



UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**Problema de Perseguição-Evasão de Pior Caso em
Ambientes Limitados Baseado em *Random Walk***
Dissertação

Daniel Augusto Santos Franco

Orientador: Prof. Dr. Elyson Ádan Nunes Carvalho

São Cristóvão-SE

Dezembro de 2023



UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA-PROEE

TERMO DE APROVAÇÃO

“Random Walk como Solução para o Problema de Perseguição-Evasão de Pior Caso em Ambientes Limitados”

Discente:

Daniel Augusto Santos Franco

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Sergipe, como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

Aprovada pela banca examinadora composta por:

Marcos Vinícius Silva Alves
Prof. Dr. Marcos Vinícius Silva Alves (PROEE/UFS)
Presidente

Jugurta R. Montalvão Filho
Prof. Dr. Jugurta Rosa Montalvão Filho (PROEE/UFS)
Examinador Interno

Phillippe Cardoso Santos
Prof. Dr. Phillippe Cardoso Santos (IFS)
Examinador Externo

Daniel Augusto Santos Franco
Daniel Augusto Santos Franco
Discente

Cidade Universitária “Prof. José Aloísio de Campos”, 22 de dezembro de 2023.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE

F825p Franco, Daniel Augusto Santos
Problema de perseguição-evasão de pior caso em ambientes limitados baseado em *random walk* / Daniel Augusto Santos Franco ; orientador Elyson Ádan Nunes Carvalho. - São Cristóvão, 2023.
66 f. : il.

Dissertação (mestrado em Engenharia Elétrica) – Universidade Federal de Sergipe, 2023.

1. Engenharia elétrica. 2. Robôs móveis. 3. Passeio aleatório (Matemática). I. Carvalho, Elyson Ádan Nunes orient. II. Título.

CDU 621.31

Agradecimentos

Agradeço, primeiramente, a minha família por sempre me apoiar e me incentivar a seguir o que quero. Um agradecimento especial aos meus pais, Cláudia e Francisco, e a minha irmã, Carla, que sempre tiveram paciência comigo e sempre me disseram o que precisava ouvir em vez de me falarem o que queria ouvir.

Agradeço aos meus amigos que sempre me ajudaram, principalmente aos amigos que levo como irmãos Levi, Antônio e Ariel.

Agradeço ao meu orientador, Elyson Carvalho, por aceitar me orientar e pelas diversas reuniões que duravam horas, mesmo quando estava tão ocupado.

Agradeço igualmente a todos os orientadores do GPR-UFS: Eduardo Freire, Elyson Carvalho, Gilmar Carvalho e Lucas Molina, por sempre me orientarem a ser uma pessoa e profissional melhor.

Agradeço também a toda família GPR-UFS por me acolher tão bem e me fazer sentir em casa, principalmente Arthur, Beq, Bunitinho, Cone, Davidi, Lázaro, Mestre, Vanessa, Thau, e Xandão, por todas as discussões fervorosas, científicas ou não, mas que sempre agregam e todos acabam melhor do que quando começaram.

Agradeço a banca por aceitar o convite de avaliar este trabalho.

Agradeço à CAPES pelo apoio financeiro durante o período de desenvolvimento deste trabalho.

Resumo

Uma das motivações de utilizar robôs se dá por sua capacidade de realizar tarefas que possuem riscos sem expor os humanos, como é o caso de tarefas de resgate ou de busca. Para essas tarefas, é comum a utilização de sistemas com múltiplos robôs para uma melhor robustez ou pela tarefa ser incapaz de ser completada com apenas um robô. Uma dessas tarefas é a de perseguição-evasão, em que o robô perseguidor tem o objetivo de capturar um evasor dentro de uma área de interesse. Quando as condições da tarefa são que o evasor possui velocidade infinita e conhecimento completo do ambiente, enquanto o perseguidor possui velocidade limitada e não possui conhecimento do ambiente, essa é considerada a perseguição-evasão de pior caso, à qual este trabalho propõe uma solução. Nessa solução, os robôs possuem uma movimentação baseada em *random walk* para garantir que se for possível completar a tarefa, essa será completada, sendo essa demonstração também realizada neste trabalho. Ainda neste trabalho são propostas duas polarizações para a movimentação baseada em *random walk* a fim de diminuir o tempo médio de execução da tarefa. Para a validação da abordagem, o método é testado em um ambiente de simulação desenvolvido. Nesse ambiente também é testado o impacto das polarizações e a variação do número de robôs no tempo médio para completar a tarefa. O método proposto é então capaz de finalizar a tarefa em um tempo finito desde que se use um número de robôs capaz de completar a tarefa.

Palavras-chaves: Perseguição-evasão. Sistemas com múltiplos robôs. Pior-caso de perseguição-evasão. *Random walk*. Perseguição-evasão com *random walk*.

Abstract

One of the motivations to use robots is their capacity to complete dangerous tasks without exposing humans, such as search and rescue tasks. In these tasks, it is common to use multiple robots for robustness or because the task is impossible to be completed with only one robot. One of these tasks is the pursuit-evasion. To complete this task the robots must capture the evader that is inside of an area of interest. If the conditions of the task are that the evader has infinite velocity and complete knowledge of the environment, while the pursuers have limited velocity and without knowledge of the environment, it is considered the worst-case pursuit-evasion, and this work proposes a method to solve it. The robots in this method have a random walk-based movement so the method has a guarantee of completing the task if it is possible to complete it and the demonstration is also shown in this work. This work also proposes two polarizations for the random walk-based movement to reduce the mean time to complete the task. To validate the method a simulation environment is implemented, where the method was tested. In this environment is also tested the polarizations proposed and how the number of robots influenced the mean time of the system to complete the task. The proposed method is capable of finalizing the task in a finite time if there are enough robots to be able to complete the task.

Key-words: Pursuit-evasion. Multiple robot system. Worst-case of pursuit-evasion. Random walk. Pursuit-evasion with random walk.

Sumário

Lista de ilustrações	I
Lista de tabelas	IV
1 Introdução	1
2 Objetivos	4
2.1 Objetivos gerais	4
2.2 Objetivos específicos	4
3 Fundamentação Teórica	5
3.1 Perseguição-evasão em grafos	5
3.2 Perseguição-evasão por visibilidade	6
3.3 Cooperação entre múltiplos robôs para perseguição-evasão	8
3.4 Pior-caso de perseguição-evasão	8
3.5 <i>Random walk</i> na tarefa de perseguição-evasão	10
3.5.1 <i>Random Walk</i>	10
3.5.2 <i>Random walk</i> aplicado ao pior-caso de perseguição-evasão	11
4 Revisão bibliográfica	13
5 <i>Random walk</i> aplicado ao pior-caso de perseguição-evasão em ambientes multiplamente conectados	19
5.1 Algoritmo proposto	19
5.1.1 Movimentação dos robôs	23
5.1.1.1 Restrições locais	24
5.1.1.2 Cálculo de probabilidades de movimento	24
5.1.2 Separação e Junção dos times	28
5.2 Discussão sobre a completude do sistema	32
5.2.1 Noção intuitiva da completude do sistema	32
5.2.2 Demonstração da completude do sistema	33
6 Resultados	37
6.1 Configuração dos experimentos	37
6.2 Análise das polarizações propostas	40
6.3 Comparação com vários números de robôs	45
7 Conclusão	50

Referências 52

Lista de ilustrações

Figura 1	– (a) Mapa a ser decomposto e (b) Mapa decomposto em grafo.	6
Figura 2	– Perseguição-evasão em uma representação por visibilidade, com a área vista pelo perseguidor em azul, o limite de visão em vermelho, áreas sem visão em amarelo e em verde os obstáculos	7
Figura 3	– Perseguidores P1 e P2 descontaminando o ambiente em cinza, com a área contaminada em vermelho e em azul a área descontaminada (a) Início do sistema; (b) Limpeza de uma região; (c) Recontaminação da região.	9
Figura 4	– Representação do <i>Random walk</i> unidimensional com probabilidade p em um sentido e q no outro em que n é a posição discretizada.	10
Figura 5	– Random walk com barreiras.	10
Figura 6	– Posição atual do robô demarcada pela cruz e possíveis posições futuras marcadas por círculos.	12
Figura 7	– (a) Probabilidades do <i>random walk</i> sem ponderação; (b) Probabilidades do <i>random walk</i> com ponderação.	12
Figura 8	– Robôs realizando a limpeza do ambiente com uma formação em linha conectando as paredes.	15
Figura 9	– Estratégia de procura por obstáculos ao não conseguir prosseguir com a formação em linha.	16
Figura 10	– Tipos de ambiente em que os obstáculos são representados em preto e em branco é a área que o robô pode navegar (a) Simplesmente conectado; (b) Multiplicamente conectado.	17
Figura 11	– Organização da linha de robôs pelas restrições estratégicas.	17
Figura 12	– Time sendo visto como barreira móvel, em que o vermelho é o ambiente contaminado, em azul é o ambiente descontaminado, em preto são os obstáculos, em cinza é a área dentro da região de descontaminação dos robôs, em branco são os líderes e cinza escuro os não líderes. (a) O time de robôs acha um obstáculo o que faz o ambiente multiplicamente conectado; (b) O time em cinza mais escuro é visto como uma barreira móvel pelo outro time.	20
Figura 13	– Fluxograma de uma iteração do algoritmo proposto com cálculo de probabilidades de movimentação dos robôs, a movimentação desses e verificação de separação e junção.	22

Figura 14	– Fluxograma até a finalização da tarefa do algoritmo proposto com cálculo de probabilidades de movimentação dos robôs, a movimentação desses e verificação de separação e junção.	23
Figura 15	– Probabilidade de movimento para cada possível célula.	25
Figura 16	– Caso em que o robô não possui células contaminadas próximas e atrapalha a movimentação do time.	27
Figura 17	– Movimentação desejada de time em região descontaminada.	28
Figura 18	– Visão geral do método, em vermelho é o ambiente contaminado, em azul é o ambiente descontaminado, em preto são os obstáculos, em cinza é a área dentro da região de descontaminação robôs e em branco são os líderes e cinza escuro os não líderes. (a) O time de robôs acha um novo obstáculo; (b) O time se separa, com o novo time sendo mostrado em tonalidade diferente; (c) Os times continuam sua movimentação pelo mapa; (d) O time completa a descontaminação do obstáculo; (e) O time se junta.	30
Figura 19	– Detalhes da separação do time. (a) Um robô encontra a parede; (b) Dois robôs encontram a mesma parede; (c) É feita a verificação das células em amarelo por contaminação entre os robôs e o obstáculo; (d) O time se separa em dois, com o novo time mostrado em tonalidade diferente e os times continuam a realização da tarefa.	30
Figura 20	– Células a serem verificadas em amarelo, com os ângulos em verde e azul do obstáculo mais próximo do robô até o outro robô e as retas em magenta que ligam do robô até os limites dos ângulos.	31
Figura 21	– Detalhes da junção do time. (a) Os dois líderes se aproximam, com os times mostrados em tonalidades diferentes; (b) É feita a verificação das células em amarelo por contaminação; (c) Os times se juntam em um time; (d) O time continua a descontaminação do ambiente.	31
Figura 22	– Representação do conjunto W em que os robôs se movimentam.	33
Figura 23	– Estados S_j referentes a cada elemento do conjunto W	34
Figura 24	– Ambiente de simulação.	38
Figura 25	– Fluxograma da simulação detalhando o processo descontaminação e contaminação do ambiente, bem como as condições de parada da simulação.	38
Figura 26	– Mapas utilizados (a) Mapa 1; (b) Mapa 2; (c) Mapa 3; (d) Mapa 4.	39
Figura 27	– Média de iterações para valores de k_{pol1} em cada mapa.	41
Figura 28	– Evolução da descontaminação pelas iterações para o teste com menor iterações das polarizações analisadas para os mapas: (a) Mapa 1; (b) Mapa 2; (c) Mapa 3; (d) Mapa 4.	43

Figura 29	– Teste realizado no mapa 3, com os obstáculos já percebidos pelos robôs em preto, em azul a área descontaminada, em vermelho a área contaminada e em cinza (com tonalidade diferente para os dois times) a área de captura dos robôs para as seguintes iterações: (a) iteração 150; (b) iteração 500; (c) iteração 2900; (d) iteração 3203.	43
Figura 30	– Exemplo em que a polaridade atrapalha a movimentação desejada. . .	45
Figura 31	– Evolução da descontaminação pelas iterações para o teste com menor iterações da polarização escolhida para os mapas: (a) Mapa 1; (b) Mapa 2; (c) Mapa 3; (d) Mapa 4.	46
Figura 32	– Exemplo de configuração dos robôs em que não é possível que a tarefa seja concluída sem que os times voltem e se separem com outra configuração.	47
Figura 33	– Evolução da descontaminação pelas iterações para o teste com menor iterações da polarização escolhida para os mapas: (a) Mapa 1; (b) Mapa 2; (c) Mapa 3; (d) Mapa 4.	48

Lista de tabelas

Tabela 1	– Taxa de completude para as polarizações e mapas propostos.	40
Tabela 2	– Média e desvio padrão da quantidade de iterações.	41
Tabela 3	– Taxa de completude para os mapas propostos.	44
Tabela 4	– Média e desvio padrão da quantidade de iterações.	44
Tabela 5	– Taxa de completude para a quantidade de robôs e mapas escolhidos para o método proposto.	45
Tabela 6	– Média e desvio padrão para a quantidade de robôs e mapas escolhidos para o método proposto e o método de Kolling.	47

1 Introdução

A robótica possui diversas características que fazem com que ela seja utilizada em favor da sociedade. Dentre tais características está a capacidade de execução de tarefas precisas e repetitivas em um curto espaço de tempo e, também, a capacidade de execução de tarefas que não são possíveis, ou são muito perigosas para o ser humano realizar.

A robótica frequentemente é dividida de forma didática em duas grandes áreas que se relacionam: a robótica de manipuladores e robótica móvel. Essa separação se dá devido à capacidade dos robôs móveis de se mover pelo ambiente [1], enquanto os manipuladores possuem uma base fixa que limita a distância em que o robô pode atuar [2].

A robótica de manipuladores é utilizada principalmente em indústrias, para a manipulação de objetos, embalagem de produtos, realização de soldas, montagem de equipamentos. Porém, em virtude de possuírem uma base fixa, eles possuem limitação quanto a sua área de atuação. Para solucionar essas tarefas é indicado o uso de robôs móveis.

Devido à capacidade de se movimentar pelo ambiente, os robôs móveis podem realizar tarefas como transporte de cargas [3], exploração de áreas de interesse [4], resgate de pessoas em áreas destruídas [5], procura de alvos de interesse [6], entre outras.

Algumas tarefas, porém, podem ser impossíveis ou inviáveis de serem realizadas com a utilização de apenas um robô, em decorrência, por exemplo da complexidade da tarefa, como em tarefas de exploração [7] e resgate [5], da limitação dos robôs disponíveis, como no transporte de objetos [8], de robôs com limitações sensoriais [9], ou da necessidade de redundância na ação, como nas redes de sensores sem fio com agentes móveis [10].

Dada a inserção da robótica nas diversas áreas de aplicação, tais características mostram a importância do desenvolvimento de pesquisas em sistemas com múltiplos robôs. Como forma de realizar essas tarefas, os robôs devem atuar de maneira coordenada a fim de completar a tarefa. Essa coordenação entre múltiplos robôs pode também ser utilizada para diminuir o tempo necessário para a conclusão de uma tarefa que poderia ser completada apenas com um robô. A utilização de múltiplos robôs possui, portanto, diversas vantagens em comparação com a utilização de um único robô [11].

A utilização de múltiplos robôs também aumenta a robustez do sistema, já que, em caso de perda, a tarefa de um robô pode ser distribuída para os demais robôs na cena, o que não é possível com a utilização de apenas um robô para a realização da tarefa desejada [12].

Outra vantagem de se utilizar sistemas com múltiplos robôs é a capacidade de

paralelização de tarefas, diminuindo assim a carga de trabalho de cada robô. Dessa forma, podem ser utilizados robôs mais simples para a execução da tarefa, o que pode diminuir os custos de fabricação do robô [13].

Essa utilização de robôs mais simples pode ser vista na realização do transporte de uma carga pesada. Para essa tarefa, pode ser utilizado um robô que suporta a carga, ou podem ser utilizados vários robôs mais simples e que suportam uma carga menor, mas que distribuem o peso a fim de carregar a carga em conjunto [14].

São observadas também aplicações em que além dos agentes cooperando na cena, existem outros agentes que atuam de forma competitiva, ou seja, atuando para que a tarefa não possa ser completada, o que pode dificultar significativamente a conclusão da tarefa, como no caso do futebol de robôs, em que um time adversário tenta impedir a tarefa, ou na perseguição-evasão, foco deste trabalho.

Na tarefa de perseguição-evasão, considera-se que enquanto um robô, ou um time de robôs, tem a função de encontrar e encurralar o evasor, este, por sua vez, deve se movimentar de forma a evitar a captura.

Quando se considera que o evasor, ao contrário dos perseguidores, tem conhecimento irrestrito do ambiente e agentes na cena, além de velocidade ilimitada, tem-se a tarefa de perseguição-evasão de pior caso, que normalmente não pode ser executada com apenas um robô e que se confunde com a tarefa de limpeza do ambiente [15].

A tarefa de limpeza do ambiente pode ser aplicada em situações como a descontaminação de uma área com gás nocivo ou a limpeza de resíduos nucleares. Dessa forma, caso o gás nocivo entre em contato com o ar, esse se torna nocivo. Para que a tarefa seja concluída, é então necessário que o robô seja capaz de eliminar todo o gás do ambiente ao mesmo tempo que impede completamente que o gás tenha interação com o ar limpo.

Esse problema de perseguição-evasão do pior caso tem soluções propostas em [16, 15, 17] que podem ser aplicadas para solução do problema, mas não garantem que a tarefa será concluída. A não conclusão da tarefa pode ser devido à incapacidade de se garantir que os robôs não permitam a recontaminação do ambiente ou aos robôs não serem capazes de se coordenar adequadamente em situações específicas, como nos casos em que é necessário uma disposição mais eficiente dos robôs durante a limpeza ou quando o time precisa se dividir para que seja possível completar a tarefa.

No entanto, a falta de garantia que a tarefa será concluída (garantia de completude), pode ser um fator limitante para diversas aplicações. A fim de resolver essa problemática, uma solução com garantia de completude foi proposta em [18], baseando-se em uma movimentação aleatória dos robôs, de acordo com o modelo de

random walk.

Porém, a solução de [18] é limitada a ambientes que não possuem obstáculos, o que resulta em sua aplicação apenas em conjuntos (ambientes) simplesmente conectados. Essa limitação faz com que a solução tenha menor interesse prático, pois a maioria dos ambientes de interesse possuem objetos internos.

Dessa forma, neste trabalho é proposta uma extensão do método apresentado em [18], abrangendo a aplicação de movimentação baseada em *random walk* para conjuntos (ambientes) multiplamente conectados e limitados, com resultados parciais discutidos em [19], além de apresentar uma demonstração formal que a tarefa é completada em um $t < \infty$.

São propostas também duas polarizações a fim de diminuir o tempo necessário para completar a tarefa. Essas polarizações são então testadas para validar como elas influenciam o tempo de conclusão da tarefa. Também foram realizados testes para verificação do impacto do número de robôs na tarefa e comparados os resultados do método proposto com os obtidos pelo método de [15].

Esta dissertação está organizada nos seguintes capítulos: no capítulo 2 são mostrados os objetivos do trabalho. No capítulo 3 é realizada uma fundamentação teórica sobre perseguição-evasão e a utilização do *random walk*. No capítulo 4 é realizada uma revisão bibliográfica sobre a perseguição-evasão. No capítulo 5 mostra a solução proposta para o problema e demonstra a convergência do método proposto. Os resultados encontrados são mostrados no capítulo 6 e, por fim, as conclusões são apresentadas no capítulo 7.

2 Objetivos

2.1 Objetivos gerais

O objetivo geral deste trabalho é desenvolver uma solução baseado em *random walk*, com prova de completude, para o problema de perseguição-evasão de pior caso em ambientes limitados.

2.2 Objetivos específicos

Como objetivos específicos, tem-se:

- Avaliar o estado da arte acerca da tarefa de pior-caso da perseguição-evasão;
- Estender o uso de *random walk* para a tarefa de perseguição evasão de pior caso em ambientes multiplamente conectados;
- Implementar a solução proposta;
- Demonstrar que a solução proposta completa a tarefa em tempo finito;
- Comparar a solução com outros métodos capazes de resolver a tarefa.

3 Fundamentação Teórica

Neste capítulo estão dispostas uma fundamentação teórica sobre o problema da perseguição-evasão na robótica e formulações clássicas do problema. Essas formulações apresentadas estão divididas em algumas classificações do problema: perseguição-evasão em grafos, perseguição-evasão por visibilidade, cooperação entre múltiplos robôs para perseguição-evasão e pior-caso de perseguição-evasão. É também apresentada uma fundamentação teórica de *random walk* e como esse é utilizado para a tarefa do pior-caso de perseguição-evasão.

O problema de perseguição-evasão também é chamado de busca adversária [20], por se tratar de dois times adversários, no qual o time de perseguidores tem como objetivo capturar o time de evasores, em que estes possuem o objetivo de permanecer o maior tempo possível sem ser capturados [21]. A forma clássica do problema se dá quando um único robô perseguidor tenta capturar um único evasor, e a captura é realizada quando o perseguidor possui a mesma posição do evasor.

Para essa formulação clássica do problema, são consideradas as restrições físicas de movimentação dos robôs e são propostos controladores que consigam concluir a tarefa [22, 23]. Essa formulação, porém, possui várias limitações, como a velocidades do perseguidor ter que ser similar à do evasor, ou o perseguidor necessitar da posição do evasor.

Essas limitações de velocidades similares e de conhecimento da posição fazem com que, na prática, as soluções propostas para essa formulação não possam ser utilizadas para todas as tarefas. A fim de conseguir solucionar essas limitações, outras formulações foram propostas. Uma dessas, é dada ao representar o mapa como um grafo, em que os robôs se movimentam entre os nós desse grafo.

3.1 Perseguição-evasão em grafos

A representação de perseguição-evasão em grafos considera que os robôs ocupam todo o espaço do nó em que estão posicionados, então a captura é dada quando o evasor ocupa o mesmo nó que o perseguidor.

Com essa formulação, o controlador do robô não é mais o foco da solução. As soluções para essa formulação possuem como foco serem capazes de limitar a movimentação do evasor para a sua captura.

Inicialmente proposto por [24], a representação em grafos decompõe as regiões do ambiente que podem ser ocupadas pelo perseguidor ou evasor em nós e utiliza arestas para definir a possibilidade de movimentação entre nós conectados, e consequentemente

entre as áreas representadas por eles.

Um exemplo desse tipo de representação é apresentado na Figura 1. Na Figura 1(a) é mostrado um exemplo de ambiente, enquanto na Figura 1(b) é mostrada sua representação em grafo. É importante notar que cada área indicada na Figura 1(a) foi representada por um nó na Figura 1(b) e que uma aresta foi utilizada para indicar a existência de conexão entre as regiões.

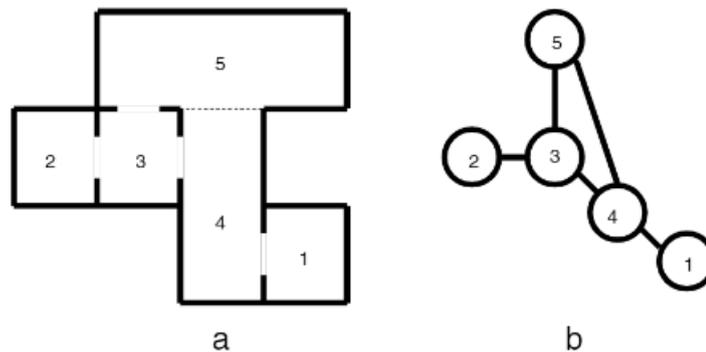


Figura 1 – (a) Mapa a ser decomposto e (b) Mapa decomposto em grafo [25].

Na Figura 1 é mostrado um exemplo em que certos problemas não podem ser solucionados com a utilização de apenas um robô, pois esse apenas irá conseguir realizar a captura se o evasor iniciar nos nós 1 ou 2 e o perseguidor nos nós 4 ou 3, respectivamente. Para as outras configurações de inicialização não é possível garantir a captura, uma vez que o evasor sempre teria duas opções de caminho e poderia se manter em fuga indefinidamente.

As abordagens para essa formulação do problema buscam então descobrir o número mínimo de perseguidores que completam a tarefa, e a construção do planejamento para finalizar a tarefa.

Uma limitação dessa formulação é a necessidade de construção do grafo que represente o ambiente. É necessário também que esse grafo seja conhecido pelos robôs perseguidores antes de iniciar a tarefa, o que torna essa formulação limitada apenas para mapas conhecidos.

A fim de resolver algumas dessas limitações, a formulação de perseguição-evasão por visibilidade foi proposta. Em que os robôs se movimentam em um mapa contínuo, sem a necessidade de decompor o mapa em grafos.

3.2 Perseguição-evasão por visibilidade

O que diferencia a formulação de perseguição-evasão por visibilidade da formulação clássica é que os perseguidores não possuem conhecimento da posição dos evasores, e a

captura é dada como concluída quando o perseguidor possui visão dos evasores.

Como a posição dos evasores não é conhecida, o perseguidor utiliza os sensores que possui para visualizar o mapa e separar em regiões conhecidas e desconhecidas. Com essas regiões, o perseguidor pode planejar uma trajetória de forma que consiga ter visualização de áreas que eram desconhecidas e passam a ser conhecidas. Dessa forma, o perseguidor diminuirá as regiões desconhecidas até possuir visualização do evasor e completar a tarefa.

Para facilitar o processamento dessas regiões, usualmente são utilizados mapas poligonais. Esses mapas diminuem o processamento por necessitar calcular apenas as linhas entre o perseguidor e os vértices vistos, como mostrado na Figura 2.

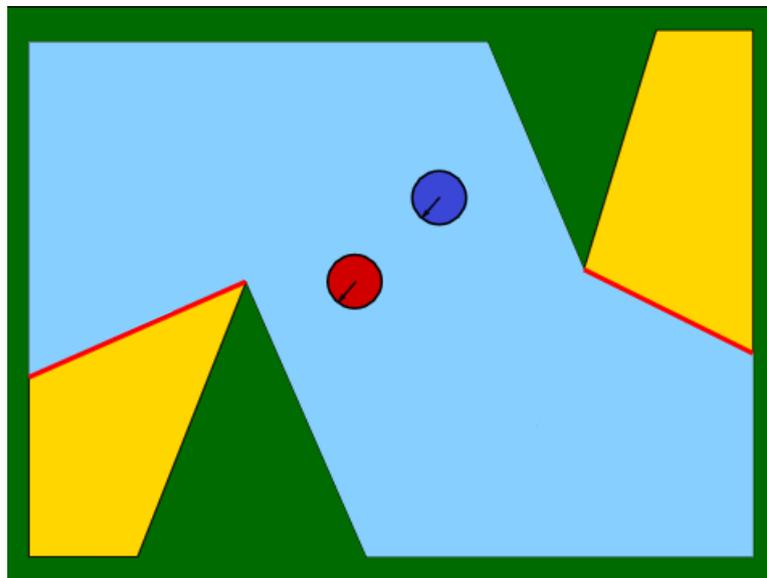


Figura 2 – Perseguição-evasão em uma representação por visibilidade, com a área vista pelo perseguidor em azul, o limite de visão em vermelho, áreas sem visão em amarelo e em verde os obstáculos [26].

Na Figura 2, o robô azul é o agente perseguidor e a área azul do ambiente é o espaço que está visível pelo robô naquele momento. Da mesma forma, o robô vermelho é o evasor e a área amarela é a área não visível ao perseguidor, que é calculada a partir dos pontos de quinas dos objetos em verde e o centro do perseguidor.

Para o exemplo da Figura 2, o robô consegue visualizar todo o mapa se ele se movimentar inicialmente para a esquerda, pois a região desconhecida na esquerda se tornaria conhecida e após isso para o canto inferior direito.

Porém, assim como a formulação de perseguição-evasão em grafos, apenas com um robô não é possível completar a tarefa para qualquer mapa. Por causa disso, é necessário que haja a cooperação entre múltiplos robôs para que seja possível completar a tarefa em mapas que não seriam possíveis ao se utilizar um único robô.

3.3 Cooperação entre múltiplos robôs para perseguição-evasão

Ao se utilizar múltiplos robôs, além de aumentar a robustez, flexibilidade e escalabilidade comparado com um sistema que utiliza apenas um robô, o sistema com múltiplos robôs torna viável a realização de algumas tarefas que não poderiam ser realizadas somente com um único agente ou se tornariam mais complexas [13]. No entanto, para que estas métricas sejam alcançadas, é necessário que os múltiplos robôs tenham coordenação ao executar cada parte da tarefa, e essa coordenação pode ser na forma de competição, em que os agentes competem para executar determinada tarefa [27], ou cooperação [28], em que os robôs cooperam entre si para completar a tarefa.

Outro fator importante para a cooperação é a comunicação entre os agentes, que pode ser de forma direta ou indireta [29]. Nessa, os robôs percebem os outros a partir de modificações no ambiente e naquela os agentes recebem informações dos demais.

Especificamente, o sistema pode ser centralizado, quando as informações de todos os robôs são enviadas para um robô central que realiza a tomada de decisão para todos os robôs e envia os comandos para serem executados [30]; ou distribuído quando cada robô toma a própria decisão com base nas informações disponíveis.

A forma de tomada de decisão centralizada acarreta em uma alta coordenação, e conseqüentemente, pode gerar uma melhor alocação de tarefas. Porém, esta forma possui uma menor tolerância para falhas e todos os robôs necessitam de comunicação com o robô central.

Uma rede descentralizada pode ser melhor para situações em que se requeira uma maior tolerância a falhas ou que a comunicação dos robôs com o robô central possa ser interrompida, já que cada agente realiza sua própria decisão apenas com informações próprias e dos robôs vizinhos [30].

A tarefa de perseguição-evasão, na prática, pode ter fatores que dificultem a conclusão da tarefa. Esses fatores são: o conhecimento parcial do mapa, a velocidade do evasor ser maior do que o esperado e os sensores dos perseguidores possuírem uma limitação de distância.

Por conta desses fatores, é necessário que haja abordagens que sejam capazes de solucionar a tarefa mesmo para o pior caso.

3.4 Pior-caso de perseguição-evasão

O pior-caso de perseguição-evasão é dado então ao considerar que a velocidade do evasor é ilimitada e esse possui conhecimento completo do mapa e posição dos perseguidores. Os perseguidores, por sua vez, possuem velocidade e distância de

sensoriamento limitadas, e não possuem conhecimento nem do mapa, nem do evasor [15].

Com todas essas considerações, o problema pode ser entendido como um problema de limpeza [31], em que o evasor é visto como um gás nocivo que contaminou todo o ambiente previamente. Os perseguidores, que tem a tarefa de limpar o ambiente, realizam essa tarefa ao buscar os evasores pelo mapa. Essa busca realizada precisa garantir que o evasor não possa se movimentar livremente para regiões em que a busca foi realizada previamente. Esse comportamento pode ser visto na Figura 3.

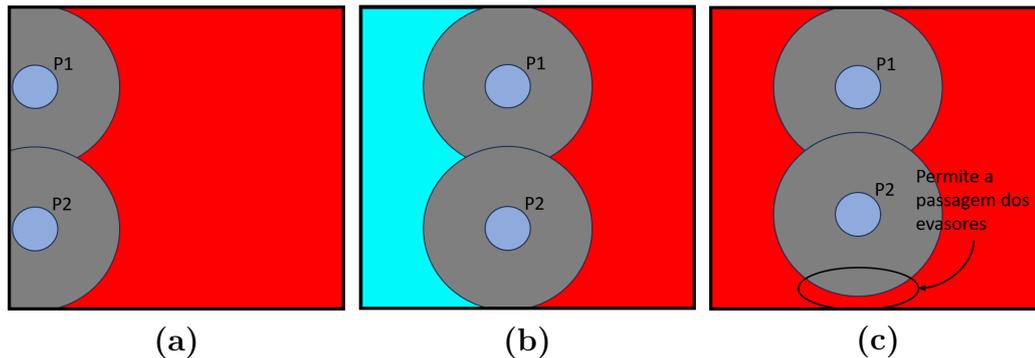


Figura 3 – Perseguidores P_1 e P_2 descontaminando o ambiente em cinza, com a área contaminada em vermelho e em azul a área descontaminada (a) Início do sistema; (b) Limpeza de uma região; (c) Recontaminação da região.

Como visto na Figura 3(a), que representa o começo da tarefa, os robôs P_1 e P_2 estão em um mapa desconhecido e toda a região em vermelho são posições em que o evasor pode estar. A região em cinza é referente a distância de sensoriamento dos perseguidores. Dessa forma, pode-se garantir que dentro da região em cinza não existem evasores, pois já teriam sido capturados.

Na Figura 3(b) os robôs se movimentaram para a direita, criando uma área que não possui evasores em azul, já que já foram previamente verificados e não foi aberta uma passagem que os evasores pudessem passar. Na Figura 3(c) é aberta uma passagem, a qual está destacada, que permite que os evasores se movimentem por ela. Como os evasores possuem velocidade infinita, eles podem estar em qualquer posição da área em vermelho, sendo entendidos assim como uma contaminação do ambiente.

A contaminação irá se espalhar para áreas descontaminadas se não for limitada por nenhuma barreira, seja ela física, no caso de um obstáculo, ou seja por um robô estar descontaminando o local por onde a contaminação iria se espalhar.

Dessa forma, os robôs precisam se coordenar para que a área em que o gás está seja reduzida com o passar do tempo, até que não tenha mais nenhuma contaminação, completando assim a tarefa.

Uma das metodologias para completar a tarefa é utilizar robôs que possuem movimentação baseada em *random walk* [18].

3.5 *Random walk* na tarefa de perseguição-evasão

Na secção 3.5.1 é realizada uma fundamentação teórica sobre a *random walk*, e tem sua aplicação na perseguição-evasão discutida na secção 3.5.2.

3.5.1 *Random Walk*

A *random walk* é um modelo estocástico que possui usos em diversas áreas, incluindo termodinâmica, economia, física, biologia, quando usado em uma dimensão, e utilizada em outras áreas, como em ecologia, quando usado de forma bidimensional [32]. No contexto da robótica, a *random walk* pode, entre outras aplicações, ser utilizada para exploração de ambientes, onde o agente possui sua movimentação definida por um modelo de probabilidade. Para o caso de apenas uma dimensão discretizada, a *random walk* pode ser representada como na Figura 4, em que p é a probabilidade de se movimentar para a direita e q a probabilidade de se movimentar para a esquerda, utilizando assim o modelo de Bernoulli [32].

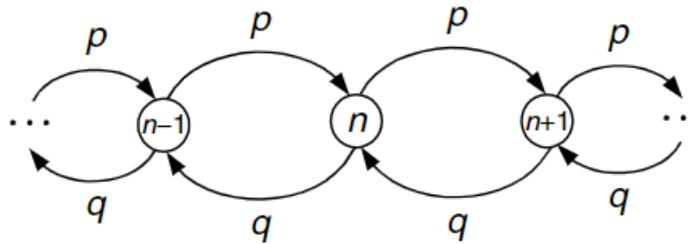


Figura 4 – Representação do *Random walk* unidimensional com probabilidade p em um sentido e q no outro em que n é a posição discretizada [32].

No entanto em aplicações de robótica, normalmente há um limite máximo que o robô pode se mover em uma direção, seja por encontrar um obstáculo ou sair da região de interesse. Na *random walk* esses comportamentos podem ser modelados como barreiras de reflexão, como mostrado na Figura 5.

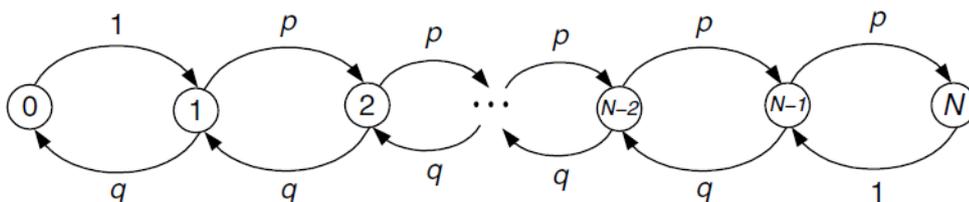


Figura 5 – *Random walk* com barreiras [32].

Essas barreiras são utilizadas de forma que se a *random walk* chegar na posição da barreira, não é possível continuar na mesma direção, fazendo com que a única movimentação possível seja retornar para a posição anterior, caso não haja também uma

barreira de absorção. A barreira de absorção faz com que a *random walk* permaneça na mesma posição para todo o tempo futuro e é normalmente utilizada no objetivo da tarefa.

Para a movimentação em mais de uma dimensão, pode-se utilizar o mesmo modelo de movimentação para a direção x e outro para a direção y , que podem ser alternados ou simultâneos, fazendo com que cada direção seja independente. Outra forma de se utilizar a *random walk* para aplicações em duas dimensões em robótica é dada por [33, 34, 35], que utilizam a *random walk* em um ambiente bidimensional e contínuo para a exploração de ambientes com múltiplos robôs, no qual os robôs seguem as direções definidas pela *random walk*.

Em [18], essa movimentação é realizada em um ambiente discreto bidimensional, em que cada robô pode se movimentar em uma das quatro direções ou permanecer parado, tendo sido primeiramente considerada a mesma probabilidade para os eventos e posteriormente realizada uma ponderação com o intuito de reduzir o tempo de convergência do sistema. Por utilizar *random walk* como estratégia de movimentação, foi possível realizar a demonstração da completude do sistema, pois a *random walk* possui a garantia de percorrer todo o espaço em $t < \infty$.

Quando utilizada em grafos, a *random walk* pode ser entendida de forma que a sequência de vértices do grafo pode ser representada como uma cadeia de Markov [36], em que um ponto do grafo tem conexão com outros pontos do grafo e pode-se construir a matriz de transição de probabilidade de se movimentar de um vértice para outro. No caso de um grafo com pesos na sua construção, a cadeia de Markov gerada será uma cadeia em que as transições são proporcionais ao peso entre os vértices no grafo.

A cadeia de Markov pode ser utilizada para diversas aplicações na robótica, como supervisão de um local [37] de forma estocástica para dificultar a predição de movimentação por parte de invasores. Também pode ser utilizada para procura de alvos [38, 39] de forma probabilística em áreas com dificuldades de conectividade, ou em patrulhas com visibilidade limitada [40].

3.5.2 *Random walk* aplicado ao pior-caso de perseguição-evasão

Uma solução para o pior-caso de perseguição-evasão foi proposta por [18], utilizando a *random walk* para garantir a completude da tarefa, e que será explicado no Capítulo 4, mas que os elementos explicados a seguir serão apresentados por se tratar de elementos importantes para a fundamentação teórica desse trabalho. Para isso, foi considerado o ambiente de forma discreta, e como mostrado na Figura 6, o robô pode ficar na mesma posição, ou se movimentar para as 4 direções de Von Neumann.

Para a movimentação de cada robô é utilizada uma estratégia de movimentação

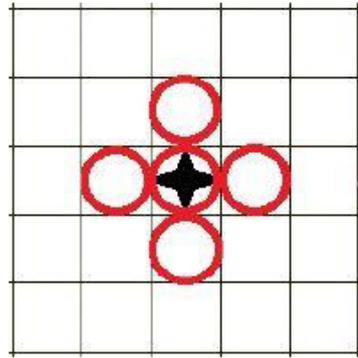


Figura 6 – Posição atual do robô demarcada pela cruz e possíveis posições futuras marcadas por círculos.

baseada na *random walk*, que é mostrada na Figura 7. As porcentagens que aparecem na Figura 7 representam as probabilidades da célula ser sorteada. Após o sorteio, o robô se movimenta para a célula sorteada.

0%	20%	0%
20%	20%	20%
0%	20%	0%

(a)

0%	P_5	0%
P_4	P_1	P_2
0%	P_3	0%

(b)

Figura 7 – (a) Probabilidades do *random walk* sem ponderação; (b) Probabilidades do *random walk* com ponderação.

Na Figura 7(a) são mostradas as probabilidades referentes a cada célula sem a utilização de uma polarização. As probabilidades com polarização são mostradas na Figura 7(b). A polarização utilizada no trabalho tem intuito de acelerar a convergência, para isso foram propostas uma polarização para os líderes e outra para os demais robôs.

A polarização para os líderes possui o intuito de explorar o obstáculo que é percebido pelos sensores. Isso é feito ao aumentar as probabilidades do robô se movimentar para posições que não foram visitadas, ou foram visitadas uma quantidade menor de vezes.

A polarização dos robôs que não são líderes aumenta a probabilidade de movimentação para posições em que ocorrerá uma maior descontaminação do ambiente. Essa polarização é utilizada a fim de aumentar a exploração do ambiente e diminuir o tempo para a tarefa ser concluída.

4 Revisão bibliográfica

Neste capítulo está disposta uma Revisão bibliográfica sobre o problema da perseguição-evasão na robótica.

Para a formulação clássica, o evasor só é considerado como capturado apenas quando o perseguidor estiver em contato com ele e o evasor possui velocidades parecidas com o perseguidor [22]. Para essas situações podem ser utilizadas diversas formas de solução como a utilização de previsão de movimento do evasor [23, 41].

Para a formulação clássica, também são usados métodos de aprendizado por reforço como controlador do robô [42, 43, 44, 45, 46].

Com a formulação de perseguição-evasão em grafos, em [25] é realizada a comparação entre 4 algoritmos para diversos mapas e com dois tipos de visibilidade, podendo ser apenas local ou estendida e a velocidade do evasor pode ser apenas de um nó ou infinitos nós.

Uma das soluções propostas por [47] é a limpeza de um ambiente, em que esse é fornecido como entrada do sistema, assim como o modelo dos sensores utilizados nos robôs e o número desses para representar o ambiente em um grafo. Esse grafo é então partido em configurações topológicas para simplificação do ambiente. A partir disso, é então realizado o planejamento, e por fim, esse é refinado no espaço de configuração. O algoritmo é comparado com um planejador que realiza uma busca exaustiva, em que o algoritmo proposto possui um tempo de execução significativamente menor.

A formulação de perseguição-evasão com visibilidade pode ser utilizada com um único robô ou com múltiplos robôs, como mostrado por [48], cujo algoritmo inicia com um único agente realizando a tarefa. Caso mais agentes sejam necessários para completar a tarefa, eles são adicionados. Também é utilizada a geração de amostras proposta por [49] e um pós-processamento para melhorar o tempo de concluir a tarefa.

Em [50] é proposto um algoritmo que computa o menor caminho para que um robô capture evasores arbitrariamente rápidos em ambientes simplesmente conectados ou reporta que não é possível realizar a captura. Outra solução para perseguição-evasão em visibilidade foi proposta por [51], em que o evasor permanece parado até que o perseguidor tenha visão direta com o evasor. Isso é feito com a decomposição do mapa em células de visibilidade para achar a solução.

Foi proposto por [52] um planejamento de movimentação que maximiza a medida de limpeza do agente realizando a tarefa. O agente proposto possui um sensor omnidirecional sem limitação de distância, porém pode possuir um erro de posição

limitado por um limiar definido. Por conta desse erro de posição, são então propostas condições suficientes para que possua a robustez considerada como necessária para o trabalho.

A utilização de aprendizado de máquina no controlador do agente também é uma forma de solucionar o problema de perseguição-evasão com visibilidade. Em [53] é proposta uma abordagem baseada em aprendizado por reforço que considera todo o ambiente não visitado como contaminado, que vai sendo limpo à medida que os robôs se movimentam pelo ambiente. Ao se achar o evasor o mapa é então considerado limpo. Essa proposta porém, trata o perseguidor e o evasor com a mesma velocidade e não obtém uma melhora significativa do método clássico que foi comparado.

Em [54] os robôs cooperam a fim de capturar um evasor em uma superfície aquática com desvio de obstáculos. Para isso os perseguidores se movem para cercar o evasor e se movimentam para uma posição estimada a partir de uma predição da trajetória. A perseguição-evasão com utilização de drones é realizada em [55], que usa aprendizado por reforço profundo para controle de múltiplos drones com predição de movimento do evasor.

A utilização de múltiplos robôs para aplicações em perseguição-evasão em contexto de visibilidade pode ser utilizado com o objetivo de reduzir o impacto que a perda de um robô causa no sistema. Em [56], o sistema utiliza a informação atual da limpeza do mapa para fazer um replanejamento e aproveita a informação do robô perdido para acelerar a busca do novo planejamento. Já em [57], o planejamento é feito de forma que seja robusto à falha de um dos robôs, isto é, de uma forma que, caso não haja a falha, tenha mais agentes se movimentando do que o necessário, porém, na ocorrência da falha, o sistema já estará precavido, não precisando assim de um replanejamento.

Uma proposta para solucionar tarefas em que o evasor possui velocidade infinita é dada em [58], em que o time de perseguidores possui conhecimento do mapa previamente. Para isso, os robôs formam uma linha entre paredes em que os robôs avançam nessa linha para garantir que o evasor não está nas áreas em que os robôs passaram. Para a realização dessa solução, porém, é necessário que os robôs possuam conhecimento completo do mapa previamente para realizar o planejamento das trajetórias. A solução foi então simulada com drones para ambientes simples e ambientes reais.

Uma abordagem de visibilidade sem conhecimento do mapa é proposta por [16], em que os robôs possuem capacidades de comunicação e sensoriamento limitadas, além de uma abordagem com coordenação descentralizada. Nessa proposta, os robôs se dividem de acordo com as funções alocadas, sendo essas guardas de fronteira e seguidores. Os robôs que funcionam como seguidores seguem passivamente os guardas de fronteira. Esses, por sua vez, possuem a tarefa de avançar o processo de limpeza se não for causar uma recontaminação. Caso possa gerar recontaminação, é comunicado para o seguidor auxiliar o avanço da fronteira.

Em [59] é proposta a utilização de aprendizado por reforço para os perseguidores capturarem os evasores em ambientes dinâmicos e desconhecidos. Utilizam também o diagrama de Voronoi como forma de separar o ambiente e guiar os perseguidores para aumentar a exploração do ambiente a procura dos evasores.

Também foi proposta a utilização de algoritmo genético por [17], que propõe que os agentes podem se mover nas quatro direções de em uma grade discretizada e possuem controle baseado em um autômato finito, sendo o algoritmo genético responsável pelo mapeamento dos estados de ações. Para as células pouco visitadas no treinamento é utilizado um sorteio aleatório como uma abordagem complementar, substituindo assim o autômato.

Uma solução para o problema de perseguição-evasão de pior caso foi proposta por [15], que será utilizada como comparação deste trabalho, em que propõe a formação de linhas de uma parede a outra, como mostrado na Figura 8. Nessa abordagem, os robôs vizinhos à parede são os líderes da linha e têm comportamento de seguidor de parede, enquanto os centrais se comportam de modo a manter a conexão entre os líderes.

Como o mapa é desconhecido, existe a possibilidade do aumento da distância entre as paredes, o que causa um aumento da linha entre elas. Para isso, é necessário que hajam robôs disponíveis para suprir o aumento do tamanho da linha. Esses robôs são chamados de reserva e permanecem atrás de um dos líderes até serem requisitados. Quando requisitados eles se movimentam para fazer parte dos robôs centrais.

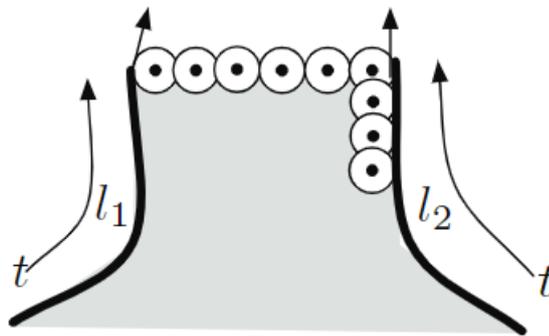


Figura 8 – Robôs realizando a limpeza do ambiente com uma formação em linha conectando as paredes [15].

Outro problema abordado por [15] na exploração em ambientes desconhecidos foi a busca por obstáculos quando a linha não possui mais agentes reserva, ou seja, chegou no tamanho máximo possível da linha. Essa busca têm o objetivo de achar novos caminhos que possam permitir que a tarefa seja completada. É mostrada na Figura 9 a sequência proposta, gerando a necessidade da separação da linha a fim de limpar o ambiente.

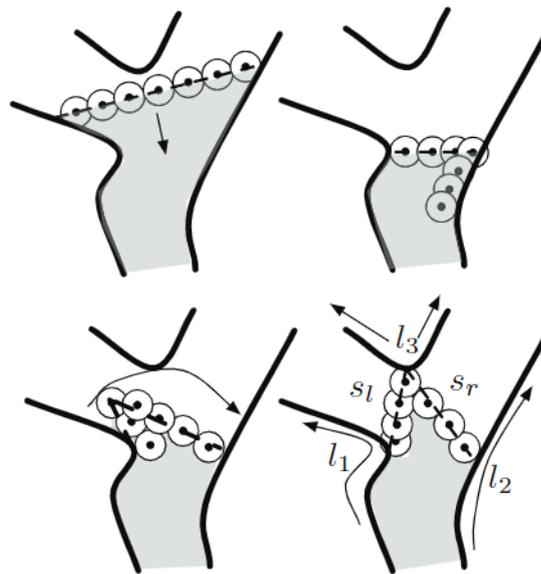


Figura 9 – Estratégia de procura por obstáculos ao não conseguir prosseguir com a formação em linha [15].

Na Figura 9 é mostrado o momento que a linha não consegue mais seguir as paredes designadas por chegar no tamanho máximo. Nesse momento, o sistema faz com que os robôs recuem de modo a ter agentes na reserva. Com a linha recuada, é então executado o procedimento de procura por obstáculo, em que os agentes são organizados em dois segmentos de linha e fazem uma varredura na parte contaminada. Caso seja achado um obstáculo, cada segmento de linha vira uma linha independente, caso não encontre um obstáculo pode-se mudar o robô que pertence aos dois segmentos ou recuar mais a linha para a obtenção de mais robôs reservas.

A utilização da *random walk* para a limpeza de ambientes simplesmente conectados foi proposta por [18], em que os agentes podem se movimentar nas quatro direções de uma grade discretizada, com a movimentação podendo ser polarizada na direção que irá limpar mais o ambiente ou não. Foi proposto ainda que os robôs formem um time, no qual os robôs centrais devem interligar os líderes, enquanto os líderes estão responsáveis por manter contato com a parede.

A solução proposta em [18], porém, é limitada para ambientes simplesmente conectados. Para que o ambiente seja considerado como simplesmente conectado, primeiramente ele precisa ser conectado, isto é, a partir de qualquer ponto dentro do ambiente que não seja obstáculo, o robô pode se movimentar para outro ponto qualquer que não seja obstáculo. Os ambientes conectados podem ser divididos em simplesmente e multiplamente conectados.

Ambientes simplesmente conectados são ambientes sem quaisquer obstáculos completamente cercados por áreas navegáveis, e podem ser vistos na Figura 10(a), enquanto os multiplamente conectados possuem obstáculos em seu interior,

representados na Figura 10(b).

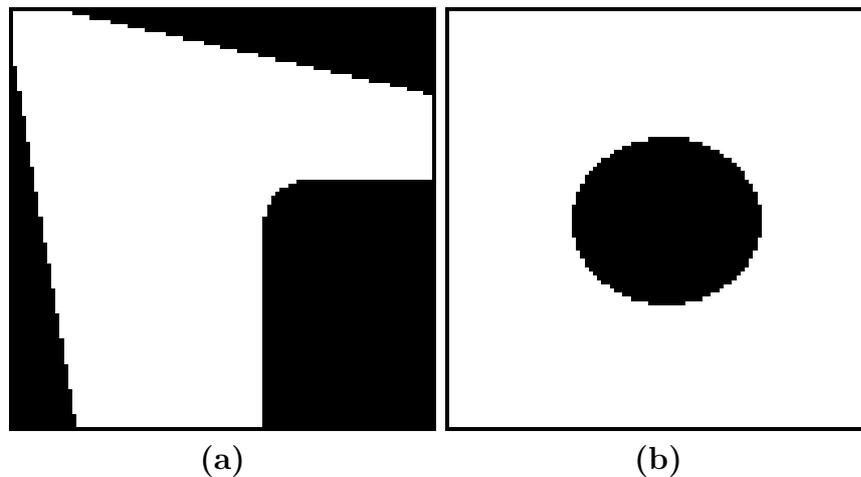


Figura 10 – Tipos de ambiente em que os obstáculos são representados em preto e em branco é a área que o robô pode navegar (a) Simplesmente conectado; (b) Multiplamente conectado.

Nas Figuras 10(a) e 10(b), os obstáculos são mostrados em preto, e a área que o robô pode se movimentar está em branco. Dessa forma, a Figura 10(b) possui um obstáculo cercado por área navegável e portanto, em [18], foram propostas restrições locais a fim de diminuir o número de iterações necessário para a completude do sistema. Essas restrições são separadas em físicas, dinâmicas e estratégicas. A restrição física impede que o robô se movimente para posições previamente ocupadas.

A restrição dinâmica limita a velocidade do robô. Por causa dessa restrição, a velocidade do robô é limitada para uma célula de distância por iteração.

As restrições estratégicas têm o propósito de organizar os robôs em uma linha para não permitir a passagem do evasor pela formação, como mostrado na Figura 11. Dessa forma, os dois robôs que estão posicionados nos limites da linha são chamados de líderes. Os líderes necessitam então, durante toda a duração da tarefa, serem capazes de detectar as paredes dos ambientes pelos sensores.

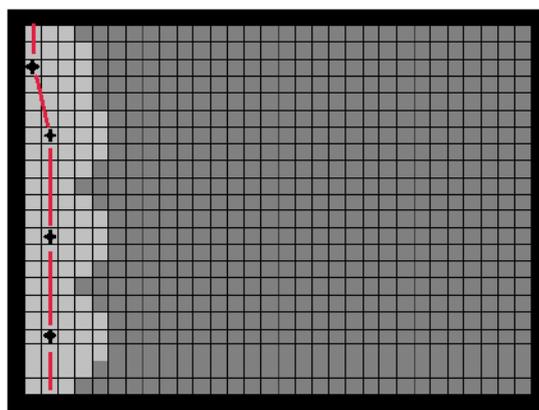


Figura 11 – Organização da linha de robôs pelas restrições estratégicas.

Esse método, porém, é limitado a ambientes simplesmente conectados devido a organização da linha, já que essa não é capaz de passar por obstáculos que estão dentro do ambiente.

Dessa forma, não foram encontradas técnicas que garantem que a tarefa de perseguição-evasão de pior case será completada para ambientes multiplamente conectados, foco dessa dissertação.

5 *Random walk* aplicado ao pior-caso de perseguição-evasão em ambientes multiplamente conectados

Nesta dissertação é apresentada uma técnica para resolução do problema de perseguição-evasão para o pior caso em ambientes conectados, com a demonstração que a tarefa é completada com a utilização da técnica de *random walk*. Para isso é feita a coordenação de múltiplos robôs em que cada robô possui um sensor limitado a uma distância r_{sensor} e é considerado um raio de captura $r_{captura}$ menor que r_{sensor} e caso exista um evasor dentro dessa área, ele é considerado como capturado.

Para o problema de pior caso, é considerado que o evasor é onisciente, sem limitação de velocidade e tem o objetivo de não ser capturado ou ficar o máximo de tempo possível sem ser capturado. Os perseguidores, por sua vez, possuem velocidade limitada, não possuem o conhecimento da posição do evasor nem do ambiente e têm o objetivo de capturar o evasor, que está dentro de um ambiente limitado.

A movimentação inspirada em *random walk* é utilizada devido à formalização matemática mostrada na seção 5.2.2 que garante de que o sistema irá finalizar a tarefa com um $t < \infty$. Para tanto, a movimentação inspirada em *random walk* é responsável por determinar qual será a movimentação do robô entre as possíveis posições.

A coordenação dos robôs é feita de forma que os robôs criem uma curva que conecta dois obstáculos entre si ou dois pontos de um mesmo obstáculo sem permitir a passagem de nenhum evasor entre os robôs, dessa forma, a região que contém o evasor pode ser chamada de região contaminada, e com o passar do tempo deve ser reduzida até o ponto em que não possui nenhuma região contaminada e a tarefa é concluída.

São também utilizadas restrições locais para que os robôs não possam quebrar a coordenação. Isso é feito ao limitar as posições que o robô pode se movimentar, como mostrada na seção a seguir.

5.1 Algoritmo proposto

O método proposto é baseado em *random walk* com restrições locais, de modo a não permitir a recontaminação de todo o espaço previamente limpo. Para isso, são escolhidos inicialmente dois robôs para serem os líderes, em que esses devem estar a uma distância máxima de $r_{captura}$ da parede, que é a distância em que os robôs conseguem

realizar a captura dos evasores. Os líderes são responsáveis por garantir que a formação consiga percorrer todas as paredes que limitam o ambiente sem que nenhum evasor consiga passar entre os robôs e as paredes.

A consideração de que os sensores possuem sensoriamento limitado para esta dissertação significa que nos ambientes onde são realizados os experimentos, um mesmo robô não consegue perceber obstáculos distintos ao mesmo tempo. Dessa forma, é necessário um mínimo de dois robôs para impedir que o evasor se movimente entre dois obstáculos. Para o ambiente, isso significa que a distância mínima entre dois obstáculos é de $2 \times r_{captura} + r$. Em que r é o tamanho da célula em que o mapa é discretizado.

Os robôs que não são líderes devem inicialmente garantir que não exista nenhuma passagem entre os dois líderes que não tenha cobertura do sensor de algum robô. Com essas restrições, [18] garante que ambientes simplesmente conectados são descontaminados (em $t < \infty$) se tiver uma quantidade de robôs capaz de realizar a tarefa.

Esse trabalho é então uma extensão do trabalho de [18] a fim de garantir que ambientes multiplamente conectados sejam capazes de serem descontaminados em um tempo finito. Para isso, é proposto que o time de robôs possa se separar ao perceber obstáculos no mapa. A separação é responsável por dividir o time anterior em dois, em que cada time possui dois líderes e funcionam como times independentes.

Essa separação faz com que um time de robôs descontamine uma das conexões do ambiente enquanto outro time descontamina o outro lado da bifurcação. Considerando ainda que um dos times funciona como uma fronteira móvel ou obstáculo móvel para os outros times, o ambiente se torna simplesmente conectado na visão de cada time, como pode ser visto na Figura 12.

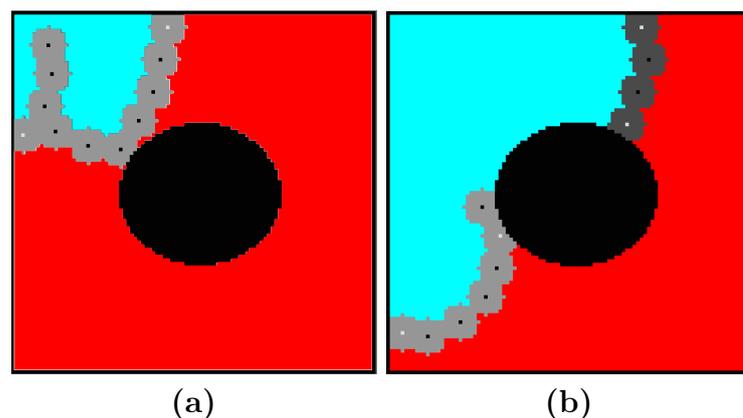


Figura 12 – Time sendo visto como barreira móvel, em que o vermelho é o ambiente contaminado, em azul é o ambiente descontaminado, em preto são os obstáculos, em cinza é a área dentro da região de descontaminação dos robôs, em branco são os líderes e cinza escuro os não líderes. (a) O time de robôs acha um obstáculo o que faz o ambiente multiplamente conectado; (b) O time em cinza mais escuro é visto como uma barreira móvel pelo outro time.

Na Figura 12(a) o time de robôs encontra um obstáculo, o que faz com que o

ambiente seja multiplamente conectado e na Figura 12(b) cada time considera o ambiente como simplesmente conectado, pois o outro time funciona como uma barreira móvel.

Dessa forma, os robôs estão limitados a se mover apenas para posições em que a recontaminação por quebra da formação não seja possível, ou seja, o ambiente não é recontaminado, com exceção para a recontaminação já esperada pela movimentação dos robôs na cena. Com essas condições, é mostrada na secção 5.2.2 a garantia que a tarefa será concluída em um $t < \infty$.

Para esta dissertação, são consideradas como células longe dos robôs as que estão a uma distância maior do que $r_{captura} + r$. Com isso, são permitidas apenas as contaminações de células perto dos robôs.

A movimentação do robô é realizada a partir da inspiração da *random walk*, em que o robô pode possuir a mesma probabilidade de se movimentar para cada célula possível, ou pode possuir probabilidades diferentes para cada célula possível. É denotado como movimentação insperada em *random walk* sem polarização quando ela possui a mesma probabilidade para cada posição possível e com polarização quando as probabilidades são diferentes.

A polarização da movimentação inspirada em *random walk* é realizada a fim de diminuir o número de iterações médias que o sistema necessita para completar a tarefa. Esta dissertação usa uma polarização para líderes e uma polarização para robôs que não são líderes baseadas em [18]. Também é testada uma nova polarização e analisado seu comportamento para os robôs não líderes.

A polarização para os líderes utilizada aumenta a probabilidade de movimentação dos líderes para células que foram visitadas em menor quantidade. Essa polarização cria uma tendência do líder se movimentar para locais não visitados previamente, a fim de aumentar a exploração dos obstáculos em que os líderes estão responsáveis.

A polarização dos não líderes tem intuito de aumentar a descontaminação das células, o que tendencia a movimentação do time para áreas contaminadas. A nova polarização aqui proposta se dá quando um robô não é líder e não possui nenhuma célula que possa descontaminar na próxima iteração. Essa polarização aumenta a probabilidade de movimentação para posições que causam um agrupamento dos robôs. As polarizações são discutidas na secção 5.1.1.

O sistema apresentado está implementado de forma descentralizada, em que os robôs compartilham as informações que possuem de posição e quais robôs estão dentro do alcance de comunicação para todos os robôs dentro desse alcance. Dessa forma, todos os robôs de um time possuem a posição dos demais e possuem comunicação entre si.

Times diferentes podem se comunicar contanto que um dos robôs do time esteja dentro do alcance de comunicação de um robô do outro time. Essa comunicação é

necessária para que possa ocorrer a junção dos times. A comunicação que é realizada para essa junção é feita quando os líderes responsáveis pelo mesmo obstáculo estão dentro do alcance de comunicação um do outro, então ambos os líderes verificam as condições para a junção.

É mostrada no fluxograma da Figura 13 uma iteração do sistema, em que cada robô sequencialmente planeja e realiza um movimento. O planejamento de movimentação do robô é dado pela verificação das posições em que ele pode se mover e em seguida é realizada a movimentação do robô.

Quando todos os robôs acabam de se movimentar, é realizada a verificação se precisa separar o time ou juntar dois times. Após essas verificações, acaba uma iteração do sistema e começa uma nova. As iterações são então realizadas até que a tarefa seja completada.

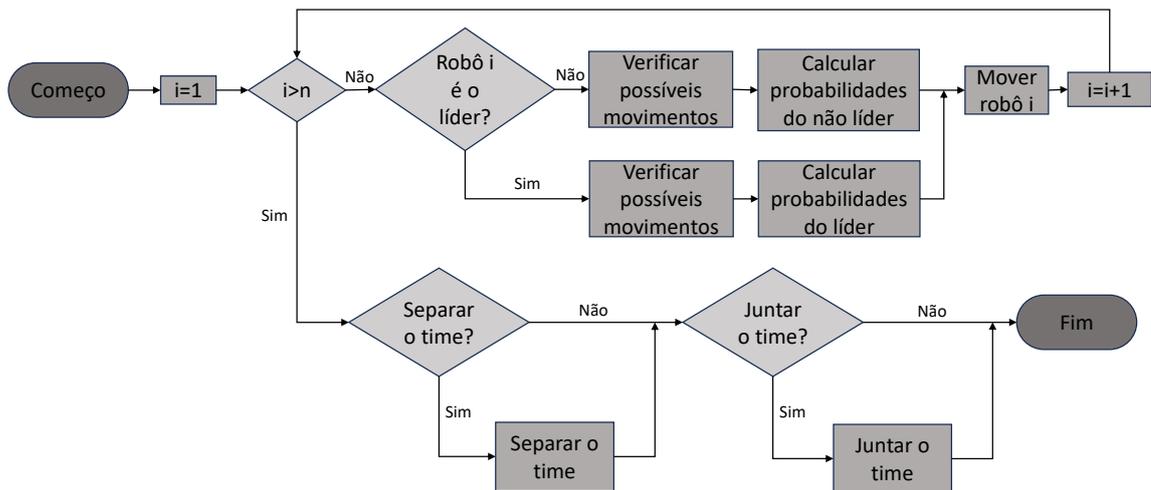


Figura 13 – Fluxograma de uma iteração do algoritmo proposto com cálculo de probabilidades de movimentação dos robôs, a movimentação desses e verificação de separação e junção.

No fluxograma da Figura 13 é mostrado que cada robô i realiza as ações sequencialmente até chegar no número de robôs n do sistema. Para cada robô, é inicialmente verificado se é um líder ou não, já que as polaridades utilizadas são diferentes.

Após isso, são verificadas para quais células que o robô pode se movimentar e calcula-se a probabilidade de movimento para cada célula possível. Com as probabilidades definidas, é então realizado o sorteio e o robô se movimenta para a célula sorteada.

Quando todos os robôs acabam de se movimentar, é feita a verificação

sequencialmente para a separação de times e junção de times. Caso alguma dessas verificações determinar que a ação correspondente precisa ser realizada, ela é realizada.

O Fluxograma mostrado na Figura 14 representa o comportamento do sistema do início ao final da tarefa.

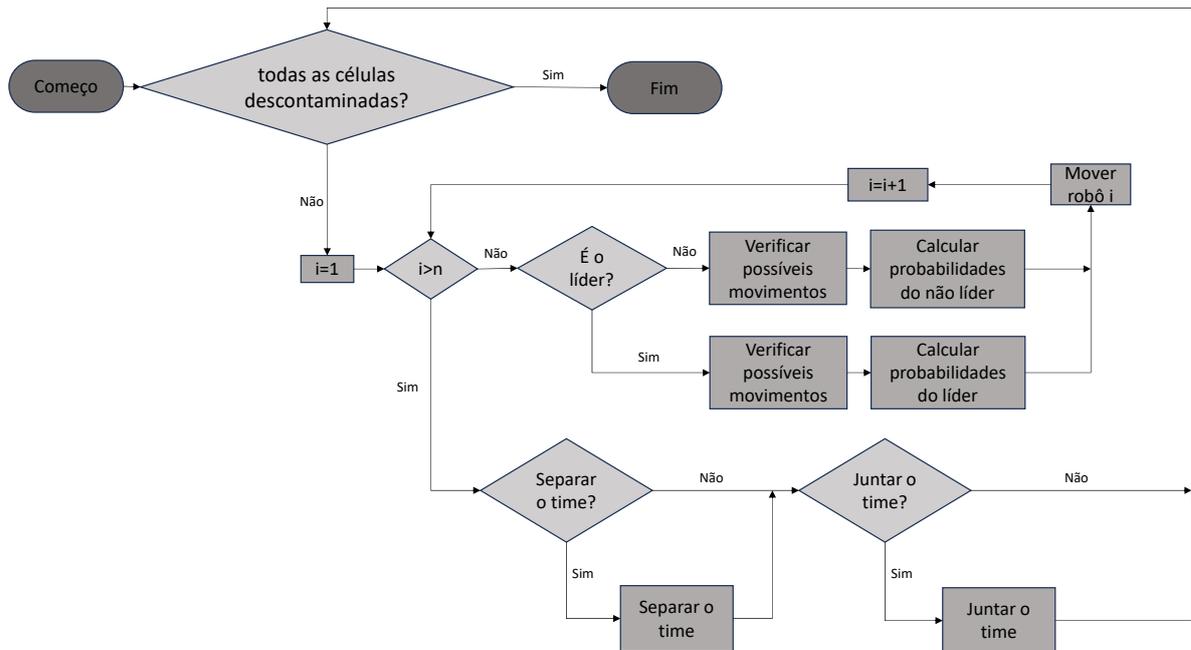


Figura 14 – Fluxograma até a finalização da tarefa do algoritmo proposto com cálculo de probabilidades de movimentação dos robôs, a movimentação desses e verificação de separação e junção.

5.1.1 Movimentação dos robôs

Para este trabalho, os robôs podem se movimentar nas 4 direções de Von Neumann ou ficar na mesma posição. São também utilizadas restrições locais propostas por [18] e apresentadas na seção 3.5.2, que não permitem a movimentação do robô para posições que causariam recontaminação.

Para que a solução seja capaz de solucionar ambientes multiplamente conectados, são então propostas condições em que o número de times podem mudar. Tais condições são apresentadas na seção 5.1.1.1.

Após as restrições limitarem as células para as quais o robô pode se movimentar, são calculadas as probabilidades de movimentação para cada célula, que podem ser utilizando polarização ou não. Esse cálculo de probabilidades para movimentação está descrito em 5.1.1.2.

5.1.1.1 Restrições locais

Existem restrições que são gerais para todos os robôs, e uma restrição que é específica para líderes. As restrições gerais são referentes a restrição física, de comunicação e sensoriamento, já que todos os robôs possuem o mesmo raio de comunicação e de sensoriamento. As restrições físicas são responsáveis por impedir a movimentação do robô para um espaço que já está previamente ocupado, seja por um obstáculo ou por outro robô.

As restrições de comunicação e sensoriamento são responsáveis por não permitir que os robôs quebrem a formação, permitindo então a recontaminação. A restrição de sensoriamento é para não permitir que a formação seja quebrada. Para isso a maior distância entre dois robôs vizinhos de forma a não quebrar a formação é $2 \times r_{captura}$.

A restrição de comunicação é necessária, pois os robôs que estão no mesmo time precisam se comunicar para obter as informações necessárias para tomar as decisões. Dessa forma, a distância máxima que pode ter entre dois robôs para que haja a comunicação entre eles é de $r_{comunicação}$.

Por isso, a distância máxima entre dois robôs vizinhos em um time é o menor valor entre $2 \times r_{captura}$ e $r_{comunicação}$. Dessa forma, os robôs não permitem a recontaminação de áreas previamente descontaminadas e não correm o risco de perder a comunicação dentro do time.

A restrição de comunicação também é responsável por manter todos os robôs de um mesmo time capazes de se comunicarem entre si. Essa comunicação pode ser de forma direta, quando a distância dos robôs é menor do que $r_{comunicação}$, ou de forma indireta, em que a informação passa por um ou mais robôs.

Como restrição específica, os líderes precisam estar a uma distância $r_{captura}$ do obstáculo, pois assim o evasor não conseguiria se movimentar entre a formação e os obstáculos.

5.1.1.2 Cálculo de probabilidades de movimento

Para a movimentação dos robôs, é utilizada uma estratégia baseada em *random walk*. A estratégia utilizada é mostrada na Figura 15, em que o robô i pode se movimentar nas 4 direções de Von Neumann. Dessa forma, as 4 direções de possível movimentação possuem uma probabilidade relacionada, que indica a probabilidade do robô se movimentar para a célula.

0%	$P(i, 1)$	0%
$P(i, 4)$	$P(i, 0)$	$P(i, 3)$
0%	$P(i, 2)$	0%

Figura 15 – Probabilidade de movimento para cada possível célula.

Essas probabilidades são determinadas a partir das restrições de cada robô e da polarização utilizada. As restrições que são utilizadas neste trabalho, podem impedir a movimentação do robô para determinada célula. Quando a movimentação não pode ser executada pelas restrições, então a probabilidade P do robô i se movimentar para a célula é igual a 0.

A probabilidade do robô permanecer na mesma posição é 1 se o robô não puder se movimentar para alguma célula, e é 0 se for possível de movimentar para pelo menos uma posição. É utilizada a probabilidade 0 do robô ficar parado se for possível de se movimentar para os robôs terem menos situações que ficam parados, o que aumenta a quantidade de movimentos, principalmente para posições de baixa probabilidade após o uso da polarização utilizada.

A polarização dos líderes é feita de forma que ele possua a tendência de se movimentar para locais que não foram visitados previamente ou foram visitados em uma menor quantidade a fim de explorar o ambiente [18]. Para isso, os robôs que são líderes precisam manter um histórico das células visitadas. Esse histórico necessita ser mantido apenas enquanto o robô é um líder.

Caso seja um robô que não era um líder e se tornou líder na separação, ele passa então a armazenar o histórico a partir do momento que se torna um líder, levando a uma maior exploração em torno do obstáculo.

Na mudança de um robô líder para um que não é líder, o histórico de posições não é mais necessário, uma vez que se o time de robôs passou pelo obstáculo, esse obstáculo está inteiramente dentro da área descontaminada, não sendo mais necessário que haja uma exploração em torno desse obstáculo.

A polarização dos robôs que não são líderes é realizada de forma diferente. Em [18] foi proposto que os robôs possuam uma tendência de se movimentar para posições em que ocorrerá a maior descontaminação de células. Neste trabalho será utilizada uma polarização baseada na de [18], com probabilidade do robô i se movimentar para a posição cel dada por

$$p(i, cel) = \frac{N_{cel} \cdot k_{pol1} + 1}{N_{total} \cdot k_{pol1} + n}, \quad (5.1)$$

em que N_{cel} é o número de células que o robô irá descontaminar ao se mover para a posição cel , N_{total} é a soma de todos os N_{cel} , n é a quantidade de posições que o robô pode se movimentar e k_{pol1} é o coeficiente de polarização que precisa ser um número real maior ou igual a 0, para que as probabilidades sejam positivas.

Como $p(i, cel)$ representa a probabilidade do robô i se movimentar para a posição cel , é necessário que o valor seja entre 0 e 1. Também é necessário que a soma de todas as probabilidades de movimentação do robô seja 1. Essas condições são satisfeitas, já que a soma de todos os N_{cel} é N_{total} .

Outra restrição para $p(i, cel)$ é que a probabilidade não pode ser 0 para posições em que o robô pode se mover. Essa restrição é devida a formalização da *random walk* em que o robô precisa permitir a movimentação em posições contrárias às células contaminadas.

É importante destacar que o número de células a serem descontaminadas para as possíveis direções (N_{cel}) podem possuir diferenças significativas, podendo inclusive não descontaminar nenhuma célula ao andar para uma direção e descontaminar o maior número possível em uma iteração para uma outra direção. Isso acarreta em uma diferença significativa entre as probabilidades de movimentação entre as células que aumenta a medida que k_{pol1} aumenta.

Como o desejado com a polarização é que o sistema tenha uma convergência mais rápida, então o coeficiente de polarização precisa ser maior do que 0, aumentando assim a probabilidade de movimentação para células que descontaminem uma quantidade maior do ambiente. No entanto, com o aumento do coeficiente, o valor de $p(i, cel)$ tende a N_{cel}/N_{total} , dificultando, assim, a movimentação em direções em que não geram descontaminação, e conseqüentemente, dificultando a solução de ambientes mais complexos.

Vale ressaltar que a polarização descrita na Equação 5.1 é utilizada quando, em alguma posição possível, tem-se pelo menos uma célula que será descontaminada. Quando não houver nenhuma célula para descontaminação, [18] propõe que todas as movimentações tenham a mesma probabilidade. Porém, em situações como mostrado na Figura 16, os robôs que estão dentro da área demarcada por 1 podem limitar a movimentação dos robôs dentro da área 2 ao se movimentar para cima. Essa restrição na movimentação dos robôs na área 2 é devido ao fato que, caso eles se movimentem para baixo, os robôs na área 1 não estariam em comunicação com o time, o que não pode acontecer considerando as restrições impostas.

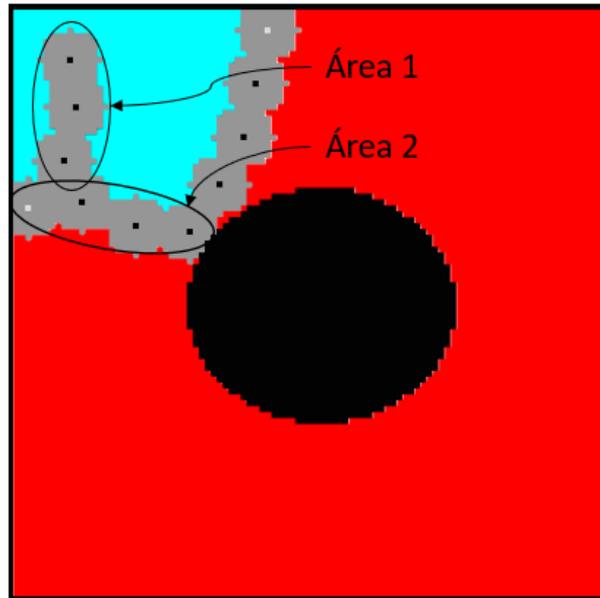


Figura 16 – Caso em que o robô não possui células contaminadas próximas e atrapalha a movimentação do time.

Esse comportamento também é visto em situações em que os times se separam por causa de algum obstáculo e um dos times consegue completar a parte do ambiente que ficou responsável. Após essa situação, é desejado que o time que completou a parte do ambiente se junte com o outro time e a polarização imposta nos líderes é capaz de fazer com que o time se movimente e se junte novamente.

Porém, nessa situação, os robôs que não são líderes teriam a mesma chance de movimentação para qualquer posição disponível já que não haveria células contaminadas para que a polarização influencie o movimento. Enquanto isso, as restrições de comunicação do time ainda são aplicadas, o que restringe a movimentação dos robôs.

Dessa forma, para que o time possua uma tendência de movimento para uma direção, todas as movimentações dos robôs teriam que ser na mesma direção. Esse comportamento causa, então, uma demora para que o time possa se movimentar e se juntar a outro, como em situações em que o time está em uma área completamente descontaminada, como mostrado na Figura 17.

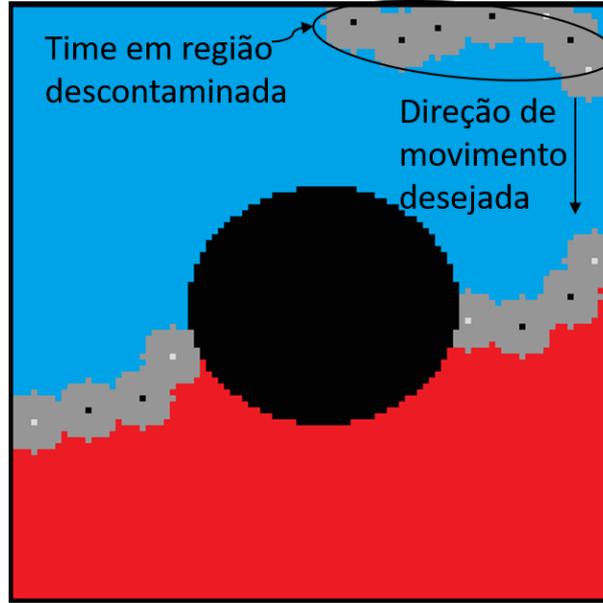


Figura 17 – Movimentação desejada de time em região descontaminada.

Por isso, é proposto que, em situações onde não tiver nenhuma célula que for descontaminada em nenhuma movimentação, que o robô tenha uma tendência a se aproximar dos robôs que estejam dentro do alcance de comunicação, para que não atrapalhe o movimento de algum robô. Com isso, é proposta a polarização

$$\begin{cases} p(i, cel) = \frac{d(i, cel) \cdot k_{pol2} + 1}{d_{total} \cdot k_{pol2} + n}, \\ d(i, cel) = \frac{N_r}{\sum^r \|p_r - p(i, cel)\|}, \end{cases} \quad (5.2)$$

em que $p(i, cel)$ é a possível posição que o robô i pode se mover, p_r é a posição do robô r dentre os robôs que estão dentro do alcance de comunicação do robô i e N_r é o número de robôs que estão dentro do alcance de comunicação. É então calculada a média das distâncias do robô i na posição cel para os outros robôs dentro do alcance de comunicação. O valor $d(i, cel)$ é então o inverso dessa média. É usado o valor inverso para que a probabilidade seja maior em direções em que a média da distância é menor.

O valor d_{total} é a soma de todos os valores $d(i, cel)$ para o robô i , n é a quantidade de posições que o robô pode se movimentar e k_{pol2} é o coeficiente de polarização com as mesmas características que k_{pol1} , porém referente à distância que o robô que está se movimentando irá ficar dos outros robôs dentro do alcance de comunicação. Valores de k_{pol2} maiores que 0 criam a tendência do robô se aproximar dos outros.

5.1.2 Separação e Junção dos times

Para que a solução seja capaz de descontaminar ambientes multiplamente conectados é necessário que o sistema seja sempre capaz de limitar a região que o evasor está.

Para isso, é proposto que o time possa se dividir de forma a funcionar como uma barreira móvel do ponto de vista do outro time. Isto pode ser entendido desta forma devido ao fato de que cada time irá impedir a recontaminação entre os obstáculos em que está posicionado. Então, do ponto de vista de cada time, eles estão responsáveis por descontaminar um ambiente simplesmente conectado, já que o outro time funciona como uma parede móvel.

Nesta dissertação optou-se por uma separação e junção dos times de forma determinística, e que é mostrado de forma simplificada como o sistema funciona na Figura 18. Os obstáculos estão sendo mostrados em preto, mas é importante ressaltar que a posição dos obstáculos não é conhecida pelos robôs.

Na Figura 18(a) o time encontra um obstáculo, fazendo com que o ambiente seja entendido como multiplamente conectado com as informações conhecidas pelo sistema, sendo necessário com que o time se separe, o que é mostrado na Figura 18(b).

Assim que os times passam pelo obstáculo o ambiente é novamente tratado como simplesmente conectado, e cada time continua sua movimentação como mostrado na Figura 18(c). Isso se dá devido ao fato que a bifurcação está em uma área descontaminada. Dessa forma, os times podem se juntar para que tenha o maior número de robôs ativamente descontaminando o ambiente. Esse comportamento é mostrado na Figura 18(d), em que o obstáculo foi contornado e não é mais necessário que os robôs se separem em dois times e por isso os times se juntam na Figura 18(e).

Um detalhamento melhor de como a separação se comporta é mostrado na Figura 19. Na Figura 19(a) é mostrado o momento em que um robô do time encontra o obstáculo. Nesse momento o time não pode se separar, já que apenas um dos robôs está em contato com o obstáculo.

Por isso, é necessário que dois robôs vizinhos estejam percebendo o obstáculo, o que é mostrado na Figura 19(b). Também é necessário que seja verificado se o ambiente entre os robôs e o obstáculo esteja descontaminado. Essa verificação é mostrada na Figura 19(c), em que os pontos verificados estão em amarelo. Por fim, na Figura 19(d) é mostrado que os times se separaram e continuam a tarefa.

A verificação das células pelo time é necessária para que, ao ocorrer a separação dos robôs, não exista uma área contida entre os robôs e o obstáculo onde o evasor poderia estar e contaminaria a parte limpa previamente. Para isso, é necessário que esta área seja verificada e completamente descontaminada antes que o time possa se separar.

Como os robôs não possuem conhecimento do ambiente para afirmar que uma área longe deles pode possuir evasores, é proposto que os robôs verifiquem as células que estão entre as distâncias de $r_{captura}$ até r_{sensor} deles à procura de evasores. Porém nem todas as células nessa distância precisam estar descontaminadas, já que podem existir células

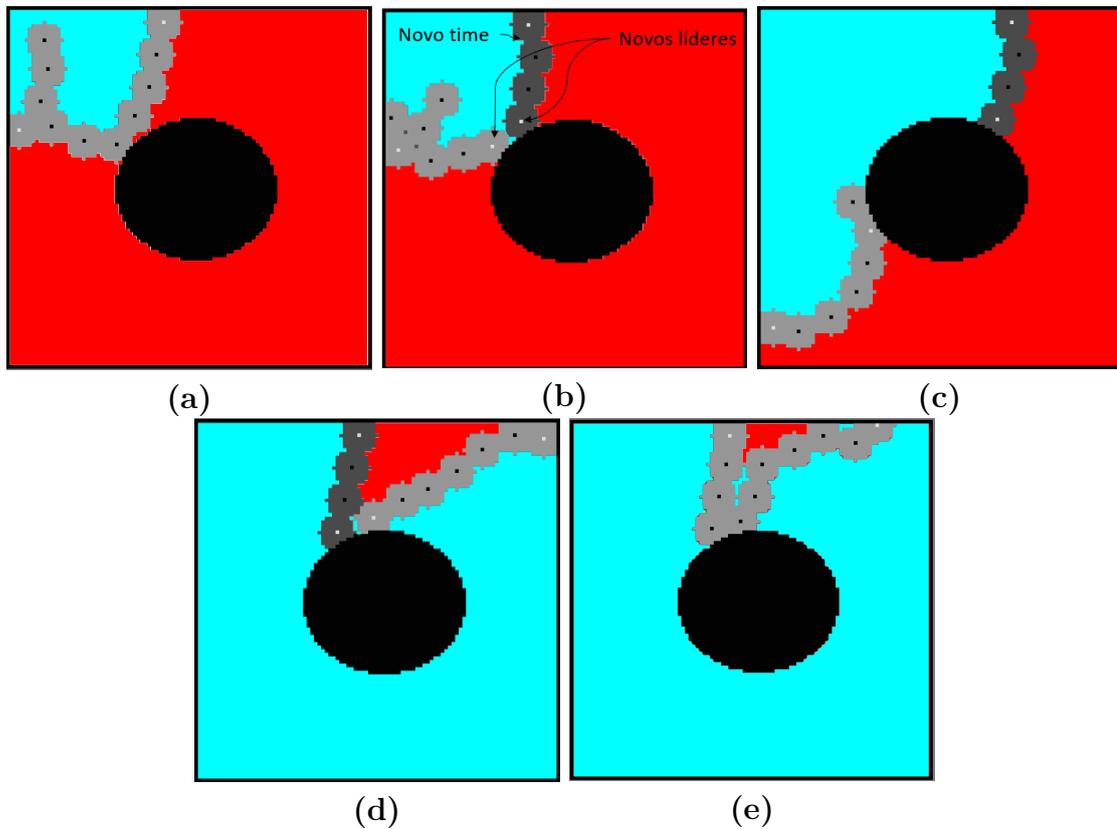


Figura 18 – Visão geral do método, em vermelho é o ambiente contaminado, em azul é o ambiente descontaminado, em preto são os obstáculos, em cinza é a área dentro da região de descontaminação robôs e em branco são os líderes e cinza escuro os não líderes. (a) O time de robôs acha um novo obstáculo; (b) O time se separa, com o novo time sendo mostrado em tonalidade diferente; (c) Os times continuam sua movimentação pelo mapa; (d) O time completa a descontaminação do obstáculo; (e) O time se junta.

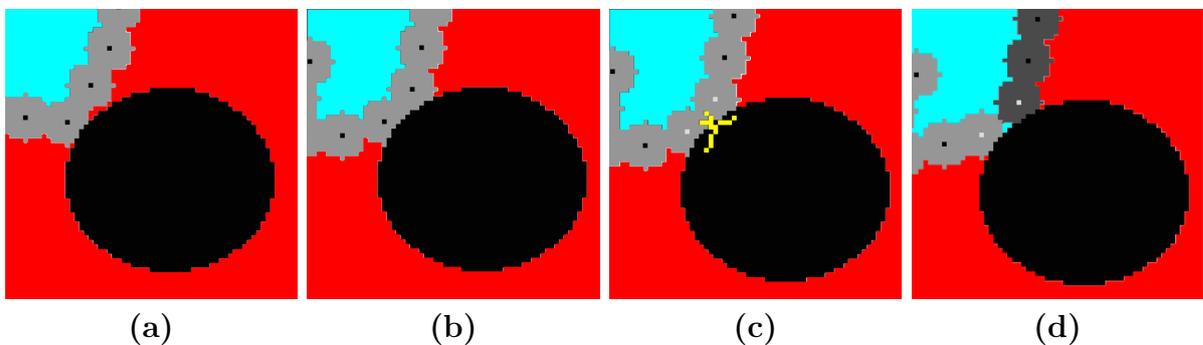


Figura 19 – Detalhes da separação do time. (a) Um robô encontra a parede; (b) Dois robôs encontram a mesma parede; (c) É feita a verificação das células em amarelo por contaminação entre os robôs e o obstáculo; (d) O time se separa em dois, com o novo time mostrado em tonalidade diferente e os times continuam a realização da tarefa.

contaminadas nessa distância que não causariam a recontaminação.

Na Figura 20 é mostrado com mais detalhes qual a área que tem que ser verificada, que corresponde às células que estão entre as linhas do robô até o obstáculo mais próximo e entre os robôs a uma distância de $r_{captura}$ até r_{sensor} . Essas linhas são criadas para os dois robôs e ambas as áreas precisam satisfazer a condição de que não possuam células contaminadas para que possa haver a separação dos times.

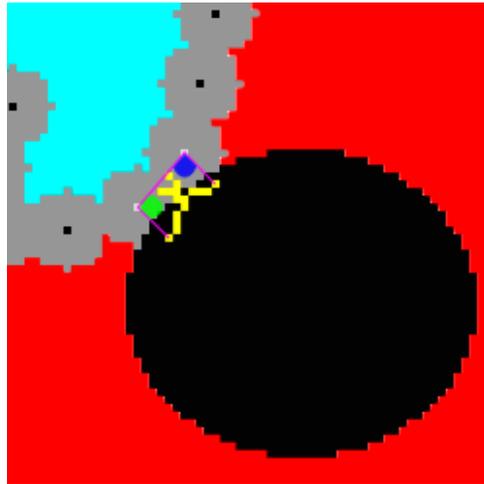


Figura 20 – Células a serem verificadas em amarelo, com os ângulos em verde e azul do obstáculo mais próximo do robô até o outro robô e as retas em magenta que ligam do robô até os limites dos ângulos.

Na Figura 20 são mostradas as linhas que delimitam essa área em magenta, fazendo com que as áreas que têm que ser verificadas sejam as áreas correspondente aos ângulos em verde para um robô e em azul para o outro.

Para a junção é realizado o mesmo procedimento realizado na separação, porém na ordem inversa, como pode ser visto na Figura 21.

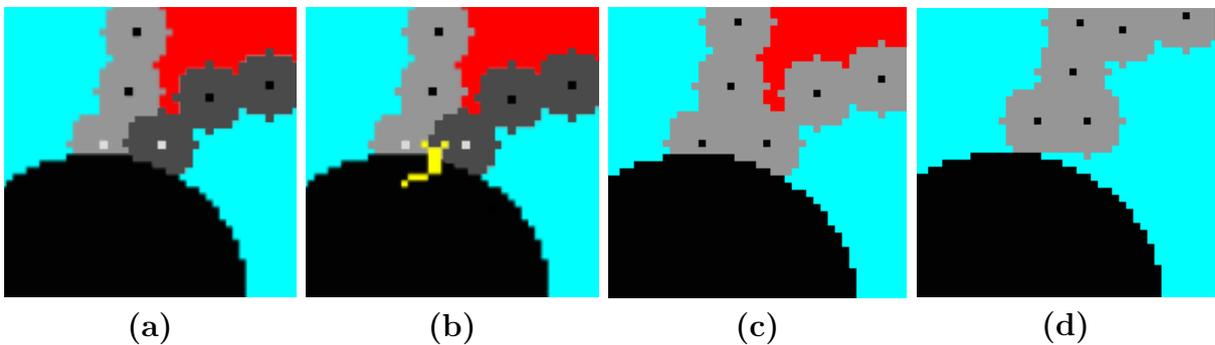


Figura 21 – Detalhes da junção do time. (a) Os dois líderes se aproximam, com os times mostrados em tonalidades diferentes; (b) É feita a verificação das células em amarelo por contaminação; (c) Os times se juntam em um time; (d) O time continua a descontaminação do ambiente.

5.2 Discussão sobre a completude do sistema

Nesta secção é discutida a completude do sistema, em que inicialmente é feita a discussão de forma intuitiva na secção 5.2.1 e após é feita a demonstração na secção 5.2.2. Em que para esta dissertação a completude do sistema é definida pela garantia do sistema de completar a tarefa em um tempo finito.

5.2.1 Noção intuitiva da completude do sistema

Um robô que tenha uma navegação inspirada em *random walk* consegue chegar no seu destino em um tempo finito se houver uma limitação no espaço em que ele pode se movimentar, esse espaço é denominado de espaço navegável. Dessa forma, o robô irá percorrer todos os pontos possíveis de serem alcançados, e como o destino é um ponto possível de ser alcançado a partir do ponto inicial, então o robô conseguirá chegar ao destino.

Essa limitação do espaço é dada ao se aplicar restrições de movimentação ou limitar a área do ambiente de interesse, são impostas barreiras de reflexão, de forma que não seja permitido que o robô se movimente em uma direção, mas seja permitida a movimento para as outras direções.

Ao se utilizar múltiplos robôs com movimentação inspirada em *random walk*, cada um deles percorrerá todo o espaço navegável em um tempo finito. Quando os robôs são utilizados para a tarefa de perseguição-evasão de pior caso, a solução da tarefa consiste em percorrer todo o ambiente sem permitir que a área limpa entre em contato com a área contaminada. Assim, movimentações dos robôs que permitam a recontaminação completa do ambiente não contribuem para a resolução do problema e, por isso, são descartadas, sendo consideradas como uma barreira de reflexão.

No entanto, embora sejam consideradas diversas barreiras de reflexão, o time de robôs continua movimentando-se de forma inspirada na *random walk*, o que os leva a percorrer todo o espaço navegável sem permitir recontaminação completa. Assim, caso exista uma combinação de trajetória que consiga limpar o ambiente, ela será realizada em um tempo finito.

Essa trajetória é possível de ser realizada mesmo com a utilização das barreiras, pois, com a movimentação inspirada em *random walk*, os robôs podem realizar movimentos que contaminam o ambiente, porém, não completamente, sendo possível a realização de movimentos no sentido de recontaminar todo o ambiente. Assim, se o sistema possui uma quantidade de robôs capaz de completar a tarefa a partir de um ambiente completamente contaminado, os robôs se movimentam em todas as possibilidades até que o ambiente esteja completamente descontaminado em um tempo finito, porém arbitrariamente grande.

Como forma de reduzir o tempo para a conclusão da tarefa, são utilizadas polarizações que aumentam a probabilidade do time de robôs navegar em direção às áreas contaminadas, mas sem impedir que o movimento em direção às áreas limpas seja também realizado, caracterizando-se ainda como movimentação inspirada na *random walk*.

5.2.2 Demonstração da completude do sistema

Seja W um conjunto aberto, limitado, discreto e conectado em que $W \subset \mathbb{Z}^2$ que representa o ambiente em que os robôs se movimentam. Como W é discreto e limitado, então o número de elementos de W é finito. Na Figura 22 pode ser vista uma representação do conjunto W para um mapa retangular discretizado de $m \times n$ células.

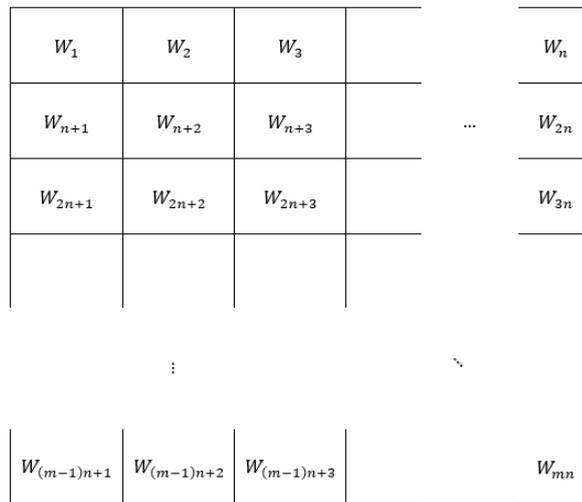


Figura 22 – Representação do conjunto W em que os robôs se movimentam.

Seja C_1 o espaço de configuração do robô 1, que indica as possibilidades de movimentação do robô 1 em W , considerando a ocupação de cada célula. Assumindo que cada robô pode se movimentar para uma das células imediatamente vizinhas ou permanecer parado, para cada elemento de W , C_1 denota 5 possibilidades de movimentação para cada posição. Os elementos de C_1 são definidos de forma que “0” denota célula livre e “1” uma célula que está ocupada por outro robô ou um obstáculo. As possibilidades de movimentação dadas por C_1 quando o robô 1 está na posição W_x são dadas por

$$C_1(W_x) = \{P_{parado}, P_{cima}, P_{baixo}, P_{esquerda}, P_{direita}\}. \tag{5.3}$$

Como C_1 é definido a partir de 5 possibilidades para cada elemento de W que é finito, então C_1 é finito e $C_1 \subset \mathbb{Z}^3$. Assim, considerando o espaço de configuração C_k

do k -ésimo robô, seja C o estado de configuração do sistema, dado pela composição dos estados de configuração de todos os N robôs.

Como cada estado de configuração dos robôs é finito, então C também é finito.

Para cada elemento de W , está associado um estado S_j que indica se a célula W_j está contaminada ($S_j = "1"$) ou não ($S_j = "0"$). Um exemplo dos estados S_j pode ser visto na Figura 23.

$S_1 = 1$	$S_2 = 0$	$S_3 = 0$		$S_n = 0$
$S_{n+1} = 1$	$S_{n+2} = 0$	$S_{n+3} = 0$...	$S_{2n} = 0$
$S_{2n+1} = 1$	$S_{2n+2} = 1$	$S_{2n+3} = 0$		$S_{3n} = 0$
	⋮		⋮	
$S_{(m-1)n+1} = 1$	$S_{(m-1)n+2} = 1$	$S_{(m-1)n+3} = 1$		$S_{mn} = 1$

Figura 23 – Estados S_j referentes a cada elemento do conjunto W .

Seja Ss_i o estado do sistema no tempo “ i ”, formado pela concatenação de todos os estados S_j de W no tempo i , assim,

$$Ss_i = \{S_1^i, S_2^i, S_3^i, \dots, S_{mn}^i\}. \tag{5.4}$$

Então, considerando cada possível posição do time de robôs em W , e a trajetória percorrida pelos robôs, tem-se um diferente estado do sistema Ss_i .

Assim, cada movimentação de um robô em W representa uma mudança no estado do sistema, associada à posição dos robôs e à contaminação de cada célula de W .

Seja Ss o conjunto com todos os possíveis estados do sistema Ss_i , que são possíveis de serem alcançados a partir do estado Ss_0 , que denota o estado que possui todas as células contaminadas. Como o número de elementos de W , as possibilidades de movimentação dos robôs e o número de robôs na cena são finitos, e cada célula de W tem apenas 2 estados possíveis (contaminado e não contaminado), então o número de elementos de Ss , embora grande, é finito.

Como o número de elementos de S_s é finito e a transição entre os estados se dá de forma sequencial, à medida que os robôs se movimentam na cena, então os estados do sistema e suas transições podem ser representados como uma cadeia de Markov.

Seja $S_{s_f} \in S_s$ o estado em que todas as células estão descontaminadas. Suponha que o sistema possui robôs suficientes para completar a tarefa, e que existe uma sequência de estados que leve de qualquer estado em S_s para o estado S_{s_f} , ou seja, que exista uma sequência de ações dos robôs que descontaminem toda a cena, então S_{s_f} é alcançável. Quando S_{s_f} é alcançado na cadeia de Markov, a tarefa é considerada concluída.

Na tarefa de perseguição-evasão de pior caso, basta que uma célula esteja contaminada para que todas as demais possam ser contaminadas novamente. Essa característica é devido ao fato de que caso haja uma célula contaminada ao lado de uma célula descontaminada, a célula descontaminada será contaminada.

Assim, a partir de qualquer estado de $S_s - \{S_{s_f}\}$, é possível alcançar os demais estados de $S_s - \{S_{s_f}\}$ desde que se tenha um número de robôs necessário para alcançá-los. Já que a partir de qualquer estado pode ocorrer uma recontaminação no sistema até que esteja completamente recontaminado e a partir desse estado é possível alcançar todos os estados de S_s .

Como todos os estados na cadeia $S_s - \{S_{s_f}\}$ são alcançáveis por todos os estados nela, essa cadeia é dita como irredutível [36].

O tempo necessário para que o sistema chegue um determinado estado y a partir de um estado x é expresso por τ . Então τ é uma variável aleatória e pode ter sua esperança calculada. Como o estado y é diferente do que o estado x , pode ser calculada a esperança apenas para o tempo positivo representado por τ^+ .

Dessa forma, de acordo com o Lema, a esperança de um estado x ser alcançado a partir de um estado y em um tempo τ é finito. Então, todos os estados $S_s - \{S_{s_f}\}$ são alcançáveis em um $t < \infty$, partindo-se de qualquer estado $S_s - \{S_{s_f}\}$.

Lema [36]. *Para quaisquer estados x e y de uma cadeia irredutível, $E_x(\tau_y^+) < \infty$.*

Como S_{s_f} é alcançável, existe ao menos um estado de $S_s - \{S_{s_f}\}$ com probabilidade não nula de levar o sistema ao estado S_{s_f} .

Então, S_{s_f} vai ser alcançado em um $t < \infty$.

Porém, o tempo médio para que o sistema alcance S_{s_f} pode ser muito grande, uma vez que existem diversas transições de estado que recontaminam todo o ambiente e, por isso, são indesejadas.

Assim, seja $C^r \subset C$ o espaço de configuração que representa quais movimentações (transições de estados) causariam a recontaminação de células em decorrência da quebra

da formação dos robôs. Seja $C^l = C - C^r$ o espaço de configurações dos robôs que não permitem a recontaminação de células em decorrência da quebra da formação dos robôs.

Considerando que após a movimentação dos robôs a partir de C^l , Ss_f se mantém e permanece alcançável, então, por repetição de argumentos prévios, Ss_f será alcançado em um $t < \infty$.

6 Resultados

Neste capítulo são apresentados a metodologia da organização dos experimentos e os resultados obtidos para o método proposto em ambiente simulado para análise do comportamento demonstrado. Inicialmente estão dispostas as características do sistema para as simulações e os mapas utilizados para os experimentos.

Estão dispostos em seguida os experimentos referentes às polarizações propostas para análise dos comportamentos resultantes e escolha dos valores para os experimentos seguintes. Após a escolha dos parâmetros de polarização, foi variado o número de robôs e comparados os resultados com os apresentados pelo método proposto em [15], por ser um método capaz de solucionar a tarefa em ambientes multiplamente conectados e com uma estratégia similar a proposta dos robôs formarem times para a descontaminação do ambiente.

6.1 Configuração dos experimentos

Para realizar os testes do método proposto e como se comportaria em um ambiente real, foi implementado um simulador para a tarefa de perseguição-evasão de pior caso, em que o simulador possui a tarefa de tratar como a área contaminada se comporta. Necessitando assim realizar a recontaminação de áreas limpas que são acessíveis para a contaminação e fornecendo os dados para os robôs que em situações reais seriam capturados por sensores.

O simulador foi implementado no Matlab, em que é carregado o ambiente e os robôs são inicializados todos na mesma célula ao lado de uma parede, a fim de simular a entrada dos robôs por uma porta, que é fechada quando os robôs acabam de entrar no ambiente. No começo do experimento, como os robôs não possuem conhecimento do mapa e não garantiram que nenhuma área foi descontaminada, então todo o mapa é inicializado como contaminado.

As imagens dos experimentos geradas pelo simulador usam o padrão de que áreas contaminadas estão destacadas em vermelho e áreas descontaminadas em azul, como mostrada na Figura 24. A área em cinza é o raio do sensor do robô, em que a área é descontaminada, os robôs são representados por pontos em preto para os não-líderes e branco para os líderes. Os obstáculos são representados em preto.

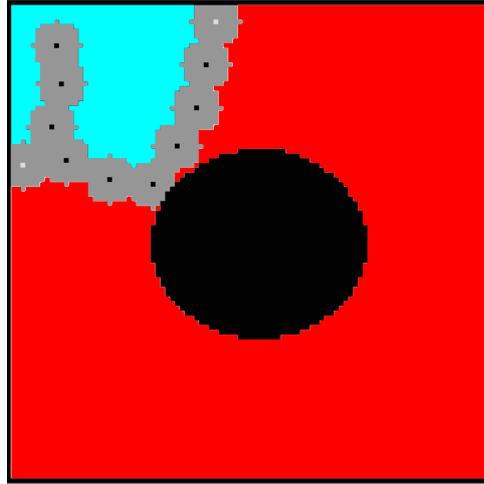


Figura 24 – Ambiente de simulação.

A cada iteração da simulação, as posições dos robôs são atualizadas juntamente com as áreas que estão dentro do raio de sensor do robô. Após isso, é realizada uma verificação se alguma célula da região contaminada está em contato com alguma célula descontaminada e, caso esteja, é considerada como contaminada. O comportamento da simulação está representado na Figura 25.

O simulador utilizado, como mostrado na Figura 25, possui duas condições de finalização: o número de iterações m ultrapassa o número máximo de iterações (n_{max}) escolhido ou todas as células do ambiente estão descontaminadas.

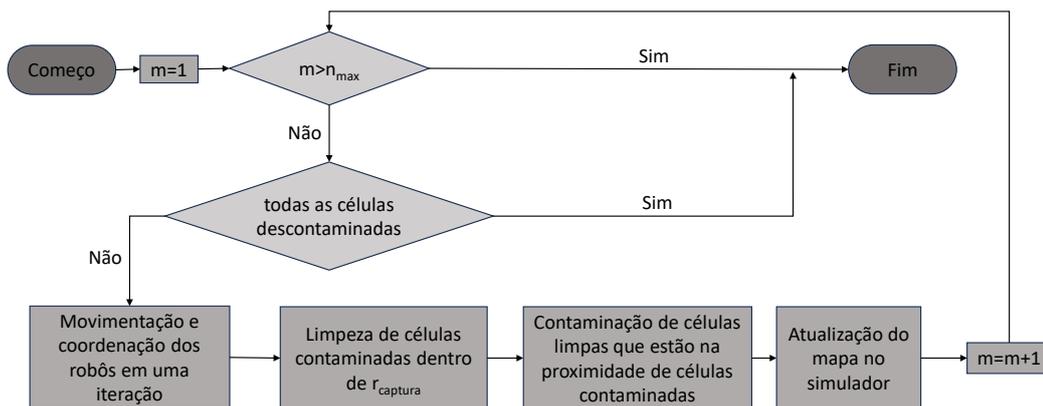


Figura 25 – Fluxograma da simulação detalhando o processo descontaminação e contaminação do ambiente, bem como as condições de parada da simulação.

Para cada iteração, é verificado se uma dessas condições foi cumprida. Caso ambas as condições sejam falsas, é executado o sistema proposto com a simulação dos

robôs. Após todos os robôs se movimentarem, o simulador contamina as células que estão descontaminadas e são vizinhas a alguma célula que está contaminada.

Devido ao fato que a abordagem proposta é dependente de componentes aleatórios, o número de iterações pode variar entre testes para o mesmo experimento. Dessa forma, cada experimento foi repetido 10 vezes.

É importante destacar que como a solução tem a garantia de completude para um tempo finito, mas arbitrariamente grande, foi definido um limite máximo de 10 mil iterações para os testes realizados. O limite escolhido de 10 mil iterações foi escolhido de forma que o sistema possua iterações suficientes para que consiga completar a solução do problema.

Para a realização dos experimentos, é necessário determinar a posição inicial dos robôs. Para a simulação da entrada dos robôs no ambiente, os robôs são inicializados a duas células do limite superior do mapa e na célula vizinha da lateral. Dessa forma, os robôs entram no ambiente e após todos os robôs entrarem, a passagem é fechada.

Os robôs foram definidos de forma que ocupam completamente a célula em que estão posicionados, não permitindo que uma mesma célula seja ocupada por dois robôs ou por um robô e um obstáculo. O $r_{captura}$ foi definido como $5 \times r$, o r_{sensor} foi definido como $r_{captura} + r$ e $r_{comunicação}$ como $2 \times r_{captura}$ de forma que quando os robôs estiverem em comunicação não é permitida a passagem de evasores entre eles. Em que r é o tamanho da célula.

Para esta dissertação são usados 4 mapas diferentes para as análises de todas as influências discutidas. Os mapas são mostrados na Figura 26.

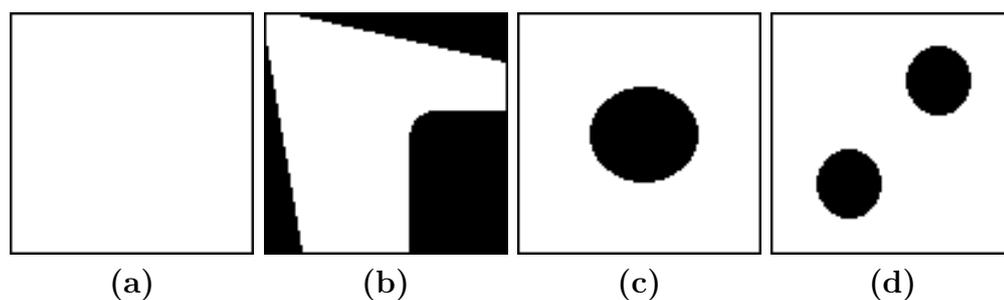


Figura 26 – Mapas utilizados (a) Mapa 1; (b) Mapa 2; (c) Mapa 3; (d) Mapa 4.

Os mapas foram selecionados de forma que os dois primeiros são referentes a ambientes simplesmente conectados e os outros dois a ambientes multiplamente conectados, sendo que todos os mapas possuem o mesmo tamanho de 100x100 células.

O primeiro mapa é um ambiente conectado mais simples enquanto o segundo é um ambiente em que as paredes são posicionadas de forma a propiciar uma maior chance de que o time possa se separar, pois seria tratado como um obstáculo diferente já que os líderes tenderiam a demorar mais para que todo o mapa fosse percebido como

simplesmente conectado. Por isso, o time perceberia esse ambiente simplesmente conectado como um ambiente multiplamente conectado e seria realizada a separação do time e os times completariam a tarefa, mostrando que mesmo que os robôs se separem em ambientes simplesmente conectados, conseguem completar a tarefa.

Dos dois mapas multiplamente conectados, um é um mapa simples com apenas um obstáculo com o intuito de mostrar o impacto da divisão do time no sistema, e o último mapa possui dois obstáculos, de forma que o time necessita de pelo menos duas separações, podendo ter vários times ao mesmo tempo.

6.2 Análise das polarizações propostas

O primeiro experimento foi realizado para verificar a influência da primeira polarização, que cria uma tendência a se movimentar para posições que geram uma maior descontaminação do ambiente. Para esse experimento a segunda polarização não foi utilizada, em que essa polarização tem o intuito de aproximar os robôs que não possuem contato com células contaminadas dos demais robôs. Foram utilizados valores de polarização entre 0 e 40 para os quatro mapas descritos, e na Tabela 1 é mostrada a taxa de experimentos que concluíram a tarefa em menos de 10 mil iterações.

	Mapa 1	Mapa 2	Mapa 3	Mapa 4
$k_{pol1} = 0$	0%	0%	0%	0%
$k_{pol1} = 0.2$	100%	100%	70%	100%
$k_{pol1} = 0.4$	100%	100%	70%	100%
$k_{pol1} = 0.6$	100%	100%	30%	100%
$k_{pol1} = 0.8$	100%	100%	90%	100%
$k_{pol1} = 1$	100%	100%	40%	60%
$k_{pol1} = 6$	100%	90%	70%	50%
$k_{pol1} = 11$	100%	100%	40%	40%
$k_{pol1} = 20$	100%	100%	70%	80%
$k_{pol1} = 30$	100%	100%	30%	80%
$k_{pol1} = 40$	100%	100%	40%	100%

Tabela 1 – Taxa de completude para as polarizações e mapas propostos.

Como mostrado na Tabela 1, para valores de polarização maiores do que 0, o método conseguiu completar a tarefa ao menos em um teste dentro do limite de 10 mil iterações. Para os casos que não foram completados é devido a um time estar dentro de uma região descontaminada e não se movimentar de encontro aos outros times dentro do limite estipulado.

Dessa forma, o método ainda é válido, já que pela *random walk* esse time terá uma movimentação de encontro ao outro e concluirá a tarefa, precisando apenas de um tempo maior para que a tarefa seja completada.

Na Tabela 2 são mostradas a média e o desvio padrão para a quantidade de iterações para cada mapa e valores de polarização escolhidos. É importante destacar que os testes em que o número de iterações passou de 10 mil não foram utilizados para a média e desvio padrão.

	Mapa 1	Mapa 2	Mapa 3	Mapa 4
$k_{pol1} = 0,2$	$3635 \pm 573,03$	$2243,3 \pm 242,89$	$7439,6 \pm 1236,1$	$6432,7 \pm 904$
$k_{pol1} = 0,4$	$2967,3 \pm 1028,4$	$1841,6 \pm 389,82$	$5480,3 \pm 1252,9$	$5028,7 \pm 1151,2$
$k_{pol1} = 0,6$	$2436,1 \pm 448,58$	$1511,2 \pm 251,47$	$4586 \pm 966,3$	$4016,5 \pm 821,17$
$k_{pol1} = 0,8$	$2004 \pm 540,41$	$1553,4 \pm 321,42$	$1553,4 \pm 321,42$	$4035,5 \pm 752,25$
$k_{pol1} = 1$	$2595,7 \pm 298,8793$	$1905,6 \pm 745,53$	$3499,8 \pm 723,59$	$4977,3 \pm 1894,8$
$k_{pol1} = 6$	$1720,1 \pm 495,3858$	$1038,6 \pm 348,35$	$2983,4 \pm 2460,3$	$2803,2 \pm 826,2$
$k_{pol1} = 11$	$1557,4 \pm 363,0769$	$958,6 \pm 236,89$	$3979,5 \pm 2286,8$	$3986 \pm 1764,1$
$k_{pol1} = 20$	$1698,5 \pm 737,56$	$1445,8 \pm 977,64$	$3849,3 \pm 2863,5$	$3067,3 \pm 2097,4$
$k_{pol1} = 30$	$1636 \pm 241,1$	$926,4 \pm 264,1$	$4242,3 \pm 2453,6$	$2437,4 \pm 601,96$
$k_{pol1} = 40$	$1592,2 \pm 419,8$	$1022 \pm 382,58$	$3984,8 \pm 1482,8$	$2741,5 \pm 1675,2$

Tabela 2 – Média e desvio padrão da quantidade de iterações.

Com esses resultados, é possível observar que o sistema é capaz de completar a tarefa com as configurações utilizadas dentro do limite estipulado. É possível observar também que mapas com maior complexidade necessitam de um número maior de iterações, devido à necessidade de garantia de não recontaminação de células longe dos robôs. Isso pode ser visto também na Figura 27.

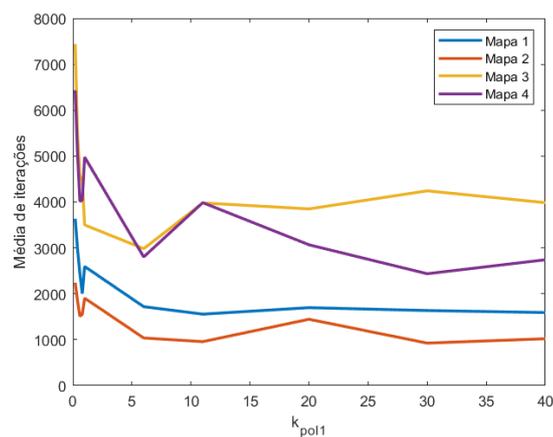


Figura 27 – Média de iterações para valores de k_{pol1} em cada mapa.

É mostrado na Figura 27 a média de iterações para os valores de k_{pol1} testados em cada mapa, e é possível observar que a média de iterações é alta para pequenos valores de k_{pol1} . Esse aumento de iterações com a diminuição do valor de polarização complementa o resultado que sem a polarização, o sistema não foi capaz de concluir a tarefa em nenhum mapa.

Na Figura 27 pode ser visto que o aumento da polarização tende a estabilizar já que as probabilidades de movimento convergem para N_{cel}/N_{total} , como discutido na seção 5.1.1.2.

Também é possível observar que com o aumento da polarização, mapas mais simples como o mapa 1 e mapa 2, tiveram uma média menor de iterações necessárias. Entretanto, para mapas mais complexos, como o mapa 3 e mapa 4, com o aumento da polarização para $k_{pol1} = 6$ foi obtida uma diminuição do número de iterações, enquanto para o valor de $k_{pol1} = 11$ obteve-se um aumento.

Esse aumento da polarização pode ser devido ao fato dos robôs que não são líderes terem a tendência de explorar mais o ambiente, o que causa que esses robôs percebam obstáculos no mapa. Esses obstáculos fazem com que haja a separação do time. Portanto, o aumento da polarização cria uma tendência de que o time de robôs se separe mais.

Esse número maior de separações cria a necessidade de que os times se juntem posteriormente para a concluir a tarefa, dessa forma é necessário que haja um maior número de iterações para que essa junção seja realizada.

Esses aumentos, porém, não são estatisticamente significativos já que os valores de desvio padrão obtidos são altos comparados com a diferença entre as médias obtidas. Dessa forma, pode ser entendido que possui realmente uma estabilização da média de iterações para valores de $k_{pol1} > 6$ para os resultados obtidos.

Na Figura 28 são mostradas as evoluções da descontaminação pelas iterações do melhor resultado para as polarizações analisadas. Nessa figura, é possível perceber que para o teste com menor número de iterações de cada polarização, o valor de $k_{pol1} = 0,2$ obteve resultados piores para todos os mapas, necessitando inclusive de mais do que o triplo de iterações em relação aos melhores valores.

Também é possível perceber que o comportamento do sistema é similar para todos os valores de polarização, de forma que a relação entre as células contaminadas e iteração do sistema não é monotonicamente decrescente. Isso é devido a utilização da *random walk* para movimentação em que os robôs precisam ser capazes de se movimentar para uma direção que irá contaminar um número de células a fim de concluir a tarefa.

Por essas razões, foi escolhido o valor de $k_{pol1} = 6$, já que o número médio de iterações possui uma estabilização acima desse valor. Também apresentou a menor média de iterações quando analisados todos mapas. Na Figura 29 são mostrados 4 instantes de um teste realizado para o valor de $k_{pol1} = 6$.

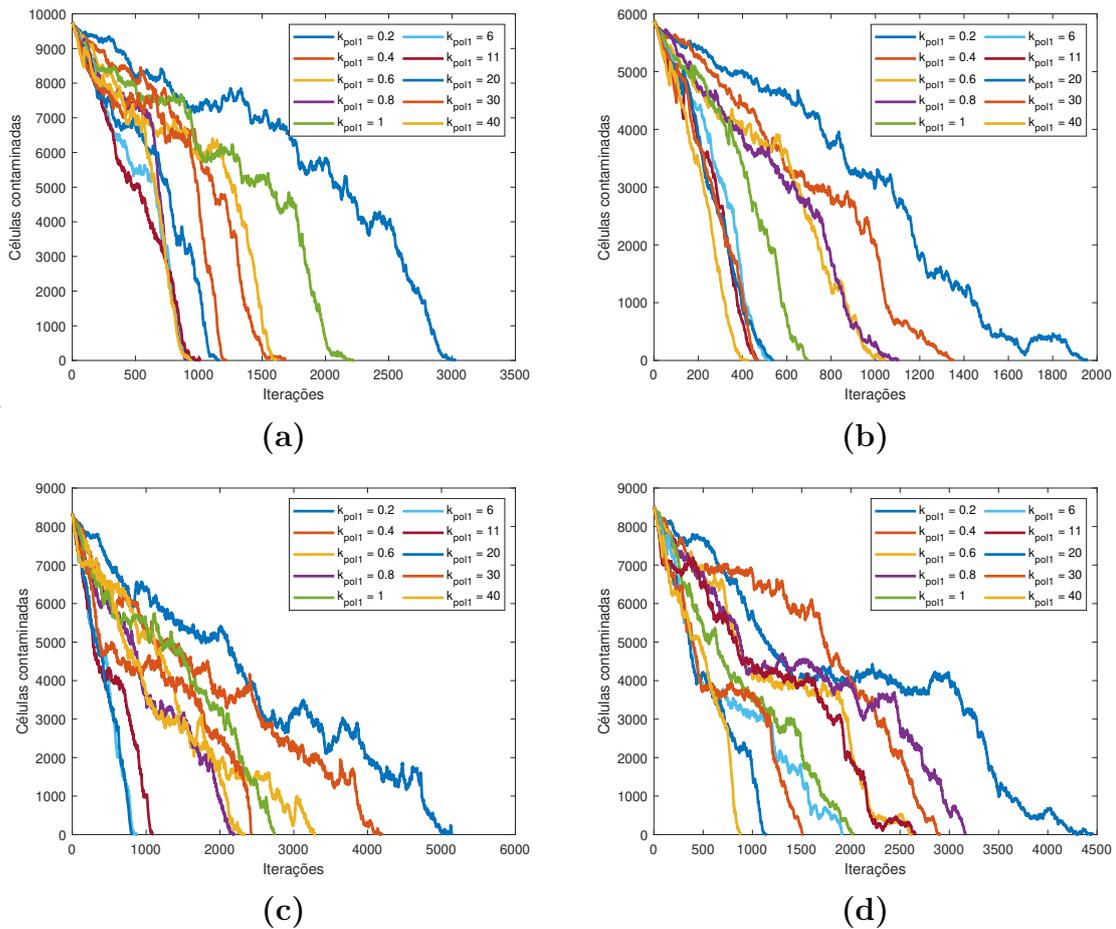


Figura 28 – Evolução da descontaminação pelas iterações para o teste com menor iterações das polarizações analisadas para os mapas: (a) Mapa 1; (b) Mapa 2; (c) Mapa 3; (d) Mapa 4.

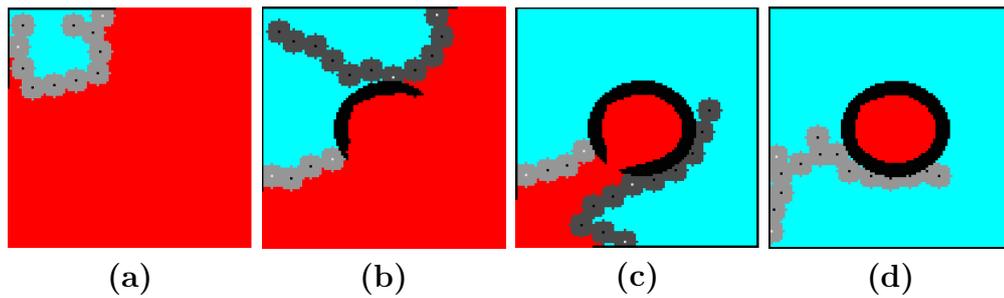


Figura 29 – Teste realizado no mapa 3, com os obstáculos já percebidos pelos robôs em preto, em azul a área descontaminada, em vermelho a área contaminada e em cinza (com tonalidade diferente para os dois times) a área de captura dos robôs para as seguintes iterações: (a) iteração 150; (b) iteração 500; (c) iteração 2900; (d) iteração 3203.

Com o valor de k_{pol1} escolhido, foi então realizado o segundo experimento de polarização para analisar o comportamento da segunda polarização. Para isso foi testado o valor de $k_{pol2} = 10$ a fim de verificar se há uma melhora no comportamento. Na Tabela 3 são mostradas as taxas de completude para os mapas propostos.

	Mapa 1	Mapa 2	Mapa 3	Mapa 4
$k_{pol2} = 0$	100%	90%	70%	50%
$k_{pol2} = 10$	100%	100%	30%	60%

Tabela 3 – Taxa de completude para os mapas propostos.

Na Tabela 3 são mostradas as taxas de completude com a segunda polarização proposta. Pode ser visto nessa tabela que os mapas 1, 2 e 4 possuem taxas semelhantes, porém o mapa 3 teve uma taxa menor do que sem a polarização.

Na Tabela 4 são mostrados os valores de média e desvio padrão para a segunda polarização.

	Mapa 1	Mapa 2	Mapa 3	Mapa 4
$k_{pol2} = 0$	1720,1 ± 495,3858	1038,6 ± 348,35	2983,4 ± 2460,3	2803,2 ± 826,2
$k_{pol2} = 10$	1928,4 ± 419,14	2076,8 ± 1756,3	5187,3 ± 2808,2	3338,3 ± 796,67

Tabela 4 – Média e desvio padrão da quantidade de iterações.

Na Tabela 4 pode ser visto que a quantidade de iterações necessárias utilizando o valor de $k_{pol2} = 10$ foi maior do que para o valor de $k_{pol2} = 0$, em todos os mapas.

Esse aumento é devido à forma que a polarização foi proposta, pois os robôs criam uma tendência a se agruparem com os outros robôs, quando não há células contaminadas na vizinhança.

Porém, com essa polarização os robôs que não possuem células contaminadas perto podem se agrupar dentro de uma área descontaminada. Esse caso específico pode ser notado na Figura 30, em que o robô analisado está em uma área sem células contaminadas.

Nesse caso, a polarização cria uma tendência de movimento do robô para a célula superior. Pois, esse movimento irá diminuir a distância para os outros robôs que estão no raio de comunicação.

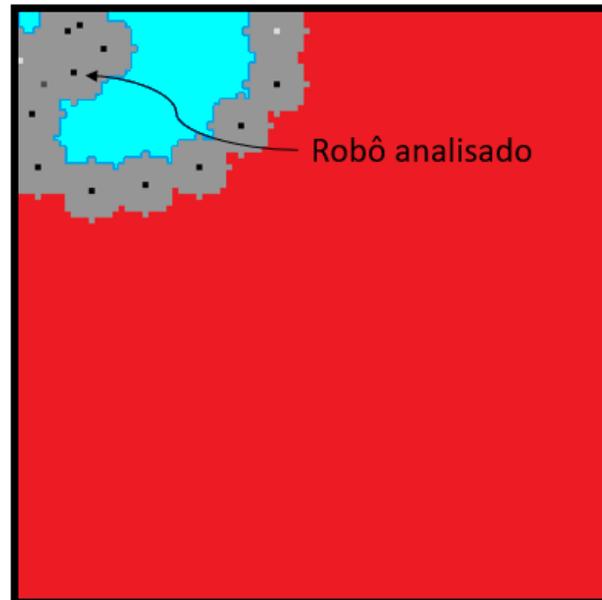


Figura 30 – Exemplo em que a polaridade atrapalha a movimentação desejada.

A evolução da descontaminação para a segunda polarização pode ser vista na Figura 31, em que apenas no mapa 2 o valor de $k_{pol2} = 10$ obteve um resultado melhor e um comportamento similar do que para o valor de $k_{pol2} = 0$.

Por essas razões, foi utilizada a segunda polarização com valor de $k_{pol2} = 0$ para os experimentos com números diferentes de robôs.

6.3 Comparação com vários números de robôs

Com os valores de polarização determinados, é possível verificar o impacto da variação do número de robôs na taxa de completude da tarefa. É importante lembrar que o número de iterações está limitado em 10 mil.

Esses resultados também são comparados com o método em [15], já que esse método é capaz de resolver a tarefa em ambientes tanto simplesmente conectado quanto multiplamente conectado, a consideração de limitação dos sensores serem similares com este trabalho e estruturar a formação dos robôs de forma similar a este trabalho.

Para a realização dessa análise, foi variado o número de robôs, e as taxas de completude para os números de robôs escolhidos são mostrados na Tabela 5.

	Mapa 1	Mapa 2	Mapa 3	Mapa 4
11 robôs	0%	100%	90%	20%
13 robôs	100%	100%	60%	30%
15 robôs	100%	90%	70%	50%
20 robôs	100%	100%	80%	100%

Tabela 5 – Taxa de completude para a quantidade de robôs e mapas escolhidos para o método proposto.

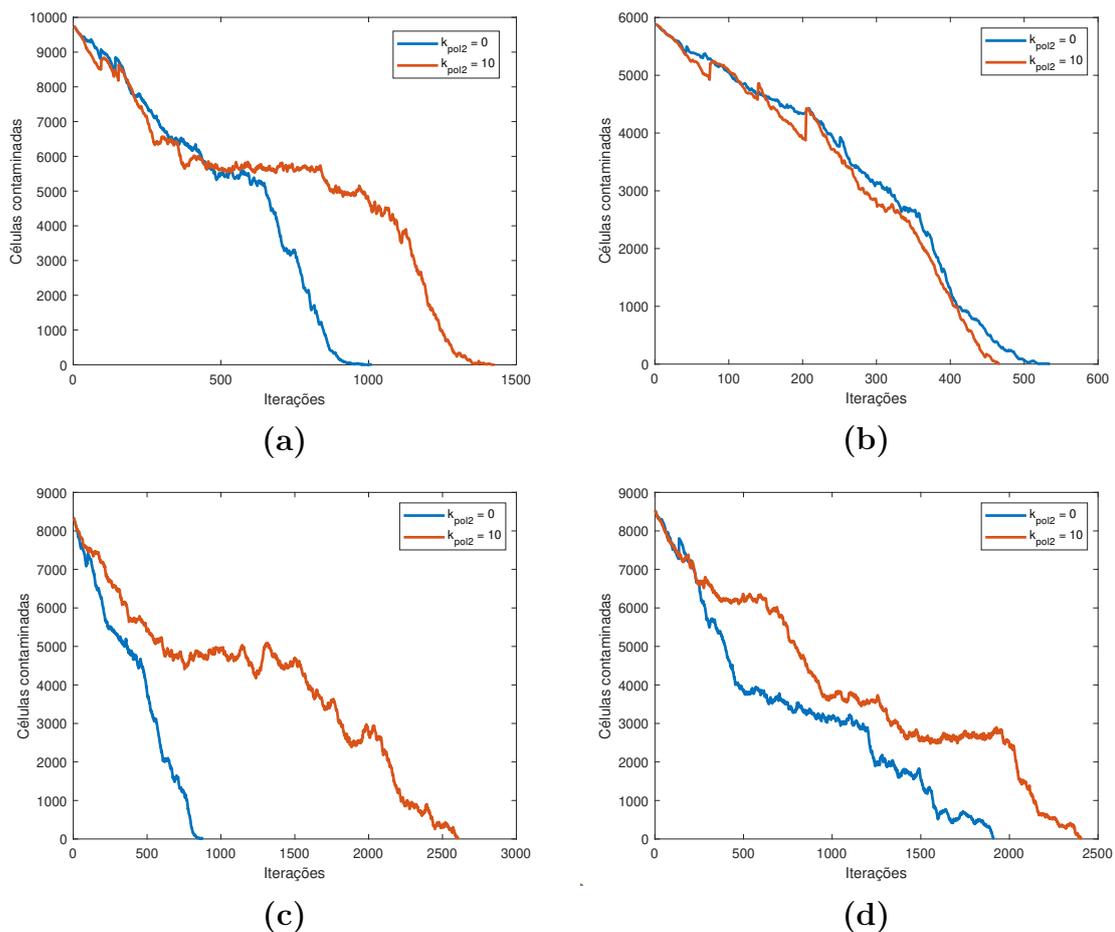


Figura 31 – Evolução da descontaminação pelas iterações para o teste com menor iterações da polarização escolhida para os mapas: (a) Mapa 1; (b) Mapa 2; (c) Mapa 3; (d) Mapa 4.

É possível ver na Tabela 5 que a variação no número de robôs possui um impacto na solução da tarefa. Isso pode ser visto no mapa 1, em que o teste com 11 robôs não foi possível concluir a tarefa em nenhum dos testes realizados por causa da limitação de iterações. Esse experimento não foi capaz de concluir devido ao fato que o número mínimo de robôs capaz de solucionar esse mapa é 11, e é necessário que os robôs se organizem de forma paralela ao lado do mapa e realizem suas trajetórias com essa formação.

Outra forma de impacto pode ser vista no mapa 4, pois com a diminuição do número de robôs o sistema pode não satisfazer mais as condições para garantia de completude. Pelo sistema não cumprir com as condições necessárias, é possível que os robôs não sejam capazes de completar a tarefa.

Esse é o caso para 11 robôs no mapa 4, em que existe uma trajetória possível da condição inicial até que o mapa esteja completamente limpo. Mas existem posições alcançáveis a partir do ponto inicial em que não é possível alcançar o estado completamente limpo depois que o time se separa, exceto para casos em que os times voltam para posições antes do obstáculo e se junta. Após essa junção o time necessita se movimentar para que a separação seja feita com uma quantidade diferente de robôs de

forma que possa completar a tarefa. Uma possível configuração que não consegue completar a tarefa é mostrada na Figura 32.

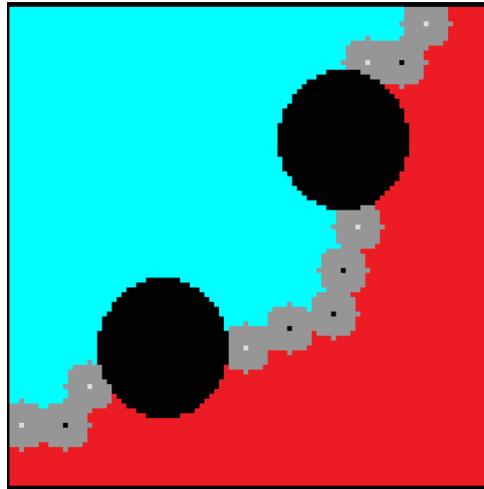


Figura 32 – Exemplo de configuração dos robôs em que não é possível que a tarefa seja concluída sem que os times voltem e se separem com outra configuração.

Na Figura 32 é possível observar que os times de robôs estão separados entre os obstáculos no ambiente, mas a divisão realizada é de tal forma que os times não possuem robôs suficientes para que haja a junção dos times após os obstáculos. Isso é devido ao fato que para esse mapa, para que um time que está entre a quina e o obstáculo, necessita de 4 robôs para que passe da quina e possa se juntar com o time entre os obstáculos.

Na Tabela 5 são mostrados a média e o desvio padrão para o método proposto e para o método de Kolling [15]. O método de Kolling foi usado como comparação por ser capaz de solucionar a tarefa em ambientes multiplamente conectados e os robôs possuem uma formação em linha. Para o método em [15] foi realizado apenas um teste, pois o método é determinístico, necessitando assim de apenas um teste.

		Mapa 1	Mapa 2	Mapa 3	Mapa 4
11 robôs	Método Proposto	Não concluiu	$793 \pm 148,99$	$1462,2 \pm 295,59$	1536 ± 32
	Método de Kolling	Não concluiu	331	417	Não concluiu
13 robôs	Método Proposto	$1834,1 \pm 600,32$	$1209,9 \pm 487,03$	$5047,3 \pm 3255,7$	$1927 \pm 428,53$
	Método de Kolling	Não concluiu	247	405	Não concluiu
15 robôs	Método Proposto	$1720,1 \pm 495,3858$	$1038,6 \pm 348,35$	$2983,4 \pm 2460,3$	$2803,2 \pm 826,2$
	Método de Kolling	202	247	405	Não concluiu
20 robôs	Método Proposto	$2058,4 \pm 345,9$	$945,1 \pm 356,72$	$2122,3 \pm 810,02$	$3482,6 \pm 819,46$
	Método de Kolling	202	247	405	Não concluiu

Tabela 6 – Média e desvio padrão para a quantidade de robôs e mapas escolhidos para o método proposto e o método de Kolling.

Na Tabela 6 é mostrado que o método de Kolling não foi capaz de resolver algumas situações que foram resolvidas pelo método proposto. Para o mapa 1 o método de Kolling não é capaz de resolver com 11 e 13 robôs devido a posição inicial dos robôs, que se encontram no canto superior esquerdo. Para que o método seja capaz de solucionar essa

situação, é necessário um número de robôs capaz de ligar a diagonal do mapa. Porém, se os robôs iniciassem a tarefa paralelos a uma das bordas, seria capaz de completar a tarefa com apenas 10 robôs.

Outra situação que o método de Kolling não foi capaz de resolver é do mapa 4, já que inicialmente o time se movimenta até achar os dois círculos. Após achar os círculos o time de robôs se separa e espera outro time. Dessa forma, o time que se movimenta precisa que a linha continue a contornar o mapa até que ache outro time de robôs que foi separado previamente. Esse comportamento necessita de uma quantidade de robôs maior do que os testados.

Na Figura 33 é mostrada a evolução da descontaminação para o método proposto ao longo das iterações para diferentes números de robôs em cada mapa do resultado com menor número de iterações.

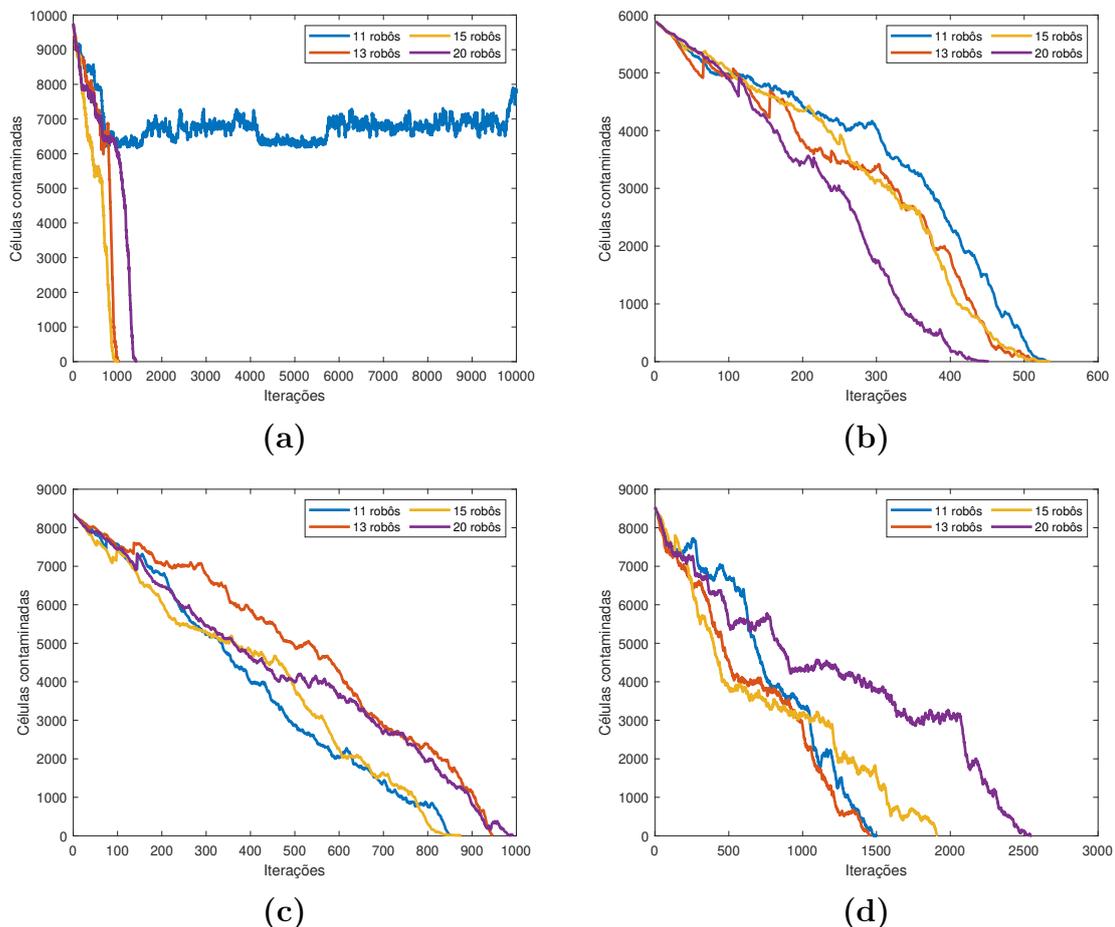


Figura 33 – Evolução da descontaminação pelas iterações para o teste com menor iterações da polarização escolhida para os mapas: (a) Mapa 1; (b) Mapa 2; (c) Mapa 3; (d) Mapa 4.

Na Figura 33 pode ser visto que a evolução da descontaminação pelas iterações possuem respostas parecidas para os melhores casos. O único caso que possui uma diferença é para o mapa 1 com 11 robôs, em que o sistema não foi capaz de completar a tarefa em nenhum dos testes realizados.

Dessa forma, embora a média de iterações possua variações para números diferentes de robôs a resposta pode ser similar devido ao alto desvio padrão nos testes.

Portanto o maior impacto encontrado para números diferentes de robôs é ao utilizar um número de robôs próximo ao mínimo possível para a conclusão da tarefa, já que para essas situações precisam de uma formação específica. Para esses casos, é desejado que a componente aleatória do sistema seja maior a fim de realizar a formação. Então é esperado que para situações com número de robôs perto do limite mínimo para a realização, valores de polarização menores possam ser melhores, sendo necessários mais testes para validação.

7 Conclusão

Nesta dissertação foi apresentada uma solução para a tarefa de perseguição-evasão do pior caso, em ambientes conectados. Também foi apresentada uma demonstração que a tarefa é concluída em um tempo finito para um time de robôs com movimento baseado em *random walk*.

A demonstração é primeiramente referente a situação em que os robôs se movimentam utilizando a *random walk* sem utilização de restrições locais. Após isso, foi realizada a demonstração com uso de restrições locais para que não haja a recontaminação completa do ambiente por quebra da formação dos robôs, para diminuição do tempo médio para completar a tarefa.

A solução apresentada está de acordo com a demonstração apresentada, de forma que possui a garantia de finalizar a tarefa em um tempo finito. Também foram propostas duas polarizações com o intuito de finalizar a tarefa mais rapidamente, em que a primeira polarização foi baseada na proposta em [18].

Para que a solução apresentada pudesse ser testada, foi implementado um ambiente de simulação para testes de perseguição-evasão de pior caso. O ambiente construído permite a alteração dos parâmetros do sistema, como o número de robôs, as constantes de polarização, o mapa utilizado, entre outros.

Com a utilização do ambiente de simulação a abordagem foi testada, inicialmente sem as polarizações propostas para análise do comportamento demonstrado. Na sequência, foram feitas as análises com a primeira polarização apresentada, com a obtenção de resultados que demonstram o efeito da polarização em diminuir o tempo para a finalização da tarefa.

Foi analisado também o impacto da segunda polarização apresentada, para diminuir ocasiões em que os robôs, que estão em uma posição em que a primeira polarização não tem efeito, dificultam a movimentação do time.

Essa polarização causou um comportamento de aglomeração dos robôs em regiões descontaminadas. Devido a essas aglomerações a movimentação em direção as células contaminadas foi prejudicada o que aumentou o número de iterações médias para que a tarefa seja completada.

Outra análise realizada nesta dissertação foi em relação ao impacto do número de robôs no sistema, em que foi demonstrado que é possível a conclusão da tarefa com diferentes números de robôs, contanto que a quantidade de robôs seja capaz de completar a tarefa.

A abordagem proposta também foi comparada com o método proposto em [15], para número de robôs variados, em que a abordagem proposta foi capaz de completar a tarefa em mapas que [15] não foi capaz de completar. Dessa forma, A abordagem proposta é capaz de solucionar situações de perseguição-evasão com a garantia da solução, com a problemática de necessitar de mais iterações para a solução ser alcançada.

Então a abordagem proposta é capaz de garantir a solução da tarefa de maneira que seja de fácil implementação para os robôs e que não necessita de uma capacidade de processamento alta.

Como proposta de trabalhos futuros, é sugerido que sejam realizados testes para avaliar melhor o efeito de k_{pol2} e realização de testes para verificar o efeito da polarização para números de robôs variados. Esses testes possuem o intuito de verificar se o valor de polarização se mantém em situações que o número de robôs é o mínimo ou muito maior do que o necessário para a realização da tarefa.

É proposto também que o sistema seja implementado de forma descentralizada para os robôs, de forma que diminua a necessidade da comunicação entre eles, a fim de obter uma simplicidade maior da solução menor custo do sistema, por permitir a utilização de robôs mais simples. Mais ainda, é proposta a análise da perda de robôs durante a tarefa e estratégias para poder concluir a tarefa com sucesso.

Ainda como trabalhos futuros podem ser realizados testes com robôs reais para verificação da viabilidade em sistemas práticos.

Por fim, é proposta uma análise para computar o tempo médio necessário que o sistema necessita para solucionar o problema, a partir da cadeia de Markov, considerando a inclusão de restrições nas movimentações dos robôs.

Referências

- 1 RUBIO, F.; VALERO, F.; LLOPIS-ALBERT, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 16, n. 2, p. 1729881419839596, 2019. 1
- 2 GOSSELIN, C. M.; JEAN, M. Determination of the workspace of planar parallel manipulators with joint limits. *Robotics and Autonomous Systems*, Elsevier, v. 17, n. 3, p. 129–138, 1996. 1
- 3 VILLA, D. K.; BRANDAO, A. S.; SARCINELLI-FILHO, M. A survey on load transportation using multirotor uavs. *Journal of Intelligent & Robotic Systems*, Springer, v. 98, p. 267–296, 2020. 1
- 4 LLUVIA, I.; LAZKANO, E.; ANSUATEGI, A. Active mapping and robot exploration: A survey. *Sensors*, MDPI, v. 21, n. 7, p. 2445, 2021. 1
- 5 DREW, D. S. Multi-agent systems for search and rescue applications. *Current Robotics Reports*, Springer, v. 2, p. 189–200, 2021. 1
- 6 TANG, Q. et al. Swarm robots search for multiple targets. *IEEE Access*, IEEE, v. 8, p. 92814–92826, 2020. 1
- 7 HU, J. et al. Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, IEEE, v. 69, n. 12, p. 14413–14423, 2020. 1
- 8 CARDONA, G.; TELLEZ-CASTRO, D.; MOJICA-NAVA, E. Cooperative transportation of a cable-suspended load by multiple quadrotors. *IFAC-PapersOnLine*, Elsevier, v. 52, n. 20, p. 145–150, 2019. 1
- 9 DUTTA, A.; GHOSH, A.; KREIDL, O. P. Multi-robot informative path planning with continuous connectivity constraints. In: IEEE. *2019 international conference on robotics and automation (ICRA)*. [S.l.], 2019. p. 3245–3251. 1
- 10 LIN, Y. et al. Optimal caching scheme in d2d networks with multiple robot helpers. *Computer Communications*, Elsevier, v. 181, p. 132–142, 2022. 1
- 11 SHENG, W. et al. Distributed multi-robot coordination in area exploration. *Robotics and autonomous systems*, Elsevier, v. 54, n. 12, p. 945–955, 2006. 1
- 12 DARMANIN, R. N.; BUGEJA, M. K. A review on multi-robot systems categorised by application domain. In: IEEE. *2017 25th mediterranean conference on control and automation (MED)*. [S.l.], 2017. p. 701–706. 1
- 13 YAN, Z.; JOUANDEAU, N.; CHERIF, A. A. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 10, n. 12, p. 399, 2013. 2, 8

- 14 SIRINTUNA, D.; OZDAMAR, I.; AJOUDANI, A. Carrying the uncarriable: a deformation-agnostic and human-cooperative framework for unwieldy objects using multiple robots. In: IEEE. *2023 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2023. p. 7497–7503. 2
- 15 KOLLING, A.; CARPIN, S. Multi-robot pursuit-evasion without maps. In: IEEE. *2010 IEEE International Conference on Robotics and Automation*. [S.l.], 2010. p. 3045–3051. 2, 3, 9, 15, 16, 37, 45, 47, 51
- 16 DURHAM, J. W.; FRANCHI, A.; BULLO, F. Distributed pursuit-evasion without mapping or global localization via local frontiers. *Autonomous Robots*, Springer, v. 32, n. 1, p. 81–95, 2012. 2, 14
- 17 GREGORIN, L. d. V. et al. Robótica evolutiva aplicada ao problema de perseguição-evasão de pior caso com múltiplos robôs. Universidade Federal de Sergipe, 2016. 2, 15
- 18 GONÇALVES, A. R. C. et al. Problema de perseguição-evasão baseado em random walk. Universidade Federal de Sergipe, 2016. 2, 3, 9, 11, 16, 17, 20, 21, 23, 25, 26, 50
- 19 FRANCO, D. A. S. et al. Worst-case pursuit-evasion using random walk for multiply connected environment. In: *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*. [S.l.: s.n.], 2023. p. 194–199. 3
- 20 CHUNG, T. H.; HOLLINGER, G. A.; ISLER, V. Search and pursuit-evasion in mobile robotics. *Autonomous robots*, Springer, v. 31, n. 4, p. 299–316, 2011. 5
- 21 KUMKOV, S. S.; MÉNEC, S. L.; PATSKO, V. S. Zero-sum pursuit-evasion differential games with many objects: survey of publications. *Dynamic games and applications*, Springer, v. 7, n. 4, p. 609–633, 2017. 5
- 22 BRAVO, L.; RUIZ, U.; MURRIETA-CID, R. A pursuit–evasion game between two identical differential drive robots. *Journal of the Franklin Institute*, Elsevier, v. 357, n. 10, p. 5773–5808, 2020. 5, 13
- 23 SANI, M.; ROBU, B.; HABLY, A. Pursuit-evasion game for nonholonomic mobile robots with obstacle avoidance using nm-pc. In: IEEE. *2020 28th Mediterranean Conference on Control and Automation (MED)*. [S.l.], 2020. p. 978–983. 5, 13
- 24 PARSONS, T. D. Pursuit-evasion in a graph. In: *Theory and applications of graphs*. [S.l.]: Springer, 1978. p. 426–441. 5
- 25 KEHAGIAS, A.; HOLLINGER, G.; SINGH, S. A graph search algorithm for indoor pursuit/evasion. *Mathematical and Computer Modelling*, Elsevier, v. 50, n. 9-10, p. 1305–1317, 2009. 6, 13
- 26 LOZANO, E. et al. A visibility-based pursuit-evasion game between two nonholonomic robots in environments with obstacles. *Autonomous Robots*, Springer, v. 46, n. 2, p. 349–371, 2022. 7
- 27 JIN, L. et al. Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach. *Automatica*, Elsevier, v. 100, p. 75–81, 2019. 8

- 28 BAXTER, J. L. et al. Shared potential fields and their place in a multi-robot co-ordination taxonomy. *Robotics and Autonomous Systems*, Elsevier, v. 57, n. 10, p. 1048–1055, 2009. 8
- 29 FARINELLI, A.; IOCCHI, L.; NARDI, D. Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 34, n. 5, p. 2015–2028, 2004. 8
- 30 LIN, S. et al. A review of path-planning approaches for multiple mobile robots. *Machines*, MDPI, v. 10, n. 9, p. 773, 2022. 8
- 31 DURHAM, J. W.; FRANCHI, A.; BULLO, F. Distributed pursuit-evasion with limited-visibility sensors via frontier-based exploration. In: IEEE. *2010 IEEE International Conference on Robotics and Automation*. [S.l.], 2010. p. 3562–3568. 9
- 32 IBE, O. C. *Elements of random walk and diffusion processes*. [S.l.]: John Wiley & Sons, 2013. 10
- 33 PANG, B. et al. A swarm robotic exploration strategy based on an improved random walk method. *Journal of Robotics*, Hindawi Limited, v. 2019, p. 1–9, 2019. 11
- 34 PANG, B. et al. Effect of random walk methods on searching efficiency in swarm robots for area exploration. *Applied Intelligence*, Springer, v. 51, p. 5189–5199, 2021. 11
- 35 NAUTA, J. et al. Enhanced foraging in robot swarms using collective lévy walks. In: IOS. *24th European Conference on Artificial Intelligence (ECAI)*. [S.l.], 2020. v. 325, p. 171–178. 11
- 36 LEVIN, D. A.; PERES, Y. *Markov chains and mixing times*. [S.l.]: American Mathematical Soc., 2017. 11, 35
- 37 DUAN, X.; BULLO, F. Markov chain-based stochastic strategies for robotic surveillance. *Annual Review of Control, Robotics, and Autonomous Systems*, Annual Reviews, v. 4, p. 243–264, 2021. 11
- 38 SHIRSAT, A.; ELAMVAZHUTHI, K.; BERMAN, S. Multi-robot target search using probabilistic consensus on discrete markov chains. In: IEEE. *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. [S.l.], 2020. p. 108–115. 11
- 39 FANG, X. et al. Cooperative pursuit with multi-pursuer and one faster free-moving evader. *IEEE transactions on cybernetics*, IEEE, v. 52, n. 3, p. 1405–1414, 2020. 11
- 40 ALAM, T. et al. Stochastic multi-robot patrolling with limited visibility. *Journal of Intelligent & Robotic Systems*, Springer, v. 97, p. 411–429, 2020. 11
- 41 CUI, J. et al. Game-model prediction hybrid path planning algorithm for multiple mobile robots in pursuit evasion game. In: IEEE. *2021 IEEE International Conference on Unmanned Systems (ICUS)*. [S.l.], 2021. p. 925–930. 13
- 42 KOKOLAKIS, N.-M. T.; VAMVOUDAKIS, K. G. Safety-aware pursuit-evasion games in unknown environments using gaussian processes and finite-time convergent reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, 2022. 13

- 43 WAN, K. et al. An improved approach towards multi-agent pursuit–evasion game decision-making using deep reinforcement learning. *Entropy*, MDPI, v. 23, n. 11, p. 1433, 2021. 13
- 44 FU, X. et al. A uav pursuit-evasion strategy based on ddpq and imitation learning. *International Journal of Aerospace Engineering*, Hindawi Limited, v. 2022, p. 1–14, 2022. 13
- 45 ZHOU, Z.; XU, H. Mean field game and decentralized intelligent adaptive pursuit evasion strategy for massive multi-agent system under uncertain environment. In: IEEE. *2020 American Control Conference (ACC)*. [S.l.], 2020. p. 5382–5387. 13
- 46 WANG, M.; WANG, L.; YUE, T. An application of continuous deep reinforcement learning approach to pursuit-evasion differential game. In: IEEE. *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. [S.l.], 2019. p. 1150–1156. 13
- 47 RAMAITHITIMA, R. *Sensor-based Topological Coverage and Mapping Algorithms for Resource-constrained Robot Swarms*. [S.l.]: University of Pennsylvania, 2019. 13
- 48 OLSEN, T. et al. A visibility roadmap sampling approach for a multi-robot visibility-based pursuit-evasion problem. In: IEEE. *2021 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2021. p. 7957–7964. 13
- 49 STIFFLER, N. M.; O’KANE, J. M. A sampling-based algorithm for multi-robot visibility-based pursuit-evasion. In: IEEE. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2014. p. 1782–1789. 13
- 50 STIFFLER, N. M.; O’KANE, J. M. Complete and optimal visibility-based pursuit-evasion. *The International Journal of Robotics Research*, SAGE Publications Sage UK: London, England, v. 36, n. 8, p. 923–946, 2017. 13
- 51 LI, A. Q. et al. A search-based approach to solve pursuit-evasion games with limited visibility in polygonal environments. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. [S.l.: s.n.], 2018. p. 1693–1701. 13
- 52 STIFFLER, N. M.; O’KANE, J. M. Planning for robust visibility-based pursuit-evasion. In: IEEE. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2020. p. 6641–6648. 13
- 53 ENGIN, S.; JIANG, Q.; ISLER, V. Learning to play pursuit-evasion with visibility constraints. In: IEEE. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2021. p. 3858–3863. 14
- 54 SUN, Z. et al. Cooperative strategy for pursuit-evasion problem with collision avoidance. *Ocean Engineering*, Elsevier, v. 266, p. 112742, 2022. 14
- 55 ZHANG, R. et al. Game of drones: Multi-uav pursuit-evasion game with online motion planning by deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, 2022. 14
- 56 OLSEN, T.; STIFFLER, N. M.; O’KANE, J. M. Rapid recovery from robot failures in multi-robot visibility-based pursuit-evasion. In: IEEE. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2021. p. 9734–9741. 14

-
- 57 OLSEN, T.; STIFFLER, N. M.; O’KANE, J. M. Robust-by-design plans for multi-robot pursuit-evasion. In: IEEE. *2022 International Conference on Robotics and Automation (ICRA)*. [S.l.], 2022. p. 10716–10722. 14
- 58 KLEINER, A.; KOLLING, A. Guaranteed search with large teams of unmanned aerial vehicles. In: IEEE. *2013 IEEE International Conference on Robotics and Automation*. [S.l.], 2013. p. 2977–2983. 14
- 59 KOUZEGHAR, M. et al. Multi-target pursuit by a decentralized heterogeneous uav swarm using deep multi-agent reinforcement learning. *arXiv preprint arXiv:2303.01799*, 2023. 15