



Universidade Federal de Sergipe
Pró-Reitoria de Pós-Graduação e Pesquisa
Programa de Pós-graduação em Engenharia Elétrica

ROBÓTICA EVOLUTIVA APLICADA AO PROBLEMA DE
PERSEGUIÇÃO-EVASÃO DE PIOR CASO COM MÚLTIPLOS ROBÔS

Lívia do Vale Gregorin

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica (PROEE) da Universidade Federal de Sergipe, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Eduardo Oliveira Freire
Elyson Ádan Nunes Carvalho

São Cristóvão
Dezembro de 2016



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

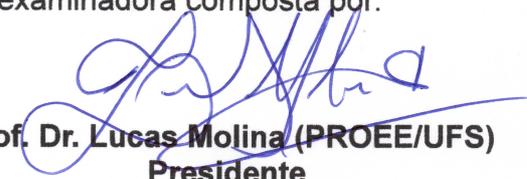
**“Robótica Evolutiva Aplicada ao Problema de Perseguição-
Evasão de Pior Caso com Múltiplos Robôs”**

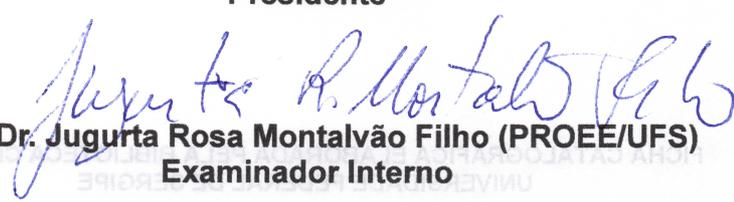
Discente:

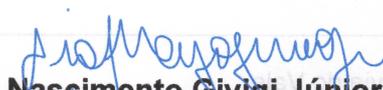
Lívia do Vale Gregorin

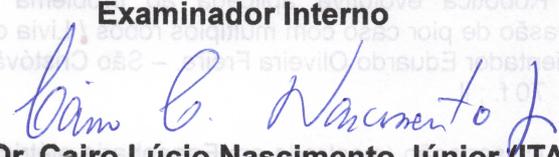
Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Sergipe, como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

Aprovada pela banca examinadora composta por:


Prof. Dr. Lucas Molina (PROEE/UFS)
Presidente


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


Prof. Dr. Sidney Nascimento Givigi Júnior (PROEE/UFS)
Examinador Interno


Prof. Dr. Cairo Lúcio Nascimento Júnior (ITA)
Examinador Externo


Lívia do Vale Gregorin
Candidata

Cidade Universitária “Prof. José Aloísio de Campos”, 20 de dezembro de 2016.

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE**

Gregorin, Livia do Vale
G821r Robótica evolutiva aplicada ao problema de perseguição-
evasão de pior caso com múltiplos robôs / Livia do Vale Gregorin ;
orientador Eduardo Oliveira Freire. – São Cristóvão, 2016.
70 f. ; il.

Dissertação (mestrado em Engenharia elétrica) – Universidade
Federal de Sergipe, 2016.

1. Engenharia elétrica. 2. Robótica. 3. Máquinas. I. Freire,
Eduardo Oliveira, orient. II. Título.

CDU: 621.3:007.52

Agradecimentos

Indubitavelmente, o período de desenvolvimento deste mestrado, considerando todos os anos da presente existência, foi o que me proporcionou as maiores e mais incríveis experiências.

Ao Criador agradeço por todas as oportunidades e pessoas estrategicamente alocadas para me fazerem progredir, e pela saúde que se sustentou em meu organismo apesar de todos os maus tratos de alimentação e movimentação.

Ao meu quási-marido Felipe Anderson, por sustentar minhas emoções em seus mínimos locais, por tanta paciência com meus períodos de mau humor e estresse, por todos os puxões de orelha, pela cumplicidade, por nunca fechar as portas do seu coração para mim, por sempre me lembrar o que é realmente importante, por ter sido sempre meu refúgio, meu porto seguro, meu amor.

Aos meus pais Iracema e Euclides, por todo o ensino moral desde meus primeiros passos, e por suportarem a ausência física e até mesmo minha negligência de atenção.

Ao meu irmão Flávio, por ter me ensinado desde a infância a dar o melhor de mim, no que quer que seja.

À Stephanie Kamarry, uma enorme gratidão pela parceria e pelo gigantesco apoio em toda minha caminhada, seja na UFS, no mestrado, em minha vida profissional, social ou pessoal. Por ter estado sempre presente, indicando caminhos e abrindo portas. Por ter compartilhado comigo seu sonho de *Irradiar* e por ter me permitido vivê-lo também. Por ter me ensinado tanto, e resgatado tanto do que eu escondia: criatividade, disciplina, coragem. A caminhada ainda é longa, mas me felicito por continuar contando com seu apoio.

Ao meu orientador Eduardo Freire, agradeço primeiramente pela confiança que sempre depositou em mim e por estar sempre disponível para ajudar quando precisei. Agradeço também por reacender a motivação quando as coisas pareciam que não iam tão bem, e pelo apoio que depositou em todas as minhas atividades de docência. Por me fazer valorizar minhas características de docente e pesquisadora e por me incentivar a sempre seguir em frente.

Ao meu co-orientador Elyson Carvalho, por extrapolar no ensino de competências técnicas e especialmente humanas. Por melhorar meu estado psicológico quando necessário, por me mostrar novos caminhos quando eu não via soluções. Por apoiar o Projeto Irradiar

com todo seu coração e conhecimento.

Ao co-orientador não-oficial Lucas Molina, agradeço pela visão crítica que contribuiu não somente para melhorar meu olhar a respeito da robótica evolutiva, mas também para ajustar minhas características de pesquisadora, tornando-me mais madura e preparada para saber contribuir efetivamente para a ciência, e não apenas publicar trabalhos.

Como parte do corpo discente do Grupo de Pesquisa em Robótica da UFS, agradeço em uníssono aos professores Eduardo, Elyson e Lucas, pela confiança e principalmente por me proporcionarem aprendizados práticos a respeito de liderança, confiança, hierarquia, método, justiça e amizade.

Aos professores Jugurta Montalvão e Jânio Canuto, pela presteza e interesse que sempre demonstraram por meu trabalho e pelas ideias que fluem naturais de suas mentes incríveis. A Jugurta agradeço ainda por fazer parte de minha banca avaliadora e mais uma vez contribuir em meu trabalho.

Ao professor Sidney Givigi, que co-orientou não formalmente, mas efetivamente, grande parte de minha pesquisa, agradeço pela dedicação, presteza, boas ideias e boa vontade que sempre demonstrou, presencialmente e também virtualmente.

Ao professor Cairo Nascimento, agradeço pela boa vontade em deslocar-se até Aracaju para contribuir em minha banca, mesmo com todos os contratempos de entrega dos Correios.

Aos amigos antigos do GPR, agradeço pela companhia em todo esse trajeto e por todos os bons momentos que passamos juntos! Um agradecimento especial a Stephanie Kamarry (*again*), Phillipe Cardoso, Mateus Cardoso, Shyenne Pinheiro, Ruan Bispo, Erik Richard, Clebeson Canuto, Jefferson Júnior e mais especialmente a Renato Gonçalves, por ter me inspirado no tema do mestrado e por todo apoio oferecido.

Aos colegas de mestrado Jeferson França, Roberto Menezes, Luiz Miranda e Dami Dória, agradeço pela ajuda que me ofereceram, principalmente no início da caminhada!

À querida Mayane por todo apoio técnico, burocrático e estrutural que sempre ofereceu e por toda a boa vontade e dedicação com que me acompanhou neste mestrado.

À CAPES, por todo o apoio financeiro!

A todos que torceram por mim de perto ou de longe, aos que cuidaram de mim com preces e sorrisos, aos que cruzaram meu caminho e fizeram de mim uma pessoa melhor...

O B R I G A D A !

Resumo

ROBÓTICA EVOLUTIVA APLICADA AO PROBLEMA DE PERSEGUIÇÃO-EVASÃO DE PIOR CASO COM MÚLTIPLOS ROBÔS

Lívia do Vale Gregorin

Dezembro/2016

Orientadores: Eduardo Oliveira Freire

Elyson Ádan Nunes Carvalho

Departamento: Engenharia Elétrica (DEL/CCET/UFS)

Com o passar dos anos, é crescente a demanda pela utilização de robôs em tarefas de auxílio aos seres humanos, em especial àquelas que envolvam riscos, como resgates e buscas. Para tais atividades, tem sido comum a utilização do trabalho conjunto e coordenado entre vários robôs, e uma das formas de estudar e modelar estas aplicações envolve o problema de perseguição-evasão. Este trabalho propõe o uso de uma abordagem baseada em robótica evolutiva para solucionar o problema de perseguição-evasão de pior caso, em que os evasores são considerados infinitamente rápidos e oniscientes, enquanto os perseguidores têm capacidade de sensoriamento e comunicação limitados. Nesta pesquisa, não é considerado nenhum conhecimento prévio a respeito dos ambientes, que são tratados de forma discreta e podem ser multiplamente conectados. É proposto um sistema de controle descentralizado para múltiplos robôs baseado em uma máquina de estados finitos cujo mapeamento de estados em ações é definido por meio de um algoritmo genético, aplicado nos termos da robótica evolutiva. Os resultados mostram que o sistema proposto é capaz de descontaminar diversos tipos de mapas, porém sem robustez para todas as inicializações de posição, devido à incompletude na definição do mapeamento do autômato. A fim de investigar parcialmente tal fato, é apresentada uma abordagem complementar em que sorteios são utilizados em conjunto com o autômato evoluído, indicando ações aleatórias nos casos de estados pouco visitados durante a evolução, o que resultou em melhora nos resultados, apontando caminhos para trabalhos futuros. Além disso, é também realizada uma análise comparativa da abordagem evolutiva proposta com outro método de solução para o mesmo problema, sendo discutidas vantagens e desvantagens de cada trabalho.

Palavras-chave: robótica evolutiva, perseguição-evasão, problema de limpeza.

Abstract

EVOLUTIONARY ROBOTICS APPLIED TO THE MULTI-ROBOT WORST-CASE PURSUIT-EVASION PROBLEM

Lívia do Vale Gregorin

December/2016

Advisors: Eduardo Oliveira Freire
Elyson Ádan Nunes Carvalho

Department: Electrical Engineering (DEL/CCET/UFS)

Over the years, there is a growing demand for the use of robots to assist humans in their tasks, especially those involving risks, such as search and rescue. For this reason, coordination among several robots has been a common option, and one of the ways to study and model these applications involves the problem of pursuit-evasion. This work proposes the use of an evolutionary robotics approach to solve the worst-case pursuit-evasion problem, in which evaders are considered arbitrarily fast and omniscient, while pursuers have limited sensing and communication capabilities. In this research, no prior knowledge is considered regarding environments, which are treated as discrete and can be multiply connected. A decentralized multi-robot control system is proposed, based on a finite state machine with state-action mapping defined by means of a genetic algorithm, applied in terms of evolutionary robotics. The results show the proposed system is able to decontaminate several types of maps, but does not generalize to all initial conditions, due to the incompleteness in the automaton mapping. Therefore, a complementary approach is presented in which draws are used alternatively with the evolved automaton, indicating random actions in cases of states not sufficiently visited during evolution. This investigation led to better results, indicating ways for future work. In addition, a comparative analysis of the evolutionary approach is also carried out with another solution method for the same problem, with advantages and disadvantages of each work being discussed.

Sumário

Lista de Figuras	viii
Lista de Tabelas	x
1 Introdução	1
2 Perseguição-Evasão	5
2.1 Origens	5
2.2 Perseguição-Evasão em Grafos	6
2.3 Perseguição-Evasão Baseada em Visibilidade	8
2.4 Coordenação de Múltiplos Perseguidores em Ambientes Desconhecidos . . .	10
3 Robótica Evolutiva	15
3.1 Algoritmos Genéticos	20
4 Abordagem Proposta	26
4.1 Ambiente e restrições de simulação	26
4.2 Modelo do controlador	30
4.3 Modelo de evolução	35
5 Resultados	38
5.1 Parâmetros do Processo Evolutivo	38
5.2 Análise das Soluções Evoluídas	41
5.3 Análise de Abordagem Complementar: aplicação de sorteios aleatórios . . .	47
5.4 Análise Comparativa	49
6 Conclusões	52
Referências Bibliográficas	54
A Pseudo-código	61

Lista de Figuras

2.1	Composição dos robôs na linha de varredura (adaptado de [1])	10
2.2	Busca por obstáculo e formação de nova linha [1]	11
2.3	Exemplo de percepção dos sensores [2]	12
2.4	Ações de movimentação do trabalho de Gonçalves [3]: (a) restrição de movimentação do robô; (b) distribuição uniforme de probabilidades do <i>random walk</i> e (c) distribuição ajustada de probabilidades segundo células já visitadas	13
3.1	Relação entre engenharia, robótica evolutiva e biologia [4]	16
3.2	Estrutura geral de um algoritmo genético	21
3.3	Exemplo de representação cromossomial em (a) <i>bits</i> e (b) números reais . .	22
3.4	Estrutura de um AG no contexto da robótica evolutiva (adaptado de [4]) .	22
3.5	Simulação de uma roleta viciada	23
3.6	Exemplo de <i>crossover</i> binário de (a) 1 ponto e (b) 3 pontos	24
3.7	Exemplo de mutação binária de um gene	24
4.1	Exemplo de um mapa (a) real; (b) discretizado e (c) como visto na simulação, inicialmente desconhecido	27
4.2	Restrição de movimentação: permanecer na célula ou deslocar-se na vizinhança de von Neumann	27
4.3	(a) Máscara de sensoriamento discreta e (b) exemplo de visão do robô . . .	28
4.4	Exemplo de configurações-limite para comunicação e descontaminação . . .	29
4.5	Exemplo de formação inicial com total comunicação entre os buscadores . .	29
4.6	Exemplo de recontaminação no ambiente simulado	30
4.7	Exemplos de situações de estado	32
4.8	Sentidos para análise de descontaminação	33
4.9	Exemplos de possibilidade de colisão	34
4.10	Mapeamento de estados em ações	34
4.11	Processo de restrição de colisão	36
4.12	Exemplificação de (a) <i>crossover</i> e (b) mutação na abordagem proposta . .	37
5.1	Mapas utilizados no processo evolutivo	39
5.2	Exemplo de inicialização da formação	39

5.3	Exemplo de finalização da formação	40
5.4	Fluxograma do processo evolutivo completo	40
5.5	Taxa de sucesso na tarefa de descontaminação	43
5.6	Continuação do processo evolutivo para um experimento <i>outlier</i>	43
5.7	Mapas multiplamente conectados usados nos testes	44
5.8	Taxa de sucesso na descontaminação dos novos mapas	45
5.9	Mapas conjugados utilizados nos testes	45
5.10	Exemplo de escolha de movimentação na abordagem complementar	47
5.11	Análise comparativa entre as abordagens evolutiva (ER) e baseada em <i>random walk</i> (RW) [5]: em escala completa e com detalhe para até 200 iterações	50

Lista de Tabelas

4.1	Descrição do Vetor de Estado	31
4.2	Possíveis ações de movimentação para o robô	35
5.1	Parâmetros do Algoritmo Genético	41
5.2	Média de sucesso dos experimentos por geração	42
5.3	Média de sucesso da última geração	42
5.4	Descontaminação de mapas multiplamente conectados	44
5.5	Resultados para os quatro mapas conjugados	46
5.6	Resultados para os quatro mapas conjugados com aplicação do sorteio para estados pouco visitados	48
5.7	Média e desvio padrão das iterações válidas e das tentativas na abordagem de Gonçalves	50

Capítulo 1

Introdução

O ser humano está cada vez mais propenso a integrar robôs ao seu cotidiano para realizar atividades de crescente complexidade. Muito tem sido feito nas últimas décadas pelos pesquisadores a respeito da robótica de manipuladores e robótica móvel, utilizando-se de robôs que cumprem tarefas de forma individual.

Mais recentemente, a partir do final da década de 1980, as atenções se voltaram para a investigação de sistemas que utilizam vários robôs em trabalho conjunto que, apesar de ser uma extensão das pesquisas de sistemas com um único robô, traz desafios particulares e relevantes que a tornaram uma nova área independente de estudo [6], chamada sistema multirrobôs (MRS - *multi-robot system*) ou ainda robótica multi-agente.

A ideia central sugere a interação entre dois ou mais robôs móveis autônomos (referenciados como times ou sociedades) para que o trabalho conjunto e coordenado resulte em objetivos bem definidos ou em soluções para problemas complexos [7], por vezes impossíveis para um único robô.

Pesquisadores concordam que há várias vantagens em coordenar múltiplos robôs ao invés de utilizar o trabalho de um robô individual [6, 7]. As motivações mais comuns para o desenvolvimento de MRSs são:

- Complexidade das tarefas: algumas são inerentemente complexas e podem ser impossíveis para um único robô. A complexidade pode estar relacionada com a natureza distribuída das tarefas, como em ações simultâneas separadas espacialmente, e/ou em termos de diferentes requisitos, quando então robôs com distintas habilidades podem trabalhar em conjunto, oferecendo versatilidade;
- Simplicidade de projeto: trabalhar com um conjunto de robôs de construção e controle simples pode ser mais fácil e barato de implementar do que um robô de mecanismo complexo, integrado com múltiplos sensores e um avançado sistema inteligente de controle;
- Aumento do desempenho do sistema: robôs trabalhando em paralelo podem resolver

problemas com redução de tempo e consumo de energia, além de evitar gargalos em horas críticas que podem comprometer a conclusão da tarefa;

- Robustez e confiabilidade: esses sistemas são normalmente distribuídos e inerentemente redundantes. Isso os torna mais tolerantes a falhas e a perda de um robô (especialmente em aplicações em ambientes hostis) pode não comprometer o cumprimento da tarefa.

Os MRSs têm sido propostos em várias configurações e estruturas, com grupos homogêneos e heterogêneos, visando diferentes objetivos de pesquisa e obtendo bons resultados em muitos domínios de aplicação [8], sendo bem adequados a tarefas que requerem comportamentos coordenados, em ambientes cooperativos e/ou competitivos.

O trabalho de Fukuda *et al.* [9] está entre os primeiros em que foram considerados times de robôs trabalhando em conjunto. Recentemente, as investigações de aplicabilidade dos MRSs têm sido voltadas a áreas como segurança inteligente [10], busca e resgate [11], vigilância e patrulha [12], desminagem humanitária [13], monitoramento de ambiente [14], entre outros.

Muitas dessas aplicações envolvendo múltiplos robôs têm suas modelagens inspiradas nos chamados jogos de perseguição-evasão, utilizados há anos para investigar questões fundamentais da Matemática e Ciência da Computação ou para soluções de problemas práticos [15].

De forma geral, o problema (referenciado por vezes como jogo, devido à sua natureza “adversária”) é definido como a ação de um ou mais agentes perseguidores tentando identificar, se aproximar e possivelmente capturar um ou mais agentes evasores, que de forma oposta tentam manter distância e não serem capturados por seus perseguidores.

Segundo Isaacs [16], futebol, esgrima, um atirador com alvo cujo movimento é voluntário, as logísticas de uma batalha, um torpedo perseguindo um navio ou um míssil interceptando um bombardeiro são todos exemplos de situações possíveis de serem modeladas como problemas de perseguição-evasão. Também podem se encaixar nestas modelagens os cenários gerais de busca e resgate, patrulha, vigilância e exploração, que têm sido de fundamental importância nas pesquisas em robótica.

Diferentes modelos podem ser obtidos a depender da dinâmica permitida aos perseguidores e evasores, condições de captura, restrições globais e de visibilidade [17, 18]. A partir destas considerações, diversas taxonomias são encontradas e diversos estudos podem ser realizados sobre comportamentos coletivos de cooperação e competição.

O problema de perseguição-evasão é classificado [19] como busca adversária, fazendo referência aos casos em que o evasor tem a intenção explícita de escapar do perseguidor. Isto difere por exemplo da busca probabilística, em que os movimentos dos buscadores e dos alvos são independentes.

No contexto da busca adversária, uma situação comum é o projeto de estratégias que visam maximizar o desempenho de busca no caso em que os evasores possuem onisciência e velocidade ilimitada, enquanto os perseguidores apresentam diversas restrições. Tal cenário, inicialmente proposto por Parsons [20], é conhecido como adversário de pior caso, e a maior parte das abordagens exige a garantia de captura no processo de busca.

A principal vantagem da busca adversária de pior caso que visa garantia de captura é que se aplica simultaneamente a um ou vários evasores, não importando o tipo de comportamento que apresentem [19].

Devido às características do problema, a literatura científica faz a equivalência da busca e detecção de todos os evasores com o ato de *limpar* um ambiente previamente contaminado com um gás nocivo, usando o termo *o problema de limpeza (the clearing problem)* [2]. Assim, os buscadores podem limpar partes menores do ambiente enquanto restringem a propagação da contaminação para áreas já limpas [1].

Uma vez que sejam almejadas aplicações com um robô real, que possui sensoramento limitado em relação à dimensão do ambiente, a limpeza de quaisquer locais que não se limitem a corredores simplesmente conectados exige a coordenação de múltiplos robôs para cumprimento da tarefa, porque do contrário o evasor teria sempre as condições de recontaminar o ambiente.

Diversos trabalhos se empenharam em definir o número mínimo necessário de buscadores (para ambientes conhecidos) ou controlar o comportamento coletivo para que os mesmos atendam às restrições do cenário [19].

Alguns trabalhos mais recentes concentram-se na coordenação de múltiplos robôs para cumprir a tarefa de limpeza de ambientes não conhecidos previamente, e distintas abordagens já foram apresentadas para solução do problema em questão. Entre as principais, um algoritmo com formação de linhas de varredura [1], desenvolvimento de protocolos de comunicação e controle para cobertura de arcos orientados de fronteiras [2] e abordagem baseada em *random walk* em conjunto com restrições locais [5].

Dentre as publicações encontradas a respeito da busca adversária de pior caso, em especial as que utilizam múltiplos perseguidores em ambientes desconhecidos, não constam referências à utilização de algoritmos evolutivos como método de solução.

Abordagens evolutivas têm demonstrado benefícios em uma ampla gama de aplicações, e possuem potencial para construir sistemas autônomos para problemas de muitas dimensões e que sejam capazes de lidar com novas informações não previstas durante o projeto, o que desperta interesse para investigar e expandir suas aplicações. Por isso uma técnica deste tipo foi escolhida para o desenvolvimento deste trabalho, e uma análise preliminar desta abordagem foi apresentada em [21].

A robótica evolutiva é uma área que se inspira na evolução biológica para construir e controlar robôs móveis autônomos [22]. Trata-se de uma técnica de busca adequada a problemas cujo espaço de soluções é amplo e tem se mostrado promissora também na

investigação de soluções não intuitivas ao projetista [23].

Com um amplo histórico de pesquisas envolvendo coordenação e competição, a robótica evolutiva estuda há décadas o problema de perseguição-evasão, neste contexto também chamado predador-presa, em esforços destinados principalmente à abordagem competitiva através da coevolução [24]. Neste cenário, predadores e presas evoluem alternadamente suas habilidades e encontram o adversário mais adaptado, de forma que a melhoria na capacidade de um grupo perseguir exerce pressão para a melhoria da capacidade do outro grupo evadir e vice-versa.

Apesar da relação antiga da robótica evolutiva com jogos de perseguição-evasão, suas técnicas ainda não foram aplicadas na solução cooperativa do problema de busca adversária de pior caso, cenário desta dissertação. Apesar disso, a robótica evolutiva tem obtido bons resultados em pesquisas de cooperação entre múltiplos robôs [25, 26], apresentando comportamentos robustos e eficazes, características de interesse para o problema em questão.

O objetivo deste trabalho, portanto, é a proposição de uma solução baseada em robótica evolutiva para o problema de perseguição-evasão de pior caso, onde múltiplos robôs com sensoriamento e comunicação limitados devem coordenar-se para garantir a captura de evasores oniscientes e de velocidade ilimitada. A tarefa corresponde à limpeza total de um ambiente de busca plano, que pode ser multiplamente conectado, limitado, previamente desconhecido, contaminado e sujeito à recontaminação imediata.

Este trabalho está dividido da seguinte forma: no Capítulo 2 é fornecida uma revisão sobre o problema de perseguição-evasão; no Capítulo 3 são apresentados os principais conceitos da robótica evolutiva e dos algoritmos genéticos, ferramenta aplicada no processo de evolução; o Capítulo 4 contém os detalhes da abordagem proposta para esta pesquisa; no Capítulo 5 são discutidos os resultados obtidos; no Capítulo 6 são apresentadas as conclusões e por fim são listadas as referências bibliográficas.

Capítulo 2

Perseguição-Evasão

Neste capítulo serão apresentados alguns trabalhos principais correlatos ao problema de perseguição-evasão em áreas de estudo cujo contexto tem relação com esta pesquisa, com especial atenção aos esforços que contribuem mais especificamente para aplicações em robótica.

Na seção 2.1 são apresentadas algumas das primeiras formulações do problema, contrapondo abordagens contínuas e discretas. A seção 2.2 contém a revisão dos trabalhos em que os jogos ocorrem em grafos e a seção 2.3 trata das abordagens baseadas em visibilidade. Na seção 2.4 são apresentados os trabalhos de busca adversária de pior caso voltados a ambientes desconhecidos, com o trabalho coordenado de múltiplos perseguidores, diretamente relacionados ao cenário abordado neste trabalho.

2.1 Origens

O primeiro registro encontrado a respeito do problema de perseguição-evasão é a proposição em 1925, pelo matemático Richard Rado, do jogo *Lion and Man*, um caso geométrico clássico de perseguição-evasão, cujas variações ainda são investigadas nos dias atuais [27], em que um leão tenta capturar um homem, ambos com a mesma velocidade, em uma arena fechada.

Em 1951, Rufus Isaacs [28], inauguraria uma série de trabalhos que exploram diversas análises geométricas de problemas gerais relacionados informalmente à perseguição-evasão com o viés da teoria dos jogos diferenciais, onde os movimentos dos jogadores são contínuos e modelados como equações diferenciais.

Dois dos casos clássicos apresentados foram *The Homicidal chauffeur*, em que um carro rápido e com determinado raio de giro tenta atingir um pedestre, e *Princess and Monster*, em que um monstro tenta capturar uma princesa em um quarto escuro com forma arbitrária [15]. Nos cenários propostos, Isaacs investigou abordagens que buscavam a otimização em relação a aspectos como o ato da captura e o tempo (ou iterações) gasto,

a depender do modelo, sempre considerando perseguidor e evasor com total conhecimento das circunstâncias e do ambiente.

As proposições de Isaacs visavam principalmente aplicações de interesse militar e utilizavam sempre representações geométricas para os ambientes. Trabalhos posteriores, principalmente de Parsons e Petrov [18], passaram a investigar soluções com representação topológica e inaugurariam uma grande e ativa área de pesquisa denominada busca em grafo (*graph searching*), que investiga uma grande variedade de problemas combinatórios relacionados a capturar um fugitivo residente num grafo utilizando o número mínimo de buscadores [17].

A linha clássica de tais pesquisas tem o foco voltado à garantia de captura, sendo por isso referenciada como busca garantida (*guaranteed search*), e está diretamente relacionada aos problemas clássicos de perseguição-evasão, sendo possível encontrar termos como buscador, perseguidor, policial, caçador ou robô, para representar o agente que deseja capturar, e evasor, fugitivo, ladrão, caça ou alvo, para referenciar o agente que evita ser capturado [17].

2.2 Perseguição-Evasão em Grafos

A primeira formulação matemática da busca em grafos foi introduzida por Torrence Parsons, em 1978 [20], buscando solucionar um problema de perseguição-evasão proposto por um amigo espeleologista anos antes, quando perguntou pelo número mínimo de buscadores necessário para encontrar um homem perdido em uma caverna escura de estrutura conhecida, independentemente do comportamento adotado pelo homem e ainda que o mesmo tenha velocidade ilimitada, intenção de não ser encontrado e conhecimento total do posicionamento e movimento de cada buscador.

Nikolai Petrov redescobriu o problema e alguns resultados de Parsons de forma independente na União Soviética a partir de 1982, estudando a busca em grafo como uma restrição natural dos jogos diferenciais em espaços Euclidianos (estudados por Isaacs), tendo sido provado posteriormente que as formulações de Parsons e Petrov são equivalentes [17]. Petrov estudou por mais de 30 anos a busca garantida e em [18] encontra-se uma revisão detalhada de suas realizações.

Um grande número de trabalhos deriva-se das primeiras proposições de busca em grafo, gerados principalmente a partir de diversas restrições ou expansões nas habilidades de perseguidores e evasores. A formulação de Parsons foi a primeira a considerar o adversário de pior caso. Nesta formulação do problema, busca-se a melhor sequência de movimentos que permite a limpeza das arestas de um grafo com o menor número de buscadores.

O número mínimo de buscadores para completar a limpeza do grafo com sucesso foi chamado por Parsons de *edge search number*, e a formulação é atualmente conhecida como *edge searching*, porque o grafo é visto como um sistema de túneis, representados

pelas arestas, onde os intrusos estão escondidos.

Na estratégia de Parsons, os movimentos possíveis são: (i) colocar ou (ii) retirar um buscador para guardar um vértice e (iii) deslizar através de uma aresta até o vértice vizinho para limpá-lo da contaminação. Assim, enquanto houver um buscador no vértice ele é considerado guardado e nenhuma contaminação pode se espalhar através dele; por outro lado, se em algum momento houver um caminho livre (não-guardado por um buscador) entre uma aresta já limpa e uma contaminada, a contaminação se espalha imediatamente (devido à velocidade ilimitada do evasor), o que é chamado de recontaminação.

A primeira variação desta abordagem para busca em grafo foi proposta por Kirousis e Papadimitriou [29], com regras mais simples, sendo permitido apenas colocar e retirar um buscador de um vértice, onde uma aresta é considerada limpa se há um buscador em cada vértice extremo (como numa armadilha). Nesse caso, o número mínimo de buscadores para limpar o grafo foi chamado de *node search number*, apesar de a contaminação também estar localizada nas arestas do grafo. Outra variação relevante foi chamada de *mixed searching* [30], unificando as duas anteriores.

Ao tentar relacionar as formulações tradicionais da busca garantida em grafos à aplicação em problemas reais, especialmente em robótica, algumas características apresentam severas desvantagens. Duas delas estão relacionadas ao movimento: é permitido que os buscadores se *teletransportem* entre os nós, ou seja, migrem entre os ambientes sem que haja um caminho direto entre eles, e também que eles sejam retirados do grafo a qualquer instante.

Em aplicações com robôs reais, conectividade e permanência do agente são características importantes, uma vez que eles não podem simplesmente desaparecer. Ao invés disso, precisam de um caminho seguro entre localizações físicas.

Interessados nestas restrições, Barrière *et al.* [31, 32] propuseram a ideia de busca conectada (*contiguous search*) em árvore, onde os perseguidores precisam manter um subgrafo conectado de nós já explorados, o que garante um caminho contínuo a partir do nó inicial, e também aplicando a propriedade de busca interna, em que a remoção dos buscadores não é permitida.

Além disso, eles também introduziram uma variante ponderada da *edge searching*, onde são determinados (e contabilizados) pesos que representam o número necessário de robôs para guardar um vértice ou deslizar por uma aresta, já que estes podem representar ambientes de tamanhos variados que requerem mais ou menos agentes para serem explorados.

Outra desvantagem com relação às formulações tradicionais da busca garantida para aplicações com agentes físicos que vale ser mencionada, diz respeito ao fato dos evasores se esconderem nas arestas do grafo. Em ambientes reais, internos e externos, os possíveis caminhos de um evasor frequentemente não podem ser precisamente representados como arestas em um grafo [19].

Neste sentido, mais recentemente Hollinger *et al.* [33] discutiram as propriedades da busca adversária para os casos em que os evasores residam nos nós ao invés de nas arestas, e encontraram uma relação formal entre as duas situações, afirmando que “qualquer estratégia de limpeza para *node search* é também uma estratégia de limpeza para *edge search*, mas o contrário não é verdadeiro”. Neste trabalho não foi considerado o adversário de pior caso.

Outro trabalho a ser mencionado a respeito da busca garantida em grafos, cuja base teórica é especialmente relevante para aplicações em robótica, é o de Kolling e Carpin [34, 35], que apresentam uma formulação referida como *graph-clear*, uma modelagem formal de perseguição-evasão em grafos, útil para representar tarefas como patrulha e vigilância, em que múltiplos robôs com capacidades sensoriais restritas são coordenados para detectar intrusos.

Trata-se de uma abordagem que conecta explicitamente aplicações robóticas reais e teoria de grafos, já que robôs podem executar ações como limpar um cômodo e guardar portas que dividem os ambientes.

A formulação tem relação com o *edge searching* ponderado, com a principal diferença de que neste último as implementações estão relacionadas a guardar um vértice e deslizar por uma aresta, enquanto no *graph-clear* implementam-se ações de varredura em vértices e bloqueio de arestas, que representam apenas conexões entre vértices.

Como cada robô possui limite de sensoriamento, mais de um pode ser necessário para executar cada uma destas tarefas e o conceito dos pesos se aplica, contudo, devido à diferença na requisição das implementações, e também porque no *graph-clear* são autorizados movimentos simultâneos, os pesos do grafo são diferentes dos encontrados para o *edge searching*.

A formulação teórica do *graph-clear* se estendeu para abarcar contextos ainda mais particulares da robótica, como em [35], em que são consideradas falhas probabilísticas ao sensoriamento, visando possibilitar a aplicação em ambientes dinâmicos e complexos.

2.3 Perseguição-Evasão Baseada em Visibilidade

Especialmente na comunidade de robótica, os pesquisadores buscaram estudar formulações para problemas de busca adversária em representações diferentes de grafos, a fim de modelar melhor os ambientes reais [19]. Diversos estudos consideram representações geométricas, em especial em ambientes poligonais.

A uma destas importantes abordagens dá-se o nome de problema de perseguição-evasão baseado em visibilidade, cujo foco é a consideração de robôs com sensores de alcance ilimitado em ambientes de duas dimensões. Os trabalhos incluem variações no tipo de ambiente, restrições de movimento, campo de visão do sensor e número de robôs disponíveis [1].

O primeiro trabalho considerando robôs móveis foi realizado por Suzuki e Yamashita [36], que investigaram as condições de busca em um polígono simples por um perseguidor equipado com um número limitado k de feixes de sensores de alcance infinito, chamado de *k-searcher*.

Posteriormente, LaValle *et al.* [37] consideraram o caso de coordenação de movimento de um ou mais ∞ -*searchers* ($k = \infty$, ou seja, robôs cujo sensoriamento é omnidirecional), em um ambiente poligonal, com o objetivo de garantir que por fim o evasor se encontre em pelo menos uma das zonas de visibilidade. Além disso, determinaram o cálculo do número necessário de perseguidores para determinado ambiente.

Em [38] foi apresentado um algoritmo completo a esse respeito para limpeza do ambiente poligonal com um único buscador, mas mostrou pouca escalabilidade para muitos buscadores e ambientes complexos. Mais recentemente, um algoritmo centralizado [39] foi apresentado como generalização para múltiplos ∞ -*searchers*.

Ainda em busca de particularidades de aplicações práticas, outros cenários vêm sendo estudados, onde são acrescentadas restrições como limitação de velocidade ou ambiente desconhecido, já que em aplicações do mundo real os mapas nem sempre estão disponíveis e os sensores para criá-los em detalhe não são perfeitos.

Uma grande culminância desses esforços foi o trabalho de Sachs *et al.* [40], em uma versão *online* de limpeza de um ambiente desconhecido, plano e simplesmente conectado, através da construção incremental de um grafo. O perseguidor tem movimentos simples, sem capacidade de localização e sem controle perfeito, dotado de sensores capazes de informar direção de descontinuidades, sem medição perfeita de profundidade.

Foi demonstrado que o algoritmo completo permite ao perseguidor limpar os mesmos ambientes que um possuidor do mapa completo e perfeita localização. São também dadas garantias teóricas de que o evasor será encontrado.

Uma vez que nas pesquisas em perseguição-evasão baseada em visibilidade considera-se robôs com sensores de alcance ilimitado, a maioria dos esforços foram despendidos para encontrar soluções para apenas um robô, ou vários, porém sem preocupação com a sincronia de posicionamento.

Considerando, contudo, os limites reais de sensoriamento e comunicação entre buscadores, é inevitável a preocupação com a coordenação do movimento de múltiplos perseguidores. A próxima seção, então, é dedicada a trabalhos que consideram o problema de perseguição-evasão de pior caso, em ambientes planos desconhecidos, com múltiplos perseguidores de sensoriamento limitado. Tais pesquisas, de alguma forma, serviram de inspiração para o desenvolvimento deste trabalho de mestrado.

2.4 Coordenação de Múltiplos Perseguidores em Ambientes Desconhecidos

No cenário em questão, o primeiro trabalho relevante encontrado foi realizado por Kolling e Carpin [1], que formularam a abordagem *Line-Clear*, uma técnica que organiza os perseguidores em múltiplas linhas de varredura, geradas a partir de obstáculos, e coordena tais linhas sem precisar de um mapa métrico.

Os autores foram os primeiros a assegurar a detecção de evasores para um time de robôs sem conhecimento a priori do ambiente. Isso porque, considerando o sensoriamento limitado de cada robô, só é permitido o avanço dos buscadores se o posicionamento entre eles não exceder o dobro do raio do sensor ou o próprio valor do raio entre o buscador e um obstáculo.

Trata-se de uma abordagem distribuída em que os robôs seguem paredes ou outros robôs que estejam próximos ao executar um de três algoritmos, a depender do papel que exercem na formação. As possíveis funções que os robôs podem exercer estão exemplificadas na Figura 2.1, em que os pontos representam os buscadores e os círculos suas regiões de sensoriamento.

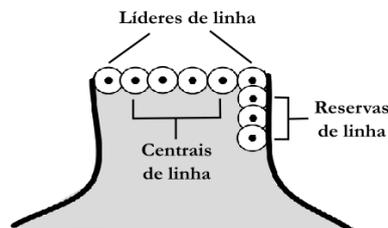


Figura 2.1: Composição dos robôs na linha de varredura (adaptado de [1])

Os líderes de linha pertencem às duas extremidades da linha de varredura e têm a função de seguir paredes, portanto, são os responsáveis por conduzir a formação. Os centrais de linha se mantêm distribuídos na linha de varredura entre os líderes de linha, e os reservas de linha seguem um líder de linha, permanecendo na retaguarda enquanto as passagens são estreitas e atendendo à formação quando a passagem é aumentada, quando então se tornam centrais de linha.

Os robôs coordenam seus movimentos de forma a cobrir cada linha entre paredes opostas, e avançam até descobrir novos obstáculos. Quando encontram, a formação é dividida em duas novas linhas e os robôs têm que decidir qual linha vão seguir.

Assim, vai sendo construído e armazenado um mapa topológico do ambiente descoberto, descrevendo possíveis movimentos de linha e seus custos associados em termos de robôs. A coordenação requer apenas comunicação local entre os líderes de diferentes linhas de robôs quando eles se encontram.

Se em determinada situação o número de robôs for insuficiente para cobrir a linha e nenhum obstáculo for encontrado, a formação retrocede e é executado um procedimento de busca com o qual os robôs tentam se deslocar para obstáculos desconhecidos através da coordenação das linhas, associando custos a uma sequência de movimentos, como mostrado na Figura 2.2.

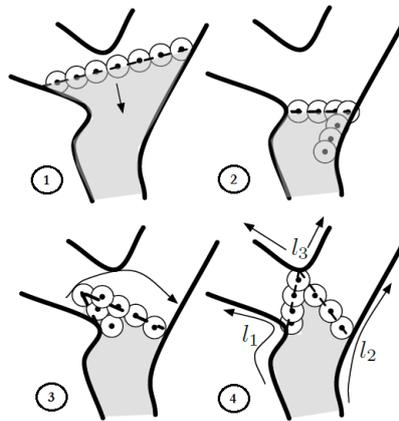


Figura 2.2: Busca por obstáculo e formação de nova linha [1]

Na primeira imagem da figura a formação é insuficiente para continuar a se mover à frente e retrocede no sentido da seta para a área que já havia sido descontaminada (indicada em cinza). A cada reserva de linha disponível o algoritmo de busca de obstáculo é executado. A segunda imagem mostra o recuo final da formação, para logo após haver sucesso na busca por um novo obstáculo, com o movimento no sentido da seta da imagem 3, quando então a formação se parte e as linhas podem seguir individualmente, como indicado na quarta imagem.

Nesta abordagem não há preocupação com o número mínimo de robôs, tendo em vista tratar-se de um ambiente desconhecido. E também por isso, não se garante uma forma ótima de limpar o ambiente ou de lidar com os obstáculos. Não existe prova de convergência, com garantias teóricas para a descontaminação do ambiente, e o algoritmo não é bem adequado a ambientes multiplamente conectados.

Posteriormente, um trabalho com resultados melhores foi publicado por Durham *et al.* [2], que também abordam o problema de perseguição-evasão de pior caso em que uma equipe de robôs móveis, com sensores e capacidade de comunicação limitados, precisa se coordenar para detectar os evasores de um ambiente plano, desconhecido, que pode ser multiplamente conectado e não-poligonal. Isso é feito desenvolvendo-se protocolos de comunicação e controle para a varredura.

O algoritmo distribuído proposto oferece garantia teórica de captura para um número suficiente de buscadores, sem construir ou armazenar nenhum tipo de mapa do ambiente e sem precisar de localização global. Funciona mantendo uma cobertura completa das fronteiras entre ambientes limpos e contaminados, enquanto expande a limpeza das re-

giões. Quando não há um número suficiente de buscadores, o algoritmo limpa a maior quantidade de área possível, garantindo que nenhuma área seja recontaminada.

Os robôs são denotados como (d, ϕ) -searchers ou d -searchers, pois possuem um sensor de alcance limitado d e um campo de visão angular $\phi \in [\pi, 2\pi]$ ou $\phi = 2\pi$. Os buscadores armazenam e atualizam unicamente sua mais recente percepção e sua fronteira local.

A percepção dos sensores para um determinado posicionamento é formada por uma região livre de obstáculos S , chamada *footprint*, usada para definir a região guardada pelo robô. Na Figura 2.3 é exemplificado tal registro em uma configuração onde quatro obstáculos estão presentes.

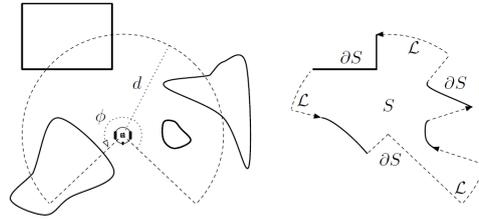


Figura 2.3: Exemplo de percepção dos sensores [2]

Na figura, os arcos orientados pontilhados dizem respeito à fronteira livre \mathcal{L} , enquanto os arcos sólidos representam as fronteiras com os obstáculos locais ∂S . Os robôs se comunicam quando há interseção de *footprints*. A união de todas as percepções dos robôs é chamada de região inspecionada e corresponde a uma área limpa, porque os robôs movimentam-se somente se não permitirem a recontaminação.

O método desenvolvido armazena e atualiza a fronteira global baseado nas interseções locais e nos arcos orientados, usando uma pequena quantidade de memória por robô. Além disso, também apresentam um método de planejamento de pontos de vista (*viewpoint*) que minimiza localmente o número de robôs necessários para expandir a limpeza da área.

Os buscadores podem ser designados a quatro tipos de atividade. Os guardas de fronteira são os robôs que permanecem no *viewpoint* e garantem a proteção dos segmentos de fronteira; expansores são destinados a alcançar o novo *viewpoint*, e então registrar um novo *footprint*; os seguidores seguem passivamente e respondem aos guardas de fronteira ou aos expansores e a última atividade é a de peregrino, quando um guarda de fronteira não tem mais uma fronteira para vigiar e, então, passeia para localizar um líder para seguir. Desta forma, os robôs sempre guardam os pontos da fronteira orientada, enquanto expandem o quanto possível a região limpa.

O trabalho apresenta ainda uma alternativa a possíveis falhas dos buscadores que são guardas de fronteira durante a execução do algoritmo, o que resultaria em recontaminação do ambiente. Para garantir a robustez do sistema, tais buscadores poderiam ser duplicados e manter constante comunicação, para que haja substituição do principal guarda em caso de falhas.

Os autores apresentam também resultados de experimentos com robôs reais, inclusive no caso de falha de um robô, deixando a cargo dos restantes o cumprimento da tarefa. Outro ponto relevante da pesquisa é sua adequabilidade para um número maior de buscadores.

Um trabalho posterior publicado nesta linha, proposto por Gonçalves *et al.* [5], apresenta uma abordagem discreta centralizada aplicando *random walk* em conjunto com restrições locais, em que os robôs têm sensoriamento e comunicação limitados.

O mapa foi discretizado com uma grade regular e cada célula que contém pelo menos um ponto representando um obstáculo é considerada ocupada. Os robôs somente são capazes de conhecer as células totalmente cobertas por um raio r_{sensor} e que não estejam ocultas pela presença de obstáculos.

Uma vez que exista obstáculo, ocorre um bloqueio de visão bastante restrito, chamado de *propagação de incertezas*, em que células adjacentes e posteriores permanecem como desconhecidas até que, por novas movimentações do robô, elas estejam completamente no ângulo de sensoriamento e sejam devidamente classificadas.

Foram propostas no trabalho três restrições para os perseguidores: a restrição física determina que o robô não poderá ir a uma célula que já esteja ocupada por um obstáculo ou por outro robô; a restrição dinâmica estabelece que os robôs só podem se mover para as quatro células imediatamente vizinhas ou permanecerem onde estão, como ilustrado na Figura 2.4(a); e a restrição estratégica é o controle das ligações entre robôs e entre robô e parede, a fim de não permitir a recontaminação, o que reduz significativamente o tempo de convergência do algoritmo.

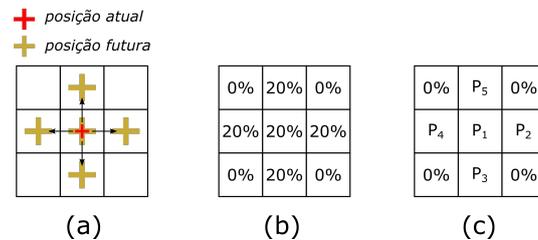


Figura 2.4: Ações de movimentação do trabalho de Gonçalves [3]: (a) restrição de movimentação do robô; (b) distribuição uniforme de probabilidades do *random walk* e (c) distribuição ajustada de probabilidades segundo células já visitadas

A movimentação dos robôs se dá de forma paralela a cada iteração, onde é feito o sorteio aleatório e uniformemente distribuído entre as células candidatas a serem os destinos, como na Figura 2.4(b), sendo então verificado por um controle central se as restrições foram atendidas. Em caso negativo, um novo sorteio é realizado até que seja permitida a movimentação dos perseguidores.

O algoritmo proposto funciona somente para ambientes simplesmente conectados e foi capaz de cumprir a tarefa de limpeza em todos os mapas apresentados. Posteriormente foi implementado um reajuste de probabilidades [3], ilustrado na Figura 2.4(c), para diminuir

o número médio de iterações para limpeza dos ambientes. Apesar de ter potencial para garantia de captura, não foi apresentada a prova de convergência.

Com o intuito de oferecer uma solução alternativa ao problema, o próximo capítulo apresenta a área de pesquisa Robótica Evolutiva, escolhida para o desenvolvimento desta dissertação, buscando também contextualizá-la ao cenário de perseguição-evasão.

Capítulo 3

Robótica Evolutiva

O comportamento de um robô é determinado pela interação entre seus aspectos morfológicos, aparelhagem sensorial, sistema motor e arquitetura de controle, sendo um grande desafio para a robótica a consideração simultânea de todos estes aspectos, já que usualmente eles são projetados de forma isolada [4]. Pesquisadores precisam antecipar quais habilidades um dado robô deve ter para cumprir um objetivo e então formulá-las em um programa ou hierarquia de controle, que podem ser bastante complexos [41].

Com um ponto de vista alternativo, em que os aspectos construtivos de um robô são considerados de forma holística, surgiu no início da década de 1990 a área de pesquisa denominada Robótica Evolutiva (*ER - Evolutionary Robotics*) [22, 23], que aplica conceitos abstraídos da teoria da evolução de Darwin no intuito de tentar projetar e construir robôs autônomos e adaptativos, em busca da eficiência apresentada pelos sistemas biológicos [22].

Enquanto a robótica tradicional objetiva projetar controladores e morfologias para que um dado robô cumpra tarefas específicas, eventualmente se utilizando de algoritmos de aprendizado de máquina para otimizar sua política de controle [22], pesquisas em ER trabalham no desenvolvimento de algoritmos que permitam evoluir o controlador (mais comumente) ou simultaneamente a morfologia para um robô autônomo, simulado ou real, muitas vezes sem pressupor antecipadamente a existência de um robô físico.

A ER distingue-se da robótica bioinspirada, caracterizada pela construção de robôs semelhantes a determinados animais, uma vez que esta usa a inspiração do resultado de bilhões de anos de evolução, enquanto que naquela o objetivo é recriar o processo evolutivo em si, e o resultado pode nem mesmo existir na natureza.

Integrando conhecimentos da robótica e da biologia, a ER fornece em contrapartida para estas áreas, respectivamente, resultados que podem contribuir para construção de melhores robôs e uma ferramenta científica para testar hipóteses que podem fornecer respostas de questões abertas sobre como e porque determinada característica ou habilidade cognitiva evoluiu [42]. Na Figura 3.1 está ilustrada a relação entre tais pesquisas.

A robótica evolutiva também divide interesses com algumas áreas correlatas da ciên-

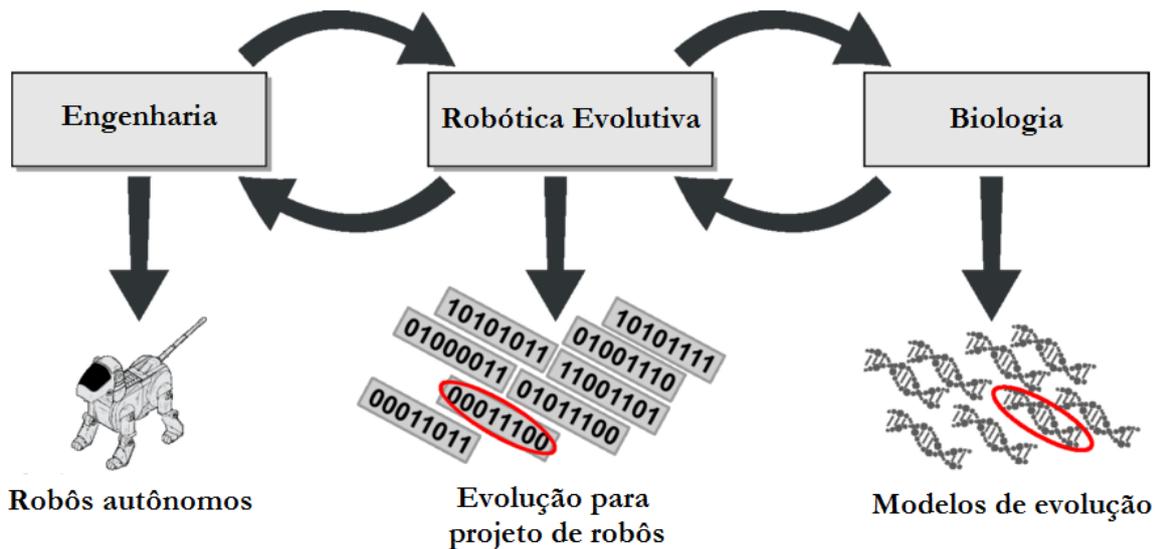


Figura 3.1: Relação entre engenharia, robótica evolutiva e biologia [4]

cia experimental, como a biorrobótica evolutiva [22] e a robótica desenvolvimental [43]. Trabalhos nestas áreas utilizam modelos físicos para investigar desenvolvimentos biológicos e simultaneamente servem à robótica, no sentido de sugerirem ideias para construir melhores robôs.

A metodologia da robótica evolutiva baseia-se na utilização das técnicas de Computação Evolutiva, em especial dos Algoritmos Genéticos, para desenvolver ou otimizar sobretudo o sistema de controle de robôs móveis.

O que principalmente caracteriza um trabalho em robótica evolutiva, diferenciando-o da tradicional aplicação dos algoritmos genéticos, é a ideia da cognição incorporada [44], que traduz o conceito de que a inteligência não é somente algo contido no cérebro de um animal ou na política de controle de um robô, mas ao invés disso é algo que emerge da interação entre cérebro, corpo e ambiente.

Em outras palavras, isso quer dizer que para avaliar uma solução durante o processo evolutivo, a mesma precisa ser incorporada em uma morfologia ou sistema de controle real ou simulado, resultando em determinado comportamento, que então poderá ter sua qualidade avaliada, como será detalhado em capítulo posterior.

Quanto ao sistema de controle a ser evoluído, o paradigma predominante consiste no uso de redes neurais artificiais [4, 45]. Isto porque, entre outras vantagens, elas permitem diferentes níveis de adaptação, são robustas a ruídos e podem oferecer um espaço de busca relativamente suave, onde pequenas alterações nos parâmetros frequentemente correspondem a pequenas mudanças no comportamento [46], dependendo do tipo e das parametrizações definidas.

As aplicações envolvem desde as tradicionais redes *feedforward* de elementos simples,

em que os pesos dos neurônios ou as regras de aprendizado são evoluídos, até as redes recorrentes dinâmicas e plásticas relativamente complexas, em que toda a estrutura da rede está sujeita à evolução [45].

Por outro lado, compreender e validar redes neurais resultantes dos processos evolutivos é algo extremamente problemático [47], e por isso encontram-se também pesquisas em robótica evolutiva baseadas em outras abordagens, como as máquinas de estado finito.

Em [48], por exemplo, é proposta a evolução de um autômato do tipo Moore para navegação de robôs móveis; em [47] é feita uma comparação entre evolução de controladores baseados em redes neurais artificiais e em um autômato do tipo Mealy para um tarefa de busca por fonte de luz e em [49] é analisado o impacto da coevolução cooperativa para autômatos do tipo Mealy e Moore para o problema *tartarus*.

As pesquisas caracterizadas como robótica evolutiva datam do início da década de 1990 [45], e desde então diversas investigações têm sido realizadas, sob distintos aspectos possíveis de serem explorados na área.

Nestas décadas, foram evoluídos com sucesso robôs que exibem diferentes funcionalidades, dos terrestres aos que voam [23, 50], e também foram apresentados trabalhos que combinam evolução com o conceito de aprendizado [4], tornando-se bem estabelecida para aplicações simples. Além disso, seus esforços têm sido ainda aplicados em linhas de pesquisa interessantes, como:

- Síntese automática de morfologias [51], para projeto e manufatura automática de robôs;
- Robótica *soft* [52], que incorpora materiais macios no projeto de robôs, explorando o potencial de reduzir a complexidade algorítmica e mecânica envolvidas;
- Robótica modular [53], em que robôs individuais podem se juntar e separar dinamicamente em uma constante mudança de formas, permitindo que eles se adaptem a novas circunstâncias, realizem novas tarefas e recuperem-se de danos;
- Robótica de enxames [54], que tem o desafio geral de definir uma política de controle que faça com que um grupo grande de robôs apresente determinado comportamento coletivo, cumprindo tarefas que um robô individual pode não ser capaz de realizar.

No que concerne ao último tópico, que mais tem relação com este trabalho, é válido destacar que a natureza distribuída dos enxames torna os projetos desafiadores, no sentido de que o comportamento coletivo precisa ser interpretado para que seja possível caracterizar o papel de cada um dos robôs individualmente.

Como destacam Francesca *et al.* [55], a tarefa do projetista é indireta, uma vez que ele deve designar as ações em nível individual, ao mesmo tempo que considera as relações entre robôs e entre robô e ambiente, a fim de obter o comportamento coletivo desejado, e não há ainda um procedimento preciso para tal finalidade [56].

Neste sentido, os autores em [55] apontam que o projeto tradicional da robótica de enxames é em sua maioria baseado em tentativa-e-erro, cujo sucesso está fortemente relacionado à experiência do projetista, além de ter alto consumo de tempo e desvantagens em relação à repetibilidade e consistência.

Os desafios no projeto de enxames e a dificuldade em estabelecer as aplicações no mundo real [56] abre espaço para pesquisas alternativas, e a robótica evolutiva, com sua proposta de projeto automático, pode ser usada para projetar enxames capazes de atuar em alguns tipos de tarefas, apesar de ainda serem grandes seus desafios [57, 58].

A robótica de enxames é caracterizada pela coordenação de um grande número de robôs, ou pelo menos pela busca de escalabilidade [59]. Quando as pesquisas são realizadas com um pequeno número de robôs, o termo mais comumente utilizado é o MRS, onde se encaixa a maior parte dos trabalhos em perseguição-evasão.

O problema de perseguição-evasão é abordado há anos pela robótica evolutiva, também sob o nome predador-presa [24, 60–62]. Contudo, o contexto sempre foi o de desenvolver e estudar comportamentos coletivos cooperativos [63] e principalmente competitivos [64, 65] através da coevolução.

Na coevolução, a habilidade de um grupo em superar um segundo grupo faz com que seja provável que este segundo grupo evolua uma defesa contra o grupo original, o que exerce uma pressão para que o grupo original desenvolva uma nova estratégia, e assim por diante. Com isso, o comportamento do grupo de presas interfere diretamente na estratégia do grupo predador.

Já no problema de perseguição-evasão relacionado à busca adversária de pior caso, que visa a garantia de captura, tema desta pesquisa, os perseguidores precisam realizar uma espécie de varredura nos ambientes porque, uma vez que os evasores são oniscientes e de velocidade ilimitada, eles serão encontrados sempre na última área a ser visitada.

Para o caso apresentado, o comportamento a ser evoluído é o de cooperação entre os perseguidores, para que desenvolvam a capacidade de interagir de forma a varrer ambientes diversos em uma formação que não permita recontaminação.

Neste sentido, as pesquisas do comportamento cooperativo realizadas pela robótica evolutiva têm mostrado que robôs evoluídos desenvolvem comportamentos robustos e eficazes e podem exibir habilidade em diferenciar seu comportamento para melhor cooperar [25, 26].

Por outro lado, algumas limitações enfrentadas na ER dificultam a larga utilização de suas técnicas, principalmente para fins de engenharia de sistemas robóticos. Uma das principais limitações da robótica evolutiva é o *reality gap* [66], que ocorre quando o sistema de controle é evoluído em simulação e só é transferido para o robô real quando uma boa solução tiver sido encontrada.

Como a simulação é uma abstração da realidade, a precisão do modelo do robô e das suas interações com o ambiente é limitada, o que pode resultar em controladores com um

pior desempenho no mundo real.

A solução óbvia para o problema seria a condução totalmente embarcada do processo evolutivo, ou seja, realizar a evolução com a atuação real do robô em seu ambiente de trabalho. Isto, porém, diz respeito a outro problema em ER: o tempo necessário para a evolução do controlador no robô real costuma ser proibitivamente longo [67].

Algumas técnicas foram então desenvolvidas ao longo dos anos para minimizar o efeito do *reality gap*. Entre as principais ações estão [4]: inserir ruídos independentes, restringir o modelo a características consideradas relevantes, coevoluir controlador e parâmetros de simulação, aplicar evolução *online* com aprendizado e oferecer abordagem de transferibilidade.

Outro grande desafio diz respeito à definição da função de custo que guia o processo evolutivo. Toda a experiência do projetista que pode ser dispensada em considerações específicas dos projetos é, neste tipo de abordagem, direcionada à decisão sobre esta função.

As dificuldades em torno deste ponto se devem à influência que tal função tem na definição do que seja uma boa solução para o problema em questão, e uma situação comum recai em assumir que o aumento da performance em determinado aspecto conduzirá a busca na direção do comportamento desejado, o que nem sempre é verdade [41, 68].

A função de custo é o cerne da computação evolutiva, e é utilizada tanto para definir o objetivo quanto para guiar a busca. Em robótica evolutiva, porém, o processo de otimização não precisa ser do tipo *caixa-preta*, porque os experimentos têm desafios e configurações em comum, e novas formas de conduzir a evolução (o que é chamado de *exercer pressão seletiva*) têm sido exploradas nas pesquisas [68].

Conduzir o processo evolutivo através de outras técnicas de pressão seletiva (que ideologicamente são, em conjunto, uma complexa função *fitness*) também não é trivial, porque o comportamento do robô quase nunca é definido por uma única habilidade. Já se sabe, contudo, que ter atenção à pressão seletiva é pelo menos tão importante quanto cuidar do processo de codificação da solução [68].

A coevolução é uma das mais antigas técnicas aplicadas, e entre as recentes estão as pressões dirigidas pelo ambiente, na tentativa de replicar a evolução sem fim (*open-ended*) que realmente ocorre na natureza [69].

Outra forma de pressão seletiva que muito tem se destacado é a busca por novidade (*novelty search*), em que o processo de busca é também guiado por medidas de quanto uma nova solução difere de soluções geradas anteriormente. Os resultados são promissores para alguns tipos de tarefas, a ponto de em algumas pesquisas ela ser a única responsável por guiar a busca do processo evolutivo [70].

Outros pontos apontados como desvantagem na aplicação da ER recaem na falta de padronização nas práticas de pesquisa [41, 71], e nas falhas ao abordar experimentos que contam com um pouco mais de complexidade [57]. Tais críticas visam conduzi-la

à aplicação de metodologias mais rigorosas e objetivam o amadurecimento da área de pesquisa.

A robótica evolutiva ainda não produz resultados melhores do que as técnicas mais tradicionais ou do que as de otimização como aprendizado por reforço, por exemplo, mas uma vez que o projeto manual de robôs móveis autônomos e adaptativos ainda é desafiador, abordagens evolutivas oferecem grandes benefícios em potencial [72], trazendo inclusive bons resultados em aplicações reais em uso conjunto com técnicas bem estabelecidas, como abordagens baseadas em comportamento [73].

Assim, considerando-se o grande espaço de soluções a ser investigado em tarefas com múltiplos robôs e por tratar-se de uma pesquisa preliminar importante para futuras investigações com grupos heterogêneos de robôs e em tarefas em ambientes tridimensionais [74, 75], cujo espaço de soluções é ainda maior, acredita-se ser relevante a pesquisa de aplicação da robótica evolutiva como possível solução para o pior caso de perseguição-evasão em busca adversária.

A seção a seguir trata da fundamentação teórica do algoritmo evolutivo utilizado para ajustar o sistema de controle proposto neste trabalho.

3.1 Algoritmos Genéticos

Processos de otimização dizem respeito à busca da melhor solução para um problema dentre todas as soluções possíveis, contidas no espaço de soluções, com mínima interferência humana [76].

Inspirados no desempenho dos organismos vivos e na teoria da evolução das espécies proposta por Charles Darwin, pesquisadores desenvolveram modelos para reproduzir em computador a metodologia de busca por soluções realizada pela natureza.

Os Algoritmos Genéticos (AGs), formalmente inaugurados no estudo de John Holland [77] estão entre os modelos computacionais de evolução (chamados algoritmos evolutivos) mais comumente aplicados, inclusive no contexto da robótica evolutiva [23].

Segundo [76], trata-se de uma técnica de busca classificada como aleatório-guiada, já que possui componentes aleatórios, mas usa as informações do estado atual para guiar a busca.

Seu uso é adequado a diversos problemas de otimização que não podem ser solucionados por métodos exatos ou técnicas numéricas tradicionais, apresentando vantagens [78] como simplicidade de conceito, paralelismo, robustez e fácil aplicação, inclusive em conjunto com outras ferramentas. Além disso, são uma abordagem alternativa a problemas complexos, cujos modelos são difíceis de serem obtidos.

A teoria da seleção natural de Darwin sugere que os indivíduos mais aptos têm maiores chances de obter recursos de sobrevivência e gerar descendentes para as próximas gerações, enquanto os menos aptos geram menos descendentes e tendem à extinção.

A reprodução computacional dessa teoria faz com que soluções candidatas sejam metaforicamente consideradas indivíduos, classificados como mais ou menos adequados à solução de determinada tarefa, e como consequência tenham ou não suas características conduzidas às próximas gerações.

O algoritmo atua em uma população de indivíduos, que são pontos no espaço de busca a ser explorado. A representação dos indivíduos se dá através do genótipo, ou seja, estrutura de codificação dos atributos (genes) em um vetor (cromossomo) que caracteriza a solução.

Durante uma geração, cada indivíduo da população é avaliado por uma função de adaptação, também chamada de função de *fitness*, aptidão ou custo, que deve ser capaz de qualificar o indivíduo segundo sua capacidade de resolver o problema para o qual busca-se a solução. Ao longo deste trabalho, o termo *função de fitness* ou simplesmente *fitness* será majoritariamente utilizado para designar a avaliação dos indivíduos.

Para formar a população da próxima geração, operadores genéticos são aplicados aos cromossomos selecionados como pais, com maior probabilidade para os melhor avaliados, com o objetivo de que, com o passar das gerações, sejam encontradas melhores soluções. Tal procedimento é repetido até que a solução ideal surja ou até que alguma outra condição seja satisfeita. O esquema geral do processo evolutivo está mostrado na Figura 3.2.

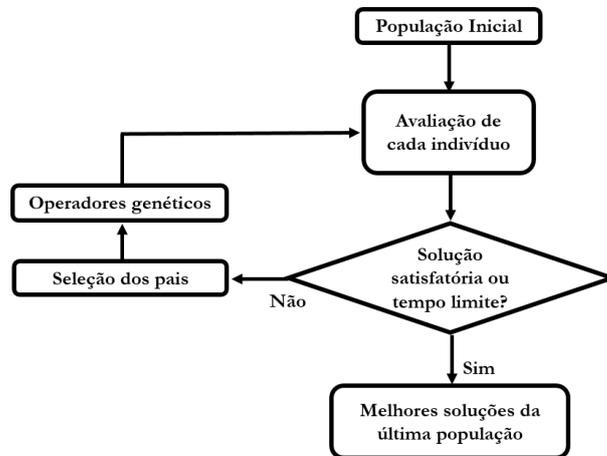


Figura 3.2: Estrutura geral de um algoritmo genético

O algoritmo parte de uma população de cromossomos geralmente inicializada de forma pseudoaleatória. O tamanho a ser escolhido para a população tem relação com a complexidade do problema, e a variabilidade dos indivíduos da primeira população é importante para garantir uma boa exploração do espaço de soluções. Na forma mais simples de aplicação, a população tem um tamanho fixo, e os filhos não convivem com os pais na geração.

O tipo de representação para o cromossomo também depende do problema abordado e, se devidamente escolhido, pode já incluir as restrições do problema [76]. Dentre as formas mais comuns estão os vetores de *bits* e os de valores reais, ambos exemplificados

na Figura 3.3.

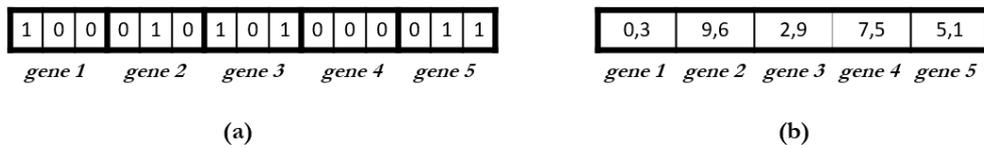


Figura 3.3: Exemplo de representação cromossomial em (a) *bits* e (b) números reais

O cromossomo deve ter uma representação simples e ser capaz de codificar qualquer ponto do espaço de soluções, para que exista a possibilidade de a solução adequada ser encontrada. Nele podem estar representados, por exemplo, os coeficientes de uma função, os pesos ou a arquitetura de uma rede neural artificial, os estados ou as transições de um autômato, as regras e funções de pertinência de um sistema nebuloso, e tudo o que possa caracterizar a resposta para um dado problema.

Assim como ocorre na vida biológica, o algoritmo deve ser capaz de decodificar o genótipo em fenótipo, que é a expressão do cromossomo nos termos do modelo, porque, ainda que a evolução seja baseada em operações sobre o genótipo, a seleção é sempre realizada com base no fenótipo [78].

Nos termos da robótica evolutiva, diferentemente da tradicional aplicação dos AGs, há ainda mais um passo intermediário entre o fenótipo e a avaliação da solução: o comportamento que emerge da interação entre robô e ambiente. Ou seja, se a decodificação do cromossomo (fenótipo) representa uma parametrização de um sistema de controle, tal sistema precisa ser efetivamente testado no robô, com a realização da tarefa de interesse. Aí então a solução candidata pode ter seu desempenho avaliado. Por meio da Figura 3.4 é possível identificar a importância da interação entre robô e ambiente.

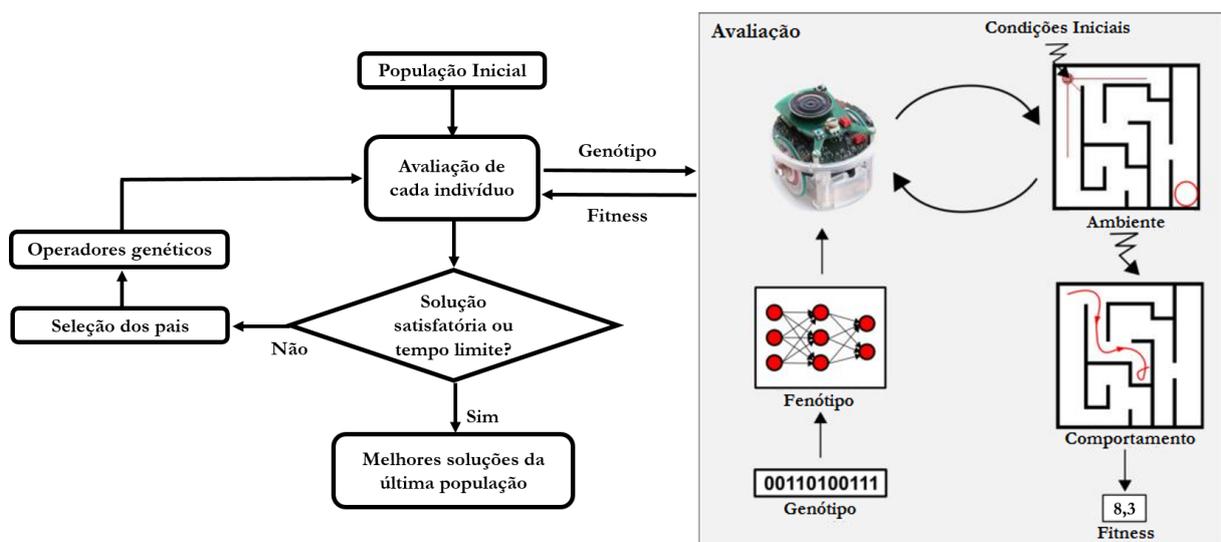


Figura 3.4: Estrutura de um AG no contexto da robótica evolutiva (adaptado de [4])

Na figura, o sistema de controle do robô, seja ele real ou simulado, é representado por uma rede neural artificial (RNA), o fenótipo. O genótipo (neste caso, um cromossomo binário) caracteriza a RNA através da codificação dos pesos, das regras ou da topologia, e é sujeito à evolução.

O fenótipo por si só não traduz a representatividade da solução, por isso o processo tradicional dos AGs $genótipo \rightarrow fenótipo \rightarrow fitness$ é substituído pelo processo real da natureza $genótipo \rightarrow fenótipo \rightarrow comportamento \rightarrow fitness$. Ou seja, o fenótipo, incorporado ao robô, é colocado para agir livremente e por determinado tempo no ambiente e, segundo suas configurações e condições iniciais, haverá um comportamento resultante.

Uma vez que o fenótipo ou o comportamento sejam devidamente definidos no algoritmo, a função de *fitness* deve ser capaz de quantificar a qualidade do indivíduo como solução para o problema abordado. Isso quer dizer que a função deve retornar um valor numérico que reflete o quão adequado é o conjunto de parâmetros do cromossomo.

Em um problema de maximização, se o cromossomo A representa uma melhor solução para o problema do que o cromossomo B, então a *fitness* de A deve ser um valor maior que a *fitness* de B, e a ordenação da avaliação da população deve ser decrescente.

Com base na ordenação das avaliações é feita a seleção dos cromossomos-pais que formarão a próxima geração de indivíduos. Dentre as técnicas mais comuns de seleção estão o método da roleta viciada e o método do torneio.

Segundo a teoria da seleção natural, pais mais adaptados ao meio se reproduzem mais, e os menos adaptados não deixam também de gerar descendentes, mas em menor número, porque partes de seus cromossomos podem ser muito adequadas para a composição da solução ideal e podem não estar presentes em nenhum outro indivíduo.

No método da roleta viciada as preferências de seleção são definidas pela proporção que as soluções representam em fatias de uma roleta virtual. Cada indivíduo é representado por sua *fitness*, e a roleta é composta pela soma das avaliações de todos os indivíduos, como mostrado na Figura 3.5.

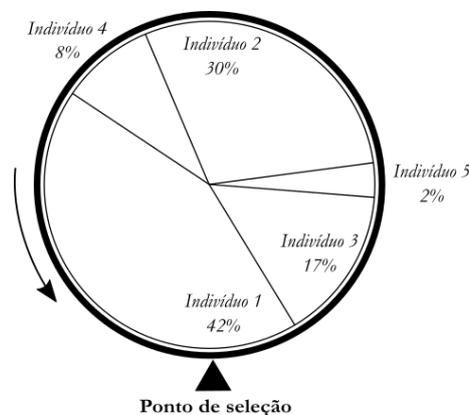


Figura 3.5: Simulação de uma roleta viciada

Com isso, ao *girar* a roleta (ao sortear-se um número ou um ângulo) o indivíduo com

maior *fitness* tem uma probabilidade maior de ser selecionado como pai. No caso da figura, o indivíduo 1 é o mais provável e o indivíduo 5 é o menos provável.

A abordagem do torneio, por sua vez, consiste na seleção de um subgrupo de indivíduos da população (de forma aleatória ou até mesmo por meio da roleta) e, destes, o indivíduo com melhor *fitness* é selecionado como pai. O subgrupo não é descartado e os indivíduos podem ser novamente sorteados. O tamanho do torneio define a quantidade de indivíduos do subgrupo.

Selecionados os pais, o algoritmo passa à aplicação dos operadores genéticos para formar novos cromossomos. Porém, como na natureza, nem sempre tais operações devem ser aplicadas, sendo necessário então um sorteio para decidir pelas modificações.

A primeira etapa é a operação de *crossover*, ou recombinação, que consiste no cruzamento de material genético dos pais, com a possibilidade de ocorrer definida pelo parâmetro *probabilidade de crossover*. A Figura 3.6 exemplifica dois tipos de operação de *crossover* para uma representação binária.

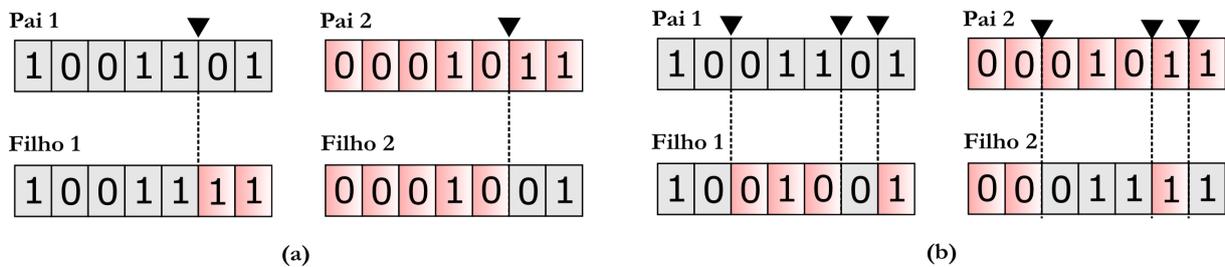


Figura 3.6: Exemplo de *crossover* binário de (a) 1 ponto e (b) 3 pontos

No exemplo da figura, dois pais geram dois filhos, cujos genótipos são o resultado da combinação de partes dos cromossomos dos pais, chamada *crossover* de n pontos, onde n é o número de partes em que os vetores são divididos para formar o novo arranjo. A figura (a) exemplifica um *crossover* de um ponto e a figura (b) de três pontos.

Os pontos de divisão são definidos de forma aleatória, e caso o sorteio defina que o recombinação não deve ser aplicada, os filhos serão a reprodução idêntica dos pais.

A segunda operação é a mutação, que é uma modificação aleatória de um ou mais genes dos descendentes, e deve ocorrer segundo o parâmetro *probabilidade de mutação*. A Figura 3.7 mostra uma operação de mutação em um cromossomo binário.

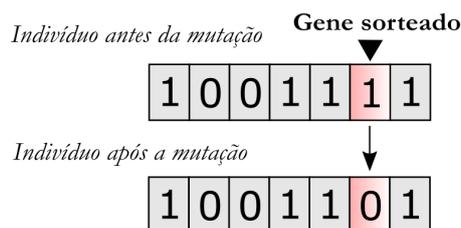


Figura 3.7: Exemplo de mutação binária de um gene

No caso da representação binária, a mutação consiste na simples substituição do *bit* por seu valor oposto considerando o conjunto $\{0,1\}$, e pode ocorrer em um ou mais genes, a depender da necessidade de conferir variabilidade genética à população, ou seja, de criar variabilidade nas soluções candidatas.

No que se refere à busca pela solução adequada, o operador de *crossover* representa em geral uma investigação de aproveitamento de soluções próximas, enquanto a operação de mutação representa uma exploração mais ampla do espaço de busca, porque uma pequena modificação pode conduzir a uma solução bastante diferente. O *crossover*, portanto, é importante na busca por ótimos locais, enquanto a mutação tem destaque para a exploração dos ótimos globais.

Tendo sido formada uma nova população geneticamente modificada, os pais são desprezados (na abordagem mais tradicional) e dá-se início a uma nova geração, que é novamente qualificada de acordo com sua adequação como solução do problema.

Em alguns casos é conveniente a aplicação da técnica elitismo, em que parte dos indivíduos melhor avaliados da população, denominada elite, passa diretamente à próxima geração, sem a atuação de nenhum dos operadores genéticos. Tal prática é útil para garantir que boas soluções não desapareçam de uma geração para outra.

Os critérios de parada do algoritmo podem ser baseados na qualidade das soluções ou na quantidade de gerações, e a eficácia da otimização está relacionada com a escolha dos parâmetros e dos métodos a serem aplicados [78].

Dentre os principais desafios no uso dos algoritmos genéticos estão: a não garantia de se obter a solução ótima (ainda que alguma solução será sempre encontrada); a definição da função *fitness*, que em geral é a responsável por guiar os processos evolutivos; a convergência prematura, quando os indivíduos convergem para uma solução sub-ótima e praticamente deixam de evoluir (a não ser pela ocorrência de mutações positivas [76]); e a necessidade de muitas avaliações da solução e várias gerações, o que demanda alto custo computacional.

Apesar disso, os AGs são um método de busca e otimização robusto, capaz de lidar com superfícies de custo mal-comportadas, com propriedades como multimodalidade, descontinuidade, variância no tempo, aleatoriedade e ruído. São portanto uma alternativa a problemas difíceis de se formalizar matematicamente, como os altamente não-lineares, estocásticos ou que ainda não são bem compreendidos [79].

Em sua implementação não há obrigatoriedade de inclusão de muitas informações estritamente ligadas ao problema específico, sendo que a complexidade fica transferida à definição da função de ajuste do algoritmo. Com isso, os AGs podem ser aplicados a uma ampla gama de problemas sem modificações significativas.

Neste trabalho, os algoritmos genéticos, nos termos da robótica evolutiva, são aplicados para evoluir um controlador simulado, baseado em um autômato finito, conforme detalhado no capítulo subsequente.

Capítulo 4

Abordagem Proposta

A presente dissertação trata do problema de perseguição-evasão em que um time de N perseguidores, também chamados robôs ou buscadores, devem coordenar seus movimentos de forma a capturar todos os evasores de um ambiente desconhecido plano, limitado e que pode ser multiplamente conectado.

Trata-se de uma abordagem cooperativa descentralizada em que os robôs têm ciência a respeito da existência e do posicionamento de outros robôs, desde que estejam dentro do alcance de comunicação.

Para solucionar a tarefa propõe-se um modelo de controlador baseado em autômato finito cujo mapeamento de estados em ações seja obtido por meio de um processo evolutivo simulado nos termos da robótica evolutiva.

A primeira seção deste capítulo trata do ambiente utilizado para evoluir o controlador, assim como dos atributos e restrições dos robôs perseguidores no ambiente. A seção 4.2 apresenta o modelo de controle para os robôs e a seção 4.3 discute as particularidades do modelo de evolução aplicado para ajustar o controlador proposto.

4.1 Ambiente e restrições de simulação

O problema é tratado em ambiente discreto simulado, desenvolvido no *software* Matlab[®] por Gonçalves *et al.* [5] e adaptado para aplicação nesta pesquisa.

Os mapas em que ocorrem os experimentos são representados na simulação por uma grade de ocupação regular de resolução r . No processo de discretização, exemplificado na Figura 4.1, se houver na célula da grade qualquer pixel que represente um obstáculo, então toda a célula será considerada ocupada.

Durante a simulação, o mapa é apresentado segundo a percepção dos robôs a respeito do ambiente, que inicialmente é desconhecido, como na Figura 4.1(c) e, à medida que os robôs vão se deslocando, vai gradualmente sendo descoberto (e as células vão sendo classificadas) conforme as movimentações. As células que são completamente cercadas

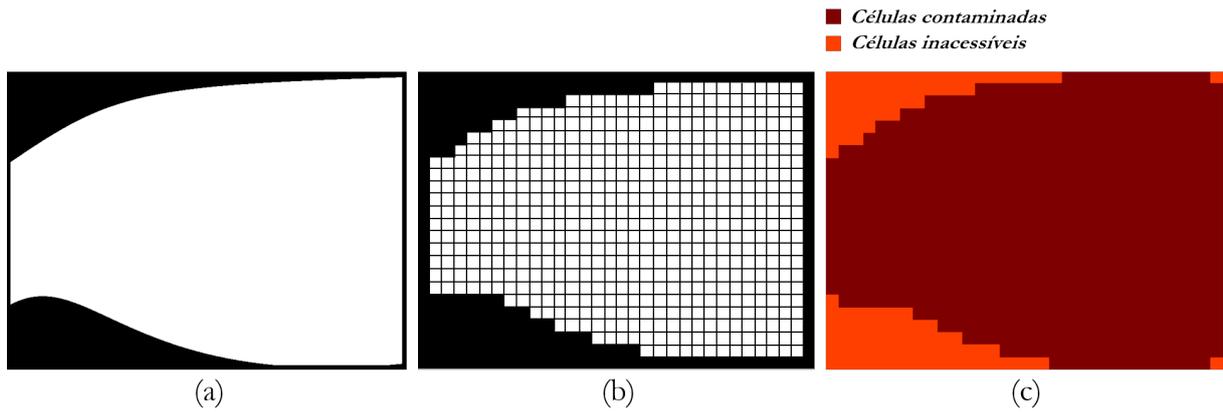


Figura 4.1: Exemplo de um mapa (a) real; (b) discretizado e (c) como visto na simulação, inicialmente desconhecido

por obstáculos são chamadas *células inacessíveis*, porque nunca serão percebidas pelo sensoriamento dos robôs, e estarão sempre destacadas na cor laranja, como na Figura 4.1.

Os evasores são considerados oniscientes a respeito do posicionamento e ações dos perseguidores, e deslocam-se continuamente pelo ambiente com velocidade ilimitada. Por isso, não precisam ser explicitamente simulados, já que eles ocuparão completamente todas as células ditas contaminadas, ou seja, ainda não inspecionadas pelos buscadores, como é o caso de toda área marrom da Figura 4.1(c). A região limite entre uma área descontaminada e uma contaminada será daqui para frente tratada como *fronteira*.

Os perseguidores, por outro lado, ocupam o centro de uma célula e são robôs com capacidade limitada de movimentação, comunicação e sensoriamento.

A restrição de movimentação define que, a cada iteração, um robô pode mover-se apenas para uma das quatro células adjacentes da vizinhança de von Neumann ou permanecer na mesma célula, como mostrado na Figura 4.2. A proibição de movimentação na diagonal foi inserida com o objetivo de restringir o espaço de busca. Não há restrição relacionada à capacidade de memória, e cada robô salva todas as células que foram visitadas, assim como a informação local sobre áreas já descontaminadas.

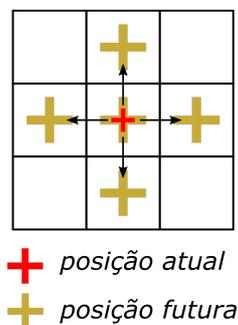


Figura 4.2: Restrição de movimentação: permanecer na célula ou deslocar-se na vizinhança de von Neumann

Os sensores são considerados perfeitos, porém de alcance restrito. O sensoriamento de um robô e sua percepção do ambiente estão ilustrados na 4.3, e foram inspirados no modelo proposto em [5], porém tratado de forma menos restrita. A classificação da situação de ocupação das células é tratada neste trabalho somente no contexto da simulação, considerando-se um processo de discretização do ambiente previamente realizado.

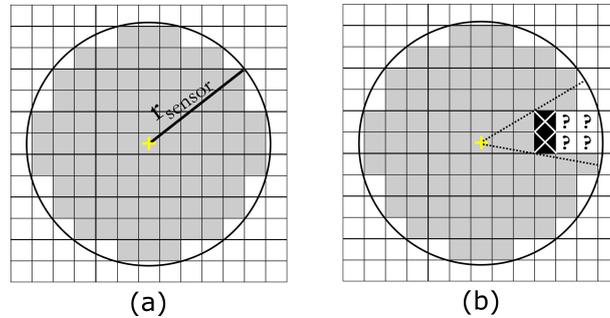


Figura 4.3: (a) Máscara de sensoriamento discreta e (b) exemplo de visão do robô

A região em cinza da figura representa o campo de visão de um robô, também chamado de máscara de sensoriamento, que é capaz de perceber somente as células completamente inscritas em um círculo definido por um raio r_{sensor} , a partir do ponto central da célula em que ele se encontra.

Uma vez que haja algum obstáculo na região de sensoriamento, como nas células pretas marcadas com X, o robô tem a percepção bloqueada e não consegue identificar as células com pontos de interrogação, que permanecem desconhecidas. As células adjacentes foram classificadas porque partes delas estão no ângulo de visão do robô. Como um processo prévio de discretização está sendo considerado, basta que parte da célula esteja visível para que a mesma seja classificada como livre ou ocupada.

A região de sensoriamento auxilia também na definição da região de comunicação, porque dois robôs comunicam-se sempre que a distância euclidiana entre eles for menor ou igual a duas vezes o raio de sensoriamento r_{sensor} . E também, para assegurar a descontaminação, em regiões de fronteira os buscadores não devem exceder a mesma distância, porque isto resultaria em uma imediata recontaminação das áreas já visitadas que estavam sendo *guardadas* pelos robôs. Alguns exemplos de configurações a tal respeito são apresentados na Figura 4.4.

Para o caso da Figura 4.4(a), dado que um dos robôs permaneça na mesma posição, qualquer movimentação do outro, que não seja no sentido do robô vizinho, resultaria em perda de comunicação e permitiria a recontaminação, caso eles estivessem protegendo uma fronteira. A configuração da figura (b) é um pouco mais flexível, e ambos os robôs ainda poderiam movimentar-se em sentidos opostos na vertical, sem que houvesse perda de comunicação ou recontaminação.

Uma vez que os robôs possuem comunicação restrita e nenhum conhecimento *a priori* do ambiente, eles serão sempre inicializados em uma formação de total comunicação, de

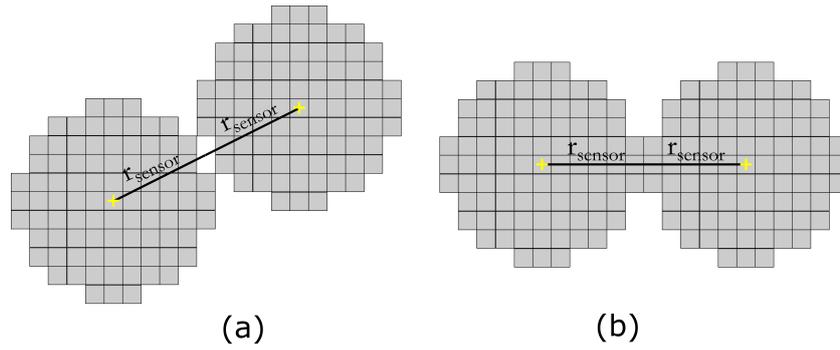


Figura 4.4: Exemplo de configurações-limite para comunicação e descontaminação

forma a terem referência do sistema de coordenadas do mapa, cuja origem é definida pela posição do primeiro robô: a primeira célula livre da parte esquerda inferior do mapa (sudoeste), como mostrado na Figura 4.5, onde são também indicadas e nomeadas as direções de movimento utilizadas neste trabalho.

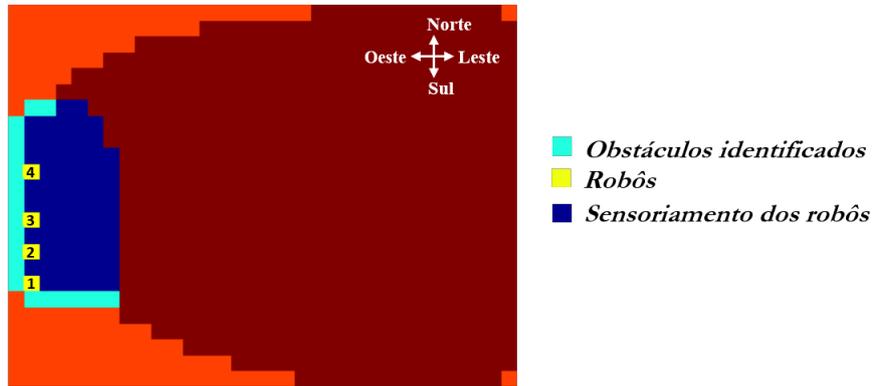


Figura 4.5: Exemplo de formação inicial com total comunicação entre os buscadores

No caso de uma fronteira em que um robô faz vizinhança com um obstáculo, a descontaminação é garantida quando a distância entre eles for menor ou igual do que o raio r_{sensor} , situação da Figura 4.6, que também ilustra uma formação de descontaminação parcial dos robôs no ambiente simulado e uma situação de recontaminação devido à movimentação.

Na Figura 4.6(a), os robôs 1 e 4 representam um caso de formação limite de descontaminação entre robô e obstáculo. É importante ressaltar que todas as células marcadas como *obstáculos identificados* assim o são porque já estiveram sob a máscara de sensoriamento de qualquer um dos robôs, e que a região *descontaminada* surgiu de uma movimentação de varredura parcial do ambiente a partir da formação da Figura 4.5. Já a recontaminação mostrada na Figura 4.6(b) é o resultado da movimentação dos robôs uma célula ao leste, quando o sensoriamento não é mais capaz de assegurar a ausência de evasores.

O objetivo deste trabalho é desenvolver um sistema autônomo de coordenação entre

buscadores para uma completa descontaminação de ambientes, e a próxima seção apresenta o modelo de sistema de controle proposto para alcançar a finalidade da pesquisa.

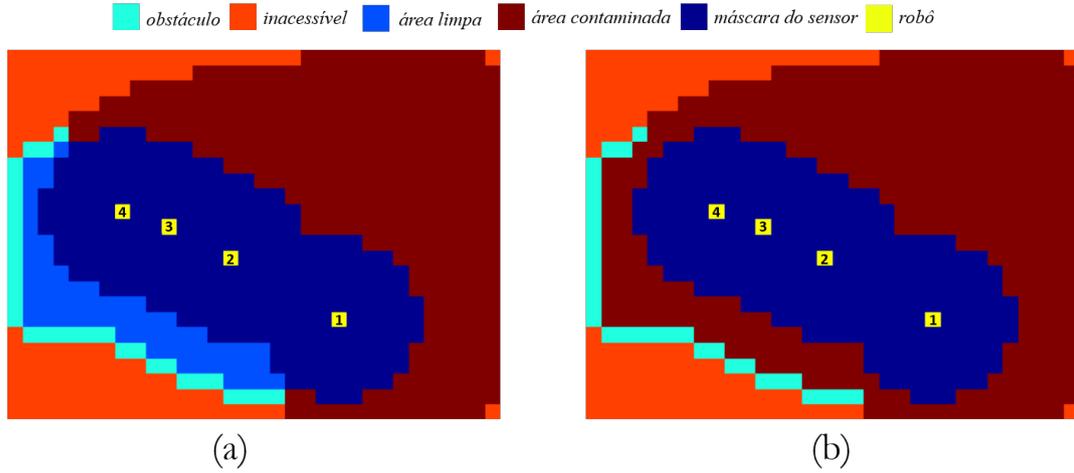


Figura 4.6: Exemplo de recontaminação no ambiente simulado

4.2 Modelo do controlador

O controlador proposto neste trabalho é um autômato discreto definido por um conjunto finito de estados possíveis, mapeados para um conjunto finito de ações de movimentação, sendo cada robô regido de forma independente pelo mesmo modelo de autômato, caracterizando a abordagem como descentralizada.

Formalmente, o autômato pode ser definido [80] como a quintupla

$$G = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F}) \quad (4.1)$$

onde:

- \mathcal{Q} é um conjunto finito de estados.
- Σ é um conjunto finito de símbolos de entrada.
- δ é uma *função de transição*, que tem como argumento um estado e um símbolo de entrada e retorna um estado.
- q_0 é um estado inicial do sistema, um dos estado em \mathcal{Q} .
- \mathcal{F} é um conjunto de estados finais, em que \mathcal{F} é um subconjunto de \mathcal{Q} .

O autômato particular deste trabalho é definido por

$$G = (\mathcal{S}^N, \mathcal{A}, \delta, {}^0s, \mathcal{F}) \quad (4.2)$$

onde:

- \mathcal{S}^N é o conjunto finito de possíveis estados $s^n \in \mathcal{S}$, com $|\mathcal{S}| = 10$ e $\mathcal{S} = \mathcal{B}^{10}$, com $\mathcal{B} = \{0, 1\}$. Consequentemente, $s_i^n \in \mathcal{B}$, ou seja, cada elemento do *vetor de estado* é binário e caracterizado segundo a percepção local do n -ésimo robô a respeito do ambiente e da fronteira de contaminação, sendo $n \in \{1, 2, \dots, N\}$.
- \mathcal{A}^N é o conjunto finito de ações $\alpha \in \mathcal{A}$ associadas a G , com $|\mathcal{A}| = 5$.
- $\delta : \mathcal{S}^N \times \mathcal{A}^N \rightarrow \mathcal{S}^N$ é a *função de transição* em que não é possível definir o estado de destino a partir da ocorrência de uma ação dado um estado atual, ou seja, para qualquer robô em um estado $s \in \mathcal{S}$, a ocorrência da ação $\alpha \in \mathcal{A}$, associada ao estado, conduzirá o robô para qualquer outro estado $s \in \mathcal{S}$.
- 0s é o estado inicial do sistema, em que qualquer estado pode ser o estado inicial.
- \mathcal{F} é o conjunto de estados finais, em que qualquer estado pode ser o estado final.

Apesar de o sistema como um todo ser não-determinístico, uma vez que, dada uma ação, não é possível conhecer o estado futuro a partir do estado atual, o autômato proposto em si é determinístico, porque, definido o estado atual, haverá *probabilidade* = 1 de ocorrência de determinada ação α .

Cada dimensão do vetor de estado é um dígito binário, ou seja, $s_i^n \in \{0, 1\}$, cujo valor é definido pela concordância à descrição afirmativa das dimensões, conforme a Tabela 4.1. O valor atribuído a cada dimensão será então 0 se a resposta à afirmação correspondente for falsa e 1 se for verdadeira.

Tabela 4.1: Descrição do Vetor de Estado

Dimensão	Descrição
1	Guardando uma fronteira
2	A fronteira está segura
3	Leste está limpo
4	Norte está limpo
5	Oeste está limpo
6	Sul está limpo
7	Leste livre de colisão
8	Norte livre de colisão
9	Oeste livre de colisão
10	Sul livre de colisão

A primeira dimensão diz respeito à situação de fronteira, conforme percebida pelo robô, que estará *guardando uma fronteira* em dois casos: (*i*) quando a interseção entre sua máscara de sensoriamento com obstáculos ou com a máscara de outro robô tem uma vizinhança de Moore (8 células adjacentes) com uma célula contaminada, mas sem que

tenha permitido recontaminação, ou (ii) quando um robô está em uma área totalmente limpa, mas ainda comunica-se com os robôs que estão *guardando a fronteira*. Para ilustrar tais percepções, seja o exemplo da Figura 4.7.

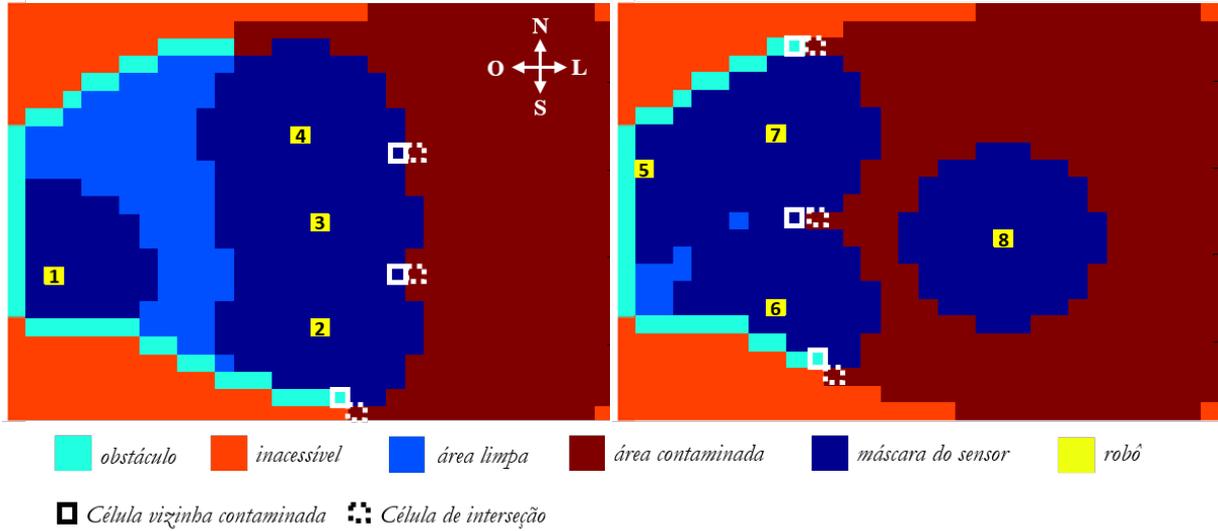


Figura 4.7: Exemplos de situações de estado

Na figura, os robôs 2, 3, 6 e 7 encaixam-se no primeiro caso, ou seja, $s_1^n = 1$ para $n \in \{2, 3, 6, 7\}$, e os robôs estão *guardando* a fronteira por meio das células destacadas, enquanto o robô 5 cai no segundo caso, e também $s_1^5 = 1$ porque, apesar de não estar na fronteira, tem comunicação com os robôs que estão, e por isso conhece a situação de descontaminação local.

No caso do robô 4, a extremidade de fronteira superior de sua máscara de sensoria-mento já não tem célula comum ao obstáculo, e por isso uma única movimentação para o sul ou para o leste pode recontaminar todo o ambiente. Por este motivo, $s_1^4 = 0$, isto é, o mesmo não está guardando a fronteira, apesar de saber, pelo histórico de movimentação, que a recontaminação não ocorreu, e apesar de o outro lado da fronteira estar sendo guardado. Esta escolha conduz a um comportamento conservativo, mas foi feita de forma a generalizar o movimento e reduzir o tamanho do vetor de estados.

Os robôs 1 e 8 estão deslocados, o primeiro em uma área descontaminada e o outro em uma área contaminada, e por isso não têm conhecimento a respeito da guarda da fronteira. Para estes casos, $s_1^n = 0$ para $n \in \{1, 8\}$.

A segunda dimensão do vetor define a situação de segurança da fronteira, ou seja, se a movimentação pode resultar em recontaminação, ou se o robô pode movimentar-se de maneira segura. Neste sentido, ainda na Figura 4.7, apenas os robôs 1, 2, 3 e 5 têm a fronteira segura, então $s_2^n = 1$ para $n \in \{1, 2, 3, 5\}$. O robô 1, apesar de não ter comunicação com os robôs da fronteira, é considerado ciente de que ficou para trás e, uma vez que o processo é interrompido quando ocorre recontaminação, como será explicado na

próxima seção, sabe que está em uma área descontaminada e, portanto, está livre para movimentar-se.

Por outro lado, $s_2^n = 0$ para $n \in \{4, 6, 7, 8\}$. O robô 8 desvinculou-se da fronteira e sabe que está em uma área contaminada, então sua movimentação não interfere na segurança da área limpa. O robô 4 já não está guardando a fronteira porque chegou em uma posição limite de segurança, então certamente pode ser responsável pela recontaminação, e os robôs 6 e 7 também correm este risco, porque, uma vez que a movimentação ocorre de forma paralela, o deslocamento do primeiro para o sul e do segundo para o norte faria com que a fronteira deixasse de existir.

As dimensões de 3 a 6 do vetor de estado dão informação da situação de descontaminação nos 4 sentidos da região de percepção local de cada buscador. As regiões da máscara de sensoriamento são divididas conforme a Figura 4.8.

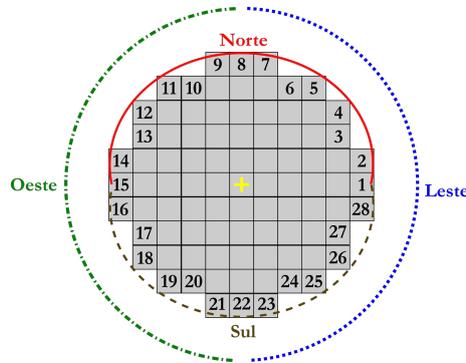


Figura 4.8: Sentidos para análise de descontaminação

As células que estão na borda da máscara sempre contribuem na informação de contaminação para mais de um sentido. Se, por exemplo, a célula de borda 3 indicar uma área contaminada, então a informação de contaminação é atribuída tanto para o sentido leste quanto para o sentido norte.

Voltando aos casos da Figura 4.7, somente os buscadores 1 e 5 indicam sensoriamento limpo em todos os sentidos, então $s_i^1 = s_i^5 = 1$ para $i \in \{3, 4, 5, 6\}$. O robô 8 está em uma região totalmente contaminada, enquanto o robô 4 contém indicação de contaminação em pelo menos uma célula do sensor em cada sentido, logo, $s_i^4 = s_i^8 = 0$ para $i \in \{3, 4, 5, 6\}$. Por fim, os robôs 2, 3, 6 e 7 podem definir como limpo somente o sentido oeste, portanto para $n \in \{2, 3, 6, 7\}$, $s_5^n = 1$ e $s_i^n = 0$ com $i \in \{3, 4, 6\}$.

As quatro últimas dimensões dizem respeito à liberdade de movimentação, no sentido de evitar colisões. O risco de colisão é definido de forma diferente entre robôs e entre robô e obstáculo. Um buscador tem risco de colidir no caso de existir um obstáculo em uma das células da vizinhança de von Neumann ou se estiver a duas células de distância de outro buscador em qualquer sentido de vizinhança, por causa da movimentação paralela. Na Figura 4.9 são apresentados exemplos de ambos os casos.

Na figura, o robô 1 possui risco de colisão em todas as direções, ou seja, $s_i^1 = 0$ para

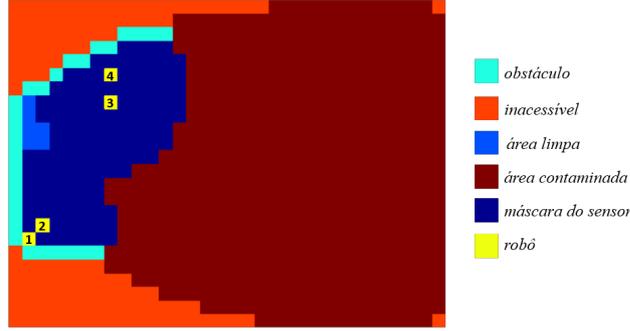


Figura 4.9: Exemplos de possibilidade de colisão

$i \in \{7, 8, 9, 10\}$. Isto porque o deslocamento ao oeste ou ao sul provocam uma colisão com a parede, e porque a movimentação para norte ou leste podem causar uma colisão em dois casos: se o buscador 1 move-se para leste e o buscador 2 para o sul, ou se o primeiro move-se para norte e o segundo para oeste. Isto faz com que a definição das dimensões 9 e 10 do robô 2 também acusem que não estão livres de colisão, ou seja, $s_9^2 = s_{10}^2 = 0$.

Essa dependência da movimentação de um outro buscador também envolvem os robôs 3 e 4, porque se o primeiro desloca-se para norte e o segundo para sul, a colisão ocorrerá. Já para todos os outros sentidos o deslocamento é livre e, portanto, $s_i^3 = 1$ para $i \in \{7, 9, 10\}$ e $s_i^4 = 1$ para $i \in \{7, 8, 9\}$.

Uma vez que cada uma das dimensões pode assumir dois valores, tem-se um total de $2^{10} = 1024$ estados possíveis para o autômato. A transição entre estados depende da ação de movimentação $\alpha \in \mathcal{A}$ e o estado futuro depende do ambiente e também da situação dos outros robôs, de forma que, como já foi dito, não é possível defini-lo a partir do estado atual.

A cada iteração, o estado dos robôs é individualmente identificado dentre os 1024 estados possíveis, definindo o cenário em que o robô está. Cada estado, marcado com um identificador (id) precisa então corresponder a uma única ação de movimentação α_{id} , como mostrado na Figura 4.10.

<i>Id do estado</i>	<i>Vetor de estado</i>	<i>Ação</i>
1	0 0 0 0 0 0 0 0 0 0	α_1
2	0 0 0 0 0 0 0 0 0 1	α_2
3	0 0 0 0 0 0 0 0 1 0	α_3
:	:	:
1022	1 1 1 1 1 1 1 1 0 1	α_{1022}
1023	1 1 1 1 1 1 1 1 1 0	α_{1023}
1024	1 1 1 1 1 1 1 1 1 1	α_{1024}

Figura 4.10: Mapeamento de estados em ações

As ações de movimentação α podem assumir valores inteiros de 0 a 4, e estão descritas na Tabela 4.2.

Tabela 4.2: Possíveis ações de movimentação para o robô

Ação	Descrição
0	Robô não se move
1	Move-se uma célula para o leste
2	Move-se uma célula para o norte
3	Move-se uma célula para o oeste
4	Move-se uma célula para o sul

A ação definida para cada estado deve ser tal que leve o robô a assumir um comportamento coletivo de cooperação para cumprir a tarefa de limpeza do ambiente de toda a contaminação inicial. Para realizar uma busca exaustiva com 4 robôs, considerando as 5 ações de movimentação possíveis e os 1024 genes, $5^{4 \cdot 1024} = 9,57 \times 10^{2862}$ possibilidades de combinação deveriam ser analisadas, o que é inviável.

Neste trabalho, foi escolhido o método evolutivo de busca para encontrar um mapeamento que se mostre adequado à solução do problema, como detalhado na próxima seção.

4.3 Modelo de evolução

Segundo Hettiarachchi *et al.* [48], operações evolutivas em máquinas de estado finito são menos intuitivas e bastante atenção precisa ser despendida no projeto da representação genotípica. No caso deste trabalho, entretanto, o processo de evolução não atua na estrutura do autômato, mas somente no mapeamento de estados em ações, ou seja, para cada estado s , espera-se encontrar uma ação correspondente $\alpha \in \mathcal{A}$, de forma que o comportamento resultante do sistema garanta a completa descontaminação do ambiente.

Por isso, o cromossomo escolhido para o algoritmo evolutivo é um conjunto de 1024 ações de movimentação, correspondentes aos possíveis estados do sistema. Cada indivíduo da população, portanto, é um vetor de ações inicializadas de forma pseudoaleatória.

A busca deste trabalho concentra-se na obtenção do comportamento coletivo, e por isso algum algoritmo de evitar colisão é assumido para os robôs, de forma que, no processo de inicialização, ações que representam colisão são não factíveis, ou seja, são desprezadas e substituídas nos genes dos cromossomos, como demonstrado na Figura 4.11.

Além disso, ações de movimentação com valor zero, que correspondem à permanência do robô na mesma célula, são permitidas somente em casos de colisão em todas as direções. Isto é feito no intuito de acelerar a evolução do comportamento de descontaminação, não permitindo que os robôs permaneçam parados sem necessidade.

Uma vez gerada a população de cromossomos, cada um deles é incorporado simultaneamente como sistema de controle de todos os robôs presentes na simulação, e o com-

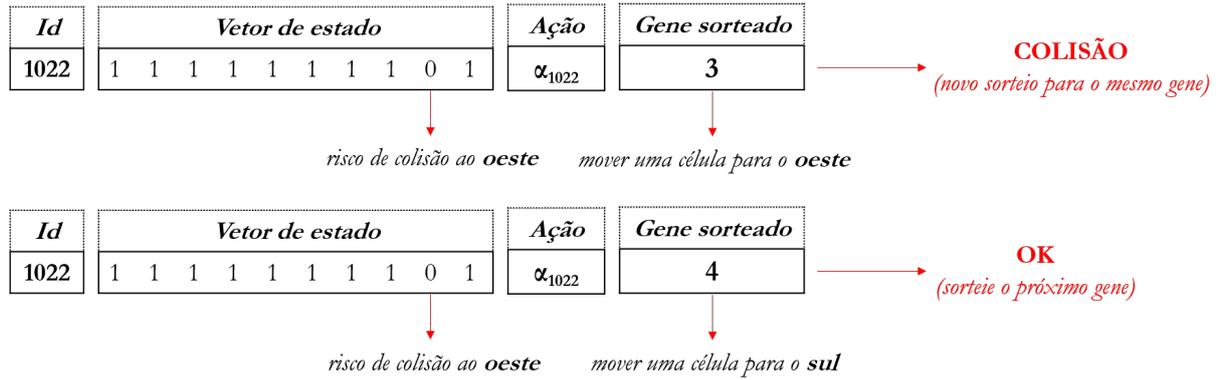


Figura 4.11: Processo de restrição de colisão

portamento emergente resultante é então avaliado segundo a função

$$fitness = \begin{cases} 0, & \text{Recontaminação} \\ \frac{1}{Contaminação}, & \text{Contaminação} > 0 \\ 1, & \text{Contaminação} = 0 \end{cases} \quad (4.3)$$

em que

- *Recontaminação* indica que os robôs abriram a formação, fazendo com que a contaminação se espalhe por todo o ambiente, caso em que a simulação do indivíduo é interrompida. Assim, ainda que não haja proibição explícita de recontaminação, a penalidade é imposta no intuito de que este comportamento desapareça no processo evolutivo; e
- *Contaminação* corresponde ao número de células contaminadas ao final da simulação, de forma que a avaliação é tanto maior quanto menor for o número de células contaminadas, e nos casos de completa descontaminação do ambiente, o indivíduo recebe *fitness* máxima.

Após a avaliação de toda a população, os indivíduos são ordenados e selecionados por meio de torneio para formarem a próxima geração, através da aplicação dos operadores genéticos. O *crossover* de n pontos é aplicado conforme exemplificado na Figura 4.12 (a), para $n = 2$, e a mutação conforme a Figura 4.12 (b), mostrada para apenas 2 genes.

É interessante notar pelo exemplo da figura que o *crossover* é realizado de forma que haja correspondência entre as posições, e portanto todas as restrições iniciais de evitar colisão são mantidas. Já no caso da mutação, o gene sorteado passa novamente pelo critério da Figura 4.11.

A simulação de cada indivíduo é interrompida nos casos de recontaminação, de total descontaminação do ambiente ou se for atingido o número limite de iterações definido. Já o

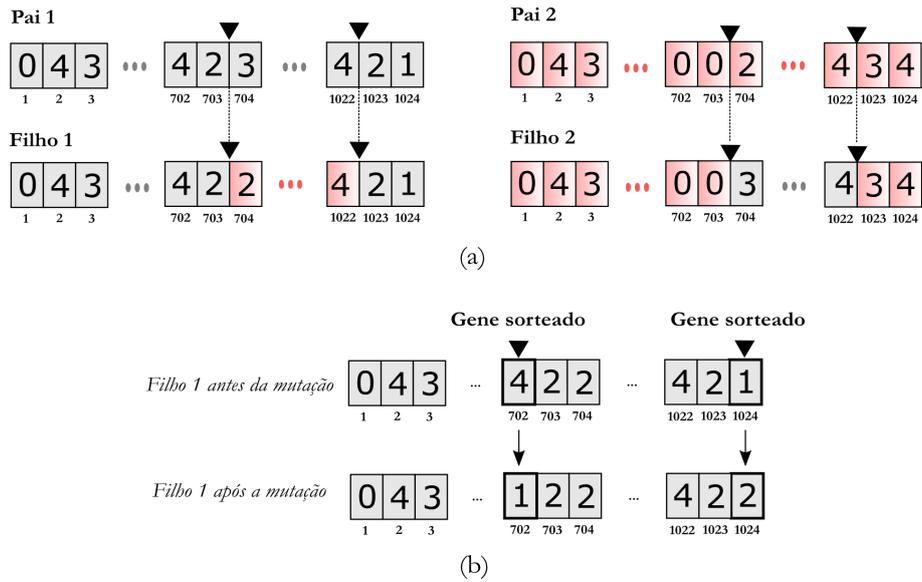


Figura 4.12: Exemplificação de (a) *crossover* e (b) mutação na abordagem proposta

critério de parada do algoritmo de evolução é o número de gerações estabelecidas, quando espera-se que hajam surgido indivíduos que satisfaçam a tarefa de descontaminação do ambiente.

No próximo capítulo são tratados dos detalhes específicos a respeito da utilização desta abordagem e são discutidos os resultados obtidos ao longo desta pesquisa.

Capítulo 5

Resultados

Neste capítulo são fornecidos, na seção 5.1, os detalhes de aplicação da abordagem evolutiva proposta neste trabalho, e na seção 5.2 são mostrados e discutidos os resultados das soluções evoluídas, em complemento a resultados preliminares apresentados em [21].

Além disso, na seção 5.3 são apresentados os resultados da abordagem complementar, que investiga a aplicação de sorteios aleatórios em conjunto com a solução evoluída, importante para o desdobramento de trabalhos futuros, e na seção 5.4 é realizada uma análise comparativa de método e resultados com a proposta apresentada por Gonçalves [3] para solução do mesmo problema.

5.1 Parâmetros do Processo Evolutivo

Para a etapa de condução da evolução, foi definida neste trabalho a utilização de quatro robôs com raio de sensoriamento e comunicação $r_{sensor} = 150$ pixels. A quantidade de buscadores com tal característica é suficiente para descontaminação de todos os mapas utilizados, que possuem 600×800 pixels. A resolução escolhida para a grade de ocupação foi de 25 pixels, o que resulta em mapas discretizados de 24×32 células. Tais escolhas foram inspiradas no trabalho de Gonçalves *et al.* [5], com o intuito de posteriormente comparar as soluções obtidas neste trabalho com as apresentadas naquele.

Os mapas escolhidos para o processo evolutivo, mostrados na Figura 5.1, são simplesmente conectados e relativamente adaptados dos mapas utilizados em [5]. Foram escolhidos de forma a representar partes de ambientes reais, como passagens estreitas, salas e corredores.

No início de cada interação da solução candidata com o ambiente, os robôs são posicionados no extremo oeste, de forma a apresentar uma formação de total comunicação e de descontaminação no sentido vertical, como exemplificado na Figura 5.2. As posições dos buscadores são sorteadas dentro de faixas que atendam às restrições apresentadas, com exceção do primeiro robô, conforme explicado no capítulo precedente.

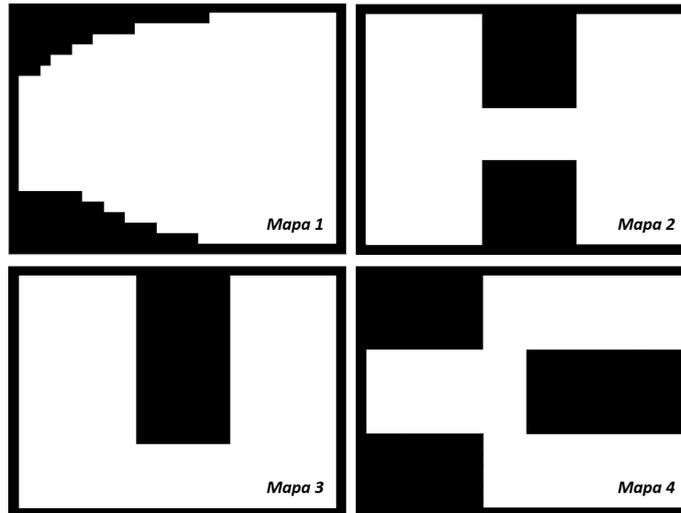


Figura 5.1: Mapas utilizados no processo evolutivo

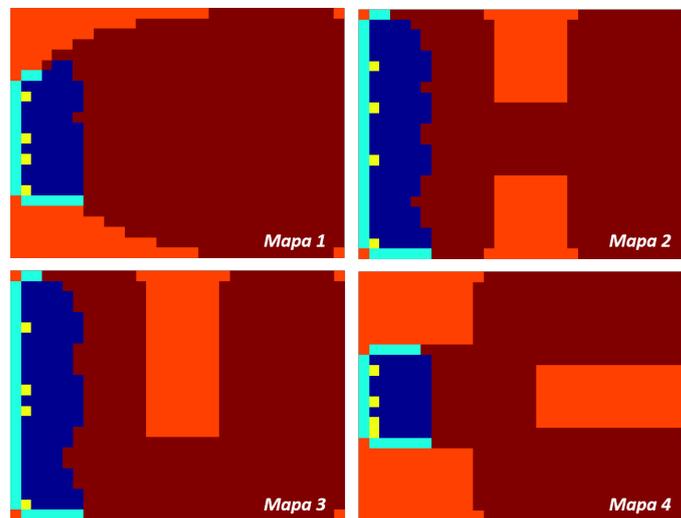


Figura 5.2: Exemplo de inicialização da formação

Na Figura 5.3 é mostrado um exemplo de configuração final de descontaminação que se deseja obter para os quatro mapas utilizados.

Em busca de uma solução mais generalizada, além das posições iniciais aleatórias foi também estabelecido que a avaliação da solução seria definida pelo comportamento resultante da interação dos buscadores nos quatro mapas apresentados, dentro do mesmo processo evolutivo. Isto quer dizer que a mesma solução candidata foi testada em cada um dos mapas da Figura 5.1, e sua avaliação deu-se pelo somatório das avaliações parciais realizadas em cada mapa segundo a equação 4.3, sendo 4 o valor máximo para a *fitness* final, no caso de descontaminação de todos os mapas.

A cada iteração de toda simulação os robôs movimentam-se guiados pelo mesmo autômato, sendo o mapeamento de estados em ações realizado de forma independente para cada robô, uma vez que os mesmos podem estar em estados distintos. A simulação em

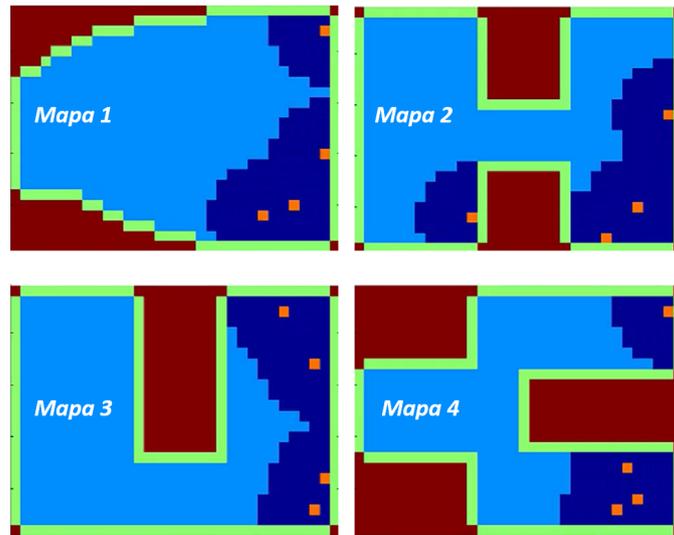


Figura 5.3: Exemplo de finalização da formação

cada mapa tem uma duração máxima de 70 iterações, podendo ser interrompida antes devido à limpeza completa do ambiente ou por recontaminação. Na Figura 5.4 está ilustrado o fluxograma completo do processo evolutivo realizado.

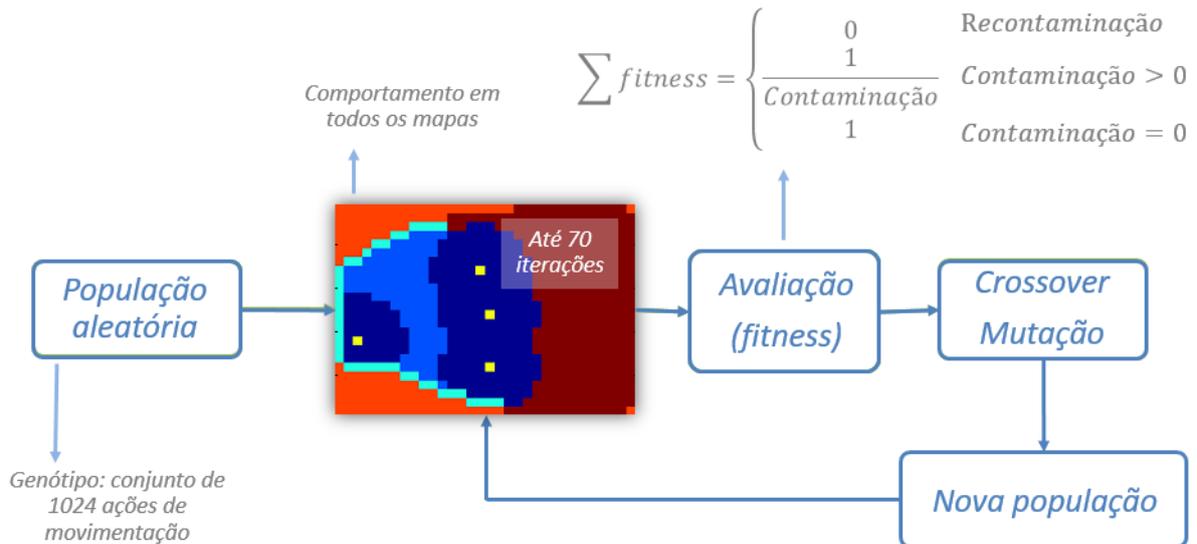


Figura 5.4: Fluxograma do processo evolutivo completo

Foi escolhida uma população de 200 indivíduos, na tentativa de obter uma relação de compromisso entre o tempo de condução do processo evolutivo, para que não fosse tão elevado, e a necessidade de variabilidade genética para o ajuste de um cromossomo de 1.024 posições.

Cada experimento evolutivo foi conduzido por 200 gerações e o elitismo foi aplicado em 10% da população de cada geração, sem que com isso haja garantia de que as melhores

soluções efetivamente sobreviveram ao longo das gerações, uma vez que elas foram testadas em posições diferentes a cada simulação.

Os pais foram selecionados por torneios entre dois indivíduos, e foram escolhidos 20 pontos para a operação de *crossover*, tendo em vista o tamanho do cromossomo utilizado, aplicado segundo uma probabilidade de 50%.

E sendo a função de custo percebida como multimodal, a probabilidade de mutação foi definida de forma experimental como 20% e, uma vez selecionada a operação, a mesma ocorre em 20% dos genes, escolhidos aleatoriamente.

A Tabela 5.1 resume os parâmetros definidos para aplicação do algoritmo genético e a estrutura resumida da metodologia de evolução aplicada nesta abordagem é apresentada em forma de pseudo-código no Apêndice A.

Tabela 5.1: Parâmetros do Algoritmo Genético

Parâmetro	Valor
Tamanho da população	200
Número de gerações	200
Elitismo	10%
Tamanho do cromossomo	1024
Probabilidade de <i>crossover</i>	50%
Pontos de <i>crossover</i>	20
Probabilidade de mutação	20%

Dez processos evolutivos foram realizados a partir da mesma estrutura indicada no Apêndice A. Em simulações realizadas em um computador com processador Intel Core i5 1.70 GHz, 8GB de RAM, cada geração levou cerca de 40 minutos para ser finalizada. Na próxima seção são apresentados os testes e correspondentes resultados que visam avaliar as soluções obtidas nos experimentos.

5.2 Análise das Soluções Evoluídas

Inicialmente a qualidade das soluções foi verificada através de testes nos mesmos mapas utilizados no processo evolutivo, porém com nova inicialização de posições, que não correspondem exatamente às utilizadas durante a evolução.

Foram aleatoriamente definidos quatro conjuntos, com 12 configurações de formação inicial cada, para aplicação nos testes com os quatro mapas da Figura 5.1, sendo um conjunto para cada mapa. É importante ressaltar que a qualidade das soluções foi julgada pelo número de sucessos na tarefa completa de descontaminação, e não mais pelos valores de *fitness* atribuídos às soluções candidatas durante o processo de evolução. Isto significa que se o ambiente não é completamente limpo ou se a formação permite a recontaminação a descontaminação não é contabilizada.

Os testes foram realizados com os dez melhores indivíduos de cada experimento evolutivo, em intervalos de 20 gerações, a partir da centésima, até a geração 200. Com isso, foi possível conhecer a variação no desempenho das soluções em cada mapa ao longo do processo.

A Tabela 5.2 contém a média de sucesso dos 100 indivíduos (dez melhores de cada experimento), testados para as doze configurações em cada mapa ao longo das gerações.

Tabela 5.2: Média de sucesso dos experimentos por geração

Geração	Mapa 1	Mapa 2	Mapa 3	Mapa 4
100	88%	71%	75%	64%
120	90%	76%	84%	70%
140	93%	75%	86%	81%
160	93%	74%	90%	85%
180	94%	76%	92%	87%
200	94%	77%	91%	91%

Acompanhando as taxas de descontaminação mostradas na Tabela 5.2, é possível perceber casos de diminuição de desempenho em alguns mapas com o aumento das gerações, como ocorreu para o mapa 2 entre as gerações 120 e 180 e na última geração do mapa 3. Estas ocorrências se devem ao ajuste da busca para obter um comportamento mais generalizado, porque a avaliação do processo evolutivo pontua melhor soluções que têm a maior média de sucesso para todos os quatro mapas. Logo, nota-se que a diminuição da taxa para um mapa é sempre acompanhada de aumento para outro.

Na Figura 5.5 tais dados são apresentados de forma gráfica, assim como é mostrada a evolução na taxa de descontaminação considerando a média de desempenho em todos os mapas, onde fica evidente o crescimento global discutido.

Já por meio da Tabela 5.3, é possível conhecer o desempenho das soluções da última geração discriminados por experimento, sendo mostrados tanto os valores médios quanto por mapa.

Tabela 5.3: Média de sucesso da última geração

Mapa	Experimento Evolutivo									
	1	2	3	4	5	6	7	8	9	10
1	99%	91%	98%	79%	86%	100%	92%	100%	100%	99%
2	86%	63%	72%	49%	77%	84%	76%	99%	90%	72%
3	83%	88%	91%	91%	93%	89%	83%	100%	94%	98%
4	95%	54%	93%	83%	100%	99%	99%	100%	98%	93%
Média	91%	74%	89%	76%	89%	93%	88%	100%	95%	90%

O fato de o desempenho geral médio ser sempre crescente, conforme apresentado na Figura 5.5, mostra-se como indicativo de que a continuação da evolução poderia conduzir a resultados mais robustos, e dois teste foram feitos a respeito. O primeiro foi efetuado

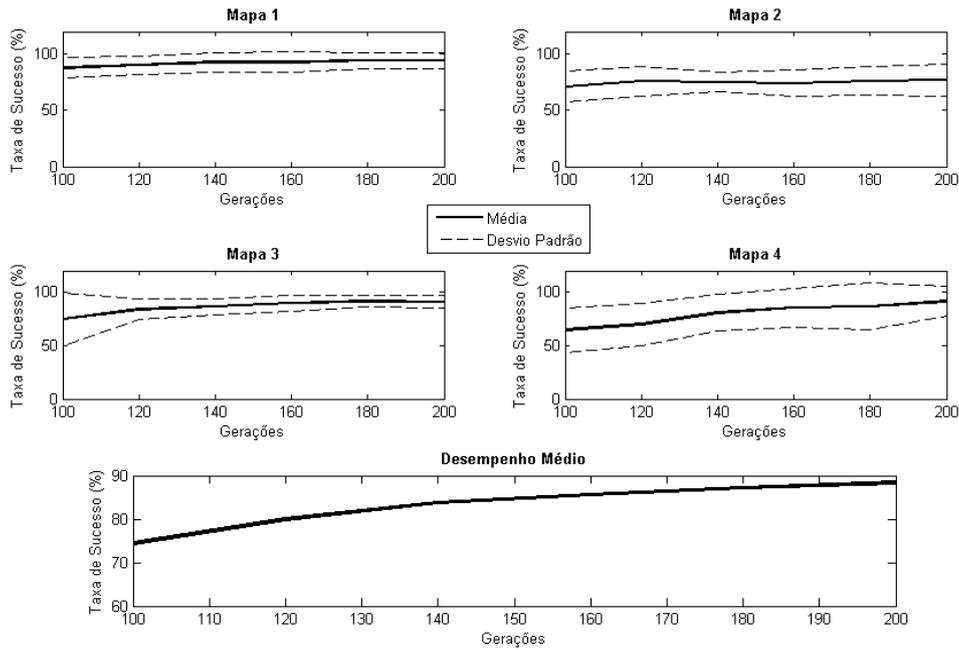


Figura 5.5: Taxa de sucesso na tarefa de descontaminação

para o experimento 8, que tinha mostrado o melhor resultado até então, e já na geração 300 os dez melhores indivíduos foram capazes de descontaminar 100% dos mapas para todas as 12 posições de teste.

Além dos dez experimentos apresentados, um a mais havia sido realizado e posteriormente descartado por ter se configurado como *outlier*, não apresentando um único sucesso na descontaminação do mapa 2 até a geração 200. Este experimento foi então utilizado como segunda tentativa, em que o processo evolutivo foi continuado até a geração 1000, quando o resultado foi considerado estabilizado, conforme apresentado na Figura 5.6.

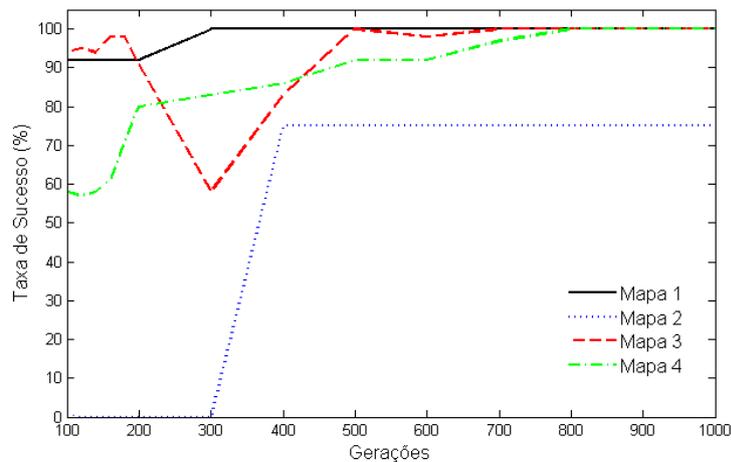


Figura 5.6: Continuação do processo evolutivo para um experimento *outlier*

A continuação do processo evolutivo neste caso mostrou a possibilidade de obter-se

uma melhor solução até um dado limite, possivelmente pela influência da inicialização dos cromossomos na primeira geração.

Posteriormente, para continuar a verificar a qualidade e robustez das soluções foram realizados testes em mapas diferentes dos utilizados na evolução. Foram escolhidos os dois mapas multiplamente conectados mostrados na Figura 5.7. Um mapa pode ser considerado como multiplamente conectado (ou não simplesmente conectado) se nele existir qualquer curva fechada que não pode ser contraída até tornar-se um ponto dentro da região, como é o caso de caminhos fechados ao redor dos obstáculos presentes nesses mapas.

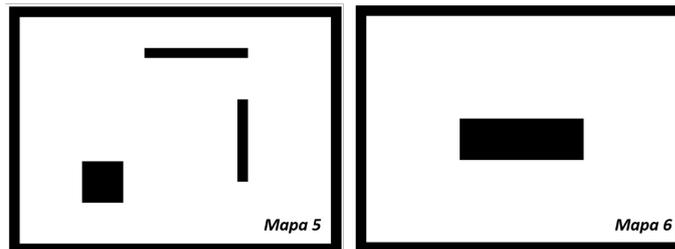


Figura 5.7: Mapas multiplamente conectados usados nos testes

Para estes testes foram criados novos conjuntos de posições para cada mapa, desta vez com 32 configurações de inicialização cada. Além disso, os experimentos não foram realizados somente com o número de robôs utilizados durante a evolução. Os testes foram também repetidos variando-se a quantidade de buscadores no ambiente, de 4 a 9 robôs (todos carregando em seus sistemas de controle a mesma solução sob análise). O principal objetivo foi verificar no sistema o efeito da redundância.

Na Tabela 5.4 são apresentados, para cada mapa e quantidade robôs, a média e o desvio padrão do sucesso de descontaminação das dez melhores soluções de cada um dos dez experimentos evolutivos.

Tabela 5.4: Descontaminação de mapas multiplamente conectados

Número de robôs	Taxa de sucesso	
	Mapa 5	Mapa 6
4	18% ± 12%	59% ± 20%
5	58% ± 17%	89% ± 10%
6	59% ± 20%	89% ± 13%
7	74% ± 20%	94% ± 13%
8	82% ± 17%	94% ± 8%
9	89% ± 14%	96% ± 6%

Os resultados mostrados na Tabela foram repetidos nos gráficos da Figura 5.8 para melhor visualização. É possível perceber com estes resultados que o sistema aceita bem a redundância, sendo que o aumento no número de robôs favorece o aumento na taxa de descontaminação dos mapas.

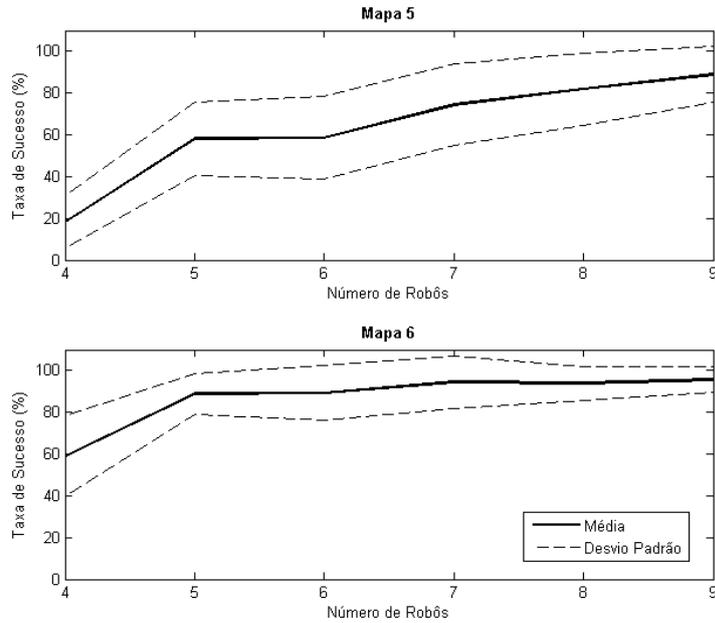


Figura 5.8: Taxa de sucesso na descontaminação dos novos mapas

Os últimos testes realizados para as soluções obtidas por meio da evolução foram feitos em mapas maiores, chamados aqui de *mapas conjugados*, por serem uma espécie de conexão dos mapas já apresentados, conforme mostrado na Figura 5.9. Os mapas conjugados 2 e 4 possuem 24×64 células e os mapas 1 e 3 têm 24×92 e 24×126 células, respectivamente.

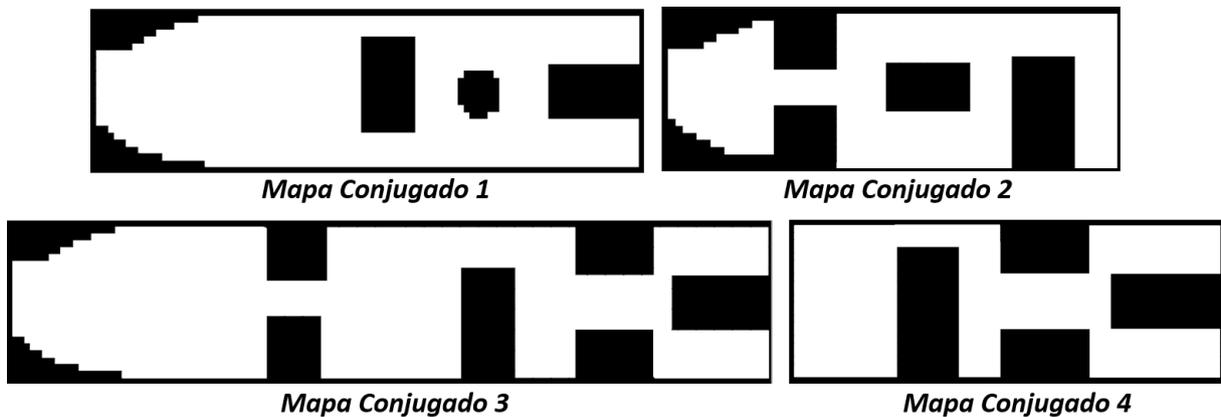


Figura 5.9: Mapas conjugados utilizados nos testes

As dez melhores soluções dos experimentos 8 e 10 foram escolhidas para os testes, que foram repetidos 30 vezes em cada mapa, com novas configurações de posição inicial, agora também com robôs começando deslocados na horizontal.

Considerando o tamanho e o aumento de complexidade dos mapas, os testes foram feitos para um grupo de 9 e outro de 15 robôs, com tolerância de recontaminação parcial

de 70 células, ou seja, se parte dos robôs abrem a formação e um número menor que 70 células sofre recontaminação, a simulação não é interrompida. Tal número foi estipulado com base na observação e nos maiores mapas utilizados, tendo em vista que em algumas ocasiões uma recontaminação parcial não compromete o cumprimento da tarefa. Foram permitidas simulações de até 400 iterações para o mapa conjugado 3, e 200 para os demais mapas.

Na Tabela 5.5 são apresentados os valores médios de sucesso na tarefa de descontaminação dos dois experimentos selecionados para cada um dos quatro mapas e para os dois grupos de robôs. Além disso, é também mostrada a média de iterações gasta para os casos bem sucedidos.

Tabela 5.5: Resultados para os quatro mapas conjugados

	Conjugado 1	Conjugado 2	Conjugado 3	Conjugado 4
9 robôs	5%	32%	1%	8%
Iterações	176	130	331	151
15 robôs	26%	54%	7%	12%
Iterações	162	122	328	166

Foi observado que na maioria dos casos de insucesso vários buscadores ficaram presos em trechos descontaminados do mapa, não atendendo à formação para que o restante dos buscadores pudesse prosseguir na descontaminação.

Buscando melhorar este resultado, foram realizadas três tentativas de continuar a evolução em um dos mapas conjugados, utilizando 7 robôs, a partir de uma população formada pelos melhores indivíduos dos experimentos anteriores. Contudo, a continuação da evolução por 200 gerações não resultou em nenhuma solução satisfatória.

Os insucessos registrados, tanto nos mapas conjugados quanto nos casos anteriores, foram atribuídos principalmente a dois fatos: (1) a função de custo mostrou-se altamente não linear, de forma que pequenas modificações no genótipo podem conduzir a grandes mudanças no comportamento do sistema, portanto o autômato não é capaz de generalizar para estados próximos; e (2) os 1.024 estados não são todos visitados para os mapas e posições de inicialização utilizados, e portanto não há garantia de que os estados sejam todos mapeados para as ações apropriadas.

Na verdade, apenas cerca de 50% dos possíveis estados foram visitados ao longo de todos os processos evolutivos. Com isso, a tentativa de alcançar a visitação de tais estados durante a evolução usaria um tempo proibitivo e exigiria a criação de diversos mapas em diversas inicializações, aproximando-se da ideia de uma busca exaustiva, o que não é o objetivo.

Por outro lado, é relevante investigar o impacto de tais estados não mapeados para o cumprimento da tarefa, e por isso é proposta uma abordagem complementar com a inserção de sorteios aleatórios para movimentar o robô nos casos em que o estado identificado

tenha sido pouco visitado durante a evolução e, portanto, não adequadamente mapeado para uma das ações α . A próxima seção comenta os resultados obtidos para esta nova análise.

5.3 Análise de Abordagem Complementar: aplicação de sorteios aleatórios

O sorteio aleatório aplicado nesta abordagem complementar tem probabilidade igualmente distribuída entre as possíveis ações de movimentação, de forma semelhante à demonstrada na Figura 2.4 para o trabalho de Gonçalves *et al.*, explicado na seção 2.4, porém sem a implementação de restrições, sem a possibilidade de permanecer na mesma célula e sem ponderação de probabilidades.

A proposta foi incluir o sorteio nesta nova abordagem para substituir o autômato evoluído sempre que o estado identificado por um robô não atender ao ponto de corte estabelecido quanto ao número de visitas, situação em que o mapeamento obtido na evolução está sendo considerado como não confiável. Na Figura 5.10 é dado um exemplo de escolha entre autômatos.

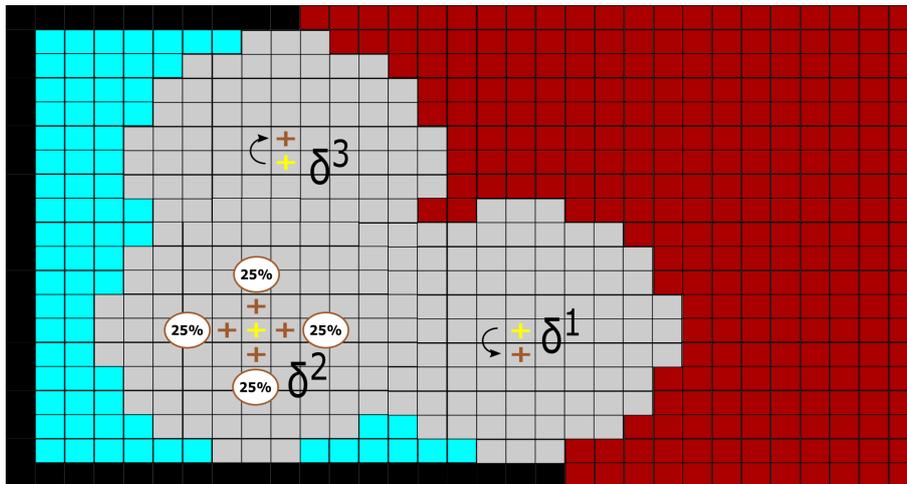


Figura 5.10: Exemplo de escolha de movimentação na abordagem complementar

No exemplo da figura, δ é a função de transição que associa um estado a uma ação, ou seja, representa o mapeamento. No caso de δ^1 e δ^3 , os estados individualmente identificados foram considerados suficientemente visitados, e portanto o mapeamento do autômato evoluído foi acessado e conduziu às ações correspondentes definidas no processo de evolução da solução incorporada: *mover uma célula ao sul* e *mover uma célula ao norte*, respectivamente. Já δ^2 representa a situação de um robô que identifica um estado cujo número de visitas não supera o ponto de corte estabelecido, portanto, a função de transição é substituída por um sorteio, em que uma das ações de movimentação indicadas é

aleatoriamente escolhida, sob igual probabilidade. Nota-se que a permanência na célula não é uma opção válida.

Se o robô na situação δ^2 utilizasse o mapeamento definido na evolução, muito provavelmente a ação de movimentação conduziria ao insucesso da tarefa, como ocorreu nos diversos testes apresentados. Aplicando esta abordagem complementar, é possível investigar parcialmente, mas de forma simples, os estados pouco, ou não visitados, verificando se a definição de novas ações resulta na melhora de desempenho e se tais estados realmente merecem uma maior atenção.

Foram escolhidos para os testes os mapas conjugados da Figura 5.9 e, assim como nas simulações anteriores com esses mesmos mapas, foram utilizados grupos de 9 e 15 robôs, sendo novamente aceita a recontaminação parcial de até 70 células.

Considerando agora a existência de uma componente aleatória, os testes foram realizados duas vezes e o limite de iterações aceito para a simulação foi passado para 500, com exceção do mapa conjugado 3, em que este valor foi estabelecido como 700.

O ponto de corte escolhido foi o de 100.000 visitas, de forma experimental e com base na observação da quantidade de visitação dos estados. A Tabela 5.6 apresenta os testes realizados também com os indivíduos dos experimentos 8 e 10 nas mesmas 30 posições utilizadas para as simulações cujos resultados foram apresentados na Tabela 5.5. Além disso, também é mostrado o percentual de iterações em que qualquer um dos robôs optou pelo sorteio, ao invés do autômato mapeado por meio da evolução.

Tabela 5.6: Resultados para os quatro mapas conjugados com aplicação do sorteio para estados pouco visitados

	Conjugado 1	Conjugado 2	Conjugado 3	Conjugado 4
9 robôs	10%	48%	8%	52%
Iterações	199	156	413	213
Sorteio	19,0%	18,9%	23,5%	18,4%
15 robôs	29%	70%	38%	57%
Iterações	201	146	398	200
Sorteio	23,3%	23,8%	27,3%	30,6%

Os resultados da Tabela 5.6 apresentam uma melhora média de 21% em relação aos testes realizados sem a aplicação do sorteio, indicando ser válida a investigação destes estados particulares, seja pela modificação do processo evolutivo, ou pela aplicação de alguma técnica de adaptação ou aprendizado *online* para ajustar os estados poucos visitados no mapeamento proposto na abordagem desta dissertação.

É interessante notar também que o percentual de escolha pelo sorteio e o número de iterações para convergência, principalmente considerando o tamanho dos mapas conjugados, aponta que os buscadores ainda foram majoritariamente guiados pela abordagem evolutiva.

Na próxima seção é apresentada uma análise comparativa da solução obtida somente

pela abordagem evolutiva neste trabalho, com uma outra metodologia para solução do problema de perseguição-evasão de pior caso [3], desenvolvida com base na técnica *random walk*.

5.4 Análise Comparativa

Ainda com a finalidade de estabelecer um parâmetro para avaliar a qualidade das soluções obtidas por meio da evolução, foi realizada uma análise comparativa com o algoritmo proposto por Gonçalves [3], que aplica *random walk* com ponderação de probabilidades, somada a restrições locais, para solução do mesmo problema de perseguição-evasão de pior caso. Os detalhes desta abordagem foram apresentados na seção 2.4.

A escolha do trabalho a ser comparado deu-se principalmente por tratar-se também de uma abordagem discreta, no mesmo ambiente de simulação utilizado nesta pesquisa, tendo sido aquele algoritmo adaptado para corresponder às novas definições de sensoriamento aqui propostas. Além disso, diversas escolhas para esta dissertação foram inspiradas no referido trabalho, de forma a facilitar esta análise comparativa.

Devido à aleatoriedade da metodologia aplicada naquela pesquisa, os testes foram realizados duas vezes, para as 12 inicializações da formação de cada mapa da Figura 5.1, mesmos conjuntos utilizados nos testes com a abordagem evolutiva aqui proposta. Os resultados foram verificados somente nos mapas simplesmente conectados porque o algoritmo de Gonçalves não é extensivo a mapas multiplamente conectados.

No caso do Mapa 4, foi necessário utilizar 5 robôs para cumprir a tarefa, conforme exigência apontada na pesquisa. Para todos os outros mapas e nas simulações das soluções evoluídas foram empregados 4 buscadores.

No que se refere ao sucesso na descontaminação dos mapas, as melhores soluções obtidas no processo evolutivo (na geração 300 do experimento 8) foram capazes de cumprir a tarefa para todos as posições em todos os mapas. O algoritmo baseado em *random walk* só foi considerado falho na descontaminação do Mapa 4 em uma das duas tentativas, devido ao número alto de iterações (170.000) sem que a tarefa fosse cumprida.

Na comparação de complexidade, foram consideradas as iterações necessárias para a descontaminação, sendo registradas apenas nos experimentos que foram bem sucedidos. Além disso, alguns casos com número muito alto de iterações foram considerados *outliers* e não foram mostrados no gráfico, para facilitar a visualização dos resultados, mostrados na Figura 5.11.

Na figura é apresentada a comparação estatística do número de iterações entre a abordagem proposta neste trabalho, identificada como ER, e a abordagem de Gonçalves, denominada RW, para cada um dos quatro mapas utilizados. O gráfico superior contempla todos os dados, com exceção dos pontos considerados *outliers*, acima de 3.500 iterações. Já o gráfico da parte inferior apresenta o detalhe para até 200 iterações, de forma que

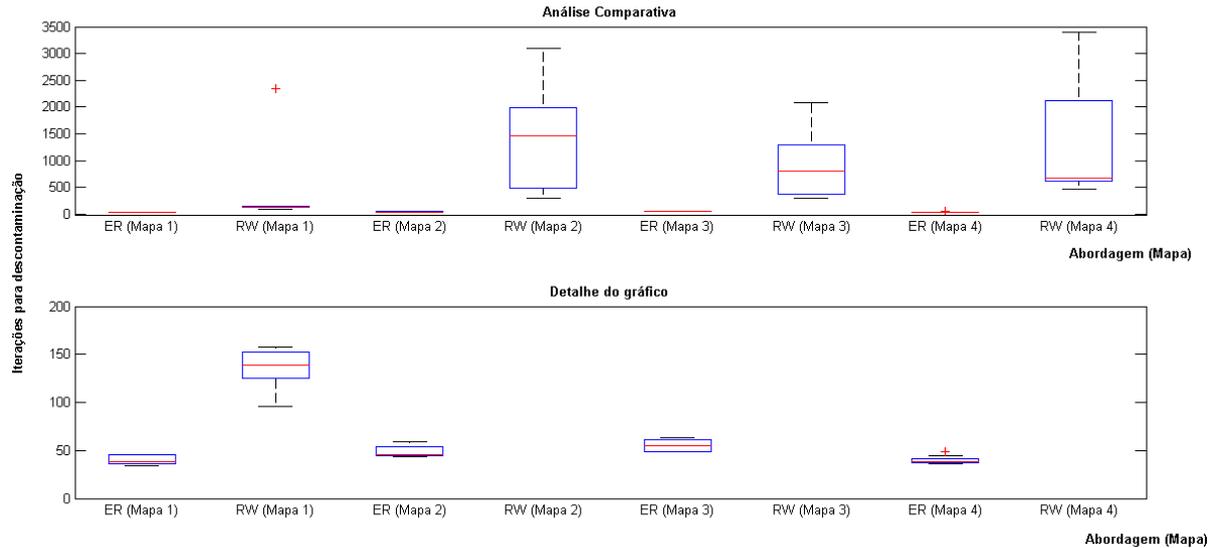


Figura 5.11: Análise comparativa entre as abordagens evolutiva (ER) e baseada em *random walk* (RW) [5]: em escala completa e com detalhe para até 200 iterações

sejam melhor visualizados os resultados da presente pesquisa.

Os gráficos da Figura 5.11 demonstram que as duas abordagens são estatisticamente diferentes. Apesar de o método proposto neste trabalho exigir uma etapa prévia de evolução, uma vez obtida a solução, o tempo de descontaminação do ambiente é consideravelmente mais baixo do que na aplicação do *random walk* ponderado, porque o completo processo de *busca* é realizado a cada novo experimento.

Além disso, não é contemplado no trabalho de Gonçalves o número de tentativas de sorteios até que todas as restrições impostas no trabalho sejam cumpridas. Este número de tentativas é muito mais elevado do que as iterações consideradas, e foi registrado nos experimentos realizados. Na Tabela 5.7 estão mostrados a média e o desvio padrão das iterações tidas como válidas e das tentativas para cada mapa.

Tabela 5.7: Média e desvio padrão das iterações válidas e das tentativas na abordagem de Gonçalves

	Iterações válidas	Tentativas
Mapa 1	318 ± 636	27.869 ± 90.602
Mapa 2	1.422 ± 870	11.320 ± 8.180
Mapa 3	888 ± 573	30.564 ± 32.030
Mapa 4	7.298 ± 17.396	123.557 ± 245.864

Como mostrado na tabela, é grande o número de tentativas que o algoritmo faz até que as restrição impostas sejam atendidas. Isto impacta bastante no tempo necessário para que a limpeza dos ambientes seja completa. Se forem considerados ainda mapas maiores e um maior número de robôs, a complexidade será significativamente aumentada, o que

não ocorre para a abordagem com autômato evoluído, como mostrado na seção anterior.

Além disso, o fato de o algoritmo não ser extensivo para mapas multiplamente conectados restringe bastante sua aplicação. Já a abordagem aqui proposta, apesar de exigir uma etapa de evolução prévia no método, uma vez realizada, apresenta soluções capazes de descontaminar mapas simples ou multiplamente conectados, com um número bem menor de iterações, sem utilizar informação global e de forma descentralizada.

Por outro lado, a abordagem de Golçalves, apesar de não possuir provas teóricas, tende a garantir a descontaminação de qualquer mapa simplesmente conectado para todas as inicializações de posição, em um tempo menor que infinito, robustez não apresentada na proposição desta dissertação.

Cada uma das abordagens apresenta, portanto, vantagens e desvantagens, e foi o objetivo desta seção levantar algumas comparações a respeito de ambas.

Neste capítulo foram apresentados principalmente os resultados para diversos experimentos e análises a respeito da abordagem evolutiva proposta, e o próximo capítulo discute as conclusões gerais obtidas com o desenvolvimento desta pesquisa, e aponta possíveis investigações para trabalhos futuros.

Capítulo 6

Conclusões

Nesta dissertação de mestrado é proposta uma abordagem evolutiva para solução do problema de perseguição-evasão de pior caso. Os robôs possuem o mesmo sistema de controle baseado em autômato finito, cujo mapeamento de estados em ações é sujeito ao algoritmo evolutivo. A evolução é guiada pela avaliação do comportamento coletivo resultante dos robôs, simulados em mapas discretos.

O objetivo foi alcançado, no sentido de que as soluções encontradas com um número razoavelmente baixo de gerações mostraram-se satisfatórias para resolver o problema em alguns, mas representativos, tipos de mapas, inclusive multiplamente conectados, que não foram apresentados durante o processo evolutivo. Além disso, foram também capazes de generalizar para diversas condições iniciais e para um maior número de robôs.

Apesar disso, nenhuma solução encontrada nos experimentos realizados foi capaz de generalizar para todas as condições iniciais em todos os tipos de mapas. A hipótese é de que a razão esteja no mapeamento incompleto realizado pelo processo evolutivo, uma vez que o mesmo não garante que todos os estados sejam visitados nem que o autômato seja corretamente definido.

Tal hipótese sustenta-se após os testes realizados em conjunto com sorteios aleatórios, aplicados para definir a ação de estados pouco visitados durante a evolução, e cujos resultados demonstraram que as soluções melhoraram sua taxa de sucesso na descontaminação dos mapas conjugados em 21%, considerando a média de todos os mapas dos grupos de 9 e 15 robôs.

Por outro lado, a análise comparativa de resultados realizada com o trabalho de Gonçalves mostrou que a abordagem evolutiva proposta, apesar de exigir uma etapa prévia no método, apresenta uma solução capaz de descontaminar os mesmos mapas com um número bem menor de iterações. Além disso, não usa informação global e é extensível a mapas multiplamente conectados, ao contrário do referido trabalho, cuja vantagem recai na possível garantia de descontaminação de todo mapa simplesmente conectado para qualquer inicialização em um tempo menor que infinito.

É possível citar que esta dissertação apresenta como contribuição: a investigação evo-

lutiva do problema de perseguição-evasão de pior caso, a proposição do autômato com sua definição local dos estados, a comparação com outro método de solução para o mesmo problema e a proposta de investigar os autômatos ajustados segundo a robótica evolutiva com os sorteios aleatórios.

Uma vez que algumas modificações nas ações dos robôs resultaram em melhoria no comportamento coletivo para descontaminação dos mapas conjugados, acredita-se que um trabalho futuro relevante seja a utilização de uma técnica de aprendizado por reforço, como o *learning automata*, a partir das soluções evoluídas, para que os estados não corretamente definidos possam ser ajustados de forma *online*.

Outras extensões da mesma abordagem podem ser também estudadas, principalmente aproveitando os diversos trabalhos que investigam variações na aplicação da robótica evolutiva, inclusive conjugando o aprendizado.

Ainda, é válido considerar outras formas de definição do vetor de estado a partir da percepção dos buscadores ou dos estados não visitados, podendo ser aplicada, por exemplo, a teoria da entropia, para analisar as dimensões dos estados propostos e possivelmente obter alguma redução ou redefinição do vetor.

É sugerido também o estudo do mesmo problema considerando a movimentação dos robôs na diagonal, assim como o estudo dos parâmetros do algoritmo genético para esta aplicação, como tamanho da população e taxa de mutação, aspectos não investigados devido ao elevado tempo necessário para repetir os experimentos evolutivos, tendo em vista a necessidade de interação com o ambiente para que o comportamento das soluções candidatas seja avaliado.

Além disso, modificações no algoritmo, como a implementação de paralelismo, são sugeridas a fim de reduzir o tempo necessário para obter as soluções por meio do processo evolutivo.

Referências Bibliográficas

- [1] Andreas Kolling e Stefano Carpin. Multi-robot pursuit-evasion without maps. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp 3045–3051. IEEE, 2010.
- [2] Joseph W Durham, Antonio Franchi e Francesco Bullo. Distributed pursuit-evasion without mapping or global localization via local frontiers. *Autonomous Robots*, 32(1):81–95, 2012.
- [3] Antonio Gonçalves. Problema de perseguição-evasão baseado em random walk. Master’s thesis, São Cristóvão, 2015.
- [4] Stephane Doncieux, Nicolas Bredeche, Jean-Baptiste Mouret e Agoston E. (Gusz) Eiben. Evolutionary robotics: What, why, and where to. *Frontiers in Robotics and AI*, 2:4, 2015.
- [5] Antonio Gonçalves, Elyson Carvalho, Lucas Molina e Eduardo Freire. Problema de perseguição-evasão baseado em random walk. In *XII Simpósio Brasileiro de Automação Inteligente (SBAI)*, pp 1460–1465, Oct 2015.
- [6] Y Uny Cao, Alex S Fukunaga e Andrew Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, 4(1):7–27, 1997.
- [7] Avinash Gautam e Sudeept Mohan. A review of research in multi-robot systems. In *2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS)*, pp 1–5. IEEE, 2012.
- [8] Alessandro Farinelli, Luca Iocchi e Daniele Nardi. Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(5):2015–2028, 2004.
- [9] Toshio Fukuda, Seiya Nakagawa, Yoshio Kawauchi e Martin Buss. Structure decision method for self organising robots based on cell structures-cebot. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp 695–700. IEEE, 1989.

- [10] Yi-Lin Liao e Kuo-Lan Su. Multi-robot-based intelligent security system. *Artificial Life and Robotics*, 16(2):137–141, 2011.
- [11] Yugang Liu e Goldie Nejat. Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent & Robotic Systems*, 72(2):147–165, 2013.
- [12] David Portugal e Rui Rocha. A survey on multi-robot patrolling algorithms. In *Doctoral Conference on Computing, Electrical and Industrial Systems*, pp 139–146. Springer, 2011.
- [13] Alaa Khamis e Asser ElGindy. Minefield mapping using cooperative multirobot systems. *Journal of Robotics*, 2012, 2012.
- [14] Florian Shkurti, Anqi Xu, Malika Meghjani, Juan Camilo Gamboa Higuera, Yogesh Girdhar, Philippe Giguere, Bir Bikram Dey, Jimmy Li, Arnold Kalmbach, Chris Prahacs et al. Multi-domain monitoring of marine environments using a heterogeneous robot team. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 1747–1753. IEEE, 2012.
- [15] Rufus Isaacs. Differential games, a mathematical theory with applications to optimization, control and warfare, 1965.
- [16] Rufus Isaacs. Differential games i: Introduction. Technical report, DTIC Document, 1954.
- [17] Fedor V FOMIN e Dimitrios M THILIKOS. An annotated bibliography on guaranteed graph searching. *Theoretical computer science*, 399(3):236–245, 2008.
- [18] TV Abramovskaya e NN Petrov. The theory of guaranteed search on graphs. *Vestnik St. Petersburg University: Mathematics*, 46(2):49–75, 2013.
- [19] Timothy H Chung, Geoffrey A Hollinger e Volkan Isler. Search and pursuit-evasion in mobile robotics. *Autonomous robots*, 31(4):299–316, 2011.
- [20] TD Parsons. Pursuit-evasion in a graph. theory and applications of graphs (pp. 426441), 1978.
- [21] Lívia Gregorin, Sidney Givigi, Eduardo Freire, Elyson Carvalho e Lucas Molina. Evolutionary robotics applied to the multi-robot worst-case pursuit-evasion problem. In *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2016.
- [22] Josh C Bongard. Evolutionary robotics. *Communications of the ACM*, 56(8):74–83, 2013.

- [23] Dario Floreano, Phil Husbands e Stefano Nolfi. Evolutionary robotics. In *Springer handbook of robotics*, pp 1423–1451. Springer, 2008.
- [24] Geoffrey F Miller e Dave Cliff. *Co-evolution of pursuit and evasion I: Biological and game-theoretic foundations*. University of Sussex, School of Cognitive and Computing Sciences, 1994.
- [25] Matt Quinn, Lincoln Smith, Giles Mayley e Phil Husbands. Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 361(1811):2321–2343, 2003.
- [26] Gianluca Baldassarre, Stefano Nolfi e Domenico Parisi. Evolving mobile robots able to display collective behaviors. *Artificial life*, 9(3):255–267, 2003.
- [27] Nikhil Karnad e Volkan Isler. Lion and man game in the presence of a circular obstacle. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 5045–5050. IEEE, 2009.
- [28] R Isaacs. Games of pursuit, paper no. *P-257*, RAND Corporation, Santa Monica, California, 1951.
- [29] Lefteris M Kirousis e Christos H Papadimitriou. Searching and pebbling. *Theoretical Computer Science*, 47:205–218, 1986.
- [30] Daniel Bienstock e Paul Seymour. Monotonicity in graph searching. *Journal of Algorithms*, 12(2):239–245, 1991.
- [31] Lali Barriere, Pierre Fraigniaud, Nicola Santoro e Dimitrios M Thilikos. Searching is not jumping. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pp 34–45. Springer, 2003.
- [32] Lali Barrière, Paola Flocchini, Pierre Fraigniaud e Nicola Santoro. Capture of an intruder by mobile agents. In *Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures*, pp 200–209. ACM, 2002.
- [33] Geoffrey Hollinger, Athanasios Kehagias e Sanjiv Singh. Gsst: anytime guaranteed search. *Autonomous Robots*, 29(1):99–118, 2010.
- [34] Andreas Kolling e Stefano Carpin. The graph-clear problem: definition, theoretical properties and its connections to multirobot aided surveillance. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 1003–1008. IEEE, 2007.

- [35] Andreas Kolling e Stefano Carpin. Probabilistic graph-clear. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp 3508–3514. IEEE, 2009.
- [36] Ichiro Suzuki e Masafumi Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal on computing*, 21(5):863–888, 1992.
- [37] Steven M LaValle, David Lin, Leonidas J Guibas, J-C Latombe e Rajeev Motwani. Finding an unpredictable target in a workspace with obstacles. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pp 737–742. IEEE, 1997.
- [38] Leonidas J Guibas, Jean-Claude Latombe, Steven M LaValle, David Lin e Rajeev Motwani. A visibility-based pursuit-evasion problem. *International Journal of Computational Geometry & Applications*, 9(04n05):471–493, 1999.
- [39] Nicholas M Stiffler e Jason M O’Kane. A complete algorithm for visibility-based pursuit-evasion with multiple pursuers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp 1660–1667. IEEE, 2014.
- [40] Shai Sachs, Steven M LaValle e Stjepan Rajko. Visibility-based pursuit-evasion in an unknown planar environment. *The International Journal of Robotics Research*, 23(1):3–26, 2004.
- [41] Andrew L Nelson, Gregory J Barlow e Lefteris Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370, 2009.
- [42] Inman Harvey, Ezequiel Di Paolo, Rachel Wood, Matt Quinn e Elio Tuci. Evolutionary robotics: A new scientific tool for studying cognition. *Artificial life*, 11(1-2):79–98, 2005.
- [43] Max Lungarella, Giorgio Metta, Rolf Pfeifer e Giulio Sandini. Developmental robotics: a survey. *Connection Science*, 15(4):151–190, 2003.
- [44] Rolf Pfeifer e Josh Bongard. *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.
- [45] Patricia A Vargas, Ezequiel A Di Paolo, Inman Harvey e Phil Husbands. *The horizons of evolutionary robotics*. MIT Press, 2014.
- [46] Stefano Nolfi e Dario Floreano. *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT press, 2000.

- [47] Agnes Pintér-Bartha, Anita Sobe e Wilfried Elmenreich. Towards the light?comparing evolved neural network controllers and finite state machine controllers. In *Intelligent Solutions in Embedded Systems (WISES), 2012 Proceedings of the Tenth Workshop on*, pp 83–87. IEEE, 2012.
- [48] Suranga Hettiarachchi, William M Spears, Lukas König, Sanaz Mostaghim e Hartmut Schmeck. Decentralized evolution of robotic behavior using finite state machines. *International Journal of Intelligent Computing and Cybernetics*, 2(4):695–723, 2009.
- [49] Grant Dick e Xin Yao. Model representation and cooperative coevolution for finite-state machine evolution. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pp 2700–2707. IEEE, 2014.
- [50] Dario Floreano, Jean-Christophe Zufferey e Jean-Daniel Nicoud. From wheels to wings with evolutionary spiking circuits. *Artificial Life*, 11(1-2):121–138, 2005.
- [51] Hod Lipson e Jordan B Pollack. Automatic design and manufacture of robotic life-forms. *Nature*, 406(6799):974–978, 2000.
- [52] Sangbae Kim, Cecilia Laschi e Barry Trimmer. Soft robotics: a bioinspired evolution in robotics. *Trends in biotechnology*, 31(5):287–294, 2013.
- [53] Mark Yim, Wei-Min Shen, Behnam Salemi, Daniela Rus, Mark Moll, Hod Lipson, Eric Klavins e Gregory S Chirikjian. Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):43–52, 2007.
- [54] Marco Dorigo, Mauro Birattari e Manuele Brambilla. Swarm robotics. *Scholarpedia*, 9(1):1463, 2014.
- [55] Gianpiero Francesca, Manuele Brambilla, Arne Brutschy, Vito Trianni e Mauro Birattari. Automode: A novel approach to the automatic design of control software for robot swarms. *Swarm Intelligence*, 8(2):89–112, 2014.
- [56] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari e Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.
- [57] Thomas Schmickl, Payam Zahadat e Heiko Hamann. Sooner than expected: Hitting the wall of complexity in evolution. *arXiv preprint arXiv:1609.07722*, 2016.
- [58] Vito Trianni e Stefano Nolfi. Engineering the evolution of self-organizing behaviors in swarm robotics: A case study. *Artificial Life*, 17(3):183–202, 2011.

- [59] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics*, pp 10–20. Springer, 2004.
- [60] Dave Cliff e Geoffrey F Miller. Co-evolution of pursuit and evasion ii: Simulation methods and results. 1995.
- [61] Stefano Nolfi e Dario Floreano. Coevolving predator and prey robots: do ?arms races? arise in artificial evolution? *Artificial life*, 4(4):311–335, 1998.
- [62] Stefano Nolfi. Co-evolving predator and prey robots. *Adaptive Behavior*, 20(1):10–15, 2012.
- [63] Geoff Nitschke. Co-evolution of cooperation in a pursuit evasion game. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pp 2037–2042. IEEE, 2003.
- [64] Fani A Tzima, Andreas L Symeonidis e Pericles A Mitkas. Symbiosis: Using predator-prey games as a test bed for studying competitive co-evolution. In *2007 International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pp 115–120. IEEE, 2007.
- [65] Geoff S Nitschke e Leo H Langenhoven. Neuro-evolution for competitive co-evolution of biologically canonical predator and prey behaviors. In *Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on*, pp 546–553. IEEE, 2010.
- [66] Nick Jakobi, Phil Husbands e Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *European Conference on Artificial Life*, pp 704–720. Springer, 1995.
- [67] Fernando Silva, Miguel Duarte, Luís Correia, Sancho Moura Oliveira e Anders Lyhne Christensen. Open issues in evolutionary robotics. *Evolutionary computation*, 24(2):205–236, 2016.
- [68] Stephane Doncieux e Jean-Baptiste Mouret. Beyond black-box optimization: a review of selective pressures for evolutionary robotics. *Evolutionary Intelligence*, 7(2):71–93, 2014.
- [69] Nicolas Bredeche e Jean-Marc Montanier. Environment-driven open-ended evolution with a population of autonomous robots. In *Evolving Physical Systems Workshop*, 2012.
- [70] Joel Lehman e Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.

- [71] Stéphane Doncieux, Jean-Baptiste Mouret, Nicolas Bredeche e Vincent Padois. Evolutionary robotics: Exploring new horizons. In *New horizons in evolutionary robotics*, pp 3–25. Springer, 2011.
- [72] Agoston E Eiben e Jim Smith. From evolutionary computation to the evolution of things. *Nature*, 521(7553):476–482, 2015.
- [73] Marco Dorigo, Dario Floreano, Luca Maria Gambardella, Francesco Mondada, Stefano Nolfi, Tarek Baaboura, Mauro Birattari, Michael Bonani, Manuele Brambilla, Arne Brutschy et al. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4):60–71, 2013.
- [74] Jianhua Yang, Yabo Liu, Zhaohui Wu e Min Yao. The evolution of cooperative behaviours in physically heterogeneous multi-robot systems. *International Journal of Advanced Robotic Systems*, 9, 2012.
- [75] Andreas Kolling, Alexander Kleiner e Piotr Rudol. Fast guaranteed search with unmanned aerial vehicles. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp 6013–6018. IEEE, 2013.
- [76] Ricardo Linden. *Algoritmos genéticos (3a edição)*. Ciencia Moderna, 2012.
- [77] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [78] SN Sivanandam e SN Deepa. *Introduction to genetic algorithms*. Springer Science & Business Media, 2007.
- [79] Peter J Fleming e Robin C Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control engineering practice*, 10(11):1223–1241, 2002.
- [80] John E Hopcroft, Rajeev Motwani e Jeffrey D Ullman. Automata theory, languages, and computation.

Apêndice A

Pseudo-código

Algorithm 1 Estrutura do algoritmo evolutivo

- 1: Definição dos parâmetros
 - 2: Inicialização da população de cromossomos
 - 3: **para** $geracao = 1 : NumeroDeGeracoes$ **faça**
 - 4: **para** $individuo = 1 : TamanhoDaPopulacao$ **faça**
 - 5: Atribua a mesma solução candidata representada pelo indivíduo ao sistema de controle de todos os robôs
 - 6: Sorteie as posições iniciais dentro das faixas de restrição para o Mapa 1
 - 7: Simule os robôs no Mapa 1
 - 8: Compute a avaliação parcial
 - 9: Sorteie as posições iniciais dentro das faixas de restrição para o Mapa 2
 - 10: Simule os robôs no Mapa 2
 - 11: Compute a avaliação parcial
 - 12: Sorteie as posições iniciais dentro das faixas de restrição para o Mapa 3
 - 13: Simule os robôs no Mapa 3
 - 14: Compute a avaliação parcial
 - 15: Sorteie as posições iniciais dentro das faixas de restrição para o Mapa 4
 - 16: Simule os robôs no Mapa 4
 - 17: Compute a avaliação parcial
 - 18: Some as avaliações parciais para obter a *fitness* final do indivíduo
 - 19: **fim para**
 - 20: Ordene os indivíduos segundo a *fitness*, do melhor avaliado para o pior
 - 21: Atribua a elite às primeiras posições da população da próxima geração
 - 22: Selecione os pais por meio do torneio entre dois indivíduos
 - 23: Aplique o operador *crossover* de 20 pontos, segundo a *Probabilidade_de_crossover* e complete a população da próxima geração
 - 24: Aplique o operador de mutação em 20% dos genes, segundo a *Probabilidade_de_mutacao*
 - 25: Despreze a população anterior
 - 26: **fim para**
-