormente em (Winfield, 2009) o autor descreve um modelo similar para robótica de enxames para cumprir a tarefa de forrageamento. O autor também parte da ideia de um sistema modelado por AF com quatro estados: procurando (uso de sensores para a procura de objetos), segurando (captura de objetos), retornando à casa (movimentação com o objeto coletado), e depositando (deposita o objeto próximo à casa), ver Figura 4.4. Além disso, o autor (Winfield, 2009) classifica os tipos de indivíduos do sistema que podem ser: caçador, recuperador ou defensor. Cada um desses tipos taxonômicos desempenham um papel diferente no sistema. O autor apresenta um cenário com um único robô para executar a tarefa de forrageamento, sendo que este pode ser tratado como parte de um time, e requer vários sensores e executa algumas ações em determinados estados. Adicionalmente, o autor (Winfield, 2009) descreve um sistema com múltiplos robôs que realizam a tarefa de forrageamento de forma coletiva. O autor cita duas abordagens para este tipo de tarefa, a primeira é a estratégia sem cooperação e a segunda cooperativa. Na primeira, os robôs não se comunicam, apesar disso, eles possuem forças repulsivas entre

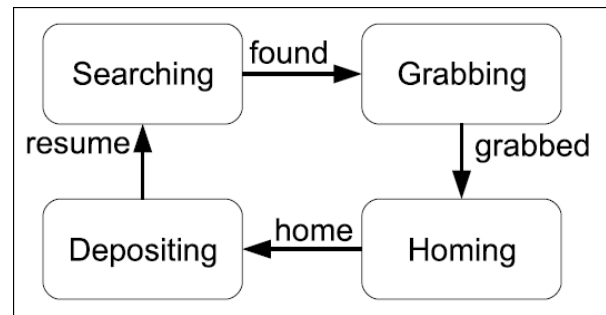


Figura 4.4: Diagrama de um controlador do robô para o cenário de forrageamento simplificado (Winfield, 2009).

si, essa característica desencadeia um espalhamento dos robôs e possui maior eficiência no ambiente de busca. Na segunda estratégia, foi mostrado que não há uma entidade central que controla os robôs. Conforme mostra a Figura 4.5, os quatro estados do modelo simples anterior foi estendido para oito estados a fim de prover a comunicação entre os indivíduos. Se um robô achar um bastão, ele irá comunicar a todos os outros. Um robô ao receber a mensagem entra no estado de “rastreamento” e depois ao estado “procurando” (Winfield, 2009). O autor também apresenta uma variação do modelo onde há divisão do trabalho e é importante quando a tarefa precisa de dois ou mais indivíduos para ser realizada com sucesso como é o caso do levantamento de bastões.

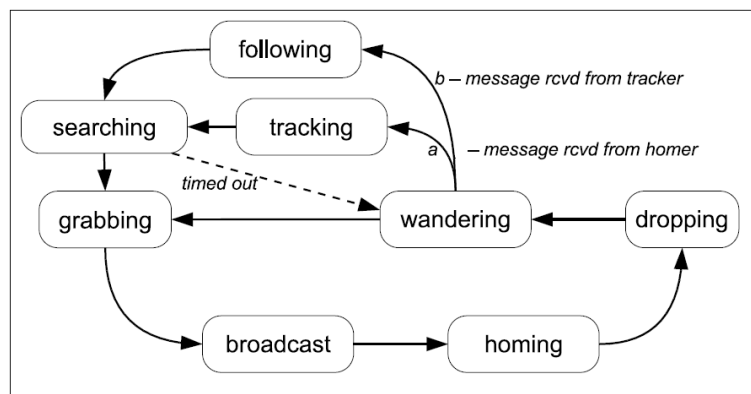


Figura 4.5: Máquina de estados para o forrageamento com comunicação (Winfield, 2009).

Outro trabalho que investiga o forrageamento foi proposto em (Hamann and Wörn, 2007), no qual os robôs devem procurar por comida e levá-la ao ninho. Todos os robôs que compõem o sistema são reativos, ou seja, repelem objetos tais como, outros robôs ou obstáculos; e se sentem atraídos por comida. Os autores apresentaram um modelo macrocópico baseado em equações diferenciais para o cenário do forrageamento e feromônios virtuais são usados como forma de comunicação indireta entre os agentes. Cada robô estudado pelos autores possui uma visão circunferencial, ou seja, pode medir distâncias dos objetos com os seus sensores e pode distinguir de outros robôs o ninho, a comida e a parede (obstáculo) que circula a área de simulação. O sistema contempla robôs que podem estar em um de dois estados: buscando comida ou carregando para casa (Winfield, 2009). Inicialmente todos os robôs são posicionados perto de casa com uma velocidade

aleatória posicionando para direções aleatórias. Se o robô está perto de casa e percebe a comida ele vai para o estado de busca por comida. Os robôs que estão buscando comida vão para perto de casa se eles percebem o ninho. Dois tipos de feromônios são lançados no ambiente, um para quando o robô está perto da comida, e outro para quando o robô está perto do ninho (em uma taxa constante). O depósito do feromônio é discreto e é regido pelas equações diferenciais. Os autores apresentaram resultados utilizando-se o simulador *Breve*. Os testes contemplam 30 execuções com 10 robôs cada e verificou-se qual é a melhor quantidade de feromônio a ser depositado. Também foi analisado o fluxo de comida pelo tempo e se os robôs possuem comportamento estável ou não estável (fluxo  $\times$  tempo). Nenhum experimento com robôs reais foi apresentado (Winfield, 2009).

No trabalho de (Vargas et al., 2012) a tarefa de coleta seletiva de lixo aplicada à reciclagem foi apresentada através da robótica de enxames utilizando-se PSO (Particle Swarm Optimization). De acordo com os autores, a técnica de utilização do PSO foi aplicada com sucesso nesta abordagem. A ideia básica do modelo foi treinar uma quantidade de robôs para interagir com os demais de tal forma que o lixo pudesse ser recolhido, similar ao que acontece com comportamentos de grupos de animais, atuando de forma a ser uma entidade cognitiva única. Os experimentos foram realizados utilizando-se o simulador Simbad (Ferreira et al., 2014). Os ambientes de simulação consideraram dois cenários principais, um que considera que os robôs deveriam coletar as unidades de lixo e depositá-las numa estação de lixo sem preocupação com a energia e outro onde estes robôs perdem energia com o passar do tempo e para isso recarregam suas energias numa estação de recarga, além de coletar o lixo. Para cada um desses cenários foram avaliadas as topologias do PSO e as principais diferenças que as mesmas provocam no desempenho do time de robôs. De acordo com (Vargas et al., 2012) os robôs atingiram um comportamento similar ao que acontece com os insetos, cooperando uns com os outros de maneira eficiente e precisa. Duas topologias de PSO foram implementadas para serem contrastadas no trabalho, apresentando diferenças de avaliação e exploração das duas topologias escolhidas.

## 4.5 Trabalhos de robótica baseados em ACs

Em diversas situações em robótica é necessário o uso de algoritmos em tempo real para a determinação de possíveis trajetórias para os robôs. Um estudo bastante promissor é o uso de modelos de ACs para prover uma dinâmica inteligente e coletiva aos robôs em ambientes complexos a fim de atingirem um determinado objetivo. Alguns dos principais estudos sobre ACs na robótica estarão detalhados nesta seção.

Uma das primeiras abordagens conhecidas para dinâmica de robôs móveis baseada em AC foi proposta por (Shu and Buxton, 1995). Assim como nos modelos de dinâmica coletiva, o autor abordou a técnica de discretização do campo de piso bidimensional, onde as células no estado “0” representam as células livres, enquanto que as células no estado

“1” representam as células ocupadas. Para que o movimento seja consolidado, o autor considera as forças vindas de outras direções até o raio 2 da vizinhança do AC (Ferreira et al., 2014).

Outra abordagem relevante foi proposta por (Marchese, 1996). Neste trabalho, também existe o conceito de campo de piso com múltiplas camadas e as células podem estar nos seguintes estados: obstáculos, posição inicial do robô (indivíduo), posição objetivo, atração para o objetivo (distância até o objetivo) e o caminho até o objetivo. Inicialmente, os obstáculos são aumentados em  $r$  (raio) células para evitar a colisão dos robôs (Ferreira et al., 2014). Posteriormente, as direções dos estados iniciais e finais são indicadas na matriz de atualização. Em seguida, explora-se o ambiente de tal maneira que seja possível construir um caminho da célula inicial até o objetivo. Por fim, a partir de regras não determinísticas e probabilísticas o robô percorre um dos caminhos possíveis até chegar ao objetivo.

No trabalho de (Behring et al., 2001), um dos mais citados da literatura, é realizado um cálculo entre qualquer célula até o objetivo. Neste trabalho, é utilizado o método de crescimento das bordas para evitar a colisão dos robôs. Os obstáculos foram aumentados iterativamente por quatro passos de tempo de acordo com a regra de transição: “se a célula está no estado livre e algum dos seus vizinhos for um obstáculo, a célula passa para o estado obstáculo, caso contrário, o estado do instante de tempo anterior é mantido”. Posteriormente, o campo de piso definido no passo de tempo anterior é utilizado para a movimentação do robô. Além disso, o sistema calcula a distância entre o objetivo e cada célula livre do ambiente (Ferreira et al., 2014). A regra de transição é dada por: “Caso exista alguma célula livre e o vizinho já teve sua distância calculada, então o valor da célula é alterado para o valor do vizinho adicionado de 1; caso contrário, mantém o estado da célula”. Posteriormente, o robô percorre o trajeto saindo de sua posição inicial até o objetivo sempre diminuindo 1 na distância já percorrida. Em (Tavakoli et al., 2008) e (Soofiyani et al., 2010), o modelo precursor de (Behring et al., 2001) foi aperfeiçoado, sendo que o primeiro adaptou o modelo para múltiplos agentes enquanto que o segundo alterou o campo de piso, sendo que no seu trabalho, os agentes tendem a mover-se em linha reta ao invés de realizarem movimentos zigzagues (Ferreira et al., 2014).

No estudo em (Oliveira et al., 2015) um modelo baseado em ACs foi desenvolvido a partir do (Behring et al., 2001). Neste trabalho, dois problemas em Behring et al. (2001) foram apontados: (i) a posição final do robô distancia-se da meta em vários cenários avaliados e (ii) quando o robô tem um estado inicial dentro de um obstáculo aumentado virtualmente (Marchese, 2011), o mesmo é incapaz de encontrar o caminho até a meta. Para resolver o primeiro problema um novo modelo baseado na difusão da distância foi apresentado e foi composta de duas fases: a primeira é baseada no recálculo do caminho até a meta a cada passo de tempo. A segunda alteração refere-se à difusão da distância na área dos obstáculos alargados, na situação que o robô não encontra o caminho até a

meta (Ferreira et al., 2014). Para a solução do problema, duas medidas foram adotadas: a primeira delas irá medir a menor distância da célula que está em uma área de alargamento até uma célula livre. A segunda medida serviu para pedir o quão próximo cada célula das regiões alargadas está de um obstáculo real. Essas duas medidas são calculadas através de ACs (Ferreira et al., 2014). Um robô do tipo e-Puck foi usado para simular cenários para avaliar o novo modelo baseado em ACs para o planejamento de caminhos. As simulações mostraram resultados promissores no desempenho de um único robô, confirmando que o modelo pode ser adaptado para um time de robôs.

Um estudo que emprega regras de atualização local baseada em ACs para controlar dinâmica dos robôs até um objetivo foi desenvolvido em (Akbarimajd and Lucas, 2006). Nesta pesquisa, os sensores dos robôs são utilizados para avaliação da sua vizinhança. Assim, é possível decidir, por meio de uma regra de decisão, para qual estado o robô se movimentará no próximo instante de tempo. O robô sabe em que posição ele se encontra no reticulado de campo de piso e também onde se encontra a meta. A cada passo de tempo, o robô avalia sua vizinhança, neste caso a vizinhança de Moore, para se verificar em qual estado a mesma se encontra: livre, obstáculo e outro agente. O robô movimenta-se para uma de suas vizinhas deixando a célula central no estado livre. Posteriormente, este trabalho foi refinado em (Akbarimajd and Hassanzadeh, 2011) para ambientes com múltiplos robôs, sendo este um dos trabalhos mais similares com os modelos de dinâmica coletiva avaliados.

Nos trabalhos de (Ioannidis et al., 2008), (Ioannidis et al., 2011b), (Ioannidis et al., 2011a) foi apresentado um sistema multi-robôs baseado em ACs a partir de uma regra de atualização local para a tarefa de controle de formação, onde a tomada de decisão é realizada a cada passo de tempo. Para isso, os modelo usa a leitura dos sensores a cada passo de tempo para identificar a vizinhança de cada célula onde está o robô. A partir dessa leitura é aplicada uma regra de atualização local que decide qual será o próximo movimento do robô. Nessa investigação, o ambiente de movimentação é um espaço euclidiano bidimensional de células idênticas de tamanho  $l$ . O tamanho da célula não pode ser pequeno a ponto de tornar o processo de movimentação muito lento e com alta exigência de memória, e nem tão grande a ponto do robô não reconhecer um obstáculo próximo. Neste trabalho, a arquitetura adotada foi o robô e-Puck (EPFL, 2013). A cada passo de tempo, o robô poderá realizar dois tipos de movimentação: (i) deslocar-se para uma célula vizinha, mantendo sua orientação atual, ou (ii) realizar uma rotação sobre seu eixo, mantendo sua posição atual (Ferreira et al., 2014). Quando a ação é de deslocamento, a distância percorrida pelo robô será a distância entre a célula em que ele está e a célula vizinha, ou seja, um passo de deslocamento equivale a percorrer a largura  $l$  de uma célula, se o deslocamento é ao norte, sul, leste e oeste, e equivale a  $l\sqrt{2}$  se o deslocamento for para as células das vizinhanças diagonais. Por outro lado, quando a ação é uma rotação, o robô efetuará um giro com 4 valores possíveis  $-90^\circ$ ,  $-45^\circ$ ,  $45^\circ$ ,  $90^\circ$ . A vizinhança do

robô é construída a partir da leitura dos sensores do robô. Além disso, é necessário saber o ângulo de rotação dos robôs. A regra de atualização do AC define a movimentação do robô pelo ambiente. Essas regras estão divididas em dois tipos: regras de controle de formação e regras de desvio de obstáculos (Ferreira et al., 2014).

O estudo em (Ferreira et al., 2014) foi elaborado a partir do trabalho de (Ioannidis et al., 2011b), onde também é adotado um AC como regra de atualização. A primeira delas refere-se à correção do *deadlock* de quina do obstáculo e para isso foi utilizada a adição de um novo estado diferenciado para identificar que o robô já vinha de uma rotação para o desvio de obstáculo ao fazer uma nova rotação e voltar para a rotação original identificando novamente um obstáculo. Outra alteração neste modelo refere-se à inserção de estados para um deslocamento ainda maior na quina do obstáculo. Essa correção foi realizada através do aumento de estados que indicam a situação de robô previamente rotacionado, para que o robô dê mais passos para desviar de uma quina de obstáculo antes mesmo de retornar ao ângulo  $0^\circ$ . Por fim, a inserção de um novo estado nas regras contribuiu para a diminuição do efeito de ziguezague após o desvio de obstáculos, melhorando o tempo de processamento do sistema. Neste trabalho, foram avaliados a performance de cada uma dessas alterações em relação ao modelo precursor. No trabalho de (Silva, 2015) uma melhoria do modelo precursor foi avaliada a partir da utilização de algoritmos genéticos. Além disso, foi investigada uma topologia para a implementação do modelo precursor para um time de robôs, utilizando-se um esquema de comunicação entre os agentes do sistema para a troca de mensagens, que continham informações sobre as variáveis do sistema.

Um dos modelos de dinâmica coletiva baseados em ACs revisados no Capítulo 3, foi também executado com robôs (Nishinari et al., 2006) apenas para apresentar uma implementação do modelo de dinâmica de pedestres com a arquitetura de um robô. Este trabalho propôs a utilização de interação de múltiplos robôs a partir de interação por feromônio (líder-seguidor). Neste experimento, um sistema de feromônio virtual foi utilizado (*V-DEAR*) com os sinais projetados no chão e os robôs são capazes de decidir suas ações a partir das cores de gráficos projetados no chão. Foram analisados o tempo total de simulação a partir das diferentes constantes do sistema e a partir do depósito do feromônio. Além disso, um comportamento anômalo com o experimento de múltiplos robôs foi apresentado devido ao fato de que o modelo é dependente do depósito de feromônio (passos de robôs) e pequenas perturbações na posição inicial dos robôs podem evitar congestionamentos e permitir uma evacuação segura do time de robôs.

Mais recentemente, em (Boukas et al., 2015) um modelo robótico foi investigado motivado pelo problema da evacuação de pessoas do interior. Modelos onde se estuda o comportamento da evacuação de multidão podem ser em nível microscópico ou macroscópico e são capazes de fornecer os meios para acionamento sobre os evacuados por sinais visuais de som para evitar o entupimento em determinadas saídas. No trabalho de (Boukas et al., 2015) uma abordagem usando um autômato celular para atrair a atenção

de pessoas evacuadas que se dirigem para saídas não saturadas é proposto através do algoritmo de planejamento de caminhos. Assim, em pesquisas futuras, os autores propõem resolver o problema da coordenação de vários robôs para a evacuação de massas em grandes auditórios ou estádios. Diferentemente de nosso trabalho, onde propomos a coordenação de vários robôs não só baseada em modelagem de dinâmica coletiva de autômatos celulares, mas também baseada em outras técnicas bio-inspiradas e no uso de algoritmos que reforçam e melhoram a estratégia de busca.

Em (Tsiftsis et al., 2016) foi apresentado um modelo para controlar a evacuação de pessoas num estádio de futebol a partir de um desenvolvimento integrado de um circuito eletrônico que permite a gestão de congestionamento a partir das circunstâncias dinâmicas. Assim, foi proposto um modelo de ACs com memória de estado baseada em (Alonso-Sanz, 2009) para evitar os caminhos recentemente percorridos. O sistema incorpora uma câmera para o controle de densidade da multidão, similarmente ao que foi proposto para o cálculo de densidade próximo aos ninhos. Apesar de este não ser um trabalho diretamente ligado à robótica, o mesmo pode ser abstraído como um sistema de controle através de câmeras e processamento de imagens distribuídas ao longo do ambiente. Assim como o trabalho de (Ando et al., 2006) que apresenta um modelo para o controle de congestionamento através de feromônio invertido de vários tipos e controle de diferentes densidades ao longo do caminho. Similarmente em (Boukas et al., 2015) apresenta uma melhoria para a evacuação de pedestres, mas não apresenta uma ideia para a coordenação de um time de robôs assim como o que está sendo apresentado nesta tese de doutorado.

## 4.6 Visão geral dos trabalhos de robótica

A Tabela 4.1 apresenta uma visão geral de cada um dos modelos de robótica coletiva investigados para a proposição dos modelos aqui apresentados. Essa tabela tem por objetivo resumir e apresentar todos os trabalhos que foram investigados na literatura para a realização desta tese, inclusive àqueles que não se relacionam com as tarefas robóticas aqui investigadas. Os fatores analisados foram: (i) a tarefa e abordagem investigadas, (ii) o tipo de grupo usado, (iii) ambiente de simulação adotado, (iv) a arquitetura robótica investigada e por fim (v) e as principais análises a partir dos resultados apresentados.

## 4.7 Arquitetura investigada

Em robótica, faz-se necessário a adoção de um modelo de arquitetura de robô que executará os experimentos para a validação dos métodos propostos. Foram encontrados vários estudos na literatura na área de robótica coletiva (Pini et al., 2007), (Capi et al., 2008), (Greeff and Nolfi, 2010), (Ioannidis et al., 2011b) que utilizaram a arquitetura e-Puck (EPFL, 2013). Além disso, o laboratório no qual esta pesquisa foi desenvolvida,

Laboratório de Computação Bio-inspirada (LCBio), possui 6 robôs e-Puck. O robô foi desenvolvido pela École Polytechnique Fédérale de Lausanne (EPFL) com o propósito de servir como auxiliar em cursos de engenharia. Conceitos e detalhes de design podem ser vistos em (Mondada et al., 2009). Portanto, o e-Puck foi o escolhido para a validação dos experimentos realizados em simulação (Ferreira et al., 2014).

Os atuadores existentes no robô são as duas rodas que contêm o diâmetro de aproximadamente 41 milímetros e com a distância em seu entorno de 53 milímetros. A velocidade máxima atingida por cada roda do robô é de 1000 passos por segundo, que corresponde a 1 giro completo da roda por segundo. O robô contém 8 leds em torno do robô e um auto-falante para geração de som (Ferreira et al., 2014). O robô contém também 8 sensores de infra-vermelho para verificar a proximidade com obstáculos. Estes sensores não estão dispostos de uma forma linear, existindo mais sensores na frente do robô para evitar a colisão com obstáculos à frente. A Figura 4.6 mostra a disposição espacial dos sensores em

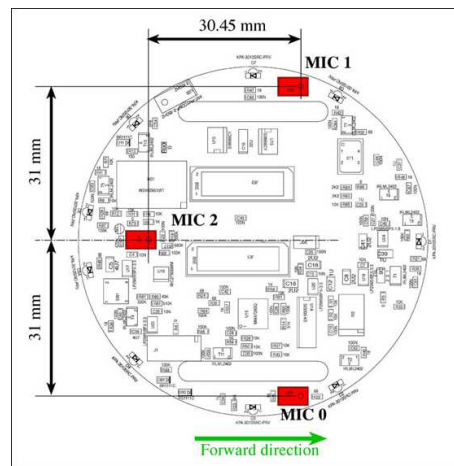


Figura 4.6: Disposição espacial dos sensores de proximidade em torno do robô e-Puck (EPFL, 2013).

torno do robô. A arquitetura do robô e-Puck contém também uma câmera com resolução de  $640 \times 480$  pixels, três acelerômetros, além de microfones para a captação de sons. O processador é um microcontrolador da família PIC. Contém uma bateria recarregável com autonomia por entre 2 e 3 horas. Está disponível a interface Bluetooth para comunicação entre os robôs (Ferreira et al., 2014).



Tabela 4.1: Principais trabalhos de robótica coletiva investigados.

Referência	Tarefa e Abordagem	Indivíduos/ Grupos	Software	Arquitetura Investigada	Resultados
(Beckers et al., 1994)	Forrageamento, Clusterização, AFD.	Homogêneo	Player/Stage	Khepera,	Formação de cluster Distribuição objetos.
(Shu et al., 1995)	Planejamento, ACs.	Heterogêneo	SIMD, DAP Fortran.	Small robot	Forças externas, Desvio de obstáculos
(Behring et al., 2001)	Planejamento, ACs.	Homogêneo	Robocup	Small Robot, QuickCam 2.	Efeito ziguezague
(Martinoli et al., 2004)	Forrageamento, AFD.	Homogêneo	Webots	Khepera	Agrupamento, Tamanho do cluster.
(Nourbakhsh et al., 2005)	Busca e resgate, I	Heterogêneo	USARsim	USARobot,	Feedback do time, Coordenação.
(Şahin, 2005)	Forrageamento ACO, PSO.	Homogêneo	Player/Stage	Khepera.	Coordenação do time, Estratégias
(Lerman et al., 2005)	Levantamento, Forrageamento, AFD.	Homogêneo	Player/Stage	Khepera,	Forças externas, Desvio de obstáculos, Interação.
(Krishnanand, 2005)	Planejamento, GLOW-WORM.	Heterogêneo	Matlab	Kinbots	Líder-seguidor, Luminescência, Espalhamento.
(Nishinari et al., 2006)	Dinâmica coletiva ACs.	Homogêneo	VDEAR	Small Robot,	Comportamento de seguir, Efeito de colisão.
(Kantor et al., 2006)	Formação, Busca e Resgate, REDE.	Heterogêneo	IEEE 802.11	GPS Robots Mote sensors	Modificação de trajetória, Diferença temperatura.
(Sugawara et al., 2007)	Forrageamento, AFD.	Homogêneo	-	DCMotor	Distribuição objetos, Tempo de comunicação.
(Liu et al., 2007)	Forrageamento, AFD.	Homogêneo	Player/Stage	Linux Bots,	Divisão trabalho, Energia do time, Equilíbrio time.
(Hamann et al., 2007)	Forrageamento, ACO.	Heterogêneo	BRave	Small Robot	Reação do time, Difusão Feromônio.
(Marjovi et al., 2009)	Identificação de objetos, A*.	Homogêneo	Player/Stage, C++	Khepera III,	Detecção de objetos, Detecção de fogo, Tempo em labirintos.
(Winfield, 2009)	Forrageamento, AFD.	Heterogêneo	-	Slugbot	Rastreamento, Espalhamento.
(Basilico et al., 2009)	Busca e resgate, MCDM.	Heterogêneo	USARSim	Pioneer P3AT	Área de cobertura, Mapas de rastros, Tempo de exploração.
(Ducatelle et al., 2010)	Forrageamento ACO.	Heterogêneo	Swarmanoid	Eyebots Footbots	Realização de trajeto, Congestionamento, Espalhamento.
(Akbarimajd, 2012)	Planejamento, ACs.	Homogêneo	Matlab	Small Robot	Comportamento coletivo, Comportamento individual.
(Ioannidis et al., 2011b)	Planejamento, ACs.	Homogêneo	Webots	e-Puck	Mestre-escravo, Desvio de obstáculos.
(Marchese, 2011)	Planejamento, ACs.	Homogêneo	-	SIMD machine	Obstáculo virtual, Colisão de robôs.
(Calvo et al., 2012)	Vigília e Exploração, ACO.	Homogêneo	Player/Stage	Pioneer 2Dx	Feromônio, Posição de depósito, Área coberta.
(Vargas et al., 2012)	Coleta seletiva PSO.	Homogêneo	Simbad, Java 3D	Eyebots Footbots	Energia do time, Topologia.
(Ferreira et al., 2014)	Formação, ACs.	Homogêneo	Webots	e-Puck	Líder-seguidor, Desvio de obstáculos. Efeito ziguezague.
(Oliveira et al., 2015)	Planejamento, ACs.	Homogêneo	Webots	e-Puck	Efeito ziguezague, Desvio de obstáculos.
(Silva, 2015)	Formação, ACs, AGs.	Homogêneo	Webots	e-Puck	Líder-seguidor, Desvio de obstáculos.
(Boukas et al., 2015)	Resgate ACs.	Homogêneo	VSLAM	Próprio Construído	Líder-seguidor, Resgate
(Tsiftsis et al., 2016)	Planejamento, ACs.	Homogêneo	Compilador VHDL	FPGA	Densidade

## 4.8 Ambientes de simulação

Os ambientes de simulação de robôs são de extrema importância para pesquisas em robótica (Tikhanoff et al., 2008). Com o auxílio deles, é possível criar modelos para teste de teorias, ideias ou eficiência de métodos, antes de embarcar a aplicação em um robô físico. Assim, a implementação de testes em ambientes de simulação antes dos testes com robôs reais é uma prática comum (Ferreira et al., 2014). Desta forma, é possível utilizar um ambiente de simulação desenvolvido especialmente para determinado robô e determinada aplicação ou utilizar algum simulador genérico. Porém, deve-se ter em mente que um modelo de simulação não provê toda a complexidade existente em um ambiente real, além de não garantirem que a aplicação seja embarcada diretamente no robô real através do ambiente de simulação (Tikhanoff et al., 2008). Contudo, é recomendável a utilização de um motor de física para simular a movimentação mais real dos robôs em simulação. Alguns simuladores genéricos utilizam a ODE (Open Dynamics Engine) (Smith, 2013), Bullet Engine (Coumans, 2013a) ou PhysX (Green, 2013). As duas primeiras são engines de código aberto testadas em dezenas de aplicações. A última é uma engine proprietária desenvolvida pela empresa Nvidia (Ferreira et al., 2014).

Para a escolha de um ambiente de simulação para a execução dos testes deste trabalho, foi definido que seria necessário que o simulador tivesse a capacidade de simular o robô e-Puck (EPFL, 2013), que foi a arquitetura escolhida para validar os modelos. Para isso, foi escolhido o simulador Webots EDU (Coumans, 2013b), que também já havia sido adquirido no Laboratório de Computação Bio-inspirada (LCBio) e no Grupo de Pesquisa Robótica Bio-Inspirada (Robo-Bio). Apesar de ser possível de implementação em seis linguagens, dentre elas, C, C++, Java, Python, Matlab e URBI, para este trabalho as implementações foram feitas em linguagem C, que facilita a embarcação em robôs reais sem a necessidade de adaptações. O Webots EDU utiliza o motor de dinâmica ODE que obteve resultados satisfatórios nos experimentos desenvolvidos neste trabalho.

## 4.9 Implementação da Movimentação do Robô

Para a movimentação do robô na plataforma Webots EDU seja por deslocamento ou por rotação, é necessário implementar a forma como as rodas irão se comportar dado um determinado comando. Em tarefas onde o robô tenha que se deslocar uma série de movimentos discretos são realizados pelo robô. É necessário que o robô execute um comando de rotação para uma determinada orientação ou que se desloque por uma distância determinada em uma translação. Assim, o robô saberá em qual posição e orientação ele está em determinado momento (Ferreira et al., 2014).

Neste trabalho, duas abordagens diferentes foram discutidas e implementadas nos experimentos usando o ambiente de simulação Webots EDU para o robô e-Puck, tanto para

a localização do robô, quanto para sua movimentação. A primeira, mais simples, foi utilizada nos experimentos utilizando-se o conceito de odometria. Por não obter sucesso com essa abordagem, uma solução foi apresentada utilizando-se o conceito de movimentação por GPS (Ferreira et al., 2014).

### 4.9.1 Movimentação por Odometria

A odometria é um dos métodos mais amplamente utilizados para estimar a posição de um robô e é um dos conceitos mais amplamente utilizados em robótica móvel. Sabe-se que a odometria proporciona uma boa precisão em curto prazo, é barata de implantar, mas gera resultados imperfeitos para grandes distâncias percorridas. A idéia geral da odometria é a integração de informação incremental do movimento ao longo do tempo, o qual envolve uma inevitável acumulação de erros. A acumulação de erros de orientação causa grandes erros na estimação da posição, os quais vão aumentando proporcionalmente com a distância percorrida pelo robô. Apesar destas limitações, muitos pesquisadores concordam que a odometria é uma parte importante do sistema de navegação de um robô, e que deve-se usar com medidas do posicionamento absolutas para proporcionar uma estimativa de posição mais confiável (Borenstein and Feng, 1996; Ferreira et al., 2014).

Para a implementação da odometria, é necessário um codificador preso nas rodas do robô que informa quantos passos de revolução foram executadas até aquele momento. Com essa informação é possível estimar a distância de fato percorrida pelo robô. A partir da medida de raio de cada roda ( $r_r$ ) do robô, e a quantidade de passos de revolução ( $p_{rev}$ ) para uma rotação completa desta roda, podemos aplicar a fórmula:  $rev = \frac{p_{rev}}{2\pi \times r_r}$  para descobrir quantos passos são necessários para percorrer determinada distância no ambiente de simulação (Ferreira et al., 2014). Para estimar a quantidade de passos para atingir uma determinada rotação  $\Theta$ , também é necessário conhecer a distância existente entre as duas rodas do robô  $base_r$ , sabendo-se isto, utiliza-se a seguinte fórmula:  $passos = \frac{p_{rev} \times base_r}{(360 \div \Theta) \times 2\pi \times r_r}$ . Especificamente para os robôs e-Puck, é sabido que uma rotação completa da roda é feita em 1000 passos de revolução, a distância entre as rodas é de 52 milímetros e o raio das rodas é em torno de 21 milímetros. O simulador Webots EDU contém um codificador de odometria implementado nativamente para os robôs e-Puck, assim, os experimentos apresentados neste trabalho foram realizados com o auxílio da odometria para a estimação da posição e orientação dos robôs (Ferreira et al., 2014).

### 4.9.2 Movimentação por GPS

O problema da localização é um dos cenários na área de robótica que tem sido bastante investigado e apresenta diversas soluções. Estas soluções incluem a utilização de sensores associados ao sistema de locomoção bem como sensores para identificação de

marcadores externos. Uma solução interessante é a utilização de um sistema de GPS (*Global Positioning Sensor*). Este sistema permite resolver o problema da localização do robô com diferentes resoluções ( $m/cm$ ) mas apresenta desvantagens importantes, dentre elas, limitações à sua utilização no interior de espaços ou em locais muito congestionados (Agrawal and Konolige, 2006), além de um custo ainda considerado alto.

Neste trabalho, um problema encontrado na utilização da abordagem de odometria foi o fato dos robôs não ficarem centralizados na célula do reticulado. Assim, para resolver este problema, foi utilizado o *slot* de GPS, que é disponibilizado no ambiente Webots EDU. Além disso, a correção da guinada (*yaw*) se fez necessária para otimizar o recurso de GPS, já que os robôs realizam vários movimentos de rotação. Para o refinamento da guinada foi necessário utilizar a biblioteca *Inertial Unit library* do Webots EDU.

## 4.10 Considerações finais

Neste capítulo, foram apresentados alguns dos principais conceitos a respeito de robótica coletiva que serão importantes para a compreensão do modelo proposto neste trabalho. Todos os modelos implementados para cada uma das tarefas adotadas neste trabalho seguem o motor de estados para a movimentação individual dos robôs representado pelo AFD proposto em (Lerman et al., 2005) e (Winfield, 2009). No entanto, neste trabalho, o processo embutido na concepção e implementação de cada estado segue regras diferentes. A tarefa de busca e vigília proposta em (Calvo et al., 2011b), mais especificamente o algoritmo denominado (IAS), foi uma importante inspiração para evitar muitas colisões entre os robôs, uma vez que o feromônio invertido tem a propriedade de repelir os agentes, assim como o trabalho de (Ando et al., 2006), que inspirou a utilização do controle de robôs realizado através em sistemas bio-inspirados. Os experimentos foram implementados no ambiente Webots EDU (Coumans, 2013b) simulando o robô e-Puck (EPFL, 2013), pela grande quantidade de trabalhos já realizados a partir desta arquitetura em robótica e também por ser a arquitetura adotada pelo laboratório ao qual esta pesquisa foi realizada.

## Parte II

# Proposta e Resultados



---

## Modelos para controle de robôs

---

A partir do estudo dos trabalhos discutidos da literatura de robótica foi possível observar que modelos para controle de robôs utilizando robótica de enxames e autômatos celulares são escassos e que os trabalhos existentes são desenvolvidos para um propósito específico. Assim, no presente trabalho, modelos de dinâmica coletiva baseados em ACs serviram como inspiração para a criação de um novo modelo baseado em ACs utilizado para sistemas que envolvem robôs móveis de busca em ambientes complexos dispondo de múltiplos recursos. Três tarefas e seus respectivos modelos de navegação propostos neste trabalho são apresentadas neste capítulo: forrageamento, coleta seletiva e busca e resgate. Além disso, as definições e adaptações necessárias para a implementação no ambiente de simulação Webots EDU foram apresentadas.

### 5.1 Modelo de forrageamento

O modelo para a tarefa de forrageamento foi denominado CAAM (Cellular Automata Ant Memory) e publicado em (Lima and Oliveira, 2017a), possui dois níveis: o nível de cada robô individual e o nível do ambiente onde o time de robôs atua. Primeiramente, o nível do robô individual será detalhado; em seguida será apresentado o nível do ambiente.

O modelo de cada robô pode ser descrito por um autômato finito determinístico de quatro estados, assim como é apresentado na Figura 5.1. Cada estado será detalhado para a compreensão do sistema de busca de maneira geral. Inicialmente, cada robô permanece no estado *searching* até que ele encontre algum objeto/comida. Ao encontrar algum objeto dentro de seu raio de visão o robô entra no estado de *grabbing* para capturar o

objeto/comida. Ao capturar a comida, o robô vai até o ninho e deposita a comida. Para tal, o robô tem um conhecimento prévio do ambiente e da localização dos ninhos e transita pelos outros dois estados: *homing* e *depositing*.

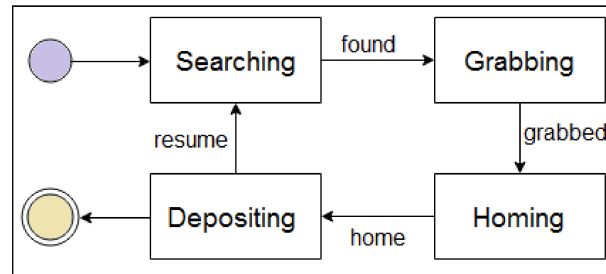
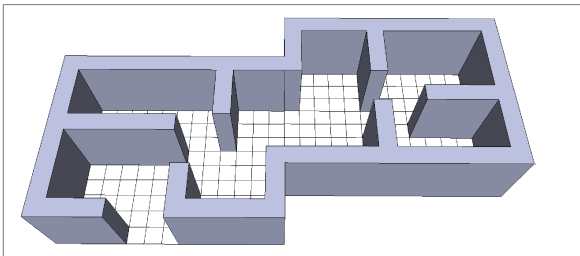
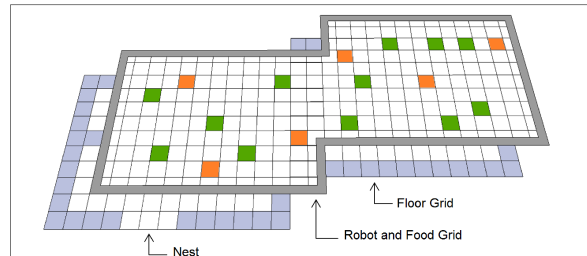


Figura 5.1: Modelo representado em uma máquina de estado finito (Winfield, 2009).

Inicialmente, a representação da estrutura predial 3D (Figura 5.2a) é transformada para uma estrutura 2D (Figura 5.2b). Posteriormente, uma nova grade com robôs e alimentos é sobreposta (Figura 5.2b) sendo que os robôs não podem sobrepor os alimentos e obstáculos e os alimentos não podem sobrepor os obstáculos. O campo de piso possui duas subdivisões. A primeira contempla o campo de piso para o estado *homing* (static and dynamic floor field). A segunda contempla o campo de piso do estado *searching* que é dinâmico e é baseado na técnica bioinspirada que utiliza os traços de feromônio deixados no ambiente para prover a comunicação global e indireta entre os agentes.



(a) Ambiente de simulação para robôs 3D com obstáculos e um único ninho.



(b) Ambiente de simulação 2D com grade de robôs e alimento e malha de piso.

Figura 5.2: Construção do ambiente de simulação com obstáculos e um único ninho para a execução da tarefa de forrageamento pelo enxame de robôs (em verde) e um conjunto de alimentos (em laranja) que deve ser procurado e depositado no ninho (Castro and Lima, 2013a).

Na Figura 5.3 é apresentada uma ideia do comportamento dos robôs durante uma simulação da tarefa de forrageamento com  $N = 3$  robôs e  $F = 21$  alimentos. Inicialmente, conforme pode ser observado na Figura 5.3a, todos os robôs estão no estado *searching* (estado vermelho) e nenhum alimento (representados em azul) foi descoberto. Posteriormente, conforme pode ser observado na Figura 5.3b, um robô entra no estado *grabbing* (robô em roxo), uma vez que o mesmo encontrou um alimento. No próximo passo, conforme pode ser observado na Figura 5.3c, o robô que anteriormente havia encontrado no estado *grabbing* entra no estado *homing* (robô verde) e os demais robôs (robôs vermelhos)



continuam no estado *searching*. Em seguida, conforme pode ser observado na Figura 5.3d, o robô que anteriormente estava no estado *homing* (robô verde) entra no estado *depositing* (robô laranja). Por fim, conforme pode ser observado na Figura 5.3e, o robô que anteriormente estava no estado *depositing* (estado laranja) volta a para o estado *searching* (robô vermelho), enquanto que um segundo robô encontra o alimento e muda de estado. A seguir, o funcionamento de cada estado da máquina de estado finito (MEF) é detalhado.

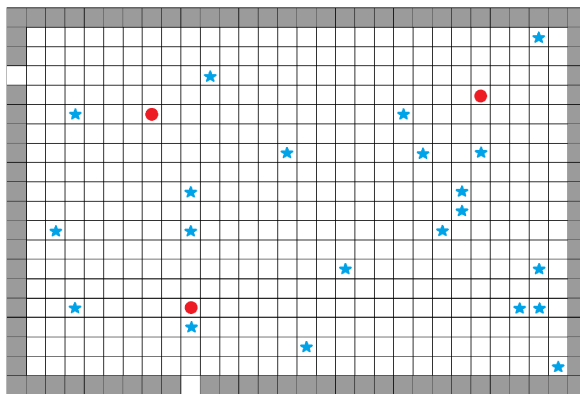
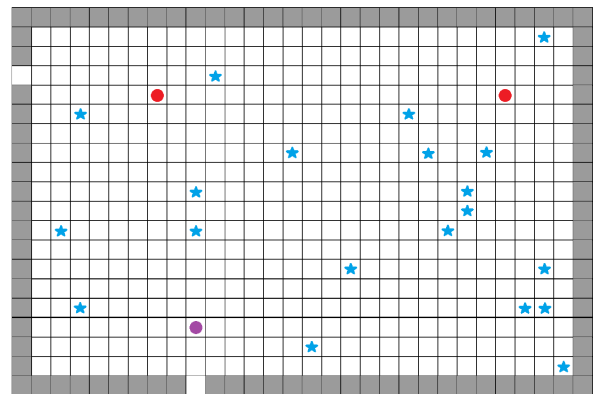
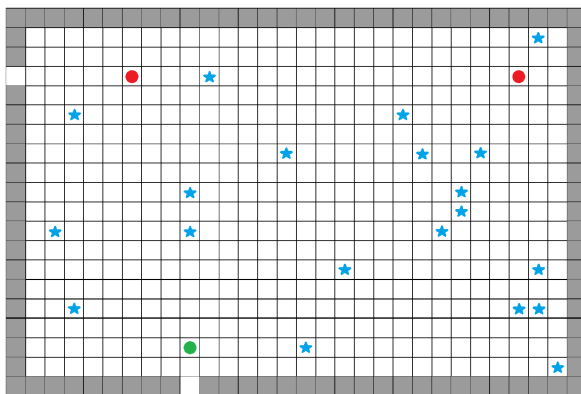
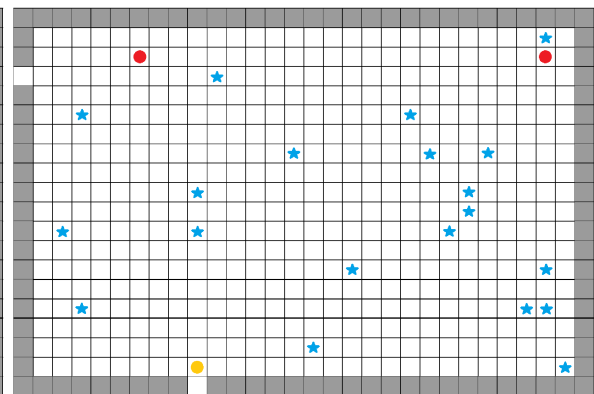
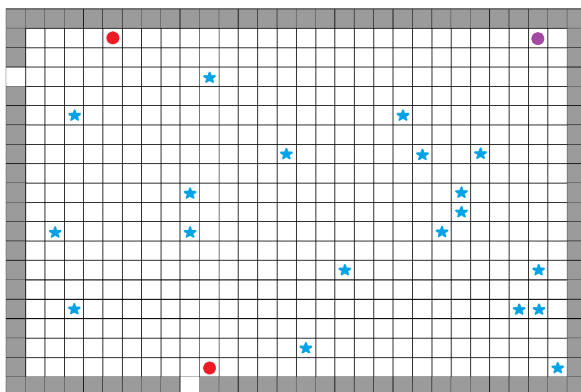
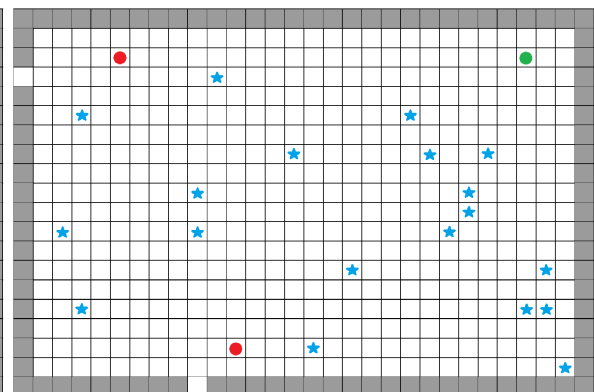
(a) Exemplo searching  $t = 0$ .(b) Exemplo grabbing  $t = 1$ .(c) Exemplo homing  $t = 2$ .(d) Exemplo depositing  $t = 3$ .(e) Exemplo grabbing  $t = 4$ .(f) Exemplo homing  $t = 5$ .

Figura 5.3: Construção do ambiente de simulação com obstáculos e dois ninhos para a execução da tarefa de forrageamento pelo enxame de robôs (vermelho, roxo, verde ou laranja) e um conjunto de alimentos (em azul) que deve ser procurado e depositado no ninho.

O nível do ambiente pode ser dividido em três etapas principais. A primeira etapa consiste na abstração do campo de piso que representa as unidades de comida. A segunda etapa refere-se à criação do campo de piso do processo de *searching*. Neste caso, o ambiente é modificado na presença de um robô em uma determinada célula. A cada passo de tempo o robô deposita feromônio na sua célula e nas células adjacentes. Além disso, esse feromônio possui uma taxa de declínio com o passar do tempo. O feromônio depositado no ambiente tem o efeito de repulsão entre os agentes, pois cada robô escolhe células de menor valor para realizarem seu movimento. O terceiro nível de abstração refere-se ao campo de piso relacionado à busca do ninho para depósito de cada uma das unidades de comida após serem coletados. Para isso, foram utilizados dois campos de piso, um que se refere à distância de cada célula à cada um dos ninhos, e outro que refere-se ao campo de piso dinâmico que dá a contagem da quantidade de robôs perto do ninho para cada célula do campo de piso que contém um robô.

A seguir os 4 estados da MEF apresentada na Figura 5.1. Posteriormente, o processo de resolução de conflitos executado em estados (*searching* e *homing*) será apresentado. Posteriormente, algumas melhorias do modelo inicial serão apresentadas.

### 5.1.1 Busca no forrageamento

No estado *searching*, somente a posição dos ninhos e a geometria do ambiente são determinadas e conhecidas pelo enxame de robôs, o que torna essa busca global. A comunicação pelos agentes é indireta, ou seja, os mesmos não trocam mensagens, mas compartilham um único ambiente geral. Cada robô comunica-se com os demais robôs do enxame de maneira indireta através de um processo coordenado denominado estigmergia, utilizando-se do feromônio, um traço deixado no ambiente enquanto eles se movem, contendo a informação global da trajetória de todos os robôs. Essas interações indiretas obtidas através de traços de feromônio depositados no ambiente criam uma sinergia entre os agentes o que resulta em um comportamento global capaz de realizar a tarefa desejada de forma mais eficiente. Sabe-se que as formigas não trocam informações diretamente, elas apenas trocam estímulos de forma indireta. O feromônio, que está representado em cada célula do campo de piso de busca, é compartilhado globalmente por todos os robôs do enxame. A cada célula é atribuído um valor de constante que representa o feromônio que cada robô deposita durante o processo de busca. Valores mais baixos correspondem às células que são menos visitadas pelos robôs, por outro lado, valores mais altos representam as células que foram mais visitadas ou tangenciadas pelos robôs. Cada célula  $x$  do campo de busca é atribuído o valor 0 no primeiro instante de tempo  $t = 0$ .

Além da informação do campo de feromônio, cada robô pode ter um campo de visão na localização dos alimentos, que simula a capacidade dos robôs processarem imagens locais e identificarem os objetos que devem coletar. Ou seja, um robô é capaz de encontrar

um alimento dentro do seu raio de visão ( $r_v$ ), e este raio de visão é definido a partir da vizinhança da célula central à qual o robô encontra-se naquele instante de tempo. Dessas situações foram utilizados nas simulações: (i) o robô não usa a leitura ( $r_v = 0$ ) ou (ii) considera-se que a precisão dos sensores é capaz de identificar objetos apenas nas células adjacentes ao robô ( $r_v = 1$ ). Assim, foram considerados campos de visão  $r_v = 0$  e  $r_v = 1$ .

A transição de um robô de uma célula para a outra é um modelo de primeira escolha, ou seja, o robô seleciona sempre a célula do campo de piso com menor valor de feromônio que está contida dentro do raio de sua vizinhança. Os conflitos são resolvidos de maneira não determinística, conforme será apresentado na seção 5.1.5. Neste caso, a ideia é justamente utilizar a estigmergia de forma inversa, ou seja, cada robô evita o caminho já percorrido por outro robô, pois isso significa que provavelmente naquela porção de campo de piso não existe nenhuma comida, uma vez que cada robô verifica a existência de alimento ao mover-se para uma determinada célula discreta, diferentemente do modelo de Calvo et al. (2011a) que contemplou um modelo contínuo. Se existem duas unidades de comida dentro do raio de visão do robô, o mesmo dirige-se para a célula onde é apresentado o menor campo de piso. Caso as unidades de comida estejam em duas células de campo de piso iguais, uma delas é escolhida de maneira aleatória. Neste caso, os robôs criam campos de força denominados repulsão (Song et al., 2006). Além de fazer a verificação, o robô ainda deixa um estímulo de feromônio no ambiente. Esse estímulo engloba a célula visitada e as células adjacentes, considerando um raio  $r_f$  do AC, que está sendo considerado na vizinhança para difusão do feromônio, similar ao modelo de dinâmica de pedestres em (Schadschneider, 2001b). Além disso, com o passar do tempo, o traço deixado por cada robô no campo de piso perde parte dos estímulos, caso não seja reforçado, e evapora gradativamente similar ao declínio de feromônio descrito em (Schadschneider, 2001a). O valor do feromônio no campo de piso da busca é determinado a cada instante. Inicialmente em  $t = 0$ , todas as células recebem um valor inicial de feromônio igual a zero. Posteriormente, cada célula visitada por um robô é acrescida por uma constante  $\delta$  na célula central ou  $\delta'$  na célula da vizinhança dentro do raio de difusão  $r_f$ , conforme é apresentado na Equação 5.1.

$$x_{(i+a)(j+b)}^{t+1} = \begin{cases} x_{(ij)}^t + \delta & \leftarrow a = 0, b = 0 \\ x_{(i+a)(j+b)}^t + \delta' & \leftarrow 0 < a \leq r_f, 0 < b \leq r_f \end{cases} \quad (5.1)$$

A etapa de declínio afeta todas as células do reticulado a cada passo de tempo  $t$ : toda célula com valor diferente de zero é decrescida por uma constante  $\beta$  conforme é apresentado na Equação 5.2.

$$x_{ij}^{t+1} = x_{ij}^t - \beta \quad (5.2)$$

Na implementação de versões preliminares do modelo de forrageamento, foi observado uma tendência à repetição de caminhos no modelo puramente baseado em feromônio

invertido. A fim de reduzir a repetição de caminhos já percorridos, um fato recorrente nesse modelo de feromônio invertido, uma memória finita  $Q$  auxiliar foi implementada, conforme apresentado na Figura 5.4, inspirada na Tabu Search (Glover, 1989), (Glover, 1990). O modelo sem a utilização da memória é denominado neste trabalho de Cellular Automata Ant (CAA) model usado na análise formal. Enquanto que a aplicação da memória faz com que o modelo seja denominado de CAAM.

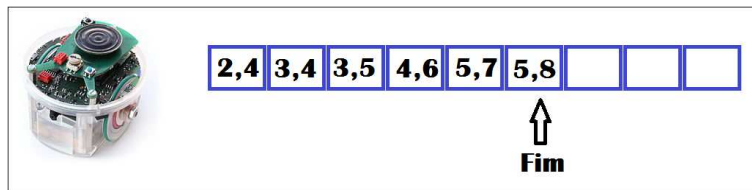


Figura 5.4: Exemplo de memória com política de fila para o robô da arquitetura e-Puck proposta para o processo de busca (EPFL, 2013).

A memória foi implementada com a política de fila, sendo muito importante para otimizar o processo de busca, já que evita que os robôs escolhessem uma célula já visitada por si mesmo recentemente, aumentando dessa forma a performance do sistema. Assim, a cada passo de tempo  $t$  quando um robô visita uma determinada célula no processo de busca, ele adiciona a posição exata dessa célula à sua fila. Antes de mover-se para qualquer célula pelo modelo de primeira escolha, o robô percorre a sua fila e verifica se aquela célula não foi previamente visitada num passo de tempo anterior. Neste caso, o robô escolhe uma outra célula para realizar o seu movimento, desde que esta não esteja na fila.

Duas situações condicionais também devem ser verificadas. A primeira refere-se à quando o robô encontra-se na situação de *deadlock*, ou seja, não existe mais nenhuma célula na vizinhança que não tenha sido visitada pelo robô. A segunda refere-se à condição de fila cheia. Nas duas situações, a fila daquele robô é totalmente esvaziada, pelo fato de que muitas das vezes torna-se dispendioso remover um elemento por vez, já que o robô pode voltar a uma situação de *deadlock* ou fila cheia em um curto passo de tempo. À medida que esta fila cresce, o processo de busca passa a consumir muito processamento, portanto, um ajuste nas constantes que determinam o tamanho desta fila torna-se essencial. Além disso, memórias muito grandes acabam gerando áreas que são bloqueadas pelo próprio robô e que poderia conter um objeto ou *deadlocks* que esvaziariam periodicamente a fila tornando-se um processo desvantajoso.

### 5.1.2 Captura no forrageamento

O processo de recolher e capturar o alimento pelo robô pode ser entendido de duas maneiras, dependendo do raio de alcance da visão do robô. Quando o robô não tem raio de visão ( $r_v = 0$ ), o mesmo deve sobrepor o alimento para então capturá-lo. Neste caso,

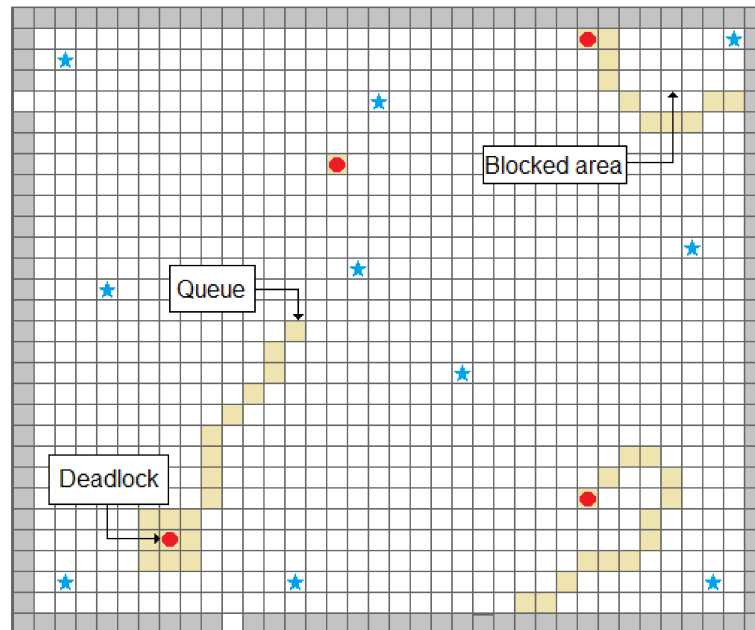


Figura 5.5: Exemplo de formação de filas pelos robôs e criação de zonas de bloqueio.

mesmo quando o robô passa numa célula adjacente ao alimento, o robô acaba não vendo o alimento, podendo ir para uma célula diferente de onde esteja aquele alimento. Neste caso, existe uma economia de processamento, mas uma perda de eficiência do algoritmo.

Por outro lado, quando o robô possui um nível de visão radial  $r_v \geq 1$ , então, apesar de existir um processamento maior, o robô é capaz de já prever em  $r_v$  passos anteriores que ele pegará o alimento, o que já é considerado o estado *grabbing*. Uma vez que esta ação possui prioridade em relação aos demais robôs e também em relação ao próprio campo de piso, o valor de campo de piso para realizar o movimento, quando o mesmo já percebe o alimento, pelos sensores. Nesse caso, o robô irá diretamente para a célula que contém o alimento em  $r_v$  passos de tempo.

Uma vez situado na célula onde o alimento identificado se localiza, o modelo faz com que o robô permaneça nesse estado pelo número de passos necessários à conclusão da operação de coleta propriamente dita. Neste trabalho, em todas as simulações, o modelo considera um único passo de tempo para a coleta, por uma questão de simplicidade. Entretanto, em uma aplicação real com arquiteturas com garras, por exemplo, a etapa de captura pode ser mais demorada ou mesmo dotada de um controle mais complexo que garante a conclusão da tarefa, sem prejuízo do restante do modelo.

### 5.1.3 Regresso no forrageamento

Cada robô no processo de *homing* carrega consigo um único objeto (alimento). Além disso, cada robô conhece o ambiente como um todo, ou seja, ele sabe onde fica cada um dos ninhos e obstáculos do ambiente. O campo de piso compartilhado pelos robôs segue o algoritmo proposto em (Varas et al., 2007). Note que qualquer outro algoritmo de distân-

cias poderia ser utilizado, tais como, distância Euclidiana, de Manhattan ou de Disjsktra (Alizadeh, 2011). Neste caso, o modelo considera que as células mais próximas ao ninho  $A_i$  assumem os menores valores, enquanto que as células mais distantes são atribuídas com valores mais altos. O campo de piso estático  $W_{static}^i(x)$  é calculado iterativamente. Inicialmente, sala é dividida em uma grade retangular e ao ninho  $A_i$  é atribuído um valor “ 1 ”, sendo considerada a primeira camada a ser calculada. Posteriormente, todas as células consideradas de segunda camada, adjacentes às células de primeira camada, são calculadas de acordo com as seguintes regras:

1. Se uma célula tem valor “ $\eta$ ”: (i) todas as células adjacentes nas direções verticais ou horizontais são atribuídas com o valor “ $\eta + 1$ ” e (ii) todas as células adjacentes nas direções diagonais são atribuídos o valor “ $\eta + \lambda$ ”.
2. Se existem conflitos na atribuição de um valor de uma célula, uma vez que cada célula é adjacente a várias outras células que podem ter valores  $\eta$  ou posições diferentes o menor valor calculado de acordo com cada vizinha é atribuído à célula em conflito.

No algoritmo, considera-se  $\lambda > 1$ , que representa o fato de que a distância entre duas células diagonais adjacentes é maior que nas direções horizontal ou vertical. Nos experimentos, foi utilizado o valor  $\lambda = \frac{3}{2}$ , sendo este o valor que mais se aproxima de  $\sqrt{2} = 1.414$ , que representa a hipotenusa do triângulo.

À medida que novas células têm o seu valor de piso atribuído, suas vizinhas terão seu valor calculado no próximo passo de tempo. O processo é repetido até que todas as células são avaliadas. Paredes também são considerados quando da definição do grid. Às células pertencentes às paredes são atribuídos valores muito elevados de campo piso. Isto assegura que os robôs nunca tentarão ocupar uma destas células.

Foi apresentado em (Alizadeh, 2011) que o uso apenas do campo de piso estático como em (Varas et al., 2007) não é apropriado a este tipo de modelo, porque longas filas foram observadas próximo às metas. De forma, similar, esse comportamento seria incompatível na tarefa de forrageamento, quando é esperado um grande número de robôs no enxame. Assim, neste trabalho foi utilizado um campo de piso estático em conjunto com um campo de piso dinâmico que ajuda a evitar a formação de filas próximas ao ninho. Para implementar esta estratégia, o algoritmo de (Alizadeh, 2011) foi utilizado, no qual o número de robôs em torno do ninho é utilizado e influencia a escolha do melhor ninho a depositar o alimento. No presente modelo, diferentemente do modelo em (Alizadeh, 2011), o robô não troca essa escolha ao longo da movimentação depois que ele decide o ninho para depositar o alimento coletado. No entanto, o campo de piso dinâmico tenta evitar a formação de longas filas e a diminuição de conflitos próximo aos ninhos. Neste modelo, é determinado o peso de uma célula  $x$  em relação à um ninho  $A$  no  $i$ -ésimo passo de tempo  $W_i^A(x)$ . Inicialmente,  $W_{static}^A(x)$  é utilizado de tal forma que ele representa a

distância de cada célula  $x$  do reticulado até o ninho  $A_i$ . Para cada célula  $x$  que é ocupada ( $v_i$ ) por um robô na etapa  $i$ -ésimo deve-se calcular  $T_i^A(x)$  de acordo com a Equação 5.3.

$$T_i^A(x) = P_i^A(x) + \frac{1}{2}E_i^A(x) \quad (5.3)$$

De tal maneira que as seguintes premissas devem ser satisfeitas:

1. if  $N < \max(W_{static}^A(x))$   
 $\rightarrow$  then  $W_i^A(x) = W_{static}^A(x) + \alpha \times T_i^A(x)$
2. else  $W_i^A(x) = \max(W_{static}^A(x))$

Conforme a seguinte configuração:

1.  $P_i^A(x) = \{y | W_{static}^A(y) < W_{static}^A(x) \text{ and } y \in V_i\}$ ,
2.  $E_i^A(x) = \{y | W_{static}^A(y) = W_{static}^A(x) \text{ and } y \in V_i\}$ .

Os valores  $P_i^A(x)$  e  $E_i^A(x)$  representam o número de robôs que estão em células de valores menores ( $P$ ) ou iguais ( $E$ ) às que possuem um robô que coletar um objeto estão no estado *homing*. Em (Alizadeh, 2011) todas as outras células são calculadas (vazias ou ocupadas por um pedestre). Neste modelo, somente as células que contêm um robô são calculadas e para o valor de cada uma dessas células é acrescido um valor que representa quantos robôs pertencem à células menores ou iguais que o robô da célula que está sendo avaliada. Se a constante  $\alpha$  for igual a 0, então o modelo torna-se o modelo proposto em (Varas et al., 2007). Quando usamos um valor para  $\alpha > 0$  os robôs não são capazes de mudar o caminho para um ninho previamente escolhido, mas com essa modificação os robôs são capazes de evitar congestionamento e formação de filas próximo aos ninhos. Finalmente, os pesos de uma célula  $x$  no passo  $i \leq 1$  é ajustado por  $W_i(x)$  e é calculado pela seguinte Equação 5.4:

$$W_i(x) = \min\{W_i^A(x) | A \text{ é um ninho do ambiente}\}. \quad (5.4)$$

Como estamos utilizando um modelo de primeira escolha, então o robô escolhe a célula de menor valor do campo de piso *homing* durante o processo de depósito e evita os conflitos, usando os mesmos critérios no estado *searching* e que serão descritos na seção 5.1.5. Além disso, o robô é conhecedor da posição à qual ele faz parte através de um compartilhamento global e centralizado de informações e adicionalmente, cada robô sabe para onde ele deve mover-se, uma vez que cada um desses robôs recebe um mapeamento do ambiente.

### 5.1.4 Depósito no forrageamento

O *depositing* é o último estado do ciclo percorrido pelo robô para coletar um objeto e levá-lo ao ninho. Quando o robô chega ao ninho e está carregando uma comida, o robô deixa a comida nessa célula e volta ao ambiente para procurar mais comida, saindo da posição de depósito no ninho e retornando ao estado *searching*. Nesse modelo, foi considerado apenas um passo de tempo. No entanto, se já não existe mais nenhuma comida no ambiente e não existe mais nenhum robô no estado *homing*, então o processo de busca é encerrado.

### 5.1.5 Resolução de conflitos e movimentos do robô

Cada robô tenta movimentar-se para uma célula de menor valor, tanto quando está no estado *searching* ou ainda quando está no estado *homing*. A diferença entre os dois está na métrica utilizada. No caso do *searching*, essa métrica é dada pelo campo de piso de busca e equivale ao valor do feromônio nas células, considerando-se um possível bloqueio pela fila  $Q$ . No caso do *homing*, essa métrica é dada pelo campo de piso de *homing* e equivale ao valor de  $W_i(x)$  dado pela Equação 5.4. No entanto, alguns conflitos podem ocorrer durante essa decisão de se deslocar para a célula de menor valor. Em (Varas et al., 2007) foram apresentados alguns desses conflitos, conforme exemplificado na Figura 5.6, onde os números correspondem ao valor do campo de piso em uma célula, os arcos representam as intenções de movimento, e os símbolos representam os pedestres em várias situações que serão descritas a seguir. (a) Círculo preenchido (representa uma situação sem conflitos): o robô movimenta-se para uma célula adjacente que apresenta o menor campo de piso (no estado *searching* ou *homing*) naquele momento, minimizando sempre a sua escolha. (b) Círculos abertos (situação de conflito entre dois robôs): dois robôs tentam movimentar-se para a mesma célula, caracterizando um conflito. Nesse caso, um valor aleatório decide qual robô realizará o movimento, enquanto os perdedores ficam parados. (c) Triângulo preenchido (situação de conflito entre duas células): duas células com o mesmo campo de piso, os robôs decidem aleatoriamente para qual célula devem realizar o movimento. É possível perceber que esse método para resolução de conflitos agrega não determinismo ao modelo de forrageamento, mesmo que a decisão seja realizada por um modelo de primeira escolha.

No modelo em (Varas et al., 2007), além dos conflitos na Figura 5.6 a situação de pânico é tratada. Entretanto, neste trabalho, devido ao fato de se tratar um modelo de enxames de robôs para a realização da tarefa de forrageamento, optamos por não contemplar o comportamento social de pânico agregado ao modelo apresentado em (Varas et al., 2007), uma vez que o objetivo do enxame de robôs é realizar a tarefa proposta no menor tempo possível e o comportamento social de pânico encontrado em pedestres numa situação de risco, não contribui para o desempenho do enxame.



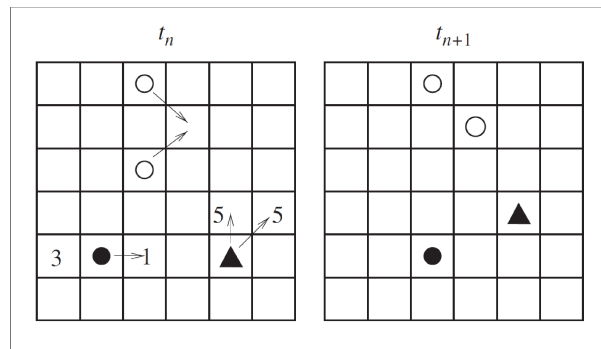


Figura 5.6: Movimentos possíveis para os robôs adaptado do modelo de (Varas et al., 2007).

Em relação à escolha da comida, se existem duas ou mais unidades de comida dentro do raio de visão do robô, a unidade de comida com o menor valor de campo de piso é escolhida. Por outro lado, se duas ou mais unidades de comida estão em células de campo de piso iguais, uma unidade de comida é escolhida aleatoriamente pelo robô. Adicionalmente, para evitar o efeito de inércia, em alguns momentos, foi necessário utilizar a cada  $t$  passos de tempo uma escolha de célula estocástica por todos os robôs.

### 5.1.6 Variações no modelo de forrageamento

O modelo inicial proposto para a tarefa de forrageamento foi denominado CAAM (Cellular Automata Ant Memory) model e as seções anteriores apresentaram suas principais características. Os principais resultados e análises dos experimentos de simulação envolvendo esse modelo são apresentados no próximo capítulo e serviram de base para a elaboração do artigo (Lima and Oliveira, 2017a). Posteriormente, quatro variações foram propostas a partir do modelo de forrageamento CAAM nas etapas de *searching* e *homing*, através de novas investigações publicadas em (Lima and Oliveira, 2016a) e em (Lima and Oliveira, 2016b). Cada uma dessas variações é detalhada nas seções a seguir.

#### Alteração da política de memória na busca

O modelo denominado Cellular Automata Ant Queue (CAAQ), contempla uma alteração na política da memória do utilizada no modelo CAAM. O algoritmo implementado no modelo inicial (CAAM) apresenta uma memória finita de tamanho  $Q$  que é esvaziada completamente quando a mesma está cheia ou quando o robô encontra-se em deadlock, situação em que as células da vizinhança estão em  $Q$  impedindo a escolha da próxima célula. Para evitar que a memória seja sempre esvaziada e que os robôs percam sua memória recente de caminhos percorridos foram avaliadas outras políticas de gerenciamento da memória dos robôs, conforme é apresentado a seguir.

1. A memória é completamente esvaziada no caso de deadlock e é completamente

- esvaziada no caso de memória cheia, e os novos elementos são inseridos à medida que o robô realiza seu movimento (política original do CAAM).
2. A memória é completamente esvaziada no caso de deadlock e remove-se apenas o primeiro elemento que entrou no caso de memória cheia, assim a nova posição é inserida ao final da fila (primeira política avaliada no CAAQ).
  3. A memória sofre alteração de apenas do primeiro elemento no caso de deadlock, assim a nova posição é inserida ao final da fila, e é completamente esvaziada no caso de memória cheia (segunda política avaliada no CAAQ).
  4. A memória não é esvaziada em caso de fila cheia ou deadlock utilizando-se a política de fila para os dois casos (terceira política avaliada no CAAQ).

Os resultados na avaliação dessas políticas são apresentadas e discutidas no próximo capítulo.

### Regresso probabilista elitista

Uma alternativa probabilística para a modelagem do estado de *homing* foi avaliada dando origem ao modelo Probabilistic Cellular Automata Ant Memory (PCAAM), que também foi inspirado por modelos probabilísticos anteriores baseados em AC para a modelagem de dinâmica de multidão durante uma evacuação (Schadschneider et al., 2011). No modelo PCAAM não é considerado o campo de piso dinâmico adotado no modelo original CAAM (Lima and Oliveira, 2017a), a partir de uma adaptação do campo de piso dinâmico em (Alizadeh, 2011). Esta nova abordagem foi criada com o intuito de melhorar o tempo que se leva para finalizar a tarefa de forrageamento e também reduzir o tempo de processamento gerado a partir do cálculo do piso dinâmico. A nova ideia de se utilizar um modelo probabilista partiu do pressuposto de que a aglomeração perto dos ninhos seria diminuída através da inserção de um elemento probabilista para a escolha das novas células. No entanto se essa escolha for extremamente aleatória, a busca tende a se afastar muito da rota ótima. Esse comportamento deve-se ao fato de que o campo de piso estático fornece a distância ótima entre qualquer célula do reticulado e cada um dos ninhos. Entretanto, para um mesmo ninho, as distâncias das células de uma vizinhança têm diferenças mínimas entre elas, podendo assumir uma diferença máxima de 3,, equivalente à distância entre 2 células em diagonais opostas, se for adotado  $\lambda = 1,5$ . Assim, se for realizado um sorteio probabilístico baseando-se apenas nas informações de campo de piso (de forma similar à ideia da roleta no AG) para escolha da próxima célula, aumente bastante a probabilidade de escolha de uma célula mais distante da meta, a cada passo. Como resultado, os robôs não conseguem chegar com facilidade aos ninhos, atrasando a finalização da tarefa de forrageamento.

No modelo PCAAM esse comportamento estático é corrigido reduzindo-se bastante a probabilidade de escolha das piores vizinhas. Cada robô escolhe um caminho a seguir no

estado de *homing*, considerando-se a distância para o ninho mais próximo em cada célula da sua vizinhança. Esta informação é armazenada em cada célula  $x_{ij}$  em  $W_{static}(x_{ij}^t)$ . O processo de decidir o movimento é uma função probabilística elitista, onde o robô escolhe com alta probabilidade as células mais próximas ao ninho, depois da leitura dos valores de  $W_{static}(x_{ij}^t)$  associados a todas as células da vizinhança baixos valores de probabilidade são associados às demais células da vizinhança. O processo de decisão do próximo passo inicia-se atribuindo valores  $c_{ij}$  para cada célula da vizinhança. Se uma ou mais células tem o menor valor  $W_{static}(x_{ij}^t)$  no  $\eta_{x_{ij}}^m$ , o valor  $c_{max}$  é atribuído às células. Por outro lado, se as células não têm o valor mais baixo, o valor  $c_{min}$  é atribuído às células. As variáveis  $c_{max}$  e  $c_{min}$  são parâmetros do modelo que devem ser ajustados pelo projetista. O ideal é que o sorteio tenha uma tendência elitista, porém permitindo que outros pontos sejam sorteados eventualmente. Cada célula tem uma probabilidade  $c_{ij}$  de ser escolhido dentre um total  $\sum_{ij=0}^m c_{ij}$ , que representa a soma de todos os valores  $c_{ij}$  na  $\eta_{x_{ij}}^m$ . A Equação 5.5 representa a função de **probabilidade decisão**  $P(x_{ij})$ .

$$P(x_{ij})^{t+1} = \begin{cases} \frac{c_{max}}{\sum_{ij=0}^m c_{ij}} & \leftarrow x_{ij}^t \leq \forall(x_{ij}) \in \eta_{x_{ij}}^m \\ \frac{c_{min}}{\sum_{ij=0}^m c_{ij}} & \leftarrow x_{ij}^t > \forall(x_{ij}) \in \eta_{x_{ij}}^m \end{cases} \quad (5.5)$$

Após o processo de decisão do próximo passo, o robô **movimenta-se** para a célula  $x_{ij}$  escolhida na sua vizinhança. Este movimento de escolha probabilístico elitista impede a formação de linhas perto dos ninhos (Varas et al., 2007), jamming (Yamamoto et al., 2007), comportamento de inércia (Yang et al., 2005), e permite escolhas com caminho alternativo, quando comparados à uma escolha puramente determinística, em um modelo de primeira escolha. A título de exemplo, esses fenômenos (filas, jamming e inércia) são observados no modelo CAAM quando se usa  $\alpha = 0$ , uma vez que o campo de piso dinâmico é ignorado e o modelo torna-se de primeira escolha. Os resultados e análises do modelo PCAAM foram apresentados em (Lima and Oliveira, 2016b) e são discutidos no próximo capítulo.

### Regresso probabilista híbrido

Uma nova estratégia de *homing*, denominada Hybrid Cellular Automata Ant Memory (HCAAM), foi avaliada e publicada em (Lima and Oliveira, 2016a) visando melhorar o desempenho da equipe e para distribuir igualmente os agentes nos ninhos durante o processo de deposição de objetos. A partir de experimentos com o modelo inicial CAAM foi possível constatar que os ninhos não são igualmente utilizados nem mesmo com a adoção do campo dinâmico. Os agentes tendem a ir diretamente para o ninho que tem a menor distância para todas as células do meio ambiente, devido à força do campo de piso estático. Este fenômeno ocorre devido à forte força física que é dada ao campo estático, ou seja, o agente sempre tende a ir para a célula de menor valor.

Visando a obtenção da utilização mais distribuída dos ninhos (ver Figura 5.7), a nova estratégia de *homing* usa uma escolha probabilística sobre uma grande parte do ambiente. No entanto, como será mostrado neste trabalho, um movimento puramente probabilista atrasa o desempenho da equipe devido à fraqueza no poder que é dado para o campo de piso estático  $W_{static}^A(x_{ij})$ , que contém o menor caminho entre qualquer célula e um ninho  $A$ . Uma vez que as distâncias em uma vizinhança de raio 1 pode diferir 2 unidades no caso extremo, quando o robô está longe de qualquer ninho as probabilidades envolvidas são muito semelhantes e torna a escolha aleatória ruim, pois mistura-se a convergência para o ninho. A Equação 5.6 apresenta o movimento probabilista se a célula é maior do que  $x_{ij} > d_k^A$  (distância estipulada), que  $c_{ij}^h = |\eta_{x_{ij}}^m| * 10 - 7 * k$ , para cada  $1 < k < |\eta_{x_{ij}}^m|$  se  $x_{ij}^t \leq \forall(x_{ij}) \in \eta_{x_{ij}}^m$ , que evita a mesma escolha ninho. O valor  $c_{ij}^h$  representa cada célula avaliada no  $\eta_{x_{ij}}^m$  vizinhança.

$$P(x_{ij})^{t+1} = \frac{c_{ij}^h}{\sum_{ij=0}^m c_{ij}^h} \leftarrow x_{ij} > d_k^A \quad (5.6)$$

Numa segunda fase, a escolha probabilística é agregada com uma estratégia de movimento probabilístico elitista Lima and Oliveira (2016a), aqui designado como estratégia híbrida. Em seguida, os valores de distância para o ninho são atribuídos  $c_{ij}$  para cada célula assim como Lima and Oliveira (2017a). Se uma célula tem o menor valor no  $\eta_{x_{ij}}^m$ , a  $c_{max}$  valor é atribuído à célula. Em todos os outros casos,  $c_{min}$  valor é atribuído à célula. Cada célula tem um  $c_{ij}^e$  probabilidade de ser escolhida de um total de  $\sum_{ij=0}^m c_{ij}$  no  $\eta_{x_{ij}}^m$ . A equação para o movimento da escolha elitista probabilística  $P(x_{ij})$  é mostrada na Equação 5.5. Quando um robô se aproxima de uma certa distância  $d_k^A$  de um dos ninhos  $A$ , o robô irá convergir para o movimento probabilista elitista como mostrado na Equação 5.5 da atual célula  $x_{ij}$  para o ninho mais próximo  $A$ . A vantagem da ideia aqui proposta combina o poder de um movimento quase aleatório, e um atrator que faz movimentos mais precisos perto do ninho selecionado usando a Equação 5.5. Se a célula de valor inferior  $x_{ij} \leq d_k^A$  ao selecionado, os agentes são guiados pela Equação 5.5. Em todos os outros casos, os robôs são guiados através da Equação 5.6.

### Regresso probabilista híbrido com cálculo de densidade

Uma última estratégia de *homing* foi avaliada e o modelo de forrageamento resultante foi denominado Density Cellular Automata Ant Memory (DCAAM). Essa estratégia visa melhorar a distribuição dos ninhos e também o desempenho médio de tempo  $T$  da equipe (ver 5.8). O estado *homing* aqui proposto baseia-se na ideia de que as formigas têm a particularidade de contagem de densidade perto dos ninhos (Gordon, 2014). Neste caso, cada formiga é capaz de reconhecer uma outra companheira-de-ninho, devido às diferentes taxas de hidrocarbonetos espalhados pelo ambiente (Gordon, 2007). Eles também

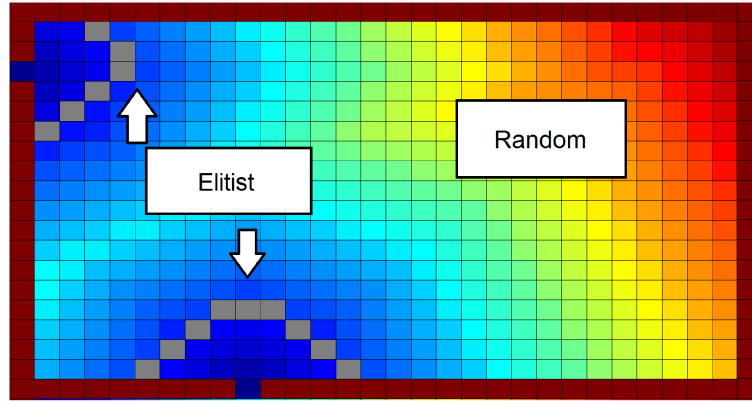


Figura 5.7: Exemplo do cálculo probabilista híbrido onde para células mais distantes do ninho uma escolha mais randomizada é realizada e à medida que os agentes chegam perto dos ninhos os agentes tendem a realizar movimentos mais elitistas, se aproximando mais do modelo determinístico de primeira escolha.

são capazes de reconhecer que formiga é responsável por cada tarefa no ninho (Gordon, 2013), e quando é o melhor momento  $t$  para sair para forragear ou voltar para o ninho. Uma formiga não será capaz de sair e procurar comida, se há várias outras formigas à procura de alimentos ao mesmo tempo. Se há várias de formigas que saem do ninho e há muitas formigas no estado *homing*, uma grande quantidade de colisões provavelmente irá acontecer.

Inspirado nesse comportamento biológico uma estratégia de coordenação da equipe de robôs foi proposta, que pode ser considerada híbrida. Nessa estratégia, os agentes são capazes de usar o cálculo de densidade  $\rho_{A_k}$  de cada ninho  $A$  para selecionar o ninho a uma certa distância  $d_{A_k}^t$  dada pela Equação 5.15.

$$\rho_{A_k}^t = \sum_{ij=0}^{A_k^v} x_{ij}^t \quad (5.7)$$

No caso em que os ninhos são igualmente lotados, o campo de piso estático  $W_{static}(x_{ij}^t)$  é usado. Deve-se lembrar que o campo de piso estático está associado ao movimento probabilístico elitista como na Equação 5.5. O agente é capaz de decidir qual é a melhor solução, ir para o ninho mais próximo  $A$  ou ir para um que não esteja congestionado como mostrado na Equação 5.16.

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t \in W_{(x_{ij}^t)}^{A_k^t} & \leftarrow \rho_{A'_k}^t < \forall \rho_{A_k}^t, k \geq 1 \\ x_{ij}^t \in \min(W_{(x_{ij}^t)}^{A_k^t}) & \leftarrow \rho_{A'_k}^t = \forall \rho_{A_k}^t, k \geq 1 \end{cases} \quad (5.8)$$

Se há muitos agentes esperando para depositar sua comida no ninho de uma fila será formada perto dele criando um efeito de bloqueio e o ninho será bloqueado. Além disso, se um lote de unidades de alimentos estão sendo depositados e vários agentes estão livres para procurar novamente, campos de forças opostas aparecem perto dos ninhos, similar

ao efeito de parentesco (Schadschneider, 2001b). Essas forças são chamadas de forças de repulsão, retardam o desempenho da equipe. Usando o modelo DCAAM, os robôs evitam a formação de linha perto dos ninhos por causa de uma boa distribuição entre eles. A abordagem do estado *homing* no modelo DCAAM impede, a formação de linhas perto dos ninhos Varas et al. (2007), obstrução dos ninhos Yamamoto et al. (2007), comportamento de inércia Yang et al. (2005), e permite um caminho para o ninho com escolhas alternativas, sendo essa uma das melhorias do nosso trabalho.

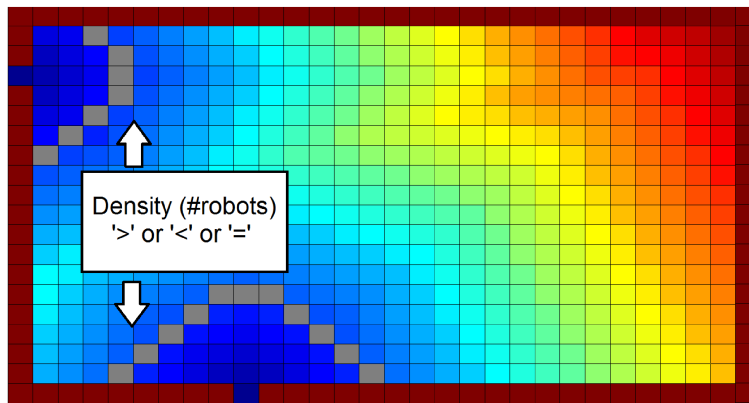


Figura 5.8: Exemplo do cálculo probabilista híbrido com cálculo de densidade onde para células mais próximas ao ninho um cálculo de densidade da quantidade de outros agentes é calculada e repassada ao agente. Com esse cálculo é possível encontrar o melhor melhor ninho para se depositar o objeto coletado pelo agente.

### 5.1.7 Comparação entre o modelo proposto e modelos da literatura

O modelo CAAM para a navegação de multi-robôs e suas respectivas variações (CAAQ, PCAAM, HCAAM, DCAAM) propostas nesta tese utiliza diversas técnicas presentes na literatura, por isso pode ser considerado um sistema híbrido. Entretanto, nenhum outro sistema investigado na literatura faz a utilização simultânea de todos os algoritmos e estratégias apresentados no presente trabalho. Dentre essas técnicas podemos destacar a utilização de: busca Tabu, autômatos celulares, dinâmica coletiva de pedestres e robótica de enxames, sendo analisados os efeitos sociais (inércia, fricção, obstrução, congestionamento, formação de filas, arqueamento) decorrentes da tarefa e também os efeitos de tempo para a conclusão da tarefa, cobertura do ambiente e a distribuição da tarefa por cada um dos agentes.

A modelagem do comportamento individual de um robô através de um AFD de quatro estados também foi adotado nos modelos de forrageamento investigados em (Lerman et al., 2005) e (Winfield, 2009). Entretanto, o funcionamento de cada estado no modelo aqui descrito é bem mais complexo e inspirado em outros modelos da literatura, conforme

descrito a seguir.

Neste contexto, podemos destacar que a elaboração do processo de *searching* baseia-se na ideia de utilização de feromônio invertido, proposto inicialmente em (Calvo et al., 2011a). Esta técnica de busca permite que exista uma melhor distribuição do time de robôs dentro do ambiente. No entanto, neste trabalho, foi utilizado um modelo de difusão de feromônio discreto e mais complexo, com novas constantes para prover um melhor espalhamento do time de robôs. Além disso, em (Calvo et al., 2011a) o modelo de busca é probabilístico, enquanto que no modelo inicial CAAM utilizamos um modelo de primeira escolha com resolução de conflitos não-determinística. Ou seja, uma célula de menor valor é escolhida pelo robô e caso existam células com o menor valor, uma delas é escolhida aleatoriamente. Além disso, os modelos de dinâmica populacional descritos em (Nishinari et al., 2006) também adotaram um modelo de feromônio. Entretanto, o mesmo foi utilizado como elemento de atração e não de repulsão. Por outro lado, os modelos (Kirchner and Schadschneider, 2002) e (Schadschneider, 2001a) usaram o conceito de declínio e difusão, sem uso da terminologia de feromônio. Adicionalmente, para evitar a repetição de caminhos no CAAM uma estratégia de memória curta baseada em fila foi adotada. A fila utilizada neste trabalho é inspirada na Tabu Search (Glover, 1989). Essa fila, além de prover uma disposição mais esparsa do time de robôs dentro do ambiente também ajuda o time evitar a visitação de locais recentemente percorridos, especialmente quando os parâmetros não estão inicialmente ajustados.

O processo de *grabbing* considera que cada robô seja capaz de coletar um único objeto por vez como em (Beckers et al., 1994). Além disso, quando o robô verifica que tem um objeto dentro do seu raio de visão ele pega o objeto dentro no próximo instante de tempo, desde que não seja detectado nenhum conflito. O processo de *depositing* é constituído de um único passo de tempo e consiste em depositar a comida coletada pelo robô dentro do ninho escolhido.

A elaboração do processo de *homing* deste trabalho pode ser associada aos modelos de dinâmica de pedestres em (Varas et al., 2007) e (Alizadeh, 2011) e (Schadschneider et al., 2011). Neste trabalho, o campo de piso estático é baseado no trabalho proposto em (Varas et al., 2007) e a distância resultante é baseada no número de passos mínimos até a saída, mas poderia ser alterado pelas métricas de Manhattan ou Euclidiana como proposto por (Schadschneider, 2001b). A inserção do campo de piso dinâmico, associado ao campo de piso estático, foi baseado no trabalho de (Alizadeh, 2011). No entanto, neste trabalho, uma alteração do piso dinâmico foi adotada. Essa alteração permite que o arqueamento investigado em (Alizadeh, 2011), (Wei-Guo et al., 2006) e (Perez et al., 2002) seja mais suave para o processo de *homing*. A Figura 5.9 exemplifica esse fenômeno apresentando dois exemplos de simulação do CAAM. O primeiro com  $\alpha = 0$  (sem o uso do piso dinâmico) e o segundo usa  $\alpha = 1$  (com o piso dinâmico interferindo na escolha dos robôs).

Posteriormente no modelo DCAAM também investigado neste trabalho, a inserção da abordagem biológica no contexto de robótica do trabalho de (Gordon, 2013) trouxe ao trabalho uma melhor utilização dos ninhos. Além disso, evita: aglomerações (Yamamoto et al., 2007), comportamento de inércia (Yuan and Tan, 2007), congestionamento (Li-jun et al., 2009), a formação de filas de espera Varas et al. (2007) e a obstrução (Song et al., 2006) em frente aos ninhos. Os trabalhos de (Nishinari et al., 2004) e (Boukas et al., 2015) fizeram uma referência à modelagem de evacuação de pedestres e robótica, mas não a utilizaram no contexto da resolução de problemas ligados à tarefas no campo de robótica como no presente trabalho.

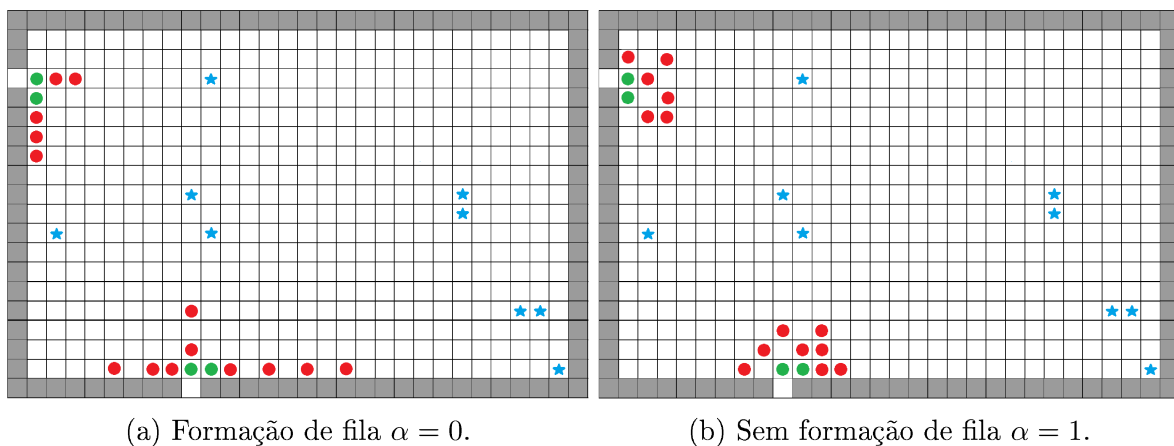


Figura 5.9: Os robôs em vermelho estão carregando um objeto enquanto que os robôs em verde acabaram de depositar um objeto, os símbolos (★) representam os focos de comida.

A resolução de conflitos adotada neste trabalho é não-determinística e considera que cada robô deve ocupar uma única célula do campo e não sobrepor os obstáculos, essa resolução foi inspirada do trabalho (Varas et al., 2007). No entanto, neste trabalho não foi adotado o parâmetro de pânico, utilizado no trabalho de (Varas et al., 2007).

## 5.2 Modelo de coleta seletiva de objetos

Esta seção descreve um modelo para o controle de um time de robôs projetado para a tarefa de coleta seletiva denominado SCAAM (Stochastic Cellular Automata Ant Memory). Mais especificamente, nessa tarefa os robôs devem coletar objetos distribuídos em um ambiente separando-os por tipo e colocando-os em depósitos específicos. Os robôs podem operar em um ambiente que pode conter obstáculos. O modelo também é baseado em comportamentos individuais simples, que combinados podem resultar num time cooperativo. Este modelo também pode ser descrito em dois níveis: nível do ambiente e nível individual do robô. O modelo do robô pode ser representado pelo autômato finito determinístico de cinco estados apresentado na Figura 5.10. Cada estado será detalhado nesta seção. Inicialmente, cada robô permanece no estado *searching* até que ele encontre algum



objeto. Ao encontrar um objeto ele faz uma detecção do tipo de objeto que ele encontrou no estado *selecting*. Após a detecção do tipo de objeto, o robô coleta e leva o objeto no depósito específico, transitando pelos estados *grabbing*, *homing* e *depositing*. A primeira abordagem considerada é quando os robôs são homogêneos, ou seja, quando todos têm o mesmo controlador e estão aptos a capturarem qualquer tipo de objeto observado no AFD da Figura 5.10. Para tornar o time heterogêneo, alguns ajustes são requeridos e o AFD é mostrado na Figura 5.11. Neste caso, uma transição de quando o robô identifica um objeto que não é do seu tipo, ele volta a procurar por outros objetos. O time de robôs conhece o ambiente previamente. Quando todos os objetos são depositados o processo é finalizado.

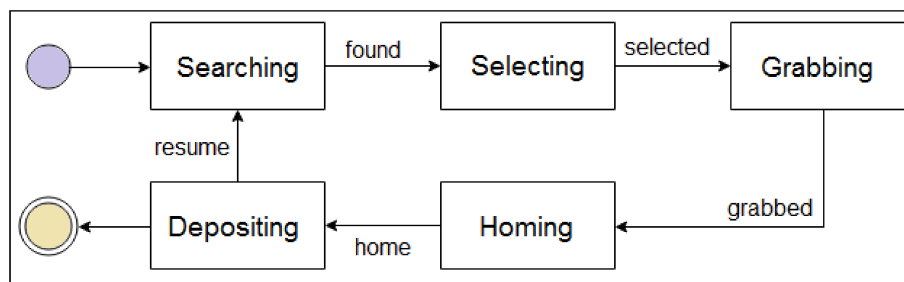


Figura 5.10: Autômato finito que representa o controlador da tarefa de coleta seletiva para um time homogêneo.

Com relação ao modelo do ambiente, assim como no modelo forrageamento, a estrutura predial para esta tarefa é uma abstração 2D da planta real. As grades de objetos, robôs e obstáculos são diferentes e sobrepostas para a execução do modelo. Se o modelo é executado com robôs homogêneos, uma única grade de piso de busca é necessária para o controle. No entanto, se o time de robôs é heterogêneo, então, a quantidade de campos de busca é a mesma quantidade de tipos de robôs. O número de grades de campo de piso de depósitos corresponde sempre à quantidade de tipos de objetos existentes no ambiente, tanto no modelo heterogêneo, quanto no homogêneo.

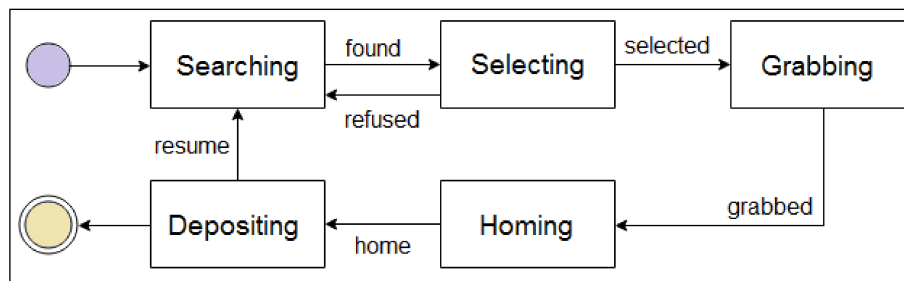


Figura 5.11: Autômato finito que representa o controlador da tarefa de coleta seletiva para um time heterogêneo.

### Coleta seletiva com time homogêneo

Na Figura 5.12 é apresentada uma ideia do comportamento de um time de robôs homogêneo durante uma simulação da tarefa de coleta seletiva com  $N = 3$  robôs (círculos).

Nessa figura, além dos robôs, existem 10 objetos do tipo “1” (estrelas azul claro) e 11 objetos do tipo “2” (estrelas azul escuro). Nesse cenário, os parâmetros serão  $N = 3$ ,  $F_1 = 10$  e  $F_2 = 11$ . O ambiente apresenta dois depósitos e como o time é homogêneo qualquer robô pode encontrar e pegar qualquer tipo de objeto. Objetos do tipo “1” são colocados em depósitos do tipo “1” e objetos do tipo “2” são colocados em depósitos do tipo “2”. A abertura mais ao norte é o depósito “1” e a abertura ao sul é o depósito 2. Inicialmente, conforme pode ser observado na Figura 5.12a todos os robôs estão no estado *searching* (estado vermelho). Logo depois, na Figura 5.12b, o robô em rosa está no processo de identificação do objeto.

Em seguida, conforme pode ser observado na Figura 5.12c, o robô percebe que o objeto é do tipo “2” e captura o objeto mudando para o estado roxo (*grabbing*). Posteriormente, o robô leva o objeto para o depósito “2” (Figura 5.12d) mudando para o estado *homing* (cor verde). Em seguida, o objeto é colocado no depósito (Figura 5.12e) no estado *depositing* (cor amarela) ao mesmo tempo que outro robô está no processo de identificação de um objeto. Logo depois, conforme pode ser observado na Figura 5.12f, o robô que estava no estado amarelo volta para o estado *searching* (vermelho) enquanto que o robô no estado de identificação captura o objeto do tipo “1” e muda de rosa para o estado roxo (*grabbing*). No final, o robô que estava no estado de *grabbing* vai para o estado *homing*, se deslocando para o ninho correspondente ao tipo de objeto ao qual ele está carregando, conforme ilustrado na Figura 5.12g e na Figura 5.12h.

### Coleta seletiva com time heterogêneo

Na Figura 5.13 é apresentada uma ideia do comportamento de um time de robôs heterogêneo durante uma simulação da tarefa de coleta seletiva com 2 robôs do tipo “1” (vermelho claro) e 1 robô do tipo “2” (vermelho escuro), e 10 objetos do tipo “1” (estrelas azul claro), e 11 objetos do tipo “2” (estrelas azul escuro). Neste caso, temos  $N_1 = 2$ ,  $N_2 = 1$ ,  $F_1 = 10$  e  $F_2 = 11$ . Os robôs vermelhos claros capturam objeto azul claro, enquanto que os robôs vermelhos escuros capturam objeto azul escuro. O ambiente apresenta dois depósitos e como o time é heterogêneo, os robôs pegam apenas objetos do tipo correspondente à sua especificação. Objetos do tipo “1” são colocados em depósitos do tipo “1” (ao norte) e objetos do tipo “2” são colocados em depósitos do tipo “2” (ao sul). Inicialmente, conforme pode ser observado na Figura 5.13a, todos os robôs estão no estado de busca (estado vermelho claro ou vermelho escuro). Em seguida o robô vermelho claro muda para o estado de processo de identificação do objeto e muda para o estado rosa, conforme está ilustrado na Figura 5.13b.

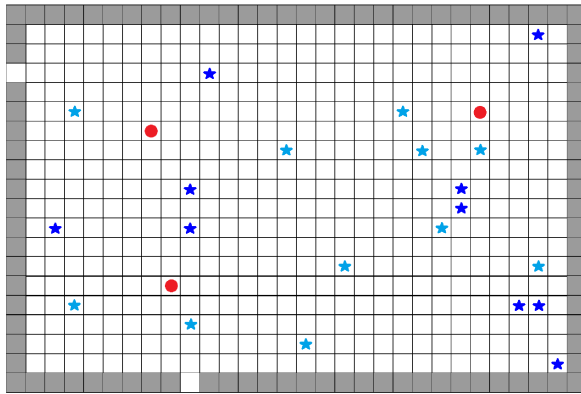
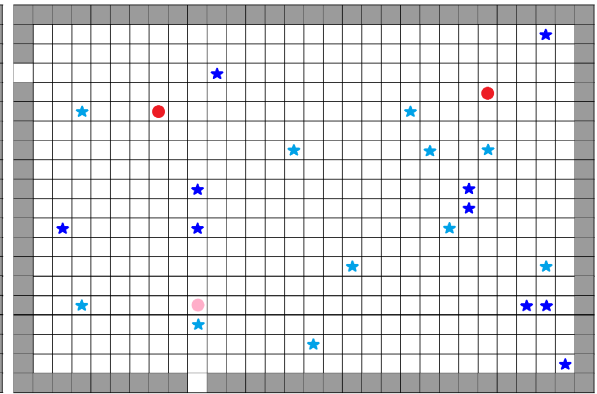
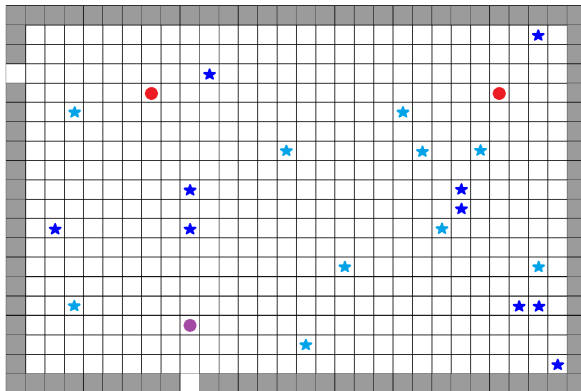
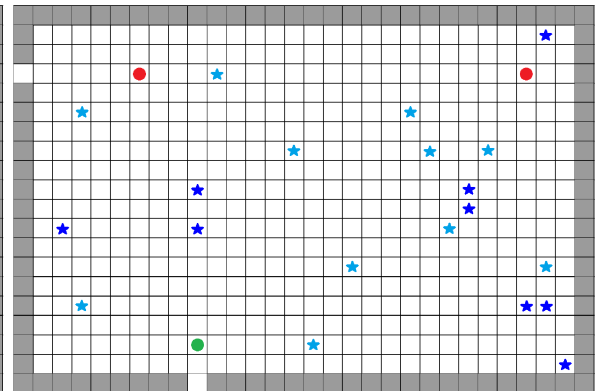
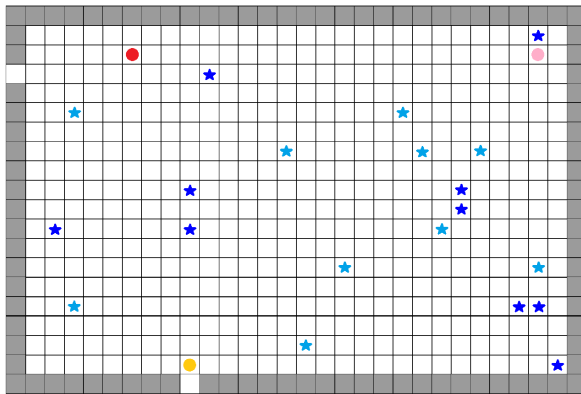
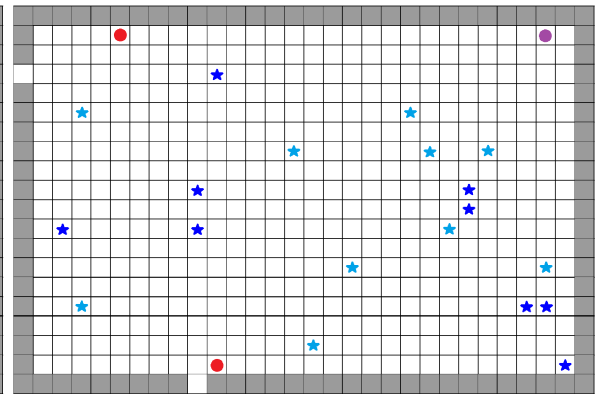
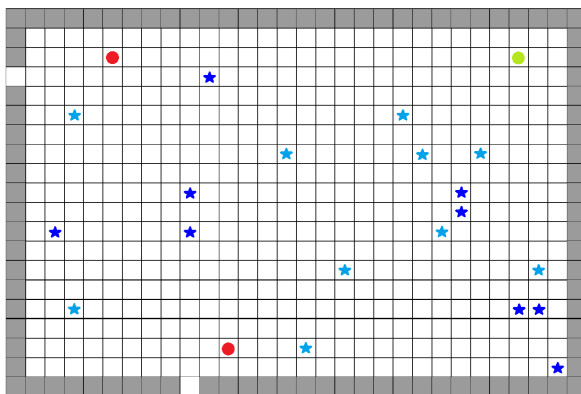
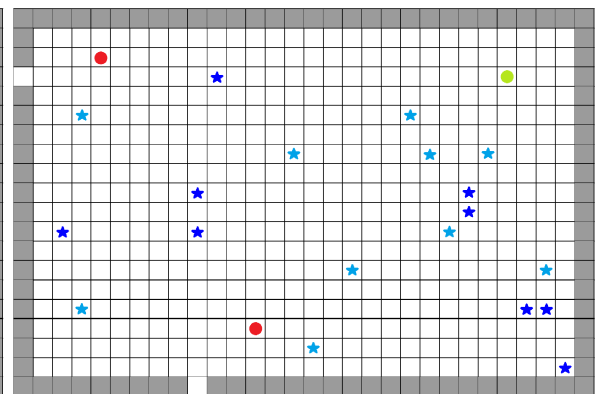
(a) Exemplo searching  $t = 0$ .(b) Exemplo selecting  $t = 1$ .(c) Exemplo grabbing  $t = 2$ .(d) Exemplo homing  $t = 3$ .(e) Exemplo depositing  $t = 4$ .(f) Exemplo grabbing  $t = 5$ .(g) Exemplo homing  $t = 6$ .(h) Exemplo homing  $t = 7$ .

Figura 5.12: Construção do ambiente de simulação com obstáculos e dois ninhos para a execução da coleta seletiva pelo time de robôs e um conjunto de alimentos (em azul).

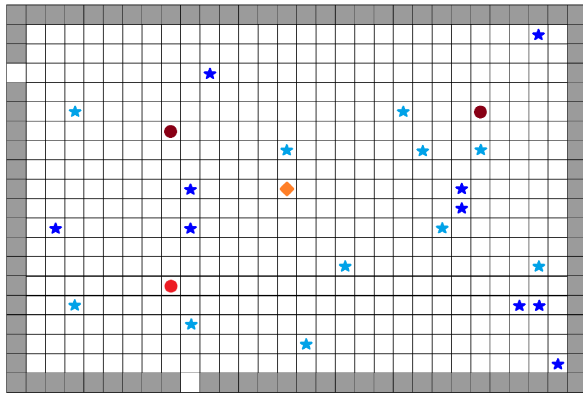
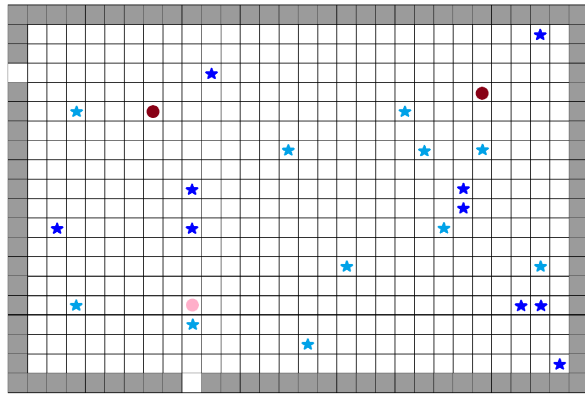
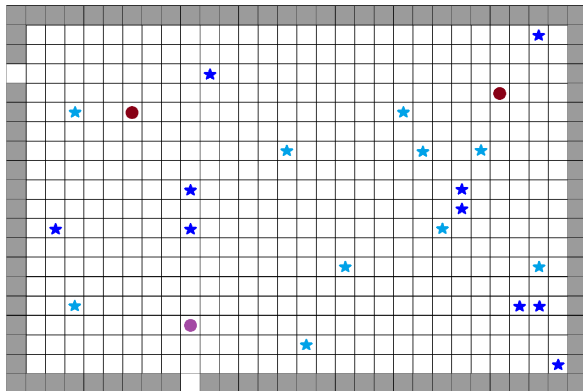
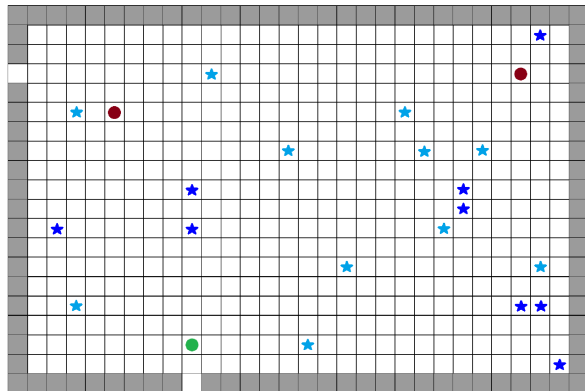
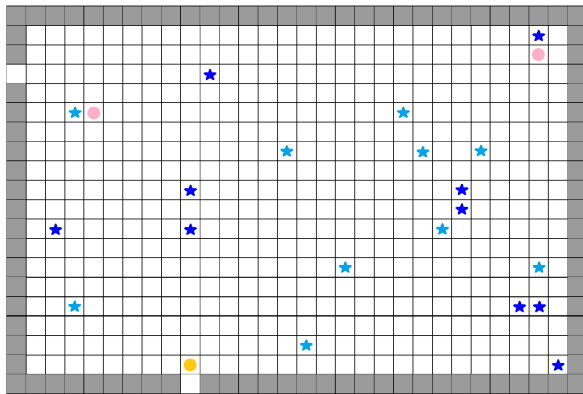
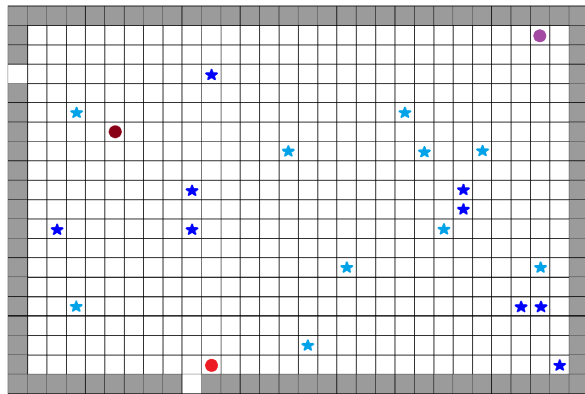
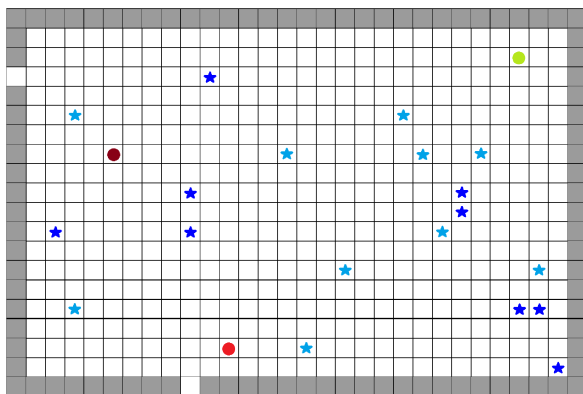
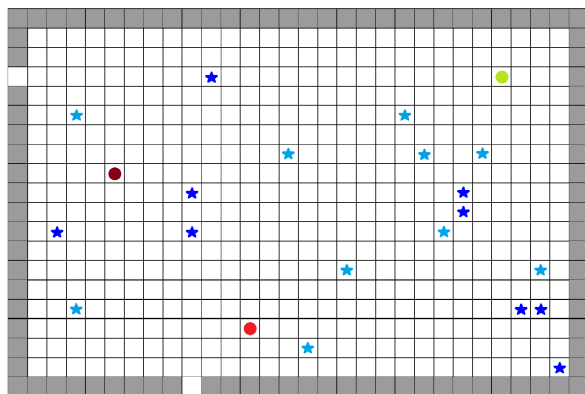
(a) Exemplo searching  $t = 0$ .(b) Exemplo selecting  $t = 1$ .(c) Exemplo grabbing  $t = 2$ .(d) Exemplo homing  $t = 3$ .(e) Exemplo depositing  $t = 4$ .(f) Exemplo grabbing  $t = 5$ .(g) Exemplo homing  $t = 6$ .(h) Exemplo homing  $t = 7$ .

Figura 5.13: Construção do ambiente de simulação com obstáculos e dois ninhos para a execução da coleta seletiva pelo enxame de robôs e um conjunto de alimentos (em azul).

Logo após, conforme pode ser observado na Figura 5.13c, o robô percebe que o objeto é do tipo “2” e captura o objeto mudando para o estado roxo. Posteriormente o robô verde (estado *homimg*) leva o objeto para o depósito “2” (Figura 5.13d). Em seguida, na Figura 5.13e o objeto é colocado no depósito (estado *depositing*) ao mesmo tempo que outros dois robôs vermelhos escuros estão no processo de identificação de objetos. Logo depois, conforme pode ser observado na Figura 5.13f, o robô que estava no estado amarelo volta para o estado de busca enquanto que o robô mais à direita no estado de identificação captura o objeto do tipo “1”. O robô mais à esquerda identifica um objeto do tipo “2”, no entanto, ele é um robô de tipo “1”, portanto o objeto não é capturado por ele. Nesse caso, o robô aumenta o valor do feromônio do tipo “1”, mas não afeta o valor do feromônio do tipo “2”. No final, o robô que estava no estado de captura vai para o estado depósito no ninho correspondente ao tipo de objeto ao qual ele está carregando, conforme ilustrado na Figura 5.13g e na Figura 5.13h.

### 5.2.1 Busca na coleta seletiva

O processo de busca na tarefa de coleta seletiva é similar ao processo de busca apresentado para a tarefa de forrageamento. A posição dos ninhos e a geometria do ambiente são determinadas e conhecidas pelo enxame de robôs. A comunicação pelos agentes é indireta, ou seja, os mesmos não trocam mensagens, mas compartilham um único ambiente geral. Cada robô comunica-se com os demais robôs do enxame de maneira indireta através da estigmergia, utilizando-se do feromônio. O feromônio, que está representado em cada célula do campo de piso de busca, é compartilhado globalmente por todos os robôs de um mesmo tipo do enxame.

Neste caso, a busca é global, no entanto podemos ter um time de robôs homogêneo ou heterogêneo. Quando o time é **homogêneo** um único campo de feromônio é necessário para a execução da tarefa, similar ao que ao modelo da tarefa de forrageamento. Por outro lado, quando o time de robôs é **heterogêneo** são necessários mais de um campo de piso de feromônio. Isso significa que não é importante para os demais robôs saberem onde estão os objetos do outro time de robôs. A cada célula é atribuído um valor de constante que representa o feromônio que cada robô deposita durante o processo de busca. Valores mais baixos correspondem às células que são menos visitadas pelos robôs, por outro lado, valores mais altos representam as células que foram mais visitadas ou tangenciadas pelos robôs. Cada célula  $x$  do campo de busca é atribuído o valor 0 no primeiro instante de tempo  $t = 0$ .

Também nesta tarefa cada robô pode ou não ter um campo de visão na localização dos alimentos, que simula o poder de precisão dos sensores ou da câmera do robô. Neste trabalho, foram considerados apenas o tipo de campo de visão definido pelo parâmetro  $r_v$  (raio de visão). Um robô é capaz de encontrar um alimento dentro do seu raio de

visão que é aplicado à vizinhança da célula central à qual o robô encontra-se naquele instante de tempo. Esse comportamento modela a capacidade do robô usar os sensores para identificar obstáculos em células adjacentes a ele.

Diferentemente do modelo CAAM para a tarefa de forrageamento, a transição de um robô de uma célula para outra é um modelo de probabilístico, ou seja, o robô seleciona com uma probabilidade não nula uma célula dentro do seu raio de vizinhança, sendo que as células com valores mais baixos têm maior probabilidade de serem selecionadas por terem sido menos visitadas tanto no estado de *searching* quanto no estado de *homing*. Portanto, o modelo é puramente estocástico e é denominado SCAAM (Stochastic Cellular Automata Ant Memory). Os conflitos são resolvidos de maneira não-determinística, conforme foi apresentado na seção 5.1.5, pois acredita-se que o não-determinismo na tarefa de busca pode prover maior cobertura do time de robôs, por permitir um maior movimento entre a vizinhança da célula vizinha à qual o robô encontra-se inserido.

A equação que representa a função de **probabilidade decisão** de escolha da célula do campo de piso de busca de feromônio  $P(x_{ij})$  é mostrado na Equação 5.9. A escolha é probabilista elitista, pois apresenta uma melhor configuração que o processo puramente de primeira escolha proposto no algoritmo CAAM. Para um dado robô, as células  $x_{ij}$  que possuem a menor quantidade de passos dentro da vizinhança em  $\eta_{x_{ij}}^m$  recebem o valor  $c_{\max}$ , enquanto que as demais recebem um valor  $c_{\min}$ . Todas as células da vizinhança  $\eta_{x_{ij}}^m$  são somadas resultando num total de  $\sum_{ij=0}^m c_{ij}$  e as células de menor valor tem uma chance  $c_{\max}$  dentro do total, enquanto que as menores tem uma chance  $c_{\min}$  de serem escolhidas. Após o processo de decisão de células, o robô **movimenta-se** para outra célula  $x_{ij}$  na sua vizinhança.

$$P(x_{ij})^{t+1} = \begin{cases} \frac{c_{\max}}{\sum_{ij=0}^m c_{ij}} & \leftarrow x_{ij}^t \leq \forall(x_{ij}) \in \eta_{x_{ij}}^m \\ \frac{c_{\min}}{\sum_{ij=0}^m c_{ij}} & \leftarrow x_{ij}^t > \forall(x_{ij}) \in \eta_{x_{ij}}^m \end{cases} \quad (5.9)$$

Assim como no modelo CAAM, no modelo SCAAM proposto para a tarefa de coleta seletiva a fim de reduzir a repetição de caminhos já percorridos, a memória finita  $Q$  também foi utilizada para otimizar a tarefa. Quando o time é **homogêneo**, apenas um campo de piso é utilizado. Assim, o mesmo campo de piso de busca é compartilhado por todos os robôs, cabendo a eles apenas identificar o tipo de objeto e o ninho. Por outro lado, quando time é **heterogêneo**, o campo de busca de cada tipo de robô, não interfere no campo de piso de busca dos outros tipos de robôs, bem como na atualização de cada campo de busca.

Uma alteração foi no processo de controle do feromônio das células inserida para a tarefa de coleta seletiva. Esse estímulo engloba a célula visitada e suas adjacentes, considerando-se um raio  $r_p$  do AC que está sendo considerado na vizinhança (difusão) (Schadschneider, 2001b). O feromônio perde sua força com o passar do tempo (declínio).

Inicialmente, todas as células do campo de piso estão configuradas com o valor 0. Com o passar do tempo o feromônio vai sendo depositado através de uma constante  $\delta$  e  $\delta'$  para as células vizinhas à célula central. Cada célula que é visitada por um robô é acrescida de uma constante de difusão  $\delta$  e de uma constante de difusão na vizinhança  $\delta'$ , que varia em função do tempo  $t$ , de acordo com a seguinte Equação 5.10.

$$x^{t+1} = \begin{cases} x^t + \delta & \leftarrow \delta = 1 \\ x^t + \delta' & \leftarrow \delta' = (1 - \frac{\tanh(\frac{t-a}{b})}{c}) \end{cases} \quad (5.10)$$

O declínio, neste caso também é realizado através de uma constante  $\beta$ , como apresentado anteriormente para o modelo CAAM. Entretanto, no modelo da coleta seletiva a quantidade de feromônio que existia previamente na célula é levado em consideração. Inicialmente em  $t = 0$ , todas as células recebem um valor inicial de feromônio igual a zero  $x^t = 0$ . Cada célula é decrementada por uma parcela resultante da função de declínio  $\beta$  de acordo com a seguinte Equação 5.11.

$$x^{t+1} = x^t - \beta, \quad \beta = (1 - \frac{\tanh(\frac{t-a}{b})}{c}) \quad (5.11)$$

### 5.2.2 Identificação na coleta seletiva

O processo de identificação na coleta seletiva refere-se à identificação do tipo de objeto que está sendo investigado. A habilidade de detectar a presença de objetos pode ser realizada através dos sensores ou da câmera do robô. No entanto, identificar objetos no ambiente consiste em após a sua detecção descobrir qual é o tipo de objeto que foi encontrado. Portanto, é importante que a mesma receba um estado particular pois exigirá um processamento mais elaborado. Esse tipo de processamento é importante se os agentes precisam executar tarefas eficientes em ambientes estruturados, como casas, escritórios ou hospitais. Se muitos tipos e formas de objetos precisam ser identificados, então um classificador mais sofisticado elaborado a partir de uma base de treinamento precisa ser elaborado para o reconhecimento de imagens. Neste trabalho será utilizado apenas uma identificação da cor do objeto, pois todos terão o mesmo formato.

### 5.2.3 Captura na coleta seletiva

O estados de captura no modelo da tarefa de coleta seletiva tem a mesma funcionalidade utilizada na tarefa de forrageamento. Após a identificação do objeto o robô passa para o estado denominado de processo de captura, sendo que o robô pode visualizar o objeto em um raio de visão  $r_v$  e capturar o objeto em  $T = r_v$  passos de tempo, quando não há conflitos. Neste caso, o robô que visualizou o objeto previamente possui prioridade em relação aos demais robôs e também em relação ao próprio campo de piso. Ou seja, uma vez que um robô identifica um objeto, ele não considera o valor de campo de piso para

realizar o movimento, e se dirige diretamente para a célula onde o alimento foi detectado.

#### 5.2.4 Regresso na coleta seletiva

O processo de regresso (*homing*) do SCAAM é baseado no modelo PCAAM para o forrageamento. Entretanto, para o caso da coleta seletiva, para cada tipo de objeto existe um tipo de depósito diferente. Assim, para tornar o processo viável, no caso do time homogêneo cada robô tem o campo de piso estático de cada tipo de depósito armazenado em sua memória. Assim que um robô captura um objeto ele sabe qual mapa deve percorrer até chegar no depósito desejado. Todo o cálculo de cada campo de piso é baseado no modelo de (Varas et al., 2007). Para a realização do movimento, escolhe-se aleatoriamente uma célula do campo de piso usando o modelo, ou seja, baseado no modelo de PCAAM. Os conflitos são resolvidos de maneira não determinística, conforme descrito na Seção 5.1.5. A equação que representa a função de **probabilidade decisão** de escolha da célula do campo de piso de estático de feromônio  $P(x_{ij})$  é mostrado na Equação 5.12. A escolha é probabilista elitista assim como no *searching* e no regresso do modelo PCAAM para o forrageamento. Para um dado robô, as células  $x_{ij}$  que possuem a menor quantidade de passos dentro da vizinhança em  $\eta_{x_{ij}}^m$  recebem o valor  $c_{\max}$ , enquanto que as demais recebem um valor  $c_{\min}$ . Todas as células da vizinhança  $\eta_{x_{ij}}^m$  são somadas resultando num total de  $\sum_{ij=0}^m c_{ij}$  e as células de menor valor tem uma chance  $c_{\max}$  dentro do total, enquanto que as menores tem uma chance  $c_{\min}$  de serem escolhidas. Após o processo de decisão de células, o robô **movimenta-se** para outra célula na sua vizinhança.

$$P(x_{ij})^{t+1} = \begin{cases} \frac{c_{\max}}{\sum_{ij=0}^m c_{ij}} & \leftarrow x_{ij}^t \leq \forall (x_{ij}) \in \eta_{x_{ij}}^m \\ \frac{c_{\min}}{\sum_{ij=0}^m c_{ij}} & \leftarrow x_{ij}^t > \forall (x_{ij}) \in \eta_{x_{ij}}^m \end{cases} \quad (5.12)$$

Adicionalmente, cada robô move-se para o seu campo de piso que corresponde ao ninho do tipo de objeto que ele coletou, conforme descrito na Equação 5.13. Esse processo de dividir os robôs para cada um dos ninhos melhora automaticamente a distribuição dos robôs de forma igualitária pelos ninhos. Dessa forma, evita-se a formação de filas perto dos ninhos. Por outro lado, podem ser criados congestionamentos, dependendo da grande quantidade sucessiva de um mesmo tipo de objeto que pode ser coletado por diferentes robôs em tempos similares.

$$W_i(x) = \min\{W_i^A(x) \mid A \text{ é um ninho do mesmo tipo do objeto coletado pelo robô } \in x\}. \quad (5.13)$$



### 5.2.5 Depósito na coleta seletiva

Após o estado de *homing* o robô para para o estado denominado *depositing*, nesta tarefa, também é o último estado que o robô permanece. Quando o robô chega ao depósito e está carregando um objeto, o robô deixa o objeto no depósito e volta para procurar mais comida. Se já não existe mais objetos no ambiente e não existe mais nenhum robô no estado *homing*, então a tarefa é finalizada.

## 5.3 Modelo de busca e resgate

Esta seção descreve um time de robôs projetado para a tarefa de busca e resgate em um ambiente com incêndio. Portanto, a tarefa desses robôs é coletar objetos dentro de um ambiente inóspito ao trânsito humano. O modelo para busca e resgate desenvolvido nesta tese de doutorado foi denominado FSACAAM (Fire Stochastic Cellular Automata Ant Memory Model). Este modelo pode ser descrito pelo autômato finito determinístico de seis estados apresentado na Figura 5.14. Cada estado será detalhado nesta seção. No instante inicial, o time de robôs é colocado no estado de busca ao mesmo tempo que um foco de incêndio é disparado no ambiente. O Foco de incêndio se alastra pelo ambiente através de uma função hiperbólica, modelado a partir de uma regra de transição de um AC. O fogo funciona como um obstáculo para o time de robôs. Ao mesmo tempo o enxame de robôs busca por objetos. Ao encontrar algum objeto ele o carrega e se desloca até um depósito, onde o mesmo é deixado. Os robôs e objetos não são afetados fisicamente pelo fogo. Os robôs são homogêneos, ou seja todos têm o mesmo controlador e, portanto, estão aptos a capturar qualquer tipo de objeto e colocá-lo em qualquer depósito. Quando todos os objetos são depositados o processo é finalizado.

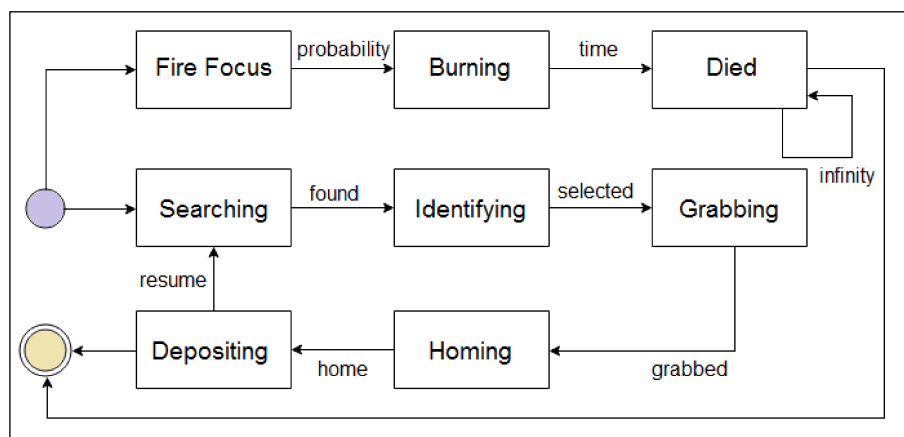


Figura 5.14: Autômato finito que representa o controlador da tarefa de busca e resgate.

Na Figura 5.15, é apresentado um exemplo de espalhamento de fogo durante a tarefa de busca e resgate de objetos, onde três robôs homogêneos foram utilizados para realizar a tarefa. Na Figura 5.15b são apresentadas as posições iniciais do robô e um foco de

incêndio com chama nova (estado laranja) na parte central do reticulado. Em seguida, na Figura 5.15b é apresentado o resultado do espalhamento inicial do fogo, sendo que este espalhamento é obtido a partir de funções probabilísticas. Adicionalmente, o robô mais ao sul entra nos estados de identificação e captura do um objeto, enquanto que os demais estão procurando por outros objetos. Na Figura 5.15c, o robô mais ao sul captura o objeto que estava dentro do seu raio de visão e o fogo se espalha no ambiente, com chamas novas (laranja) e chamas velhas (amarelas). Na Figura 5.15d, o robô ao sul no estado *homing*, os demais permanecem no estado *searching* e algumas chamas mortas (cinzas), surgem como resultado do espalhamento e evolução do fogo. Em seguida, na Figura 5.15e o robô mais ao sul deixa o seu objeto dentro do depósito e os outros dois robôs identificam novos objetos entrando no estado *identifying*. Enquanto isso, o fogo se espalha pelo ambiente e algumas chamas morrem. Por fim, na Figura 5.15f, o robô mais ao sul volta para estado *searching* e os outros dois robôs carregam os objetos, entrando no estado *grabbing*. Nessa figura é possível observar o efeito do espalhamento e da evolução do fogo após 5 passos de tempo a partir do foco inicializado em  $t = 0$ . Vemos que 35 células do reticulado já foram afetadas, sendo 20 chamas novas, 5 chamas antigas e 11 cinzas.

### 5.3.1 Modelagem de propagação do fogo

A modelagem da propagação do incêndio proposta neste trabalho é baseada em autómatos celulares. No trabalho precursor (Zheng et al., 2011) a dinâmica populacional é baseada em ACs, mas utiliza uma distribuição coordenada para a simulação do fogo. O modelo FSCAAM é executado no ambiente de simulação e considera que todas as células são homogêneas, ou seja, todas têm a mesma probabilidade de queimar. O modelo é híbrido, porque considera uma regra de transição que é probabilística e temporal (Dilão, 1993). Inicialmente todas as células estão intactas, ou seja, não há nenhum foco de incêndio no ambiente de simulação. As células que não estão no estado queimando, não apresentam diferença na coloração de suas células. Uma célula no estado laranja ou amarelo são células no estado queimando, e a diferença entre as duas cores se dá pela quantidade de tempo transcorrido entre um estado e outro. Inicialmente, todas as células recebem um valor inicial de zero “0” (célula sem coloração). Uma célula do ambiente é sorteada para ser o foco de incêndio no estado queimando do tipo “1” (cor laranja). A cor cinza representa células que já queimaram. A seguir é apresentada a regra de atualização de propagação de incêndio dentro do ambiente de simulação.

1. Cada célula tem uma probabilidade não nula  $P(x)$  de começar a queimar caso tenha uma célula adjacente no estado queimando ou estado morto, tal que  $P(x) = \frac{1}{m} \times (0.7\varphi + 0.3\varepsilon) \times \frac{\sec(\frac{t-a}{b})}{c}$ , sendo  $\varphi$  e  $\varepsilon$  a quantidade de células queimando e mortas, respectivamente, ao redor da célula central,  $a$ ,  $b$  e  $c$  são constantes de inclinação da curva da secante hiperbólica,  $t$  representa o tempo da iteração corrente e  $m$

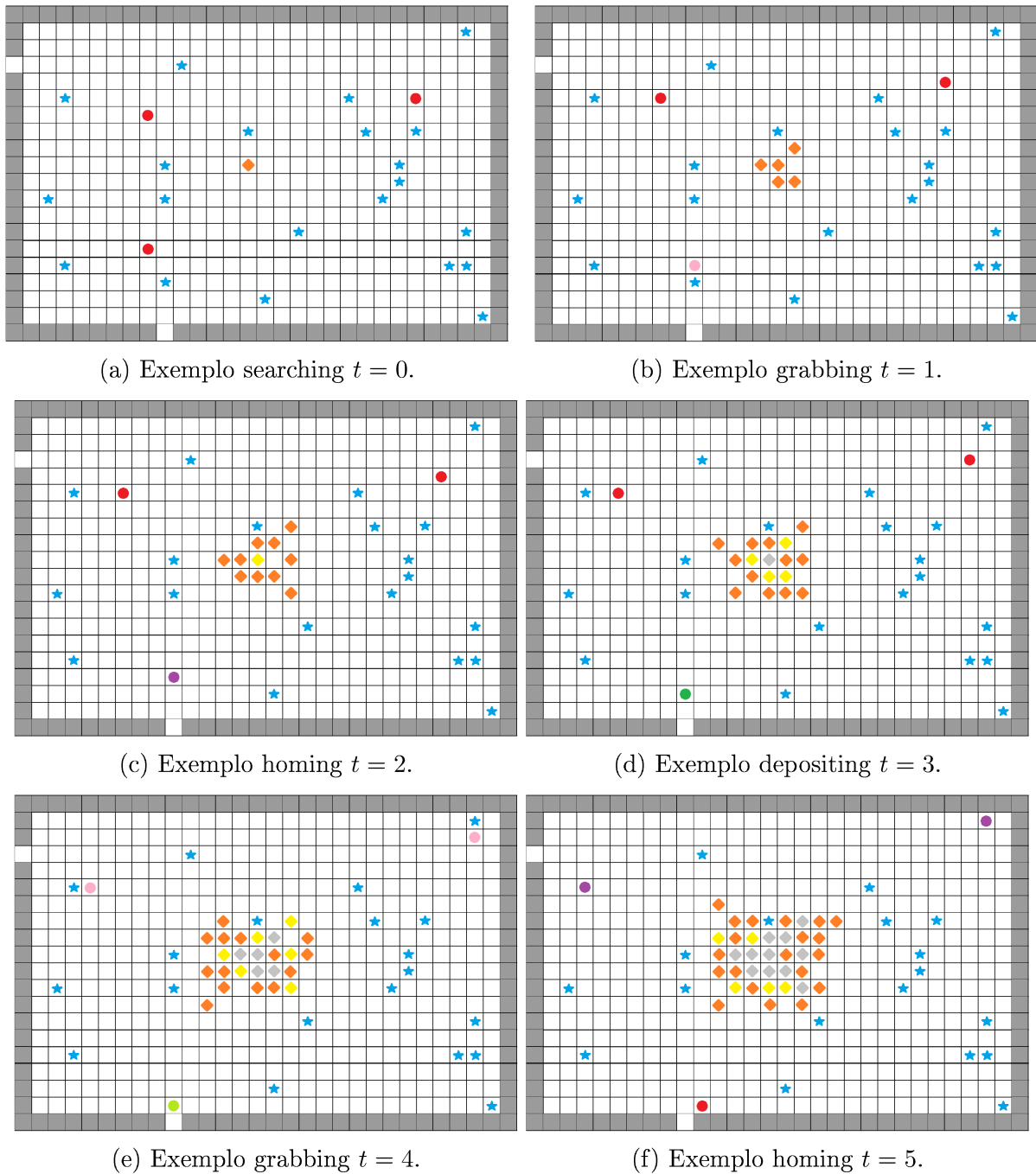


Figura 5.15: Construção do ambiente de simulação com obstáculos e dois ninhos para a execução da tarefa de forrageamento pelo enxame de robôs (vermelho, roxo, verde ou rosa) e um conjunto de alimentos (em azul) que deve ser procurado e depositado no ninho. Os losangos laranja, amarelo e cinza claro estão apresentados os estados que a chama do incêndio dentro do ambiente de simulação modelado através de um autômato celular pode alcançar.

representa o tamanho da vizinhança bidimensional  $m = (2r_f + 1)^2$ , sendo que  $r_f$  representa o raio de espalhamento do fogo.

2. Uma célula permanece no estado queimando do tipo “1” (cor laranja) por  $T_1$  passos de tempo e muda para o estado queimando do tipo “2” (cor amarela).

3. Uma célula permanece no estado queimando do tipo “2” (cor amarela) por  $T_2$  passos de tempo e muda para o estado morta (cor cinza).
4. Uma célula permanece no estado morta (cor cinza) por tempo indefinido.

De acordo com (Rothermel, 1972) a intensidade das chamas é fator de exponencial agravamento para alteração da coloração e propagação das chamas. A intensidade das chamas é implementada apenas para efeito visual (Hernandez et al., 2007), ou seja, a medida que a célula queima, altera-se seu estado de coloração. Neste trabalho é proposta uma modelagem da propagação de incêndio com grande probabilidade de espalhamento utilizando-se a função da secante hiperbólica apresentada na Figura 5.16.

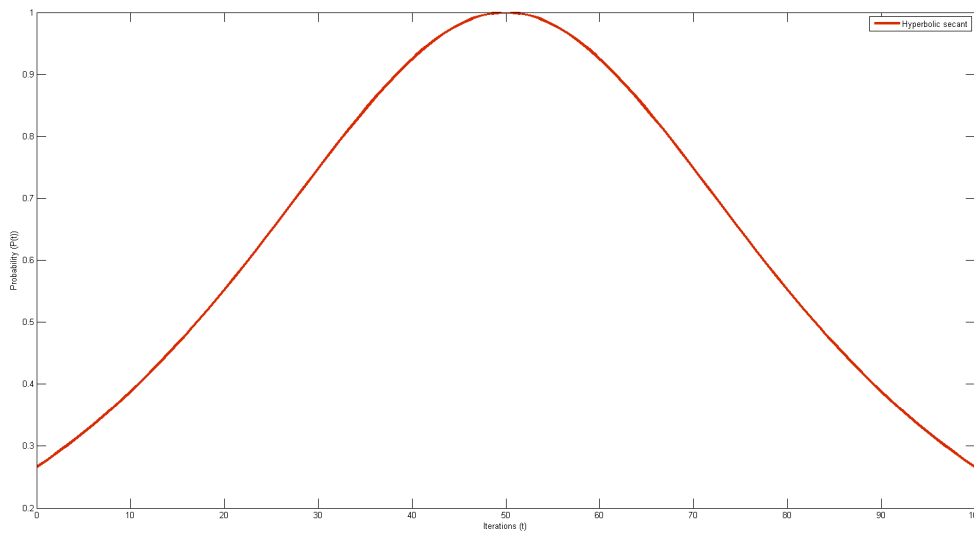


Figura 5.16: Função da secante hiperbólica.

Um modelo de espalhamento de fogos baseado em ACs foi apresentado em (Lima and Lima, 2014), que propôs a função de intensidade de espalhamento através de uma função discreta criada manualmente. Nesta tese de doutorado, a implementação foi realizada através de uma função da secante hiperbólica ou função logística. Além disso, nos trabalhos de (Lima and Lima, 2014) e (Dilão, 1993) o fator vento foi considerado, já que o espalhamento era realizado em uma floresta com árvores homogêneas e em um ambiente propício a essas condições climáticas. Neste trabalho, o ambiente de simulação é fechado e desconsidera-se o fator vento. A ideia apresentada neste trabalho é que a intensidade de poder de propagação do fogo aumenta ao longo do tempo ( $t$ ), atinge um ápice e posteriormente volta a perder força de propagação. Ou seja, inicialmente o fogo tem maior intensidade para propagar chamas e à medida que o tempo passa, essa intensidade é reduzida.

### 5.3.2 Busca na tarefa de busca e resgate

O estado *searching* para a tarefa de busca e resgate é similar ao apresentado na tarefa de coleta seletiva. Ou seja, a escolha é probabilista elitista. A equação que representa a função de **probabilidade decisão** de escolha da célula do campo de piso de busca de feromônio  $P(x_{ij})$  é mostrado na Equação 5.14. A escolha é probabilista elitista, pois apresenta uma melhor configuração que o processo puramente de primeira escolha proposto no algoritmo CAAM. Para um dado robô, as células  $x_{ij}$  que possuem a menor quantidade de passos dentro da vizinhança em  $\eta_{x_{ij}}^m$  recebem o valor  $c_{\max}$ , enquanto que as demais recebem um valor  $c_{\min}$ . Todas as células da vizinhança  $\eta_{x_{ij}}^m$  são somadas resultando num total de  $\sum_{ij=0}^m c_{ij}$  e as células de menor valor tem uma chance  $c_{\max}$  dentro do total, enquanto que as menores tem uma chance  $c_{\min}$  de serem escolhidas. Após o processo de decisão de células, o robô **movimenta-se** para outra célula na sua vizinhança.

$$P(x_{ij})^{t+1} = \begin{cases} \frac{c_{\max}}{\sum_{ij=0}^m c_{ij}} & \leftarrow x_{ij}^t \leq \forall(x_{ij}) \in \eta_{x_{ij}}^m \\ \frac{c_{\min}}{\sum_{ij=0}^m c_{ij}} & \leftarrow x_{ij}^t > \forall(x_{ij}) \in \eta_{x_{ij}}^m \end{cases} \quad (5.14)$$

Assim como nos modelos CAAM e SCAAM, no modelo FSCAAM proposto para a tarefa de busca e resgate a memória finita  $Q$  também foi utilizada para otimizar a tarefa, a fim de reduzir a repetição de caminhos já percorridos.

Uma alteração no algoritmo foi proposta com o intuito de apresentar uma função de depósito de feromônio que varia não somente a partir da interação de um robô e o campo de piso, mas também a partir da iteração do tempo. A função escolhida foi a função da tangente hiperbólica, apresentada na Figura 5.17. Essa função foi escolhida devido ao fato de que à medida que uma quantidade suficiente de comida é encontrada pelas formigas, elas tendem a diminuir a inserção de feromônio no ambiente (Gordon et al., 1993). Esse tipo de comportamento é importante para os robôs, pois ao final da busca acaba sendo mais difícil de encontrar alimentos, e uma busca com campo de piso uniforme pode acelerar o processo. Outras funções com comportamento similar, como a função logística, poderiam ser utilizadas. Inicialmente, todas as células recebem um valor inicial de feromônio igual a zero  $x^t = 0$ . Em seguida, é previsto que a regra de difusão e declínio de feromônio seja realizada através de uma função que inicialmente deposite muito feromônio dentro do campo de piso e posteriormente, essa taxa é decrementada, conforme descrito a seguir:

1. todas as células recebem um valor inicial de feromônio igual a zero  $x^t = 0$ .
2. cada célula que é visitada por um robô é acrescida de uma parcela resultante da função de difusão  $\delta$  de acordo com a seguinte equação  $x^{t+1} = x^t + \delta$ , sendo que  $\delta = p(x) = \frac{1 - \tanh(\frac{t-a}{b})}{c}$ ,  $a$ ,  $b$  e  $c$  são constantes de inclinação da curva da tangente hiperbólica e  $t$  representa o tempo da iteração corrente.

3. cada célula é decrementada por uma parcela resultante da função de declínio  $\beta$  de acordo com a seguinte equação  $x^{t+1} = x^t - \beta$ , sendo que  $\beta = \frac{1 - \tanh(\frac{t-a}{b})}{c}$ ,  $a$ ,  $b$  e  $c$  são constantes de inclinação da curva da tangente hiperbólica e  $t$  representa o tempo da iteração corrente.

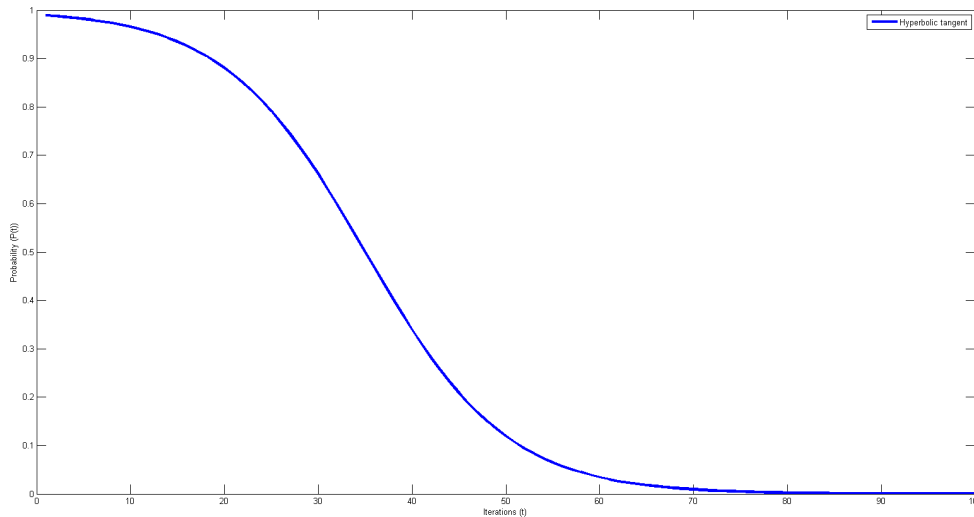


Figura 5.17: Função da tangente hiperbólica com  $a = 35$ ,  $b = 15$  e  $c = 2$ .

### 5.3.3 Identificação na tarefa de busca e resgate

O estado *identifying* prevê a diferenciação de um objeto de um foco de incêndio dentro do ambiente de simulação. Assim, quando o robô se aproxima de um objeto dentro do raio de sua visão, ele identifica através de seu sistema de câmeras, o tipo de objeto encontrado. Se o robô identificar como um objeto, então, é possível que o mesmo entre no estado de captura. Caso o robô identifique um foco de incêndio ele considera o foco como um obstáculo. Assim, o robô não é capaz de mover-se para uma célula com foco de incêndio, a menos que ele não tenha outra possibilidade.

### 5.3.4 Captura na tarefa de busca e resgate

Os estado *grabbing* no modelo da tarefa de busca e resgate tem a mesma funcionalidade utilizada na tarefa de forrageamento. Após a identificação do objeto o robô passa para o estado denominado de processo de captura o robô pode ver o objeto em um raio de visão  $r_v$  e capturar o objeto em  $T = r_v$  passos de tempo, quando não há conflitos. Neste caso, o robô que visualizou o objeto previamente possui prioridade em relação aos demais robôs e também em relação ao próprio campo de piso, ou seja, ele não considera o valor de campo de piso para realizar o movimento, quando o mesmo já percebe o alimento.

### 5.3.5 Regresso na tarefa de busca e resgate

O estado *homing* proposto para o modelo de busca e resgate é baseado na ideia de que as formigas se comunicam diretamente através do toque de suas antenas. Num ninho as formigas se cobrem de uma camada de gordura que carrega um odor específico daquela colônia (Gordon et al., 1993). Por exemplo, as formigas que ficam expostas ao ambiente possuem diferentes odores específicos de hidrocarboneto cuticulares. Por esse motivo, uma formiga é capaz de reconhecer suas companheiras de ninhos. A partir disso, as formigas usam o padrão de seus contatos de antenas à medida que encontra formigas de outras tarefas, para decidir o que fazer. Esse padrão é a mensagem trocada por elas. Diante disso, é possível diferenciar, por exemplo, se uma formiga é coletora, responsável pela manutenção ou pela limpeza do ninho (Wagner et al., 2000). Além disso, de acordo com (Gordon, 2002), as formigas saem do ninho para suas várias tarefas e a maneira com que elas têm essa frequência de contatos, saem ou entram no ninho, influencia ou determina a decisão de cada formiga sobre sair ou não, e sobre qual tarefa adotar, e o padrão que deve fazer quando volta para o ninho. Outra particularidade é que embora as formigas não saibam contar precisamente, elas são capazes de diferenciar seus trabalhos e também de dividir a quantidade de formigas para cada tipo de tarefa (Gordon, 2013).

Dessa forma, neste trabalho é proposta a ideia de que quando um robô captura um objeto esse robô usa não só o campo de piso estático e o piso dinâmico para decidir o caminho até o ninho, mas também uma combinação da troca de mensagens com o sensor de proximidade do ninho. A partir dessa troca de mensagens, que pode ocorrer a cada no momento de captura do objeto, tornando o processo dinâmico, o robô pode escolher o melhor ninho. Existe um sensor em cada ninho com raio de visão  $r_n$ , e este é o responsável por avisar aos outros robôs sobre a quantidade de robôs ao redor desse ninho num raio de visão  $r_v$ . Então, um dos dois ninhos é escolhido. Quanto maior é o raio de visão  $r_n$ , melhor é a chance de saber qual é o ninho menos congestionado a ser escolhido, no entanto, se esse raio é muito grande, ele acaba fornecendo uma informação distorcida da fusão dos dois ninhos simultaneamente. Assim, a estratégia segue a mesma definição do modelo DCAAM, tal que  $\rho_{A_k}$  de cada ninho  $A$  serve para selecionar o ninho a uma certa distância  $d_{A_k}^t$  conforme a Equação 5.15.

$$\rho_{A_k}^t = \sum_{ij=0}^{A_k^v} x_{ij}^t \quad (5.15)$$

No caso em que os ninhos são igualmente lotados, o campo de piso estático  $W_{static}(x_{ij}^t)$  é usado. Deve-se lembrar que o campo de piso estático está associado ao movimento probabilístico elitista como na Equação 5.5. O agente é capaz de decidir qual é a melhor solução, ir para o ninho mais próximo  $A$  ou ir para um que não esteja congestionado como

mostrado na Equação 5.16.

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t \in W_{(x_{ij}^t)}^{A_k^t} & \leftarrow \rho_{A_k}^t < \forall \rho_{A_k}^t, k \geq 1 \\ x_{ij}^t \in \min(W_{(x_{ij}^t)}^{A_k^t}) & \leftarrow \rho_{A_k}^t = \forall \rho_{A_k}^t, k \geq 1 \end{cases} \quad (5.16)$$

A utilização desse processo de *homing* com cálculo de densidades próximo aos ninhos é de extrema importância, já que o fogo cria obstáculos virtuais para os robôs, que são capazes de identificar quando as chamas estão no estado mais abrasivo. Quando a intensidade das chamas torna-se menos intensa, os robôs são capazes de infiltrarem por entre o fogo. Sem a utilização do DCAAM no processo de *homing* do FSCAAM, a execução do algoritmo seria mais lenta, uma vez que os robôs além de encontrarem um maior nível de congestionamento próximo aos ninhos devido à má distribuição advindo do algoritmo de (Varas et al., 2007), ainda teria um outro problema que seria a resolução de conflitos entre os obstáculos móveis, que neste trabalho são as chamas em processo de combustão elevado.

### 5.3.6 Depósito na tarefa de busca e resgate

Após o estado de *homing* o robô para para o estado denominado *depositing*, nesta tarefa, também é o último estado que o robô permanece. Quando o robô chega ao depósito e está carregando um objeto, o robô deixa o objeto no depósito e volta para procurar mais comida. Se já não existe mais objetos no ambiente e não existe mais nenhum robô no estado *homing*, então a tarefa é finalizada.

### 5.3.7 Resolução de conflitos na tarefa de busca e resgate

A tarefa de busca e resgate apresenta a mesma resolução de conflitos apresentada na Seção 5.1.5. Ou seja, não somente a tarefa de busca é não determinística, mas também as regras que regem a resolução de conflitos. No entanto, um fator adicional foi adicionado ao ambiente de simulação que é o **tratamento de obstáculos móveis**. Assim, quando um robô identifica uma chama de fogo o mesmo trata essa chama como um obstáculo e então realiza um movimento de desvio. Quando a chama do incêndio morre, o robô pode então entrar nessa região para a procura de objetos.

## 5.4 Implementação do modelo no Webots

Esta seção apresenta os detalhes de implementação do modelo RCAAM (Robot Cellular Automata Ant Memory) para a tarefa de forrageamento no ambiente de simulação Webots utilizando-se o robô e-Puck. Algumas adaptações do modelo inicial foram realizadas para que fosse possível sua execução no simulador, tais como, detecção de cruzamento



e criação de obstáculos virtuais (Marchese, 1996). Além disso, um modelo para sincronização do time de robôs foi proposto neste trabalho. A descrição formal desse modelo foi detalhada e publicada em (Lima et al., 2017b).

### 5.4.1 Modelo de comunicação para sincronização do time de robôs

Para que fosse possível sincronizar, ou seja, fazer com que o enxame realizasse o movimento no mesmo instante de tempo, foi necessário compartilhar globalmente a informação entre os robôs através de um servidor e as mensagens eram enviadas através de arquivos de texto. Cada arquivo de texto contém o tempo  $t$  em que o robô se encontra e a posição na matriz. Inicialmente, cada robô envia a informação para o servidor sobre o seu tempo  $t$  e posicionamento na matriz, que representa o ambiente de simulação. Posteriormente, o servidor realiza a leitura de cada arquivo, e insere os valores numa matriz interna para calcular para onde cada robô deve realizar o seu movimento no instante de tempo  $t + 1$ . Esse servidor resolve os possíveis conflitos e também fornece num diretório compartilhado um arquivo de texto que indica onde cada robô deve realizar o movimento. No entanto, cada robô decide seu ângulo de rotação. Neste arquivo de texto fornecido pelo servidor existe uma informação sobre as próximas posições de todos os robôs e também o instante de tempo  $t + 1$ . Em seguida, cada robô realiza a leitura desse arquivo de forma paralela e independente. O robô recebendo a informação da sua posição no tempo  $t + 1$ , começa, então, a realizar o seu movimento. Ao mesmo tempo, o servidor realiza leituras contínuas nos arquivos de cada robô para então verificar se cada um deles já completou o seu movimento. Um robô só completa o seu movimento, quando ele escreve no arquivo a sua posição no instante  $t + 1$ . A partir da escrita, o servidor poderá acessar aquela informação. Por fim, o servidor acessa informação de todos os robôs no instante de tempo  $t + 1$  e calcula as novas posições de cada robô. O processo continua, até que não tenha mais nenhum robô que não se tenha depositado o seu alimento. O diagrama que representa o modelo de sincronização está ilustrado na Figura 5.18.

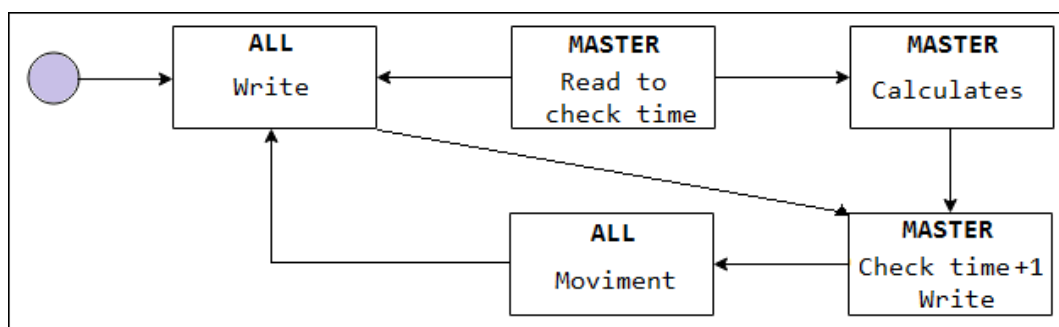


Figura 5.18: Diagrama que representa a sincronização de movimento do time de robôs (Lima et al., 2017b).

### 5.4.2 Resolução de conflitos

A resolução de conflitos foi adaptada para a implementação no ambiente de simulação Webots. Além das três condições de conflito apresentadas na Figura 5.6, que se referem aos conflitos considerando-se apenas às posições finais dos robôs, uma quarta condição foi adicionada a fim de evitar uma colisão entre dois ou mais robôs. A resolução de conflitos prevê a detecção de movimentos do tipo “X” entre dois robôs, conforme está apresentado na Figura 5.19 através dos triângulos abertos. Inicialmente, um primeiro robô é sorteado aleatoriamente e realiza o seu movimento enquanto o segundo robô espera o primeiro robô completar o seu movimento. Posteriormente, o segundo robô realiza o movimento e assim o time de robôs é sincronizados. Toda a movimentação é realizada em um único passo discreto de tempo.

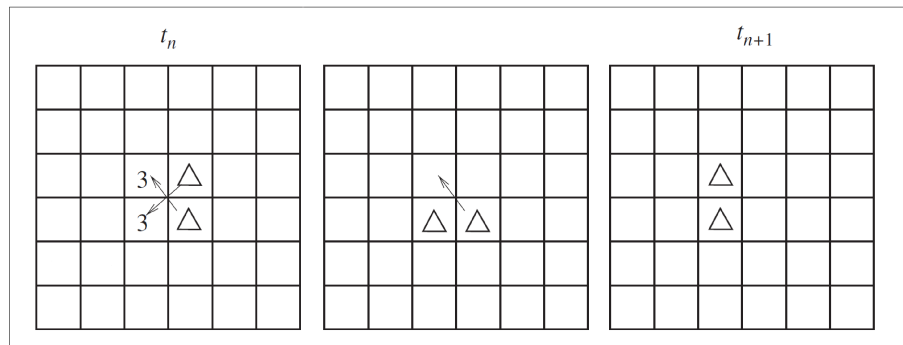


Figura 5.19: Detecção de conflitos do tipo “X” (Lima et al., 2017b).

### 5.4.3 Criação de obstáculos virtuais

O conceito de obstáculos virtuais foi apresentado no trabalho de (Marchese, 2011). Devido à grande quantidade de colisões provocadas em movimentos adjacentes a um obstáculo, a criação de obstáculos virtuais torna-se necessária quando o ambiente de simulação apresenta obstáculos internos além das paredes internas, ou mesmo se as paredes possuem quinas.

A Figura 5.20 ilustra um exemplo de colisão contra obstáculos reais. Na Figura 5.20a é apresentado um exemplo de um robô colidindo contra um obstáculo, quando deseja fazer uma movimentação diagonal. Para resolver este problema, uma técnica de duplo movimento poderia ser utilizado (Pereira et al., 2011). No entanto, essa solução poderia gerar um atraso na simulação, em especial porque todos os robôs eem estar sincronizados. Para resolver este problema de forma efetiva, a adição de obstáculos virtuais foi ser incorporada ao ambiente de simulação, fazendo com que este problema fosse resolvido. Um exemplo de obstáculos virtuais está ilustrado na Figura 5.20b.

Nesse caso, os obstáculos reais para os quais a localização é conhecida a priori são alargados de forma que o mapa do ambiente considera células adicionais como obstáculos pelos quais os robôs não devem trafegar. Esse alargamento pode ser um parâmetro do

sistema a ser ajustado de acordo com as dimensões do ambiente, do robô, da célula e da precisão. Nos experimentos realizados o alargamento utilizado foi de 1 célula em torno dos obstáculos reais.

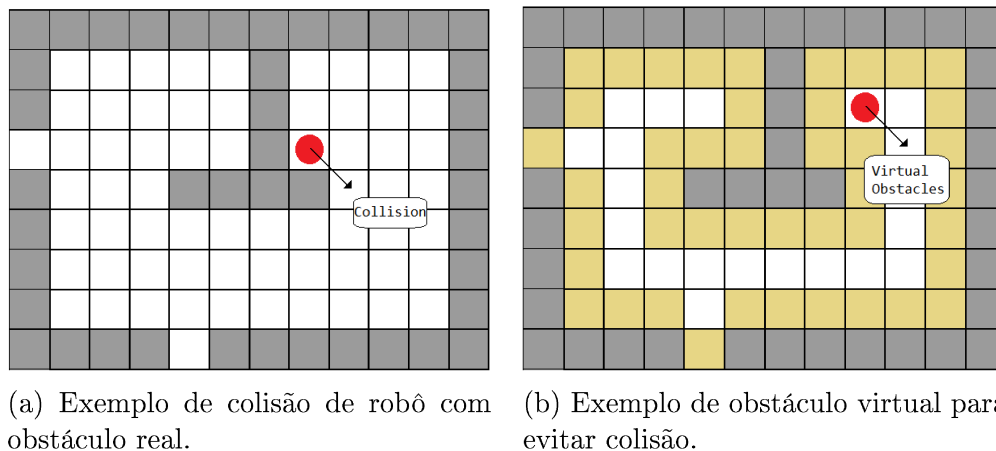


Figura 5.20: Exemplo da estrutura de obstáculos virtuais proposta em (Marchese, 2011).

#### 5.4.4 Utilização da odometria, GPS e rotação dos robôs

Como o modelo é baseado em ACs, então a sequência de movimentos do robô é discreta. Para que ele possa realizar o movimento é necessário que o robô receba um comando de rotação para uma determinada orientação, ou translação para uma nova célula. A primeira implementação no Webots do modelo RCAAM utiliza a odometria para controlar a movimentação do time de robôs. A odometria é um método de estimação de posição e orientação muito utilizado em robótica móvel. A função de odometria utilizada neste trabalho foi a mesma utilizada no trabalho de (Ferreira et al., 2014). No entanto, para este trabalho, esta função não trouxe a precisão necessária para coordenar os robôs exatamente no centro da célula. Neste trabalho, apesar do movimento de uma célula para outra ser um passo relativamente pequeno, o deslocamento final é a sequência de diversos passos. Assim, o erro acumulado prejudica a precisão do robô, pois o mesmo não centraliza na célula e pode colidir com outros robôs durante o experimento. Assim, uma segunda implementação do modelo RCAAM foi realizada usando GPS (Global Position Sensor).

O GPS é um sensor que pode obter a informação absoluta da posição a partir do controlador do programa. Este slot de tecnologia de GPS pode ser usado através de informações de satélite ou de laser. A precisão do GPS é expressa em metros. Na linguagem de programação C utilizando-se a biblioteca de GPS do Webots, o vetor retornado é um ponteiro com os valores de cada coordenada. Cada valor representa uma chamada da função (*step()*) e a mesma retorna a coordenada a cada passo de tempo. Neste modelo, esse slot foi necessário para que a acurácia da centralização na célula fosse atingida, uma vez que o modelo precisa sincronizar os passos de tempo e a centralização dentro de cada célula para impedir colisões.

Inicialmente, o time de robôs era disposto na mesma orientação, ou seja, todos os robôs ficavam num ângulo de  $0^\circ$  orientados para frente. Após realizar o movimento, cada um dos robôs do time voltava para suas posições iniciais. Ou seja, quando o robô rotacionava para fazer um movimento em diagonal, após a realização do movimento, este robô rotacionava para sua posição original. Essas rotações sequenciais em que o robô retorne à sua posição original a cada movimento realizado, provoca um aumento no tempo total da execução da tarefa.

Posteriormente, um ajuste na rotação desses robôs foi realizada para que o time alcançasse um melhor desempenho. Para isso, uma variável na estrutura do robô foi adicionada. Essa variável guarda o estado do ângulo do robô naquele momento. Assim, quando o robô rotaciona para a realização do movimento ele permanece no mesmo ângulo em que realizou o movimento, até que a próxima ação seja decidida.

O slot IMU (Inertial Measurement Unit) simula uma unidade de medição inercial. Esse nó calcula e retorna ângulos suas rotação, de passo e de guinada com respeito a um sistema de coordenadas global definida no slot da WorldInfo. Com esse slot é possível medir a uma aceleração ou a uma velocidade angular, através do acelerômetro ou giroscópio. No modelo RCAAM, este slot foi utilizado para o cálculo do erro do ângulo de guinada que cada robô comete ao realizar movimentos de giro. Assim, esse slot permitiu calcular através de um limiar de erros, e a partir desse cálculo o robô é capaz de corrigir seu ângulo de guinada, corrigindo a precisão de centralização para cada robô dentro de cada célula.

#### 5.4.5 Processamento de imagens para identificação de objetos e mundo inteligente

O processamento de imagens foi inserido no estado de **searching** para tornar a simulação com os robôs mais realista. Durante o processo de busca cada robô utiliza sua câmera para encontrar objetos ao longo do caminho. Para processar as informações capturadas pela câmera do robô a partir do seu raio de visão  $r_v$ , o robô tem que girar em  $360^\circ$  e verificar para cada célula da vizinhança de tamanho  $m$  se existe um alimento em sua visão de raio  $r_v$ . Este processo compreende o processamento de imagem ambiente. A comida é representada por um círculo azul [0 0 1] no padrão RGB, conforme apresentado na Figura 5.21. Para simplificação, qualquer outro objeto no ambiente tem a cor diferente do círculo da comida e a câmera tem a rotação voltada para o piso térreo, minimizando o tempo de processamento. O tamanho da câmera é de 1 *vezes* 14, que 14 representa o tamanho da célula de ambiente  $l$ . Se um pixel azul é detectado a partir da rotação do robô, então o robô faz um movimento e vai para o estado de *grabbing*, que consiste na única etapa que representa a pegar o objeto detectado.

Um **smart grid** inclui uma variedade de medidas operacionais e de energia, de toque,

incluindo medidores inteligentes e aparelhos inteligentes, que são capazes de medir diferenças dinâmicas com o passar do tempo. Para a realização desse controle do ambiente, foi utilizado uma rede de sensores disposta por cada uma das células do reticulado, conforme apresentado na Figura 5.21. Neste nosso trabalho, o objetivo é fazer um controle em nível do ambiente, de tal maneira que quando um robô se sobrepõe a uma célula, após verificar com sua câmera que a mesma é um foco de comida, ela desaparece do ambiente simulado. Esse controle é realizado a partir de leituras do arquivo onde são armazenadas as unidades de comida a partir do sistema geral de controle das demais variáveis do ambiente (ferômonio, obstáculos). No Webots EDU, a criação se deu a partir da criação de uma malha de sensores interconectados com o mundo através de uma matriz, cada um desses sensores é dotado de um LED no modo de luz ligada, quando o robô passa sobre esses sensores, detectado pelo arquivo de controle da variável de comida do ambiente, o LED é por sua vez desligado. Quando não há mais nenhum LED ligado no ambiente e todos os robôs estão no estado livre, então o processo de busca é finalizado e a tarefa de forrageamento também é completada.

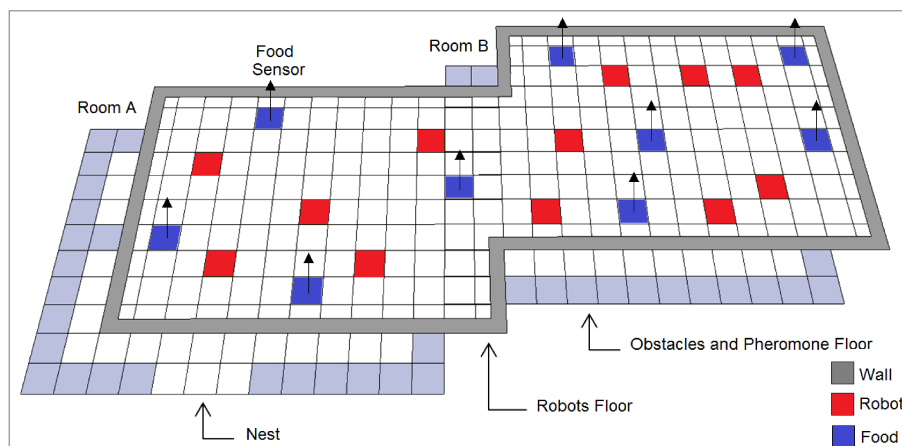


Figura 5.21: Ambiente inteligente simulação do mundo com ■ obstáculos e um ninho A, o enxame robô é representado por ■ e unidades de alimentação são representados pela ■, o sensor de comida é representado pela ↑, e cada unidade de alimentação tem apenas um 90° sensor de posição vertical (Castro and Lima, 2013a).

#### 5.4.6 Melhorias na tarefa de forrageamento do Webots

Quatro melhorias foram propostas para integrarem o modelo de forrageamento e suas respectivas implementações no Webots EDU, sendo que essas melhorias se deram nos algoritmos de *searching* e *homing*, publicadas em (Lima and Oliveira, 2016b) e em (Lima and Oliveira, 2016a). Essas melhorias contemplam utilizar não somente as abordagens de investigação de localização, sincronização, controle, processamento de imagens e utilização de mundos inteligentes, mas também as melhorias impostas nos algoritmos CAAQ (Lima and Oliveira, 2016b), PCAAM (Lima and Oliveira, 2016b), HCAAM (Lima and Oliveira,

2016a), DCAAM (Lima and Oliveira, 2016a). Na seção de experimentos, foi possível comprovar que a partir das melhorias do algoritmos teóricos, houve uma significativa melhoria no ambiente de simulação do Webots EDU e os respectivos foram denominados Robot Cellular Automata Ant Queue (RCAAQ) model, Robot Probabilistic Cellular Automata Ant Memory (RPCAAM), Robot Hybrid Cellular Automata Ant (RHCAAM) model e Robot Density Cellular Automata Ant (RDCAAM) model.

## 5.5 Comentários finais sobre os modelos propostos

Neste capítulo foram apresentados os modelos CAA, CAAM, CAAQ, PCAAM, HCAAM, SCAAM, FSCAAM e RCAAM elaborados para as três tarefas de times de robôs: forrageamento, coleta seletiva, busca e resgate, e a adaptação do forrageamento para a implementação em robôs reais. Adicionalmente, as definições de implementação de cada um dos três modelos foram apresentadas. Além disso, foi apresentado um novo modelo para a simulação de propagação de incêndios em ambientes fechados através da técnica de autômatos celulares. Por fim foram apresentadas as principais adaptações que devem ser realizadas para a implementação do modelo RCAAM no ambiente de simulação Webots EDU para os robôs e-Puck, e suas respectivas melhorias denominadas apresentadas nos algoritmos RCAAQ, RPCAAM, RHCAAM e RDCAAM. Os resultados dos experimentos para a validação destes modelos serão apresentados nos capítulos posteriores.

---

## Análise da formal da proposta

---

Uma das investigações desenvolvidas neste trabalho foi o estudo do modelo de feromônio invertido utilizado em todas as tarefas para enxames de robôs estudadas neste trabalho do ponto de vista teórico, através da transformação desse modelo em um grafo. Adicionalmente, uma análise da busca competitiva pelos ninhos foi analisada segundo a teoria das filas. A análise formal apresentada neste capítulo foi publicada em (Lima and Oliveira, 2017b), cujo principal objetivo desse estudo é investigação do potencial de cobertura e convergência do algoritmo Cellular Automata Ant (CAA). Sabe-se que o CAA é tido como modelo base para a aplicação nas tarefas de robótica investigadas no presente trabalho, onde não será considerada a memória adicional, para facilitar a análise dos dados. Sabe-se que CAAM é uma generalização de CAA, quando a memória possui comprimento nulo.

### 6.1 Contextualização sobre cobertura de ambientes

A utilização de sistemas de múltiplos robôs pode ser considerada de extrema vantagem em uma gama de áreas e domínios de aplicação (Cortes et al., 2005) e pode ser visto como uma estratégia para a exploração (Choset, 2001), conforme nas seções anteriores. Em um sistema de múltiplos robôs, cada um apresenta um conjunto de componentes para sensoriamento, computação e comunicação. Assim, as estratégias de coordenação apresentam as vantagens de minimização de esforços do robô (distância e energia consumida pelo robô) e tempo para manter a topologia adequada, minimizando o custo computacional e, por consequência maximiza-se a área de cobertura (Choset, 2005). Quando o mapa

do ambiente é conhecido, o desenvolvimento de uma estratégia ótima para a coordenação do time de robôs torna-se relativamente menos complexo. A resolução de um sistema multi-robôs para o problema de cobertura minimal é NP-hard (Zheng et al., 2010). Em outros trabalhos esse problema é resolvido através da inteligência coletiva baseados em algoritmos polinomiais bio-inspirados (Calvo et al., 2012). Por este motivo, neste trabalho, surgiu a motivação para modelar um algoritmo de tempo polinomial, baseado em algoritmos de inteligência coletiva e comportamento social de enxames, para a tarefa de forrageamento. Para o cumprimento desta tarefa é necessário a realização da cobertura de todo o ambiente e o recolhimento de todos os objetos. Neste contexto, a eficácia de muitos modelos é avaliada considerando-se o grau de cobertura obtido (Zheng et al., 2010). A cobertura neste trabalho é dada pelo algoritmo polinomial de feromônio de formigas invertido com memória. Embora seja muito citado como algoritmos de otimização que exploram forma do comportamento social animal, até recentemente este tema foi pouco explorado sob o ponto de vista teórico. Em virtude do que foi mencionado, a análise da eficiência do algoritmo utilizado neste trabalho torna-se fundamental para a verificação se o modelo proposto é capaz de cumprir as tarefas modeladas a partir de qualquer instância de entrada.

O problema de cobertura é definido como a maximização da área pertencente ao campo de visão do menor número de robôs em um determinado instante (Kong et al., 2006). Dessa forma, existem diversos trabalhos que utilizam abordagens para solucionar o problema da cobertura. Esses trabalhos seguem duas linhas distintas. A primeira, considera um ambiente particionado, onde cada robô do time se movimenta de maneira independente e isolada de tal maneira que a combinação das áreas visitadas por eles corresponda a toda a área do ambiente. A partir da troca de informações compartilhadas por esses robôs através de mecanismos de comunicação é possível descobrir as áreas já cobertas. Alguns trabalhos relacionados a essa primeira perspectiva serão apresentados em sequência.

Para a primeira abordagem apresentada em (Kong et al., 2006), o ambiente é dividido virtualmente em células de tamanho fixo. Os robôs são dispostos neste ambiente em posições aleatórias. Cada uma dessas posições são representadas por um vértice do grafo. Se duas dessas áreas são adjacentes, existe uma aresta que liga essas duas regiões. Em todo instante que uma área é coberta, esta informação (vértice) é inserida no grafo e é distribuída e compartilhada entre todos os robôs do time. O processo é finalizado no instante em que o ambiente está coberto. Neste trabalho, ambientes com configurações complexas não podem ser completamente explorados pelo time de robôs.

Outro trabalho dentro da ideia de que os caminhos percorridos pelos robôs podem ser representados por grafos foi apresentado em (Agmon et al., 2008). Neste trabalho, a ideia foi utilizar uma árvore geradora para a construção de uma cobertura eficiente do ambiente a partir da granularização do ambiente em áreas de formato quadrangular. A



ideia inicial surgiu no trabalho de (Gabriely and Rimon, 2001), onde os autores utilizaram a ideia de caminho hamiltoniano que fosse capaz de cobrir todo o ambiente com apenas um robô. A cobertura, neste caso, é obtida percorrendo-se as arestas da árvore. De acordo com (Agmon et al., 2008), quando sistemas multi-robôs são utilizados, a escolha da árvore geradora pode modificar substancialmente a distância entre os robôs, impactando o tempo de cobertura. (Agmon et al., 2008) descreve um algoritmo heurístico composto de duas fases que maximiza a distância entre robôs ao longo do caminho da árvore gerado. Na primeira fase, sub-árvores são geradas iterativamente para cada robô do time, de tal maneira que cada uma dessas árvores é expandida em direção contrária às árvores dos demais robôs. Na segunda fase, o objetivo é conectar essa floresta de tal maneira a não conter ciclos formando-se uma única árvore. A partir da construção dessa árvore, o time de robôs pode percorrê-la segundo os algoritmos propostos em (Hazon and Kaminka, 2005).

Na segunda linha de pesquisa, os robôs devem evitar desperdícios de recursos sensoriais que são dados através da sobreposição dos mesmos. Todavia, a dispersão deve atender o princípio de cobertura total do ambiente. De acordo com (Gage, 1992), essa linha pode ser dividida em três abordagens principais: *blanket*, *barrier* e *sweep*. Na primeira abordagem, o time de robôs deve alcançar um conjunto de pontos estáticos de tal forma a maximizar a área total coberta por eles. Na segunda, os robôs são dispostos em um arranjo estático formando-se uma barreira satisfazendo dois critérios, a maximização da área de cobertura e a preservação de uma determinada área. Na terceira abordagem, a maximização da área de cobertura é dada através da movimentação do time de robôs. Alguns dos principais trabalhos referentes a essas abordagens serão brevemente apresentados a seguir.

Considerando-se a primeira abordagem denominada *blanket*, em (Howard et al., 2002) um sistema multi-robôs foi utilizado para a tarefa de cobertura onde o time desconhece a configuração do ambiente. Para isso, cada robô do time deve movimentar-se de forma a atingir a dispersão, aumentando em consequência disso a cobertura do ambiente. As abordagens em campos potenciais citadas anteriormente também fazem parte dessa estratégia, (Barraquand and Latombe, 1991). Nestes trabalhos os robôs são móveis e guiam-se no ambiente através de forças de repulsão virtuais, tais como, obstáculos ou outros robôs. Esses robôs também sofrem a força da atração guiados através de um ponto de destino. Num ambiente onde somente as forças de repulsão atuam, os robôs tendem a distanciar-se um dos outros até o equilíbrio do sistema, fazendo com que as forças de repulsão sejam anuladas. Neste instante, os robôs permanecem imóveis e então é analisada a cobertura do ambiente. Com essa abordagem, mesmo que o time de robôs inicie o procedimento aglomerados num mesmo local é possível, através das forças de repulsão, que com o passar do tempo esses robôs estejam esparsos no ambiente e realizando uma cobertura do mesmo.

Na abordagem denominada *barrier*, (Kumar et al., 2005) uma formação de arranjo de robôs adequado foi fundamentada teoricamente a fim de proteger uma determinada

região. A quantidade de robôs é definida para que nenhum intruso consiga derrubar essa proteção. Essa análise foi realizada segundo a teoria dos grafos tanto para detectar se uma determinada região é coberta quanto para verificar se a disposição do time de robôs seja ótima. Devido às restrições de comunicação apresentadas no trabalho anterior, em (Cheng and Savkin, 2011) foi proposto um trabalho que minimiza este problema. Todavia, esse tipo de cobertura é mais restrito e espera-se que à medida que o tamanho do ambiente seja reduzido, esta cobertura torne-se mais próxima à da primeira proposta.

Os métodos da abordagem *sweep* possibilitam o deslocamento dos robôs, diferente da metodologia anterior, onde os robôs eram dispostos estaticamente no ambiente. Neste cenário, alguns estudos tem focado no planejamento de caminhos enquanto que outros promovem um deslocamento do time de forma coesa (Gage, 1992). No trabalho de (Alboul et al., 2010) um time de robôs é encarregado de cobrir um ambiente desconhecido com o objetivo de manter a proximidade entre eles. Durante este processo o time de robôs constrói um mapa da topologia do ambiente através de um grafo, cujos vértices representam os triângulos virtuais, que são as áreas que podem ser sensoriadas pelo robô. Os robôs devem se manter próximos o suficiente para que a comunicação entre eles seja garantida. Além disso, o erro de odometria é corrigido a cada instante de tempo e de troca de informações. Em (Cheng et al., 2011) foi proposto um algoritmo orientado através de regras inspiradas na agregação animal observada na natureza. Esse esquema é descentralizado e é capaz de controlar robôs dispostos em linha. Cada robô é orientado de acordo com uma ponderação entre os estados atuais de seus próprios sensores e dos seus vizinhos, fazendo com que um comportamento consensual seja estabelecido pelo time de robôs.

## 6.2 Modelagem da busca no algoritmo CAA através da teoria dos grafos

Um dos objetivos deste trabalho foi a concepção de um modelo baseado em autômatos celulares e em teorias de computação bio-inspirada, para a realização da tarefa de busca de objetos por múltiplos robôs num ambiente bidimensional. Nesta seção será apresentado um modelo discreto baseado na teoria dos grafos e este será associado ao algoritmo CAA através de algumas adaptações ao problema da cobertura de vértices. Com essa adaptação, será possível encontrar o número mínimo de robôs para realizarem uma determinada tarefa num tempo ótimo. Posteriormente, uma redução polinomial do problema CAA será apresentada com o clique, mostrando que o problema pertence à classe de algoritmos NP-completos.

Em ciência da computação, uma cobertura de vértices em um grafo  $G$  é um conjunto de vértices tal que cada aresta do grafo é incidente a pelo menos um vértice do conjunto. Ou seja, é um conjunto de vértices que contém pelo menos uma das pontas de cada aresta. Em

outras palavras, uma cobertura de vértices é um conjunto  $V$  de vértices dotado da seguinte propriedade: toda aresta  $E$  do grafo tem pelo menos uma ponta em  $V$ . O problema de encontrar uma cobertura de vértices mínima é um clássico problema de otimização em ciência da computação e é um exemplo típico de um problema de otimização NP-hard que tem um algoritmo de aproximação. Uma linguagem está contida em NP se e somente se ela é decidida por alguma máquina de Turing não-determinística de tempo polinomial (Sipser, 2006). Esta versão de problema de decisão, o problema da cobertura de vértices, foi um dos 21 problemas NP-completos de Karp (Karp, 1977) e, portanto, um problema NP-completo clássico da teoria da complexidade computacional. Além disso, o problema de cobertura de vértices é tratável em parâmetros fixos e é um problema central na teoria da complexidade parametrizada (Sipser, 2006).

O problema referente à cobertura de vértices, na disciplina de teoria dos grafos, consiste em descobrir: “qual uma cobertura de vértices de um grafo é um conjunto de vértices tal que cada aresta do grafo é incidente, a pelo menos, um vértice do conjunto?”. O problema referente ao algoritmo proposto neste trabalho consiste em verificar considerando-se um grid com  $m \times n$  células com células ponderadas (constantes feromônio), uma quantidade  $F$  de comida, um raio de visão  $r_2$  e uma fila  $Q$ , “qual é o número mínimo de robôs para que todas as células sejam cobertas?”. Dessa maneira, os dois problemas aparentemente possuem características semelhantes, e portanto, faz-se necessária a modelagem do problema de busca em CAAM segundo a teoria dos grafos.

Para isso algumas considerações devem ser realizadas. Primeiro, um algoritmo de cobertura com tempo ótimo para um sistema com  $N$  robôs e tempo total de execução  $T$  vai resultar, teoricamente, num tempo de cobertura de  $\frac{T}{N}$ . Algoritmos de cobertura e de exploração de ambientes aplicados a qualquer tipo de tarefa vão resultar numa cobertura ótima se os robôs estão uniformemente posicionados ao longo da árvore de cobertura, a uma distância de no máximo  $\frac{T}{N}$  do outro robô. Isso significa que à medida que se aumenta a quantidade de robôs é possível que a cobertura tenha o tempo ( $T$ ) de execução total reduzido. Assim, reduzir o problema de CAAM em uma instância ótima consiste também em verificar a quantidade de unidades de comida ( $F$ ) contidas no ambiente de busca. Ou seja, para que a busca seja realizada em tempo imediato  $T = 1$ , deve-se ter mais robôs que unidades de comida  $N \geq F$ . Essa restrição deve-se ao fato de que em nosso modelo cada robô é capaz de encontrar apenas um objeto por vez e mesmo que haja um desperdício de recursos robóticos, com  $N \geq F$  ainda é possível encontrar todos os objetos de uma única vez.

A ideia da transformação do problema da busca em um grafo leva em consideração alguns aspectos importantes. Inicialmente, o reticulado consiste de  $m \times n$  células  $x$  de lado  $l$ . Cada célula do reticulado pode ser considerada bloqueada ou não-bloqueada (livre). As células bloqueadas são as células que contêm um obstáculo. Cada célula livre possui um peso que se refere ao valor advindo das constantes de feromônio em um determinado

instante  $t$ . As células em cinza apresentadas na Figura 6.1 (a) representam os obstáculos, enquanto que as demais células são livres e estão representadas em branco e as células que apresentam um símbolo (★) em azul representam os focos de comida. As unidades de comida estão numa célula de menor valor em sua janela de vizinhança. Uma janela de vizinhança  $\eta_x^t$  consiste em todas as células  $x$  que pertencem ao raio de visão  $r_v$  do robô num instante  $t$ . Todos os robôs são idênticos em tamanho e características (time homogêneo). Cada passo de movimento do robô é atômico, isto é, precisa ser executado por completo por um robô. O tempo de circuito de um robô é o somatório de todos os tempos gastos para percorrer as células de menor peso, e o valor de passo é  $l$  (vertical ou horizontal) ou  $l\sqrt{2}$  (diagonal).

Para a transformação deste problema genérico  $G = (V, E)$  em um grafo não-direcionado com vértices ponderados,  $G = (V, E, \omega)$ , tal que,  $\omega = \{\delta, \delta', -\beta\}$ . Todas as células do reticulado devem ser transformadas em um vértice  $v_i \in V$  no grafo  $G$ . Cada um desses vértices, que fazem parte da vizinhança  $\eta_x^t$  de um célula central, são interligadas através de uma aresta  $e_{ij}$  neste grafo,  $e_{ij} \in E$ . Ou seja, uma aresta  $e_{ij}$  liga dois vértices  $v_i$  e  $v_j$ , formando-se a aresta  $(v_i, v_j)$ . Como os vértices em cinza, que representam os obstáculos não podem ser acessados pelos robôs, na Figura 6.1 (b), então os mesmos podem ser excluídas do grafo com suas respectivas arestas. O grafo que representa o ambiente resultante está apresentado na Figura 6.1 (c). Para a devida simplificação e evitar a duplicação de valores que atrapalham a escolha, cada célula do ambiente possui um peso diferente (feromônio  $\omega = \{\delta, \delta', -\beta\}$ ) e este peso, que corresponde à quantidade de feromônio, está representado por um inteiro. Como o robô possui raio de visão  $r_v = 1$ , então o mesmo consegue ver e armazenar as células adjacentes  $\eta_x^t$ , e sabe quais são as células que não podem ser acessadas através de  $Q$  (que em CAA  $Q = 0$  ausência de memória), bem como verificar as unidades de comida. Assim, para a devida simplificação, apenas a célula de menor valor pertencente à janela da vizinhança  $\eta_x^t$  precisa ser armazenada. Todas as janelas possuem uma única célula de menor valor, e conseqüentemente, uma única célula de alimento (caso exista). Nem toda janela possui uma unidade de alimento, mas se essa unidade existir, ela é única por janela e está contida na célula de menor valor, para fins de simplificação. Uma janela de vizinhança está apresentada no quadrado da Figura 6.1 (c). Assim, o grafo não-direcionado resultante, que representa o ambiente, pode ser mostrado na Figura 6.1 (d). Assim, foi mostrado que uma instância do problema  $CAA \leq_P VERTEX\_COVER$ . Logo um conjunto de vértices  $S$  que satisfazem o conjunto de vértices no problema da cobertura representa uma instância de  $N$  robôs que vai realizar em tempo ótimo a busca no algoritmo CAAM para a tarefa de forrageamento.

Para a transformação do problema da cobertura vértices no problema de busca do algoritmo CAA, proposto neste trabalho, basta realizar o procedimento inverso. Ou seja, cada célula  $x$  de menor peso representada por sua vizinhança  $\eta_x^t$  no através de um vértice  $v_i$  no grafo ilustrado na Figura 6.1 (d) possui um índice para essa vizinhança. Para a

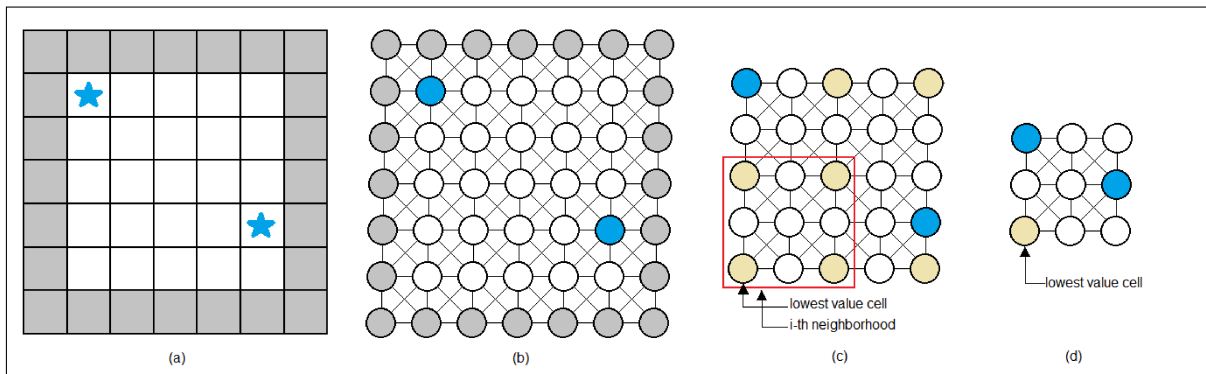


Figura 6.1: Redução do problema. (a) Representação real do ambiente com obstáculos e comida. (b) Representação do ambiente através de um grafo não-orientado. (c) Grafo não-orientado com exclusão dos obstáculos. (d) Grafo final com as células de menor valor que representam todas as vizinhanças possíveis de raio 1 do grafo anterior.

transformação, basta resgatar os índices de suas respectivas vizinhanças e fazer as respectivas ligação de arestas  $e_i$  aos vértices  $v_i$  remanescentes. A partir desta transformação, o algoritmo de cobertura pode ser aplicado ao grafo resultante. Assim, se a instância inicial de entrada, que representa a quantidade de robôs e suas posições, correspondesse ao conjunto de vértices resultante. Esse algoritmo teria tempo de execução  $T = 1$  no processo de busca, desconsiderando-se, então, o processo de *homing*. Se o tempo  $T$  de conclusão fosse no máximo  $T = 1$  numa cobertura mínima, isso significa que o tempo de circuito é o tempo gasto para percorrer as células de menor peso, e o valor de passo é  $l$  (vertical ou horizontal) ou  $l\sqrt{2}$  (diagonal). Como todos os robôs realizam seu movimento de forma síncrona e paralela, o tempo total seria  $\max(l\sqrt{2}, l)$ . Se algum movimento em diagonal, o tempo para a execução é  $l\sqrt{2}$ , caso contrário, é  $l$ . Esse fato, ocorre, porque todas os vértices, que representam as células, com pesos não minimais e tal que  $v_i \in Q$  foram eliminados. Assim, mostra-se que o algoritmo de busca proposto foi transformado numa instância em tempo polinomial  $P$  ao algoritmo de cobertura de vértices  $VERTEX\_COVER \leq_P CAA$ .

### 6.3 Análise de complexidade do algoritmo CAA

Após a modelagem do da busca da tarefa de forrageamento referente ao modelo CAA em um grafo não direcionado com suas respectivas alterações, vamos mostrar que esse problema é NP-completo e se equivale ao problema de clique máximo, através de uma redução polinomial ( $CLIQUE \equiv_P CAA$ ). O problema do clique máximo é um problema conhecido por ser NP-completo (Sipser, 2006). Um clique em um grafo não-direcionado  $G = (V, E)$  é um subconjunto de vértices  $S \subseteq V$ , tal que para cada dois vértices em  $S$ , existe uma aresta os conectando. Isso se equivale a dizer que um subgrafo induzido de  $S$  é completo (em alguns casos, o termo clique também pode ser referência ao subgrafo). Um clique máximo é o maior clique possível em um dado grafo. O número do clique  $\omega(G)$  de um grafo  $G$  é o número de vértices de um clique máximo em  $G$ . O número da intersecção

de  $G$  é o menor número de cliques que, juntos, cobrem todas as arestas de  $G$  (Sipser, 2006). O problema em CAAM consiste em encontrar um número mínimo de robôs  $N$  que resolvam o problema da busca na tarefa de forrageamento no menor tempo possível.

Deseja-se provar que  $CLIQUE \equiv_P CAA$ , para isso, primeiramente, devemos considerar um grafo não-direcionado  $G = (V, E)$ . O seu respectivo complemento é dado por um grafo  $G'$ , onde  $G' = (V, E')$  sendo que  $E' = \{(v_i, v_j) : (v_i, v_j) \notin E\}$ . Devemos considerar também que um grafo  $G$  tem um clique de tamanho  $N$ , se e somente se ( $\leftrightarrow$ ),  $G'$ , que representa CAA, tem uma cobertura de vértices de tamanho  $|V| - N$ . A Figura 6.2 (a) representa um grafo  $G$  e o respectivo grafo  $G'$  está ilustrado na Figura 6.2 (b).

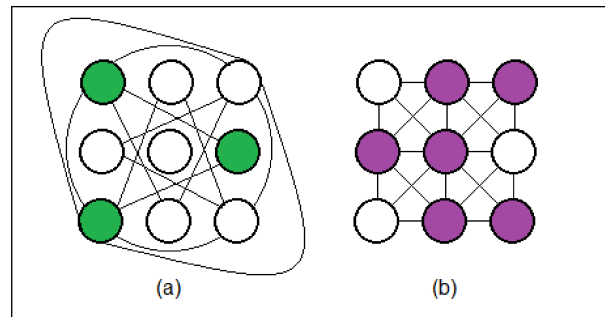


Figura 6.2: Redução do problema. (a) Grafo com a representação do grafo de cobertura de vértices mínima. (b) Representação do grafo da clique maximal (Sipser, 2006).

A ideia da prova do “ $SE$ ” ( $\rightarrow$ ) consiste em supor inicialmente que o grafo  $G$  apresenta um clique  $S$  com  $|S| = |V| - |S'|$  e que  $|S'| = N$ , logo o comprimento de  $S'$  é dado por  $|S'| = N$  (Sipser, 2006). Para mostrar que  $S'$  é uma cobertura, considere que qualquer aresta  $(v_i, v_j) \in E'$ , então  $(v_i, v_j) \notin E$ . Por isso, pelo menos um vértice  $v_i$  ou  $v_j$  não está em  $S$ , considerando-se que  $S$  forma um clique. Além disso, pelo menos um  $v_i$  ou  $v_j$  estão em  $S'$ , dessa forma,  $(v_i, v_j)$  é coberto por  $S'$ .

Para provar o “ $SOMENTE SE$ ” ( $\leftarrow$ ) devemos supor que o grafo  $G$  apresenta uma cobertura  $S'$  com  $|S'| = N$  e considerar que  $S = |V| - |S'|$ , assim temos que  $|S| = |V| - N$  (Sipser, 2006). Para mostrarmos que  $S$  é uma clique, consideramos que algumas arestas  $(v_i, v_j) \in E'$ . Se a aresta  $(v_i, v_j) \in E'$ , então, tanto o vértice  $v_i \in S'$  quanto  $v_j \in S'$  ou ambos  $\in S'$ . Por contradição, se  $v_i \notin S'$  e  $v_j \notin S'$ , então a aresta  $(v_i, v_j) \in E$ . Assim, temos que  $S$  é um clique no grafo  $G$ .

Assim, foi provado que  $CLIQUE \equiv_P CAA$  e sabe-se que  $CLIQUE \equiv_P VERTEX\_COVER$  (Sipser, 2006), portanto o  $CAA \equiv_P VERTEX\_COVER$ . Isso significa que um instância em CAAM que representa um conjunto de  $N$  robôs, satisfaz o problema de cobertura de vértices para um conjunto de  $S$  vértices que por sua vez satisfaz o problema do clique com um conjunto  $S'$  de vértices. Portanto, torna-se importante a utilização de algoritmos que minimizam a complexidade desse tipo de tarefa, pois problemas NP são computacionalmente difíceis de serem resolvidos (Sipser, 2006).

Assim, fundamenta-se a importância da criação do algoritmo polinomial baseado em ACs com memória como uma solução aproximada para a tarefa de forrageamento de um

time de robôs através do algoritmo denominado neste trabalho de CAA. Lembrando-se que a busca ótima seria executada em  $T = 1$  caso todos os robôs fossem colocados de forma a cobrir os vértices do grafo (instância da cobertura de vértices). Ou seja, seria utilizada a quantidade  $N$  mínima de robôs para realizar a busca em  $T = 1$ . Além disso, justifica-se a importância da aplicação de  $T$  iterações até que a tarefa seja completamente concluída, uma vez que, encontrar a instância ótima num tempo  $T = 1$  é um problema NP-completo.

## 6.4 Razão de aproximação do da busca do modelo CAA segundo o problema da árvore de cobertura

Como foi visto anteriormente, uma cobertura de multi-robôs num tempo ótimo resulta num tempo de cobertura menor que o tempo gasto para a cobertura de um único robô. Para resolver o problema de instanciação  $I$  da busca num tempo ótimo  $S^*$ , um algoritmo NP-completo é necessário (Sipser, 2006). Os computadores atuais demorariam milhares de anos para resolver instâncias simples do problema num tempo ótimo. Assim, algoritmos aproximados de tempo polinomial com fatores constantes de aproximação são necessários para a solução  $S(I)$  do problema. Por isso, surgiu a motivação para a construção de um algoritmo baseado em autômatos celulares e teorias baseadas em teorias de inteligência coletiva para resolver este problema (Sipser, 2006).

Um algoritmo aproximado para um problema  $\Gamma$  é um algoritmo polinomial que produz uma solução aproximada  $S(I)$  para uma instância  $I$  de  $\Gamma$  (Cormen, 2009). O comportamento do algoritmo é descrito pela razão de aproximação  $R(I) = \frac{S(I)}{S^*(I)}$ , onde  $S^*(I)$  é a solução ótima. No caso de um problema de maximização, a razão é invertida. Em ambos os casos,  $R(I) \geq 1$ . O problema da cobertura de vértices é um problema de minimização. A análise será feita, primeiramente, considerando-se um único robô e posteriormente um time com  $N$  robôs.

Para a realização da cobertura na tarefa de busca, o robô precisa visitar todas as células do reticulado ou percorre-las até encontrar todas as unidades de comida. Duas análises podem ser realizadas. A primeira consiste em aceitar que cada nó do grafo será visitado uma única vez (melhor caso) e as posições das comidas estão em posições intermediárias no caminho percorrido. A segunda considera que os nós podem ser visitados mais de uma vez pelo mesmo robô (pior caso) (Cormen, 2009). Em todas as análises a quantidade de unidades de alimento  $F$  é igual à quantidade  $N$  de robôs posicionados.

A análise do melhor caso considera-se a construção de um grafo  $G$  e as unidades de comida serão representadas em nós intermediários da árvore (nós que não são folha) (Cormen, 2009). Primeiramente, deve-se construir um grafo no qual os vértices correspondem às células do reticulado livres e as arestas conectam essas células não bloqueadas. O grafo

gerado precisa ser conexo. Em seguida uma árvore de cobertura  $A$  é construída iterativamente. Um robô está posicionado numa célula arbitrária que é considerada o nó raiz do grafo  $G$ , apresentado em vermelho na Figura 6.3 (a). As unidades de comida estão representadas em azul na Figura 6.3. Cada nó pertencente ao conjunto da árvore  $A$  é dado partir da configuração de pesos de cada célula, que correspondem ao feromônio. A cada passo de tempo  $t$ , enquanto nem todas as unidades de comida tiverem sido capturadas ou enquanto o grafo  $G$  não for uma árvore de cobertura, escolher uma aresta segura que liga o vértice  $v_i$  atual do robô com o vértice de menor custo  $v_j$ . Uma árvore de cobertura de menor custo não gera ciclos (Cormen, 2009). Além disso, para a escolha do novo vértice  $v_i \in A$ , a cada passo de tempo  $t$ , o robô escolhe uma aresta do grafo e atualiza a sua célula  $x$ , representada pelo vértice  $v_i$  do grafo, e de suas vizinhas  $\eta_x^t$  de acordo com o raio  $r_2$  de incidência do feromônio. Assim, tem-se um exemplo arbitrário possível de uma árvore de custo mínimo, que representa a solução aproximada  $S(I)$  do problema ilustrada na Figura 6.3 (b). O tempo  $t$  da solução aproximada para a captura de objetos por um único robô é dada pela altura da árvore  $h$ , que neste caso representa a quantidade de vértices  $V$  (caminho sem ciclos), e temos que  $S(I) = |V| - \frac{|V|}{F+1}$ . O algoritmo ótimo retorna solução  $S^*(I) = 1$ , que representa o posicionamento ótimo do robô. Assim, a razão de aproximação para o algoritmo aproximado CAAM é  $R(I) = |V| - \frac{|V|}{F+1}$ , já que temos um problema de minimização.

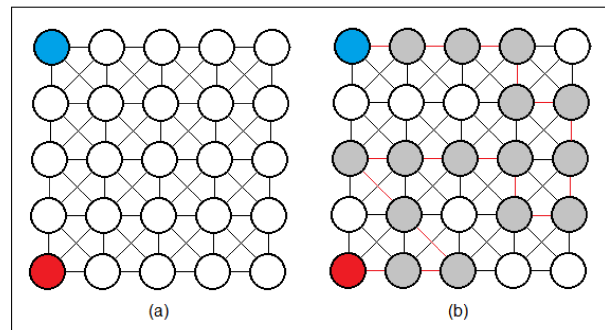


Figura 6.3: Redução do problema. (a) Grafo com a representação da instância inicial do problema. (b) Árvore para a resolução do problema.

O pior caso, é representado pela ideia de que o robô precisa visitar grande parte das células, ou seja, as unidades de comida estão posicionadas nos nós mais profundos da árvore gerada. Para a representação do problema, será considerado um grafo  $G$  apresentado na Figura 6.4 (a). Os nós em azul representam as unidades de comida e o nó vermelho representa o robô. Em determinados momentos é necessário que o robô retorne alguns vértices já previamente visitados por encontrar deadlocks. Além disso, vértices podem apresentar pesos iguais no tempo  $t$  e podem ser visitados novamente. Na Figura 6.4 (b) os visitados mais de uma vez (deadlock) estão ilustrados em cinza mais escuro, enquanto que os demais nós em cinza claro representam a árvore de profundidade percorrida pelos robôs. Para a escolha do novo vértice  $v_i \in A$ , a cada passo de tempo  $t$  o robô escolhe



uma aresta do grafo, atualiza a sua célula  $x$ , representada pelo vértice  $v$  do grafo, e de suas vizinhas  $\eta_x^t$  de acordo com o raio  $r_2$  de incidência do feromônio. Assim, tem-se uma árvore de custo mínimo, que representa a solução aproximada  $S(I)$  do problema ilustrada na Figura 6.3 (b). O tempo  $t$  da solução aproximada para a captura de objetos por um único robô é proporcional ao número de vértices  $V$  somados ao número de arestas  $E$  do grafo  $G$  ao qual o robô atravessa. O percurso definido pelo robô é similar à busca em profundidade. O algoritmo ótimo retorna solução  $S^*(I) = 1$ , que representa o posicionamento ótimo do robô. Assim, a razão de aproximação para o algoritmo aproximado CAA é  $R(I) = |V + E|$ , já que temos um problema de minimização.

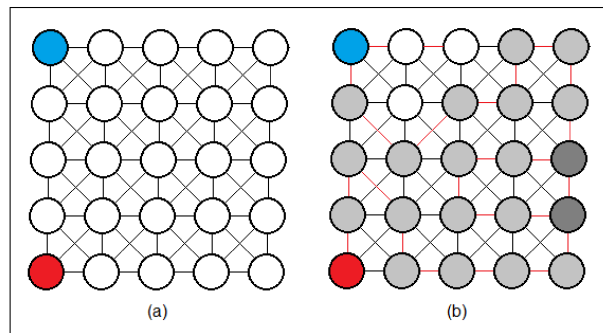


Figura 6.4: Redução do problema. (a) Grafo com a representação da instância inicial do problema. (b) Árvore para a resolução do problema.

A análise da razão de aproximação em um algoritmo de cobertura com tempo ótimo para um sistema com  $N$  robôs vai resultar, teoricamente, num tempo de cobertura de  $\frac{T}{N}$ . Algoritmos de cobertura e de exploração de ambientes aplicados a qualquer tipo de tarefa vão resultar numa cobertura ótima se os robôs estão uniformemente posicionados ao longo da árvore do cobertura. Isso significa que à medida que se aumenta a quantidade de robôs é possível que a cobertura tenha o tempo ( $T$ ) de execução total reduzido. Para a análise da razão de aproximação entre a solução do algoritmo ótimo e o tempo apresentado pelo algoritmo aproximado CAA, vamos considerar o pior caso da análise com um único robô, apresentada anteriormente. Neste caso, se o tempo para terminar a tarefa era de  $S(I) = |V + E|$ , então ao se incrementar a quantidade de robôs o tempo reduzido passa a ser  $\frac{|V+E|}{N}$ . O algoritmo ótimo retorna solução  $S^*(I) = 1$ , que representa o posicionamento ótimo do time de robôs. Assim, a razão de aproximação para o algoritmo aproximado CAA é  $R(I) = \frac{|V+E|}{N}$ , já que se trata de um problema de minimização.

## 6.5 Análise do sistema baseada em teoria das filas

Na última seção a partir da análise formal para modelar o processo de busca, foi mostrado que o desempenho do time de robôs aumenta à medida que o número de robôs também aumenta. A tarefa de forrageamento é composta principalmente em dois **estados**: busca e regresso. Alguns parâmetros tem que ser configurados para executar a

tarefa de forrageamento de forma satisfatória, não só considerando um estado, mas ambos trabalhando de forma sincronizada. A busca (*searching*) multi-objetiva é cooperativa, no entanto o regresso (*homing*) para o ninho é uma busca competitiva, que dispõe de poucos recursos (ninhos). Essa pouca quantidade de recursos faz com que muitos robôs fiquem ociosos numa fila esperando para depositar o objeto. Essa espera provoca um atraso no sistema, devido ao ponto de gargalo ou estrangulamento (ninhos). Considerando o fluxo de robôs nos ninhos, então, o que limita o seu normal funcionamento são os múltiplos objetos a serem depositados em um mesmo instante de tempo, pois contribui para o estreitamento do fluxo e conseqüentemente para a redução da velocidade deste sistema. Além disso, o efeito de fricção ocorre devido ao fato de que alguns robôs estão no momento *depositing*  $\rightarrow$  *searching*, e forças opostas tornam o sistema ainda mais lento. Para isso, uma análise desses efeitos será realizada através da teoria das filas.

Os experimentos aqui apresentados foram realizados utilizando ambientes virtuais implementados na linguagem C com o objetivo de: (i) avaliar diferentes taxas de evaporação, (ii) avaliar diferentes tamanhos  $N$  de equipe para executar a tarefa e que o tempo  $T$  na busca é reduzido quando temos  $N$  robôs para um tempo  $\frac{T}{N}$ , (iii) avaliar o desempenho da equipe relacionado com a cobertura de meio ambiente, (iv) avaliar o desempenho da equipe na tarefa de forrageamento. Todas as experiências foram conduzidas em simulação de ambientes com  $20 \times 30$  células utilizando dois tipos diferentes de obstáculos. As simulações aqui apresentadas foram baseadas nos experimentos propostos em Lima et al. (2016a) na tarefa de vigilância. Neste trabalho a análise vai além do trabalho anterior, considerando-se também a análise formação de filas de espera e o tempo de entrega.

O primeiro experimento foi realizado para demonstrar a diferença no tempo que um robô ( $N = 1$ ) demora para encontrar todos os objetos ( $F = 30$ ) no ambiente utilizando-se 30 robôs ( $N = 30$ ) com  $\delta = 1$ ,  $\delta' = 0.25$  e  $\beta = 0.2$ . A Figura 6.5 mostra claramente a diferença destas duas configurações, onde o eixo  $x$  representa o tempo gasto para o time concluir a tarefa e o eixo  $y$  representa o label do robô que realizou a tarefa. A primeira abordagem mostra que o uso 30 robôs é possível capturar todas as unidades de alimentos rapidamente, próximo de um valor baixo com  $\frac{T}{N}$  ao passo de que usando apenas um robô esse tempo torna a execução da tarefa mais lenta. Esse fato ocorre devido aos 30 robôs serem capazes de se espalhar sobre o ambiente fazendo a cobertura do mesmo. Usando 30 robôs, a equipe pegou todas as unidades de alimentos usando  $T = 59$  iterações, e quase todos os robôs foram usados para coletar objetos. Assim, o algoritmo também é apropriado para distribuir a tarefa para que cada robô fique a menor unidade de tempo  $t$  ocioso. Por outro lado, quando foi utilizado apenas um robô, todas as unidades de alimentos foram encontrados usando  $T = 1900$ , conforme analisado anteriormente temos que  $N \times \frac{T}{N} = 30 \times 59 = 1770 \approx 1900$  iterações.

Um segundo experimento verifica se o ritmo de entrega da comida é semelhante ao o tempo gasto no estado de busca, usando  $N = 1$  e  $N = 30$  robôs. Se o número de iterações

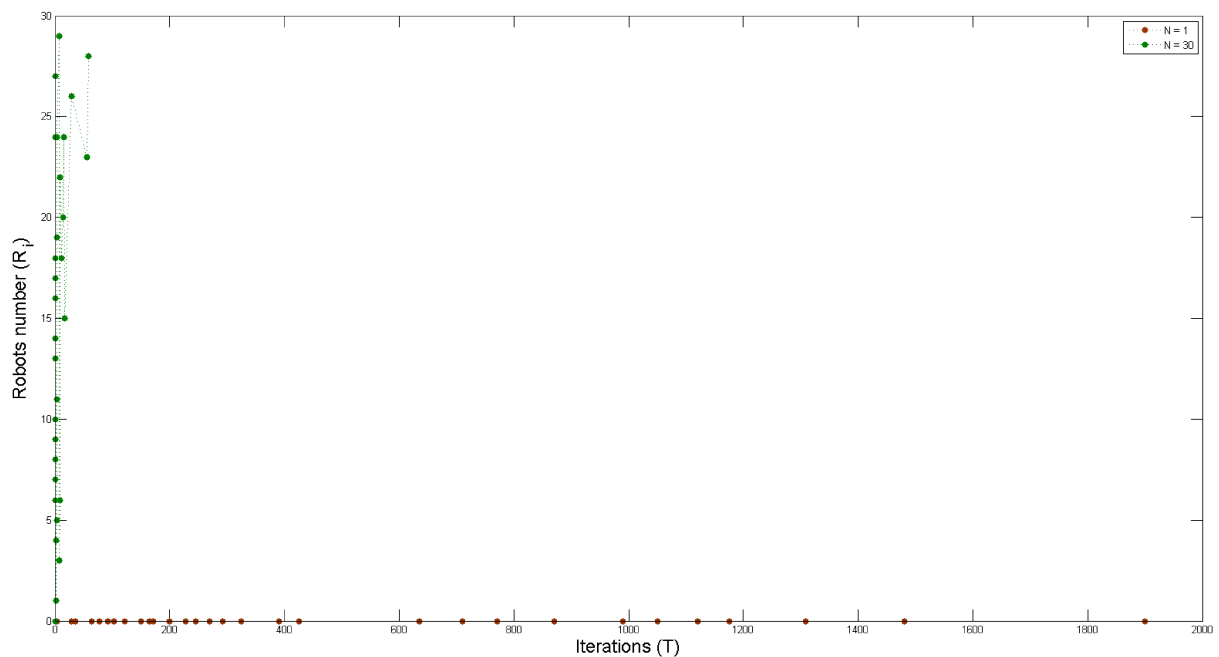


Figura 6.5: Número total de iterações usando  $N = \{1, 30\}$  robôs para coletar todos os objetos.

para depositar todos os objetos for semelhante em ambos os estados (busca e regresso), então o modelo de coordenação proposto não permite a formação de filas próximas aos ninhos. É possível observar na Figura 6.6 que o tempo de entrega e de concluir a tarefa ao usar  $N = 1$  robôs é quase o mesmo do último experimento, ou seja, o tempo para concluir a tarefa é de  $T = 1918$  iterações. Por outro lado, quando é usado  $N = 30$  robôs o tempo gasto para a entrega a comida é cerca de 10 vezes maior, o tempo gasto para concluir a tarefa foi  $T = 412$  iterações. O robô passou um longo tempo com o objeto capturado à espera na fila perto dos ninhos, o que atrasa a realização de tarefas do sistema quando o número de robôs é amplamente aumentado.

Finalmente, um terceiro experimento verifica se o tempo total para a tarefa de forrageamento, ou seja, para entregar todos os alimentos no ninho usando  $N = \{3, 6, 9, 12, 15\}$ . A Figura 6.7 mostra no eixo  $x$  a iteração correspondente em que a comida é capturada e no eixo  $y$  rótulo do robô correspondente que captura a comida. É possível observar que quando o número de robôs é aumentado o número de iterações para terminar o recolhimento de alimentos é reduzido. Quando  $N = 3$  o número de iterações para encontrar todos os objetos é de  $T = 410$ , no outro extremo, quando  $N = 15$  o tempo para recolher todos os objetos é de  $R T = 98$ . Quando o número de robôs aumenta extensivamente ( $N = 30$ ), o sistema ficar sobrecarregado devido à formação de linha, degradando o desempenho da equipe, porém o tempo para encontrar todas as unidades de comida é muito rápido.

Depois disso, uma outra simulação foi realizada para analisar: (i) a cobertura do ambiente usando o número de passos utilizando diferentes configurações de  $\delta'$  and  $\beta$ , e

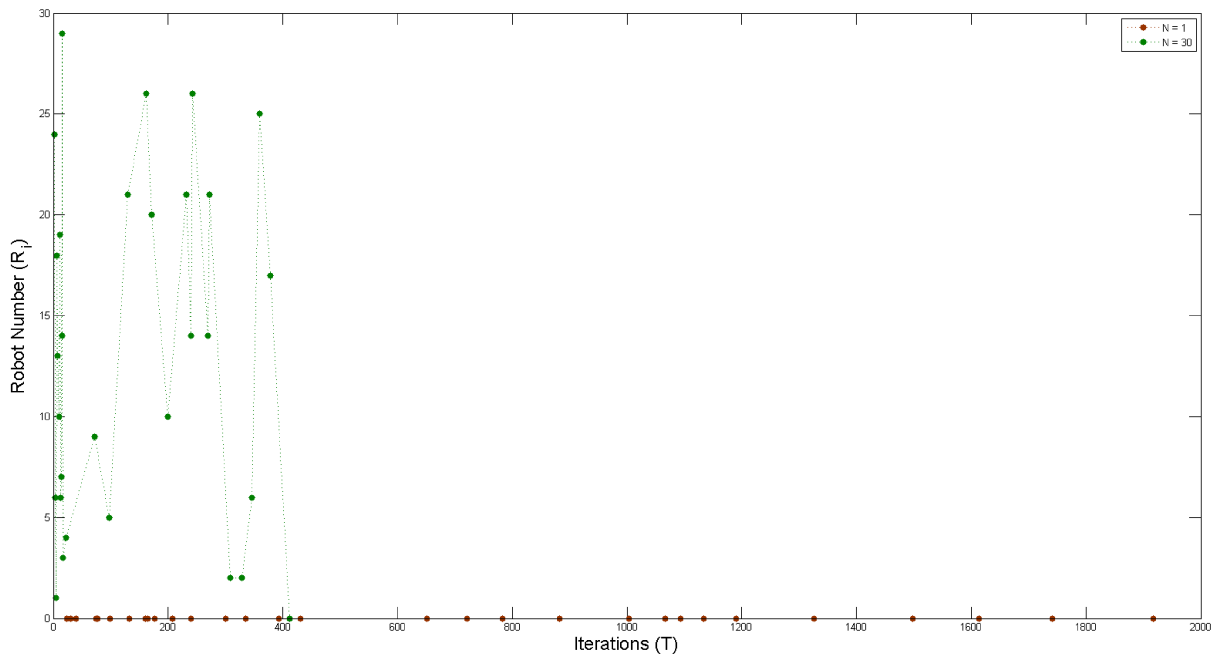


Figura 6.6: Número total de iterações usando  $N = \{1, 30\}$  robôs para todos os objetos a deposição.

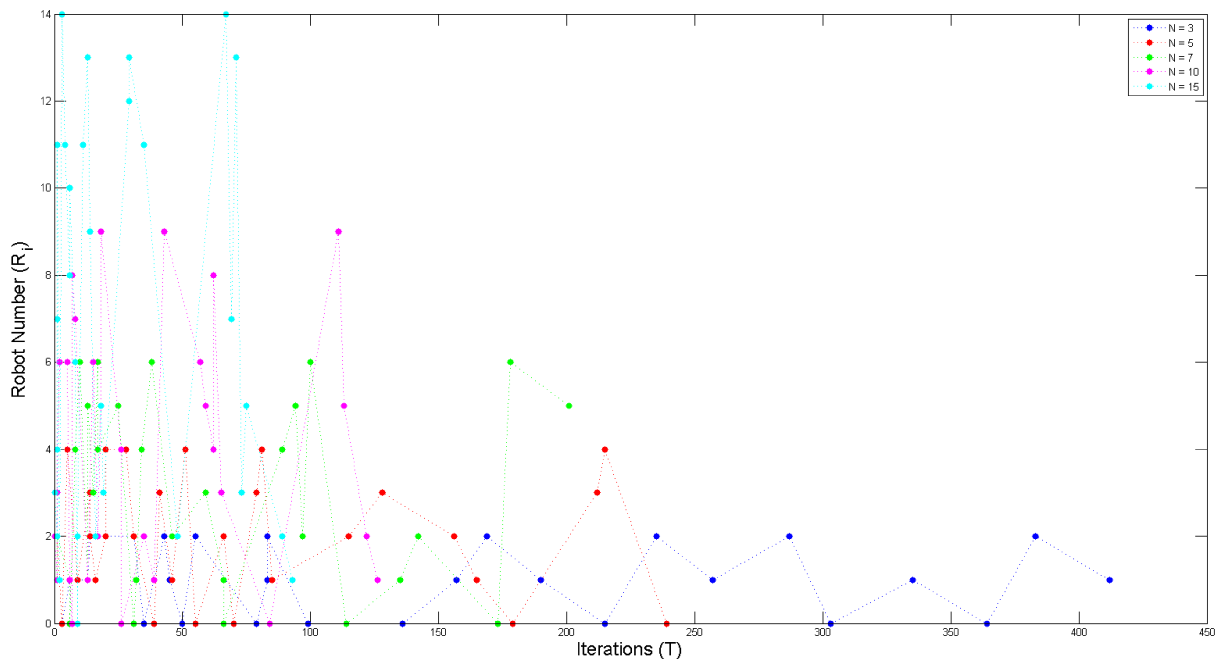


Figura 6.7: Número total de iterações usando  $N = \{3, 6, 9, 12, 15\}$  robôs para encontrar todos os objetos.

(ii) a formação de fila perto dos ninhos quando o número de robôs é aumentado, usando o traço de feromônio na última etapa da tarefa no último instante de tempo  $t = T$ . O experimento considera  $N = \{1, 3, 5, 7, 10, 15, 20, 30\}$  e as duas piores abordagens apresentadas em Lima and Oliveira (2017a) usando  $(\delta' = 0.9, \beta = 0.2)$  e  $(\delta' = 0.25, \beta = 0.5)$ , e a melhor abordagem  $(\delta' = 0.25, \beta = 0.2)$ . A Tabela 6.1 apresenta os dados dos experimentos da Figura 6.8 e da Figura 6.9, que representa o ambiente com  $20 \times 30$  células usando dois

ninhos  $\delta = 1$ . Cada coluna da Tabela 6.1 representa um  $\delta', \beta$ .

A Tabela 6.1 também apresenta o número máximo  $v_{max}$  de passos que uma determinada célula recebeu em uma simulação durante o tempo  $T$  de realização da tarefa, o número de iterações  $T_g$  que o enxame de robôs esperou para capturar todas as unidades de alimentos, e a taxa máxima feromônio acumulado que foi capturado na última  $T$  iteração.

Esta simulação apresenta a cobertura ambiente com  $30 \times 20$  células com  $\delta = 1$ . Uma boa cobertura ambiente é quando todas as células recebem quase a mesma quantidade de passos. O número de vezes que cada célula  $x_{ij}$  é visitado durante  $T$  passos de tempo foi calculado em cada passo  $t$ . A quantidade máxima de passos que uma célula recebe é representada pela cor vermelho escura ao passo que o azul escuro representa as células menos visitadas. Estes valores foram utilizados para construir gráficos de temperatura mostrado na Figura 6.8, um número elevado de células azuis escuros representa uma simulação com baixa cobertura, indicando um desempenho cobertura deficiente. A célula mais visitada definida na Tabela 6.1 pelo valor  $v_{max}$ , para cada variação  $(\delta', \beta, N)$ .

De acordo com as simulações, quando  $N = 1$  o sistema realizou um processo de busca lento, deve-se ao fato de que o tempo para concluir as três variações de  $\delta', \beta$  foi maior do que  $T \geq 1312$ , já para depositar todas as unidades de alimento  $T_g \geq 1323$ , a diferença  $T_g - T$  representa a distância entre a célula da comida e o ninho. Usando  $N = 3$  robôs, tem-se que  $v_{max} = 28$  representa os melhores valores de  $\delta', \beta$ . Neste caso, a distância entre a última célula de alimentos a partir do ninho é 18 passos. Usando  $N = \{5, 7, 10, 15\}$  robôs, a formação de fila observada não é expressiva, pois o tempo para agarrar a última unidade de alimentos  $T_g$  e o tempo da tarefa realização  $T$  é quase o mesmo para todas as variações de  $\delta', \beta$ .

A Figura 6.8c mostra a melhor cobertura da rede. As Figuras 6.8b e 6.8a as cores observadas no ambiente não são homogêneas. À medida que o número de robôs aumenta em  $N = \{1, 3, 5, 7, 10, 15\}$ , o número de iterações diminui, quase proporcionalmente. Quando o número de robôs é  $N = \{20, 30\}$ , o processo de formação de linha começa a ser observado, devido ao fato de que muitas vezes os robôs ficam na fila com o seu objeto durante o estado de *homing*. Além disso, muitos robôs ficam presos na fila atrapalhando o desempenho da equipe. Para todas as variações  $\delta', \beta$  é produzida uma média na linha - além da distância percorrida para a célula de alimentos e o ninho - pelo menos  $\bar{x} \approx 150$  iterações. Este processo aumenta  $T$ , degradando o desempenho da equipe.

A Figura 6.8 mostra que os robôs fizeram uma boa cobertura do ambiente para quase todos  $\delta', \beta$  valores. Mas quando o número de robôs diminui, um pior desempenho cobertura é observado, devido ao fato de que algumas células apresenta poucas visitas, enquanto outras estão sobrecarregadas de visitas. Também é possível observar que com  $N = \{20, 30\}$  existem vários pontos alaranjados e vermelhos perto dos ninhos, indicando que os passos perto dos ninhos extrapolam as outras áreas visitadas - devido à formação de filas. Portanto, considerando as variações em  $\delta', \beta$  e  $N$ , podemos concluir que para

Tabela 6.1: Os dados coletados para verificar a formação de linha perto dos ninhos e a cobertura do ambiente.

Quantidade de Robôs	(a) $\delta' = 0.9$ $\beta = 0.2$	(b) $\delta' = 0.25$ $\beta = 0.5$	(c) $\delta' = 0.25$ $\beta = 0.2$	Média na Coluna
$N = 1$	$v_{max} = 23$ $\tau_{max} = 64.05$ $T_g = 1312$ $T = 1323$	$v_{max} = 28$ $\tau_{max} = 1.20$ $T_g = 2678$ $T = 2695$	$v_{max} = 23$ $\tau_{max} = 16.06$ $T_g = 1900$ $T = 1918$	$\bar{x}_1 = 15.6$ $\sigma_1 = 3.2$
$N = 3$	$v_{max} = 23$ $\tau_{max} = 70.27$ $T_g = 824$ $T = 827$	$v_{max} = 22$ $\tau_{max} = 1.20$ $T_g = 928$ $T = 945$	$v_{max} = 28$ $\tau_{max} = 29.31$ $T_g = 412$ $T = 430$	$\bar{x}_3 = 12.3$ $\sigma_3 = 9.8$
$N = 5$	$v_{max} = 24$ $\tau_{max} = 74.58$ $T_g = 473$ $T = 474$	$v_{max} = 31$ $\tau_{max} = 1.65$ $T_g = 444$ $T = 462$	$v_{max} = 23$ $\tau_{max} = 32.29$ $T_g = 239$ $T = 257$	$\bar{x}_5 = 345.6$ $\sigma_5 = 12.7$
$N = 7$	$v_{max} = 23$ $\tau_{max} = 82.25$ $T_g = 232$ $T = 250$	$v_{max} = 28$ $\tau_{max} = 1.70$ $T_g = 345$ $T = 362$	$v_{max} = 23$ $\tau_{max} = 29.62$ $T_g = 201$ $T = 219$	$\bar{x}_7 = 18.6$ $\sigma_7 = 0.5$
$N = 10$	$v_{max} = 28$ $\tau_{max} = 100.30$ $T_g = 257$ $T = 270$	$v_{max} = 29$ $\tau_{max} = 23.60$ $T_g = 156$ $T = 173$	$v_{max} = 27$ $\tau_{max} = 44.83$ $T_g = 126$ $T = 144$	$\bar{x}_{10} = 16.0$ $\sigma_{10} = 2.6$
$N = 15$	$v_{max} = 32$ $\tau_{max} = 133.63$ $T_g = 214$ $T = 217$	$v_{max} = 23$ $\tau_{max} = 22.70$ $T_g = 96$ $T = 114$	$v_{max} = 36$ $\tau_{max} = 68.91$ $T_g = 93$ $T = 111$	$\bar{x}_{15} = 13.0$ $\sigma_{15} = 8.6$
$N = 20$	$v_{max} = 134$ $\tau_{max} = 660.59$ $T_g = 156$ $T = 161$	$v_{max} = 362$ $\tau_{max} = 789.04$ $T_g = 129$ $T = 388$	$v_{max} = 260$ $\tau_{max} = 626.87$ $T_g = 121$ $T = 287$	$\bar{x}_{20} = 143.3$ $\sigma_{20} = 128.5$
$N = 30$	$v_{max} = 497$ $\tau_{max} = 3236.69$ $T_g = 169$ $T = 522$	$v_{max} = 362$ $\tau_{max} = 783.34$ $T_g = 55$ $T = 386$	$v_{max} = 545$ $\tau_{max} = 1257.54$ $T_g = 59$ $T = 412$	$\bar{x}_{30} = 345.6$ $\sigma_{30} = 12.7$
Média na Coluna	$\bar{x}_a = 50.87$ $\sigma_a = 122.21$	$\bar{x}_b = 86.75$ $\sigma_b = 129.96$	$\bar{x}_c = 65.87$ $\sigma_c = 117.69$	Desvio Padrão

este ambiente de configuração o melhor de deposição  $\delta', \beta$  de taxas/evaporação e usando um  $3 \leq N \leq 15$  leva a equipe para os melhores comportamentos. Outra simulação usada para refinar a taxa de evaporação correspondente ao  $\delta', \beta$  de CAA, a partir das três combinações diferentes mostrados na Tabela 6.1, e o mesmo ambiente de simulação.

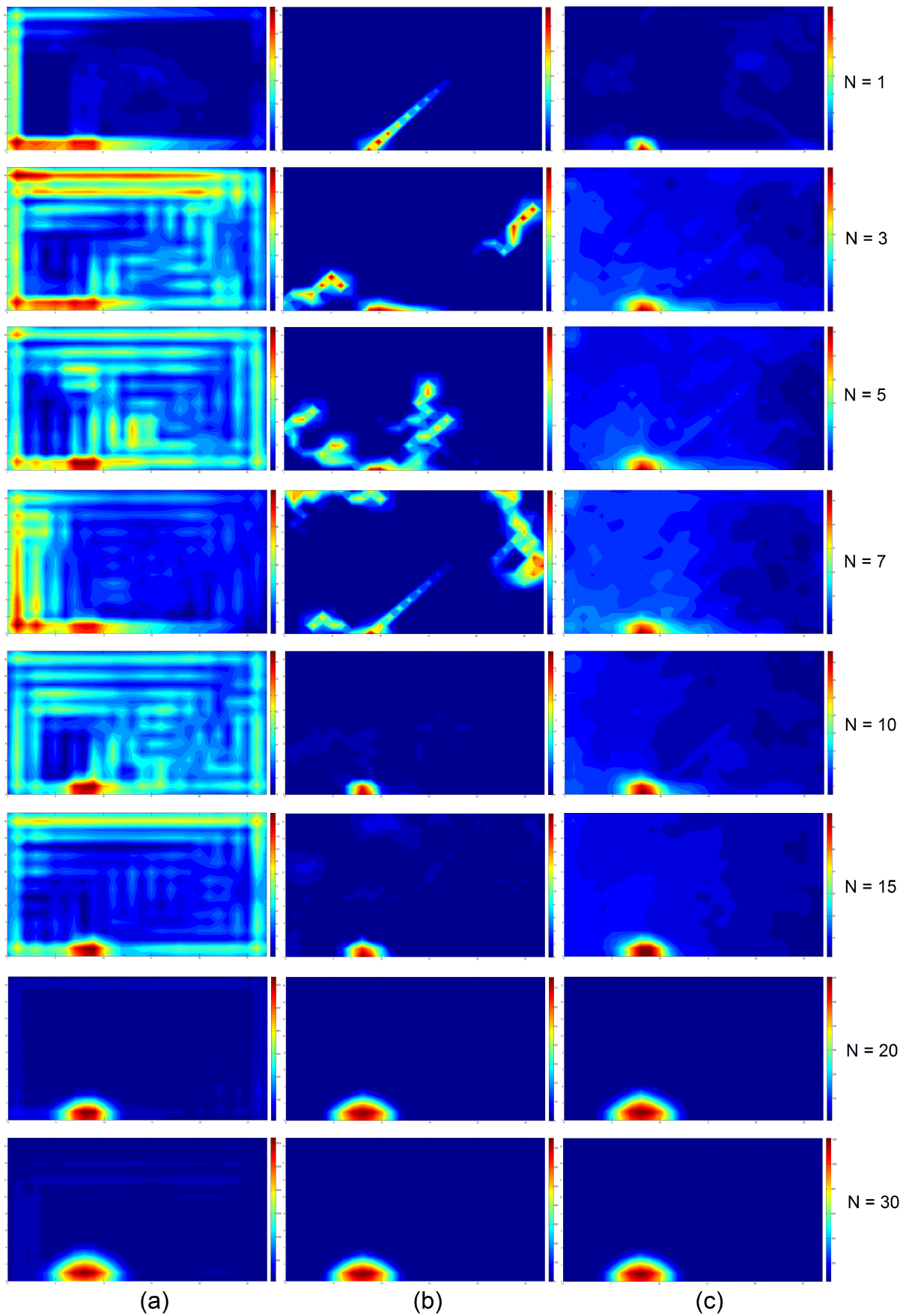


Figura 6.9: Mapas de temperatura para cada uma das abordagens de  $\beta, \delta'$  utilizando-se  $N = \{3, 6, 9, 12, 15, 18, 21\}$  robôs.

A quantidade de feromônio de cada célula  $x_{ij}$  foi capturada no  $T$ -ésimo passo para cada  $\delta', \beta$  com  $N$  robôs. A Figura 6.9 mostra os gráficos de temperatura para  $\delta', \beta$  utilizando diferentes quantidades  $N$  de robôs. Os locais vermelhos do mapa de temperatura têm níveis elevados de feromônios, enquanto as células azuis escuras têm níveis baixos ou nulos de feromônio, em comparação com as outras áreas. É possível observar na Figura 6.9a, usando  $N = \{1, 3, 5, 7, 10, 15\}$  robôs, que os reticulados finais não tem uma variação significativa na  $x_{ij}$  valor, porque todas as células possuem níveis elevados de feromônios (células azuis claros e amarelos claros), exceto para as células vermelhas, próximas aos ninhos do ambiente.

Por outro lado, a Figura 6.9b mostra que apenas as células recentes tem feromônio depositado. Estas situações não são adequados para uma boa estratégia de desempenho porque transforma a escolha do movimento em um processo aleatório, uma vez que as células têm quase a mesma probabilidade de serem escolhidas, não levando em conta as células previamente visitadas pelo enxame.

As melhores taxas de deposição/evaporação foram os utilizadas nas simulações relacionadas às Figuras 6.9c, porque as células têm uma boa variabilidade nas escalas de cores e todas as células foram visitadas pelos robôs. O tamanho da equipe  $N$  adequada para desempenhar a tarefa é uma característica importante a ser considerada. Uma vez que existem  $F = 30$  unidades de alimentos, a tarefa poderia ser resolvida colocando um robô para cada alimento.

Portanto, quando o  $N$  foi expressivamente aumentado, o desempenho da equipe foi negativamente afetado. Em todos os casos, usando um robô ( $N = 1$ ). Foi possível notar que o uso de apenas um robô no ambiente não pôde ser completamente reforçado com taxas as taxas de feromônios utilizadas. Além disso, a Figura 6.9 mostra o gráfico resultante a temperatura utilizando feromônios  $N = \{3, 5, 7, 10, 15\}$  robôs da equipe.

Nas Figuras 6.9a e 6.9b a taxa de feromônio capturado na última etapa mostra a que a taxas de difusão e declínio atrapalham a equipe. A Figura 6.9c indica que o desempenho da equipe é bom, especialmente por causa da distribuição de  $\delta', \beta$  taxas no meio ambiente. Embora tenhamos um bom desempenho da equipe com estes  $\delta', \beta$  taxas na Figura 6.9c com uma grande distribuição sobre o ambiente (não considerando a área perto dos ninhos), o número de robôs ideais, de acordo com as simulações, é entre  $N = \{7, 10\}$ . Isso deve-se essencialmente por causa da distribuição de cores ótimas sobre o ambiente de simulação, que indica que todo o ambiente recebeu uma boa exploração recente. Por fim, conclui-se que o tamanho da equipe apropriado para esta configuração do ambiente usa  $N = \{7, 10\}$  robôs com  $\delta' = 0.25$  e  $\beta = 0.2$ .



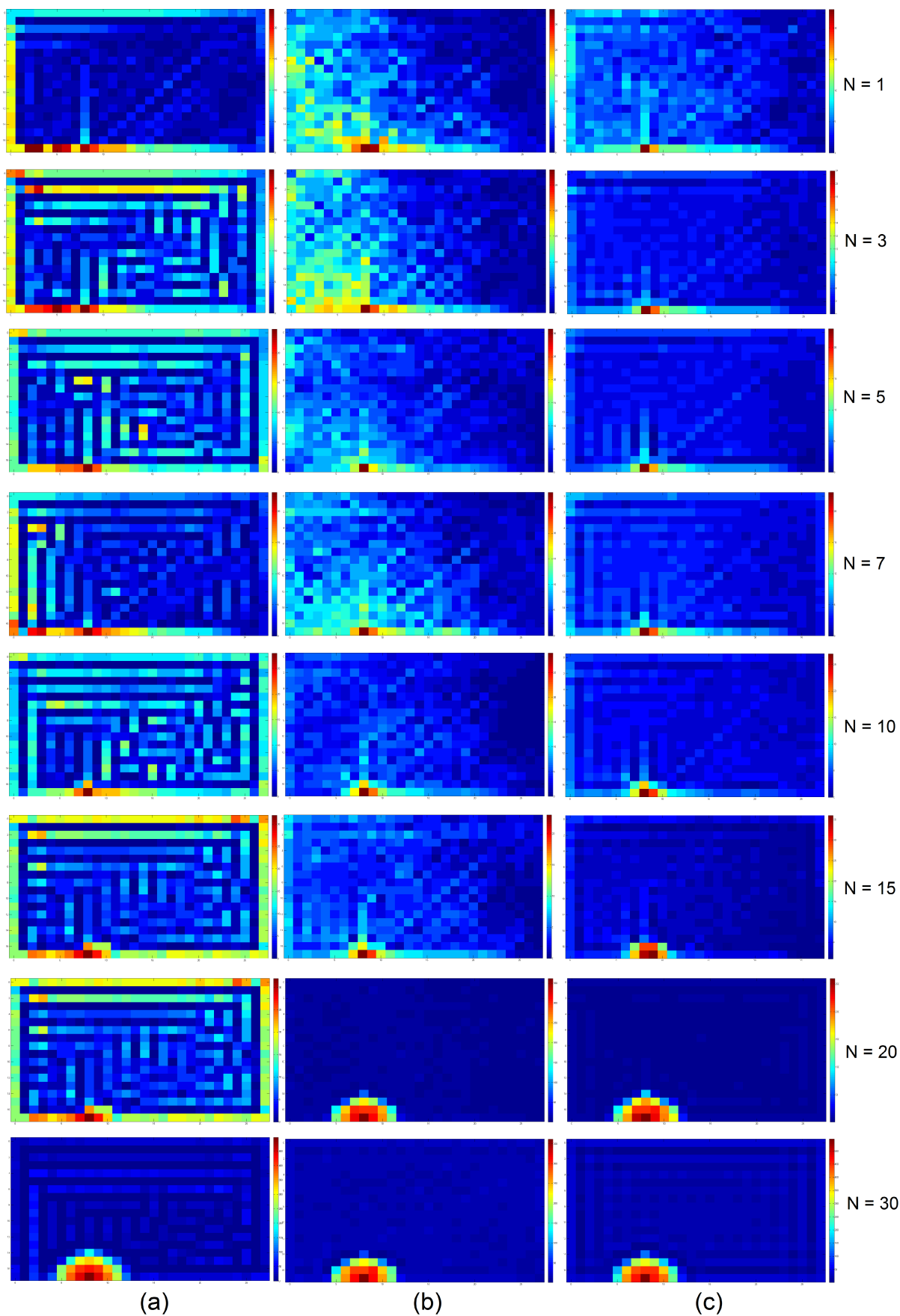


Figura 6.8: Mapas de passos por célula para cada uma das abordagens de  $\beta, \delta'$  utilizando-se  $N = \{3, 6, 9, 12, 15, 18, 21\}$  robôs.

## 6.6 Análise da complexidade dos métodos SCAAM e FSCAM

Embora não seja apresentado neste trabalho, vislumbramos também ser possível construir grafos para modelar as tarefas de coleta seletiva e de busca e resgate, modelados através dos algoritmos SCAAM e FSCAM. Dessa forma, acreditamos que com as devidas adaptações necessárias aos dois algoritmos, os resultados alcançados através da modelagem do problema da busca na tarefa de forrageamento poderão ser ampliados para os algoritmos SCAAM e FSCAM. Deixamos essa análise como proposta para trabalhos futuros que venham a dar continuidade à presente investigação.

---

## Experimentos e análise de resultados

---

Neste capítulo serão apresentados os principais resultados obtidos a partir da implementação dos diversos modelos apresentados no Capítulo 5. Dois tipos de simulação foram realizadas. A primeira foi realizada em ambientes específicos elaborados na linguagem de programação C padrão e C++. A segunda foi realizada no ambiente de simulação Webots EDU, que por ser voltado à simulação robótica, leva em conta diversos aspectos físicos do ambiente e da arquitetura utilizada (e-Pucks) e retorna uma simulação mais próxima da realidade. Neste capítulo, os resultados da tarefa de forrageamento são apresentados e discutidos detalhadamente. Os primeiros experimentos foram elaborados utilizando-se o ambiente de simulação na linguagem de programação C. Posteriormente, são discutidos os experimentos para a tarefa de coleta seletiva, e em seguida para a tarefa de busca e resgate, em um ambiente elaborado na linguagem de programação C++. Por fim, a tarefa de forrageamento foi utilizada para a simulação no ambiente Webots EDU e os principais resultados obtidos são apresentados.

### 7.1 Considerações iniciais

Todos os experimentos nos ambientes implementados na linguagem de programação C foram realizados a partir de  $10^4$  simulações, exceto nos casos em serão especificados separadamente. Todos os testes usam um campo de piso estático para uma sala  $20 \times 30$  células representadas na Figura 7.1. Existem dois ninhos; eles estão na parede à esquerda e na parede inferior e correspondem às células com valor 1 na figura. As células que representam as paredes/obstáculos são atribuídas com o valor 500, uma vez que este valor



Os primeiros parâmetros ajustados foram as constantes que determinam a difusão e o declínio do feromônio no processo de *searching*. Estes parâmetros são importantes para evitar a formação de filas e aglomerações.

### 7.2.1 Resultados de refinamento do modelo CAAM

Inicialmente, um teste foi realizado a fim de explorar o modelo de busca baseado em ACs para a tarefa de forrageamento sem utilização da memória para guardar o caminho percorrido por cada robô. Ou seja, a política de fila  $Q$  descrita na seção 5.1.1 não foi empregada ( $Q = 0$ ). Neste teste, foram realizados experimentos denominados  $E_1$ ,  $E_2$ ,  $E_3$  e  $E_4$  com 7 robôs e 6 objetos a serem recolhidos (comida), representados por  $N = 7$  e  $F = 6$ . Todos os experimentos utilizaram a constante de difusão da célula central  $\delta = 1$  e a constante de difusão da vizinhança  $\delta'$  foi variada avaliando-se 4 valores possíveis de  $\delta' = \{0.9, 0.75, 0.5, 0.25\}$ . Cada experimento  $E_i$  possui 16 combinações diferentes  $\{C_1, C_2, C_3, \dots, C_{16}\}$ , sendo que cada combinação representa um par  $(\delta', \beta)$  diferente, onde  $\beta$  é a constante de declínio do feromônio. Conforme será detalhado a seguir um diferente conjunto de valores de  $\beta$  foi avaliado, dependendo do raio de visão  $r_v$  do robô utilizado. O motivo dessa especificação refere-se à resultados de experimentos preliminares onde observou-se que quando o robô não possui raio de visão dos conteúdos, é importante utilizar valores de  $\beta$  mais suaves. Ou seja, a ausência de visão faz com que seja importante uma taxa de evaporação do feromônio mais baixa, pois essa informação tem maior relevância na busca no robô.

O primeiro experimento denominado  $E_1$  refere-se à variação que utiliza  $\alpha = 0$  (sendo  $\alpha$  a constante que pondera a importância da informação do piso dinâmico e raio de visão  $r_v = 0$ ). Além disso, nesse experimento, a constante de declínio do feromônio ( $\beta$ ) foi variada e avaliada a partir de quatro valores possíveis  $\beta = \{0.2, 0.25, 0.3, 0.5\}$ . Assim, temos que a primeira combinação  $C_1$  de  $E_1$  refere-se à  $C_1 = (\delta' = 0.9, \beta = 0.2)$ . A combinação  $C_2$  de  $E_1$  refere-se à  $C_2 = (\delta' = 0.75, \beta = 0.2)$ . A combinação  $C_{16}$  de  $E_1$  refere-se  $C_{16} = (\delta' = 0.25, \beta = 0.5)$ , e assim por diante. Em todas as combinações de  $E_1$  temos os seguintes valores fixos:  $\delta = 1$ ,  $\alpha = 0$  e  $r_v = 0$ .

O experimento denominado  $E_2$  refere-se à variação de  $\alpha = 0$  e raio de visão  $r_v = 1$ . Nesse segundo experimento, os valores de  $\beta$  avaliados foram  $\beta = \{0.01, 0.05, 0.1, 0.15\}$ . Assim, temos que a primeira combinação de  $E_2$  refere-se à  $C_1 = (\delta' = 0.9, \beta = 0.01)$  e a última combinação do experimento  $E_2$  refere-se à  $C_{16} = (\delta' = 0.25, \beta = 0.15)$ . Em todas as 16 combinações, temos  $\delta = 1$ ,  $\alpha = 0$  e  $r_v = 1$ .

O experimento denominado  $E_3$ , refere-se à variação de  $\alpha = 1$  e sem raio de visão ( $r_v = 0$ ). A constante de declínio sendo variada dentre os valores  $\beta = \{0.2, 0.25, 0.3, 0.5\}$ . Por exemplo,  $C_4 = (\delta' = 0.25, \beta = 0.2)$  com  $\delta = 1$ ,  $\alpha = 1$  e  $r_v = 0$ .

Por fim, o experimento denominado  $E_4$ , refere-se à variação de  $\alpha = 1$  e com raio

de visão ( $r_v = 1$ ). A constante de declínio foi variada:  $\beta = \{0.01, 0.05, 0.1, 0.15\}$ . Por exemplo,  $C_4$  refere-se à  $C_4 = (\delta' = 0.25, \beta = 0.01)$  de  $E_4$  e  $C_{16}$  refere-se à  $C_{16} = (\delta' = 0.25, \beta = 0.15)$  do experimento  $E_4$ , tendo os valores fixos:  $\delta = 1$ ,  $\alpha = 1$  e  $r_v = 1$ .

A Figura 7.3 apresenta os resultados médios desses 4 experimentos  $\{E_1, E_2, E_3, E_4\}$  sendo que o eixo  $x$  corresponde às 16 combinações avaliadas e o eixo  $y$  corresponde ao número total de iterações necessárias para que o time executasse toda a tarefa: recolher os  $F = 6$  objetos e depositá-los em um dos ninhos. Percebe-se que os experimentos com raio de visão  $r_v = 1$  foram superiores em relação aos testes realizados onde a constante  $r_v = 0$ . Além disso, pode-se notar que a variação adotada neste modelo utilizando-se campo de piso dinâmico, é superior em relação à variação que utiliza apenas piso estático. Grande parte dessa melhoria refere-se à diminuição de formação de filas durante o processo de homing, o que facilita a escolha de outros caminhos, mesmo que mais longos, porém com menos atraso devido à aglomeração. Além disso, a combinação  $C_4$  teve um melhor desempenho durante a execução para os quatro experimentos  $E_i$ . Neste caso, nota-se que uma taxa de declínio mais suave, proporciona um melhor espalhamento dos robôs no ambiente de piso, e conseqüentemente, um melhor desempenho de busca. Nos experimentos sem visão ( $r_v = 0$ ) é possível observar que taxas de  $\delta'$  mais suaves tendem a convergir para uma média de iterações mais baixas, quando combinadas com valores de  $\beta$  também suaves. Esse fenômeno ocorre devido ao fato de que quando não há visão, não podemos agregar uma punição (valor de célula mais alto) no campo de piso. Por outro lado, quando temos visão ( $r_v = 1$ ), significa que a vizinhança é observada e verificada, o que resulta em valores mais altos para  $\delta'$  como sendo os de melhor desempenho para a diminuição da média das iterações para a conclusão da tarefa.

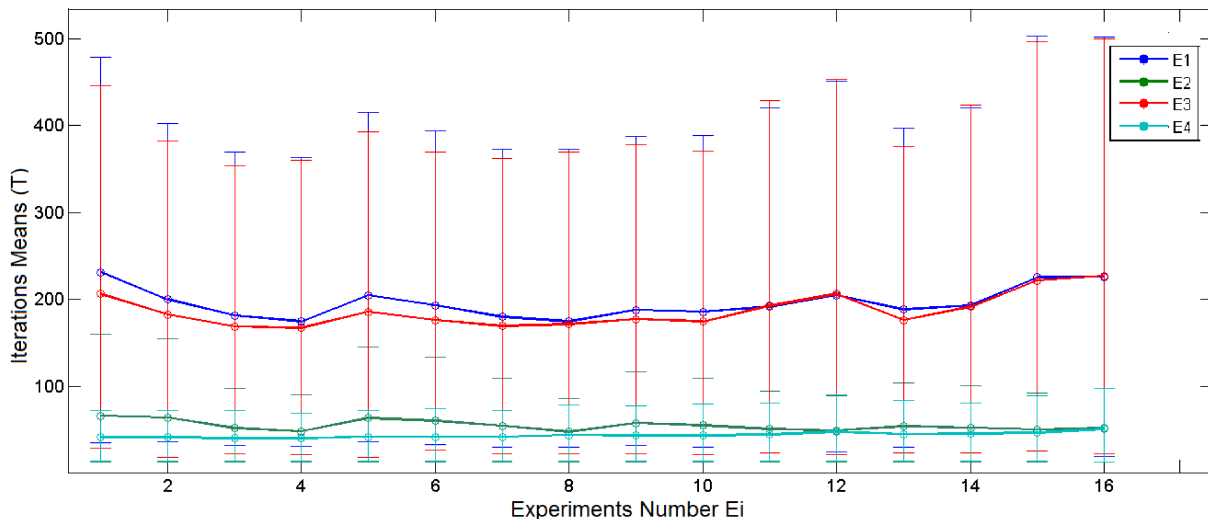


Figura 7.3: Número médio de execuções para executar o forrageamento com  $N = 6$  robôs,  $F = 7$  unidades de comida e  $\delta = 1$ .  $E_1 = \{\alpha = 0, r_v = 0\}$ ,  $E_2 = \{\alpha = 0, r_v = 1\}$ ,  $E_3 = \{\alpha = 1, r_v = 0\}$  e  $E_4 = \{\alpha = 1, r_v = 1\}$ .

Os piores resultados podem ser observados nos extremos, ou seja, quando colocamos uma taxa de difusão de feromônio na vizinhança ( $\delta'$ ) alto e uma taxa de declínio ( $\beta$ ) muito

suave, o que significa que o modelo ignora casos onde existe uma comida mais próxima do robô, no entanto ele não pode observá-la devido à evaporação lenta do ferômonio nos seus vizinhos. Outro caso que não obteve um resultado significativo foi a variação em que  $\delta'$  é alto e a taxa de declínio  $\beta$  também é alta, neste caso, ocorre que a informação global sobre os caminhos já percorridos deixados no campo de piso de busca pelo enxame é perdida, o que acarreta em uma perda de informação que acaba gerando uma diminuição no desempenho do enxame para a tarefa de forrageamento.

## 7.2.2 Variação do tamanho da fila

O segundo tipo de experimento realizado com o modelo CAAM tem por objetivo avaliar o quanto o tamanho da memória associada ao modelo melhora a eficiência do mesmo. Para tal, uma memória finita com política de fila é associada a cada robô, evitando que o mesmo percorra caminhos que já foram previamente visitados por ele, ou seja, a memória do robô registra em uma fila, quais foram as últimas células visitadas.

Foram realizados experimentos com 10 robôs ( $N = 10$ ) e 30 objetos ( $F = 30$ ) a serem recolhidos pelo time de robôs. Todos os experimentos possuem as constantes  $\delta = 1$ ,  $\alpha = 0$  e  $r_v = 0$  utilizados no experimento  $E_1$  descrito na Seção 7.2.1 e utilizou-se a melhor variação  $C_4 = (\delta' = 0.9, \beta = 0.15)$  e a pior variação  $C_{16} = (\delta' = 0.25, \beta = 0.15)$ . Para avaliar a interferência da utilização da memória no desempenho do time foram realizados 8 experimentos nomeados  $\{Cq_1, Cq_2, Cq_3, \dots, Cq_8\}$ , que se referem ao tamanho da fila avaliada  $Q = \{0, 5, 10, 15, 30, 50, 100, 200\}$ . Os resultados deste experimento podem ser visualizados na Figura 7.4, sendo que o eixo  $x$  corresponde às 8 combinações avaliadas e o eixo  $y$  corresponde ao número total de iterações necessárias para que o time executasse toda a tarefa de recolher os 30 objetos e depositá-los nos ninhos. Neste gráfico, temos duas variações: a variação 1 representa a combinação  $C_{16}$  e a variação 2 representa a combinação  $C_4$ .

Podemos perceber que o aumento do tamanho da fila é diretamente proporcional ao desempenho do time de robôs. A combinação  $C_{16}$  apresentou melhor resultado ao variar as filas, uma vez que o coeficiente de variação foi de 14.55%, já para a combinação  $C_4$  o coeficiente de variação foi de apenas 5.22%. Entretanto, cabe ressaltar que a combinação  $C_{16}$  difere-se ao pior ajuste de  $\delta'$  e  $\beta$  do experimento  $E_1$  (Seção 7.2.1), enquanto que  $C_4$  refere-se ao melhor ajuste avaliado. Portanto, os resultados da Figura 7.4 podem ser interpretados como não sendo tão significativo utilizar-se uma fila quando os parâmetros  $(\delta', \beta)$  do modelo já foram previamente refinados. Entretanto, como muitas vezes não é possível realizar esse refinamento prévio, esse experimento mostra que o uso da fila torna o método mais eficiente ainda que os parâmetros  $\delta'$  e  $\beta$  relacionados à difusão e evaporação do feromônio não estejam funcionando bem ajustados.

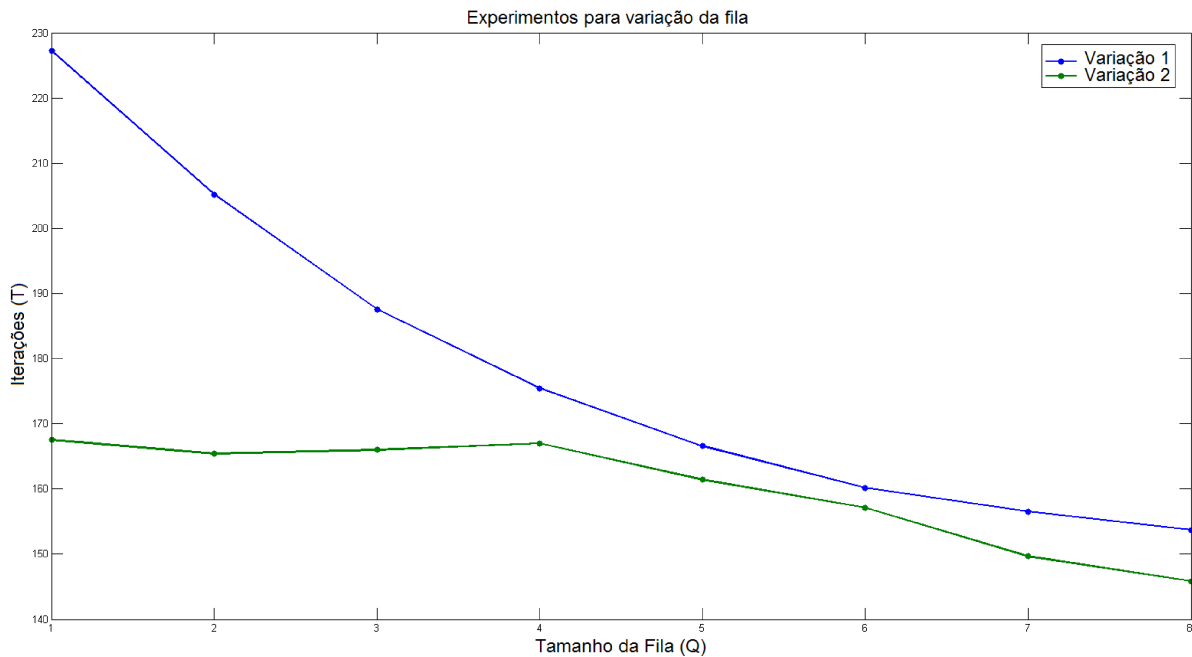


Figura 7.4: Gráficos para determinar as diferenças de médias entre o tamanho das filas para dois tipos de simulações referentes à combinação  $C_4$  e  $C_{16}$ .

### 7.2.3 Explorando a configuração do ambiente de forrageamento

Os experimentos descritos nesta seção têm como objetivo configurar o ambiente de simulação dos robôs, para que seja possível estimar as melhores características do modelo em relação a situações específicas. Estes experimentos também foram realizados a partir de  $10^4$  simulações, exceto nos casos em que estarão comentados separadamente. O tamanho da fila utilizado foi  $Q = 10$  e foi utilizada a configuração  $C_4$  que foi considerada a melhor configuração do experimento  $E_4$  da seção 7.2.1,  $\delta = 1$ ,  $\beta = 0.01$ ,  $\delta' = 0.25$ ,  $\alpha = 1$  e  $r_v = 1$ . Todas simulações desta seção também usam um campo de piso estático para uma sala de  $20 \times 30$  células representadas na Figura 7.1.

#### Quantidade de robôs e comida

O objetivo deste experimento é avaliar o quanto a variação de comida no ambiente e a quantidade de robôs podem afetar o desempenho do enxame em relação à quantidade de tempo gasta para a realização da tarefa. É importante efetuar uma análise sobre a quantidade de robôs em relação à quantidade de comida distribuída pelo ambiente, uma vez que dependendo do total de robôs a serem utilizados pode aumentar excessivamente o gargalo próximo aos ninhos, ou ainda, esse total pode estar aquém do tamanho de enxame suficiente para capturar toda a comida em um tempo satisfatório.

Para se investigar a relação entre a quantidade de robôs e o tempo total de conclusão da tarefa, foram realizados 8 experimentos  $\{Cn_1, Cn_2, Cn_3, \dots, Cn_8\}$  onde variou-se a quantidade de robôs,  $N = \{1, 3, 6, 9, 12, 15, 18, 21\}$  e a quantidade de comida foi mantida fixa ( $F = 30$ ). Como pode-se observar através dos boxplots apresentados na



Figura 7.5 os símbolos “+” representam as médias com intervalos de confiança para as médias de 95%. O eixo  $x$  refere-se ao número de robôs e o eixo  $y$  refere-se ao número de iterações médias. Neste caso, é possível perceber que o desempenho de um único robô ( $N = 1$ ) é muito discrepante em relação aos demais resultados observados e retorna um número de iterações muito superior em relação aos demais casos analisados, uma vez que um único robô não utiliza todos os recursos do ambiente (tais como os dois ninhos) e nem a vantagem de um comportamento coordenado. Ao triplicarmos a quantidade de robôs  $N = 3$  o desempenho é 4 vezes superior em relação ao experimento com um robô. À medida que a quantidade de robôs aumenta, pode-se verificar que o desempenho é aumentado, até que em  $N = 9$  temos um desempenho ótimo, dentre os valores avaliados. A partir desse ponto o desempenho do time vai se degradando retornando um tempo cada vez maior, apesar de utilizar mais recursos do sistema. Um dos principais motivos dessa degradação é devido ao fato de que em um ambiente com vários robôs é criado um efeito de acúmulo (jamming effect) de robôs próximo aos ninhos (Alizadeh, 2011). Isso provoca um arqueamento impróprio para um robô sem objetos que fica preso na presença de robôs com objetos. Além disso, podemos observar que ao aumentar em muito a quantidade de robôs acabamos criando um efeito de sobrecarga dos ninhos (clogging effect) (Song et al., 2006), conforme observado na Figura 7.6.

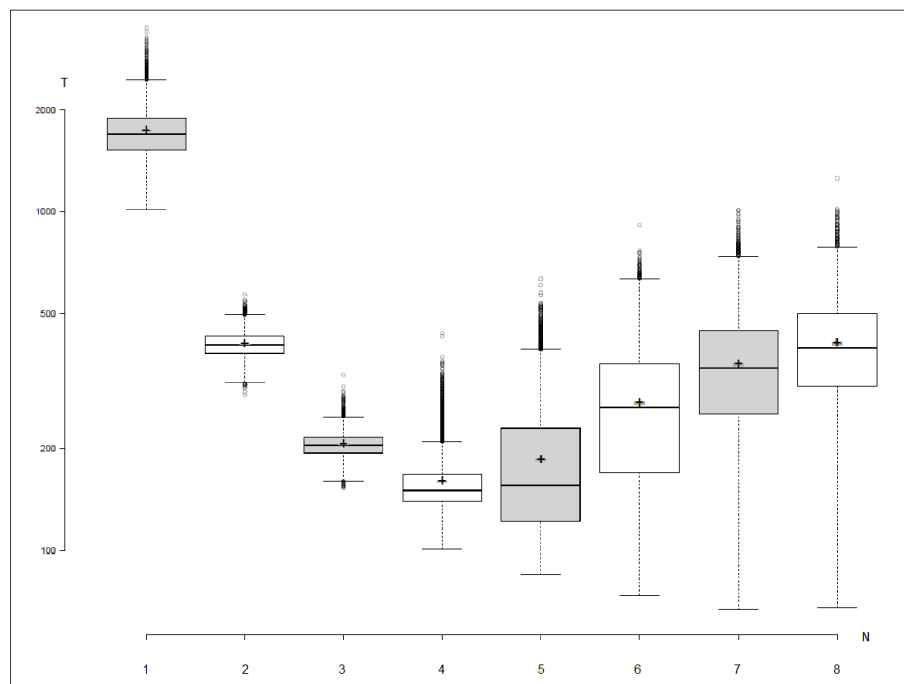


Figura 7.5: Boxplots para determinar as diferenças entre diferentes números de robôs e iterações no ambiente.

O coeficiente de correlação entre a quantidade de robôs  $N$  e a quantidade média de iterações  $T$  é de 0.4861 no intervalo de  $3 \leq N \leq 21$ . Assim, foi possível realizar a regressão polinomial dos valores obtidos, podemos dizer que os dados estão relacionados da seguinte forma, tal que  $x$  representa o número de robôs  $N$  e  $3 \leq x \leq 28.176$  com

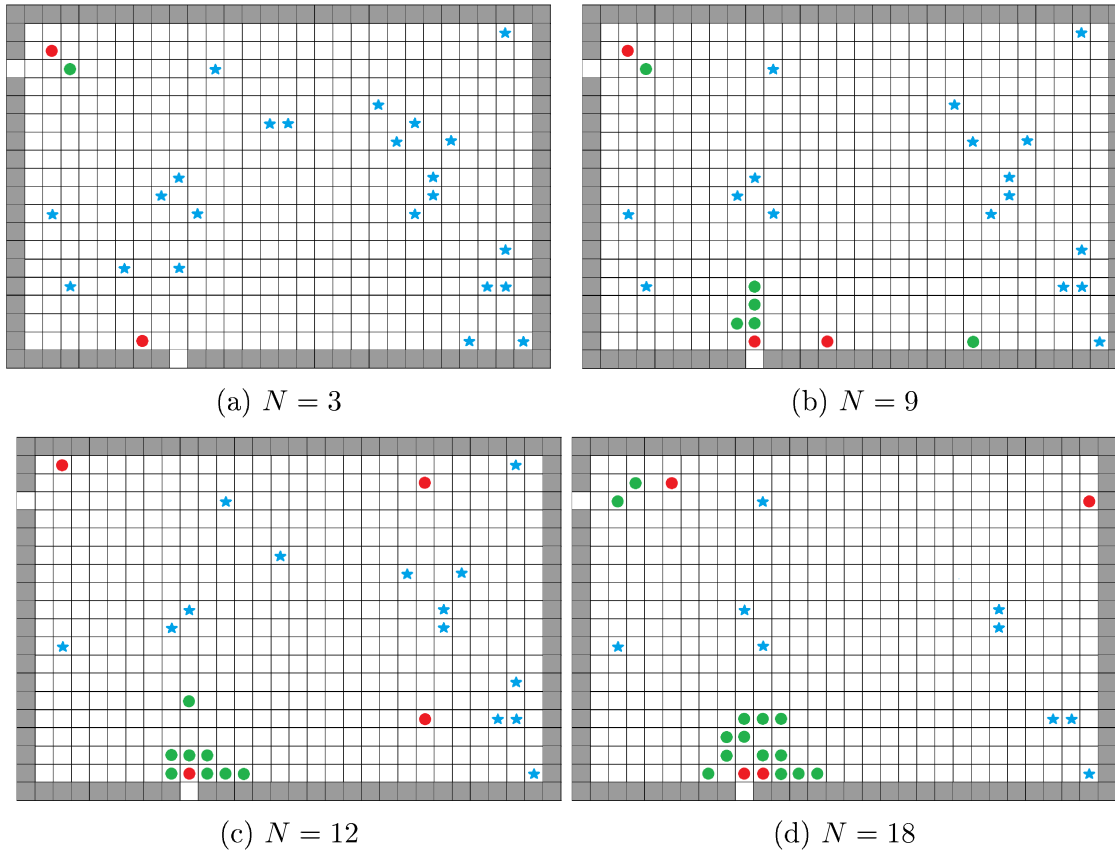


Figura 7.6: Efeito de sobrecarga dos ninhos (clogging effect), à medida que aumentamos a quantidade de robôs  $N = \{3, 9, 12, 18\}$ . Robôs vermelhos não carregam alimento (estrelas em azul), robôs em verde carregam alimento.

intervalo de confiança de 95%, e  $y_1$  representa a média da quantidade de iterações  $T$  gastas para realizar o experimento, tal que  $43.672 \leq y_1 \leq 1640.026$  está 95% no intervalo de confiança. A função linear que expressa a correlação é dada na Equação 7.1. A reta que expressa a função da função que da Equação 7.1 corta diagonalmente o gráfico. Os pontos que pertencem ao intervalo  $1 \leq x \leq 256$  foram os pontos plotados a partir da função apresentada na Equação 7.1 e os pontos pertencentes ao intervalo  $1 \leq x \leq 21$  correspondem aos pontos coletados no experimento apresentado na Figura 7.7.

$$y_1 = 16.0865x + 47.4392 \quad (7.1)$$

Em virtude do comportamento do modelo com regiões de declínio e crescimento, outra regressão foi feita para encontrar o valor do número ótimo de robôs  $N$  para este cenário de simulação e está representada pela reta  $y_2$ . A Equação 7.2 representa a regressão que corresponde aos valores de  $1 \leq x \leq 9$  robôs. O ponto de intersecção  $(7.4419, 167.1535)$  entre as funções  $y_1$  e  $y_2$  que representa o número ótimo de robôs ( $N \approx 7$ ). Os pontos do intervalo  $1 \leq x \leq 21$  corresponde aos pontos médios de iterações coletados através das

$10^4$  simulações.

$$y_2 = -170.5x + 1436.0 \quad (7.2)$$

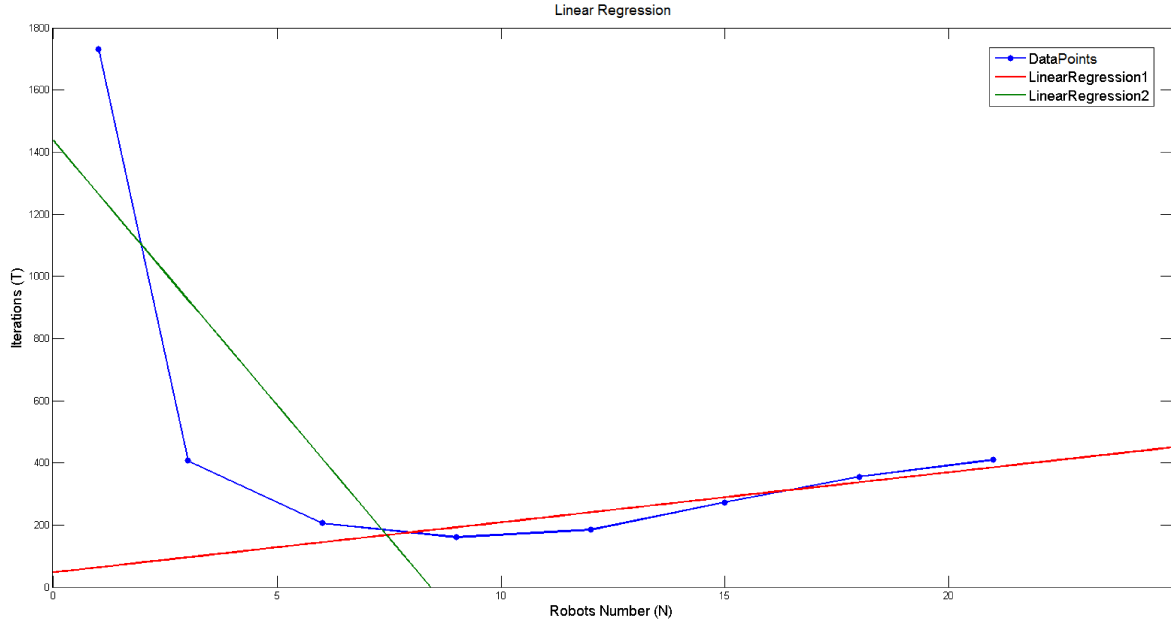


Figura 7.7: Regressão linear para robôs entre iterações.

Na segunda investigação realizada a variação na quantidade de comida foi avaliada para um número fixo de robôs. Neste caso, é importante determinar como a quantidade de comida pode influenciar o tempo total para a conclusão da tarefa de forrageamento. Para este experimento foram utilizados  $N = 10$  robôs e a quantidade de comida foi avaliada em 8 valores distintos em  $F = \{2, 4, 8, 16, 32, 64, 128, 256\}$  e os experimentos nomeados como  $\{Cf_1, Cf_2, Cf_3, \dots, Cf_8\}$ . A Figura 7.8 apresenta o resultado do experimento de variação da quantidade de comida. Podemos observar os boxplots de cada experimento e o símbolo “+” representa a média com intervalo de confiança para as médias de 95%. Neste caso, o experimento mostrou que à medida que se aumenta a quantidade de comida, aumenta-se também a quantidade de iterações para conclusão da tarefa. A Equação 7.3 descreve tal relação, sendo que,  $x$  representa a variação de comida  $F$  e  $y_3$  representa a quantidade média de iterações  $T$  de cada experimento.

$$y_3 = 5.3965x + 7.3887 \quad (7.3)$$

O coeficiente de correlação entre a quantidade de robôs  $F$  e a quantidade média de iterações  $T$  é de 0.9994 no intervalo de  $1 \leq F \leq 256$ . Isso significa que os valores de  $F$  e  $T$  estão fortemente associados. Assim, foi possível realizar a regressão linear dos valores obtidos. Considerando-se tal que  $x$  representa o número de robôs  $N$  e  $1 \leq x \leq 26.993$  com intervalo de confiança de 95%, e  $y$  representa a média da quantidade de iterações  $T$  gastas para realizar o experimento, tal que  $5.209 \leq y \leq 5.584$  com um intervalo de

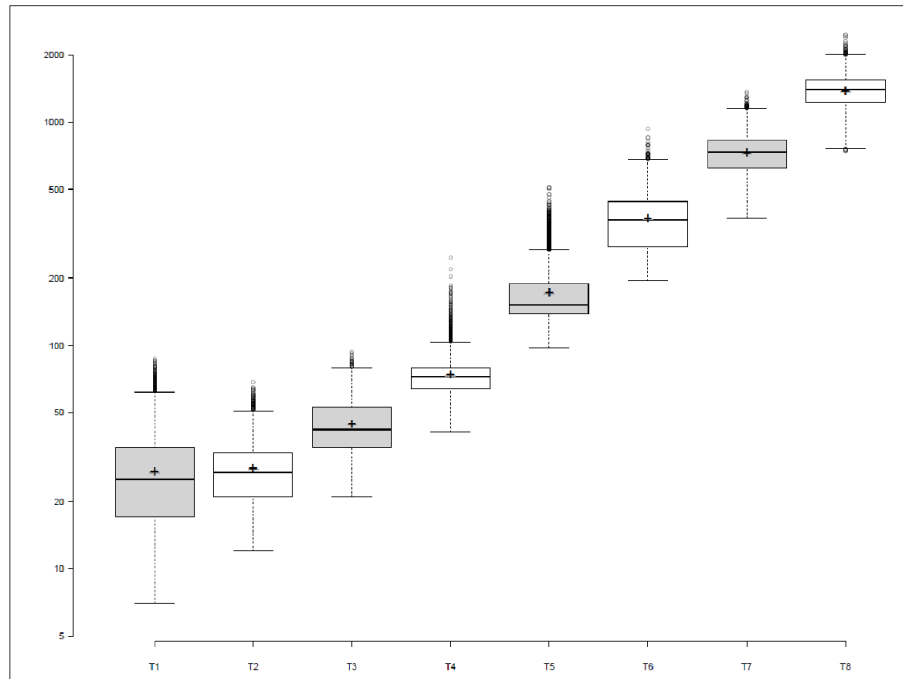


Figura 7.8: Boxplots para determinar as diferenças entre um número diferente de comida e iterações no ambiente.

confiança de 95%, a função linear que expressa a correlação é dada na Equação 7.1. A reta que expressa a função da função que da Equação 7.1 corta diagonalmente o gráfico. Os pontos que pertencem ao intervalo  $3 \leq x \leq 21$  foram os pontos plotados a partir da função apresentada na Equação 7.1 e os pontos pertencentes ao intervalo  $1 \leq x \leq 21$  correspondem aos pontos coletados no experimento apresentado na Figura 7.9.

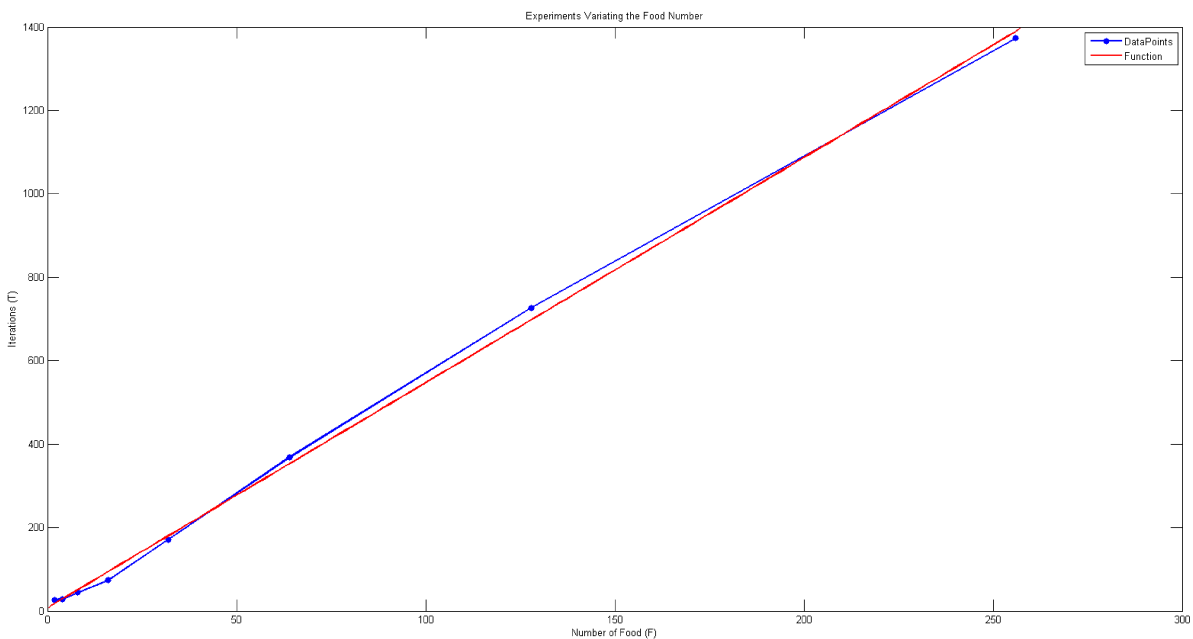


Figura 7.9: Regressão linear para variação de unidades de comida durante as iterações.

### Largura e posição dos ninhos

Experimentos de análise estrutural do ambiente de execução são relevantes quando o enxame vai executar tarefas que devem ser realizadas no menor tempo possível. Com esse tipo de experimento, é possível determinar qual é a melhor estrutura para a inserção do enxame de robôs.

Para investigar a relação entre a largura dos ninhos e o tempo para completar a tarefa de forrageamento, foram realizados 6 experimentos nomeados  $\{Cl_1, Cl_2, Cl_3, Cl_4, Cl_5, Cl_6\}$  alternando um único ninho. De acordo com a Figura 7.10, o experimento  $Cl_1 = \{1\}$  corresponde ao ninho com abertura na célula  $x_1$ . O experimento  $Cl_2 = \{1, 2\}$  corresponde ao ninho com abertura nas células  $x_1$  e  $x_2$ , já o experimento  $Cl_6 = \{1, 2, 3, 4, 5, 6\}$  corresponde ao ninho com abertura nas células  $x_1, x_2, x_3, x_4, x_5$  e  $x_6$ . Para cada variação de largura, também foi avaliado o desempenho de cada conjuntos com número diferente de robôs:  $N = \{3, 5, 7, 10, 15\}$ . A quantidade de comida foi mantida igual a  $F = 10$ .

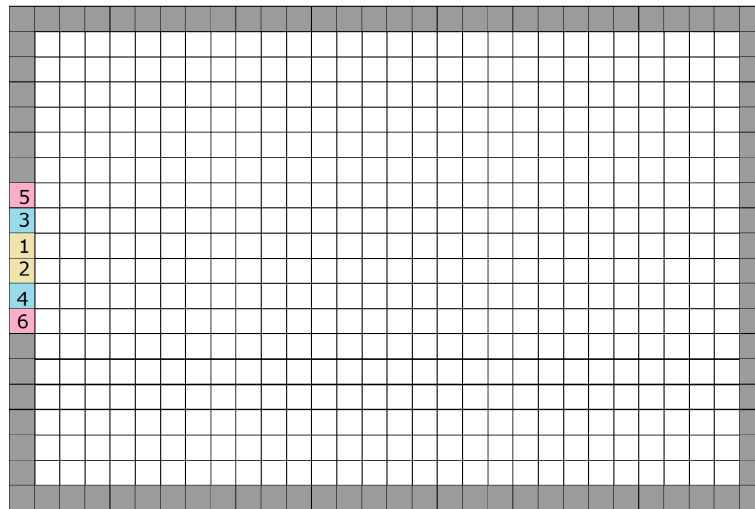


Figura 7.10: Variação de largura de ninhos no campo com  $20 \times 30$  células.

De acordo com o gráfico da Figura 7.11, ao incrementar a largura da porta, nota-se uma melhoria significativa no desempenho do time de robôs para executar a tarefa de forrageamento, diminuindo a quantidade de iterações necessárias para executar a tarefa. No entanto, para os tamanhos do time investigados, não é interessante aumentar a largura da porta em um valor maior que 2, pois embora exista uma queda no número de iterações, aumenta significativamente o tempo de processamento para o cálculo de cada uma das saídas, diminuindo, conseqüentemente o desempenho do sistema. Por exemplo, para  $N = 3$  com a abertura  $Cl_2$  são necessárias em média 248.10 iterações e o tempo total da simulação ( $10^4$  execuções) foi de 189 segundos. Por outro lado, ao usarmos a abertura  $Cl_6$ , embora o valor médio de iterações tenha caído para 217.17 o tempo total subiu para 312 segundos.

Uma segunda investigação a respeito do posicionamento dos ninhos foi realizada. Para tal, 10 experimentos foram realizados e nomeados  $\{Cr_1, Cr_2, Cr_3, \dots, Cr_{10}\}$ . Nesses ex-

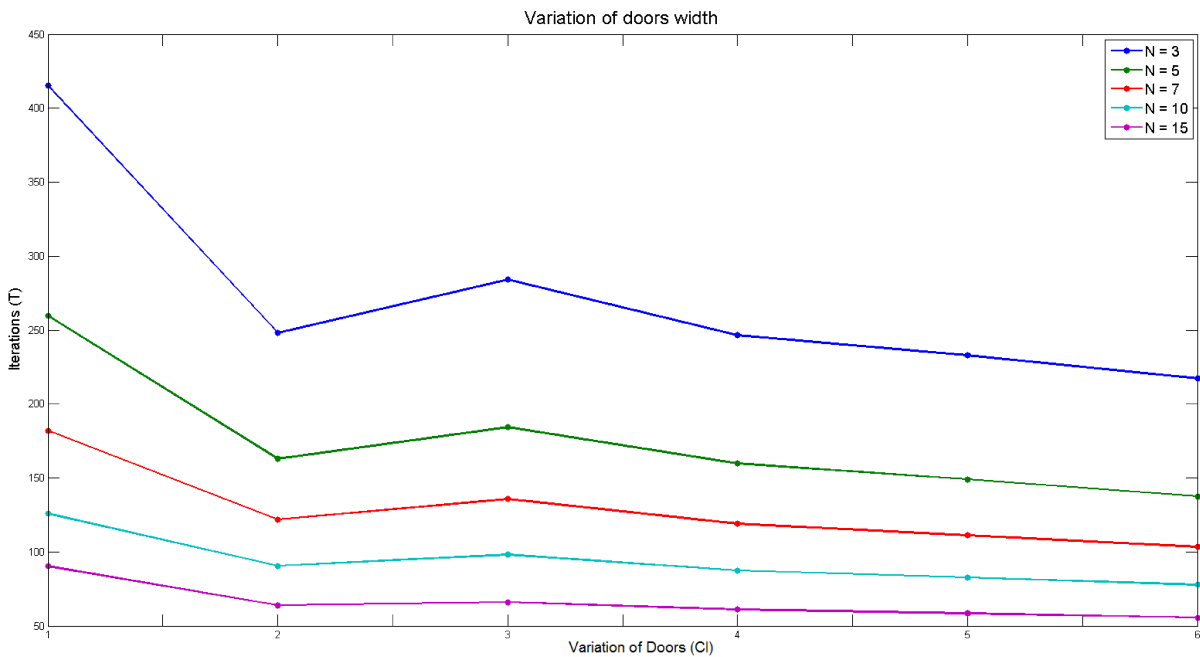


Figura 7.11: Gráficos para determinar as diferenças entre a largura das portas e o número de iterações.

perimentos, um ninho com abertura de 1 célula foi fixado em uma posição do ambiente, enquanto que um segundo ninho de 1 célula também foi avaliado, variando-se a sua posição entre 10 células da borda do ambiente, conforme indicado na Figura 7.12. De acordo com a Figura 7.10, o experimento  $Cr_1 = \{0, 1\}$  corresponde ao ninho com abertura nas células  $x_0$  e  $x_1$ . O experimento  $Cr_2 = \{0, 2\}$  corresponde ao ninho com abertura nas células  $x_0$  e  $x_2$ , já o experimento  $Cr_{10} = \{0, 10\}$  corresponde ao ninho com abertura nas células  $x_0$  e  $x_{10}$ . Para cada variação, também foram avaliados o desempenho do time usando diferentes conjuntos de robôs com  $N = \{3, 5, 7, 10, 15\}$ .

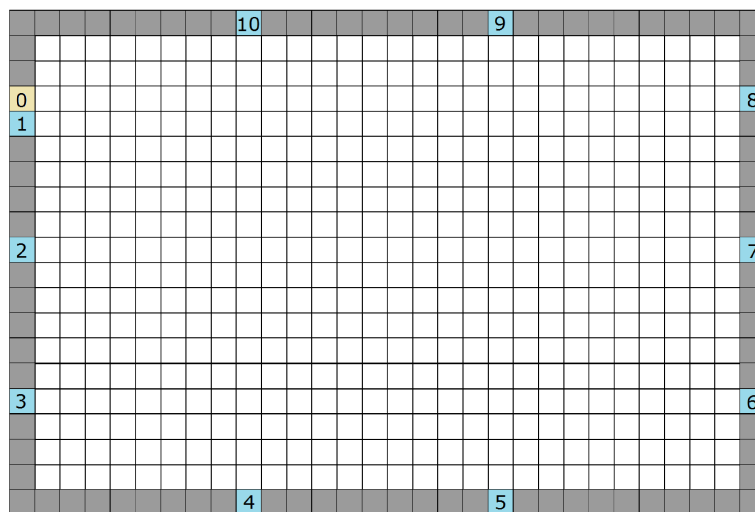


Figura 7.12: Variação de ninhos no campo de piso com  $20 \times 30$  células.

O gráfico da Figura 7.13 apresenta os resultados dos experimentos, sendo que o eixo  $x$  representa os diferentes posicionamentos da segunda abertura e o eixo  $y$  representa o

total de iterações para executar a tarefa. É possível observar que para a maioria dos conjuntos de robôs, ao se alterar as posições dos ninhos, a melhor posição para o segundo ninho é a posição  $Cr_9$  (resultado ótimo). Contudo, os resultados entre  $Cr_4$  e  $Cr_8$  foram bastante eficientes (sub-ótimos), a exemplo do resultado apresentado em  $Cr_5$  que é o segundo melhor resultado obtido. Os resultados que apresentaram o menor desempenho foram justamente as posições de ninhos muito próximas ao ninho fixo situado na célula  $x_0$  da Figura 7.12. Neste caso, há um aumento do efeito de colisões (*jamming*) (Li-jun et al., 2009) próximo aos ninhos testados acarretando um aumento no gargalo (Yamamoto et al., 2007) aumentando o arqueamento (Alizadeh, 2011) próximo aos ninhos durante o processo *depositing*. O pior resultado obtido é o  $x_3$ , uma vez que as duas saídas estão próximas e em regiões de quinas.

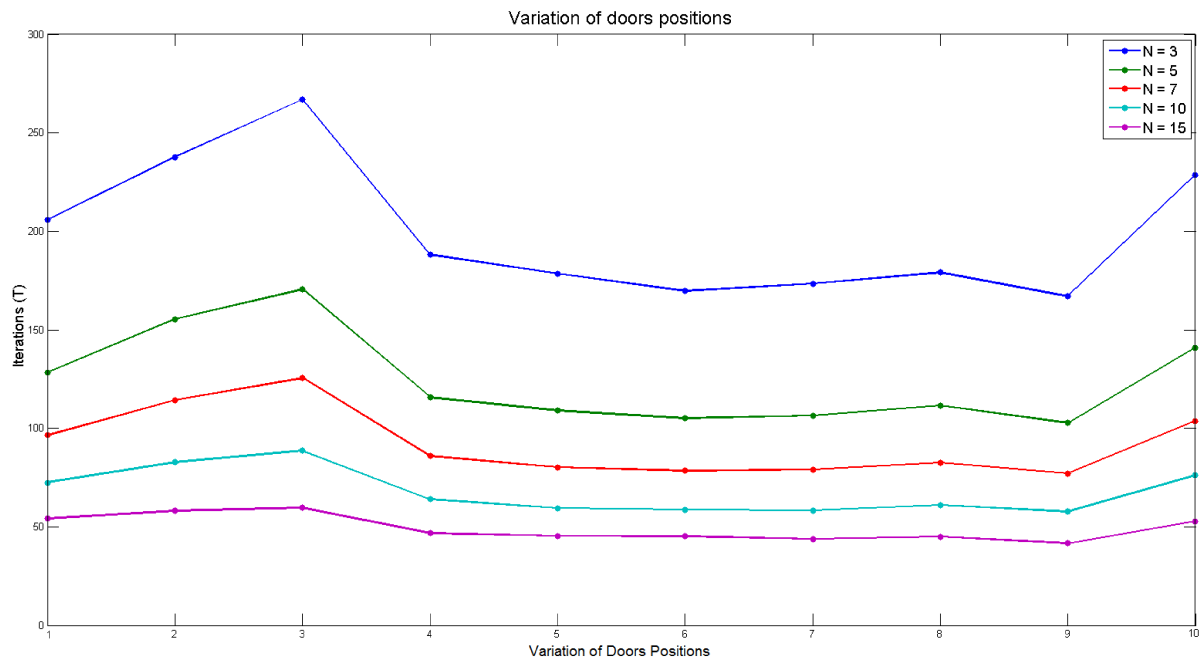


Figura 7.13: Gráficos para determinar as diferenças entre diferentes posições de pares de portas e número de iterações.

### Posicionamento da comida

O principal objetivo deste experimento é verificar o quanto a disposição da comida dentro do ambiente pode influenciar o desempenho do enxame. Neste experimento, também foram utilizadas as constantes da combinação de parâmetros  $C_4$  obtido no experimento  $E_4$  da seção 7.2.1 ( $\delta = 1, \delta' = 0.9, \beta = 0.15, \alpha = 1, r_v = 1$ ) e tamanho da fila  $Q = 10$ . Este experimento será similar ao experimento proposto em (Sugawara and Sano, 1997), onde foram alteradas as posições da comida em porcentagens de cobertura do ambiente. Foram avaliados 7  $\{Cp_1, Cp_2, Cp_3, Cp_4, Cp_5, Cp_6, Cp_7\}$  disposições diferentes de 9 objetos/comida ( $F = 9$ ) no campo de piso. Essas disposições são ilustradas na Figura 7.14.

Além disso, para cada disposição de comida, foram avaliados times de robôs de diferentes tamanhos  $N = \{3, 5, 7, 10, 15\}$  na tarefa de forrageamento.

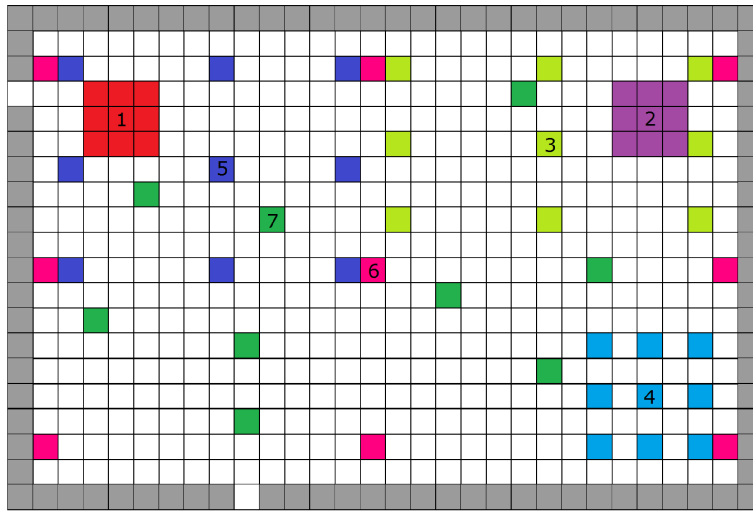


Figura 7.14: Diferentes disposições iniciais de comida ( $F = 9$ ) em um piso  $20 \times 30$ .

De acordo com o gráfico apresentado na Figura 7.15 é possível verificar novamente que a quantidade de robôs inserida no sistema aumenta o desempenho médio na tarefa de forrageamento, para todas as disposições analisadas. Além disso, nota-se que a posição dos recursos também influencia no desempenho do sistema. Nota-se que o melhor desempenho foi obtido no experimento  $Cp_5$ , onde os objetos encontram-se ocupando em torno de 25% do campo de piso e próximos aos dois ninhos. Os piores desempenhos encontram-se nas distribuições de comida mais distantes dos ninhos e mais concentradas, neste caso,  $Cp_2$  e  $Cp_4$ . Por outro lado, o time de robôs apresentou melhor desempenho quando os objetos estavam bem distribuídos ao longo do reticulado, ou quando estavam concentrados perto de um dos ninhos, como observado no experimento  $Cp_1$ . Uma das explicações para que os robôs apresentem melhores desempenhos com distribuições de comida mais perto dos ninhos é devido ao fato de que quando um robô deposita uma comida no ninho, ele já está apto novamente para buscar alimentos, o que acaba proporcionando uma facilidade para retornar ao ninho caso ele já encontre outro objeto. Os experimentos  $Cp_6$  e  $Cp_7$  possuem um bom desempenho, uma vez que a distribuição mais esparsa, facilita a busca e o espalhamento desses robôs ao longo do ambiente. Assim, ao fazer uma melhor cobertura do ambiente, esses robôs podem usufruir melhor dos objetos mais espalhados pelo ambiente, aumentando significativamente o desempenho do sistema. Além disso, o uso do feromônio invertido fez com que o time demore mais tempo para realizar buscas próximas a células já visitadas. O experimento  $Cp_3$  obteve um desempenho melhor que  $Cp_1$ , no entanto, por estar um pouco mais distante do ninho e não ter uma distribuição tão abrangente no campo de piso, este experimento, apresenta um desempenho inferior em relação aos experimentos  $Cp_5$ ,  $Cp_6$  e  $Cp_6$ , mas superior aos experimentos  $Cp_2$  e  $Cp_4$ . Os experimentos  $Cp_5$ ,  $Cp_6$  e  $Cp_6$ , por fazer com que o time de robôs usufrua da distribuição



esparsa da comida, acaba apresentando os melhores desempenhos.

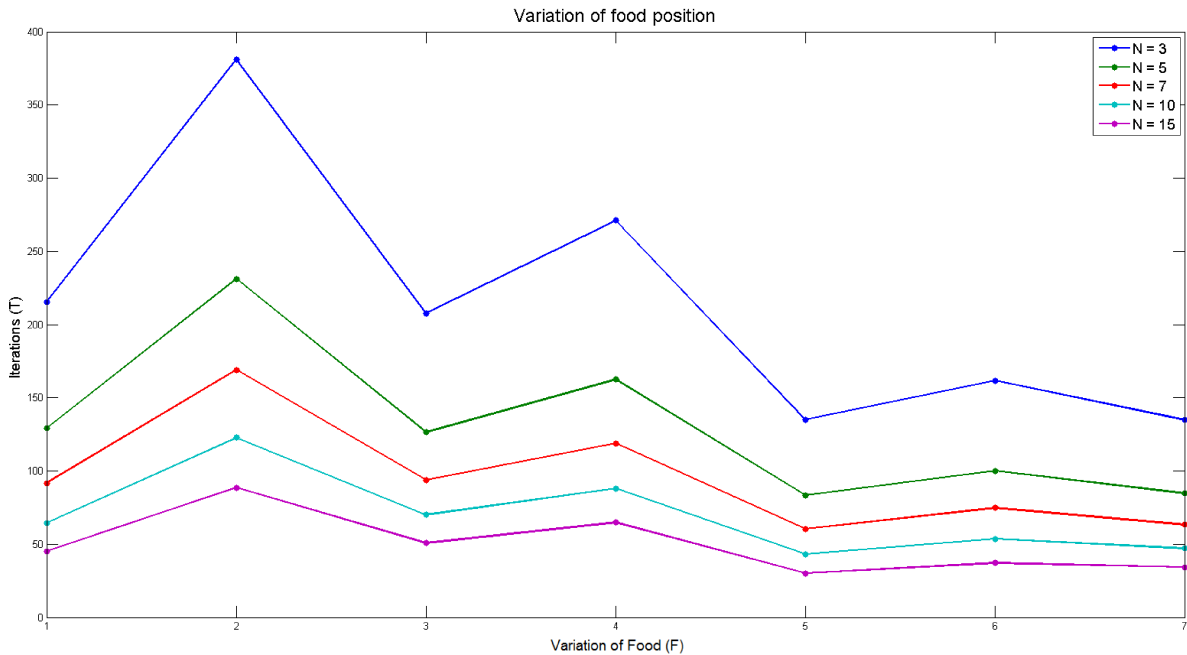


Figura 7.15: Resultados da variação do número de robôs, tempo e variação da comida no campo de piso.

#### 7.2.4 Exploração e cobertura do ambiente

O experimento apresentado nesta seção busca avaliar a cobertura do ambiente realizada pelos robôs, à medida que se aumenta o tamanho máximo o tamanho máximo da fila  $Q$  através de gráficos de temperatura similar ao teste proposto em (Calvo et al., 2011a). No entanto, naquele trabalho os autores propuseram avaliar a quantidade de feromônio deixada no ambiente. Neste trabalho, foi proposto avaliar as trajetórias realizadas pelos robôs no ambiente. Ou seja, verificou-se a cobertura e o nível de exploração atingido pelo enxame de robôs durante a tarefa de forrageamento registrando-se o número de vezes que cada célula foi visitada por algum dos robôs do time. Nas simulações efetuadas, os seguintes parâmetros foram mantidos fixos: números de robôs  $N = 7$ , quantidade de alimentos  $F = 6$ , raio de visão  $r_v = 0$ ,  $\alpha = 0$  e  $\delta = 1$ . Além disso, as constantes  $\delta'$  e  $\beta$  foram avaliados para a melhor combinação do experimento  $E_1$  discutido na Seção 7.2.1  $C_4 = \{\delta' = 0.25, \beta = 0.5\}$ . Note que, embora as variação  $C_4$  de  $E_1$ , seja a melhor encontrada, ela ainda apresenta um desempenho muito inferior ao experimento  $C_4$  de  $E_4$ , que foi utilizada nas seções anteriores. Essa configuração foi adotada, uma vez que configurações muito refinadas para o modelo não conseguem cobrir todo o ambiente de 600 células, uma vez que o time executa a tarefa rapidamente ( $T \approx 50$  passos). Para cada simulação, o time de robôs parte de uma disposição inicial aleatória. Além disso, foram avaliados diferentes tamanhos de fila  $Q = \{0, 5, 10, 15, 30, 50\}$  observando-se o desempenho dos robôs na

execução da tarefa de forrageamento. A Figura 7.16 apresenta os experimentos com diferentes tamanhos de fila, utilizando-se os parâmetros  $\delta = 1, \delta' = 0.25, \beta = 0.2, \alpha = 0, r = 0$ . Entendemos que quanto mais uniforme for a área explorada, mais eficaz é o conjunto de parâmetros utilizados, uma vez que, o feromônio foi utilizado para criar campos de força de repulsão entre os robôs do enxame e espera-se que eles façam a melhor cobertura do ambiente. Por outro lado, se todos os focos de comida forem encontrados rapidamente, pode ser que não se tenha uma boa cobertura do ambiente. Além disso, à medida em que aumentamos o comprimento da fila, o time de robôs pode encontrar áreas cegas formadas pelo próprio tabu da fila. Adicionalmente, quando temos valores mais refinados para as constantes do modelo, também podemos ter uma área com menor cobertura pelos robôs, uma vez que a quantidade de iterações menor produz uma cobertura menor do ambiente.

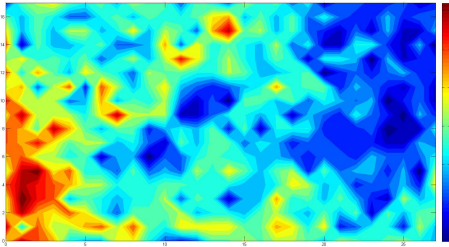
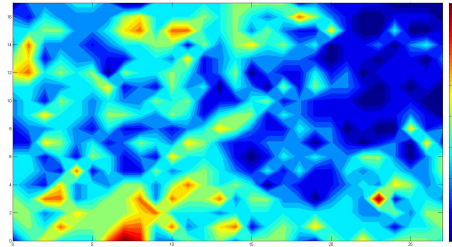
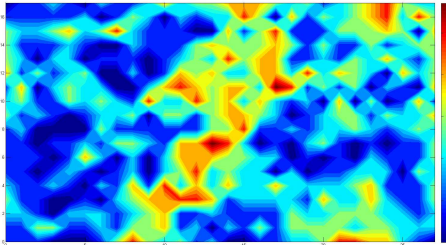
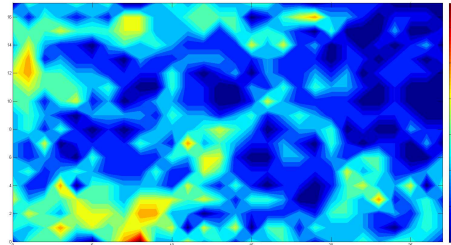
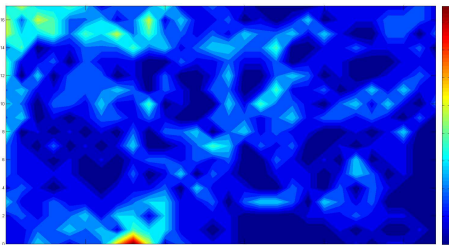
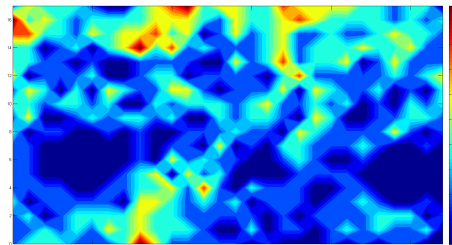
(a)  $Q = 0, \tau_{max}(x) = 13, T = 364$ (b)  $Q = 5, \tau_{max}(x) = 9, T = 146$ (c)  $Q = 10, \tau_{max}(x) = 6, T = 130$ (d)  $Q = 15, \tau_{max}(x) = 7, T = 118$ (e)  $Q = 30, \tau_{max}(x) = 7, T = 97$ (f)  $Q = 50, \tau_{max}(x) = 5, T = 94$ 

Figura 7.16: Gráficos dos experimentos com a variação  $C_4$ , constante de  $\delta' = 0.25$ , constante de declínio  $\beta = 0.2$  e  $\alpha = 5$  e raio de visão  $r = 0$ .

A Figura 7.16 apresenta os resultados do experimento variando-se o tamanho de fila. Neste experimento além de verificar que o aumento da fila  $Q$  provocou uma queda no número total de iterações, também observou-se que em  $Q = 0$  foi realizada uma grande co-

bertura do ambiente pelos robôs através do gráfico de temperatura. Nesse caso, o valor de pico observado foi de  $\tau_{max}(x) = 13$ . Apesar do experimento apresentar maior cobertura do ambiente (número maior de células visitadas), por ter memorização do caminho pelo robô ( $\delta' > \beta$ ), o número de iterações foi superior. Para as variações de fila  $Q = \{5, 10, 15, 30, 50\}$  temos valores de pico  $\{\tau_{max}(x) = 9, \tau_{max}(x) = 6, \tau_{max}(x) = 7, \tau_{max}(x) = 7, \tau_{max}(x) = 5\}$  apresentou resultados mais simétricos em relação aos valores de pico. Além disso, podemos observar que em todos os experimentos aqui apresentados a distribuição de cobertura dos robôs foi mais uniforme. Ou seja, à medida que aumentou-se o tamanho da fila foi possível observar que o número de iterações para completar a tarefa diminuiu significativamente, o que corrobora que o uso da fila torna-se necessário quando os parâmetros não estão ajustados.

### 7.2.5 Desempenho do time e distribuição dos robôs nos ninhos

Este experimento visa detalhar se o enxame está realizando as tarefas de forma cooperativa, sem sobrecarga de um único agente. Para os experimentos foram considerados  $r_v = 1$ ,  $\delta' = 1$ ,  $\delta' = 0.25$ ,  $\beta = 0.01$ ,  $\alpha = 1$ ,  $F = 30$  e  $Q = 10$ . Este experimento representa o estudo de caso de uma única execução para cada configuração definida pela quantidade de robôs  $N = \{3, 5, 7, 10, 15\}$  sendo que  $F = 30$  e  $Q = 10$ . A Figura 7.17 apresenta os resultados contrastando o tempo de iterações  $T$  e o número do robô que encontrou e depositou a comida no ninho e o tempo  $t$  em que esse objeto foi depositado no ninho. De acordo com o gráfico é possível perceber que os objetos foram distribuídos de maneira uniforme entre os robôs para cada experimento.

Este experimento representa o estudo de caso de uma única execução onde  $N = \{3, 5, 7, 10, 15\}$  com  $F = 30$  e  $Q = 10$ . De acordo com os resultados apresentados na Figura 7.18, percebe-se que o ninho da parte inferior (ninho 2) da Figura 7.1 foi mais utilizado em relação à outra saída (ninho 1). Acreditamos que esse comportamento deve-se ao fato de que não foi utilizada a estratégia de alteração da dinâmica de piso para fazer com que o robô trocar a sua posição durante o *homing* como foi proposto em (Alizadeh, 2011).

Uma compilação dos resultados está apresentada no gráfico da Figura 7.19 que o momento  $t$  que cada robô deposita o objeto coletado durante a busca no ninho, além disso, também é possível visualizar em qual ninho este robô depositou o objeto. Com isso, pode-se perceber que o enxame apresenta uma distribuição de tarefas uniforme entre os agentes, no entanto, ele não apresenta uma distribuição uniforme em relação aos ninhos depositados.

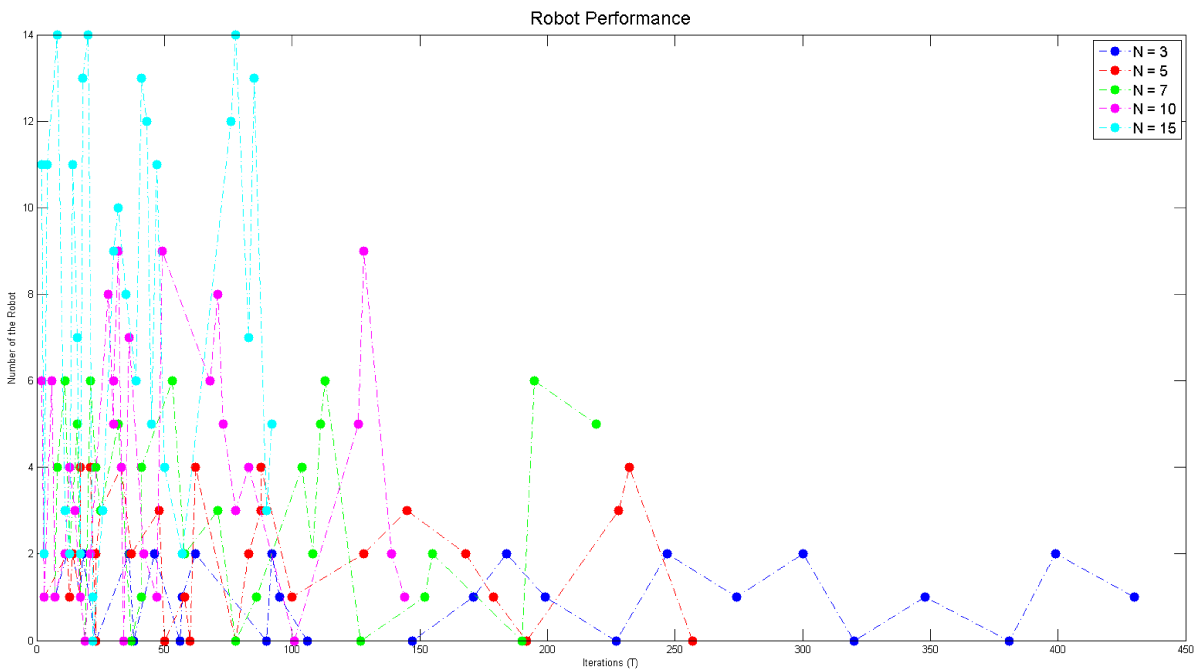


Figura 7.17: Resultados que mostram a relação entre o desempenho de cada robô em cada tempo.

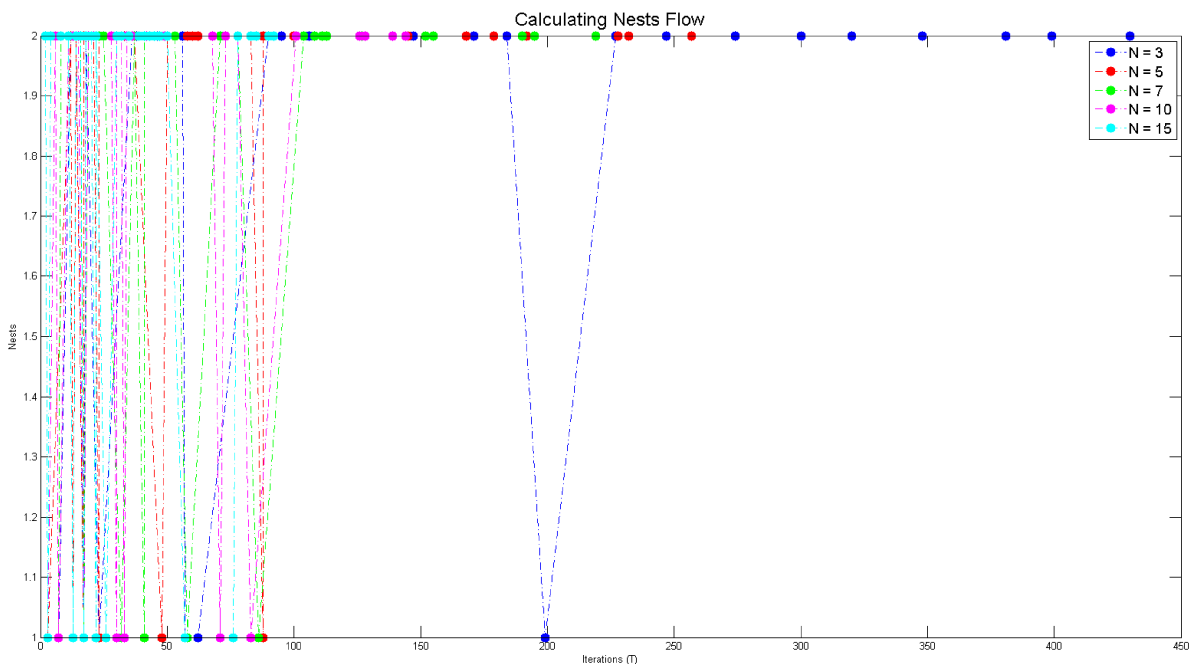


Figura 7.18: Relação entre o fluxo nos ninhos em cada tempo e o número de iterações.

### 7.2.6 Análise visual do efeito de arqueamento

Em ambientes com saída limitada (poucos ninhos) e uso de sistemas multi-robôs, o fenômeno de arqueamento é produzido próximo aos ninhos, porque todos os robôs tendem a mover-se em direção à saída, tornando-se um gargalo no depósito do alimento pelos robôs. Este é um fenômeno típico observado, caracterizando o fato de que quanto

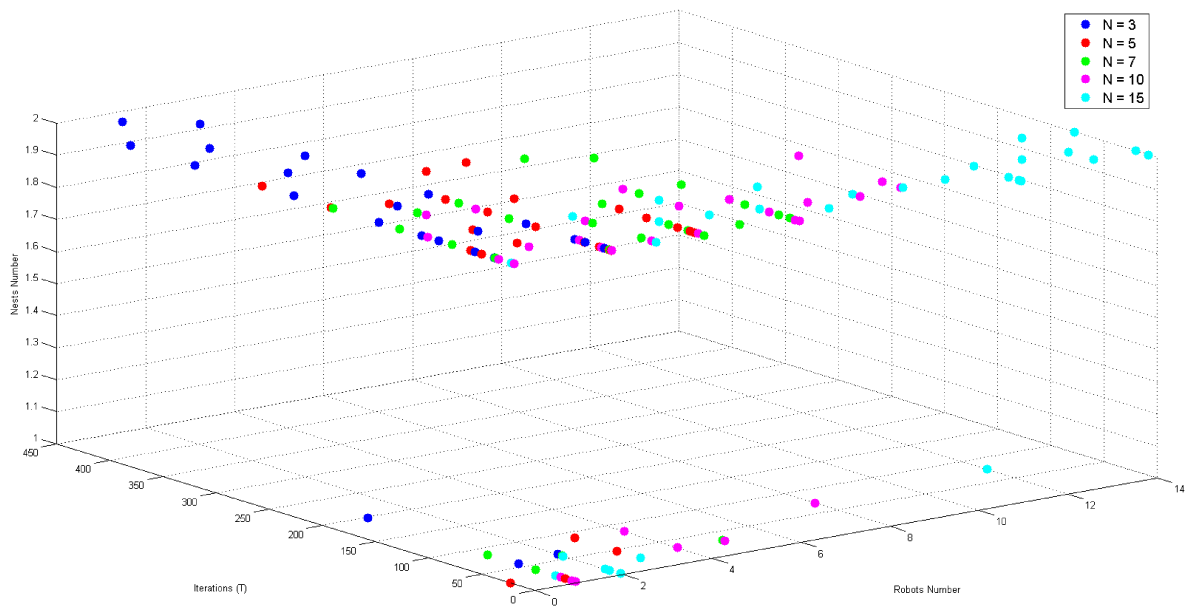


Figura 7.19: Relação entre o fluxo nos ninhos e o desempenho de cada robô e o número de iterações.

mais rápido os robôs desejarem se mover próximo ao ninho para depositar um alimento, mais lenta pode se tornar a execução da tarefa. Além disso, a forma com que os robôs interagem e a forma com que eles se distribuem próximo ao ninho, pode influenciar o tempo de execução da tarefa.

O primeiro algoritmo implementado, foi o modelo de primeira escolha de (Varas et al., 2007) que propôs a utilização do piso estático para direcionar o caminho do robô até o ninho, que consistiu em observar o comportamento de 50 robôs em um ambiente de dimensão  $20 \times 30$  células no processo *homing*. A Figura 7.20 representa alguns instantes de tempo do reticulado durante esse processo. Neste caso, foi observado que houve formação de filas frente aos ninhos e que o tempo necessário para a evacuação total é de  $T = 37$  passos de tempo. Essa formação de fila foi observada por (Alizadeh, 2011), que propôs o uso do piso dinâmico para evitá-la. O trabalho de (Alizadeh, 2011) inspirou o processo de *homing* implementado no modelo CAAM.

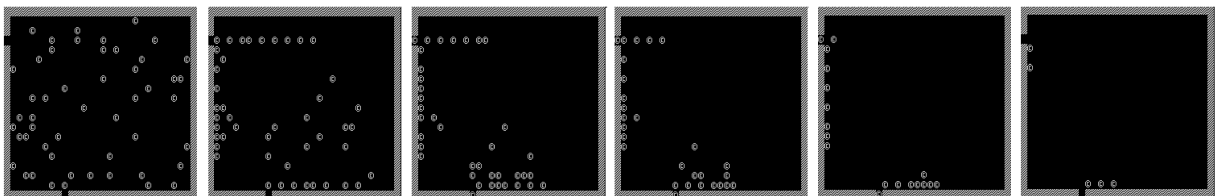


Figura 7.20: Modelo de Varas simulado em uma sala grande com dimensão de  $20 \times 30$  e  $N = 50$  robôs alocados e tempo total de evacuação foi de  $T = 37$  passos de tempo. Os tempos observados na figura são  $t = \{1, 5, 10, 15, 30\}$ .

Par fazer essa análise, fizemos duas implementações diferentes do processo de *homing*, a primeira é baseada apenas no piso estático  $\alpha =$  e a segunda utiliza também o piso

dinâmico  $\alpha = 5$ . Posteriormente, o modelo proposto neste trabalho para o processo de *homing* foi implementado e executados para exatamente o mesmo cenário utilizado para a Figura 7.20 com o mesmo número de robôs ( $N = 50$ ) a mesma distribuição inicial dos mesmos no ambiente. Verificou-se que o algoritmo de fato minimiza problema de formação de filas do algoritmo baseado em (Varas et al., 2007), justamente por adicionar um campo de piso dinâmico que faz com que os robôs tendam a evitar a formação de filas durante o processo de homing. Nos experimentos realizados, o valor de  $\alpha = 5$  foi adotado. A Figura 7.21 ilustra o experimento realizado. Neste caso,  $N = 50$  robôs tentam levar o alimento para um dos ninhos. Neste caso, é possível visualizar que o arqueamento do modelo apresentado ( $\alpha = 5$ ) é mais suave (Guo and Huang, 2008), enquanto que em ( $\alpha = 0$ ) existe formação de filas (Varas et al., 2007).

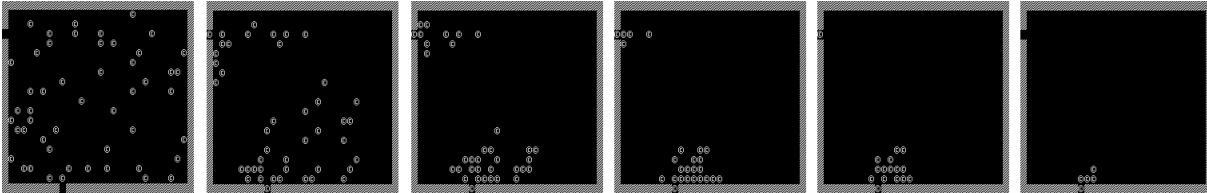


Figura 7.21: Modelo de homing simulado em uma sala grande com dimensão de  $20 \times 30$ ,  $\alpha = 5$  e 50 robôs alocados e tempo total de evacuação foi de  $T = 34$  passos de tempo. Os tempos observados na figura são  $t = \{1, 5, 10, 15, 30\}$ .

### 7.3 Tarefas de forrageamento: modelos PCAAM, D-CAAM e HCAAM

Alguns experimentos foram realizados para testar o desempenho da equipe de robôs usando o movimento PCAAM probabilista no estado *homing* em contraste com o processo de homing determinista utilizado no modelo CAAM. Os resultados dessa seção foram publicados em (Lima and Oliveira, 2016b).

O ambiente tem  $22 \times 32$  células e  $F = 30$  unidades de alimentos foram distribuídos sobre ele. obstáculos virtuais foram empregadas para evitar colisões. Os parâmetros do modelo utilizados foram  $Q = 10$ ,  $\alpha = 0$ ,  $r_v = 1$ ,  $r_p = 1$ ,  $\delta' = 0.25$  and  $\beta = 0.01$ . Cada experimento foi realizado utilizando-se  $10^3$  simulações com números diferentes de robôs  $N$ . Os parâmetros relacionados com o *homing* probabilístico foram: ( $c_{\min} = 1$  e  $c_{\max} = 100$ ). Adicionalmente, foram avaliadas diferentes quantidades de robôs:  $N = \{3, 6, 9, 12, 15\}$ . No primeiro experimento, o *homing* usa o modelo determinista, correspondente à CAAM sem o campo de piso dinâmico ( $\alpha = 0$ ), também chamado de CAA. O segundo experimento considera o *homing* usando o modelo probabilístico, correspondente ao modelo PCAAM. A Figura 7.22 mostra as diferenças entre os dois métodos, onde o eixo  $x$  representa o número de robôs  $N$  e o eixo  $y$  representa as iterações  $T$ . Usando um pequeno

número de robôs  $N = \{3, 6\}$ , embora o número de iterações em ambos os casos seja elevada, o modelo determinístico tem um desempenho melhor do que o probabilístico. Este fato ocorre porque não há conflitos perto dos ninhos com poucos robôs. Assim, a escolha do time através do campo de piso estático, que representa o menor caminho, supera a força da escolha probabilista. Por outro lado, quando o modelo é probabilístico, os robôs escolhem caminhos aleatórios e isto pode atrasar a chegada do robô no ninho. Quando o número de robôs é aumentado para  $N = \{9, 12, 15\}$ , o número de conflitos perto dos ninhos também aumentam. Usando um grande número de robôs, o modelo probabilístico ( $T = 157$ ) é superior em comparação ao modelo determinista ( $T = 175$ ), porque um robô no PCAAM é capaz de escolher caminhos alternativos. Os comportamentos de inercia Yang et al. (2005) e geração de filas em Varas et al. (2007) foram solucionados com PCAAM.

Em ambos os casos, o ninho 0 foi usado 6 vezes em contraste com as 24 vezes que o ni-

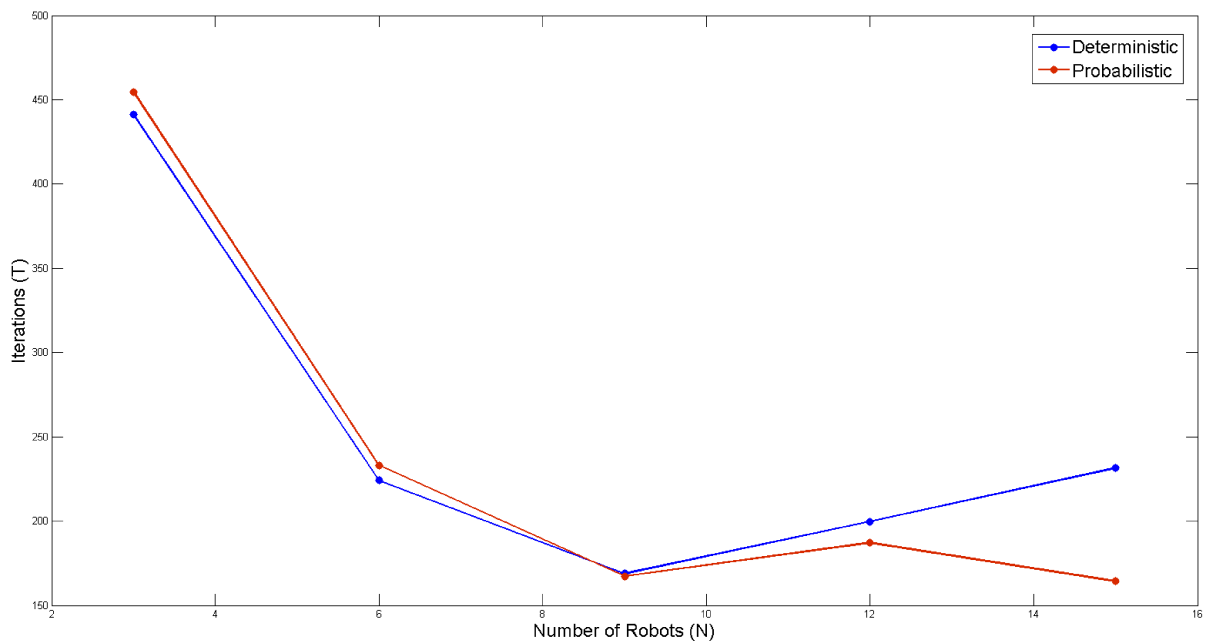


Figura 7.22: Número total de iterações (em média) usando  $N = \{3, 6, 9, 12, 15\}$ . (a) Linha vermelha: modelo determinístico. (B) Linha Azul: PCAAM.

nho 1 foi usado. Esta característica é uma consequência da distância do chão utilizado no processo homing, que é uma propriedade inerente deste ambiente com essas configurações. O tempo médio de chegada do time de robôs para a escolha determinista, considerando o ninho  $A_0$  é  $\mu_0 = 0.076$  e para  $A_1$  é  $\mu_1 = 0.13$ , e o rendimento do sistema é  $\lambda = 0.17$ . Considerando o modelo PCAAM, o rendimento do sistema é  $\lambda = 0.19$ , e o tempo médio de chegada é  $\mu_0 = 0.032$  e  $\mu_1 = 0.15$ , para os ninhos  $A_0$  e  $A_1$ , respectivamente. O desempenho ótimo do modelo usando 9 robôs refere-se a uma captura por robô de  $\approx 3$  objetos. No modelo PCAAM, cada robô capturou pelo menos 2 objetos, isso significa que cada um dos robôs está cooperando com o time. Por outro lado, no modelo determinista existe um robô captura apenas um objeto, isto significa que pelo menos um robô é severamente



subutilizado. Este fato pode ser explicado devido a alguns robôs ficarem presos em filas, devido a conflitos perto dos ninhos. Este comportamento foi reduzido com homing probabilístico.

A última análise do PCAAM foi realizada para visualizar o efeito arqueamento (Ya-

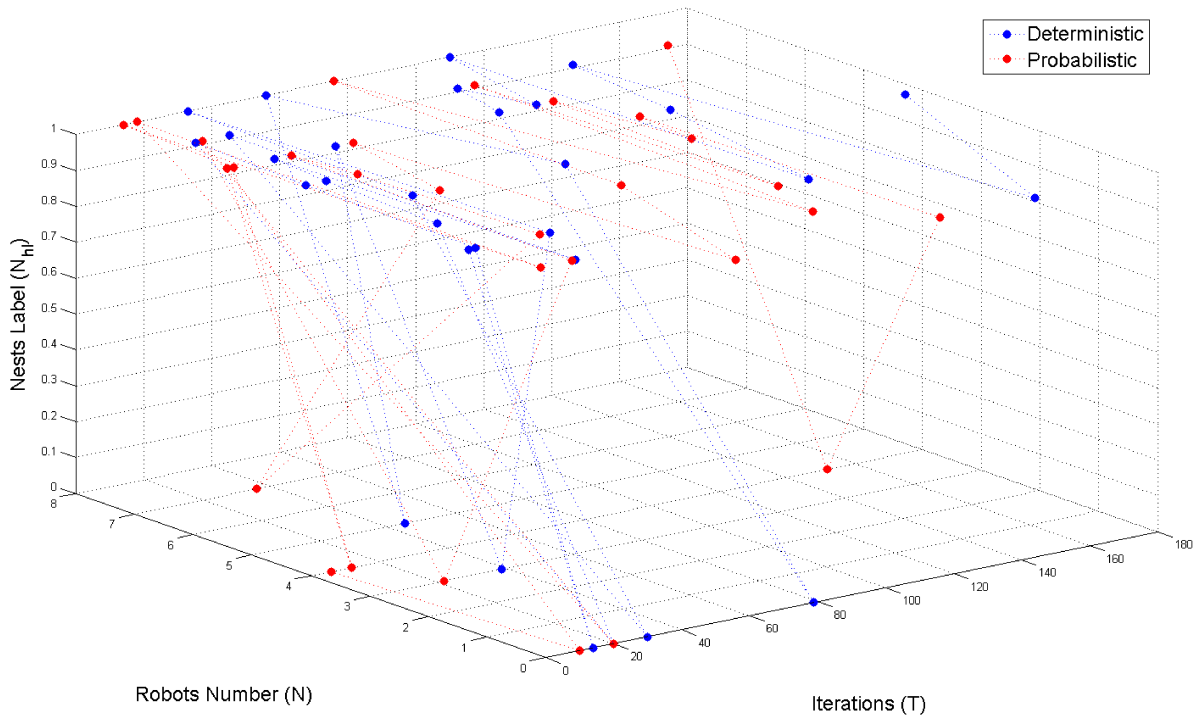


Figura 7.23: Iterações  $T$  comparação da equipe desempenho  $N = 9$  e uso ninho. (a) A linha azul representa o modelo determinista. (b) A linha vermelha representa o modelo PCAAM.

mamoto et al., 2007) perto dos ninhos e também a formação de linha Varas et al. (2007), Alizadeh (2011). Com o objetivo de realizar a análise, dois experimentos foram realizados utilizando  $N = 9$ ,  $N = 12$  e  $N = 15$  robôs para o PCAAM e CAAM. Usando o modelo determinístico (Figura 7.24 (a), (c) e (e)), com base apenas na distância proposta em (Varas et al., 2007), o efeito de arqueamento observado é um arco de fila, para todos os  $N$ . Este efeito de fila ocorre devido às características do piso estático e o efeito de inércia (Yang et al., 2005). Uma formação arqueamento mais suave é observada usando o modelo PCAAM (Figura 7.24 (b), (d) e (e)), devido aos robôs tentarem rotas alternativas em caso de conflito e evitando o comportamento inercial.

Em seguida, algumas simulações foram realizadas com o intuito de avaliar o comportamento das abordagens CAAQ, HCAAM e DCAAM. Estes experimentos tem por objetivo verificar as melhorias do modelo CAAM em função de novas políticas de gerenciamento da memória bem como melhorias no homing. Estes resultados foram publicados em (Lima and Oliveira, 2016a). Os experimentos foram realizados para testar o desempenho da equipe robô usando as diferentes abordagens de cada política de memória robô no estado *searching*. Além disso, no estado *homing*, o desempenho da equipe e a utilização



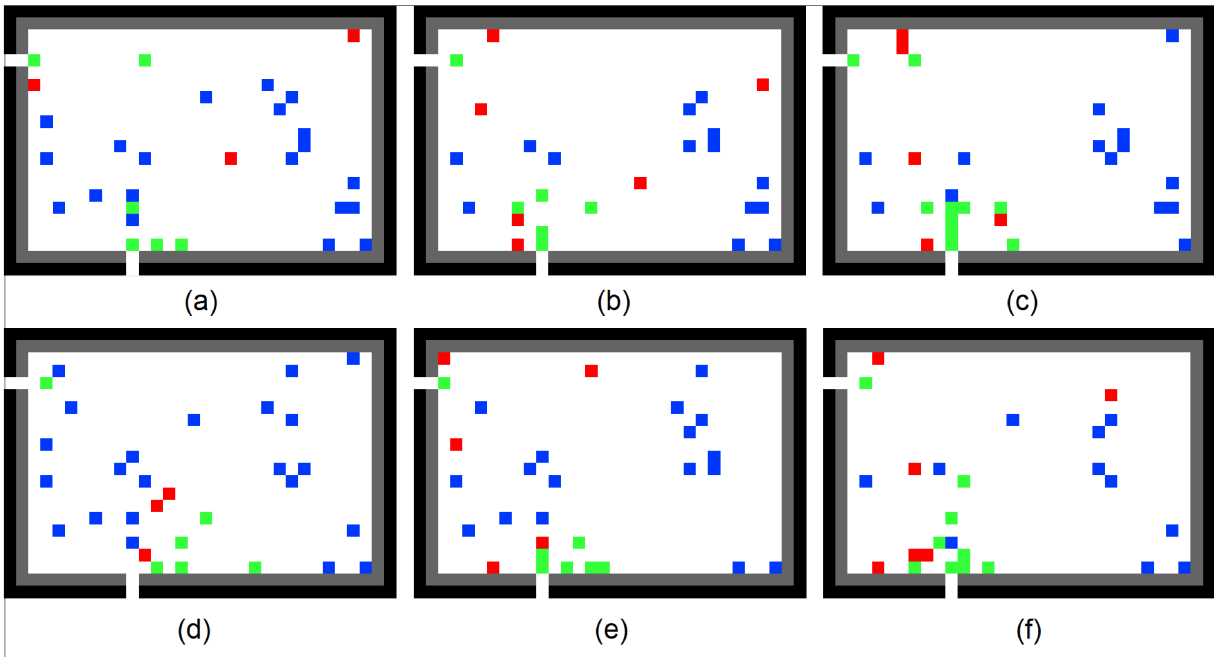


Figura 7.24: Arqueamento análise de formação visual e linha, que ■ representa robôs que buscam, ■ representa robôs no estado homing ■ representa unidades alimentares. (a) Determinístico  $N = 9$ . (b) PCAAM  $N = 9$ . (c) Determinístico  $N = 12$ . (d) PCAAM  $N = 12$ . (e) Determinístico  $N = 15$ . (f) PCAAM  $N = 15$ .

ninhos foi analisada contrastando o número de iterações  $T$ . O ambiente utilizado tem  $22 \times 32$  células, o mesmo utilizado em (Alizadeh, 2011) e CAAM, exceto pelo fato da adição dos obstáculos virtuais, e  $F = 30$  unidades de alimentos foram distribuídos sobre esse ambiente. Os parâmetros do modelo foram utilizados  $Q = 10$ ,  $\alpha = 0$ ,  $r_v = 1$ ,  $r_p = 1$ ,  $\delta' = 0.25$  e  $\beta = 0.01$ . Cada experimento foi realizado usando  $10^3$  simulações com números diferentes de robôs  $N$ .

Posteriormente, foram avaliadas diferentes números de robôs:  $N = \{3, 6, 9, 12, 15\}$ . Os parâmetros relacionados com a fila foram variados utilizando-se as combinações mostradas na Tabela 7.1. Neste trabalho, quatro combinações de fila foram implementadas. A primeira combinação (número 0) apresenta a mesma política de memória usada em CAAM e as outras combinações foram comparadas entre si.

Tabela 7.1: Diferentes combinações de memória durante o processo de pesquisa.

Número da técnica	Deadlock Case (Fila?)	Caso cheia (Fila?)
QP <sub>0</sub>	Não ( $M_0$ )	Não ( $L_0$ )
QP <sub>1</sub>	Não ( $M_0$ )	Sim ( $L_1$ )
QP <sub>2</sub>	Sim ( $M_1$ )	Não ( $L_0$ )
QP <sub>3</sub>	Sim ( $M_1$ )	Sim ( $L_1$ )

O gráfico da Figura 7.25 considera o processo de busca usando a variação política de memória, onde o eixo  $x$  representa o número de robôs  $N = \{3, 6, 9, 12, 15\}$  e o eixo  $y$

representa a média de iterações  $T$ . Todas as quatro combinações mostradas na Tabela 7.1 são destacadas no gráfico e cada ponto representa a média do tempo  $T$  das simulações  $10^3$ . O objetivo aqui é mostrar que a estratégia converge para um melhor desempenho. Uma média das  $5 \times 10^3$  simulações para cada  $T$  foi extraído conforme é mostrado na Figura 7.26. A melhor solução foi a combinação que esvazia a memória quando está cheia e remove apenas a primeira coluna e linha inserida na memória quando o robô está no estado de bloqueio.

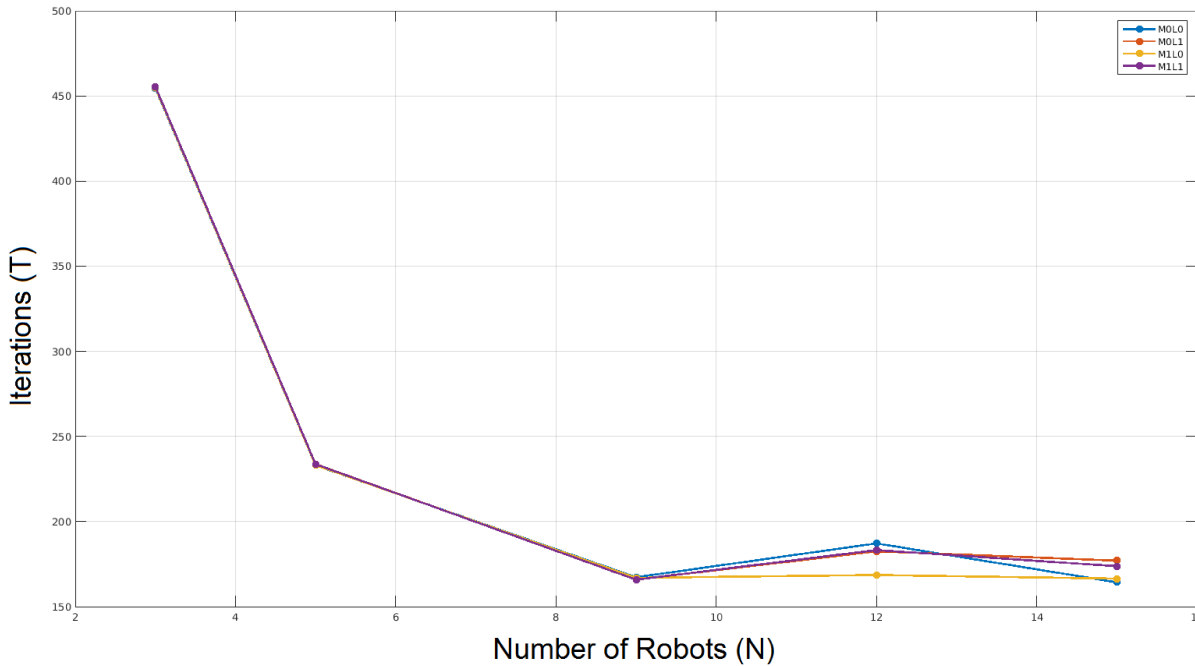


Figura 7.25: Número médio total de iterações usando  $N = \{3, 6, 9, 12, 15\}$  robôs com diferentes políticas de variação de memória (QP).

Posteriormente, um novo experimento foi conduzido para contrastar as abordagens que foram descritas neste documento com o descrito no modelo anterior CAAM com HCAAM e DCAAM. Os resultados foram publicados em (Lima and Oliveira, 2016a). Todas as simulações que testam as novas abordagens foram realizadas utilizando-se  $10^3$  simulações diferentes para cada  $N = \{3, 6, 9, 12, 15\}$ . A política de memória usada nesta considera a abordagem  $QP_2$ . A Figura 7.27 considera o HCAAM através da variação  $d_k^A = \{3, 7, 10, 14, 18\}$ , que representa a distância usada para a regra de transição quase aleatória para a abordagem de probabilidade elitista. Nesta tese,  $c_{max} = 1000$  e  $c_{min} = 1$  e os resultados são mostrados no gráfico ilustrado na Figura 7.27. Assim que a distância  $d_k^A$  é aumentada a média de iterações  $T$  é reduzida, aumentando o desempenho da equipe. Este fato ocorre devido ao fato de que o desempenho da equipe é aumentado, usando a distância mais curta. Quando a técnica HCAAM é inserida um movimento aleatório para fazer a equipe alterar a escolha do ninho durante o processo de homing. Esta técnica aleatória é responsável por causar um atraso em termos de desempenho da equipe, já que os robôs não escolhem as rotas ótimas proporcionadas pelo campo de piso estático

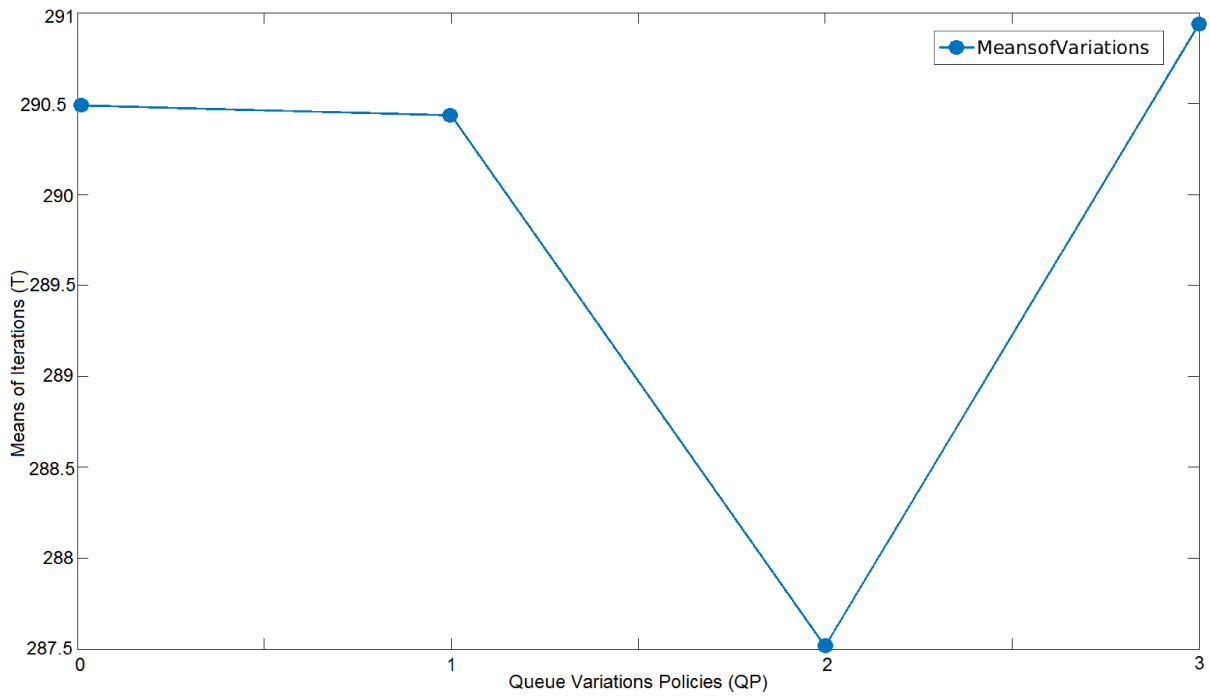


Figura 7.26: Cada ponto  $\{QP_0, QP_1, QP_2, QP_3\}$  representa a média de  $5 \times 10^3$  para cada abordagem descrita na Tabela 7.1.

proposto por (Varas et al., 2007).

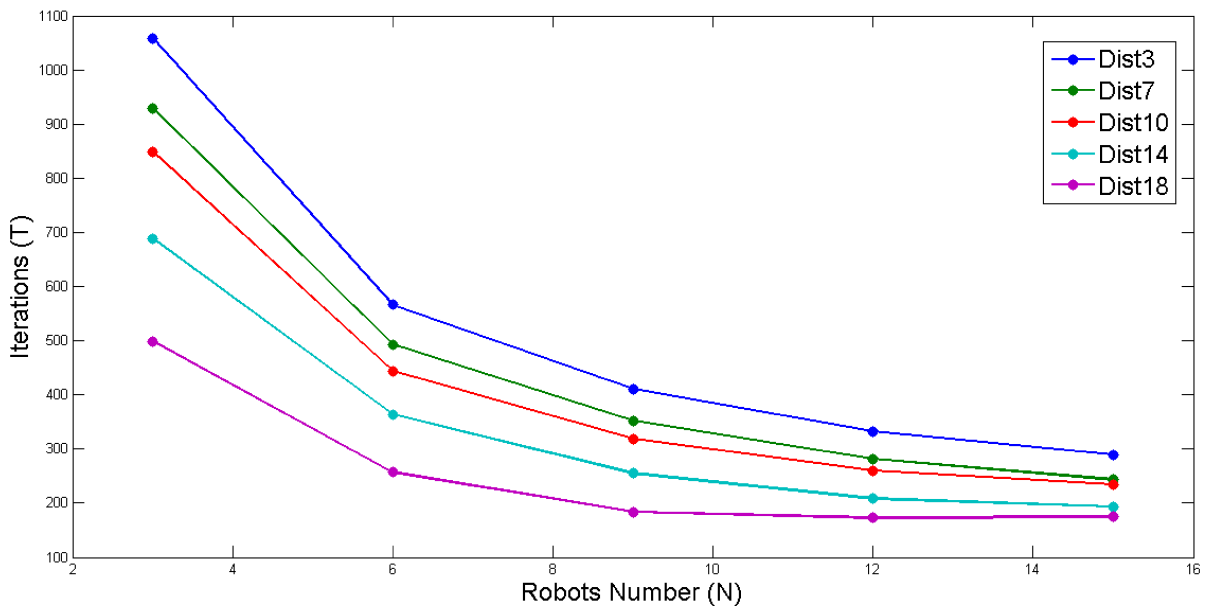


Figura 7.27: Número total de iterações (em média) usando Homing com HCAAM e  $N = \{3, 6, 9, 12, 15\}$  Robôs de um e variando  $d_k^A = \{3, 7, 10, 14, 18\}$  usando  $c_{max} = 1000$ .

As simulações mostradas nos gráficos da Figura 7.28 destinam-se a avaliar a abordagem homing DCAAM proposta neste trabalho. As simulações consideram uma variação de distâncias valendo  $d_k^A = \{3, 7, 10, 14, 18\}$  e para o movimento probabilístico elitista tem-se  $c_{max} = 1000$ ,  $c_{min} = 1$ . Em ambos os casos, quando a distância  $d_k^A$  é reduzida, o número

de iterações também diminui. A principal razão deste fenômeno é que o time tende a ir para o caminho mais próximo e apenas uma pequena parte da densidade perto do ninho realmente está sendo considerada.

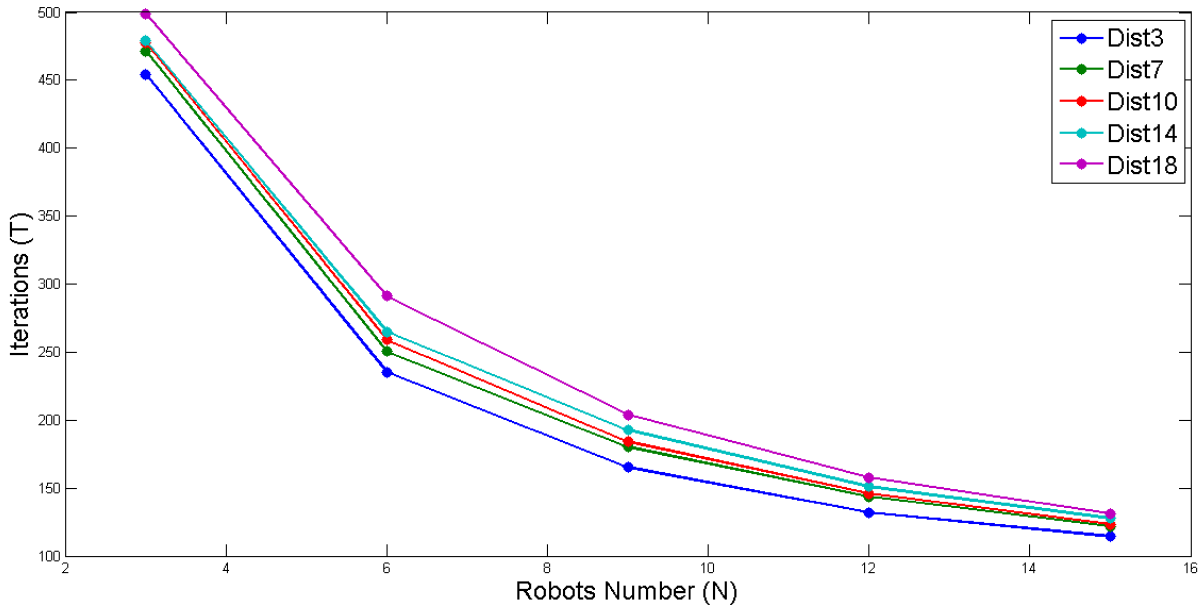


Figura 7.28: Número total de iterações (em média) usando Homing com DCAAM e com  $N = \{3, 6, 9, 12, 15\}$  robôs e variando  $d_k^A = \{3, 7, 10, 14, 18\}$  com  $c_{max} = 1000$ .

Para contrastar HCAAM e o DCAAM, com  $c_{max} = 100$ , um boxplot foi construído e apresentado na Figura 7.29. O experimento considera  $d_k^A = 10$  (linha vermelha) em ambas as abordagens homing dos dois modelos: HCAAM e DCAAM. A linha vermelha foi escolhida porque está mais próxima à média dos dois modelos entre  $d_k^A$  que foram avaliados. Cada boxplot final gerado representa a variação de  $N = \{3, 6, 9, 12, 15\}$  usando todos os  $5 \times 10^3$  valores. As combinações  $Hyb_1$  e  $Hyb_2$  representam o HCAAM usando  $c_{max} = 1000$  e  $c_{max} = 100$ , respectivamente. As combinações  $Vis_1$  e  $Vis_2$  representam o DCAAM usando  $c_{max} = 1000$  e  $c_{max} = 100$ , respectivamente. É importante ressaltar o fato de que a melhor abordagem em ambos os casos foi obtida utilizando o DCAAM, que tem o melhor desempenho quando comparado com os dois casos da HCAAM. A abordagem  $Vis_1$  de DCAAM usando  $c_{max} = 1000$  teve o melhor desempenho em contraste com todas as abordagens consideradas neste experimento. O método HCAAM embora tenha um tempo razoável, a equipe é degradada pela aleatoriedade da parte inicial do modelo, que uniformemente distribuído entre os robôs os ninhos. Esse fato acaba atrasando o depósito ninho, já que não considera o melhor caminho do campo de piso estático. Por outro lado, o homing no DCAAM cria dentro do sistema, uma boa distribuição dos agentes entre os dois ninhos, proporcionando, assim, através da utilização do cálculo da densidade, a melhor solução. A melhor maneira para cada robô chegar ao ninho é utilizando o campo chão estático, uma vez que os robôs são bem distribuídos dentro dos ninhos. Os robôs

utilizam  $c_{max} = 1000$  para usar o caminho mais curto, evitando a formação linha perto os ninhos.

Para comparar as novas abordagens em relação ao anterior, um gráfico está represen-

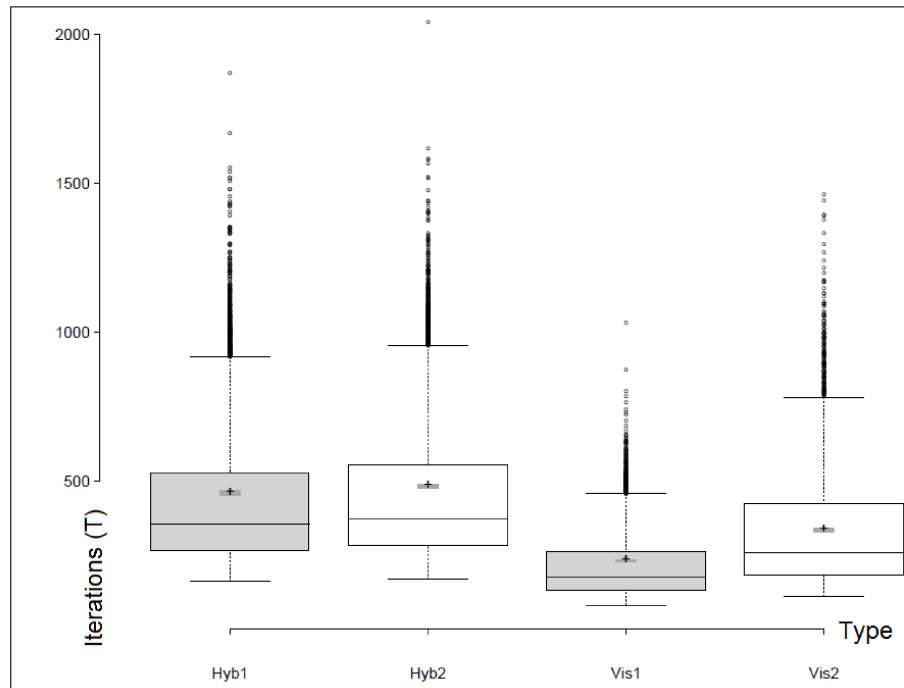


Figura 7.29: Boxplots para  $T$  passos em  $N = \{3, 6, 9, 12, 15\}$  para  $5 \times 10^3$  pontos: o HCAAM políticas homing e DCAAM, e  $c_{max} = \{100, 1000\}$ .

tado na Figura 7.30. A abordagem determinística foi implementada em Lima and Oliveira (2017a) e é possível ver que ele tem o melhor desempenho quando há poucos robôs. Neste caso, todos os robôs escolhem o caminho mais curto que é obtido pelo campo de piso estático proposto no Varas et al. (2007). Por outro lado, quando o número de robôs aumenta, todos os robôs tendem a ir para o mesmo ninho, devido à característica física das distâncias entre os ninhos. Essa característica faz com que exista formação de filas de robôs perto do ninho, aumentando o tempo para que o time complete a tarefa. A abordagem elitista foi implementada separadamente e publicada em PCAAM (Lima and Oliveira, 2016b), mostra que o modelo teve um bom desempenho em relação ao CAAM. A abordagem aleatória foi implementada usando a Equação 5.6 e apresentou um desempenho ruim em todas as circunstâncias, devido ao campo piso estático ter pouca influência para a convergência para o ninho. O homing utilizando HCAAM foi melhor que PCAAM, devido à possibilidade de a abordagem aleatória fornecendo ao sistema os meios para que o time também se distribua a outros ninhos e, no final, os robôs são geralmente atraídos para o ninho mais próximo. Por outro lado, DCAAM teve melhor desempenho sobre todas as situações em que o número de robôs foi aumentado, devido ao fato de que todos os robôs tendem a ir para diferentes ninhos para evitar interferência Yamamoto et al. (2007) e formação de filas de robôs perto dos ninhos Alizadeh (2011).

Com o objetivo de certificar que os robôs não tem apenas o objetivo de alcançar um

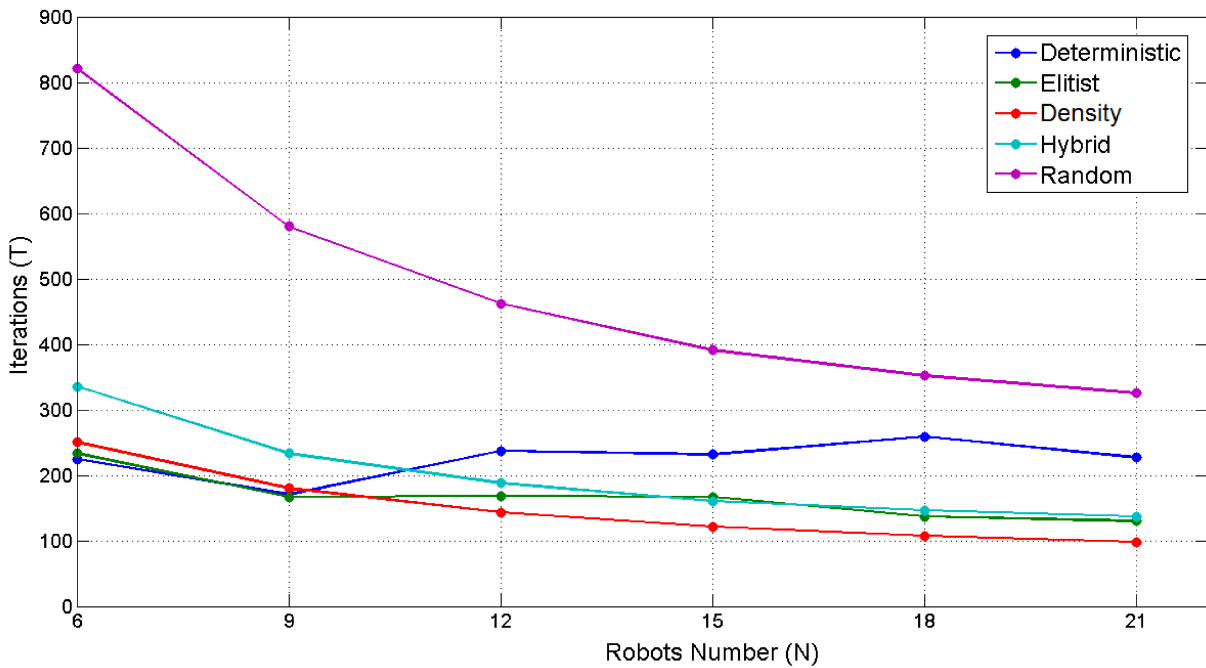


Figura 7.30: Gráfico que contrasta todas as abordagens implementadas neste trabalho em termos do desempenho da equipe variando o número de robôs  $N = \{6, 9, 12, 15, 18, 21\}$ .

melhor tempo de desempenho  $T$ , mas também em termos de utilização de recursos (ninhos  $A$ ), alguns experimentos foram conduzidos para validar a distribuição dos robôs em cada ninho. Todas os experimentos considerados utilizaram  $N = 15$  robôs. Em primeiro lugar, a Tabela 7.2 mostra o contraste dos cinco processos homing que foram discutidos neste trabalho. Os dados da tabela apresentam as distâncias  $d_k^A$  utilizadas nas abordagens HCAAM e DCAAM, a média de  $10^3$  iterações e a utilização de um único ninho. Para este experimento, utilizaram-se 15 pontos de dados  $d_k^A = \{3, 4, \dots, 18\}$  para a análise do comportamento de cada modelo. Para um bom desempenho do time, os casos em que os ninhos têm utilização iguais são mais adequados. Em termos de análise para o presente trabalho, uma boa distribuição ninho representa aquela onde cada ninho é utilizado, em média, 0.5 (50%) do tempo.

A Figura 7.31 mostra os pontos de Tabela 7.2 plotados em um gráfico, estas variações de distâncias para ambos os modelos propostos podem alterar o desempenho da equipe e distribuição ninho. Para mostrar o comportamento das distâncias no processo de direção, uma função de regressão  $y_1$  foi mostrado na Equação 7.4 para a abordagem dos HCAAM e a Equação 7.5 mostra a função  $y_2$  para DCAAM. Observa-se que a função  $y_2$  mostrado na DCAAM tem  $T$  menores que o HCAAM. Além disso, o uso ninho em DCAAM é mais uniformemente distribuído para robôs no processo de deposição de alimentos. Outro fenômeno importante que tem de ser destacado é o fato de que, quando  $d_k^A$  aumenta para DCAAM, a média de  $T$  diminui e a utilização dos ninhos de uma maneira uniforme também é diminuída. Este evento ocorre, à medida que mais movimentos aleatórios ocorrer ao longo de uma distância maior, maior é a probabilidade de alterações do ninho por um

Tabela 7.2: Distribuição dos ninhos de acordo com cada uma das abordagens investigadas.

Distancia entre $x_{ij}$ e ninho	Distribuição de ninhos	Passos de tempo	Distribuição de ninho	Passos de tempo
	Cálculo Híbrido		Cálculo de Densidade	
3	0.2752	300.531006	0.3946	114.399002
4	0.2727	283.441986	0.4416	115.810997
5	0.2726	274.571014	0.4513	117.727997
6	0.2680	266.239014	0.4730	121.412003
7	0.2671	259.157013	0.4851	121.747002
8	0.2669	248.686005	0.5010	122.904999
9	0.2622	241.960999	0.5110	123.350998
10	0.2592	233.899994	0.5243	123.171997
11	0.2574	225.639999	0.5388	123.514000
12	0.2556	219.126007	0.5386	125.773003
13	0.2537	209.207993	0.5447	127.079002
14	0.2537	198.871994	0.5492	127.538002
15	0.2533	193.934006	0.5461	128.222000
16	0.2449	177.057999	0.5546	127.963997
17	0.2442	168.826996	0.5635	128.791000
18	0.2424	160.425995	0.6031	131.242004
Resultados de outras abordagens investigadas	Cálculo Determinístico		Cálculo Randomico	
	0.2166	231.664993	0.2794	391.080994
	Cálculo Elitista		Média Geral	
	0.2335	151.923004	0.3692	184.994588

robô. Ao mesmo tempo, o campo estático andar Varas et al. (2007) perde a sua força e começa a aumentar o valor de  $T$  significativamente para concluir a tarefa.

$$y_1 = 0.0002 \times x + 0.2030 \quad (7.4)$$

Usando o homing DCAAM, uma distância ótima  $d_k^A$  também deve ser considerada. Como a distância  $d_k^A$  para o cálculo da densidade é aumentada, a força de campo piso é diminuída, uma vez que todas as estratégias são conduzidas para evitar que os ninhos sejam aglomerados. Com este experimento, podemos concluir que existe uma distância ideal que pode produzir o mais alto desempenho da equipe. Usando HCAAM é bom para equilibrar a distribuição ninho com cálculo não elitista. Por outro lado, a distância usada em DCAAM tem o efeito oposto, neste caso, o cálculo de distância correta pode medir a quantidade de robôs nos ninhos de modo a não afetar de forma significativa o desempenho da equipe.

$$y_2 = 0.0108 \times x - 0.8233 \quad (7.5)$$

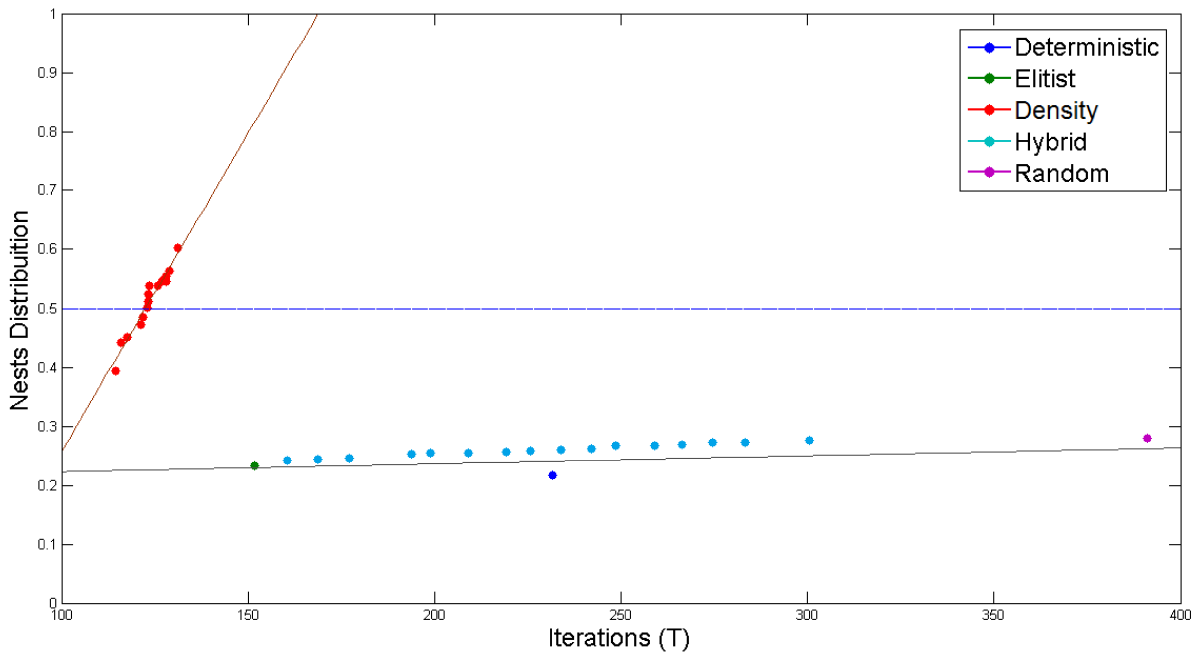


Figura 7.31: Variação de métodos implementados comparando a distribuição de ninhos sobre as iterações  $T$  usando  $N = 15$  robôs.

O experimento considerou que  $F = 30$  de unidades de alimentos têm de ser entregues aos ninhos ( $A_0$  e  $A_1$ ) no tempo diferentes passos de  $t$ . O modelo probabilista elitista, (Lima and Oliveira, 2016b) foi escolhido para avaliação devido ao seu melhor desempenho quando comparados com o modelo proposto Lima and Oliveira (2017a). A Figura 7.32 contrasta o modelo com homing elitista, que não apresenta uma boa utilização ninho  $A_0$ , com 7 casos de utilização do mesmo. Além disso, o gráfico compara os dois modelos

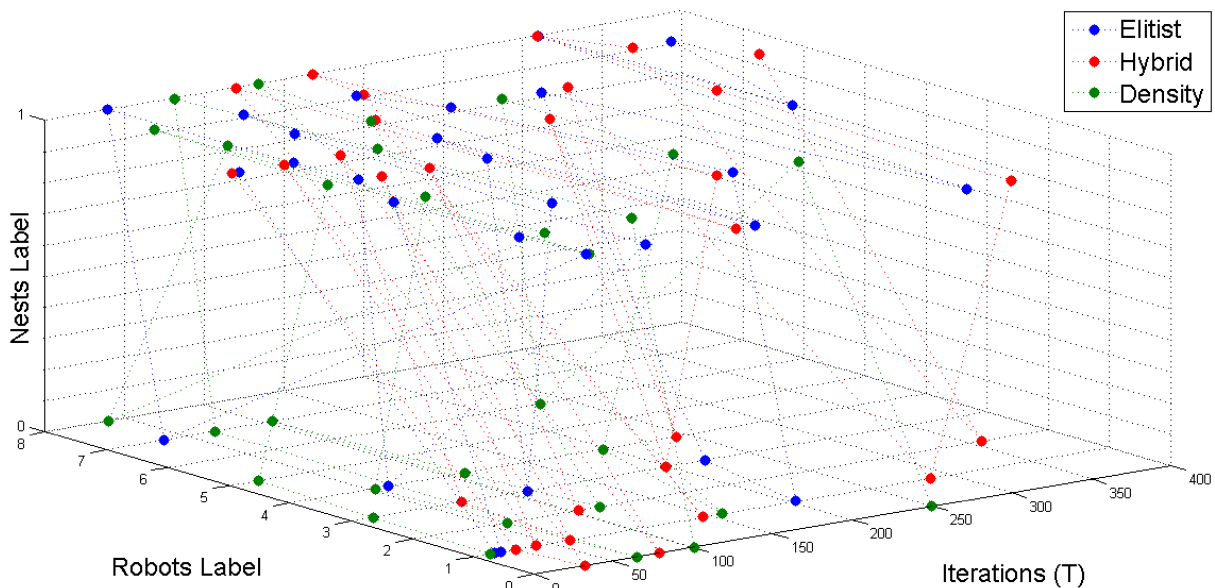


Figura 7.32: Iterações  $T$  comparação da equipe desempenho  $N = 9$  e uso ninho. (A) A linha azul representa a abordagem elitista. (B) A linha vermelha representa a abordagem HCAAM. (C) A linha verde representa a abordagem DCAAM.



propostos: HCAAM e DCAAM. Também foi possível mostrar que a HCAAM tem um bom desempenho da equipe aumentando a distribuição do ninho (12 casos de utilização do ninho  $A_0$ ). Por outro lado, o DCAAM tem a melhor distribuição da equipe nos ninhos (16 casos de utilização do ninho  $A_0$ ) e também o desempenho médio  $T$  da equipe é aumentado. Com este experimento, é possível reafirmar que o modelo homing de DCAAM alinhado à política de memória investigados em CAAQ, fornece um bom modelo para sistemas multi-agente.

## 7.4 Tarefa de coleta seletiva (SCAAM)

Todos os experimentos nos ambientes implementados na linguagem de programação C foram realizados a partir de  $10^2$  simulações, exceto nos casos que serão especificados separadamente. Todos os testes usam um campo de piso estático para uma sala  $20 \times 30$  células representadas na Figura 7.1. Assim como na maioria dos experimentos para a tarefa de forrageamento, existem dois ninhos; eles estão na parede à esquerda e na parede inferior e correspondem na figura às células com valor 1. Duas abordagens de implementação foram consideradas, uma primeira que considera um time de robôs homogêneo, e outro que considera um time de robôs heterogêneo.

### 7.4.1 Time homogêneo

Para determinar os parâmetros mais apropriados do modelo SCAAM, alguns experimentos iniciais foram realizados com o objetivo de ajustar as constantes envolvidas nos estados do modelo em especial: *searching* e *homing*.

Assim como no CAAM, neste modelo também é usado o feromônio invertido, que faz com que se evitem as colisões e repetições de maneira global a partir do depósito de feromônio realizado por todos os robôs. Os primeiros parâmetros ajustados são as constantes que determinam a difusão e o declínio do feromônio no processo de *searching*. Estes parâmetros são importantes para evitar a formação de filas e aglomerações. Além disso, as constantes  $\delta'$  e  $\beta$  sofrem alterações com o passar do tempo  $t$  a memória utilizada é constante e vale  $Q = 10$  (com a política que esvazia em caso de fila cheia ou deadlock, como no modelo CAAM) e  $F = 30$  objetos, sendo que  $F_1 = 15$  e  $F_2 = 15$  para cada tipo de objeto em todos os experimentos desta seção. O primeiro experimento leva em consideração um time homogêneo de robôs, onde qualquer robô pode encontrar e depositar qualquer tipo de objeto. A única diferença do modelo em relação ao CAAM é que este time homogêneo deve depositar cada tipo de objeto em um ninho do mesmo tipo do objeto capturado. Inicialmente, um modelo de primeira escolha (determinístico) foi utilizado tanto no homing quanto no *searching*.

Inicialmente, quatro possibilidades para as combinações  $\delta$ ,  $\delta'$  e  $\beta$  foram investigados.

A função utilizada para prover a difusão e declínio utilizada neste modelo é a função da tangente hiperbólica denotada por  $y = \left(\frac{1 - \tanh\left(\frac{t-a}{b}\right)}{c}\right)$ . A partir dos experimentos realizados em CAAM foi possível identificar que o valor do número de iteração  $T$  quando usa  $N = 3$  robôs resultou em 400 iterações. Essa média foi levada em consideração ao se configurar algumas variáveis da função. Assim, os valores de  $a$ ,  $b$  foram constantes  $\{200, 100\}$ , respectivamente, e  $c = \{2, 8, 200\}$  variou conforme os melhores valores de CAAM, que foram  $\delta = 1$ ,  $\delta' = 0.25$  e  $\beta = 0.01$ . Por exemplo, quando  $c = 8$ , temos que  $y$  vale no máximo 0.25 (e os demais valores da saída decrescem conforme a curva da função), que coincide com o valor fixo de  $\delta'$  em CAAM. Quando  $c = 2$ , temos que  $y$  vale no máximo 1 ( $\delta$ ) e quando  $c = 200$ , temos que  $y$  vale no máximo 0.01 ( $\beta$ ).

Assim, temos as quatro variações investigadas neste experimento conforme especificadas a seguir:

1. Os valores de  $\delta$ ,  $\delta'$  e  $\beta$  são fixos e valem respectivamente 1.0, 0.25 e 0.01.
2. Os valores de  $\delta$ ,  $\delta'$  valem respectivamente 1.0, 0.25 e  $\beta$  varia de acordo com a função  $\left(\frac{1 - \tanh\left(\frac{t-200}{100}\right)}{200}\right)$
3. O valor de  $\delta$  vale 1.0 e os valores de  $\delta'$  e  $\beta$  variam de acordo com a função  $\left(\frac{1 - \tanh\left(\frac{t-200}{100}\right)}{c}\right)$ , onde  $c = 8$  para  $\delta'$  e  $c = 200$  para  $\beta$ .
4. Os valores de  $\delta$ ,  $\delta'$  e  $\beta$  variam de acordo com a função  $\left(\frac{1 - \tanh\left(\frac{t-200}{100}\right)}{c}\right)$ , onde  $c = 2$  para  $\delta$ ,  $c = 8$  para  $\delta'$  e  $c = 200$  para  $\beta$ .

A Figura 7.33 apresenta os resultados em boxplots dessas quatro variações utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs para cada uma das quatro avaliações. No gráfico são primeiro apresentados os resultados dos experimentos com a variação  $v_1$  ( $\delta, \delta'$  e  $\beta$  constantes) referenciadas no eixo  $x$  por  $v_{1,12}$  refere-se à variação  $v_1$  com 12 robôs. O eixo  $y$  representa o número de iterações  $T$ . Neste gráfico é apresentado a média dos resultados representada pelo símbolo +, bem como o seu intervalo de confiança de 95% representado por ■ que sombreia o símbolo da média, quanto menor o comprimento deste sombreado, maior é a confiança da média, ou seja, menor é o desvio padrão. Os valores de *outliers* estão representados pelo símbolo o e a mediana está representada pelo traço do meio de cada caixa do gráfico. A partir deste conjunto de dados, conclui-se que a melhor variação é a  $V_3$ , pois para todos os valores de  $N$  temos que todas as médias se aproximam mais da mediana de cada um dos diagramas de caixa, mostrando uma maior tendência de distribuição simétrica dos dados. Adicionalmente, pode-se observar que a maioria das caixas que mostram o intervalo de confiança em  $V_3$  apresentam comprimentos curtos, mostrando maior confiança da média e conseqüentemente, um menor desvio padrão dos dados dessas simulações. Por fim, os dados em  $V_3$  apresentam o menor número de *outliers*, dentre os demais dados observados.

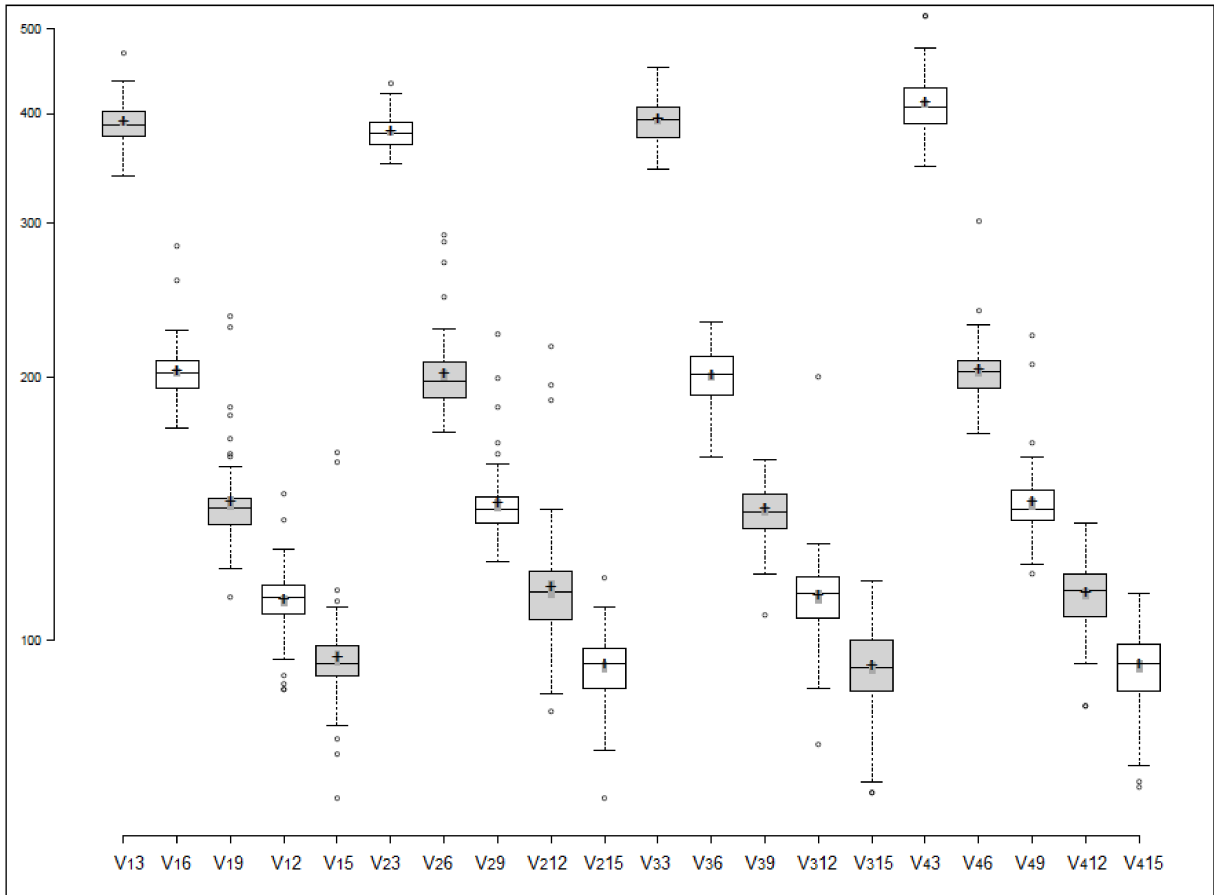


Figura 7.33: Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de coleta seletiva utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs com variações das constantes para  $\delta$ ,  $\delta'$  e  $\beta$ .

A principal razão para os resultados referentes ao número de iterações para a execução da coleta de objetos convergirem melhor em  $V_3$  deve-se ao fato de que é interessante que os valores de  $\delta'$  e  $\beta$  sofram pequenos decréscimos com o passar do tempo. Inicialmente,  $\beta$  pode ser evaporado mais rapidamente, já que no começo existem muitos focos de comida e os mesmos podem ser facilmente capturados até mesmo considerando-se uma busca com tendência mais aleatória. Com o passar do tempo, é importante que esse decréscimo de feromônio seja menos impulsor, já que com os poucos focos de comida restantes, é importante que o sistema de feromônio ajude mais na busca. Da mesma maneira  $\delta'$  deve sofrer decréscimos suaves, pois altos valores de  $\delta'$ , conforme apresentado em CAAM, prejudicam o desempenho do sistema.

Um segundo experimento foi realizado com o intuito de apresentar uma variação para o tipo de regra de movimentação nos principais estados do sistema: *homing* e *searching*. Foi utilizado a melhor configuração obtida no experimento anterior, que empregou apenas decisões determinísticas. Para a realização deste experimento, foram consideradas as seguintes variações:

D Nos estados *homing* e *searching* os movimentos são considerados ambos de primeira

escolha, assim como foi realizado o experimento anterior.

H No estado *homing* o movimento é estocástico e no estado *searching* o movimento considerado é de primeira escolha.

V Nos estados *homing* e *searching* os movimentos são ambos estocásticos V.

Na Figura 7.34 os resultados das variações investigadas na regra estocástica de movimentação dos estados *homing* e *searching* são apresentados. Nesses experimentos, o número médio de robôs também foi variado:  $N = \{3, 6, 9, 12, 15, 18, 21\}$ . O eixo  $x$  refere-se ao número de iterações  $T$  e o eixo  $y$  refere-se à combinação de experimentos. A partir

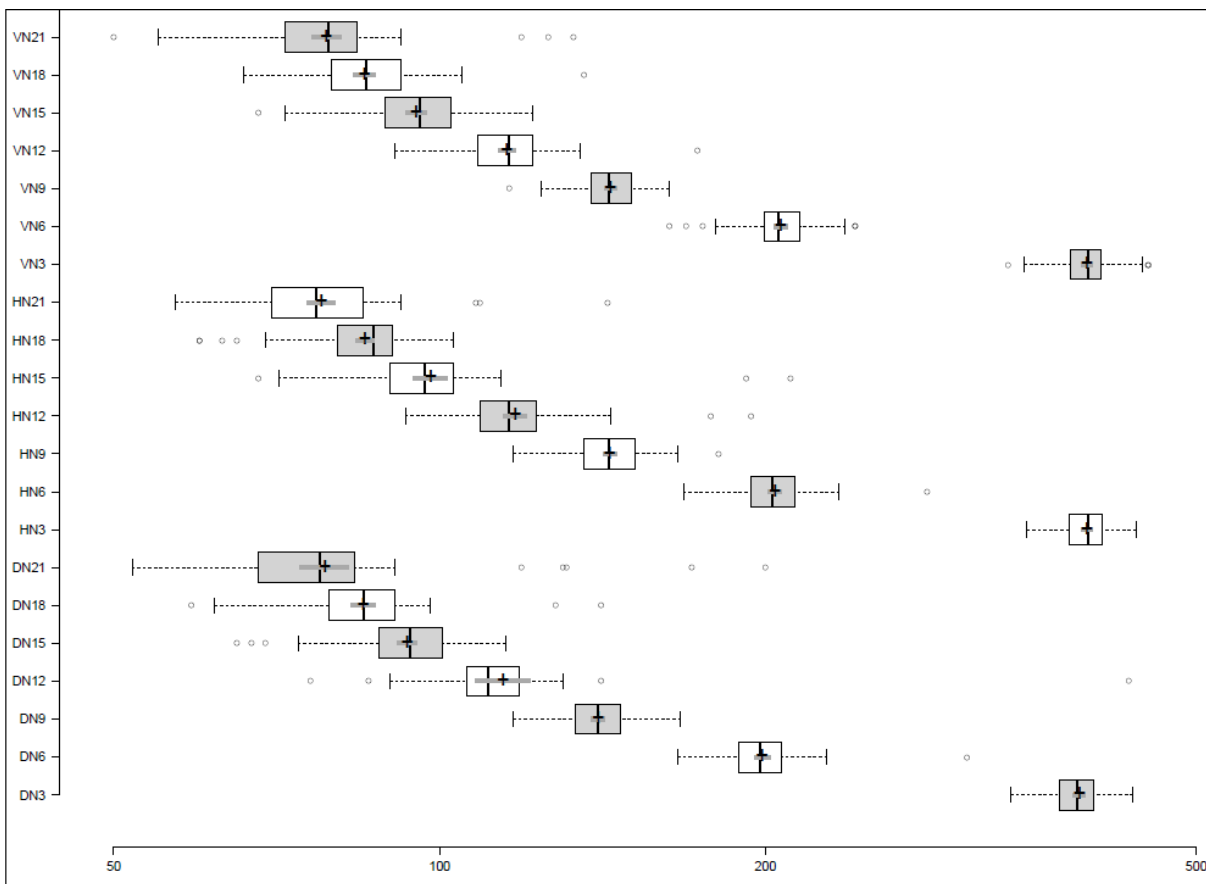


Figura 7.34: Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15, 18, 21\}$  robôs, variações nos *searching* e *homing*.

dos diagramas de caixa é possível observar que os resultados do modelo *homing* e *searching* apresentam os intervalos de confiança suas médias mais precisos. Adicionalmente, as médias e medianas estão mais próximas. Os *outliers* existentes para este conjunto de dados também encontram-se mais perto dos fios de bigode (*whiskers*) do diagrama de caixa, mostrando que esse conjunto de dados é o mais promissor dentre os demais apresentados, principalmente quando o número de robôs aumenta.

Esse resultado deve-se ao fato de que a combinação de um movimento mais estocástico elitista durante o processo de busca alinhado a um processo estocástico elitista em *homing*

produz resultados mais promissores, por tornar os efeitos de inércia ao longo do processo de busca menos frequentes. Dessa, forma produz-se resultados mais satisfatórios e com médias mais precisas.

### 7.4.2 Time heterogêneo

O próximo experimento refere-se à utilização de um time heterogêneo de  $N$  robôs dividido em dois tamanhos iguais a  $N/2$ , utilizando-se as melhores configurações do experimento anterior. Os robôs do primeiro time são especialistas em coletarem objetos do tipo 1 e os robôs do segundo time são especialistas em coletarem objetos do tipo 2. Um primeiro experimento considera um único campo de busca para os dois times de robôs. Em seguida, dois campos de busca são considerados. A Figura 7.35 apresenta os resultados do experimento com o time heterogêneo com 1 campo de busca e com o time heterogêneo e dois campos de busca, sendo que a primeira delas refere-se ao melhor resultado do experimento anterior, apenas para efeito de comparação. Pode-se observar que o time de

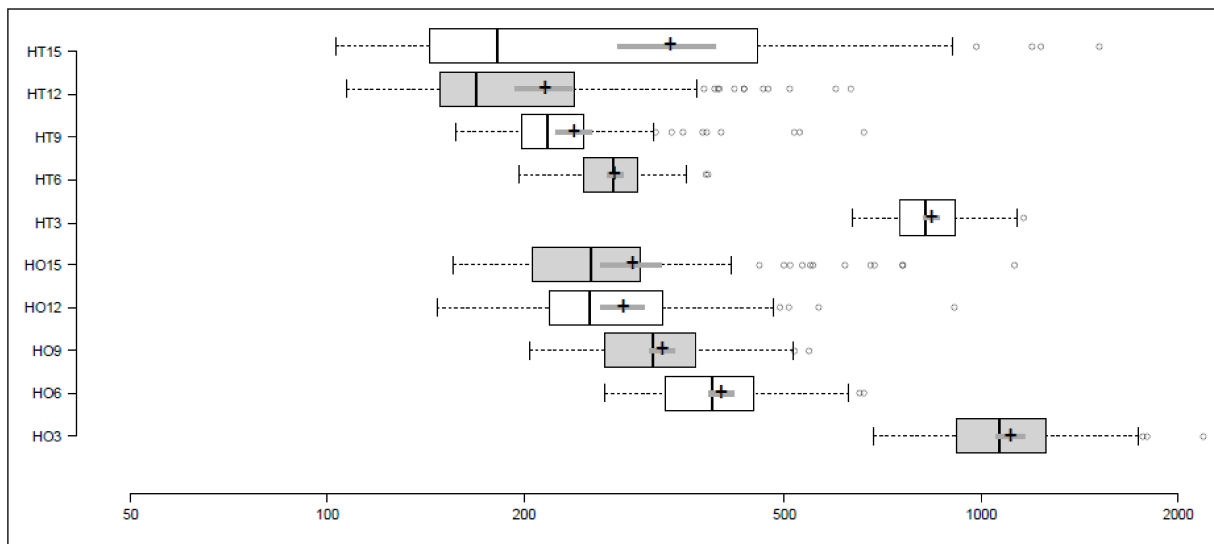


Figura 7.35: Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs, time heterogêneo com um campo de busca ( $HO$ ), time heterogêneo com dois campos de busca ( $HT$ ).

robôs heterogêneo que utiliza dois campos de busca, um para cada tipo de robô obteve melhores resultados do que àqueles utilizando-se apenas um campo de piso de busca. Esse resultado deve-se ao fato de que os robôs de tipos diferentes não podem compartilhar do mesmo campo de piso, uma vez que, eles possuem estratégias de busca diferentes. Exceto quando o time de robôs aumenta, então a ausência de ferômonio em alguns trechos provoca um maior número de colisões entre o time, já que os robôs compartilham de campos diferentes. Então pode ser que um time tente se guiar para as mesmas posições que do outro time e como não há composição de forças na intersecção dos dois campos, acontece um maior número de colisões. Por outro lado, os resultados obtidos neste experimento

são inferiores aos resultados obtidos no experimento onde foram considerados um time homogêneo de robôs. Esse fato acontece que quando os robôs tornam-se especializados demais, um time de robôs ociosos é gerado, caso os mesmos já tenham terminado de selecionarem e guardarem todos os objetos de um mesmo tipo. Sendo que eles não são capazes de ajudarem os demais robôs quando estão ociosos devido às suas especializações.

### 7.4.3 Cobertura do ambiente para os modelos homogêneo e heterogêneo

Inicialmente, foi medido o nível de cobertura do ambiente em termos do número de passos que o time de robôs exerceu sobre cada célula durante a tarefa de seleção e coleta dos objetos. Duas abordagens foram consideradas, a primeira delas mede o nível de cobertura do ambiente para um time homogêneo de robôs e a segunda mede o nível de cobertura do ambiente para um time heterogêneo de robôs. A Figura 7.36 apresenta os resultados dos experimentos com  $N = \{3, 6, 9, 12, 15\}$ .

Os experimentos do lado esquerdo da figura referem-se à uma cobertura do ambiente para um time homogêneo, enquanto que os experimentos do lado direito referem-se aos experimentos para um time heterogêneo. Pode-se observar que em todos os experimentos o algoritmo convergir para uma boa cobertura do ambiente. Em alguns casos do time heterogêneo, houveram casos de inércia provocado por diferentes campos de piso e por isso, em alguns casos, há regiões fora da área do ninho que apresentam uma coloração mais forte. Esse fato ocorre devido ao fato de que existem diferentes campos de piso que controlam os robôs e como não há nenhuma comunicação de intersecção entre eles, os mesmos podem aglomerar-se em áreas ao longo do reticulado.

A Figura 7.37 apresenta os resultados dos experimentos com  $N = \{3, 6, 9, 12, 15\}$  que fornece a cobertura do ambiente em termos dos mapas de feromônio capturados no último instante  $T$  da simulação. Os experimento do lado esquerdo refere-se a um time de robôs homogêneo que deposita os objetos de cada tipo em cada um dos respectivos ninhos.

Dessa forma é possível observar uma boa distribuição de depósito ao longo do tempo nos dois ninhos. Isso significa o time foi capaz de encontrar objetos de diferentes tipos ao longo de todo o reticulado durante toda a simulação. Assim, o ambiente encontra-se com uma distribuição de feromônio esparsa ao longo do ambiente (tons em azul mais claro) e tons em vermelho próximo aos dois ninhos.

Por outro lado, quando é utilizado um time heterogêneo onde as matrizes de feromônio não se comunicam, para cada um dos tipos de objetos há uma aglomeração de agentes mais próxima a cada um dos ninhos que representa um tipo de objeto. Por exemplo, o ninho superior  $A_1$  é o responsável por receber os objetos do tipo 1, assim a matriz de feromônio para robôs do tipo 1 tem áreas mais quentes (vermelho) próximo ao ninho  $A_1$  e menos quentes próximo aos ninhos  $A_2$  e vice versa.

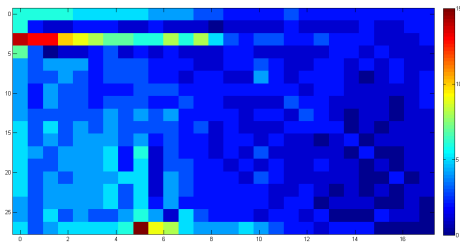
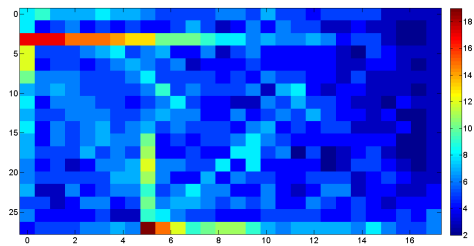
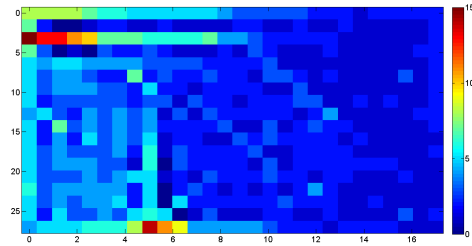
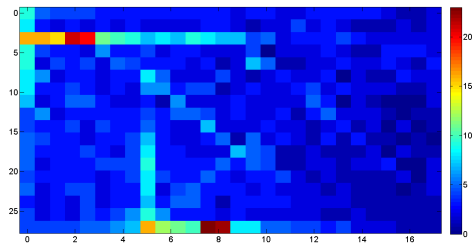
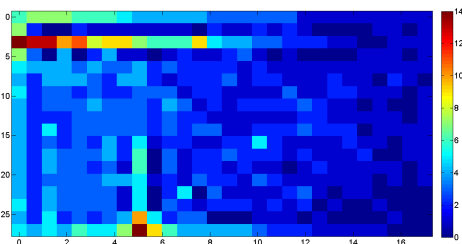
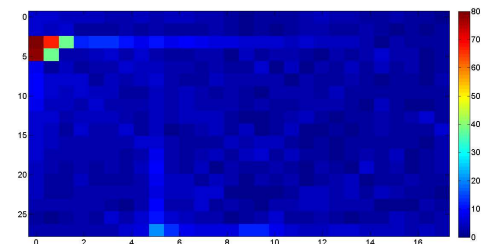
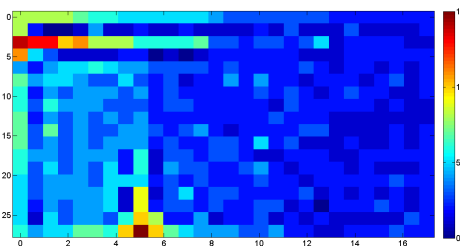
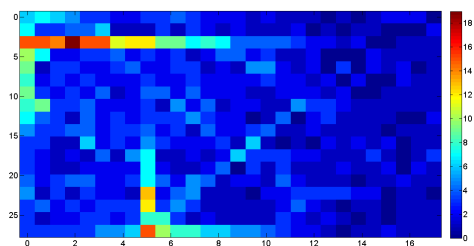
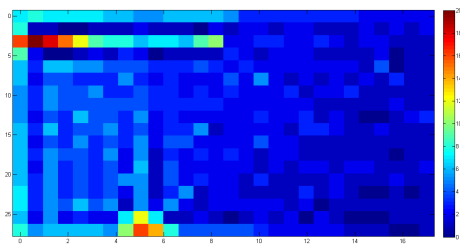
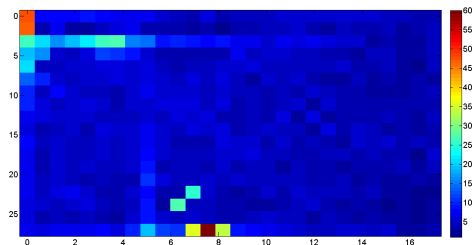
(a)  $N = 3, T = 408$ (b)  $N = 3, T = 887$ (c)  $N = 6, T = 201$ (d)  $N = 6, T = 267$ (e)  $N = 9, T = 233$ (f)  $N = 9, T = 212$ (g)  $N = 12, T = 122$ (h)  $N = 12, T = 107$ (i)  $N = 15, T = 94$ (j)  $N = 15, T = 174$ 

Figura 7.36: Mapas de temperatura que apresentam a cobertura do ambiente em relação ao número de passos do time de robôs utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs. A coluna da esquerda representa uma cobertura do time homogêneo, enquanto que a da esquerda representa um time de robôs heterogêneo.

Outro aspecto que pode ser observado é que a intersecção das duas matrizes mostra uma distribuição uniforme de depósito de objetos nos dois ninhos. Quando o número de robôs é par a distribuição tende a ser mais igualitária do que quando o número de robôs é ímpar, isso porque quando o número é par há uma mesma quantidade de robôs para procurar e depositar cada um dos tipos de objetos.

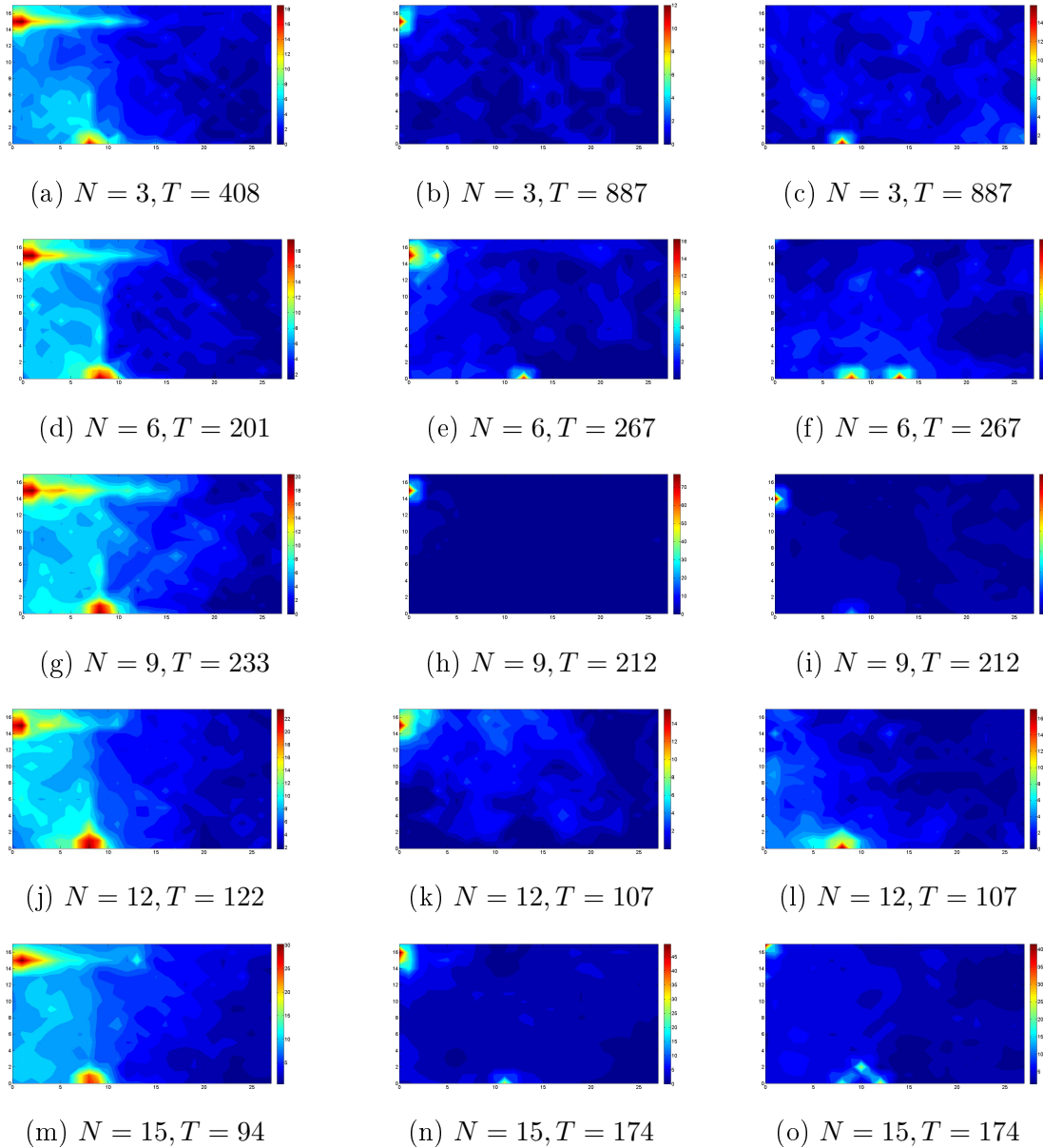


Figura 7.37: Mapas de temperatura que apresentam a cobertura do ambiente em relação à quantidade de feromônio do time de robôs utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs. A coluna da esquerda representa uma cobertura do time homogêneo, as colunas central e da direita representam a matriz de feromônio do para cada um dos tipos de robôs.



## 7.5 Tarefa de busca e resgate (FSCAAM)

Todos os experimentos com o modelo FSCAAM, descrito anteriormente, foram realizados para cada experimento nos ambientes implementados na linguagem de programação C++ foram realizados a partir de  $10^2$  simulações, exceto nos casos em serão especificados separadamente. Todos os testes usam um campo de piso estático para uma sala  $20 \times 30$  células representadas na Figura 7.1. Assim como na maioria dos experimentos para a tarefa de forrageamento, existem dois ninhos; eles estão na parede à esquerda e na parede inferior e correspondem na figura às células com valor 1. Além disso, testes de análise das condições do ambiente também foram levados em consideração.

### 7.5.1 Testes de análise estrutural

Inicialmente, foi feito um experimento para a análise do tempo  $T$  para a conclusão da tarefa de busca e resgate variando-se o número de robôs  $N = \{3, 6, 9, 12, 15\}$  robôs. Nesse experimento, foi empregada a fila  $Q = 10$  com a mesma política de memória adotada em CAAM e uma abordagem de primeira escolha para todos os casos de simulação,  $F = 30$  unidades de comida,  $\delta = 1$ ,  $\delta' = 0.25$ ,  $\beta = 0.01$ ,  $\alpha = 0$  e  $r_v = 1$  que foram os mesmos utilizados nos experimentos do modelo CAAM sem uso do piso dinâmico e que retornaram os valores mais altos de  $T$  sendo adequados para medir as condições mais inóspitas de trabalho. Este experimento foi realizado com o intuito de medir a distribuição da propagação do fogo utilizando-se  $b = 10$  e  $c = 2$  para as cinco variações da tangente hiperbólica  $y = \left(\frac{1 - \tanh\left(\frac{t-a}{b}\right)}{c}\right)$  em relação ao parâmetro  $a = 5$  para  $V_1$ ,  $a = 10$  para  $V_2$ ,  $a = 25$  para  $V_3$ ,  $a = 50$  para  $V_4$  e  $a = 100$  para  $V_5$  foram realizadas. Quanto maior for o valor de  $a$ , maior é o tempo de distribuição que a curva da tangente hiperbólica pode fazer com que o fogo permaneça, pois há valores distantes de zero. Quanto menores são os valores de  $a$ , menor é o prazo de distribuição da probabilidade que pode ocasionar fogo, por causa da amplitude da curva. Se os resultados desta primeira avaliação podem ser observados na Figura 7.38. Os resultados indicam que uma melhor distribuição de médias ao longo das simulações indicam que o fogo apresenta uma combustão mais rápida. Por outro lado, que uma combustão mais lenta, gera uma rede de obstáculos móveis para os robôs o que gera um pior desempenho do time de robôs.

Já os experimentos apresentados na Figura 7.39 foram realizados de forma similar as da Figura 7.38 apenas alterando o valor da constante da função que controla a propagação de fogo  $b = 20$ . Assim como no experimento anterior, os resultados indicam que uma melhor distribuição de médias ao longo das simulações indicam que o fogo apresenta uma combustão mais rápida, enquanto que uma combustão mais lenta, gera uma rede de obstáculos móveis para os robôs o que gera um pior desempenho do time de robôs. Quanto maiores os valores de  $b$ , maior é o tempo de distribuição que a curva da tangente hiperbólica pode fazer com que o fogo se propague ao longo do reticulado. Atrasando

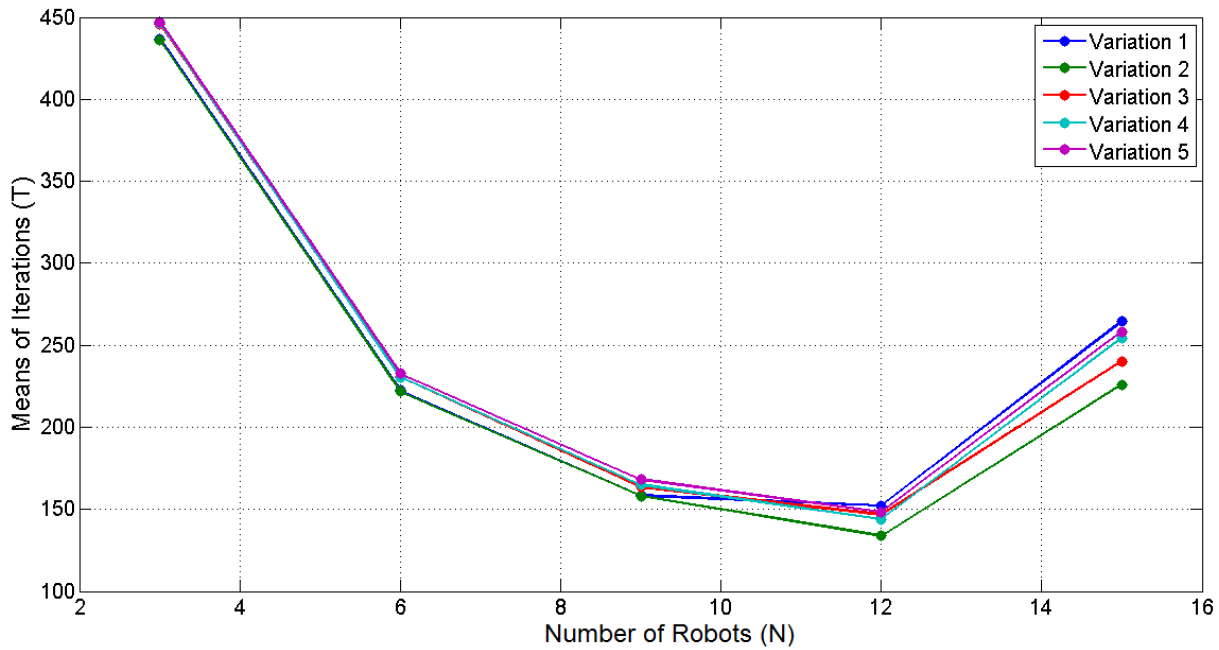


Figura 7.38: Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs, com  $b = 10$  e  $c = 2$ , para as variações da tangente hiperbólica com  $V_1 = 5$ ,  $V_2 = 10$ ,  $V_3 = 25$ ,  $V_4 = 50$  e  $V_5 = 100$ .

assim, o início de propagação do fogo e fazendo com que os robôs encontrem menos obstáculos móveis ao longo do reticulado.

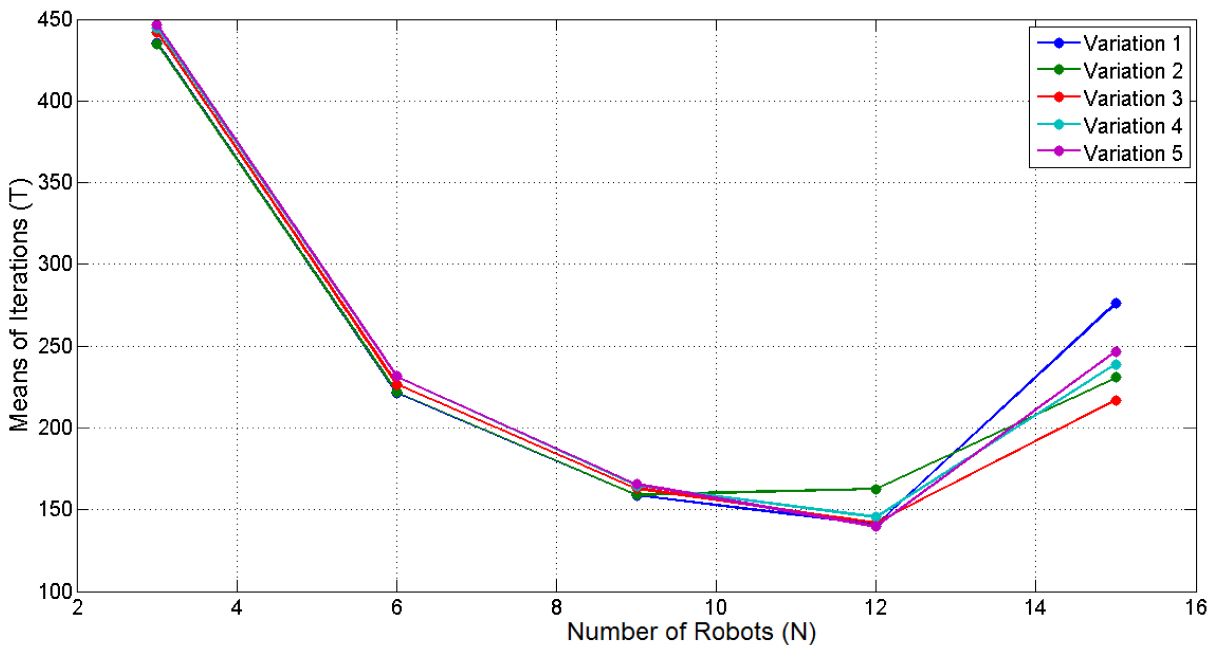


Figura 7.39: Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs, com  $b = 20$  e  $c = 2$ , para as variações da tangente hiperbólica com  $V_1 = 5$ ,  $V_2 = 10$ ,  $V_3 = 25$ ,  $V_4 = 50$  e  $V_5 = 100$ .

Tendo em vista que as variações dos experimentos  $a, b, c$  apresentam modificações no tempo de convergência do algoritmo, um novo experimento com uma nova função de probabilidade para propagação do fogo foi realizada com o intuito de avaliar se diferentes

funções podem alterar mais significativamente o comportamento do time de robôs. A Figura 7.40 apresenta o gráfico da média de  $T$  para que o time realize a tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs, com as seguintes funções para propagação do fogo: tangente hiperbólica  $a = 25$ ,  $b = 10$  e  $c = 2$  e a secante hiperbólica  $y = a \times \sec(\frac{t}{b})$  com  $a = 2$ ,  $b = 10$ .

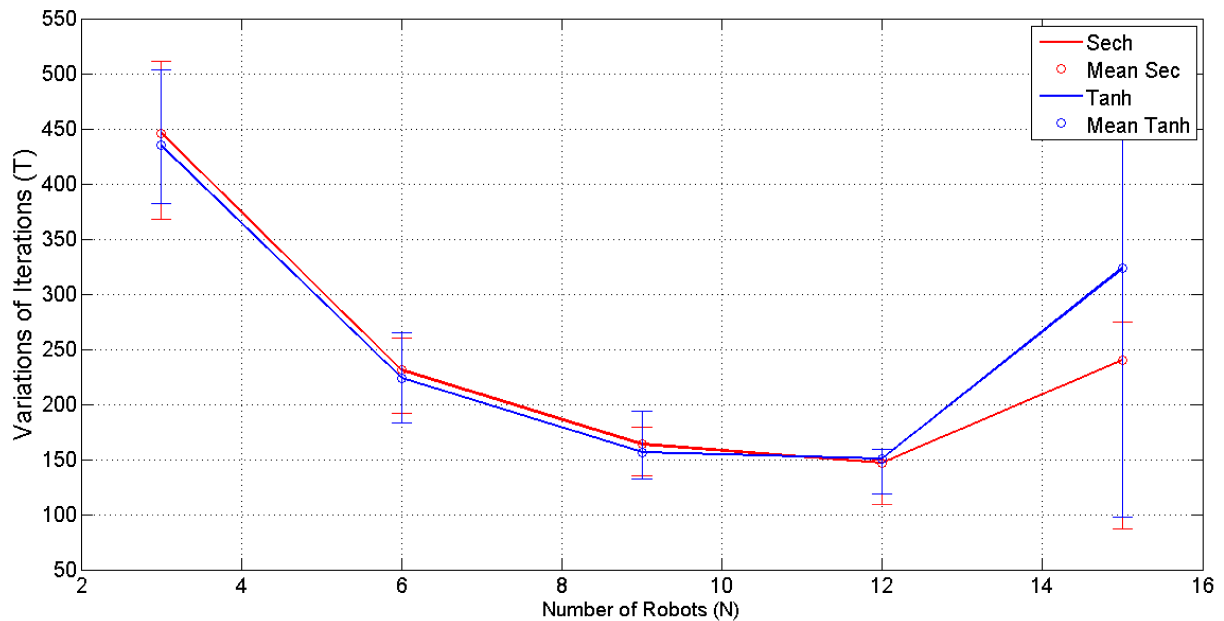


Figura 7.40: Gráfico de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs, com as seguintes funções para propagação do fogo: tangente hiperbólica  $a = 25$ ,  $b = 10$  e  $c = 2$  e a secant hiperbólica com  $a = 1$ ,  $b = 10$  e  $c = 2$ .

Posteriormente, um outro experimento baseado em (Zheng et al., 2012) foi projetado a fim de avaliar o quanto esse posicionamento inicial do foco de incêndio pode afetar diretamente a simulação da forrageamento. Neste experimento foram considerados 5 posições para o incêndio: noroeste (NW), nordeste (NE), central (CE), sudeste (SE) e sudoeste (SW). A tangente hiperbólica  $a = 25$ ,  $b = 10$  e  $c = 2$  foi considerada para a simulação. Os experimentos utilizando-se  $N = \{3, 6, 9, 12, 15\}$  foram organizados através de bean-plots, onde a maioria dos dados estão concentrados próximos às medianas, excetuando-se para  $N = 15$  em todos os experimentos, onde a concentração dos dados está distante da mediana (em vermelho). Os melhores resultados encontrados foram com focos nas posições iniciais: nordeste e central, porque estão mais distantes dos ninhos, portanto, não atrapalham a entrada dos robôs aos ninhos criando efeito de inércia (Yuan and Tan, 2007) próximo aos ninhos e formação de filas (Alizadeh, 2011). Por outro lado, os resultados com inicialização à noroeste e sudoeste apresentam os piores resultados, já que esses apresentam as inicializações próximas aos dois ninhos, dificultando a entrada dos robôs.

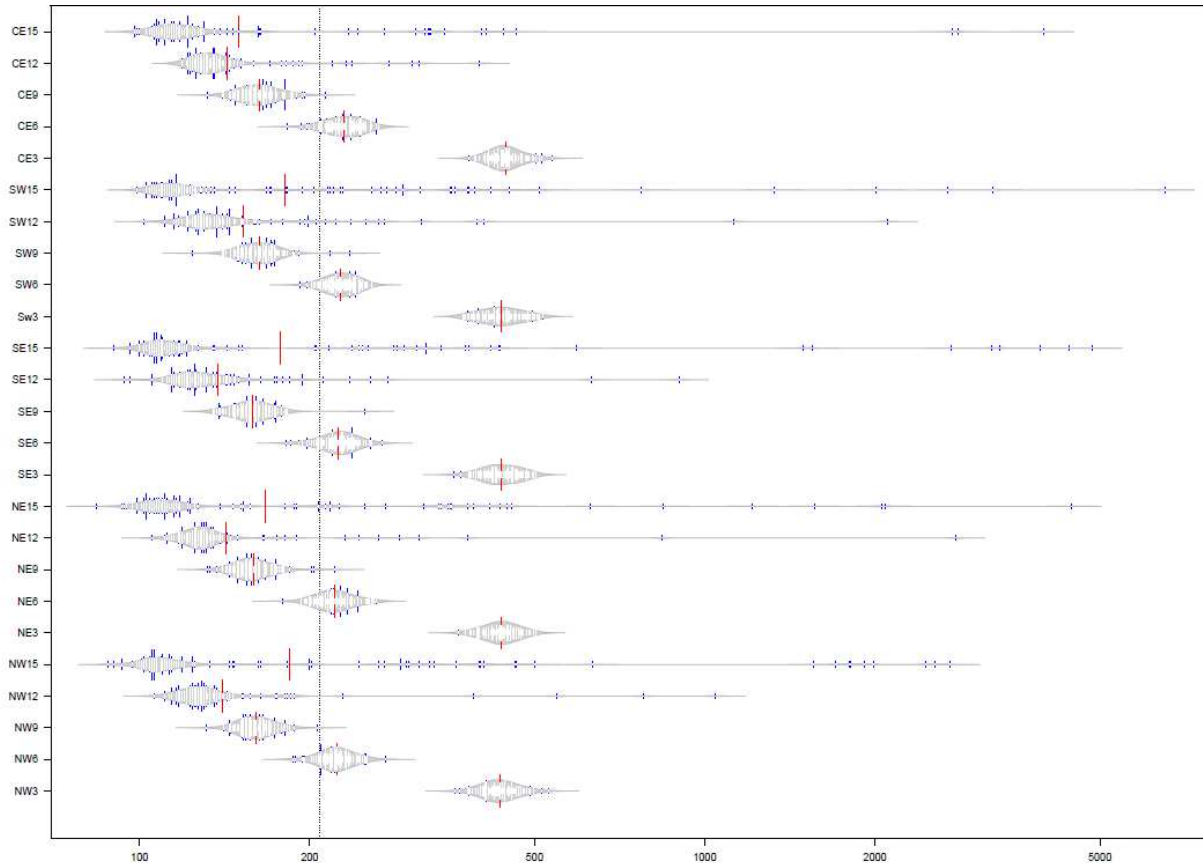


Figura 7.41: Bean plot de análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs para as posições de início de foco de incêndio nas posições: noroeste (NW), nordeste (NE), sudeste (SE), sudoeste (SW) e central (CE).

### 7.5.2 Testes de configuração do modelo

Testes de refinamento foram realizados para ajustar os parâmetros do modelo. Num primeiro experimento, realizamos testes para refinar as constantes da função de declínio  $\beta$ . Os resultados foram realizados com  $N = \{3, 6, 9, 12, 15\}$  robôs, e os estados *homing* e *searching* são modelos de primeira escolha,  $\alpha = 0$ ,  $\delta = 1$ ,  $\delta' = 0.25$  com memória  $Q = 10$ . Inicialmente, foi utilizada uma função secante hiperbólica para modelar o declínio do feromônio  $\beta = \frac{a}{\cosh(x)}$  foi definida para  $\beta$ , variando-se  $a_i = \{0.01, 0.05, 0.1, 0.2\}$ . Os resultados da Figura 7.42 indicam que  $\beta = 0.2$  apresentou os melhores resultados (menores valores médios e próximos às medianas com intervalos para a média mais confiáveis), isso se deve ao fato de que valores mais altos de declínio devem compensar o decaimento final ao longo do tempo, demonstrando que a existência de um valor constante que define a função  $\beta$  é importante para a definição do sistema. Como esse valor é decrementado ao longo do tempo, um valor suficientemente grande de  $\beta$  ( $< \delta'$ ) é interessante para que ele possa sofrer esse declínio com o passar do tempo sem que o mesmo possa sofrer uma perda de valor significativo.

Posteriormente, um experimento para avaliar a política de memória de fila foi realizado

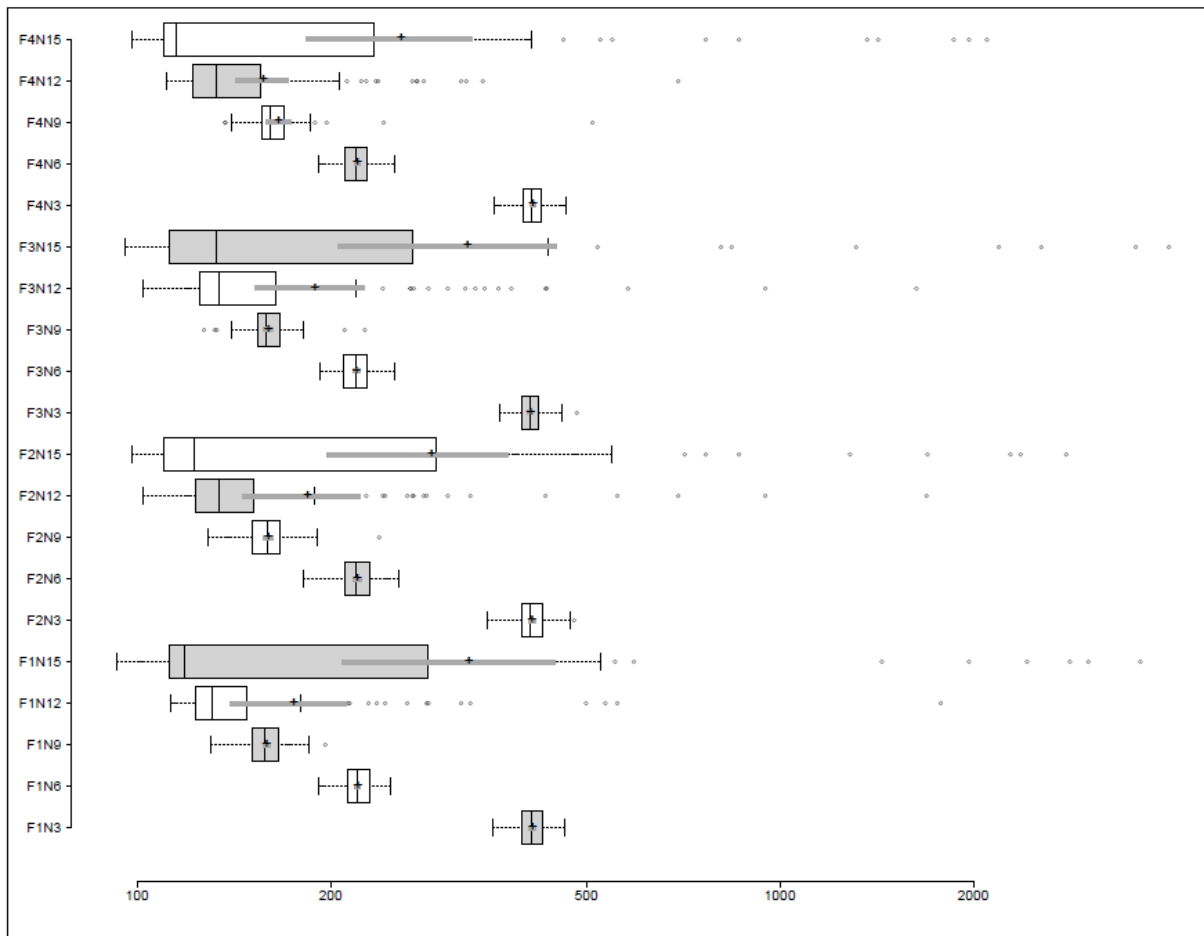


Figura 7.42: Análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs para as seguintes variações de cosseno hiperbólico:  $\alpha = 0.01, 0.05, 0.1, 0.2$ .

em  $N = \{3, 6, 9, 12, 15\}$  para verificar se os valores apresentados para os refinamentos de CAAM, também podem ser avaliados para o FSCAAM. Esses resultados são importantes, pois na modelagem para o FSCAAM é apresentado um nível de complexidade do ambiente em termos de obstáculos muito maior que em CAAM e nos algoritmos precursores, já que o fogo é uma dificuldade a mais. Portanto, mais deadlocks durante a evacuação podem ocorrer. Assim, a Figura 7.43 apresenta os resultados para a variação da política de fila de tamanho  $Q = 10$  conforme os seguintes casos de avaliação.

- $Q_0$  A memória é completamente esvaziada no caso de deadlock e é completamente esvaziada no caso de memória cheia, e os novos elementos são inseridos à medida que o robô realiza seu movimento.
- $Q_1$  A memória é completamente esvaziada no caso de deadlock e remove-se apenas o primeiro elemento que entrou no caso de memória cheia, assim a nova posição é inserida ao final da fila.
- $Q_2$  A memória sofre alteração de apenas do primeiro elemento no caso de deadlock, assim a nova posição é inserida ao final da fila, e é completamente esvaziada no caso

de memória cheia.

$Q_3$  A memória é alterada utilizando a política de fila para os casos de deadlock e também no caso de memória cheia.

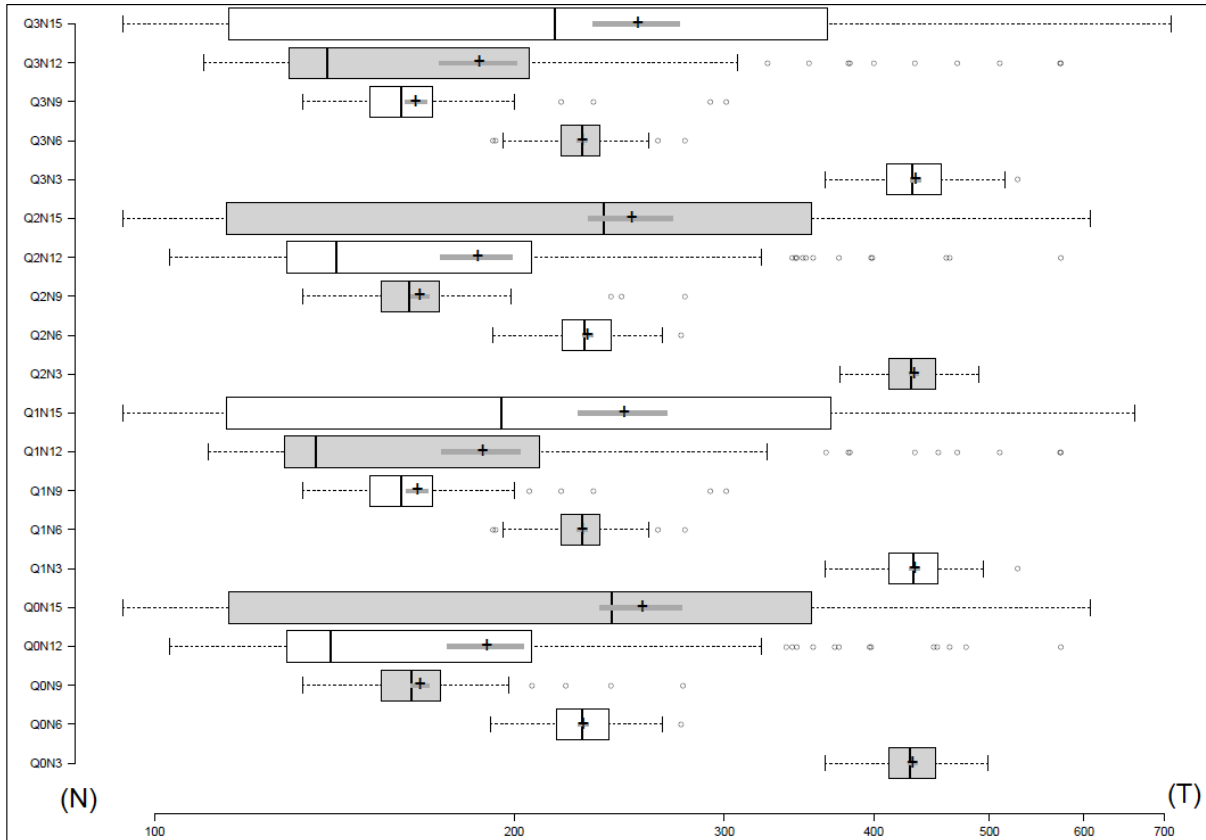


Figura 7.43: Análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs para as seguintes variações de políticas de fila:  $Q_0 = M_0L_0$ ,  $Q_1 = M_0L_1$ ,  $Q_2 = M_1L_0$  e  $Q_3 = M_1L_1$ ,  $M_i$  representa esvaziar completamente, se  $i = 0$  ou remover, se  $i = 1$ .  $L_i$  representa esvaziar completamente, se  $i = 0$  ou remover, se  $i = 1$ .

Os resultados da Figura 7.43 que apresentam os melhores resultados (menores valores médios e próximos às medianas com intervalos para a média mais confiáveis) são as combinações  $Q_2 = M_1L_0$  e  $Q_0 = M_0L_0$ , pois a memória deve ser esvaziada quando a mesma está completamente cheia, pois permite uma trajetória livre de conflitos entre os robôs e os obstáculos móveis de fogo exibidos mais ao início da simulação.

Posteriormente, quatro possibilidades para as combinações  $\delta$ ,  $\delta'$  e  $\beta$  foram investigadas para verificar o quanto essas combinações podem afetar o desempenho do time de robôs. As melhores configurações dos experimentos anteriores foram consideradas. Sabe-se que a função utilizada para prover a difusão e declínio utilizada neste trabalho é a função da cosseno hiperbólica denotada por  $y = \left(\frac{a - \cosh\left(\frac{t}{b}\right)}{c}\right)$ , no entanto, outras funções podem ser avaliadas. A partir dos experimentos realizados em CAAM foi possível notar que o valor máximo de variação de  $T$  dos experimentos ocorre quando  $N = 3$  com uma média

de 400 iterações, sendo que essa média foi levada em consideração ao configurar algumas variáveis da função. Assim, os valores de  $b = 200$ ,  $c = 200$  e  $a = \{1, 0.25, 0.01\}$  variou conforme os melhores valores de CAAM, que foram  $\delta = 1$ ,  $\delta' = 0.25$  e  $\beta = 0.01$ . Por exemplo, quando  $c = 8$ , temos que  $y$  vale no máximo 0.25 valor fixo de  $\delta'$  em CAAM (e os demais valores da saída decrescem conforme a curva da função). De forma similar,  $c = 2$ , temos que  $y$  vale no máximo 1 ( $\delta$ ) e quando  $c = 200$ , temos que  $y$  vale no máximo 0.01 ( $\beta$ ).

Assim, temos as quatro avaliações usadas neste experimento conforme especificadas a seguir:

1. Os valores de  $\delta$ ,  $\delta'$  e  $\beta$  valem respectivamente 1.0, 0.25 e 0.01.
2. Os valores de  $\delta$ ,  $\delta'$  valem respectivamente 1.0, 0.25 e  $\beta$  varia de acordo com a função  $\left(\frac{1 - \cosh\left(\frac{t-200}{100}\right)}{200}\right)$
3. O valor de  $\delta$  vale 1.0 e os valores de  $\delta'$  e  $\beta$  variam de acordo com a função  $\left(\frac{1 - \cosh\left(\frac{t-200}{100}\right)}{c}\right)$ , onde  $c = 8$  para  $\delta'$  e  $c = 200$  para  $\beta$ .
4. Os valores de  $\delta$ ,  $\delta'$  e  $\beta$  variam de acordo com a função  $\left(\frac{1 - \cosh\left(\frac{t-200}{100}\right)}{c}\right)$ , onde  $c = 2$  para  $\delta$ ,  $c = 8$  para  $\delta'$  e  $c = 200$  para  $\beta$ .

A Figura 7.44 apresenta os resultados em termos das médias e das variações dessas médias, no eixo  $y$  utilizando-se  $N = 12$  robôs para cada uma das quatro avaliações, apresentadas no eixo  $x$ . Neste gráfico é apresentado a média dos resultados representada pelo símbolo  $\circ$ , e a os fios de bigode dos dados também são apresentados mostrando até onde os dados alcançaram. A partir deste conjunto de dados, conclui-se que a melhor variação é a  $V_3$ , pois a média bem como os valores máximos e mínimos alcançados são os menores dentre as demais combinações. Essa combinação tende a ser a melhor, pois mesmo que os valores de  $\delta'$  e  $\beta$  tornem-se muito próximos de 0 com o passar do tempo, ainda tem-se a variável  $\delta$  desempenhando a função de prover o espalhamento dos robôs evitando colisões pelo agrupamento em áreas próximas.

Posteriormente, foi realizado um experimento para configurar o melhor ambiente de homing utilizado nos experimentos posteriores. Uma boa configuração do sistema é demanda uma utilização uniforme de todos os ninhos, pois evita o efeito de inércia (Yuan and Tan, 2007), efeito de obstrução (Yamamoto et al., 2007) e o efeito de formação de filas (Alizadeh, 2011). No entanto, é importante apresentar usar uma estratégia de distância de cálculo de densidade que considere os obstáculos móveis criados a partir do espalhamento das chamas do fogo alteram a dinâmica do sistema (conforme o baseado em (Zheng et al., 2012)) e o comportamento emergente do mesmo, conforme foi apresentado em experimentos anteriores. As melhores configurações dos experimentos anteriores foram consideradas. Para isso, um experimento que utilizou-se  $N = 15$  robôs foi o responsável por guiar nossos experimentos. Nesse experimento é relatado qual é a melhor distância

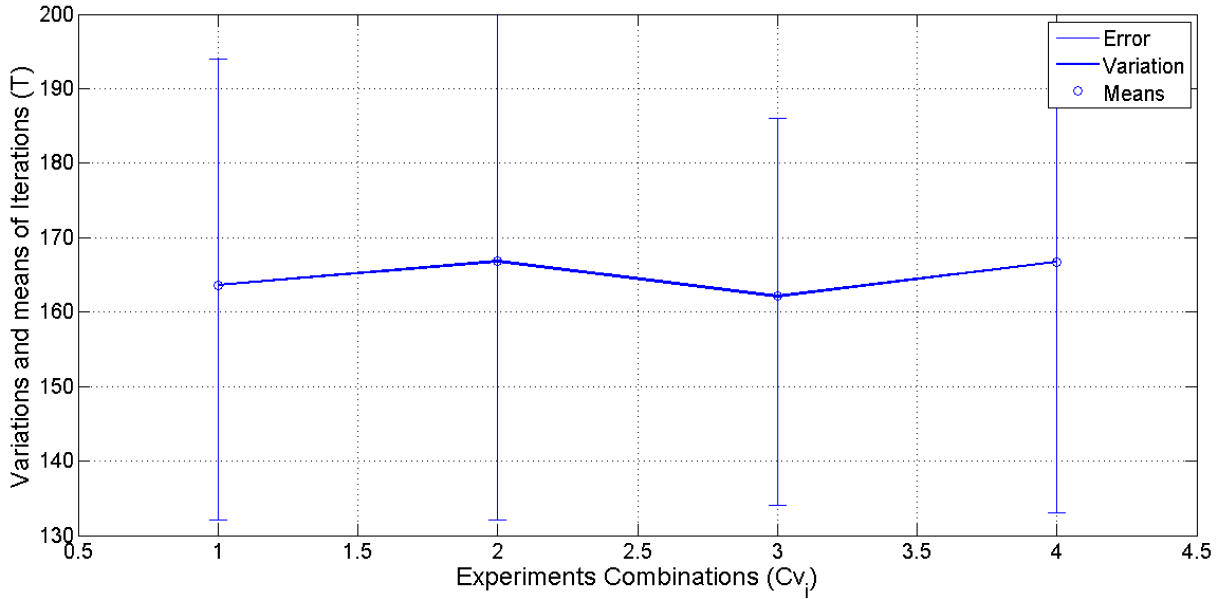


Figura 7.44: Análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = 12$  para as seguintes variações de atualização de  $\delta$ ,  $\delta'$  e  $\beta$ :  $CV_1 = \{\delta = 1, \delta' = 0.25, \beta = 0.01\}$ ,  $CV_2 = \{\delta = 1, \delta' = 0.25, \beta = \cosh\}$ ,  $CV_3 = \{\delta = 1, \delta' = \cosh, \beta = \cosh\}$ ,  $CV_4 = \{\delta = \cosh, \delta' = \cosh, \beta = \cosh\}$

$d_t^A$  onde passa a ser importante o cálculo da quantidade de robôs próximos (densidade  $\rho_{A^k}^t$ ) a cada um dos ninhos. A Figura 7.45 apresenta os resultados referentes às distâncias  $d_t^A = \{3, 7, 10, 14, 18\}$ . Neste experimento a melhor distância a ser considerada é  $d_t^A = 10$ . Distâncias muito grandes podem apresentar intersecções das faixas de distâncias entre os dois ninhos e distâncias muito pequenas podem não ser suficientes para evitar as aglomerações.

A partir da configuração da melhor distância do *homing* no experimento anterior, um novo experimento foi realizado a fim de contrastar diferentes modelos de escolha de célula para os estados de *homing* e *searching*, bem como a combinação de ambos. As configurações do sistema consideram  $Q = 10$  com política de fila que esvazia a memória quando a mesma está cheia e é usada a política de fila quando o robô encontra-se em deadlock. As melhores configurações de atualização de  $\delta$ ,  $\delta'$  e  $\beta$  também foram consideradas. Adicionalmente, o foco de incêndio encontra-se na posição central do reticulado. Cinco diferentes combinações de realização de movimento foram investigadas:  $D_k$  ( $k$  representa a variação  $N = \{3, 6, 9, 12, 15\}$ ) *homing* e *searching* de primeira escolha,  $P_k$  *homing* probabilista elitista  $c_{max}^t(ij) = 1000$  e *searching* de primeira escolha,  $V_k$  *homing* probabilista  $c_{max}^t(ij) = 100$  e *searching* de primeira escolha,  $A_k$  *searching* e *homing* probabilistas  $c_{max}^t(ij) = 100$  e  $E_k$  *searching* elitista  $c_{max}^t(ij) = 1000$  e *homing* com probabilista  $c_{max}^t(ij) = 100$ . A Figura 7.46 apresenta os resultados das simulações, onde o eixo  $x$  apresenta as diferentes combinações para cada valor de  $N$  e o eixo  $y$  apresenta o valor do tempo  $T$  gasto para a simulação. Neste gráfico é apresentado a média dos resultados representada pelo símbolo  $+$ , bem como o seu intervalo de confiança de 95% representado



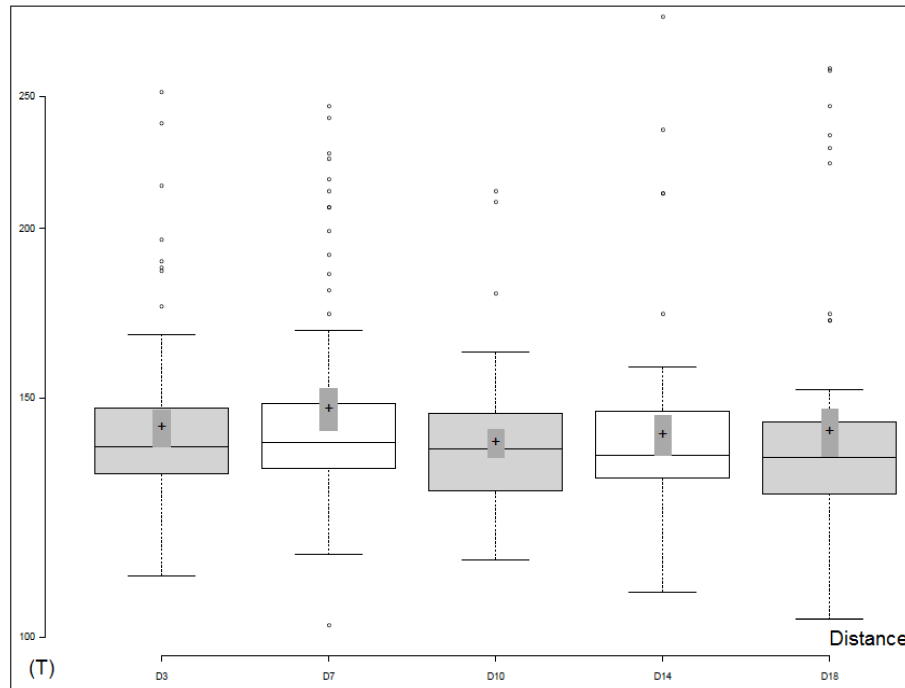


Figura 7.45: Análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = 12$  para o cálculo de densidades utilizando-se as seguintes distâncias para a mudança de cálculo probabilista e visão de ninho  $d_i^A = \{3, 7, 10, 14, 18\}$ .

por ■ que sombreia o símbolo da média, quanto menor o comprimento deste sombreado, maior é a confiança da média, ou seja, menor é o desvio padrão. Os valores de *outliers* estão representados pelo símbolo  $\circ$  e a mediana está representada pelo traço do meio de cada caixa do gráfico. A partir deste conjunto de dados, conclui-se que a melhor variação é  $E_k$  considera *searching* e *homing* probabilistas  $c_{max}^t(ij) = 100$ , pois para todas as variações de  $N$  temos que todas as médias se aproximam mais da mediana de cada um dos diagramas de caixa, mostrando uma maior tendência de distribuição simétrica dos dados. Adicionalmente, pode-se observar que a maioria das caixas que mostram o intervalo de confiança para esta combinação apresentam comprimentos curtos, mostrando maior confiança da média e conseqüentemente, um menor desvio padrão dos dados dessas simulações. Por fim, os dados apresentam o menor número de *outliers*, dentre os demais dados observados.

Posteriormente, a Figura 7.47 apresenta um experimento para validar se mesmo com o efeito de fogo investigado no presente trabalho, criando obstáculos temporários ao longo do reticulado, ainda é possível obter uma boa distribuição de tarefa para cada robô. Para efeitos de comparação, foi utilizado o melhor experimento apresentado anteriormente com o experimento onde considera-se apenas o *homing* probabilista com elitismo. A partir do gráfico da Figura 7.47 é possível observar que o modelo de escolha do *homing* probabilista apresentou um robô de label 3 que não foi capaz de encontrar nenhum objeto e o robô 1 foi capaz de encontrar 5 objetos. No modelo que agrega *homing* e *searching* com processos estocásticos para a escolha da célula apresentou uma melhor distribuição e cooperação

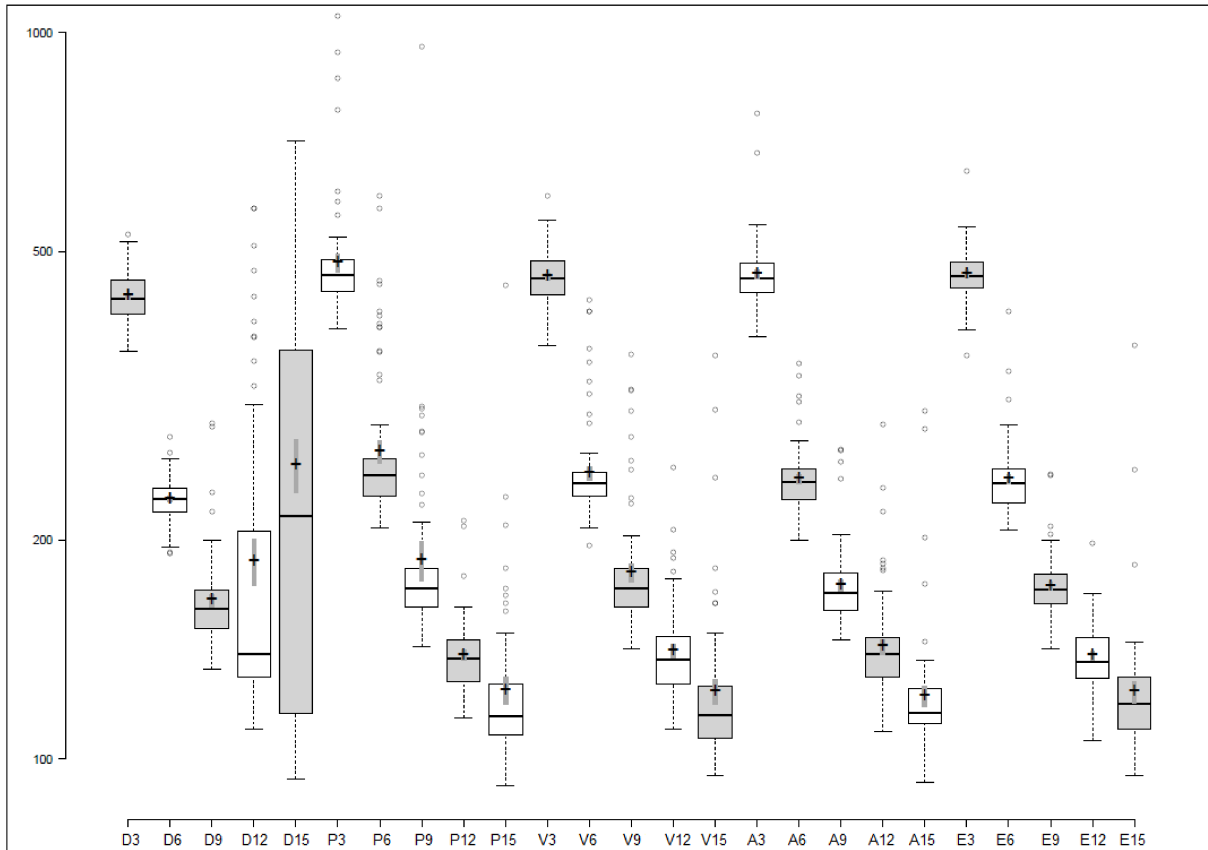


Figura 7.46: Análise de variação do tempo  $T$  para a conclusão da tarefa de busca e resgate utilizando-se  $N = \{3, 6, 9, 12, 15\}$  para as seguintes combinações:  $D_i$  determinístico,  $P_i$  probabilista elitista,  $V_i$  *homing* com visão de ninho,  $A_i$  *searching* e *homing* com processos estocásticos e  $E_i$  *searching* elitista e *homing* com processos estocásticos.

da tarefa pelos agentes (Winfield, 2009), pois todos conseguem cooperar ativamente na tarefa, já que eles pegam em média 3 objetos, exceto o robô 8 que pega apenas 2 objetos e os robôs 5 e 6 que capturam 4 objetos, sendo este o mais adequado para este tipo de tarefa.

A Figura 7.48 apresenta um experimento a fim de verificar se a configuração do experimento anterior foi capaz de realizar uma boa distribuição dos ninhos para ambas as configurações. A partir dos resultados plotados, é possível observar que a partir da utilização da estratégia que utiliza apenas o *homing* probabilista com elitismo, não foi possível obter uma boa distribuição dos ninhos, uma vez que o mesmo considera uma sub-utilização do ninho  $A_0$  (utilizado somente 4 vezes), enquanto o ninho  $A_1$  foi sobrecarregado. Adicionalmente, é possível observar que a estratégia com *homing* probabilista elitista apresenta mais passos de tempo  $T$  para a execução, do que o caso em que ambos os estados são de escolhas estocásticas, já que neste caso, observa-se uma distribuição igualitária 50% entre a utilização dos ninhos ( $A_0$  e  $A_1$  foram utilizados 15 vezes).

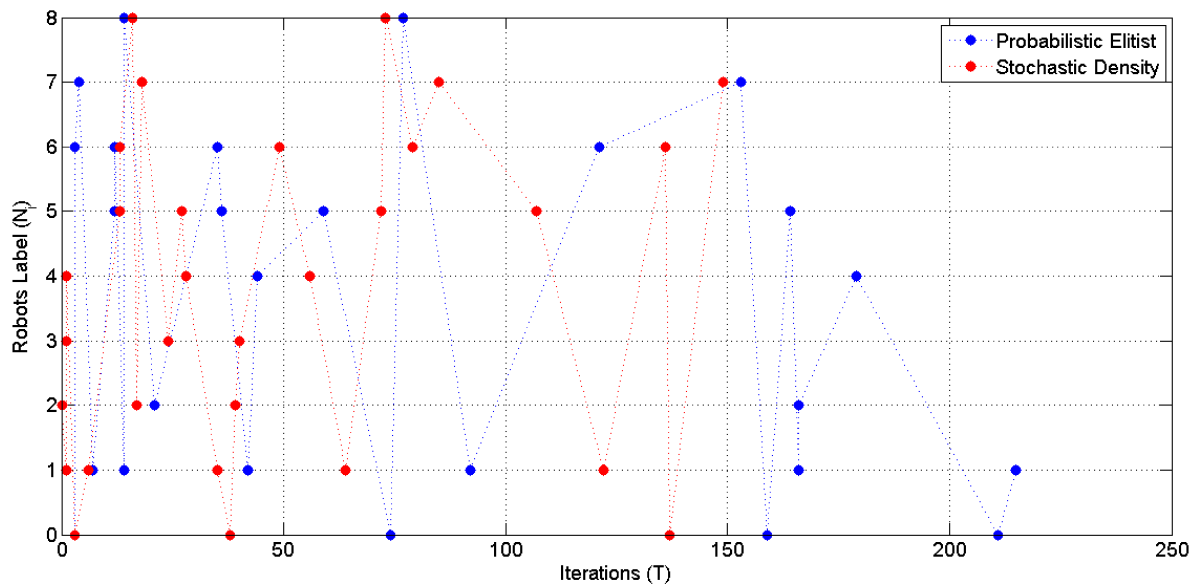


Figura 7.47: Análise de variação do tempo  $T$  para a conclusão da tarefa de busca utilizando-se  $N = 9$  para as seguintes combinações: probabilista elitista e estocástico com densidade.

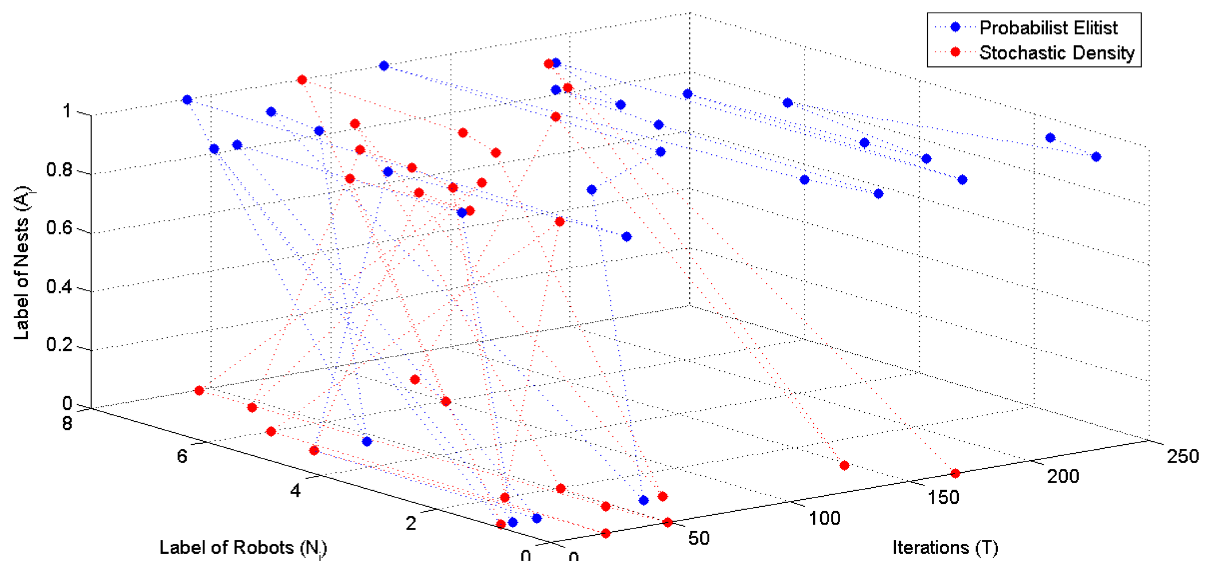


Figura 7.48: Análise de variação do tempo  $T$  para a conclusão da tarefa de depósito utilizando-se  $N = 9$  para as seguintes combinações: probabilista elitista e estocástico com densidade.

### 7.5.3 Testes de análise de cobertura do ambiente

Inicialmente, foi medido o nível de cobertura do ambiente em termos do número de passos que o time de robôs exerceu sobre cada célula durante a tarefa de busca e resgate. As Figuras 7.49 e 7.50 apresentam os resultados dos experimentos com  $N = \{3, 6, 9, 12, 15\}$ , que serão analisados a seguir.

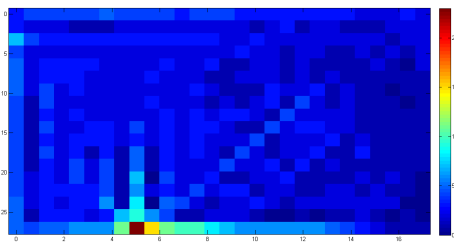
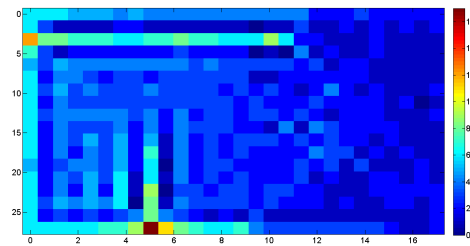
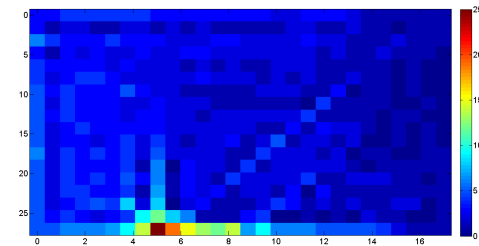
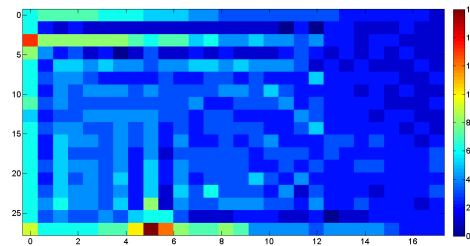
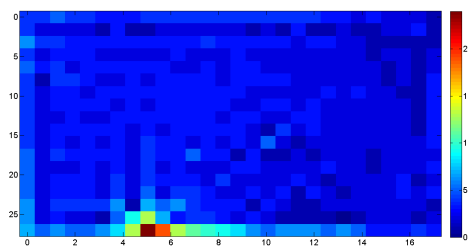
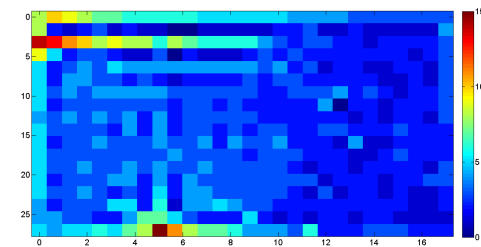
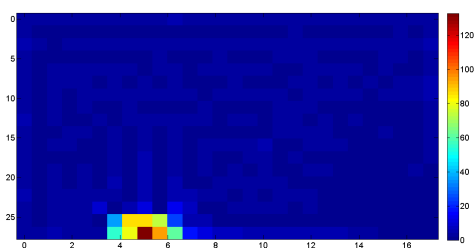
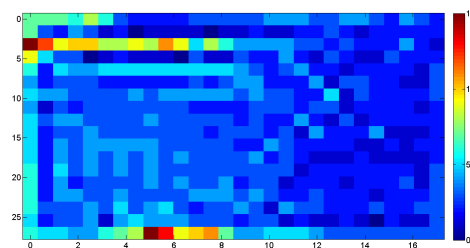
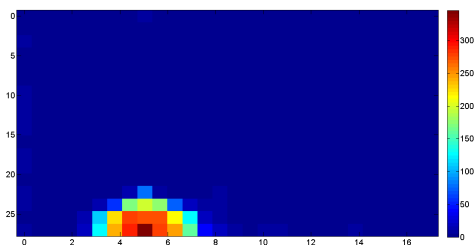
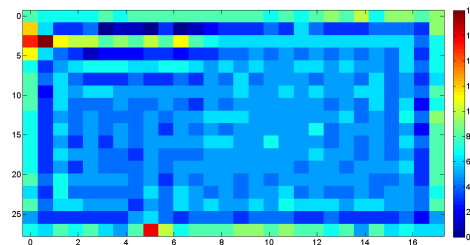
(a)  $N = 3, T = 408$ (b)  $N = 3, T = 482$ (c)  $N = 6, T = 201$ (d)  $N = 6, T = 258$ (e)  $N = 9, T = 233$ (f)  $N = 3, T = 169$ (g)  $N = 12, T = 118$ (h)  $N = 12, T = 129$ (i)  $N = 15, T = 374$ (j)  $N = 15, T = 174$ 

Figura 7.49: Mapas de temperatura que apresentam a cobertura do ambiente em relação ao número de passos do time de robôs utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs. A coluna da esquerda representa uma cobertura do usando homing probabilista elitista e a coluna da direita representa uma cobertura do usando homing com cálculo de densidades.

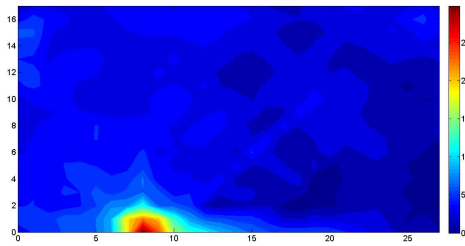
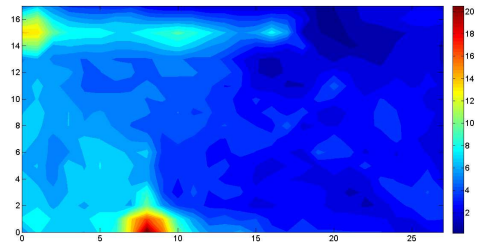
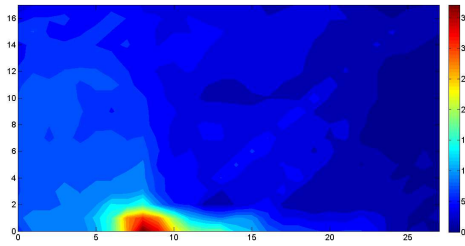
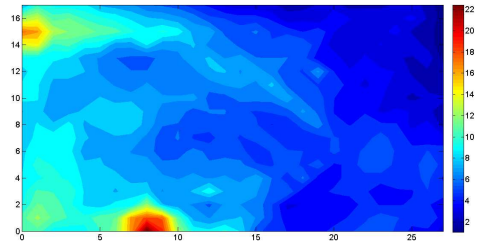
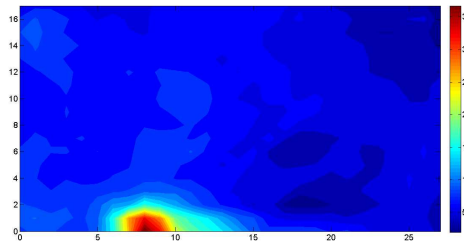
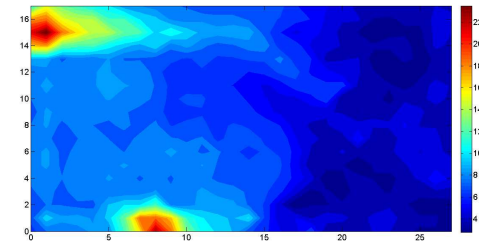
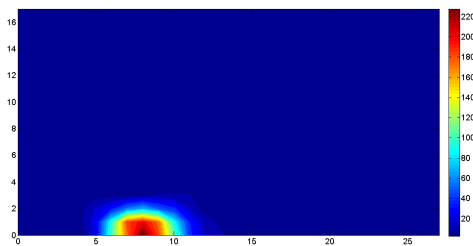
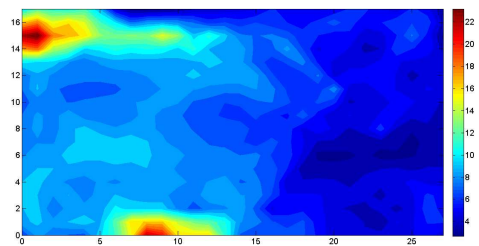
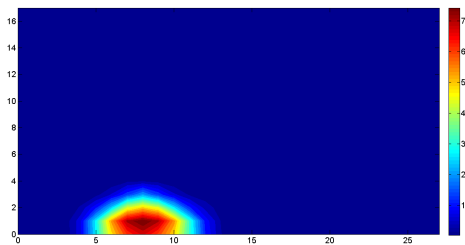
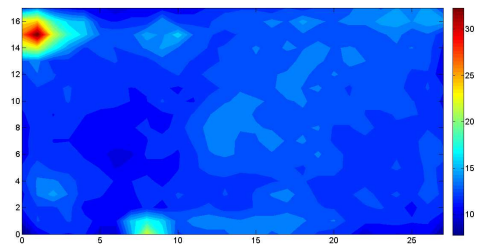
(a)  $N = 3, T = 408$ (b)  $N = 3, T = 482$ (c)  $N = 6, T = 201$ (d)  $N = 6, T = 258$ (e)  $N = 9, T = 233$ (f)  $N = 3, T = 169$ (g)  $N = 12, T = 118$ (h)  $N = 12, T = 129$ (i)  $N = 15, T = 374$ (j)  $N = 15, T = 174$ 

Figura 7.50: Mapas de temperatura que apresentam a cobertura do ambiente em relação à quantidade de feromônio do time de robôs utilizando-se  $N = \{3, 6, 9, 12, 15\}$  robôs. A coluna da esquerda representa uma cobertura do usando homing probabilista elitista e a coluna da direita representa uma cobertura do usando homing com cálculo de densidades.

Os experimentos do lado esquerdo da Figura 7.49 referem-se à uma cobertura do ambiente para o mesmo time com probabilismo apenas no homing utilizado no experimento anterior, enquanto que os experimentos do lado direito referem-se aos experimentos para time que apresenta escolha probabilista para ambos os estados e também campo de visão  $E_k$ , também apresentado no experimento anterior. Pode-se observar que em todos os experimentos do lado direito o algoritmo convergiu para uma boa cobertura do ambiente. O mesmo não é observado no experimento do lado esquerdo, onde à medida que o time de robôs cresce é observado também um crescimento de regiões vermelhas próximo aos ninhos, uma vez que, vários robôs ficam parados pelo efeito de inércia (Yuan and Tan, 2007) e efeito de obstrução (Guo and Huang, 2008), sendo fenômenos típicos prejudiciais à convergência do time de robôs. A Figura 7.50 apresenta os resultados dos experimentos com  $N = \{3, 6, 9, 12, 15\}$  que fornece a cobertura do ambiente em termos dos mapas de feromônio capturados no último instante  $T$  da simulação. Os experimentos dos lados esquerdo e direito referem-se às mesmas configurações do time de robôs e ambiente do experimento anterior. Assim como no experimento da figura anterior, também é possível observar uma melhor distribuição de feromônio para o experimento do lado direito, enquanto que o lado esquerdo, à medida que o número de robôs aumenta significativamente, também é possível observar um aumento de aglomerações ao longo de um dos ninhos, resultando em simulações mais lentas e menos propícias para a aplicação em robótica.

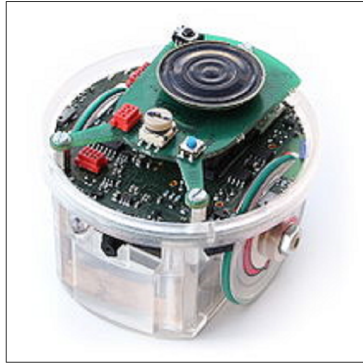
## 7.6 Experimentos no ambiente Webots (RCAAM)

Os experimentos realizados no ambiente de simulação Webots EDU foram implementados a partir do modelo CAAM para a tarefa de forrageamento. Nos testes iniciais apenas o estado de homing foi considerado para a simulação em um ambiente de  $5 \times 5$  células para dois times de robôs  $N = 1$  e  $N = 3$ . Posteriormente, os testes mais extensos consideraram  $N = 6$  robôs e  $F = 7$  alimentos espalhados dentro do ambiente de simulação de  $20 \times 30$  células. Neste segundo teste, os quatro estados (homing, grabbing, searching, depositing) foram executados e as seguintes variações foram avaliadas: modelo com odometria sem rotação de robôs, modelo com odometria e rotação de robôs, modelo com GPS e sem correção de guinada de robôs, modelo com GPS e com correção de guinada dos robôs.

### 7.6.1 Implementação do estado *homing*

Primeiramente, foi realizado um experimento no ambiente de simulação Webots EDU com o robô e-Puck (dimensão  $l = 7$  cm) apresentado na Figura 7.51a. Neste caso, a simulação apresenta apenas o processo de *homing* com  $\alpha = 0$ , ou seja, apenas o piso estático foi utilizado para direcionar o robô até o ninho mais próximo. Algumas alterações foram necessárias para que fosse possível implementar o passo dinâmico do robô. Inicialmente,

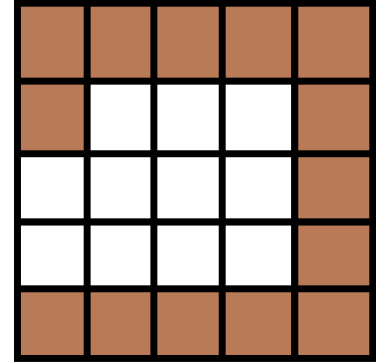
apenas um robô ( $N = 1$ ) foi utilizado nos experimentos. O ambiente utilizado no experimento inicial tem dimensão de  $5 \times 5$  e o tamanho da célula é de  $l = 20 \text{ cm}$  e está apresentado na Figura 7.51b com seus respectivos valores e a Figura 7.51c apresenta uma grid usado apenas para representar o campo de piso realçando as células da simulação. Os resultados desses experimentos foram publicados em (Lima et al., 2017b).



(a) Robô e-Puck utilizado na simulação (EPFL, 2013).

500	500	500	500	500
500	2.5	3.5	4.5	500
1.0	2.0	3.0	4.0	500
1.0	2.0	3.0	4.0	500
500	500	500	500	500

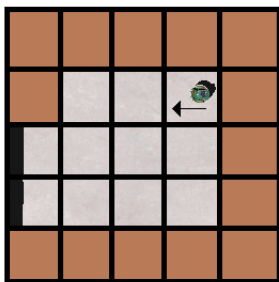
(b) Campo de piso estático usado na simulação.



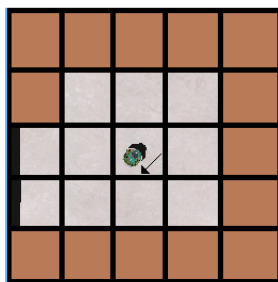
(c) Ambiente de experimentação Webots.

Figura 7.51: Campo de piso utilizado na simulação do robô e-Puck no software Webots com os valores calculados a partir do algoritmo proposto em (Varas et al., 2007).

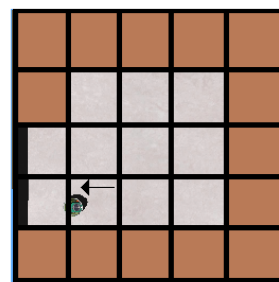
O resultado da implementação do modelo utilizando-se o robô e-Puck e o ambiente de simulação Webots EDU está apresentado na Figura 7.52. Nessa figura, algumas setas foram inseridas manualmente para facilitar a visualização da orientação da frente do robô. Inicialmente o único robô é posicionado na posição central da célula de valor  $W_{static}^A = 4.5$  (ver Figura 7.52a), com uma orientação voltada para o oeste. Posteriormente, o robô é rotacionado  $45^\circ$  sentido anti-horário e caminha  $d = 28.3 \text{ cm}$  para a célula central do reticulado cujo valor é  $W_{static}^A = 3.0$  (ver Figura 7.52b) com uma orientação voltada para o sudoeste. Em seguida o robô caminha  $d = 28.3 \text{ cm}$  para a célula de valor  $W_{static}^A = 2.0$  mantendo a mesma orientação. Por fim o robô rotaciona  $45^\circ$  sentido horário (ver Figura 7.52c) e caminha por  $d = 20 \text{ cm}$ , chegando à saída de valor  $W_{static}^A = 1.0$  (ver Figura 7.52d), que é a sua meta, o que significa que ele conseguiu depositar o alimento no ninho.



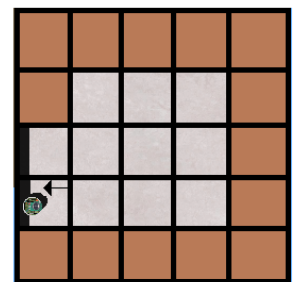
(a)  $t = 0$ .



(b)  $t = 1$ .



(c)  $t = 2$ .



(d)  $t = 3$ .

Figura 7.52: Simulação do robô e-Puck no software Webots  $N = 1$ .

O resultado da implementação do modelo utilizando-se  $N = 3$  robôs e-Puck e o am-

biente de simulação Webots EDU é apresentado na Figura 7.53. Neste experimento, os 3 robôs realizam o processo de *homíng*, e cada robô carrega um único alimento. Inicialmente eles estão dispostos nas células do campo de piso com valores  $W_{static}^A = \{4.5, 4.0, 4.0\}$ , respectivamente, todos inicialmente orientados para o oeste.

Na Figura 7.53a temos a posição e orientação inicial dos três robôs alinhados no campo de piso  $W_{static}^A = \{4.5, 4.0, 4.0\}$  no tempo  $t = 0$ , denominados robôs  $N = \{R_1, R_2, R_3\}$ , respectivamente. Posteriormente, cada robô envia sua posição atual para o servidor e ele calcula as novas posições de todos os membros do time (movimento baseado na menor escolha no piso estático). Os robôs  $R_2$  e  $R_3$  entraram em conflito. Após um sorteio realizado pelo servidor para resolução do conflito, apenas o robô  $R_2$  realiza o movimento em  $t = 1$  e o robô  $R_3$  deve aguardar. Concomitantemente, o robô  $R_1$  também realiza o seu movimento. Para ambos, é necessário realizar uma rotação de  $45^\circ$  antes de realizar o movimento. Em seguida, apenas os robôs  $R_1$  e  $R_2$  realizam movimentos, uma vez que o robô  $R_3$  ainda se encontra sem possibilidade de realizar movimento, por não ser capaz de encontrar nenhuma célula de menor valor livre. No instante de tempo  $t = 3$ , os robôs  $R_1$  e  $R_2$  encontram-se em suas posições finais, portanto, não realizarão mais movimentos, e o robô  $R_3$  realiza o seu primeiro movimento para a célula central do reticulado. No próximo passo, em  $t = 4$ , o robô realiza o movimento em linha reta. Por fim, em  $t = 5$  o robô  $R_3$  realiza o seu último movimento e o processo de *homíng* é finalizado.

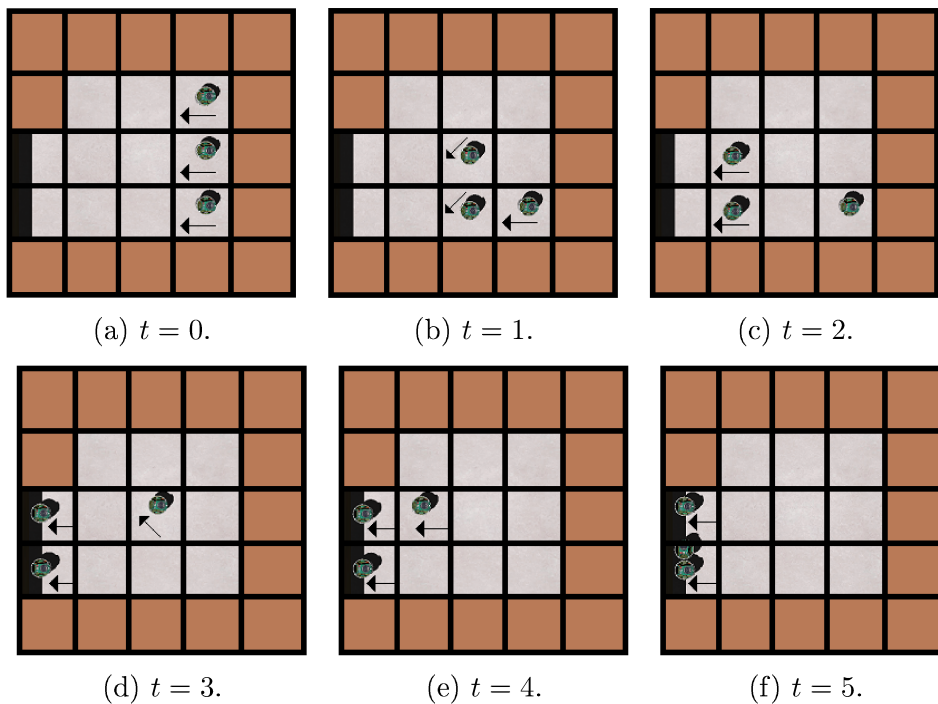


Figura 7.53: Simulação de 3 robôs do tipo e-Puck no software Webots ( $N = 3$ ).



### 7.6.2 Modelo com odometria sem manutenção do ângulo de rotação corrente

O primeiro experimento realizado para a tarefa de forrageamento completa, contemplando os 4 estados do modelo do robô (*searching*, *grabbing*, *homing* e *depositing*), utilizou a odometria para a movimentação do time. O principal objetivo desse experimento é avaliar o desempenho do modelo RCAAM em um ambiente que se aproxima mais das restrições reais de um ambiente físico. Para a visualização dos estados do modelo CAAM para a tarefa de forrageamento, inicialmente todos os robôs ficam com seus leds desligados. Quando um robô está procurando comida ou no estado de captura os leds ficam no estado vermelho. Quando um robô está no estado de homing ou de depósito, os leds ficam no estado verde. Além disso, os 6 robôs são inicializados em células contíguas próximas ao canto superior direito do ambiente, conforme apresentado na Figura 7.54a, todos com a mesma orientação: com sua frente voltada a oeste.

Neste primeiro experimento no ambiente de simulação Webots EDU foi utilizada uma função de odometria para estimar a posição e o Ângulo de rotação dos robôs pelo ambiente durante sua movimentação. A função de odometria utilizada foi baseada na função implementada no trabalho de (Oliveira et al., 2014). Além disso, foi utilizada a ideia de que todos os robôs devem retomar ao ângulo inicial de  $0^\circ$  após realizarem o seu movimento de rotação entre os possíveis ângulos  $\{-135^\circ, -90^\circ, -45^\circ, 0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ . Embora a estratégia seja simples de ser implementada, o movimento de retorno para o ângulo inicial provoca um atraso na simulação, o que pode ser desvantajoso para simulações onde existe um grande número de unidades de comida. Sabe-se que a odometria proporciona uma boa precisão em curto prazo, e é barata de implantar, além de permitir taxas de amostragem muito altas. No entanto, o acúmulo de vários de erros de orientação pode provocar grandes erros na estimação da posição, os quais vão aumentando proporcionalmente com a distância percorrida pelo robô, o que pode ser corroborado visualmente neste experimento. A captura das imagens foi realizada após a finalização do movimento, ou seja, quando todos os robôs voltam para o ângulo inicial de  $0^\circ$ . Assim, é possível perceber que todos os robôs estão na mesma direção.

A Figura 7.54 apresenta o resultado da simulação até o tempo  $T = 7$  com robôs e 7 unidades de comida. Para este experimento foi extraída a imagem de cada instante de tempo, bem como o tempo de execução em segundos (s) observado nos instantes de tempo  $t = \{0, 1, 2, \dots, 7\}$ .

No instante de tempo  $t = 0$  (na Figura 7.54a  $s = 0$ ) todos os robôs mantêm-se com os leds desligados e eles se preparam para a realização do primeiro movimento. Em seguida, na Figura 7.54b, no instante de tempo  $t = 1$ , os robôs ligam os seus leds na cor vermelha que indicam o único estado *searching* e realizam o primeiro movimento e utilizando a função de odometria.

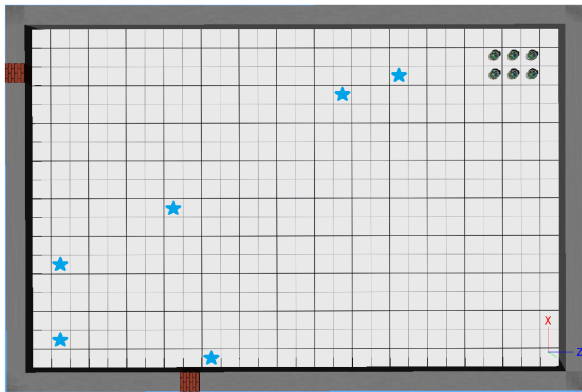
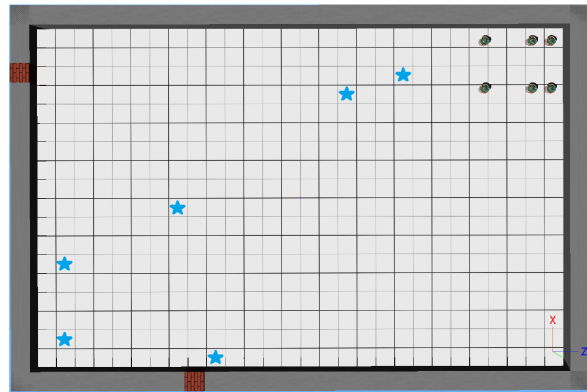
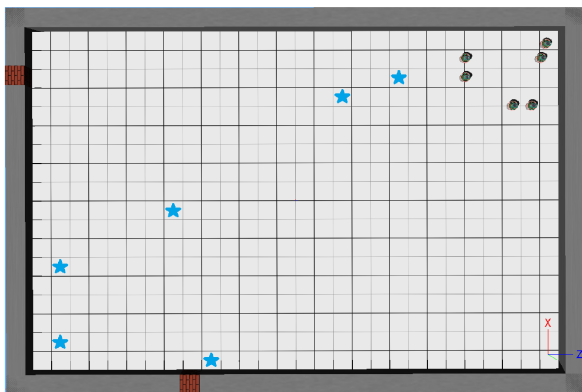
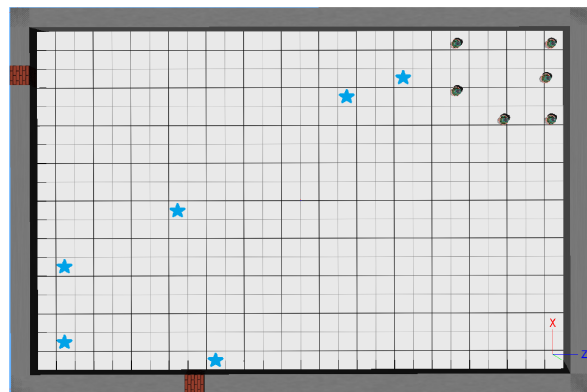
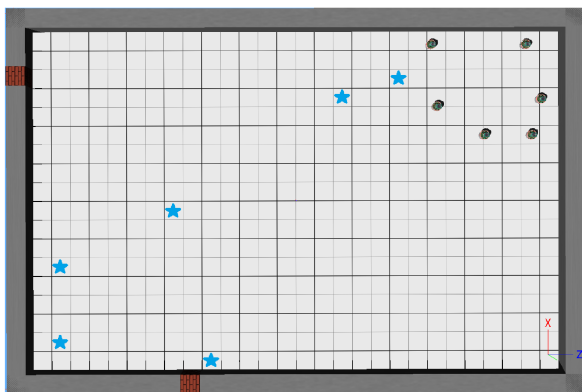
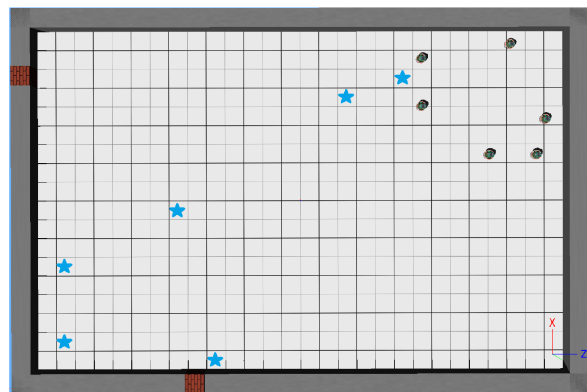
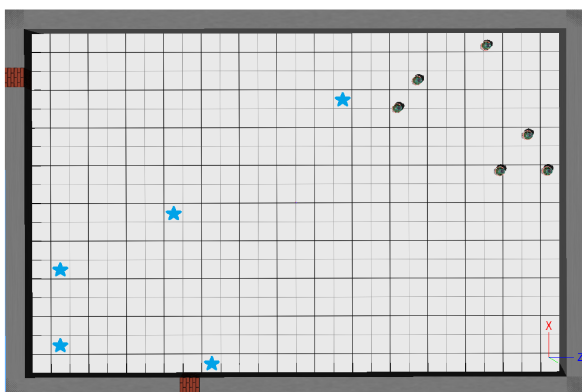
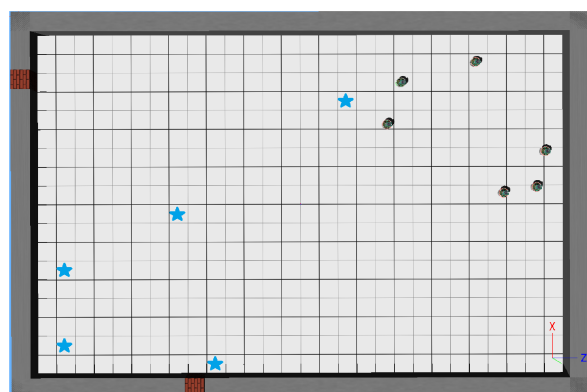
(a)  $t = 0$  e  $s = 0$ .(b)  $t = 1$  e  $s = 18.464$ .(c)  $t = 2$  e  $s = 36.512$ .(d)  $t = 3$  e  $s = 55.536$ .(e)  $t = 4$  e  $s = 69.760$ .(f)  $t = 5$  e  $s = 84.160$ .(g)  $t = 6$  e  $s = 102.576$ .(h)  $t = 7$  e  $s = 121.168$ .

Figura 7.54: Experimentos relativos à tarefa de forrageamento para 6 robôs utilizando-se odometria para localização do robô sem gravação de rotação.

Em  $t = 1$  já é possível perceber que os robôs não permaneceram centralizados nas suas respectivas células de destino. O tempo para que todos os robôs realizassem seus movimentos nesse primeiro passo de tempo e em seguida retornassem ao ângulo inicial foi de 18.464s. No instante de tempo  $t = 2$  o erro foi acumulado e os robôs no canto superior direito, que deveriam estar alinhados em suas respectivas células acabam se aproximando, conforme apresentado na Figura 7.54c. O tempo para a realização de todos os movimentos para  $t = 2$  foi de  $s = 36.512$ .

Nos instantes de tempo  $t = \{3, 4, 5, 6\}$  apresentados nas Figuras 7.54d, 7.54e, 7.54f, 7.54g os tempos de execução foram, respectivamente,  $s = \{55.536, 69.760, 84.160, 102.576\}$  e todos os robôs estão no estado de busca com os leds vermelhos ligados e não estão no centro de suas respectivas células. Por fim, no instante de tempo  $t = 7$  apresentado na Figura 7.54h, o tempo percorrido foi  $s = 121.168$ . Todos os robôs estão com os leds vermelhos ligados, exceto o robô que carrega uma unidade de comida e apresenta o led verde ligado. Assim, é possível verificar que todos os robôs estão bastante distantes do centro de suas células o que inviabiliza a implementação deste modelo porque os robôs podem colidir com os demais robôs ou com os obstáculos.

### 7.6.3 Modelo com odometria e manutenção do ângulo de rotação corrente

Para a realização deste segundo experimento no ambiente de simulação Webots EDU também foi utilizada a função de odometria, a mesma do experimento anterior. Entretanto, diferentemente do experimento anterior, quando o robô realiza uma rotação ao se deslocar para uma nova célula, após o deslocamento, o robô não retorna automaticamente à sua orientação original (ângulo  $0^\circ$ ). Ao invés disso, o robô mantém o ângulo corrente até que o próximo passo seja decidido, para tal, cada robô possui um atributo que guarda o seu ângulo de rotação somente entre os possíveis ângulos  $\{-135^\circ, -90^\circ, -45^\circ, 0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ . Essa estratégia reduz o tempo de simulação em relação ao experimento anterior, pois quando o robô realiza 2 ou mais passos em uma mesma direção, o robô mantém o deslocamento sem mudar a orientação. A captura das imagens foi efetuada após a finalização do movimento, ou seja, quando todos os robôs ficam parados para a sincronização do próximo movimento.

A Figura 7.54 apresenta o resultado da simulação nos instantes de tempo  $t = \{0, 1, 2, \dots, 7\}$ . Foi extraída a imagem de cada instante de tempo, bem como o tempo de execução em segundos (s) observados em cada um dos instantes de tempo  $t = \{0, 1, 2, \dots, 7\}$ . No instante de tempo  $t = 0$  (na Figura 7.54a  $s = 0$ ) todos os robôs mantêm seus leds desligados e eles preparam-se para a realização do primeiro movimento. Em seguida, na Figura 7.54b no instante de tempo  $t = 1$ , os robôs ligam os seus leds na cor vermelha e realizam o primeiro movimento utilizando a função de odometria. Neste passo já é possí-

vel perceber que os robôs não permanecem centralizados nas suas respectivas células de destino. O tempo para que todos os robôs realizassem seus movimentos foi de  $s = 12.233$ . Conforme apresentado na Figura 7.54c, no instante de tempo  $t = 2$  o erro foi acumulado durante a trajetória e os robôs no canto superior direito também estão desalinhados. O tempo para a realização de todos os movimentos para  $t = 2$  foi de  $s = 23.453$ . É possível observar que nos instantes de tempo  $t = \{3, 4, \dots, 7\}$  apresentados nas Figuras 7.54d, 7.54e, 7.54f, 7.54g os tempos de execução em segundos ( $s$ ) foram, respectivamente, de  $s = \{33.234, 44.668, 56.096, 65.088\}$  e todos os robôs estão no estado de busca com os leds vermelhos ligados. Por fim, no instante de tempo  $t = 7$  apresentado na Figura 7.54h corresponde ao tempo de execução  $s = 75.776$ , que é inferior ao tempo apresentado no primeiro experimento. Todos os robôs estão com os leds vermelhos ligados, exceto o robô que carrega uma unidade de comida e apresenta o led verde ligado. Assim como no experimento anterior, é possível verificar que todos os robôs também estão bastante distantes do centro de suas células o que inviabiliza a implementação desta abordagem, uma vez que os robôs vão colidir entre si ou com os obstáculos ao longo de suas trajetórias. Como é possível observar, embora tenha ocorrido uma queda no tempo de simulação, a odometria retornou um tempo satisfatório de execução.

#### 7.6.4 Modelo com GPS e sem correção de giro

Para a realização deste experimento no ambiente de simulação Webots EDU foi utilizado o *slot* de extensão dos robôs e-Pucks denominado GPS (*slot*). Como foi observado nos experimentos anteriores, a função de odometria não garante uma boa coordenação do time de robôs por apresentar um erro de posicionamento muito grande. Por outro lado, o GPS, funciona como um atributo que garante a correção das coordenadas do robô a partir de um limiar de erro aceitável. Adicionalmente, assim como no experimento anterior, foi utilizada a ideia de que cada robô possui um atributo que guarda o seu ângulo de rotação entre os possíveis ângulos  $\{-135^\circ, -90^\circ, -45^\circ, 0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ . Essa abordagem reduz o tempo de simulação em relação ao experimento que força o robô a retornar à orientação inicial, como demonstrado no experimento anterior. A captura das imagens foi feita após a finalização do movimento, ou seja, quando todos os robôs ficam parados para a sincronização do próximo movimento. A Figura 7.56 apresenta o resultado da simulação até o instante de tempo  $t = 7$ . Para este experimento foi extraída a imagem de cada instante de tempo, bem como o tempo de execução em segundos ( $s$ ). No instante de tempo  $t = 0$  (na Figura 7.56a,  $s = 0$  segundos) todos os robôs mantêm os leds desligados e eles se preparam para a realização do primeiro movimento. Em seguida, na Figura 7.56b no instante de tempo  $t = 1$ , os robôs ligam os seus leds na cor vermelha e realizam o primeiro movimento. Neste passo é possível perceber que os robôs permanecem centralizados nas suas respectivas células de destino, diferentemente dos experimentos anteriores.

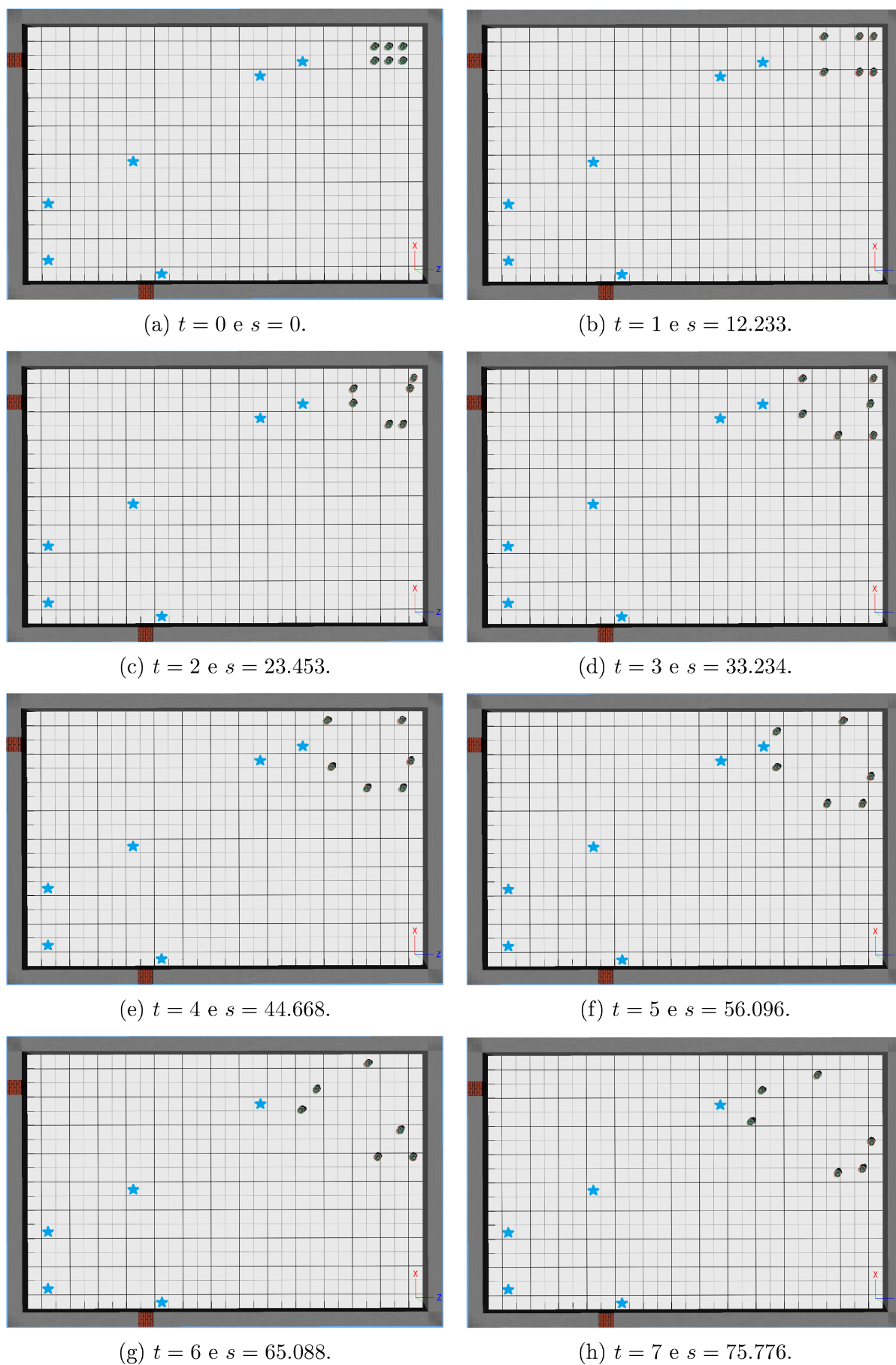


Figura 7.55: Experimentos relativos à tarefa de forrageamento para 6 robôs utilizando-se odometria para localização do robô com gravação de rotação.

O tempo para que todos os robôs realizassem seus primeiros movimentos foi de  $s = 14.045$ . A Figura 7.56c mostra o time de robôs no instante de tempo  $t = 2$  e o tempo para a realização do movimento por todos os robôs foi de  $s = 26.023$  segundos. É possível observar que nos instantes de tempo  $t = \{3, \dots, 7\}$  apresentados nas Figuras 7.56d, 7.56e, 7.56f e 7.56g os tempos de execução em segundos foram, respectivamente, de  $s = \{37.303, 44.668, 50.190, 70.557\}$  e todos os robôs estão no estado de busca com os leds vermelhos ligados. Por fim, no instante de tempo  $t = 7$  apresentado na Figura 7.56h possui tempo em segundos  $s = 81.593$ , que é superior ao apresentado no experimento anterior. Todos os robôs estão com os leds vermelhos ligados, exceto o robô que carrega uma unidade de comida e apresenta o led verde ligado. É possível verificar que o time de robôs encontra-se centralizado em suas respectivas células. Dessa forma, o uso do GPS foi importante para a correção da posição do robô. Embora a figura só apresente os primeiros 7 passos de tempo, o erro final apresentado neste experimento só é perceptível em instantes de tempo superiores. Como veremos na próxima seção, esse problema ocorre devido ao erro de guinada.

### 7.6.5 Modelo com GPS e com correção de giro

O quarto experimento realizado no ambiente de simulação Webots EDU utilizou o *slot* de extensão dos robôs e-Pucks denominado GPS (*slot*) e o *slot* de inércia (*InertialUnit*) que é capaz de fazer o controle de guinada do robô. Como foi observado no experimento anterior, o GPS por si só não garante que ao final da trajetória, através do balanceamento de correção do erro das coordenadas, seja possível corrigir o problema do carregamento do erro. Assim, o *slot* de correção de guinada foi adicionado com o propósito de corrigir o ângulo de rotação do robô que acaba sofrendo pequenas alterações devido ao atrito com o ambiente. Cada robô possui um atributo que guarda o seu ângulo de rotação entre os possíveis ângulos  $\{-135^\circ, -90^\circ, -45^\circ, 0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ . A captura das imagens foi executado após a finalização do movimento, ou seja, quando todos os robôs ficam parados para a sincronização do próximo movimento.

A Figura 7.56 apresenta o resultado da simulação até o instante de tempo  $t = 7$ . Para este experimento foi extraído a imagem de cada instante de tempo, bem como o tempo de execução em segundos ( $s$ ) observados em cada um dos instantes  $t = \{0, 1, 2, \dots, 7\}$ . No instante  $t = 0$  (na Figura 7.56a  $s = 0$  segundos) todos os robôs mantêm-se com os leds desligados e eles preparam-se para a realização do primeiro movimento. Em seguida, na Figura 7.56b no instante de tempo  $t = 1$  os robôs ligam os seus leds na cor vermelha e realizam o primeiro movimento num tempo de  $s = 14.234$  segundos. A Figura 7.56c mostra o time de robôs no instante  $t = 2$  e o tempo para a realização do movimento por todos os robôs foi de  $s = 26.992$  segundos.

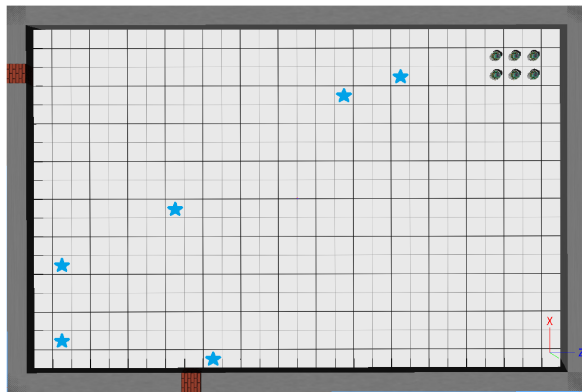
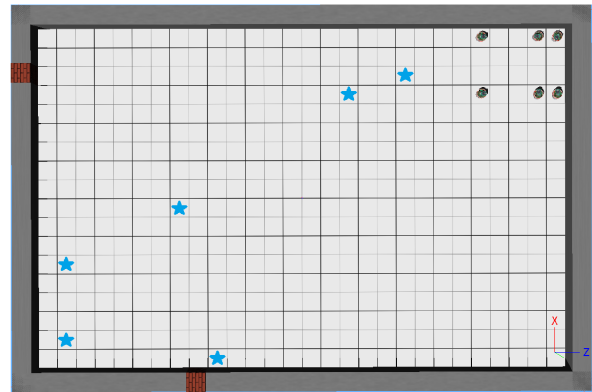
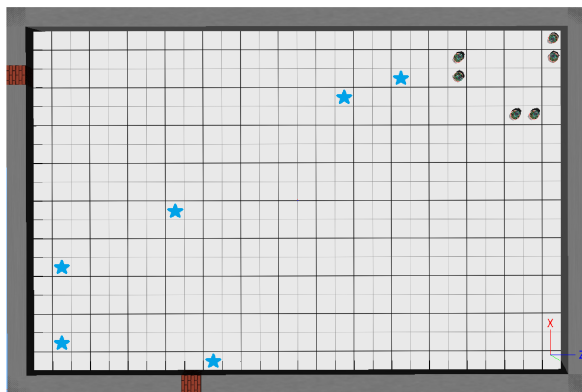
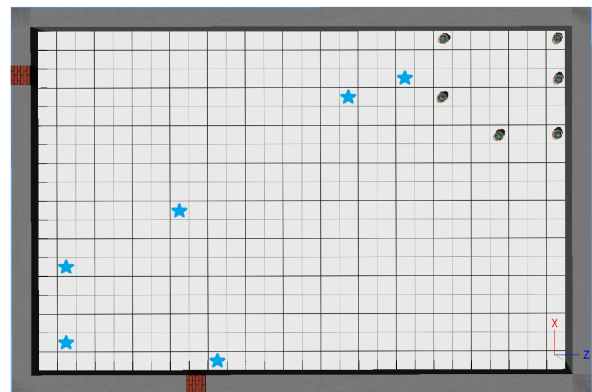
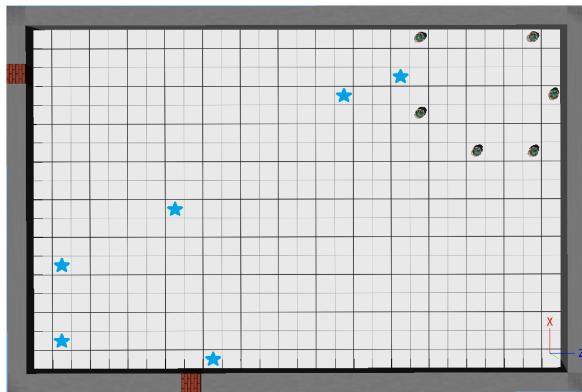
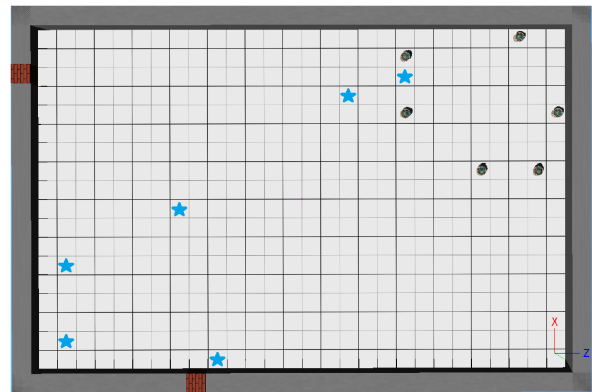
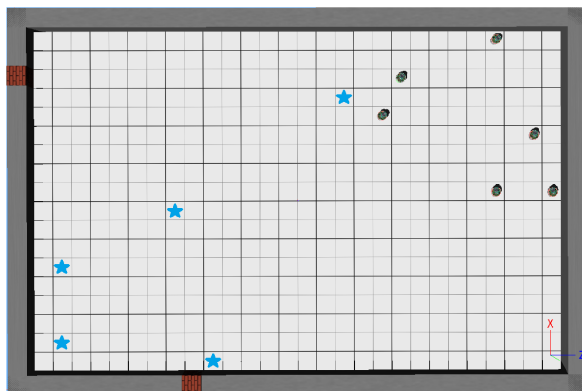
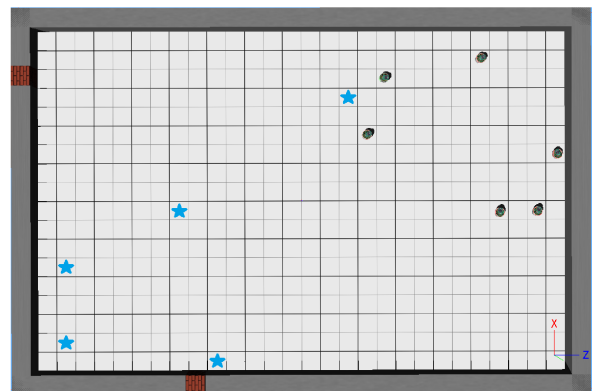
(a)  $t = 0$  e  $s = 0$ .(b)  $t = 1$  e  $s = 14.045$ .(c)  $t = 2$  e  $s = 26.023$ .(d)  $t = 3$  e  $s = 37.303$ .(e)  $t = 4$  e  $s = 44.668$ .(f)  $t = 5$  e  $s = 50.190$ .(g)  $t = 6$  e  $s = 70.557$ .(h)  $t = 7$  e  $s = 81.593$ .

Figura 7.56: Experimentos relativos à tarefa de forrageamento para 6 robôs sem rotação e com utilização de GPS para correção do movimento.

É possível observar que nos instantes de tempo  $t = \{3, \dots, 7\}$  apresentados nas Figuras 7.56d, 7.56e, 7.56f, 7.56g, onde os tempos de execução em segundos foram, respectivamente, de  $s = \{37.945, 53.575, 72.254, 88.236\}$  e todos os robôs estão no estado de busca com os leds vermelhos ligados. Por fim, no instante de tempo  $t = 7$  apresentado na Figura 7.56h possui tempo em segundos  $s = 101.229$ , que é superior ao apresentado no experimento anterior. Todos os robôs estão com os leds vermelhos ligados, exceto o robô que carrega uma unidade de comida e apresenta o led verde ligado. É possível observar que, assim como no experimento anterior, os robôs encontram-se centralizados em suas células após 7 passos de tempo. Entretanto, diferentemente do experimento anterior foi possível controlar de forma mais adequada o time de robôs devido à utilização do *slot* adicional de inércia, que permite ajustar cada robô através de uma correção discreta no ângulo de rotação. Para melhor visualizar essa diferença, a Figura 7.57 apresenta a situação do time de robôs no tempo  $t = 30$ . Na Figura 7.57a não foi aplicada a correção de guinada (experimento da seção anterior), enquanto que na Figura 7.57b foi feita com o auxílio do *slot* de inércia. É possível observar que 7.57b, os robôs encontram-se mais centralizados em suas respectivas células.

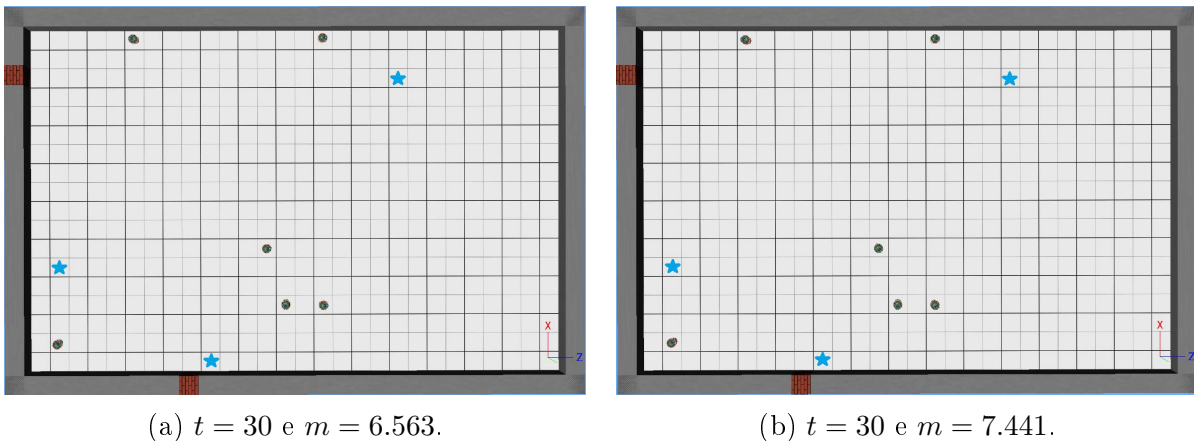


Figura 7.57: Experimentos relativos à tarefa de forrageamento para 6 robôs para contrastar a melhoria imposta ao empregar a correção da guinada.

### 7.6.6 Comparação das abordagens implementadas no Webots

Para resumir o tempo de simulação em relação ao custo benefício de implementação de cada uma das abordagens foi elaborado um gráfico que avalia o tempo ( $s$ ) de simulação na plataforma Webots EDU e o instante de tempo ( $t$ ) discreto do algoritmo. O gráfico de comparação está apresentado na Figura 7.59. A primeira abordagem implementada utiliza odometria e é a mais intuitiva e simples de ser implementada, no entanto, apresenta o maior tempo ( $s$ ) de simulação em relação às demais abordagens. Essa característica é em virtude do tempo do time retomar ao ângulo inicial em cada passo de tempo.



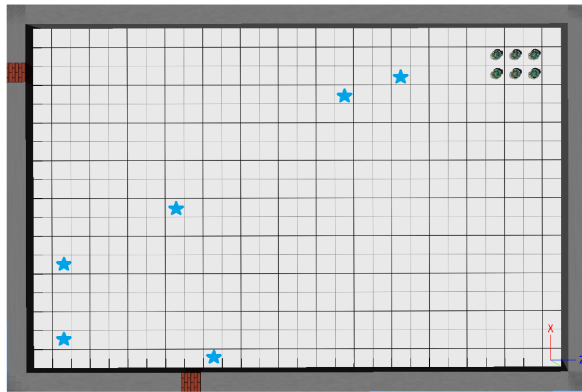
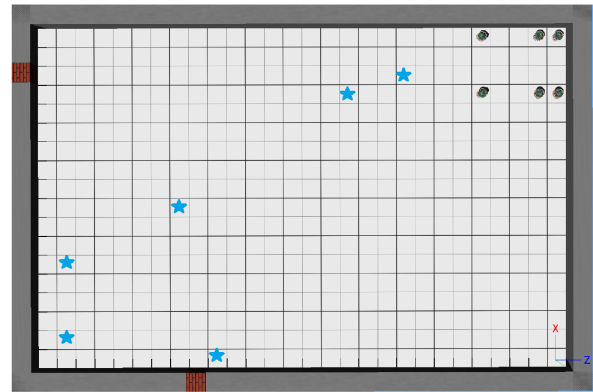
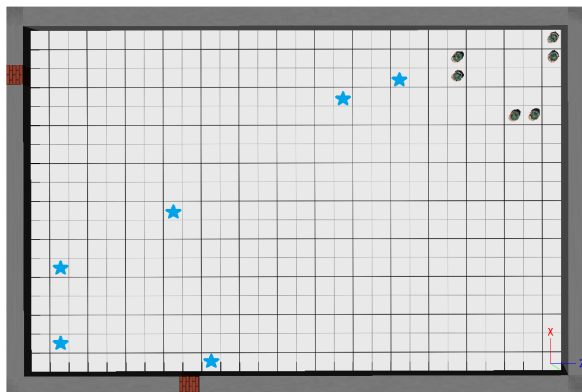
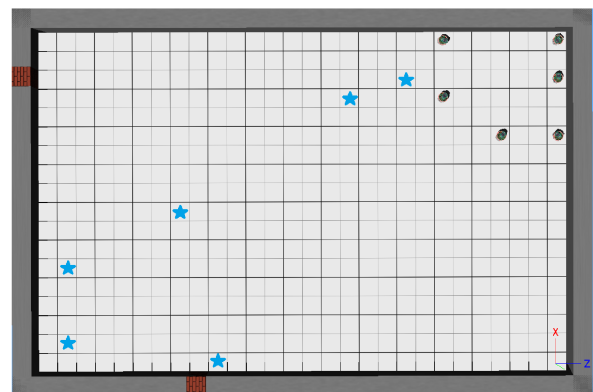
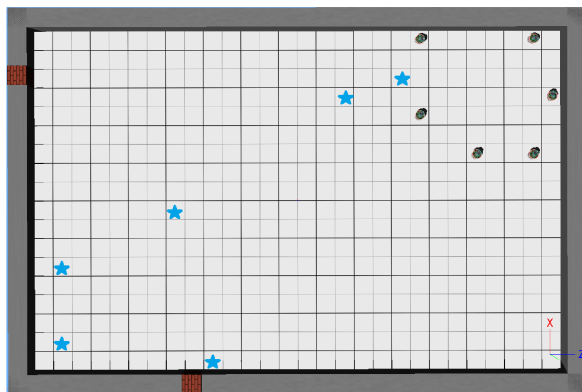
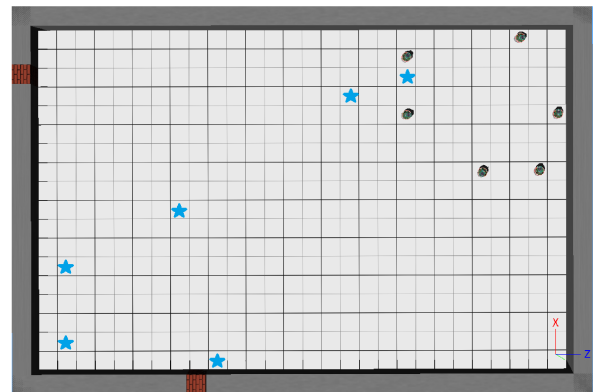
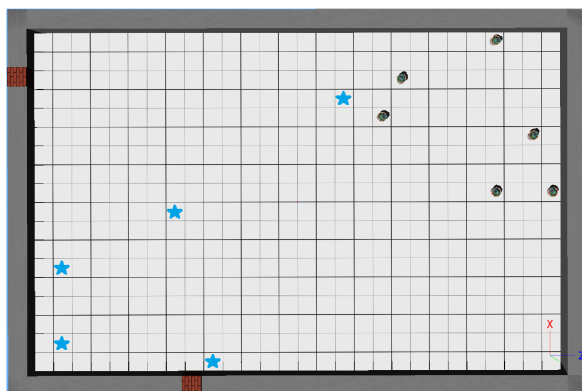
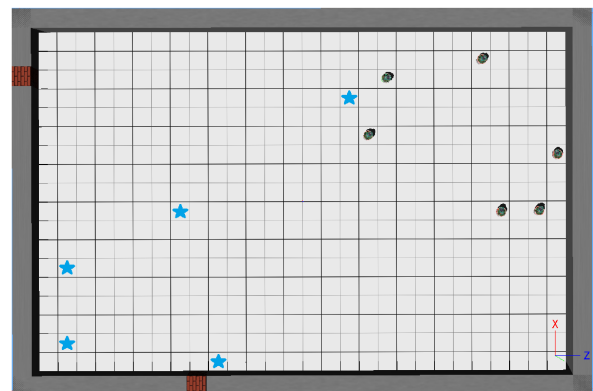
(a)  $t = 0$  e  $s = 0$ .(b)  $t = 1$  e  $s = 14.234$ .(c)  $t = 2$  e  $s = 26.992$ .(d)  $t = 3$  e  $s = 37.945$ .(e)  $t = 4$  e  $s = 53.575$ .(f)  $t = 5$  e  $s = 72.254$ .(g)  $t = 6$  e  $s = 88.236$ .(h)  $t = 7$  e  $s = 101.229$ .

Figura 7.58: Experimentos relativos à tarefa de forrageamento para 6 robôs utilizando-se rotação e GPS para correção do movimento.

A segunda abordagem implementada, refere-se ao modelo com localização do time de robôs através da odometria e o time não retoma ao ângulo inicial após cada movimento. Como é possível perceber no gráfico, essa abordagem é a que gasta menos tempo ( $s$ ) em cada passo de tempo ( $t$ ), devido ao fato de que a odometria é uma abordagem que é realizada em tempo real e é uma estratégia largamente utilizada em sistemas contínuos. Portanto, torna-se aliada à estratégia de armazenar as posições angulares de cada robô do time, a mais eficiente em termos de tempo de execução. No entanto, conforme observado nos experimentos anteriores, essa estratégia não centraliza os robôs em suas respectivas células.

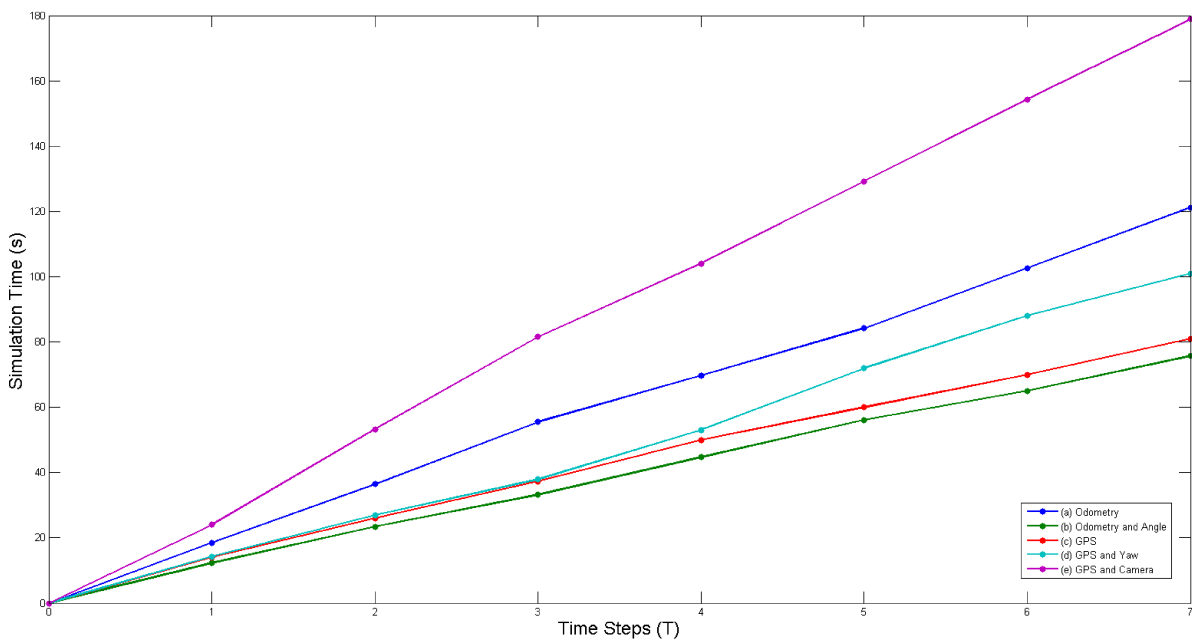


Figura 7.59: Gráfico de comparação do tempo de execução em segundos ( $s$ ) em relação ao instante de tempo discreto ( $t$ ) do algoritmo da tarefa de forrageamento para cada uma das abordagens implementadas no Webots.

A terceira estratégia implementada considera a utilização da localização do time de robôs através do *slot* de GPS, essa estratégia apresenta um desempenho de tempo ( $s$ ) de simulação muito próximo à odometria, no entanto, quando a tarefa envolve muito tempo de simulação, pode ser que o time colida com obstáculos ou outros robôs. O penúltimo experimento utiliza a estratégia de ajustar a localização do time de robôs através do GPS e do *slot* de inércia (controle de guinada) e apresenta um tempo ( $s$ ) intermediário de simulação para cada instante de tempo ( $t$ ). Nos instantes de tempo ( $t$ ) iniciais os tempos de execução ( $s$ ) do último experimento coincidem com os tempos ( $s$ ) terceiro experimento, uma vez que nenhum robô do time precisou fazer o ajuste de guinada.

À medida que o erro no ângulo de rotação passa a ser encontrado, esses ajustes provocam um atraso no tempo ( $s$ ) de execução no Webots. Apesar do tempo do penúltimo experimento ser relativamente alto se comparado às demais abordagens, esta abordagem

é a mais apropriada uma vez que o sistema deve ser inerentemente preciso em relação à localização de seus robôs dentro do ambiente de simulação. Finalmente, uma abordagem do modelo que utiliza câmera pode visualizada na linha superior do gráfico, este experimento apresenta o maior tempo de execução, uma vez que todos os robôs do sistema devem girar e processar o ambiente através de câmera com um algoritmo de processamento de imagens.

### 7.6.7 Melhorias da tarefa de forrageamento no Webots

As simulações a seguir que utilizam o simulador Webots são importantes para mostrar que o modelo PCAAM é capaz de ser implementado numa abordagem prática com o robô e-puck, denominada Robot Cellular Automata Ant Memory model (RPCAAM), publicadas em (Lima and Oliveira, 2016b). Para implementar o RPCAAM no simulador Webots algumas considerações iniciais foram feitas. Primeiro, o ambiente foi construído utilizando um mundo inteligente, em que cada alimento é um sensor de bola colorida 90°. Quando um robô passa em um sensor, ela é desligada. Esta grade de unidades de comida é controlada pela leitura e gravação de arquivos através do servidor. Para sincronizar a equipe, também foi utilizado um mecanismo de leitura gravar arquivos.

A simulação Webots versão 7.3 foi usado para contrastar o modelo RPCAAM e o modelo determinístico proposto em (Lima and Oliveira, 2017a) em termos de tempo de simulação  $s_T$ . Estes foram avaliados usando 5 simulações para cada modelo. O experimento foi realizado em um ambiente  $10 \times 15$  células usando  $N = 6$  robôs,  $F = 20$  unidades de alimentos, obstáculos virtuais (Marchese, 2011) e um solucionador de conflito perpendicular. Os outros valores de parâmetros do modelo foram os mesmos que aqueles utilizados na seção anterior. A disposição inicial do modelo é mostrado na Figura 7.60 (a). A configuração intermediária após  $t = 15$  passos é mostrados na Figura 7.60 (b).

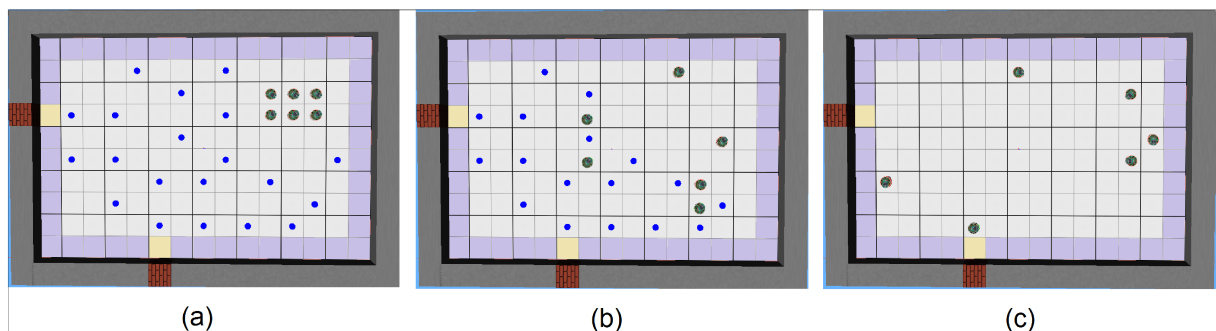


Figura 7.60: Instantâneos da simulação RPCAAM em Webots usando e-Pucks (comida é representado em azul): (a) inicial  $t = 0$ , (b)  $t = 15$  and (c) final  $t = 57$  momentos.

Além disso, a figura 7.60 (c) mostra a configuração no passo final. Os experimentos mostraram que as iterações modelo determinístico gastaram em média  $s_T = 764.6$  segundos ( $\sigma = 144.45$ ) para realizar uma média de  $T = 56.5$  iterações ( $\sigma = 11.52$ ). Por outro

lado, o modelo RPCAAM gasto em média  $s_T = 716.4$  segundos ( $\sigma = 29.14$ ) para realizar uma média de  $T = 53.4$  iterações ( $\sigma = 3.36$ ). Neste caso, o modelo RPCAAM tem um desempenho melhor do que o RCAAM que usa um homing determinista. Esta observação confirma o que foi observado em simulações C. Ou seja, RPCAAM também é capaz de escolher caminhos alternativos durante o processo de homing e também evita o comportamento inercial Yamamoto et al. (2007). Por sua vez, o modelo determinístico passou mais tempo resolvendo conflitos perto dos ninhos, especialmente para resolver o cruzamento de conflitos perpendiculares, porque um robô tem que esperar para o outro para completar o seu movimento, e esta resolução é lenta, causando um atraso na simulação.

Para implementar o Robot Density Cellular Automata Ant Memory, ou RDCAAM, no simulador Webots temos que considerar RDCAAM com  $c_{max} = 1000$  e o modelo de pesquisa baseia-se na Pesquisa Tabu usando  $QP_2$ , que representou a melhor avaliação.

A simulação Webots foi usada para contrastar os modelos probabilísticos elitistas RCAAM e o modelo RDCAAM em termos de tempo de simulação  $s_T$ . Estes modelos foram avaliados usando 5 simulações para cada abordagem. O experimento foi realizado em um ambiente com  $10 \times 15$  células:  $N = 6$  robôs e-Puck (círculos verdes),  $F = 20$  unidades de alimentos (círculos), obstáculos virtuais e um solucionador de conflitos perpendiculares.

Os outros valores de parâmetros do modelo foram os mesmos que aqueles utilizados na seção anterior. A disposição inicial da simulação é mostrado na figura 7.60 (a) e (b) são os mesmos para ambos os modelos. Além disso, alguns screenshots intermediários e finais são mostrados para os dois modelos na Figura 7.60. Os cinco experimentos executados para cada modelo mostrou que o modelo elitista RPCAAM (Lima and Oliveira, 2016b) gasta em média  $s_T = 1413.4$  segundos ( $\sigma = 282.84$ ) para realizar uma média de  $T = 74.0$  iterações ( $\sigma = 22.88$ ). Por outro lado, o modelo RDCAAM (Lima and Oliveira, 2016a) gasta em média  $s_T = 1181.8$  seconds ( $\sigma = 85.50$ ) para realizar uma média de  $T = 56.2$  iterações ( $\sigma = 5.11$ ). Neste caso, o modelo RDCAAM tem um desempenho melhor do que o modelo probabilístico elitista. Além disso, o tempo gasto para robôs completarem a tarefa foi calculado.

Uma análise de teoria de filas compara as abordagens elitistas RPCAAM e a RDCAAM da Figura 7.61. A Figura 7.62 mostra que o modelo HCAAQ passou quase  $\approx 20$  minutos (em média 22 segundos para cada passo concluir a tarefa ( $T = 55$  que representa o tempo de observação), utilizando o ninho  $A_0 = 9$  vezes (45%) e  $A_1 = 11$  vezes (55%). Cada robô passou em média  $s_l = 5.4$  iterações segurando comida, contando a partir do momento em que ele pega a comida até que deposite no ninho (média de 104,97 segundos). O rendimento do sistema RDCAAM é  $\mu = 0.36$ , e para cada ninho é  $\mu_0 = 0.16$  e  $\mu_1 = 0.2$ . O tempo ninho média de chegada é  $\lambda_0 = 0.16$  e  $\lambda_1 = 0.2$ . O tempo de ocupação do sistema (ninhos) é  $B = 20$  passos. Para cada ninho, o tempo de ocupação é  $B_0 = 9$  e  $B_1 = 11$  iterações e o tempo de utilização é  $U_0 = 0.16$  and  $U_1 = 0.2$  do total de iterações. O tempo médio de serviço para cada ninho é  $S_0 = 1$  e  $S_1 = 1$  passo, isso significa que cada ninho

só é ocupada durante o estado de depósito  $t = 1$ .

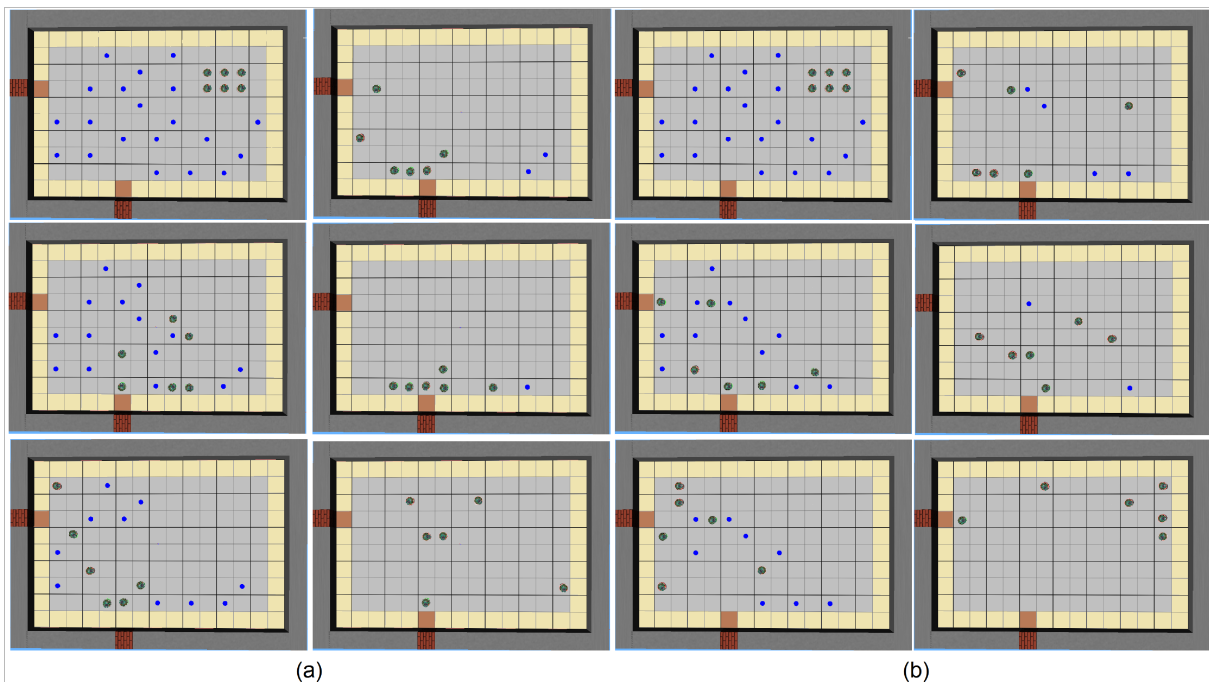


Figura 7.61: Snapshots Webots (a) desempenho elitista na  $t = \{0,6,15,35,59,109\}$ , and (b) performance do modelo RHCAAQ em  $t = \{0,7,15,30,40,55\}$ .

Por outro lado, a simulação do modelo probabilístico elitista RPCAAM gastou pouco mais de  $\approx 30$  minutos (em média, 17.4 segundos para cada passo concluir a tarefa (durante time de observação  $T = 109$ ), usando o ninho  $A_0 = 6$  vezes (30%) e  $A_1 = 14$  vezes (70%), o que significa que o ninho  $A_0$  foi subutilizado. Cada robô passou em média  $s_l = 18.3$  iterações segurando a comida (318,14 segundos em média). O rendimento do sistema probabilístico elitista é  $\mu = 0.18$ , e para cada ninho é  $\mu_0 = 0.05$  e  $\mu_1 = 0.12$ . O tempo médio de chegada é  $\lambda_0 = 0.05$  e  $\lambda_1 = 0.12$ . O tempo de ocupação do sistema é  $B = 95$  passos. O tempo ocupado para ninho  $A_0$  é  $B_0 = 6$  e para  $A_1$  é  $B_1 = 89$  iterações e o tempo de utilização é  $U_0 = 0.05$  e  $U_1 = 0.81$  do total de iterações. O comprimento médio de serviço para cada ninho é  $S_0 = 1$  passos e  $S_1 = 6.35$  passos, isso significa que cada ninho é ocupado por alguns robôs que estão bloqueados no ninho devido à formação de linha, perturbando outros robôs que gostariam de depositar suas unidades de alimentos em um daqueles ninhos ocupados.

Ambas as abordagens têm um bom uso dos recursos, isso significa que cada robô está contribuindo quase igualmente para a tarefa, portanto, nenhum robô está ocioso. Esta observação confirma o que foi observado nas simulações C: devido ao modelo RDCAAM ser capaz de escolher caminhos alternativos durante o processo de homing e também a evitar o comportamento inercial. Por sua vez, o modelo de elite que é muito semelhante ao proposto na determinista RCAAM, e por isso não é possível distribuir os robôs entre os ninhos, já que esse apenas evita o efeito de inércia do sistema, por outro lado degrada o desempenho enxame por causa da longa ocupação de um dos ninhos causada pela formação

de filas e o efeito de obstrução.

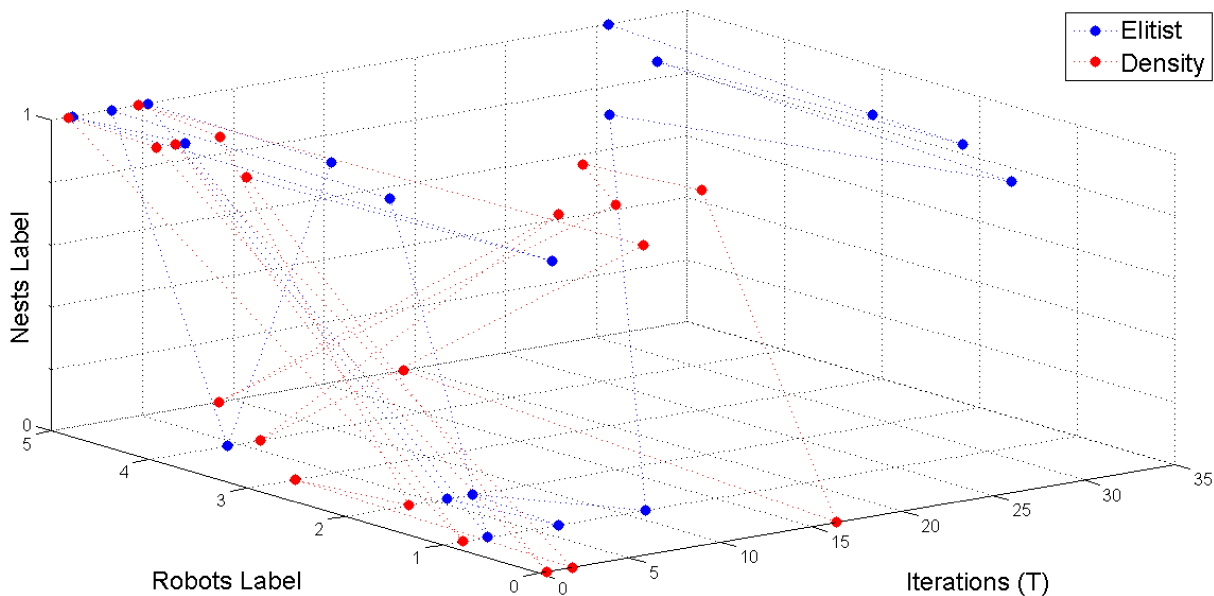


Figura 7.62: Iterações  $T$ ,  $N = 6$  robôs e uso dos ninhos no Webots. (a) O azul representa a abordagem elitista. (b) O vermelho representa a abordagem RDCAAM.

## 7.7 Discussões finais sobre os resultados

Neste capítulo foram apresentados os principais resultados referentes aos experimentos realizados para cada um dos modelos das três tarefas investigadas neste trabalho. O modelo CAAM, proposto neste trabalho, mostrou-se bastante promissor para a tarefa de forrageamento e através dos experimentos de simulação em C, foi possível ajustar diversos parâmetros do modelo, além de compreender melhor a contribuição de cada componente do modelo e sua interação com o ambiente e cenário de forrageamento. As melhorias mais promissoras do modelo CAAM foram apresentadas nos modelos CAAQ, PCAAM e DCAAM. Nesses modelos uma melhor distribuição dos robôs entre os ninhos foi observada e também o efeito de congestionamento foi reduzido significativamente a partir da inserção do elemento probabilista na escolha da célula. A partir destas contribuições outros dois modelos foram desenvolvidos, o SCAAM e o FSCAAM e também foram obtidos bons resultados para as tarefas de coleta seletiva e busca e resgate. A partir daí, o modelo RCAAM, que simula a tarefa de forrageamento, se mostrou eficiente na tarefa de forrageamento quando implementada no ambiente de simulação Webots, especialmente tiver ajuste na posição e no ângulo de guinada a partir das informações fornecidas pelo GPS e pelo *slot* de inércia. Além disso, as contribuições dos modelos PCAAM e DCAAM serviram de inspiração para a criação do modelo RPCAAM e RDCAAM, que também foram implementados no ambiente de simulação Webots. Essas simulações apresentaram resultados bastante promissores, assim como os resultados das simulações dos modelos iniciais.

Nas simulações realizadas com os modelos, os robôs foram distribuídos aleatoriamente ao longo da ambiente, todos eles a partir do estado de searching. A partir do momento que já não há qualquer unidade de comida sobre o ambiente, o processo é finalizado, exceto na simulação com o Webots, onde os robôs são colocados em posições fixas sobre o ambiente. O ambiente atualizado com feromônio a cada passo de tempo e as informações impressas a partir de outros robôs ficam disponíveis como dados globalmente compartilhados. As principais conclusões sobre os resultados dos modelos nas 3 tarefas investigadas são: (a) a habilidade visão do robô na busca e a informação dinâmica em homing acelerar a eficiência da equipe para completar a tarefa; no entanto, o modelo pode ser aplicado mesmo se essas habilidades não são disponíveis; (b) a tarefa de homing é melhorada no uso de um cálculo de densidades perto dos ninhos e a partir da utilização da regra de movimentação probabilista elitista, sendo que a primeira provê melhor esparsamento do time pelos ninhos e a segunda provê a ausência do efeito de inércia provocado pelo time; (c) um ajuste adequado dos parâmetros envolvidos na difusão e declínio do feromônio adequado é importante ter o melhor desempenho dos robôs do exame; (d) a memória de curto prazo na busca também acelera a eficiência da equipe, sendo que esta melhoria é ainda mais perceptível quando as constantes feromônio não foram bem ajustados; no entanto, uma utilização excessiva dos recursos da mesma demanda um tempo de processamento excessivo, tornando tamanhos intermediários da memória mais adequados (neste trabalho, foi utilizado  $|Q| = 10$ ); (f) um número excessivo de robôs na equipa provoca uma degradação no desempenho da equipe, dependendo dos objetos a serem recolhidos e sobre as limitações coleção pontuais (número e a largura dos ninhos). Adicionalmente, (g) o uso do piso dinâmica no homing, o algoritmo impede a formação de linhas (Alizadeh, 2011), favorecendo uma curvatura mais homogênea perto dos ninhos; evita a formação de congestionamento e comportamento inercial, e impede a obstrução área (Li-jun et al., 2009); (h) com base no efeito repulsivo de feromônio e a utilização da memória para evitar regiões recentemente visitado, o modelo retorna uma boa cobertura do ambiente na busca, (i) algumas observações relacionadas com as características do ambiente são: a distribuição concentrada de alimentos sobre o ambiente pode ser pior do que uma disposição muito longe dos ninhos; vários pontos de coleta devem ser distribuídos ao longo do ambiente para dar uma boa vantagem deste recurso; ninhos finas são inadequadas, mas o aumento da largura ninho também podem degradar o desempenho da equipe, devido ao tempo de processamento elevada; (j) uma especificação adequada do tamanho da memória é relevante para a cobertura: memórias pequenos ou inexistentes levar a um número excessivo de iterações e um monte de regiões sobrecarregados, enquanto memórias de grandes dimensões elevar o tempo de processamento de cada iteração e levar a regiões sem cobertura desde o seu acesso pode ser obstruído; (k) a utilização da modelagem do fogo cria obstáculos virtuais ao longo da simulação e gera um atraso no tempo de conclusão da tarefa dependendo de onde a inicialização do foco é colocada; (l) a utilização de um time

heterogêneo e especializado em coletar objetos de um determinado tipo gera um atraso no sistema, uma vez que os robôs que terminaram suas tarefas tendem a ficar ociosos, atrapalhando a convergência do sistema, (m) a implementação do ambiente de simulação Webots para a arquitetura de robô e-puck demandou a adoção de algumas restrições, tais como a resolução de conflitos através de obstáculos virtuais e a resolução de conflitos do tipo "X". No entanto, a implementação mais realística dos modelos apresenta os mesmos resultados de convergência da modelagem matemática.



---

## Conclusão e trabalhos futuros

---

O mercado da robótica cresce anualmente em virtude das inúmeras possibilidades aplicação em diversos setores (Russell and Norvig, 2005). Robôs são responsáveis por diversas atividades industriais, tais como, produção de biotecnologia e produtos farmacocômicos, exploração aeroespacial, soldagem e pintura, transporte terrestre e aéreo, manipulação de componentes eletroeletrônicos e até mesmo na telemedicina, realizando essas tarefas em um menor espaço de tempo com alta acurácia, repetitividade e eficiência muito além das capacidades humanas (Russell and Norvig, 2005). A maioria dos robôs industriais é composta de apenas um braço robótico que tem uma quantidade de graus de liberdade para se movimentar em seu volume de trabalho (Russell and Norvig, 2005). Para compensar a falta de mobilidade a robótica móvel vem sendo largamente utilizada (Lerman et al., 2005; Winfield, 2009). Para a coordenação de apenas um robô, a tarefa é mais simples, no entanto, quando passa a trabalhar com um time, que realiza tarefas em paralelo, problemas como, comunicação, controle de obstáculos e entre robôs passam a ser necessários (Winfield, 2009). Várias estratégias que abordam o controle tradicional e técnicas clássicas de inteligência artificial já foram investigadas por diversos cientistas (Calvo et al., 2012; Ducatelle et al., 2010; Winfield, 2009). Nesta tese de doutorado a proposta foi de estudar outras abordagens híbridas e bio-inspiradas promissoras para o controle de um time de robôs, incluindo-se modelagens baseadas em enxames e autômatos celulares.

Em resumo, com a realização desta tese de doutorado foi apresentada uma revisão da literatura de maneira detalhada buscando encontrar os principais modelos que norteiam a temática de modelagem de dinâmica inteligente de enxames, tanto para a modelagem de dinâmica populacional quanto para a robótica. Primeiramente foi apresentada uma

revisão sobre os modelos baseados em autômatos celulares desenvolvidos para a simulação da dinâmica coletiva de pedestres (Varas et al., 2007), (Alizadeh, 2011), (Wei-Guo et al., 2006). Posteriormente, foram apresentados conceitos sobre sistemas de busca e otimização, incluindo-se a busca Tabu e a otimização por colônias de formigas. Além disso, foi realizada uma revisão sobre robótica coletiva (Lerman et al., 2005), (Winfield, 2009), (Calvo et al., 2012), (Ferreira et al., 2014), um campo que tem atraído a atenção de diversos pesquisadores. A partir dessas investigações foi traçado um paralelo entre algumas tarefas investigadas para sistemas multi-robôs e os trabalhos que abordam evacuação de pedestres, e assim foi possível construir um modelo para a navegação de time de robôs. Posteriormente, o modelo foi refinado no ambiente de simulação Webots, a partir das adaptações que foram necessárias para a resolução de alguns problemas pertinentes às abstrações do mundo real, mostrando que a nova proposta é aplicável ao contexto de robótica. Além disso, uma análise do problema sob o ponto de vista teórico foi realizada para mostrar que a solução adotada para a navegação do time de robôs é de fato eficiente e se a partir de qualquer configuração inicial do ambiente é possível que a tarefa seja cumprida.

Nosso modelo é dedicado à tarefa forrageamento, mas também pôde ser adaptado às tarefas de coleta seletiva e busca e resgate em ambientes com fogo. Nessas tarefas existem duas fases principais - a busca e o regresso, sendo que, no primeiro processo, o robô precisa encontrar um objeto em um local desconhecido e no segundo, já recolheu o objeto e deve depositá-lo num ponto de depósito denominado ninho - sendo que essas tarefas são muito relevantes para robótica coletiva (Winfield, 2009), já que a fase de busca e regresso (i) é uma metáfora para uma ampla classe de problemas de integração de exploração, navegação e transporte de objetos; (ii) a busca de multi-objetiva é um problema canônico para o estudo da cooperação robô-robô em sistemas multi-robô, (iii) muitas aplicações do mundo real são exploradas em robótica, tais como, limpeza, colheita, busca e salvamento, limpeza de minas terrestres e exploração planetária, são exemplos em que robôs em algum estágio estão envolvidos com estados de busca e regresso. A busca em nosso trabalho é guiada pelo feromônio (Calvo et al., 2012), (Lima et al., 2016a) - distribuído pelos robôs sobre o ambiente, enquanto percorrem esse ambiente a cada passo de tempo - e de uma memória individual de curto prazo com base no algoritmo de busca Tabu (Glover, 1989), (Glover, 1990). Assim, a otimização da busca é guiada pelo algoritmo global baseada em colônias de formigas e a pela estratégia local denominada busca Tabu.

Outros trabalhos na área da robótica também usam a metáfora da propagação formigas feromônio para controlar a trajetória de um robô enxame (Ioannidis et al., 2011b), (Hecker and Moses, 2015), (Ioannidis et al., 2015). No entanto, o emprego habitual da feromônio é semelhante ao utilizado pelas formigas na natureza: criando um efeito de atração que faz com que os robôs se comportem como seguidores atraídos pelo feromônio. Na modelagem do processo de busca nos modelos aqui investigados, a maior inspiração foi

o modelo precursor proposto em (Calvo et al., 2012), (Lima et al., 2016a) que usa a idéia da propagação de um feromônio invertido aplicadas nas tarefas de vigilância, que tem em comum com o forrageamento a necessidade de uma boa cobertura do ambiente. Esta cobertura é obtida por meio do efeito de repulsão entre os robôs, o que leva a uma difusão natural do aglomerado. No entanto, no presente trabalho, esta dinâmica foi reforçada pelo uso de uma memória de curto prazo, que também impede o regresso antecipado de robôs para células visitadas recentemente, forçando os mesmos a explorarem novas áreas. Outra diferença entre o presente trabalho e o modelo em (Calvo et al., 2012) é que a modelagem aqui foi completamente discreta, tanto para o depósito de feromônio como para a etapa de movimento, permitindo um controle local dos conflitos por regras de transição, enquanto no trabalho anterior do autores mencionam o uso de um algoritmo de desvio de obstáculos para evitar colisões com o meio ambiente e outros robôs. O estado de *grabbing* depende do raio de visão do robô, pois ele se inicia na identificação do objeto na redondeza do robô e finaliza após o robô se deslocar com prioridade até a célula que contém o objeto para coletá-lo.

Por outro lado, o *homing* foi inspirado por modelos baseados em CA anteriores da dinâmica multidão durante o processo de evacuação (Varas et al., 2007), (Alizadeh, 2011) e (Schadschneider et al., 2011). Neste estado, cada robô escolhe uma rota ótima para seguir, ao considerar um campo de piso estático e dinâmico para evitar a formação de filas para os ninhos. Como o modelo de *homing* também é de primeira escolha, então o mesmo tenta deslocar-se para outra célula de menor valor que não seja também alvo de outro robô. Entretanto, se a célula já estiver ocupada por outro robô deve aguardar para que a célula seja desocupada na próxima iteração de tempo. Além das características anteriores, comuns aos três modelos, algumas particularidades das respectivas tarefas foram inseridas em cada modelo. É importante notar que a meta nesses trabalhos era reproduzir o comportamento humano observado em situações específicas, não sendo objetivo desses estudos a proposição de maneiras de mudar esses comportamentos, porque é basicamente instintiva. O objetivo desses modelos é apresentar uma simulação o mais próximo possível do fenômeno observado, em seguida, procurar compreender os fatores ambientais e sociais que podem afetar cada um desses comportamentos, a fim de propor melhorias no ambiente que poderiam facilitar a evacuação em situações de emergência. No presente trabalho, estes modelos anteriores do comportamento humano foram adaptados para fornecer um comportamento eficiente equipe do robô durante a pesquisa para o ponto de coleta mais próximo. Tanto quanto sabemos, nenhum trabalho anterior traçou este paralelo entre a evacuação pedestres e a *homing* no forrageamento de robôs, a fim de usar a dinâmica de multidão para orientar o comportamento desejado do enxame de robôs executando uma tarefa específica em sistemas de enxames de robôs. Para obter o efeito desejado, a similaridade entre os dois processos foi adaptada de modo a obter uma equipe de robôs eficiente nesta tarefa, mesmo porque, no caso de robôs é possível obter

uma arquitetura de controle eficiente.

O estado de *depositing* compreende um único passo de tempo, que depende da disponibilidade do ninho estar livre. A partir dessa disponibilidade o robô deposita o objeto e volta para o estado de busca por outros objetos. O ninho é considerado um ponto de gargalo no sistema e por isso foi dedicada uma análise especial para a investigação de formação de filas de robôs perto desse local. Justamente, por haver poucos recursos destinados à coleta de objetos e a formação de filas, que o time atrasa a finalização da tarefa. Além disso, o efeito de fricção (efeito de ir e vir) perto desses ninhos também gera um atraso na conclusão da tarefa. A capacidade do sistema, bem como a disciplina de atendimento desse sistema foram avaliadas. Adicionalmente, foi discutido que a estratégia usada no *homing* afeta diretamente a formação de filas perto dos ninhos.

De modo geral, existem alguns passos que são comuns às três tarefas analisadas, as principais características dos nossos modelos são: (i) o meio ambiente é modelado como uma estrutura composta por células quadradas idênticas de mesmo tamanho formada por autômatos celulares - esse mapa é a base à qual ocorrerá toda a simulação da tarefa de busca é repassada para cada robô do enxame; (ii) cada robô é controlado por uma máquina de estados finitos individual que muda ao longo de um ciclo de 4 estados principais: em busca  $\rightarrow$  *grabbing*  $\rightarrow$  *homing*  $\rightarrow$  *depositing*  $\rightarrow$  *searching*; (iii) cada movimento do robô corresponde em alterar a sua posição atual para uma célula adjacente e é decidido por uma regra local, que leva em conta a vizinhança do robô - que deixa um traço no ambiente por onde percorre e também na vizinhança de Moore ao qual a célula central é adjacente; (iv) em nosso modelo principal CAAM o próximo passo de decisão na busca é determinista e com base no nível de feromônio das células vizinhas - a informação de outros robôs impressa no ambiente é uma informação que deve ser compartilhada globalmente - enquanto que nos modelos SCAAM e FSCAAM é baseado em uma transição estocástica, (v) em todos os modelos investigados, exceto em CAA, a mudança de estado no *searching* é baseada também em um memória de curto prazo, que evita o retorno para locais visitados recentemente; (vi) cada vez que um robô passa na célula do reticulado durante a busca, deixa um traço no ambiente, sendo que o feromônio repulsivo é depositado na célula corrente, mas também nas adjacentes (num valor atenuado); (vii) feromônio da célula é submetida a um processo de evaporação com o objetivo de permitir que as células visitadas só podem ser percorridas novamente depois de algum período de tempo, exceto no modelo CAA, onde a fila não foi investigada; (viii) a capacidade de visão do robô é considerada de tal maneira que é capaz de identificar uma unidade de alimento próximo do seu campo de visão, e a mudança para o estado de captura é realizada a partir de passos diretos para coletar o objeto; (ix) o próximo passo de decisão no retorno (*homing*) é determinista com base na composição de duas informações: a distância do piso estático para o ninho mais próximo e do piso dinâmico que conta com a ocorrência de outros robôs na forma da sua posição atual e o ninho; (x) nos modelos PCAAM, HCAAM, CAAQ, DCAAM, SCAAM,

FSCAAM a escolha é probabilista e não apresenta a presença do campo de piso dinâmico para controlar os robôs, sendo que essa escolha levou à ausência do efeito de inércia próximo aos ninhos (Yuan and Tan, 2007); (xi) no modelo SCAAM e FSCAAM tanto o *homing* quanto o *searching* são guiados por processos estocásticos; (xii) no modelo SCAAM - resultados de análise visual e estatística foram implementados e esses resultados demonstraram que o modelo é promissor - apresenta uma modelagem dos agentes de maneira heterogênea (Schultz et al., 2007); (xiii) no modelo FSCAAM é apresentada uma modelagem do fogo para dar mais realismo à simulação (Zheng et al., 2011) e apresentou o efeito de fogo na simulação - diferentes formas de função de ativação foram implementadas para a propagação do incêndio; (xiv) o estado de depósito é quando o robô deixa o objeto no ninho e aqui corresponde apenas a um passo de tempo, em que o robô deve permanecer na célula assumido como sendo o ninho; no entanto, pode ser particularizada à arquitetura robótica dos modelos RCAAM, RPCAAM, RDCAAQ sem prejuízo do modelo matemático CAAM; (xv) as decisões determinísticas tanto no estado *searching* quanto no estado *homing* do modelo CAAM podem ser superados quando situações de conflito são identificadas (existência de outros robôs que se deslocam no mesmo sentido ou a existência de mais de uma célula com o menor valor da métrica em questão); em tais situações, as decisões aleatórias são feitas imprimindo uma característica estocástica com o modelo resultante, embora a maioria das decisões são tomadas na ausência de conflitos e são puramente determinísticas.

O processo de navegação do time como um todo se inicia com os robôs posicionados arbitrariamente no reticulado, sendo que todos iniciam no estado *searching*. Esse processo global se encerra quando não existe mais nenhum objeto o processo é finalizado. Com a execução do modelo descrito foi possível gerar dados visuais, onde os comportamentos de padrões observados em colônias de formigas e até mesmo observado em dinâmicas de pedestres. Além disso, dados estatísticos foram obtidos utilizando-se as linguagens de programação C padrão e C++ relacionando a média das iterações com a quantidade de robôs alocados no ambiente de simulação. Além disso, uma análise das características do modelo desenvolvido foi abordada através do programa elaborado que poderá ser utilizado em estudos posteriores, tais como, verificar se a densidade de robôs poderá afetar significativamente o tempo de iterações  $T$  para completar a tarefa de forrageamento.

O modelo RCAAM referente à tarefa de forrageamento foi implementado no ambiente de simulação Webots para a arquitetura de robôs e-Puck. Para atingir uma simulação completa da tarefa de forrageamento, algumas adaptações em relação ao modelo central e base da tarefa de forrageamento tiveram que ser considerados. As melhorias do modelo CAAM também foram incorporadas nesse trabalho, tais como a inserção do elemento probabilista na escolha da decisão da célula, bem como o cálculo de densidade. A primeira adaptação foi em relação à forma de comunicação entre os robôs. Neste trabalho a solução adotada baseia-se no compartilhamento de arquivos de texto entre os robôs, uma vez que

cada um deles recebe um código para controle. Posteriormente, algumas adaptações no controle de conflitos foi realizada para evitar que os robô colidissem ao realizarem cruzamentos durante um determinado instante. A partir dessas adaptações, o algoritmo CAAM foi implementado no Webots e foi realizada uma comparação entre as abordagens de localização por odometria e GPS, sendo que a última apresentou maior precisão.

As simulações foram realizadas no modelo usando diferentes cenários de forrageamento para um bom ajuste de parâmetros e uma melhor compreensão do comportamento do modelo. As principais conclusões dos experimentos foi resumida no último capítulo. A implementação do modelo proposto possibilitou a geração de dados visuais dos padrões de comportamento observados em colônias de formigas e até mesmo observadas na dinâmica de pedestres. Além disso, os dados estatísticos relativos foram obtidos da média de iterações com a quantidade de alimentos e robôs atribuídos no ambiente de simulação. Os resultados foram avaliados de maneira qualitativa e quantitativa. Para a análise qualitativa, foi observado a maneira como os algoritmos de enxames se comportam em relação aos demais algoritmos propostos, sendo que estes testes foram avaliados tanto em software quanto em um ambiente de simulação projetado para diversas plataformas de robôs reais e também através da modelagem do fogo. Nos testes quantitativos, através de gráficos e planilhas esses resultados foram contrastados com os modelos presentes na literatura correlacionada. Ou seja, foram avaliados o tempo total de execução e a melhor localização que a meta (recurso) deve estar localizado no ambiente estrutural e o tipo de arqueamento que os agentes fazem ao redor da meta. Assim, foi elaborado um conjunto de novas estratégias que permita que os agentes tomem melhores decisões de escolha em sua vizinhança a partir de regras locais e baseados em algoritmos de dinâmica coletiva aumentem a performance de algoritmos para dinâmica de robôs de busca móveis e cooperativo.

Por fim, uma análise teórica do modelo de depósito de feromônio invertido utilizado no estado *searching* de todas da tarefa de forrageamento para o modelo CAAM foi realizada através da transformação desse modelo em um grafo. O principal objetivo desse estudo foi investigar o potencial de cobertura e convergência desse algoritmo base para a aplicação nas tarefas de robótica estudadas. De acordo com os resultados apresentados, o algoritmo apresentado neste trabalho converge para uma solução desde que os parâmetros sejam devidamente ajustados.

## 8.1 Comentários sobre a relevância

A novidade dos modelos presentes neste trabalho refere-se à utilização de autômatos celulares, dinâmica de pedestres, Busca Tabu, ACO usados de maneira agrupada como um modelo para a execução de várias tarefas robóticas em um modelo híbrido. Uma das principais fases de forrageamento tarefa é o *homing*. Aqui o modelo *homing* foi inspirado

por modelos de evacuação de pedestres previamente investigados na literatura. Embora alguns trabalhos utilizam autômatos celulares e evacuação multidão como para a evacuação com a ajuda de um robô guiado (Boukas et al., 2015), o autor não apresentou um enxame de robôs. Tanto quanto sabemos, nenhum trabalho anterior traçou um paralelo entre a evacuação pedestre e a *homing* no forrageamento robôs, a fim de usar o comportamento natural observada na dinâmica multidão para orientar o comportamento desejado do enxame robô. Outra importante etapa da tarefa de forrageamento é a busca. Aqui o estado de busca do modelo usa a idéia de feromônio repulsivo para espalhar o enxame de robôs. O emprego de feromônio repulsivo foi previamente investigado na literatura de uma forma diferente e mais simples na tarefa robótica denominada vigilância (Calvo et al., 2012), ou clustering (Moere and Clayden, 2005). Tanto quanto sabemos, nenhum trabalho anterior usou feromônio repulsivo no forrageamento robôs agregado à modelagem com ACs. Além disso, a memória de curto prazo com base em Busca Tabu também foi empregada no processo para evitar células visitadas recentemente pesquisando, diferentemente do modelo de (Alonso-Sanz, 2009) que utiliza somente os dois últimos estados e a aplicação não consiste numa tarefa robótica. Esta melhoria retorna uma melhor dispersão dos robôs no ambiente. A combinação de todos esses algoritmos de forma integrada: autômatos celulares, busca Tabu, modelagem de evacuação de pedestres inteligente, formigas coletivas utilizando feromônio invertido tornando o modelo proposto uma abordagem híbrida para cenários de cooperação de robôs. Também não foi encontrado em nenhum trabalho de robótica uma abordagem que realiza não só uma análise qualitativa visual e estatística do modelo, mas também uma análise do ponto de vista de comportamento social do conjunto de robôs - efeito de inércia, efeito de formação de filas, efeito de fogo, efeito de heterogeneidade do time, efeito de fricção, efeito de repulsão, efeito de rebanho, efeito de congestionamento, efeito de obstrução, efeito de aglomeração e efeito de arqueamento - fazendo uma analogia aos modelos de evacuação de pedestres. Nesta tese foi investigada a tarefa de forrageamento no contexto robótica e de outras abordagens, tais como, busca e resgate e coleta seletiva. Sabe-se que muitos sistemas artificiais, incluindo abordagem robótica está interessado em imitar comportamentos que são bem sucedidos na natureza para resolver tarefas específicas de forma cooperativa, tornando-se este novo modelo uma abordagem interessante que combina não somente um trabalho de êxito do comportamento natural como formigas, mas muitas abordagens coletivas realistas, modelagem de pedestres e robótica de enxames. Este composto estratégias sociais se encaixa em robótica para resolver tarefas de forrageamento, utilizando multi-agentes que interagem cooperativamente através de interações locais comportamentos globais para tarefas realização emergentes. Nenhum outro trabalho da literatura integrou a modelagem de dois sistemas naturais, tais como modelagem de fogos e pedestres modelados com ACs. Embora o trabalho de (Zheng et al., 2012) tenha iniciado uma análise similar em dinâmica de pedestres, o mesmo não modelou o fogo o a dinamicidade proporcionada pelos

ACs e que foi investigada nesta tese. Além disso, este trabalho serve não somente para a aplicação em robótica, mas em qualquer abordagem que inclua tarefas de sistemas multi-agentes, tais como em jogos ou em otimização de busca multi-objetiva, conforme apresentado no capítulo teórico desta tese. Adicionalmente, este trabalho também pode ser incorporado à modelagem pura de dinâmica de pedestres já que aqui foram apresentadas diversas análises sob este ponto de vista, bem como a melhoria e contorno de situações conflitos que ainda não haviam sido explorados em trabalhos precursores, como o controle de “x”. Adicionalmente, o AC com memória é uma generalização muito mais promissora que o trabalho de (Alonso-Sanz, 2009) onde àquele considera apenas os últimos estados recentemente visitados, já neste trabalho foi possível modelar o sistema de uma maneira muito mais genérica, e com as devidas adaptações o presente trabalho também poderá ser aplicado em outras tarefas, tais como clusterização ou modelagem de outros sistemas naturais.

## 8.2 Respostas às questões de pesquisa

As respostas em relação às questões de pesquisa estão abaixo relacionadas, seguindo a ordem que foram apresentadas na introdução desta tese, e foram obtidas observando os resultados experimentais e a análise formal do modelo proposto:

1. Foi possível mostrar que o primeiro modelo CAAM apresentado e todos os posteriores que foram propostos a partir dele surgiram de modelos precursores baseados em dinâmica de pedestres em situação de risco ou emergência. Adicionalmente, o modelo aqui proposto auxilia na modelagem de simulação de pedestres, pois uma particularidade de cruzamento perpendicular não havia sido contemplada em tais modelos.
2. O algoritmo que representa a busca multi-objetiva dos agentes, inicialmente proposto no modelo para forrageamento CAAM, foi modelado a partir dos sistemas artificiais de colônias de formigas. Esse modelo se mostrou bastante efetivo no que se diz respeito à otimização global da busca realizada pelos agentes, especialmente, quando o mesmo foi combinado com o modelo de pesquisa local denominado busca Tabu.
3. A partir dos diversos modelos computacionais que envolvem técnicas de busca baseados em meta-heurísticas de computação bio-inspiradas, englobando técnicas globais para a otimização da busca com algoritmos baseados em colônias de formigas, bem como a utilização de técnicas de inteligência artificial, dentre elas a busca local gulosa probabilista para a determinação da próxima célula e a busca local Tabu, também foram determinantes para os bons resultados do modelo. Adicionalmente, foram utilizados os modelos de autômatos celulares para a realização do controle geral dos



agentes, bem como a coordenação e sincronização do time de robôs independentemente da tarefa selecionada: forrageamento, coleta seletiva, busca e resgate, e até mesmo a tarefa de vigilância.

4. A partir da formulação do modelo inicial, que foi proposto inicialmente para o controle do time de robôs na tarefa de forrageamento, foi possível adaptá-lo para a realização de outras tarefas específicas, mas que partem da mesma metáfora de busca, dentre elas podemos citar as demais tarefas que foram implementadas e testadas nesta tese, coleta seletiva e busca e resgate, pois todas partem do mesmo conceito inicial. Adicionalmente, também foi possível usar a metáfora da busca na forrageamento para a aplicação na tarefa de vigília utilizando-se um time de robôs.
5. Foi possível mostrar que o nosso modelo polinomial aproximado, a partir de algumas adaptações, pôde ser adaptado para o problema da cobertura de vértices, um problema NP clássico de teoria dos grafos, e vice-versa. Adicionalmente, foi possível mostrar através de uma redução polinomial do modelo de busca do CAA para o problema da cobertura de vértices que o modelo CAA é uma metáfora do problema da cobertura de vértices, onde o maior objetivo é encontrar um tempo número mínimo de robôs que podem se espalhar mais pelo ambiente e encontrar o maior número de alimentos no menor tempo possível.

### 8.3 Contribuição bibliográfica

As principais contribuições deste trabalho em termos de publicações foram relativas aos modelos da tarefa de forrageamento. A primeira submissão (Lima and Oliveira, 2017a) para um periódico internacional Applied Mathematical Modelling (APM) - Qualis A2 da editora Elsevier, refere-se ao modelo CAAM para a tarefa de forrageamento. Neste artigo descrevemos todas as características do modelo CAAM, a inspiração e modelagem inicial matemática do trabalho e também uma análise qualitativa, estatística e numérica do mesmo. Adicionalmente, as características do ambiente de simulação e do time de robôs também é detalhada neste artigo.

No artigo publicado na conferência IEEE International Conference on Systems, Man, and Cybernetics (SMC - Qualis A2) (Lima and Oliveira, 2017b), uma análise formal do modelo CAA foi relatado, onde a memória não é utilizada. Posteriormente, no artigo publicado e apresentado (San Jose, CA, Estados Unidos) pela autora no congresso 14th IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV) - Qualis B1 (Lima and Oliveira, 2016b) uma adaptação na escolha da célula do estado *homing* foi alterada para uma transição probabilística - denominado PCAAM - provendo que o time evitasse o efeito de inércia e formação de filas próximo aos ninhos. Além disso, uma adaptação do modelo foi proposta para a implementação do mesmo no ambiente de

simulação Webots e o modelo é conhecido como RPCAAM. Até o trabalho de PCAAM os robôs ainda não apresentavam uma boa distribuição para os ninhos provocando efeitos de obstrução e congestionamento nos mesmos. Assim, no trabalho publicado e apresentado (Phuket, PK, Tailândia) pela autora no congresso 28th IEEE International Conference on Tools with Artificial Intelligence (ICTAI) - Qualis A2 (Lima and Oliveira, 2016a) foram apresentadas duas alternativas para uma boa distribuição dos robôs próximo aos ninhos. A primeira delas considera a distribuição híbrida onde todos os robôs caminham aleatoriamente até chegarem em uma certa distância para convergirem para uma ação de escolha mais elitista para evitar o atraso perto aos ninhos, e posteriormente foi utilizada a ideia do cálculo de densidade que ainda obteve melhores resultados, sendo que esta última abordagem foi baseada em comportamentos naturais observados em formigas forrageiras.

Uma análise do mundo inteligente (smart world) e dos componentes da biblioteca do Webots foram investigados e publicados no congresso 9th International Conference on Agents and Artificial Intelligence (ICAART) - Qualis B1 (Lima et al., 2017b). Adicionalmente, nesse trabalho foram investigadas diferentes formas para controlar o time de robôs, englobando sincronização e localização. A tarefa de vigilância robótica, uma abstração inicial do modelo de forrageamento, foi publicada no congresso 12th International Conference on Cellular Automata for Research and Industry (ACRI) - Qualis B2 (Lima et al., 2016a) como capítulo de livro na Lecture Notes in Computer Science (LNCS) pelo grupo Springer e no periódico International Journal of Parallel, Emergent and Distributed Systems (IJPEDS) - Qualis B1 (Tinoco, Lima and Oliveira, 2017). Essa tarefa serve como uma aplicação indireta de todas as regras do modelo CAAM referentes ao estado *searching* da tarefa de forrageamento. Por isso, esse estado de busca (*searching*) e cobertura do ambiente foram aplicados e investigados numa nova abordagem para a tarefa de vigilância, que também apresentou resultados iniciais promissores do modelo base da tarefa de forrageamento investigado nesta tese.

A partir de todas essas publicações, é possível perceber que a presente tese apresenta um modelo genérico bastante promissor, que pode ser aplicado não só em uma diversidade de tarefas robóticas, tais como forrageamento, vigília, busca e resgate, e coleta seletiva, mas também em diversas outras abordagens tais como a busca de multi-agentes para otimização de busca, e aplicação desse modelo na modelagem de pedestres. Além disso o trabalho ficou entre os finalistas do Postgraduate Theses Contest (PDT) da 4th IEEE Latin American Conference on Computational Intelligence (LA-CCI), que aconteceu na cidade de Arequipa, no Peru em 2017.

## 8.4 Trabalhos futuros

Como continuidade a este trabalho, pode-se adotar uma implementação em robôs reais na arquitetura e-Puck, que foi fortemente estudada e investigada. A vantagem de realizar

simulações em tal tipo de plataforma é que eles tomam em conta os aspectos físicos de arquiteturas e ambientes, permitindo uma análise mais realista do desempenho do modelo. Além disso, um ambiente tridimensional ou um bidimensional mais complexo com diferentes tipos de obstáculos poderia ser levado em consideração para formular o ambiente de simulação. Experimentos com tempo fixo poderiam ser estabelecidos com a finalidade de verificar quantos objetos seriam coletados pelo time de robôs. Adicionalmente, outras arquiteturas robóticas poderiam ser utilizadas, tais como, drones, robôs humanoides ou utilizar a mistura deles, formando um time heterogêneo com uma abordagem ainda mais híbrida. Questões relevantes, tais como a necessidade de sincronização, comunicação e precisão das informações de dependência pode ser melhor avaliado em tais simulações.

Outro trabalho futuro seria a continuação do presente trabalho levando-se em consideração a utilização de um AC assíncrono. Para o controle não-determinista de colisões entre robôs e obstáculos, uma rede neural artificial (RNA) poderia ser utilizada através da câmera do e-Puck. A RNA é adequada, quando as imagens capturadas apresentam ruídos ou quando o conjunto de regras para fazer a classificação é extenso. Adicionalmente, o conjunto de sensores do tipo IR poderia também ser usado para o controle de pequenos obstáculos e ruídos. A tarefa de planejamento de caminhos também poderia ser investigada através de um algoritmo genético (AG), dado uma configuração do ambiente. Assim o conjunto de robôs poderia fazer a melhor rota do ponto em que ele estava até o ninho ou até o objeto (*searching*). A desvantagem seria que o tempo de processamento do AG atrasaria o processo de conclusão da tarefa.

Outro trabalho interessante seria trabalhar com a melhoria do *smart world* que representa o ambiente de simulação, até mesmo utilizando-se outros simuladores. O mundo poderia ser o responsável pelo controle dos robôs, que leriam as variáveis de ambiente na forma de células do grid em forma de semáforos de controle, indicando se um determinado robô poderia ou não ir para uma determinada célula adjacente. Dessa forma, o comportamento de controle dos robôs seria mais descentralizado. Essa estratégia também poderia ser realizada através da projeção de imagens, que guiaria os robôs os controlando. Outra alteração que poderia ser realizada é o controle do time de robôs feito através de leitores de temperatura (ou outro mecanismo), onde a leitura de informação seria através de um hardware que conseguisse ler a temperatura do ambiente e detectar onde foram os lugares mais ou menos visitados. Adicionalmente, um controle feito por câmeras de celular ou drones móveis poderia melhorar a simulação realista do modelo, evitando dessa maneira a utilização do GPS ou da odometria para este propósito. Dessa forma, a utilização do processamento de imagens e da captura de movimento, guiada por marcadores, seria utilizada em substituição ao GPS, que também dá autonomia em tempo real aos agentes.

As demais tarefas SCAAM e FSCAAM também podem ser implementadas na plataforma de simulação Webots pode ser vista como continuidade deste trabalho. Além

disso, acreditamos que o fato de que o modelo é gerido por regras de transição simples nos permitirá adicionar novos parâmetros para o comportamento social ou físico diferente, o que pode melhorar o desempenho dos robôs da equipe. Adicionalmente, novos parâmetros podem ser inseridos ao modelo, tais como a utilização de feromônio para atrair os robôs aos ninhos bem como a utilização de um PSO ou de recozimento simulado (SA) para este mesmo propósito. Além disso, outras tarefas podem ser investigadas, tais como controle de fluxo em carros, busca guiada por robôs e a melhoria da técnica de vigília. Trabalhos na área de dinâmica de pedestres também poderão ser modeladas a partir desse trabalho, já que novas técnicas foram investigadas com o presente trabalho. Uma regressão linear poderia ser utilizada para informar o número inicial do robô e uma análise poderia ser feita para garantir a melhor posição inicial possível.

---

## Referências Bibliográficas

---

- Agmon, N., Hazon, N., Kaminka, G. A., Group, M., et al. (2008). The giving tree: constructing trees for efficient offline and online multi-robot coverage. *Annals of Mathematics and Artificial Intelligence*, 52(2-4):143–168.
- Agrawal, M. and Konolige, K. (2006). Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1063–1068. IEEE.
- Akbarimajd, A. and Hassanzadeh, A. (2011). A novel cellular automata based real time path planning method for mobile robots. *Int. Journal of Engineering Research and Applications*, 1(4):1262–1267.
- Akbarimajd, A. and Hassanzadeh, A. (2012). Autonomously implemented versatile path planning for mobile robots based on cellular automata and ant colony. *International Journal of Computational Intelligence Systems*, 5(1):39–52.
- Akbarimajd, A. and Lucas, C. (2006). A new architecture to execute cas-based path-planning algorithm in mobile robots. In *Mechatronics, 2006 IEEE International Conference on*, pages 478–482. IEEE.
- Alboul, L., Abdul-Rahman, H. S., Haynes, P. S., Penders, J., and Tharin, J. (2010). An approach to multi-robot site exploration based on principles of self-organisation. In *Intelligent Robotics and Applications*, pages 717–729. Springer.
- Alizadeh, R. (2011). A dynamic cellular automaton model for evacuation process with obstacles. *Safety Science*, 49(2):315–323.
- Alonso-Sanz, R. (2009). Cellular automata with memory. In *Encyclopedia of Complexity and Systems Science*, pages 823–848. Springer.
- Altshuler, E., Ramos, O., Nunez, Y., Fernández, J., Batista-Leyva, A., and Noda, C. (2005). Symmetry breaking in escaping ants. *The American Naturalist*, 166(6):643–649.

- Ando, Y., Fukazawa, Y., Masutani, O., Iwasaki, H., and Honiden, S. (2006). Performance of pheromone model for predicting traffic congestion. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 73–80. ACM.
- Arkin, R. C. (1998). *Behavior based robotics*. Massachusetts Institute of Technology (MIT) press.
- Asimov, I. and de Naveira, R. S. (1987). *Robots e imperio*. Círculo de lectores.
- Bandini, S., Federici, M. L., Manzoni, S., and Vizzari, G. (2006). Towards a methodology for situated cellular agent based crowd simulations. In *Engineering societies in the agents world VI*, pages 203–220. Springer.
- Barlovic, R., Santen, L., Schadschneider, A., and Schreckenberg, M. (1998). Metastable states in cellular automata for traffic flow. *The European Physical Journal B-Condensed Matter and Complex Systems*, 5(3):793–800.
- Barraquand, J. and Latombe, J.-C. (1991). Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*, 10(6):628–649.
- Basilico, N. and Amigoni, F. (2009). Exploration strategies based on multi-criteria decision making for an autonomous mobile robot. In *ECMR*, pages 259–264.
- Beckers, R., Holland, O., and Deneubourg, J.-L. (1994). From local actions to global tasks: Stigmergy and collective robotics. In *Artificial life IV*, volume 181, page 189.
- Behring, C., Bracho, M., Castro, M., and Moreno, J. (2001). An algorithm for robot path planning with cellular automata. In *Theory and practical issues on cellular automata*, pages 11–19. Springer.
- Beni, G. (2005). From swarm intelligence to swarm robotics. In *Swarm Robotics*, pages 1–9. Springer.
- Beni, G. and Wang, J. (1993). Swarm intelligence in cellular robotic systems. In *Robots and Biological Systems: Towards a New Bionics?*, pages 703–712. Springer.
- Berlekamp, E., Conway, J., and Guy, R. (1992). Winning ways for your mathematical plays. *Scientific American*.
- Bhattacharya, S., Likhachev, M., and Kumar, V. (2012). Topological constraints in search-based robot path planning. *Autonomous Robots*, 33(3):273–290.
- Blum, C. and Merkle, D. (2008). *Swarm intelligence*. Springer.
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm intelligence*. Oxford.
- Borenstein, J. and Feng, L. (1996). Measurement and correction of systematic odometry errors in mobile robots. *Robotics and Automation, IEEE Transactions on*, 12(6):869–880.
- Bosque, M. M., Pimenta, L. C., Mesquita, R. C., and AS, G. (2009). Implementação do controle de enxames de robôs utilizando a hidrodinâmica de partículas suavizadas.

- 
- Boukas, E., Kostavelis, I., Gasteratos, A., and Sirakoulis, G. C. (2015). Robot guided crowd evacuation. *IEEE Transactions on Automation Science and Engineering*, 12(2):739–751.
- Braun, A., Bodmann, B. E., and Musse, S. R. (2005). Simulating virtual crowds in emergency situations. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 244–252. ACM.
- Brogan, D. and Hodgins, J. (2002). Simulation level of detail for multiagent control. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*.
- Burstedde, C., Klauck, K., Schadschneider, A., and Zittartz, J. (2001). Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3):507–525.
- Calvo, R., Constantino, A. A., and Figueiredo, M. (2015). A multi-pheromone stigmergic distributed robot coordination strategy for fast surveillance task execution in unknown environments. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–8. IEEE.
- Calvo, R., de Oliveira, J. R., Figueiredo, M., and Romero, R. A. F. (2011a). Bio-inspired coordination of multiple robots systems and stigmergy mechanisms to cooperative exploration and surveillance tasks. In *Cybernetics and Intelligent Systems (CIS), 2011 IEEE 5th International Conference on*, pages 223–228. IEEE.
- Calvo, R., de Oliveira, J. R., Figueiredo, M., and Romero, R. A. F. (2011b). A distributed, bio-inspired coordination strategy for multiple agent systems applied to surveillance tasks in unknown environments. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 3248–3255. IEEE.
- Calvo, R., de Oliveira, J. R., Romero, R. A., and Figueiredo, M. (2012). A bioinspired coordination strategy for controlling of multiple robots in surveillance tasks. *International Journal on Advances in Software Volume 5, Number 3 & 4, 2012*.
- Čapek, K. (2004). *RUR (Rossum's universal robots)*. Penguin.
- Capi, G., Pojani, G., and Kaneko, S.-I. (2008). Evolution of task switching behaviors in real mobile robots. In *Innovative Computing Information and Control, 2008. ICI-CIC'08. 3rd International Conference on*, pages 495–495. IEEE.
- Carneiro, L. O. (2012). Simulação de evacuação de multidão por autômato celular - estudo de caso em um estádio de futebol. Master's thesis, Universidade Federal do Ceará.
- Carvalho, L. B., Mariano, A. S., and de Oliveira, G. M. B. (2008). Evolução de autômatos celulares para resolução de tarefas computacionais. *Horizonte Científico*, 2(2).
- Castro, A. P. and Lima, D. A. (2013a). Autômatos celulares aplicados a modelagem de dinâmica populacional em situação de risco. In *4th Workshop of Applied Computing for the Management of the Environment and Natural Resources (WCAMA) 2014*, pages 57–64.

- Castro, A. P. and Lima, D. A. (2013b). Simulação baseada em autômatos celulares para a evacuação emergencial de pedestres em ambientes com geometria complexa. In *Anais do X Encontro Nacional de Computação - ENACOMP*, pages 1090–1102.
- Castro Silva, L. N. (2005). *Computação natural e suas aplicações*.
- Chen, Y., Chen, N., Wang, Y., Wang, Z., and Feng, G. (2015). Modeling pedestrian behaviors under attracting incidents using cellular automata. *Physica A: Statistical Mechanics and its Applications*, 432(0):287 – 300.
- Cheng, T. M. and Savkin, A. V. (2011). Decentralized control for mobile robotic sensor network self-deployment: barrier and sweep coverage problems. *Robotica*, 29(02):283–294.
- Cheng, T. M., Savkin, A. V., and Javed, F. (2011). Decentralized control of a group of mobile robots for deployment in sweep coverage. *Robotics and Autonomous Systems*, 59(7):497–507.
- Choset, H. (2001). Coverage for robotics—a survey of recent results. *Annals of mathematics and artificial intelligence*, 31(1-4):113–126.
- Choset, H. M. (2005). *Principles of robot motion: theory, algorithms, and implementation*. MIT press.
- Chung, S.-J. and Slotine, J.-J. (2009). Cooperative robot control and concurrent synchronization of lagrangian systems. *Robotics, IEEE Transactions on*, 25(3):686–700.
- Clarke, K. C., Brass, J. A., and Riggan, P. J. (1994). A cellular automation model of wildfire propagation and extinction. *Photogrammetric Engineering and Remote Sensing*, 60(11):1355–1367.
- Colombo, R. M. and Rosini, M. D. (2005). Pedestrian flows and non-classical shocks. *Mathematical Methods in the Applied Sciences*, 28(13):1553–1567.
- Conway, J. H. (1970). Jogo da vida. *Scientific American*. Jogos matematicos.
- Cormen, T. H. (2009). *Introduction to algorithms*. MIT press.
- Cortes, J., Martinez, S., and Bullo, F. (2005). Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation and Calculus of Variations*, 11(04):691–719.
- Couceiro, M. S., Vargas, P. A., and Rocha, R. P. (2014a). Bridging the reality gap between the webots simulator and e-puck robots. *Robotics and Autonomous Systems*, 62(10):1549–1567.
- Couceiro, M. S., Vargas, P. A., Rocha, R. P., and Ferreira, N. M. (2014b). Benchmark of swarm robotics distributed techniques in a search task. *Robotics and Autonomous Systems*, 62(2):200–213.
- Coumans, E. (2013a). Real-time physics simulation. In *Webots*.
- Coumans, E. (2013b). Robot simulator. In *Webots*.



- 
- Craig, J. J. (1989). *Introduction to robotics*, volume 7. Addison-Wesley Reading, MA.
- Crosby, M. E., Scholtz, J., and Ward, P. (2006). Special issue of interacting with computers: Symbiotic performance between humans and intelligent systems. *Interacting with Computers*, 18(6):1165–1169.
- Daoliang, Z., Lizhong, Y., and Jian, L. (2006). Exit dynamics of occupant evacuation in an emergency. *Physica A: Statistical Mechanics and its Applications*, 363(2):501–511.
- Das, A. K., Kantor, G., Kumar, R. V., Pereira, G. A., Peterson, R., Rus, D., Singh, S., and Spletzer, J. (2003). Distributed search and rescue with robot and sensor teams. *Departmental Papers (MEAM)*, page 57.
- Deneubourg, J. L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chretien, L. (1990). The dynamics of collective sorting: robot-like ants and ant-like robots. In *Simulation of Adaptive Behaviour: from animals to animats MIT Press*, pages 356–365.
- Devol, G. C. and Martin, P. S. (1975). Dual-armed multi-axes program controlled manipulators. US Patent 3,890,552.
- Dilão, R. (1993). Autômatos celulares, máquinas de turing ou a natureza como máquina de cálculo. *Colóquio Ciências*, 12:3–20.
- Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., and Winfield, A. (2008). *Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008, Brussels, Belgium, September 22-24, 2008, Proceedings*, volume 5217. Springer.
- Dorigo, M., Maniezzo, V., and Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1):29–41.
- Dowling, K. (1999). Limbless locomotion: learning to crawl. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 4, pages 3001–3006. IEEE.
- Ducatelle, F., Di Caro, G. A., and Gambardella, L. M. (2010). Cooperative self-organization in a heterogeneous swarm robotic system. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 87–94. ACM.
- Eberhart, R. and Kennedy, J. (1995a). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE.
- Eberhart, R. C. and Kennedy, J. (1995b). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, volume 1, pages 39–43. New York, NY.
- EPFL, E. P. F. (2013). E-puck education robot.
- Esser, J. and Schreckenberg, M. (1997). Microscopic simulation of urban traffic based on cellular automata. *International Journal of Modern Physics C*, 8(05):1025–1036.

- Falleiros, E. L. S., Calvo, R., and Ishii, R. P. (2015). Pheroslam: A collaborative and bioinspired multi-agent system based on monocular vision. In *Computational Science and Its Applications–ICCSA 2015*, pages 71–85. Springer.
- Fang, Z., Song, W., Zhang, J., and Wu, H. (2010). Experiment and modeling of exit-selecting behaviors during a building evacuation. *Physica A: Statistical Mechanics and its Applications*, 389(4):815–824.
- Farina, F. and Dennunzio, A. (2008). A predator-prey cellular automaton with parasitic interactions and environmental effects. *Fundamenta Informaticae*, 83(4):337–353.
- Feliciani, C. and Nishinari, K. (2016). An improved cellular automata model to simulate the behavior of high density crowd and validation by experimental data. *Physica A: Statistical Mechanics and its Applications*, 451:135–148.
- Ferreira, G. B., Vargas, P. A., and Oliveira, G. M. (2014). An improved cellular automata-based model for robot path-planning. In *Advances in Autonomous Robotics Systems*, pages 25–36. Springer.
- Ferri, G., Manzi, A., Salvini, P., Mazzolai, B., Laschi, C., and Dario, P. (2011). Dustcart, an autonomous robot for door-to-door garbage collection: From dustbot project to the experimentation in the small town of peccioli. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 655–660. IEEE.
- Frisch, U., Hasslacher, B., and Pomeau, Y. (1986). Lattice-gas automata for the navier-stokes equation. *Physical review letters*, 56(14):1505.
- Fu, L., Luo, J., Deng, M., Kong, L., and Kuang, H. (2012). Simulation of evacuation processes in a large classroom using an improved cellular automaton model for pedestrian dynamics. *Procedia Engineering*.
- Fu, Z., Zhou, X., Zhu, K., Chen, Y., Zhuang, Y., Hu, Y., Yang, L., Chen, C., and Li, J. (2015). A floor field cellular automaton for crowd evacuation considering different walking abilities. *Physica A: Statistical Mechanics and its Applications*, 420:294–303.
- Fujisawa, R., Imamura, H., and Matsuno, F. (2013). Cooperative transportation by swarm robots using pheromone communication. In *Distributed Autonomous Robotic Systems*, pages 559–570. Springer.
- Fukui, M. and Ishibashi, Y. (1999). Self-organized phase transitions in cellular automaton models for pedestrians. *Journal of the physical society of Japan*, 68(8):2861–2863.
- Funge, J., Tu, X., and Terzopoulos, D. (1999). Cognitive modeling knowledge, reasoning and planning for intelligent character. In *Proceedings of ACM SIGGRAPH99*.
- Gabriely, Y. and Rimon, E. (2001). Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):77–98.
- Gage, D. W. (1992). Command control for many-robot systems. Technical report, DTIC Document.
- Georgoudas, I. G., Sirakoulis, G. C., and Andreadis, I. T. (2006). A simulation tool for modelling pedestrian dynamics during evacuation of large areas. In *Artificial Intelligence Applications and Innovations*, pages 618–626. Springer.

- 
- Glover, F. (1989). Tabu search part i. *ORSA Journal on computing*, 1(3):190–206.
- Glover, F. (1990). Tabu search part ii. *ORSA Journal on computing*, 2(1):4–32.
- Gordon, D. M. (2002). The organization of work in social insect colonies. *Complexity*, 8(1):43–46.
- Gordon, D. M. (2007). Control without hierarchy. *Nature*, 446(7132):143–143.
- Gordon, D. M. (2013). The rewards of restraint in the collective regulation of foraging by harvester ant colonies. *Nature*, 498(7452):91–93.
- Gordon, D. M. (2014). The ecology of collective behavior. *PLoS Biol*, 12(3):e1001805.
- Gordon, D. M., Paul, R. E., and Thorpe, K. (1993). What is the function of encounter patterns in ant colonies? *Animal Behaviour*, 45(6):1083–1100.
- Greff, J. and Nolfi, S. (2010). Evolution of implicit and explicit communication in mobile robots. In *Evolution of Communication and Language in Embodied Agents*, pages 179–214. Springer.
- Green, S. (2013). Physx geforce. In *NVidia*.
- Guo, F., Li, X., Kuang, H., Bai, Y., and Zhou, H. (2016). An extended cost potential field cellular automata model considering behavior variation of pedestrian flow. *Physica A: Statistical Mechanics and its Applications*, 462:630–640.
- Guo, R. and Huang, H. (2008). A mobile lattice gas model for simulating pedestrian evacuation. *Physica A: Statistical Mechanics and its Applications*, 387(2):580–586.
- Guo, R.-Y. (2014). New insights into discretization effects in cellular automata models for pedestrian evacuation. *Physica A: Statistical Mechanics and its Applications*, 400:1–11.
- Guo, R. Y., Huang, H. J., and Wong, S. C. (2011). Collection, spillback, and dissipation in pedestrian evacuation: a network-based method. *Transportation Research Part B*.
- Hamann, H. and Wörn, H. (2007). An analytical and spatial model of foraging in a swarm of robots. In *Swarm Robotics*, pages 43–55. Springer.
- Hazon, N. and Kaminka, G. A. (2005). Redundancy, efficiency and robustness in multi-robot coverage. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 735–741. IEEE.
- Hecker, J. P. and Moses, M. E. (2015). Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. *Swarm Intelligence*, 9(1):43–70.
- Helbing, D., Farkas, I., and Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490.
- Helbing, D., Isobe, M., Nagatani, T., and Takimoto, K. (2003). Lattice gas simulation of experimentally studied evacuation dynamics. *Physical Review E*, 67(6):067101.
- Helbing, D. and Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282.

- Henderson, L. (1971). The statistics of crowd fluids.
- Henein, C. M. and White, T. (2006). Information in crowds: The swarm information model. In *Cellular Automata*, pages 703–706. Springer.
- Hernandez, L., Hoya, S., MartaÂn, A., and RodriÂguez, G. (2007). Modelling forest fire spread using hexagonal cellular automata. *Applied Mathematical Modelling*, 31(6):1213 – 1227.
- Holland, J., Spindler, K., Horodyski, F., Grabau, E., Nichol, S., and VandePol, S. (1982). Rapid evolution of rna genomes. *Science*, 215(4540):1577–1585.
- Hoogendoorn, J. and Bovy, P. (2003). Pedestrian travel behavior modeling. In *Moving through Nets: The Physical and Social Dimensions of Travel 10th International Conference on Travel Behavior Research*.
- Hou, L., Liu, J.-G., Pan, X., and Wang, B.-H. (2014). A social force evacuation model with the leadership effect. *Physica A: Statistical Mechanics and its Applications*, 400:93–99.
- Howard, A., Matarić, M. J., and Sukhatme, G. S. (2002). Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5*, pages 299–308. Springer.
- Huang, H., Zhang, L., Guan, Y., and Wang, D. (2008). A cellular automata model for population expansion of spartina alterniflora at jiuduansha shoals, shanghai, china. *Estuarine, Coastal and Shelf Science*, 77(1):47 – 55.
- Huang, P. and Liu, Z. (2012). A model of pedestrian crowd behavior for evacuation simulation. *Letters in Advanced Science*.
- Hughes, R. L. (2002). A continuum theory for the flow of pedestrians. *Transportation Research Part B: Methodological*, 36(6):507–535.
- Hughes, R. L. (2003). The flow of human crowds. *Annual review of fluid mechanics*, 35(1):169–182.
- Hwang, Y. K. and Ahuja, N. (1992). Gross motion planning a survey. *ACM Computing Surveys (CSUR)*, 24(3):219–291.
- Ioannidis, K., Sirakoulis, G. C., and Andreadis, I. (2008). A cellular automaton collision-free path planner suitable for cooperative robots. In *Informatics, 2008. PCI’08. Panhellenic Conference on*, pages 256–260. IEEE.
- Ioannidis, K., Sirakoulis, G. C., and Andreadis, I. (2011a). Cellular ants: A method to create collision free trajectories for a cooperative robot team. *Robotics and Autonomous Systems*, 59(2):113–127.
- Ioannidis, K., Sirakoulis, G. C., and Andreadis, I. (2011b). A path planning method based on cellular automata for cooperative robots. *Applied Artificial Intelligence*, 25(8):721–745.
- Ioannidis, K., Sirakoulis, G. C., and Andreadis, I. (2015). Cellular robotic ants synergy coordination for path planning. In *Robots and Lattice Automata*, pages 197–228. Springer.

- 
- Ishibashi, Y. and Fokui, M. (1994). Temporal variations of traffic flow in the biham-middleton-levine model. *Journal of the Physical Society of Japan*, 63(8):2882–2885.
- Isobe, M., Helbing, D., and Nagatani, T. (2004). Experiment, theory, and simulation of the evacuation of a room without visibility. *Physical Review E*, 69(6):066132.
- Itoh, T. and Nagatani, T. (2002). Optimal admission time for shifting the audience. *Physica A: Statistical Mechanics and its Applications*, 313(3):695–708.
- Ji, L., Qian, Y., Zeng, J., Wang, M., Xu, D., Yan, Y., and Feng, S. (2013). Simulation of evacuation characteristics using a 2-dimensional cellular automata model for pedestrian dynamics. *Journal of Applied Mathematics*, 2013.
- Jian, L., Lizhong, Y., and Daoliang, Z. (2005). Simulation of bi-direction pedestrian movement in corridor. *Physica A: Statistical Mechanics and its Applications*, 354:619–628.
- Kantor, G., Singh, S., Peterson, R., Rus, D., Das, A., Kumar, V., Pereira, G., and Spletzer, J. (2006). Distributed search and rescue with robot and sensor teams. In *Field and Service Robotics*, pages 529–538. Springer.
- Karp, R. M. (1977). Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane. *Mathematics of operations research*, 2(3):209–224.
- Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer.
- Kirchner, A., Klüpfel, H., Nishinari, K., Schadschneider, A., and Schreckenberg, M. (2003a). Simulation of competitive egress behavior: comparison with aircraft evacuation data. *Physica A: Statistical Mechanics and its Applications*, 324(3):689–697.
- Kirchner, A., Nishinari, K., and Schadschneider, A. (2003b). Friction effects and clogging in a cellular automaton model for pedestrian dynamics. *Physical review E*, 67(5):056122.
- Kirchner, A. and Schadschneider, A. (2002). Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, pages 260 – 276.
- Kong, C. S., Peng, N. A., and Rekleitis, I. (2006). Distributed coverage with multi-robot system. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2423–2429. IEEE.
- Kubo, M., Sato, H., Yoshimura, T., Yamaguchi, A., and Tanaka, T. (2014). Multiple targets enclosure by robotic swarm. *Robotics and Autonomous Systems*, 62(9):1294–1304.
- Kucharska, E., Grobler-Dębska, K., Rączka, K., and Dutkiewicz, L. (2016). Cellular automata approach for parallel machine scheduling problem. *Simulation*, 92(2):165–178.
- Kumar, S., Lai, T. H., and Arora, A. (2005). Barrier coverage with wireless sensors. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 284–298. ACM.

- Lerman, K., Martinoli, A., and Galstyan, A. (2005). A review of probabilistic macroscopic models for swarm robotic systems. In *Swarm robotics*, pages 143–152. Springer.
- Lévy, P. (1997). *Collective intelligence*. Plenum/Harper Collins.
- Li-jun, J., Jin-chang, C., and Wei-jie, Z. (2009). A crowd evacuation simulation model based on 2.5-dimension cellular automaton. In *Virtual Environments, Human-Computer Interfaces and Measurements Systems, 2009. VECIMS'09. IEEE International Conference on*, pages 90–95. IEEE.
- Lima, D. A. (2010). Modelos de criptografia baseados no cálculo de pré-imagem de autômatos celulares. *Trabalho de Conclusão de Curso em Ciência da Computação - Universidade Federal de Uberlândia*, pages 20 – 21.
- Lima, D. A. (2011). Análise do método de criptografia hca baseado em autômatos celulares segundo a teoria dos grafos. In *Divulgação Online dos Anais do Workshop de Teses e Dissertações em Ciência da Computação (WTDCC)*, Uberlandia/MG, Brasil.
- Lima, D. A. (2012). Modelo criptográfico baseado em autômatos celulares tridimensionais híbridos. Master's thesis, Universidade Federal de Uberlandia (UFU) Faculdade de Computacao (FACOM).
- Lima, D. A., Oliveira, C. C., Pestili, L. C., Silva, E. C., Bezerra, M. A., and Lima, H. A. (2017a). Uma proposta de sistema de aprendizagem com conteúdo gamificado e com reforço guiado por algoritmos bio-inspirados. *Anais do Computer on the Beach*, pages 140–149.
- Lima, D. A. and Oliveira, G. M. (2017a). A cellular automata ant memory model of foraging in a swarm of robots. *Applied Mathematical Modelling*, 47:551 – 572.
- Lima, D. A. and Oliveira, G. M. B. (2015). Modelo de criptografia baseado em autômatos celulares tridimensionais híbridos. BR Patent App. BR1,020,130,100,447.
- Lima, D. A. and Oliveira, G. M. B. (2016a). New bio-inspired coordination strategies for multi-agent systems applied to foraging tasks. In *28th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2016)*, volume 2, pages 117–124. IEEE.
- Lima, D. A. and Oliveira, G. M. B. (2016b). A probabilistic cellular automata ant memory model for a swarm of foraging robots. In *14th International Conference on Control, Automation, Robotics and Vision (ICARCV 2016)*, volume 3, pages 245–250. IEEE.
- Lima, D. A. and Oliveira, G. M. B. (2017b). Formal analysis in a cellular automata ant model using swarm intelligence in robotics foraging task. In *Proceedings of the IEEE International Conference On Systems, Man, and Cybernetics (SMC 2017)*, pages 1793 – 1798.
- Lima, D. A., Tinoco, C. R., and Oliveira, G. M. (2016a). A cellular automata model with repulsive pheromone for swarm robotics in surveillance. In *International Conference on Cellular Automata Research and Industry (ACRI)*, pages 312–322. Springer Publishing.
- Lima, D. A., Tinoco, C. R., Viedman, J. M. N., and Oliveira, G. M. B. (2017b). Coordination, synchronization and localization investigations in a parallel intelligent robot

- cellular automata model that performs foraging task. In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, pages 355–363.
- Lima, H. A. and Lima, D. A. (2014). Automatos celulares estocasticos bidimensionais aplicados a simulacao de propagacao de incendios em florestas homogeneas. In *5th Workshop of Applied Computing for the Management of the Environment and Natural Resources (WCAMA) 2014*, Brasilia/DF, Brazil.
- Lima, M. V., Oliveira, C. C., and Lima, D. A. (2016b). Uma ferramenta computacional para simulaco de espalhamento de fluidos baseada em autmatos celulares bidimensionais estocsticos. In *Anais do XXXVI Congresso da Sociedade Brasileira de Computaco (CSBC)*, volume 36, pages 1876–1885. Editora Universitria da Pontifcia Universidade Catlica do Rio Grande do Sul (EDIPUCRS).
- Lin, Q., Ji, Q., and Gong, S. (2006). A crowd evacuation system in emergency situation based on dynamics model. In *Interactive Technologies and Sociotechnical Systems*, pages 269–280. Springer.
- Liu, L., Yang, T. F., and Li, J. (2009). Evacuation from a classroom considering the occupant density around exits. *Physica A*.
- Liu, W., Winfield, A. F., Sa, J., Chen, J., and Dou, L. (2007). Towards energy optimization: Emergent task allocation in a swarm of foraging robots. *Adaptive Behavior*, 15(3):289–305.
- Lo, S., Huang, H.-C., Wang, P., and Yuen, K. (2006). A game theory based exit selection model for evacuation. *Fire Safety Journal*, 41(5):364–369.
- Lovas, G. (1994). Modeling and simulation of pedestrian traffic flow. *Transportation Research B*.
- Marchese, F. (1996). Cellular automata in robot path planning. In *Advanced Mobile Robot, 1996., Proceedings of the First Euromicro Workshop on*, pages 116–125. IEEE.
- Marchese, F. (2011). Mrs motion planning: the spatiotemporal multilayered cellular automata approach.
- Marcolino, L. S. and Chaimowicz, L. (2008). Experiments in the coordination of large groups of robots. In *Advances in Artificial Intelligence-SBIA 2008*, pages 268–277. Springer.
- Marjovi, A., Nunes, J. G., Marques, L., and de Almeida, A. (2009). Multi-robot exploration and fire searching. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1929–1934. IEEE.
- Markus, M. and Kusch, I. (1995). Cellular automata for modelling the shell pigmentation of molluscs, journal of biological systems. *Journal of Biological Systems*, 3:999 – 1011.
- Martinoli, A., Easton, K., and Agassounon, W. (2004). Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *The International Journal of Robotics Research*, 23(4-5):415–436.

- Martinoli, A., Ijspeert, A. J., and Gambardella, L. M. (1999). A probabilistic model for understanding and comparing collective aggregation mechanisms. In *Advances in artificial life*, pages 575–584. Springer.
- Masár, M. (2013). A biologically inspired swarm robot coordination algorithm for exploration and surveillance. In *Proceedings of the IEEE 17th international conference on intelligent engineering systems, INES*, pages 271–275.
- McLurkin, J. and Smith, J. (2007). Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In *Distributed autonomous robotic systems 6*, pages 399–408. Springer.
- Milazzo, J., Roupail, N., Hummer, J., and Allen, D. (1998). The effect of pedestrians on the capacity of signalized intersections. *Transportation Research Record 1646*, (5):1646. National Research Council (Washington DC).
- Mitchell, M., Crutchfield, J. P., and Das, R. (1997). Computer science application: evolving cellular automata. *Handbook of evolutionary computation*, Oxford University Press, Oxford.
- Moere, A. V. and Clayden, J. J. (2005). Cellular ants: Combining ant-based clustering with cellular automata. In *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, pages 8–pp. IEEE.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65. IPCB: Instituto Politécnico de Castelo Branco.
- Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., and Erkmen, A. M. (2008). Search and rescue robotics. In *Springer Handbook of Robotics*, pages 1151–1173. Springer.
- Nagai, R., Fukamachi, M., and Nagatani, T. (2005). Experiment and simulation for counterflow of people going on all fours. *Physica A: Statistical Mechanics and its Applications*, 358(2):516–528.
- Nagai, R., Fukamachi, M., and Nagatani, T. (2006). Evacuation of crawlers and walkers from corridor through an exit. *Physica A: Statistical Mechanics and its Applications*, 367:449–460.
- Nagai, R., Nagatani, T., Isobe, M., and Adachi, T. (2004). Effect of exit configuration on evacuation of a room without visibility. *Physica A: Statistical Mechanics and its Applications*, 343:712–724.
- Nishinari, K., Kirchner, A., Namazi, A., and Schadschneider, A. (2004). Extended floor field ca model for evacuation dynamics. *IEICE Transactions on information and systems*, 87(3):726–732.
- Nishinari, K., Sugawara, K., Kazama, T., Schadschneider, A., and Chowdhury, D. (2006). Modelling of self-driven particles: Foraging ants and pedestrians. *Physica A: Statistical Mechanics and its Applications*, 372(1):132–141.



- 
- Nourbakhsh, I., Sycara, K., Koes, M., Yong, M., Lewis, M., and Burion, S. (2005). Human-robot teaming for search and rescue. *Pervasive Computing, IEEE*, 4(1):72–79.
- Oliveira, G., Vargas, P. A., and Ferreira, G. (2015). Investigating a cellular automata model that performs three distance diffusion on a robot path planning. In *Proceedings of the European Conference on Artificial Life 2015 (ECAL 2015)*, pages 271–278.
- Oliveira, G. M., Vargas, P. A., and Ferreira, G. B. (2014). A local decision making cellular automata-based path-planning. In *Proceedings of 11th Encontro Nacional de Inteligencia Artificial e Computacional*, pages 129–136. Biblioteca Digital Brasileira de Computacao (BDBComp).
- Oliveira, G. M. B. (2003). Automatos celulares: aspectos dinamicos e computacionais. *III Jornada de Mini-cursos em Inteligencia Artificial (MCIA) - Sociedade Brasileira de Computacao*, 8:297 – 345.
- Oliveira, P. P. and Vaiano, R. B. (2005). Buscando um autômato celular para resolver o problema da paridade. *Encontro Nacional de Inteligência Artificial(ENIA2005)*, pages 1202–1205.
- Ortigoza, G. M. (2015). Unstructured triangular cellular automata for modeling geographic spread. *Applied Mathematics and Computation*, 258:520–536.
- Pan, X., Han, C. S., Dauber, K., and Law, K. H. (2007). A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *Ai & Society*, 22(2):113–132.
- Parisi, D. and Dorso, C. (2005). Microscopic dynamics of pedestrian evacuation. *Physica A: Statistical Mechanics and its Applications*, 354:606–618.
- Pelechano, N., Allbeck, J. M., and Badler, N. I. (2007). Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 99–108. Eurographics Association.
- Pereira, L., Duczmal, L., and Cruz, F. (2011). Simulacao de evacuacao emergencial via automatos celulares: Uma proposta de modificacao do modelo de schadschneider. *Anais do CMAC*, pages 21 – 30.
- Perez, G., Tapang, G., Lim, M., and Saloma, C. (2002). Streaming, disruptive interference and power-law behavior in the exit dynamics of confined pedestrians. *Physica A: Statistical Mechanics and its Applications*, pages 609 – 618.
- Pini, G., Tuci, E., and Dorigo, M. (2007). Evolution of social and individual learning in autonomous robots. In *Ecal Workshop: Social Learning in Embodied Agents*.
- Powathil, G. G., Gordon, K. E., Hill, L. A., and Chaplain, M. A. J. (2012). Modelling the effects of cell-cycle heterogeneity on the response of a solid tumour to chemotherapy: Biological insights from a hybrid multiscale cellular automaton model. *Journal of Theoretical Biology*, 308(0):1 – 19.
- Puliafito, J. L. (2007). A transport model for the evolution of urban systems. *Applied Mathematical Modelling*, 31(11):2391 – 2411.

- Purcaru, C., Precup, R.-E., Iercan, D., Fedorovici, L.-O., David, R.-C., and Dragan, F. (2013). Optimal robot path planning using gravitational search algorithm. *International Journal of Artificial Intelligence*, 10(S13):1–20.
- Reynolds, C. (1987). Flocks, herds, and schools: a distributed behavior model. In *Proceedings of ACM SIGGRAPH*.
- Reynolds, C. (1999). Steering behaviors for autonomous characters. *Game Developers Conference*.
- Rothermel, R. C. (1972). A mathematical model for predicting fire spread in wildland fuels. Master's thesis, U.S. Department of Agriculture.
- Rothermel, R. C., Deeming, J. E., et al. (1980). *Measuring and interpreting fire behavior for correlation with fire effects*. Intermountain Forest and Range Experiment Station, US Department of Agriculture, Forest Service.
- Russell, S. and Norvig, P. (2005). *Artificial Intelligence a modern approach*, volume 2.
- Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In *Swarm robotics*, pages 10–20. Springer.
- Şahin, E. and Winfield, A. (2008). Special issue on swarm robotics. *Swarm Intelligence*, 2(2):69–72.
- Saigol, Z., Ridder, B., Wang, M., Dearden, R., Fox, M., Hawes, N., Lane, D. M., and Long, D. (2015). Efficient search for known objects in unknown environments using autonomous indoor robots. In *IROS-15 Workshop on Task Planning for Intelligent Robots in Service and Manufacturing*.
- Saloma, C., Perez, G. J., Tapang, G., Lim, M., and Palmes-Saloma, C. (2003). Self-organized queuing and scale-free behavior in real escape panic. *Proceedings of the National Academy of Sciences*, 100(21):11947–11952.
- Santos, Z., Maria, R., and Coutinho, S. (2001). Dynamics of hiv infection: A cellular automata approach. *Phys. Rev. Lett.*, 87:168102.
- Sarkar, C. and Abbasi, S. A. (2006). Enhancing the accuracy of forecasting impact of accidents in chemical process industry by the application of cellular automata technique. *Process Safety and Environmental Protection*, 84(5):355 – 370.
- Schadschneider, A. (2001a). Bionics-inspired cellular automaton model for pedestrians dynamics traffic and granular flow. pages 110 – 114.
- Schadschneider, A. (2001b). Cellular automaton approach to pedestrian dynamics-theory. *arXiv preprint cond-mat/0112117*.
- Schadschneider, A., Klingsch, W., Klüpfel, H., Kretz, T., Rogsch, C., and Seyfried, A. (2011). Evacuation dynamics: Empirical results, modeling and applications. In *Extreme Environmental Events*, pages 517–550. Springer.
- Schadschneider, A. and Schreckenberg, M. (1993). Cellular automation models and traffic flow. *Journal of Physics A: Mathematical and General*, 26(15):L679.

- 
- Schultz, M., Lehmann, S., and Fricke, H. (2007). A discrete microscopic model for pedestrian dynamics to manage emergency situations in airport terminals. In *Pedestrian and Evacuation Dynamics 2005*, pages 369–375. Springer.
- Seyfried, A., Steffen, B., and Lippert, T. (2006). Basics of modelling the pedestrian flow. *Physica A: Statistical Mechanics and its Applications*, 368(1):232–238.
- Shu, C. and Buxton, H. (1995). Parallel path planning on the distributed array processor. *Parallel computing*, 21(11):1749–1767.
- Silva, E. C., Soares, J. A., and Lima, D. A. (2016). Autômatos celulares unidimensionais caóticos com borda fixa aplicados à modelagem de um sistema criptográfico para imagens digitais. *Revista de Informática Teórica e Aplicada*, 23(1):250–276.
- Silva, R. G. O. (2015). Um modelo baseado em automatos celulares e algoritmos geneticos para a navegacao de um time de robos visando o controle de formacao e o desvio de obstaculos. Master’s thesis, Universidade Federal de Uberlandia (UFU) - Faculdade de Computacao.
- Sipser, M. (2006). *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston.
- Smith, R. (2013). Open dynamics engine. In *Robots and Systems*.
- Somló, J., Lantos, B., and Cat, P. T. (1997). *Advanced Robot Control Advances in Electronics 14*. Akadémiai Kiadó.
- Song, W., Xu, X., Wang, B.-H., and Ni, S. (2006). Simulation of evacuation processes using a multi-grid model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, 363(2):492–500.
- Soofiyani, F. R., Rahmani, A. M., and Mohsenzadeh, M. (2010). A straight moving path planner for mobile robots in static environments using cellular automata. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on*, pages 67–71. IEEE.
- Sugawara, K. and Sano, M. (1997). Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system. *Physica D: Nonlinear Phenomena*, 100(3):343–354.
- Tajima, Y. and Nagatani, T. (2001). Scaling behavior of crowd flow outside a hall. *Physica A: Statistical Mechanics and its Applications*, 292(1):545–554.
- Tajima, Y. and Nagatani, T. (2002). Clogging transition of pedestrian flow in t-shaped channel. *Physica A: Statistical Mechanics and its Applications*, 303(1):239–250.
- Tajima, Y., Takimoto, K., and Nagatani, T. (2001). Scaling of pedestrian channel flow with a bottleneck. *Physica A: Statistical Mechanics and its Applications*, 294(1):257–268.
- Takimoto, K. and Nagatani, T. (2003). Spatio-temporal distribution of escape time in evacuation process. *Physica A: Statistical Mechanics and its Applications*, 320:611–621.

- Tavakoli, Y., Javadi, H. H. S., and Adabi, S. (2008). A cellular automata based algorithm for path planning in multi-agent systems with a common goal. *International Journal of Computer Science and Network Security*, 8(7):119–123.
- Tikhanoff, V., Cangelosi, A., Fitzpatrick, P., Metta, G., Natale, L., and Nori, F. (2008). An open-source simulator for cognitive robotics research: the prototype of the icub humanoid robot simulator. In *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pages 57–61. ACM.
- Tinoco, C. R., Lima, D. A., and Oliveira, G. M. B. (2017). An improved model for swarm robotics in surveillance based on cellular automata and repulsive pheromone with discrete diffusion. *International Journal of Parallel, Emergent and Distributed Systems*, 25(1):1–25.
- Toyama, M. C., Bazzan, A. L., and Da Silva, R. (2006). An agent-based simulation of pedestrian dynamics: from lane formation to auditorium evacuation. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 108–110. ACM.
- Tsiftsis, A., Georgoudas, I. G., and Sirakoulis, G. C. (2016). Real data evaluation of a crowd supervising system for stadium evacuation and its hardware implementation. *IEEE Systems Journal*, 10(2):649–660.
- Varas, A., Cornejo, M., Mainemer, D., Toledo, B., Rogan, J., Munoz, V., and Valdivia, J. (2007). Cellular automaton model for evacuation process with obstacles. *Physica A: Statistical Mechanics and its Applications*, 382(2):631 – 642.
- Vargas, P. A., Benhalen, A. M., Pessin, G., and Osório, F. S. (2012). Applying particle swarm optimization to a garbage and recycling collection problem. In *Computational Intelligence (UKCI), 2012 12th UK Workshop on*, pages 1–8. IEEE.
- von Neumann, J. and Burks, A. (1966). Theory of self-reproducing automata. *University of Illinois Press*.
- Wagner, D., Tissot, M., Cuevas, W., and Gordon, D. M. (2000). Harvester ants utilize cuticular hydrocarbons in nestmate recognition. *Journal of Chemical Ecology*, 26(10):2245–2257.
- Wei-Guo, S., Yan-Fei, Y., Bing-Hong, W., and Wei-Cheng, F. (2006). Evacuation behaviors at exit in ca model with force essentials: A comparison with social force model. *Physica A: Statistical Mechanics and its Applications*, 371(2):658–666.
- Weidmann, U. (1992). Transporttechnik der fugnger. *IVT Institut fr Verkehrsplanung Transporttechnik*, 2(1):1–23.
- Weifeng, F., Lizhong, Y., and Weicheng, F. (2003). Simulation of bi-direction pedestrian movement using a cellular automata model. *Physica A: Statistical Mechanics and its Applications*, 321(3):633–640.
- Weng, W., Chen, T., Yuan, H., and Fan, W. (2006). Cellular automaton simulation of pedestrian counter flow with different walk velocities. *Physical Review E*, 74(3):036102.

- 
- Winfield, A. F. (2009). Foraging robots. In *Encyclopedia of Complexity and Systems Science*, pages 3682–3700. Springer.
- Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of Modern Physics*.
- Wolfram, S. (1986). *Cellular Automata*. Los Alamos Science.
- Wolfram, S. (2002a). *A New Kind of Science*. Wolfram Media - (1st edition): 1197 - 2006-09-19T07:35:05.000+0200.
- Wolfram, S. (2002b). *A New Kind of Science*. Wolfram Media.
- Woolley, A. W., Chabris, C. F., Pentland, A., Hashmi, N., and Malone, T. W. (2010). Evidence for a collective intelligence factor in the performance of human groups. *science*, 330(6004):686–688.
- Wu, J., Wang, X., Chen, J., Shu, G., and Li, Y. (2015). The position of a door can significantly impact on pedestrians evacuation time in an emergency. *Applied Mathematics and Computation*, 258:29–35.
- Yamamoto, K., Kokubo, S., and Nishinari, K. (2007). Simulation for pedestrian dynamics by real-coded cellular automata (rca). *Physica A: Statistical Mechanics and its Applications*, 379(2):654–660.
- Yang, L., Fang, W., and Fan, W. (2003). Modeling occupant evacuation using cellular automata - effect of human behavior and building characteristics on evacuation. *Journal of Fire Sciences*.
- Yang, Z., Fang, W., Huang, R., Huang, R., and Deng, z. (2002). Occupant evacuation model based on cellular automata in fire. *Chinese Science Bulletin.*, pages 1484 – 1488.
- Yang, Z., Zhao, L., Li, J., and Fang, Y. (2005). Simulation of the kin behavior in building occupant evacuation based on cellular automaton. *Building and Environment*, pages 411 – 415.
- Yu, Y. and Song, W. (2007). Cellular automaton simulation of pedestrian counter flow considering the surrounding environment. *Physical Review E*, 75(4):046112.
- Yuan, W. and Tan, K. H. (2007). An evacuation model using cellular automata. *Physica A: Statistical Mechanics and its Applications*, 384(2):549–566.
- Zarboutis, N. and Marmaras, N. (2004). Searching efficient plans for emergency rescue through simulation: the case of a metro fire. *Cognition, Technology & Work*, 6(2):117–126.
- Zarita, Z. and Aik, L. E. (2012). Intelligent exit-selection behaviors during a room evacuation. *Chinese Physics Letters*.
- Zheng, M., Kashimori, Y., and Kambara, T. (2002). A model describing collective behaviors of pedestrians with various personalities in danger situations. In *Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference on*, volume 4, pages 2083–2087 vol.4.

- Zheng, X., Koenig, S., Kempe, D., and Jain, S. (2010). Multirobot forest coverage for weighted and unweighted terrain. *Robotics, IEEE Transactions on*, 26(6):1018–1031.
- Zheng, X., Zhong, T., and LIU, M. (2009). Modeling crowd evacuation of a building based on seven methodological approaches. *Building and Environment*, pages 437 – 445.
- Zheng, Y., Chen, J., Wei, J., and Guo, X. (2012). Modeling of pedestrian evacuation based on the particle swarm optimization algorithm. *Physica A: Statistical Mechanics and its Applications*, 391(17):4225–4233.
- Zheng, Y., Jia, B., Li, X.-G., and Zhu, N. (2011). Evacuation dynamics with fire spreading based on cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 390(18):3147–3156.
- Zieliński, C. and Winiarski, T. (2010). General specification of multi-robot control system structures. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 58(1):15–28.
- Zong, X., Xiong, S., and Fang, Z. (2014). A conflict–congestion model for pedestrian–vehicle mixed evacuation based on discrete particle swarm optimization algorithm. *Computers & Operations Research*, 44:1–12.