



ANÁLISE DE DESEMPENHO DE DIFERENTES FUNÇÕES DE FITNESS PARA O ALGORITMO PSO APLICADO A COORDENAÇÃO DE ENXAME DE ROBÔS DE SOLO

Trabalho de Conclusão de Curso

Engenharia da Computação

**Aluno: Clodomir Joaquim de Santana Junior
Orientador: Prof. Dr. Sergio Campello Oliveira**



**UNIVERSIDADE
DE PERNAMBUCO**

**Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação**

**CLODOMIR JOAQUIM DE SANTANA
JUNIOR**

**ANÁLISE DE DESEMPENHO DE
DIFERENTES FUNÇÕES DE FITNESS
PARA O ALGORITMO PSO APLICADO
A COORDENAÇÃO DE ENXAME DE
ROBÔS DE SOLO**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, Julho de 2016.

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 1 de julho de 2016, às 14:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente CLODOMIR JOAQUIM DE SANTANA JUNIOR, orientado pelo professor Sérgio Campello Oliveira, sob título ANÁLISE DE DESEMPENHO DE DIFERENTES FUNÇÕES DE FITNESS PARA O ALGORITMO PSO APLICADO A COORDENAÇÃO DE ENXAME DE ROBÔS DE SOLO, a banca composta pelos professores:

Gustavo Oliveira Cavalcanti

Sérgio Campello Oliveira

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 9,5 (nove e meio)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 15 dias para entrega da versão final da monografia a contar da data deste documento.

GUSTAVO OLIVEIRA CAVALCANTI

SÉRGIO CAMPELLO OLIVEIRA

Dedicatória

Dedico esse trabalho a Deus que permitiu que pudesse ser realizado, a meus pais e amigos pela força, conselhos e por ter acreditado em mim, e a todos que de forma direta ou indireta contribuíram para que este trabalho se concretizasse.

Agradecimentos

Agradeço a Deus por ser minha fonte de força, coragem, fé e abrigo. A Ele toda glória e louvor, pois sem Ele nada disso seria possível.

Agradeço à minha mãe, Nerize Santana, meu pai, Clodomir Santana e minha irmã, Monaliza Santana, pelo apoio que sempre me deram, pelos conselhos, incentivos, e por tudo o que fizeram e fazem por mim.

Agradeço aos meus amigos, por terem acreditado em mim e por estarem comigo em todos os momentos, pelo suporte, e bons momentos.

Ao meu orientador, o professor Dr. Sérgio Campello Oliveira, por estar trabalhando comigo desde meus primeiros períodos na faculdade e por ser um excelente orientador.

Ao meu colega de pesquisa, Caio Albuquerque, que enfrentou comigo os desafios das nossas pesquisas.

Resumo

Este trabalho tem como objetivo apresentar uma análise de desempenho de três funções de avaliação (*Fitness Function*) para o Algoritmo de Otimização por Enxame de Partículas (do inglês, PSO) aplicado à coordenação de um enxame de robôs terrestres autônomos. O enxame é composto por robôs que possuem o objetivo de percorrer um ambiente desconhecido em busca de alvos fixos, cobrindo a maior área possível.

Objetivando obter dados que auxiliassem a verificação do grau de influência que a função de avaliação tem sobre o desempenho do algoritmo, foram utilizadas três diferentes estratégias de função: *fitness* inversamente proporcional à distância Euclidiana entre o robô e o alvo mais próximo, *fitness* diretamente proporcional à distância Euclidiana entre a partícula e seu ponto de partida no ambiente, e a geração pseudoaleatória de alvos chamados “alvos fantasmas” que são perseguidos pelo robô até que alvos reais sejam detectados.

Sabendo que a comunicação entre os robôs é fundamental em um enxame, um protocolo de comunicação baseado em pacotes que permite a troca informações entre os robôs, foi desenvolvido e incorporado ao PSO. O PSO também foi modificado para incluir um mecanismo capaz de detectar e evitar colisões com obstáculos. Para fazer a modelagem do cenário e dos robôs, assim como para a execução e o gerenciamento das simulações, a ferramenta V-REP (*Virtual Robot Experimentation Platform*) foi adotada.

Os resultados das análises feitas, permitiram identificar que tanto a área de cobertura quanto a taxa de sucesso são influenciadas pelo tamanho e pela capacidade de comunicação do enxame. Os resultados ainda permitiram identificar a viabilidade da aplicação de enxame de robôs para busca de alvo fixos em ambientes desconhecidos. Além disso, as três funções de *fitness* apresentaram desempenho de área de cobertura e taxa de sucesso satisfatórios, mesmo para as duas funções de *fitness* não necessitam de informações da localização dos alvos.

Abstract

This work aims to present an analysis of three fitness functions used on Particle Swarm Optimization algorithm applied to the coordination of a swarm of autonomous ground robots. The swarm is composed of robots which have the objective of to go explore an unknown environment searching for fixed targets, covering the largest area possible.

In order to obtain data that would help to check the degree of influence that the fitness function has on the performance of the algorithm, three different function strategies were used: fitness inversely proportional to the Euclidean distance between the robot and the nearest target, fitness directly proportional to the Euclidean distance between the particle and its starting point in the environment, and the pseudorandom generation of targets called "phantom targets" that are pursued by the robot until real targets are detected.

Given the importance the communication between the robots in a swarm, a packet-based communication protocol that allows robots to exchange information was developed and incorporated into the PSO. Moreover, the PSO was also modified to include a mechanism to detect and avoid collisions with obstacles. In order to model the environment and the robots, as well as for the implementation and management of the simulations, the V-REP (Virtual Robot Experimentation Platform) framework was adopted.

The results of the analysis, showed that both the coverage area as the success rate are influenced by size and communication skills of the swarm. They also helped to identify the feasibility of robot swarm application to search fixed target in unknown environments. Moreover, the three fitness functions presented satisfactory performance of coverage area and success rate, even when the fitness functions do not need information of location of the targets.

Sumário

Capítulo 1 Introdução	1
Capítulo 2 Fundamentação Teórica	5
1.1 Inteligência de Enxames	5
1.2 Robótica de Enxames	8
1.3 Protocolo de Comunicação	11
1.4 V-REP	13
Capítulo 3 Trabalhos Relacionados	16
1.1 Algoritmo das Abelhas (Bees Algorithm - BA)	16
1.1.1 Algoritmo de Abelhas Modificado (Modified Bees Algorithm – MBA)	16
1.1.2 Algoritmo das Abelhas + Módulo de Detecção e Desvio de Obstáculos	17
1.2 Otimização por Colônia de Formigas (Ant Colony Optimization – ACO)	17
1.2.1 Colônia de Formigas + Enxame de Partículas	18
1.2.2 Colônia de Formigas + Função de Potencial Artificial - APF	19
1.3 Otimização por Cultura de Bactéria (Bacterial Foraging Optimization - BFO)	20
1.3.1 Algoritmo Modificado de Otimização por Cultura de Bactéria (Modified Bacterial Foraging Optimization – MBFO)	20
1.4 Otimização por Enxame de Partículas - PSO	21
1.4.1 Otimização por Enxame de Partículas com Limitação de Velocidade e Método Lagrangiano Aumentado (VL-ALPSO)	21
1.4.2 Otimização por Enxame de Partículas Dinâmico (Dynamic Particle Swarm Optimization – DPSO)	22

Capítulo 4 Experimentos	23
Capítulo 5 Resultados	27
Capítulo 6 Conclusões	33
Referências Bibliográficas	35

Índice de Figuras

Figura 1. Fluxograma do PSO.....	6
Figura 2. Estrutura de uma mensagem.....	12
Figura 3. Fluxograma do PSO com Protocolo de Comunicação.....	13
Figura 4. Interface do V-REP Pro Edu.....	14
Figura 5. Ambiente e componentes criados no V-REP.....	23
Figura 6. Robô K-Junior Utilizado nas Simulações.....	25
Figura 7. Área de Cobertura para Função de Avaliação 1.....	27
Figura 8. Área de Cobertura para Função de Avaliação 2.....	28
Figura 9. Área de Cobertura para Função de Avaliação 3.....	28
Figura 10. Taxa de Sucesso para Função de Avaliação 1.....	29
Figura 11. Taxa de Sucesso para Função de Avaliação 2.....	30
Figura 12. Taxa de Sucesso para Função de Avaliação 3.....	30
Figura 13. Trajetórias traçadas pelos robôs.....	32

Índice de Tabelas

Tabela 1. Parâmetros utilizados no PSO	7
Tabela 2. Comparação entre SR e MRS.....	9
Tabela 3. Especificações do K-Junior	24
Tabela 4. Parâmetros utilizados nas execuções	26

Tabela de Símbolos e Siglas

ACO – *Ant Colony Optimization* (Otimização por Colônia de Formigas)

AL - *Augmented Lagrangian* (Lagrangiano Aumentado)

APF - *Artificial Potential Function* (Função de Potencial Artificial)

API - *Application Programming Interface* (Interface de Programação de Apicativo)

BA – *Bees Algorithm* (Algoritmo das Abelhas)

BFO – *Bacteria Foraging Optimization* (Otimização por Cultura de Bactéria)

DPSO – *Dynamic Particle Swarm Optimization* (Otimização por Enxame de Partículas Dinâmica)

ER – Enxame de Robôs

GA – *Genetic Algorithm* (Algoritmo Genético)

gBest – *Global Best* (Melhor resultado obtido pelo enxame)

IA – Inteligência Artificial

IE – Inteligência de Enxame

lBest – *Local Best* (Melhor resultado encontrado na vizinhança de uma partícula)

MBA – *Modified Bees Algorithm* (Algoritmo das Abelhas Modificado)

MBFO – *Modified Bacteria Foraging Optimization* (Otimização por Cultura de Bactéria Modificado)

MRS – *Multi-robot Systems* (Sistemas Multi-robôs)

NSGA II - *Non-Dominated Sorting Genetic Algorithm II* (Algoritmo Genético com Seleção Não-Dominada II)

pBest – *Particle's Best* (Melhor resultado encontrado pela partícula)

PSO – *Particle Swarm Optimization* (Otimização por Enxame de Partículas)

PIC - *Peripheral Interface Controller* (Interface Controladora de Periféricos)

SI – *Swarm Intelligence* (Inteligência de Enxames)

SR – *Swarm Robotics* (Robótica de Enxame)

VL-ALPSO - *Augmented Lagrangian Particle Swarm Optimization with Special Velocity Limits* (PSO Lagrangiano Aumentado com Limite de Velocidade)

V-REP - *Virtual Robot Experimentation Platform* (Plataforma Virtual para Experimentos com Robôs)

Capítulo 1

Introdução

Devido aos avanços nas tecnologias de desenvolvimento de sistemas computacionais, bem como os avanços nas áreas de inteligência computacional e robótica, o uso de robôs para realização de tarefas de busca e mapeamento é algo que vem se tornando cada vez mais frequente [1]. Busca por sobreviventes dentro de prédios em chamas, busca por artefatos explosivos nos casos de alerta de bomba [2], exploração de planetas ou ambientes de difícil acesso [3], são exemplos de problemas do mundo real em que robôs vem sendo empregados em atividades de busca e mapeamento.

Embora haja aumento no número de aplicações de robôs para a realização de atividades de busca e mapeamento, grande parte das soluções existentes dependem de supervisão ou intervenção remota de humanos [4,5,6,7]. Um exemplo disso são os robôs utilizados para busca e desativação de artefatos explosivos, que são remotamente controlados por um operador humano [8]. Objetivando criar uma solução mais autônoma para esse tipo de problema, pesquisadores têm aplicado conceitos e algoritmos da Inteligência Computacional (IA) em robôs [9,10,11].

A partir da estratégia de se combinar Inteligência Computacional com robótica, surgiu um ramo chamado Robótica de Enxame, do inglês *Swarm Robotics (SR)* [12]. Ou seja, Robótica de Exame pode ser defina como a aplicação de conceitos e técnicas da Inteligência de Enxame (ramo da IA) na coordenação de sistemas multi-robôs. Esses sistemas multi-robôs recebem o nome de enxame de robôs e, assim como na Inteligência de Enxame (IE), os indivíduos (robôs) de um enxame executam tarefas simples, interagem com o ambiente e entre si com o intuito de alcançar determinado objetivo e têm seu desempenho individual avaliado por uma função de avaliação (*fitness*). Os indivíduos são relativamente homogêneos e, na maioria dos casos, o comportamento deles é descrito em termos de funções probabilísticas que dependem do local que o indivíduo tem e da sua vizinhança. Além dessas características,

sistemas de múltiplos robôs ainda devem manter três propriedades funcionais que são observadas em grupos naturais: robustez, flexibilidade e escalabilidade [13].

Dentro da Inteligência de Enxames existem vários algoritmos como o de Otimização por Colônia de Formigas (*Ant Colony Optimization* – ACO) [14], Otimização por Enxame de Partículas (*Particle Swarm Optimization* – PSO) [15], que são utilizados para a coordenação do enxame e podem ser adaptados para uso em enxames de robôs. Independentemente do algoritmo selecionado, os indivíduos do enxame deverão ter algum mecanismo de interação que pode variar de acordo com o algoritmo selecionado, e ele é fundamental para que o conhecimento do enxame emergja das iterações diretas (comunicação direta entres os elementos do enxame) ou indireta (comunicação através do ambiente). Outro fator de importância para o enxame é a função de avaliação adotada pois, como o nome sugere, é uma indicação de que o indivíduo está ou não próximo de alcançar o seu objetivo.

Existem algumas diferenças entre simular um enxame de partículas e um enxame de robôs, uma delas é que, no enxame de robôs se faz necessária a existência de um protocolo de comunicação, visto que os robôs do enxame usarão alguma tecnologia de transmissão de dados (*Bluetooth*, Infravermelho, *WiFi*, etc.) e precisarão de um mecanismo de gerenciamento e tratamento dessas informações. Além disso, se tratando de robôs, existem limitações no raio de detecção de alvos/obstáculos, raio de comunicação e também na mobilidade dos robôs. Todos esses fatores devem ser considerados e devidamente tratados para que o enxame de robôs desempenhe suas funções de maneira satisfatória. Porém, fatores como esse não se aplicam necessariamente a enxames de partículas e por isso frequentemente são desconsiderados.

O objetivo deste trabalho é aplicar o algoritmo PSO para coordenar um enxame de robôs terrestres que tem como objetivo explorar um ambiente desconhecido a procura de alvos fixos, e comparar o desempenho desse enxame com três diferentes funções de avaliação. A primeira função de avaliação adota o *fitness* inversamente proporcional à distância Euclidiana entre a partícula e o alvo mais próximo. Nessa função, é necessário saber as posições dos alvos no ambiente e o objetivo do robô é reduzir a distância entre ele e o alvo mais próximo. A segunda função de avaliação

usa o *fitness* diretamente proporcional à distância Euclidiana entre a partícula e sua posição inicial. Nesse caso, elimina-se a necessidade de se utilizar informações sobre as posições dos alvos no cálculo do *fitness* e o robô tem o objetivo de se distanciar da sua posição inicial. Já na terceira função de avaliação, enquanto os sensores da partícula não detectam alvos, o *fitness* é inversamente proporcional à distância Euclidiana entre partícula e um falso alvo gerando em uma posição pseudoaleatória do ambiente. Esse comportamento faz com que o robô percorra caminhos aleatórios explorando o ambiente e também elimina a necessidade de se saber a localização dos alvos para poder calcular o seu *fitness*.

Além disso, como no PSO a interação do enxame ocorre por meio da comunicação entre as partículas, um protocolo simples de comunicação foi desenvolvido para esse propósito. Esse protocolo baseado em pacotes foi incorporado ao PSO e é utilizado para transmissão de informações referentes ao *lBest*. Quando um robô precisa atualizar seu *lBest*, ele envia um pacote de solicitação para seus vizinhos (robôs que estão dentro do seu raio de comunicação), e ao receber a requisição, os vizinhos enviam um pacote com seu *pBest* para o robô que fez a solicitação.

Para simular o enxame de robôs em um cenário semelhante a um ambiente do mundo real, a ferramenta V-REP foi utilizada. O V-REP é uma plataforma que permite o desenvolvimento e execução de simulações envolvendo robôs [16]. A ferramenta se destaca por oferecer modelos pré-configurados de robôs e equipamentos como sensores e atuadores. Além de dispor de motores de simulação de física para a execução de cálculos e simulação de elementos de maneira realística. No V-REP foi desenvolvido um cenário com obstáculos onde foram distribuídos alvos. O enxame foi dividido em dois grupos, inicializados em posições que são tratadas como acesso ao ambiente.

Também foram desenvolvidos *scripts* na linguagem Lua para implementar o PSO, automatizar as simulações e exportar os dados das simulações em um arquivo para posterior geração de gráficos. O V-REP também foi utilizado para modelar uma interface gráfica usada para exibir informações dos parâmetros e outros elementos das simulações. Para compor o enxame de robôs, o robô K-Junior foi selecionado por

ter um modelo pronto disponível no V-REP, pelas suas dimensões e por possuir sensores e atuadores que permitem que ele se desloque pelo ambiente, possa detectar alvos e obstáculos e também sensores que permitem comunicação com outros robôs.

Este trabalho foi organizado em 6 capítulos. No Capítulo 2, são tratados conceitos relativos a inteligência de enxames, PSO, protocolo de comunicação, robótica de enxame e a ferramenta utilizada para modelagem e execução das simulações. O Capítulo 3, lista trabalhos relacionados. Em seguida, no Capítulo 4, é feita uma descrição dos experimentos e das métricas de avaliação do desempenho. Seguido do Capítulo 5, que apresenta e analisa os resultados obtidos. Por fim, no Capítulo 6, são apresentadas as conclusões deste trabalho e são listadas propostas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo, são abordados os conceitos de inteligência de enxame, robótica de enxame, protocolos de comunicação e ferramenta de simulação V-REP. Esses conceitos embasaram este trabalho e que por isso, são necessários para a melhor compreensão dele.

1.1 Inteligência de Enxames

A Inteligência de Enxame (*Swarm Intelligence* - SI) é um ramo da Inteligência computacional que estuda a inteligência que emerge do comportamento social dos indivíduos em sistemas naturais ou artificiais.

Para a Inteligência de Enxame, o enxame é um conjunto de indivíduos (ou agentes) simples e relativamente homogêneos que interagem entre si e com o ambiente, executando ações simples para alcançar um objetivo [1].

Um dos algoritmos da inteligência de enxames é o Algoritmo de Otimização por Enxames de Partículas. Apresentado em 1995 por James Kennedy e Russell Eberhart, o PSO foi descoberto através da simulação de um modelo social simplificado inspirado em comportamentos sociais como bando de pássaros e cardumes de peixes [15]. Desde 1995, várias versões do algoritmo surgiram, entretanto, todas elas baseiam-se na ideia inicial de um enxame de partículas que interagem entre si a procura de um ótimo global.

No PSO cada particular representa uma solução possível e elas são inicializadas de forma aleatória. A Figura 1 apresenta o fluxograma do PSO.

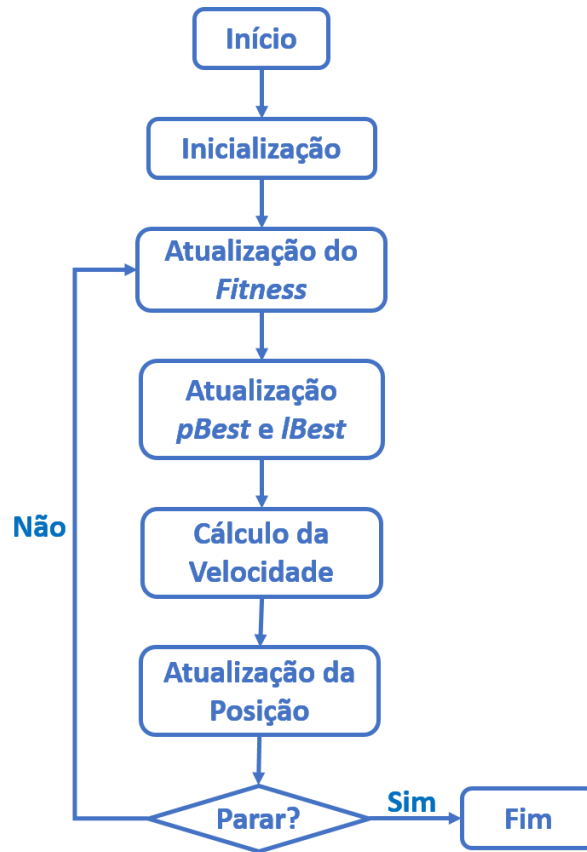


Figura 1. Fluxograma do PSO

Na Figura 1, $pBest$ e $lBest$ representam, respectivamente, o melhor resultado obtido por uma determinada partícula e o melhor resultado entre todas as partículas do enxame ou de um subgrupo de partículas de um enxame dependendo da topologia de comunicação adotada. Exemplos de topologias são a local, global e focal. Durante a fase de inicialização, as partículas são inicializadas em uma localização dentro do espaço de busca com uma determinada velocidade. Após a fase de inicialização, a partícula atualiza seu *fitness* que é uma indicação do quão perto a partícula está de atingir o seu objetivo. Para atualizar o seu $pBest$, a partícula verifica e salva a informação da melhor posição já ocupada por ela. Para atualizar o $lBest$ a partícula verifica qual o melhor $pBest$ entre os seus vizinhos, ou seja, a melhor posição já ocupada por um de seus vizinhos. Em seguida, a partícula atualiza sua velocidade e posição e verifica se o critério de parada foi atingido. Esse critério pode ser um número de iterações (ciclos de execução do algoritmo) ou quando um valor de *fitness* é atingido.

O Cálculo de atualização da velocidade e da posição são dados por

$$v_{i,d}(t + 1) = \omega v_{i,d}(t) + r_1 c_1 (pBest_{i,d} - x_{i,d}(t)) + r_2 c_2 (lBest_{i,d} - x_{i,d}(t)) \quad (1)$$

e

$$x_{i,d}(t + 1) = x_{i,d}(t) + v_{i,d}(t) \quad (2)$$

Sendo $v_{i,d}$ a componente da velocidade da partícula ‘ i ’ na dimensão ‘ d ’, $x_{i,d}$ é a posição atual da i -ésima partícula na dimensão ‘ d ’, c_1 é chamado de coeficiente cognitivo e representa o peso que o “ $pBest$ ” tem no cálculo da velocidade da partícula, c_2 chamado de coeficiente social e determina o peso que o “ $gBest$ ” (ou “ $lBest$ ”) tem no cálculo da velocidade. As variáveis r_1 e r_2 representam valores aleatórios entre 0 e 1 seguindo uma distribuição normal. O ω representa o peso da inércia, utilizado para controlar movimentos de busca em largura e profundidade [17]. A Tabela 1 ilustra o valor dos parâmetros utilizados no PSO.

Tabela 1. Parâmetros utilizados no PSO

Parâmetro	Símbolo	Valor
Coeficiente Cognitivo	c_1	2,05
Coeficiente Social	c_2	2,05
Peso da Inércia	ω	0,8
Valores Aleatórios	r_1 e r_2	[0,1]

Para a realização da fase de atualização do $pBest$ e do $lBest$, foi necessário desenvolver um simples protocolo de comunicação, descrito na seção 1.3, que fosse permitisse que um robô enviasse requisições dessa informação a seus vizinhos e recebesse as respostas deles.

Outra modificação feita no PSO foi a inclusão de um módulo de detecção e desvio de obstáculos. Esse módulo atua na fase de atualização da velocidade,

verificando se os sensores do robô detectam algum obstáculo (paredes ou outros robôs), em caso positivo a direção da partícula é gerada uma força em uma direção livre de obstáculos. Além disso, para garantir que os robôs não colidam com os obstáculos, foi determinado que a componente linear da velocidade do robô é anulada quando ele chega a uma distância de 4 cm de um obstáculo. Esse valor representa a distância mínima que o robô necessita para poder parar sem risco de colidir com o obstáculo.

1.2 Robótica de Enxames

Robótica de Enxames é o estudo de como coordenar grandes grupos de robôs utilizando uma inteligência de enxames [1]. Em outras palavras, a robótica de enxames consiste em aplicar os conceitos e princípios da inteligência de enxame para a coordenação de sistemas multi-robôs (*Multi-robot systems - MRS*).

Inicialmente, o foco da robótica de enxames foi estudar e validar pesquisas envolvendo o comportamento de coletivo em grupos de indivíduos, principalmente insetos, na natureza [12]. Porém, recentemente o foco mudou para o desenvolvimento de sistema multi-robôs bio-inspirados, capazes de solucionar problemas e executar tarefas [18].

Marco Dorigo, criador do ACO, e Erol Sahin são considerados os fundadores da Robótica de Enxames devido aos seus trabalhos com enxames de robôs [19,20]. Dorigo e Sahin estabeleceram uma lista de característica que diferencia a robótica de enxames dos outros sistemas multi-robôs:

- **Autonomia:** o enxame é composto por robôs que possuem autonomia e capacidade de fisicamente interagir e modificar o ambiente no qual estão inseridos.
- **Grande População:** o enxame de ser constituído de um número limitado de grupos homogêneos de robôs. Cada grupo é composto por um número elevado de robôs.

- Capacidades Limitadas: os robôs são relativamente incapazes e ineficientes quando executam tarefas isoladamente, contudo se tornam altamente eficientes quando cooperam.
- Escalável e robusto: ER devem ter seu desempenho melhorado quando se adiciona novos robôs, porém a remoção de robôs de um enxame não deve provocar o colapso dele.
- Coordenação distribuída: em ER a coordenação dos robôs deve ser descentralizada. Um robô possui uma percepção local de sua vizinhança, controle limitado e capacidade de comunicação.

Tabela 2. Comparação entre SR e MRS

Característica	Robótica de Enxame	Sistema Multi-robôs
Tamanho da população	Grande	Pequeno
Controle	Descentralizado e Homogêneo	Centralizado/Remoto
Homogeneidade	Homogêneo	Heterogêneo
Escalabilidade	Altamente escalável	Baixa Escalabilidade
Flexibilidade	Alta	Baixa
Ambiente	Desconhecido	Conhecido/Desconhecido

Como pode ser visto na tabela Tabela 2, as características dos Sistemas Multi-robôs tornam essa solução mais indicada para problemas que requerem o trabalho em conjunto de robôs que possuem diferentes funcionalidades, como, por exemplo, uma linha de montagem de produtos. Já a Robótica de Enxame tem grande potencial para ser aplicado nas seguintes atividades:

- Atividades em grandes áreas: devido a sua escalabilidade e autonomia, enxames de robôs podem ser utilizado para realização de atividades em

espaços físicos de grandes dimensões. Exemplo: busca, monitoramento, resgate e mapeamento.

- Atividades que ofereçam riscos aos robôs e a humanos: robustez e autonomia conferem a SR a capacidade de operação em atividades ariscadas como: localização e desarmamento de artefatos explosivos, resgate de sobreviventes em caso de incêndio, etc. Além disso, os robôs utilizados são mais simples, de fácil substituição e possuem um preço de produção inferior aos dos robôs atualmente utilizados para realização dessas tarefas.
- Tarefas que exijam grande população e redundância: Escalabilidade e robustez, permitem que enxames de robôs sejam utilizados para executar tarefas de difícil estimativa da demanda de recurso e onde o sistema deve continuar operando mesmo diante de perda de elementos. Exemplos são a contenção e limpeza de manchas de óleo em caso de vazamento em alto-mar, patrulha e mapeamento de regiões.

Para evitar que os robôs colidissem contra obstáculos, o robô dispõe de sensores infravermelhos distribuídos longitudinalmente ao longo de seu chassi a equação do cálculo da velocidade (Equação 1) foi atualizada para

$$v_{i,d}(t + 1) = !hasObstacles[(\omega v_{i,d}(t) + r_1 c_1 (pBest_{i,d} - x_{i,d}(t)) + r_2 c_2 (lBest_{i,d} - x_{i,d}(t)))] + hasObstacles(moveFreeLoc) \quad (3)$$

incluindo o módulo de detecção e ultrapassagem de obstáculos. Na Equação 3 o termo *hasObstacles* representa uma função que retorna '0' quando não foram detectados obstáculos e '1' caso contrário. *moveFreeLoc* representa uma função inspirada em algoritmos seguidores de paredes que entra em ação para contornar o obstáculo. Essa alteração pode ser vista como uma nova força que atua quando obstáculos são detectados e faz com que os robôs desviem dos obstáculos.

1.3 Protocolo de Comunicação

Tendo em vista que a capacidade de interação entre os robôs do enxame é algo essencial, se fez necessário o desenvolvimento de um protocolo que cumprisse esse papel.

No processo de transmissão e recepção de dados, o protocolo de comunicação é o elemento responsável por especificar o formato de dados e as regras a serem seguidas [21]. Ele determina todo o processo que deve ser feito antes de se enviar um dado, assim como todo o processamento necessário para a extração das informações após o recebimento dos dados.

Um protocolo de comunicação deve gerenciar basicamente os seguintes itens:

- Formato de dados: Como os dados são codificados, armazenados e transmitidos.
- Formato de endereço: como os endereços são codificados, armazenados e transmitidos.
- Roteamento: forma de se fazer com que os dados sejam entregues ao endereço de destino.
- Detecção de falhas de transmissão: o protocolo deve ser capaz de identificar situações de falhas durante a transmissão de dados e possuir uma estratégia de ação para esses casos.
- Controle de sequência e fluxo de dados: Necessários para evitar que informações fiquem presas na fila de envio, e que o emissor transmita em frequências que o receptor não consegue acompanhar, causando perda de dados.

Como o foco desse trabalho não é o desenvolvimento de um protocolo de comunicação, foi adotado um protocolo de comunicação incluso no V-REP. Esse protocolo faz todo o gerenciamento de roteamento, controle de sequência e fluxo das informações, e assume que não há falhas ou erros durante a transmissão.

Tomando o protocolo base oferecido pelo V-REP, foi determinado que as informações devem ser transmitidas através de mensagens. Mensagens são pacotes

compostos por três valores: remetente, destinatário e informação, como pode ser visto na Figura 2.

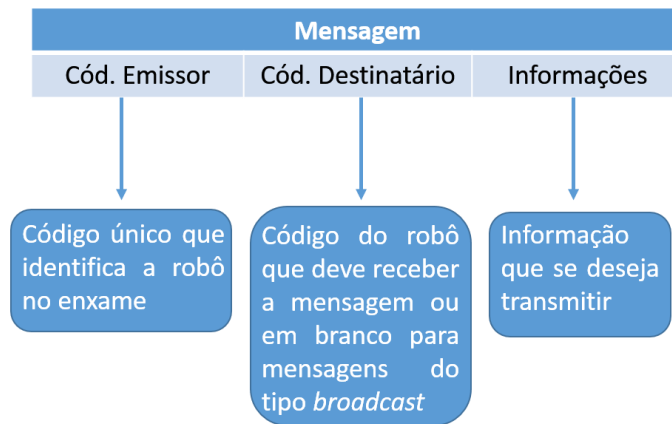


Figura 2. Estrutura de uma mensagem

Quando uma robô deseja saber vizinhos *fitness*, durante a fase de atualização do *pBest* e *lBest*, ele envia uma mensagem com a seguinte configuração: o remetente é o seu identificador, o campo receptor em branco para que todos os vizinhos (robôs no raio de alcance de comunicação) recebam a mensagem, e a conteúdo da mensagem é a palavra "fit". Após o envio, o robô aguarda 4 segundos por respostas e então o algoritmo prossegue atualizando o *lBest* com o melhor valor recebido.

Também durante a fase de atualização do *pBest* e *lBest*, o robô monitora o recebimento de mensagens sem destinatário, e caso o conteúdo da mensagem seja a palavra "fit", ele monta uma mensagem de resposta onde o emissor é o seu código, o receptor é o código do robô que enviou a mensagem, e o conteúdo da mensagem é seu *fitness*. A Figura 3 ilustra o fluxograma do PSO após a inclusão do protocolo de comunicação.

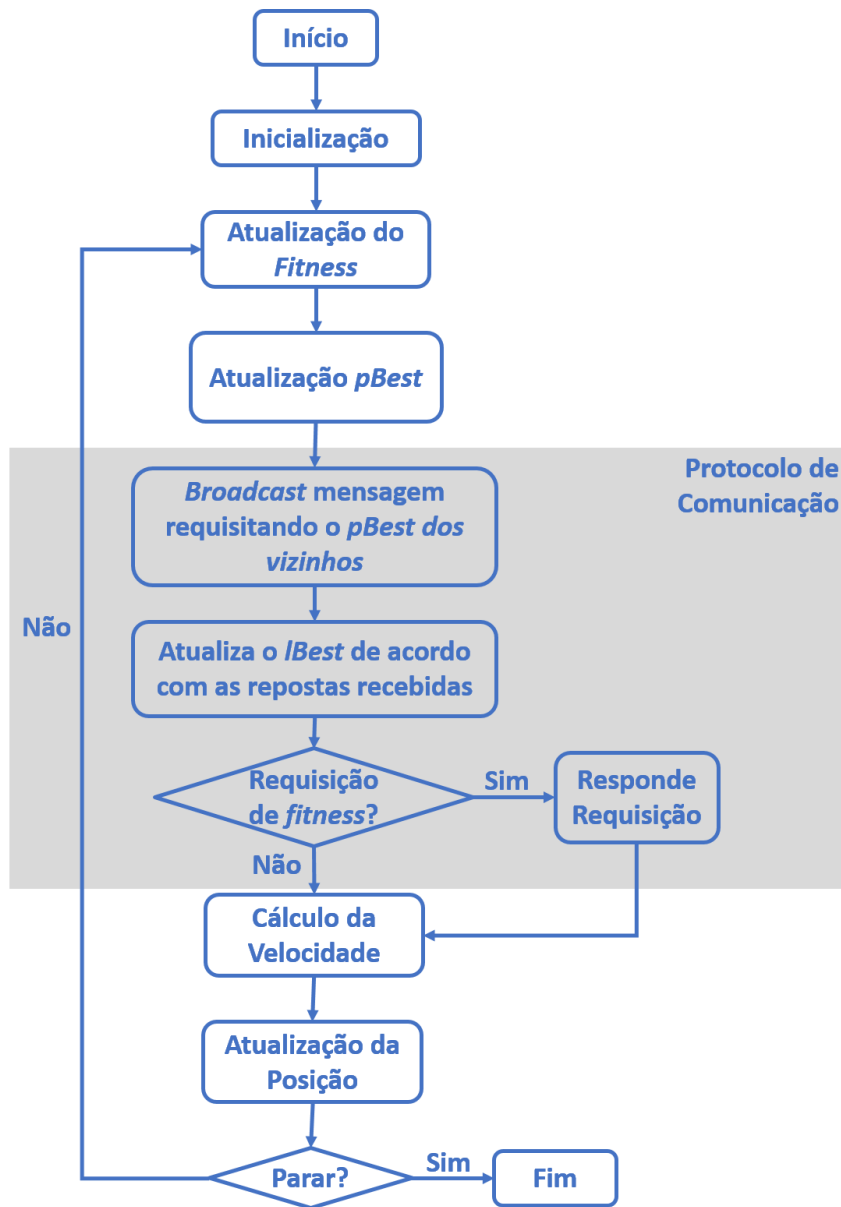


Figura 3. Fluxograma do PSO com Protocolo de Comunicação

1.4 V-REP

O V-REP é uma plataforma para criar, compor e simular cenários para simulações com robôs. A Coppelia Robotics, responsável pela plataforma, disponibiliza quatro versões da plataforma [16]:

- V-REP Pro Edu: gratuita e sem limitações. Disponibilizada para fins educacionais.

- V-REP Pro Eval: Versão de avaliação da ferramenta paga. Não é possível salvar as simulações ou modelos criados com essa versão.
- V-REP Pro: Versão destinada para fins comerciais.
- V-REP player: Ferramenta gratuita e de livre distribuição. Possui funcionalidades limitadas, sendo utilizada principalmente para a execução de simulações feitas em outras versões do V-REP.

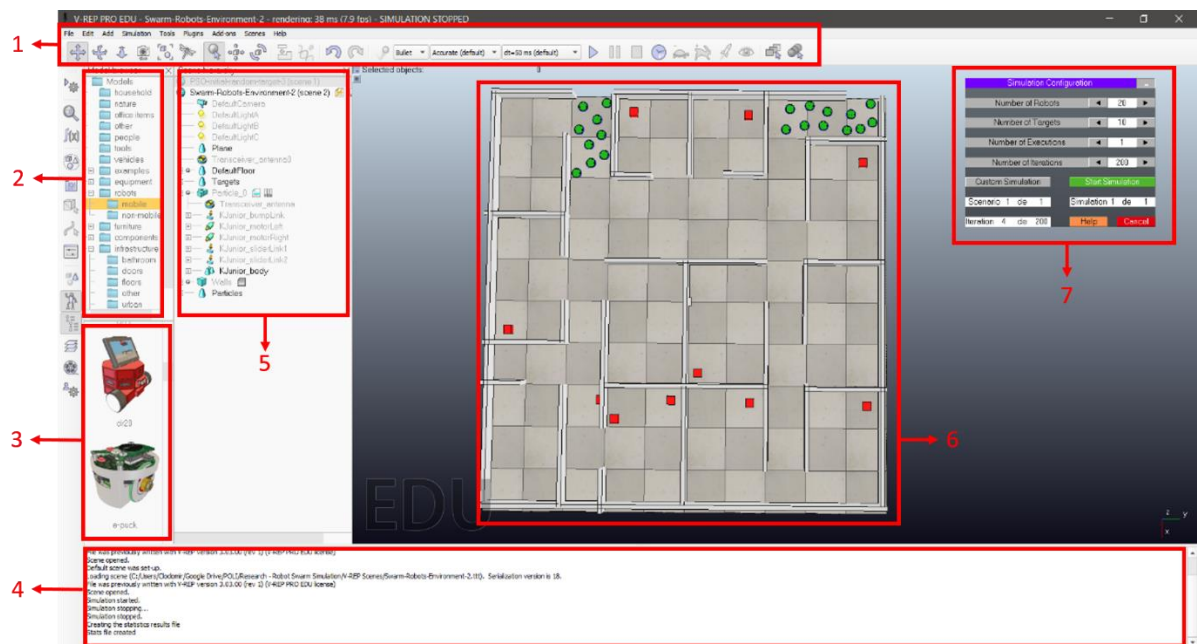


Figura 4. Interface do V-REP Pro Edu

Na Figura 4 é possível ver a barra de ferramentas do simulador (1) que oferece funcionalidades para criação, edição e visualização. O item 2 destaca as bibliotecas de elementos que podem ser utilizadas em uma simulação, como paredes, robôs (item 3), portas, janelas, etc. A ferramenta dispõem de um console (item 4), que é utilizado para acompanhamento da execução dos *scripts*. O item 6 mostra um cenário desenvolvido utilizando o V-REP e o item 7 mostra uma interface gráfica, também desenvolvida no V-REP e utilizada para controlar parâmetros das simulações.

Possui uma API (Application Programming Interface) pré-instalada que fornece ferramentas que simplificam os processos de criação, controle e execução de simulação tais como gravação de vídeo, *scripts* de controle personalizáveis e modelos de robôs que podem ser modificados para atender às necessidades do usuário [22].

A plataforma ainda conta com um ambiente de desenvolvimento integrado baseado em uma arquitetura de controle distribuído: cada objeto/modelo pode ser controlado individualmente através de um script incorporado, um *plug-in* ou API remota. Essas características tornam o V-REP uma plataforma extremamente versátil.

Os *scripts* de controle podem ser escritos em C, C ++, Python, Java, Lua, MATLAB, Octave ou Urbi, essa variedade de linguagens de programação suportadas pela ferramenta facilita e simplifica a integração com outros sistemas [23].

Ao lado dos elementos de simulação mencionados anteriormente, o simulador também suporta a simulação de sensores, transmissores, atuadores e objetos pré-definidos, como paredes, mesas, cadeiras, portas, janelas, etc. Além disso, o usuário pode definir as propriedades físicas de um elemento e especificar a interação entre os componentes e o ambiente.

As principais aplicações do V-REP estão relacionadas com o desenvolvimento rápido de algoritmos, simulações de automação industrial, prototipagem rápida, verificação de modelos, ensino de robótica, monitoramento remoto e segurança de duplo controle [23]. Por todas as razões listadas, V-REP foi a plataforma de simulação selecionada adotada.

Capítulo 3

Trabalhos Relacionados

Com o objetivo de entender e avaliar soluções relacionadas à coordenação de enxame de robôs aplicados à problemas de busca por alvos, foram analisados diversos trabalhos de temática semelhante. Os trabalhos foram agrupados de acordo com o algoritmo de enxame empregado à coordenação dos robôs. Essa divisão foi feita com objetivo de facilitar o entendimento das diferentes abordagens existentes para resolução de problemas de busca por alvos utilizando técnicas da inteligência enxames.

1.1 Algoritmo das Abelhas (Bees Algorithm - BA)

O *Bees Algorithm* foi apresentado em 2005 pelo time de pesquisa da *Cardiff University* no Reino Unido liderado por Afshin Ghanbarzadeh. O BA simula o comportamento de uma colônia de abelhas em busca por comida. Ele possui agentes simples, auto-organização, divisão do trabalho e algumas características específicas como exploração constante do espaço de busca, especialização dos agentes e recrutamento [24]. O algoritmo das abelhas serviu de base para a construção das soluções apresentadas na Subseção 1.1.1 e na Subseção 1.1.2.

1.1.1 Algoritmo de Abelhas Modificado (Modified Bees Algorithm – MBA)

O Algoritmo de Abelhas Modificado foi proposto em 2010 por Jevtic [25]. Esse algoritmo é uma versão adaptada do *Bees Algorithm* para enxame de robôs. O MBA simplifica o número de abelhas para apenas dois tipos: Abelhas disponíveis e abelhas indisponíveis. As abelhas disponíveis são aquelas que não encontraram fonte de alimento e ainda estão procurando. Abelhas indisponíveis são aquelas que encontraram uma fonte de alimento ou receberam a informação da localização de uma fonte de alimento e estão indo em direção a ela.

Para o MBA cada robô é uma abelha e inicialmente todos são abelhas disponíveis. Na fase inicial as abelhas executam uma busca aleatória por alvos, ao localizar um alvo a abelha se torna indisponível e recruta (informa a localização aproximada do alvo detectado) abelhas disponíveis para irem na direção do alvo. Ao receber a mensagem de recrutamento, a abelha verifica se há outras mensagens e vai na direção do alvo mais próximo.

1.1.2 Algoritmo das Abelhas + Módulo de Detecção e Desvio de Obstáculos

Em [26] o problema envolve a utilização de enxame de micro robôs que são responsáveis por efetuar busca por determinada célula onde devem depositar certo medicamento. Os robôs devem evitar áreas biologicamente restritas, que são áreas vistas como obstáculos que devem ser evitados para que não se provoque danos a células saudáveis.

A localização da célula alvo é conhecida e o BA é utilizado para gerar possíveis rotas que levem o micro robô à célula alvo. A detecção e desvio dos obstáculos é feita através da filtragem do conjunto de possíveis movimentos gerados pelo BA, excluindo trajetos onde há colisão ou entrada em áreas biologicamente restritas.

A solução desenvolvida obteve bons resultados, entretanto as simulações foram executadas desconsiderando forças que atuam nos robôs dentro da corrente sanguínea, e considerando que alvos e obstáculos são fixos.

1.2 Otimização por Colônia de Formigas (Ant Colony Optimization – ACO)

O Algoritmo de Otimização por Colônia de Formigas é um algoritmo distribuído no qual os agentes, as formigas, cooperam e se comunicam através do depósito de feromônio no ambiente [14]. Devido a suas características, o ACO é empregado principalmente na resolução de problemas de busca que possam ser modelados utilizando a estrutura de grafos.

Inspirado no comportamento das formigas que durante a busca por fontes de alimento, percorrem inicialmente um caminho aleatório e uma vez achado alimento, elas retornam ao formigueiro depositando feromônio no caminho. Quando outras formigas detectam o feromônio depositados, elas seguem a trilha até o alimento e, no caminho de volta para a colônia, reforçam a trilha de feromônio. Esse comportamento confere a colônia de formigas a habilidade de encontrar um caminho curto entre a fonte de alimento e a sua colônia.

O ACO adaptou esse comportamento mapeando as formigas para os agentes, o caminho entre a colônia e a fonte de alimento como uma solução, e a quantidade de feromônio no caminho representando a qualidade da solução encontrada. Uma solução que combina o ACO com o PSO é descrita na Subseção 1.2.1, enquanto que a Subseção 1.2.2 descreve uma solução que associa o ACO com uma técnica de Função de Potencial Artificial.

1.2.1 Colônia de Formigas + Enxame de Partículas

Um algoritmo híbrido de ACO + PSO proposto por Meng e Kazeem [27], aplica o conceito de comunicação através de depósito de feromônio como forma de comunicação entre os robôs do enxame.

As mensagens trocadas entre robôs ocorrem por meio de um “feromônio virtual”, que é descrito como um pacote, contendo informações sobre a localização de um certo alvo. Uma rede *ad hoc* sem fio é utilizada para a difusão dos pacotes de feromônio e, assim como os feromônio depositados por formigas, o feromônio virtual também evapora.

Ao detectar um alvo, o robô monta e transmite um pacote de feromônio, e quando um robô recebe um pacote desse tipo, ele inclui as informações da localização do alvo em um mapa individual que ele possui. Utilizando as informações da localização dos alvos detectados e a evaporação do feromônio virtual, o robô atualiza seu *lBest* com a localização do alvo que possui mais feromônio depositado.

A principal vantagem dessa estratégia é que o feromônio não depende do meio físico e por isso não é afetado por mudanças nele. Além disso, os pacotes são

transmitidos para uma determinada área do mapa, considerada a vizinhança do robô permitindo a comunicação local entre os membros do enxame.

1.2.2 Colônia de Formigas + Função de Potencial Artificial - APF

Gade e Joshi [28] apresentam uma solução que engloba a estrutura do ACO com os seguintes diferenciais:

- Situações em que o robô não tem informação da localização de alvos, ele adota um comportamento exploratório até que um alvo seja detectado ou que informações sobre a localização de alvos sejam recebidas de outro robô.
- Quando um robô localiza um alvo ele faz o depósito do feromônio, o que é na verdade uma mensagem transmitida via *gossiping* para outros robôs. *Gossiping* ou Protocolo de Comunicação por Boatos, é um protocolo onde um nó pode repassar uma informação para um número pequeno de outros nós. A principal vantagem desse protocolo, é evitar a sobrecarga da rede de comunicação com informações redundantes.
- Cada robô possui um mapa de alvos, montado com informações recebidas e descobertas pelos seus sensores. Os alvos recebem prioridades de acordo com a sua distância e confiabilidade da sua localização (quanto mais robôs relatarem a descoberta de um alvo 'x', maior confiabilidade das informações desse alvo).
- O APF (*Artificial Potential Function*) incorporado ao ACO funciona criando um mapa iterativo do ambiente onde obstáculos, outros robôs e áreas que oferecem risco, recebem um valor (ou potencial) positivo. Enquanto que os alvos recebem valor negativo. Dado esse mapa, são criadas forças que atraem o robô para áreas de potencial negativo, fazendo com que as soluções geradas pelo GA não passem por áreas de risco ou com obstáculos.

1.3 Otimização por Cultura de Bactéria (Bacterial Foraging Optimization – BFO)

A Otimização por Cultura de Bactéria foi introduzida em 2002 por Passino [29] e tem inspiração no comportamento social de coleta de alimentos observados na bactéria *Escherichia coli* (*E. coli*) dentro do intestino. O BFO possui quatro etapas:

- Quimiotaxia: habilidade que faz com que a *E. coli* tenha a tendência de migrar para regiões ricas em nutrientes. Essa etapa determina o movimento da bactéria.
- Movimentação em grupo: processo em que bactérias são atraídas para perto de outras. Quanto mais saldável for a bactéria, maior será a força de atração que ela exerce sobre as outras.
- Reprodução: Após vários ciclos de quimiotaxia, o enxame é separado em dois grupos de mesmo tamanho. As bactérias do grupo mais saldável são duplicadas na mesma região, enquanto que as bactérias do outro grupo morrem.
- Eliminação e dispersão: São eventos inesperados que eliminam parte da população de bactéria ou as espalha por regiões ainda não exploradas. Esse mecanismo previne que o enxame caia em mínimos locais.

A Subseção 1.3.1 descreve como o BFO foi modificado e aplicado à coordenação de enxame de robôs.

1.3.1 Algoritmo Modificado de Otimização por Cultura de Bactéria (Modified Bacterial Foraging Optimization – MBFO)

O Algoritmo Modificado de Otimização por Cultura de Bactéria foi proposto em 2014 por Bin Yang, Yongsheng Ding e Kuangrong Hao [30]. O MBFO trata cada robô como uma bactéria e adapta a quimiotaxia para fazer a busca por alvos e o planejamento de trajetórias.

Os alvos estão localizados em regiões de alta concentração de nutrientes, e a bactéria (robô) é atraída para essas regiões. Reprodução e eliminação são conceitos não aplicados aos robôs. Já a dispersão ocorre quando o robô está em uma área onde

ele não encontra alvos. Assim como no BFO, as bactérias são atraídas por outras mais saudáveis.

1.4 Otimização por Enxame de Partículas - PSO

Devido a sua simplicidade e eficiência em problemas de busca e otimização, algoritmos baseados no PSO tem sido bastante empregados em enxames de robôs. O PSO serviu de base para a criação do algoritmo VL-ALPSO descrito na Subseção 1.4.1 assim como para o DPSO, uma versão dinâmica do PSO detalhada na Subseção 1.4.2.

1.4.1 Otimização por Enxame de Partículas com Limitação de Velocidade e Método Lagrangiano Aumentado (VL-ALPSO)

Em seu artigo, Tang apresenta um algoritmo batizado de VL-ALPSO [31]. Esse algoritmo utiliza o PSO para fazer o planejamento da trajetória robôs, e para fazer evitar colisões e obstáculos ele utiliza o método de Lagrangiano Aumentado (AL) em associação com uma estratégia de redução da velocidade. O método de Lagrangiano Aumentado, ou método dos multiplicadores, é bastante empregado em problemas de minimização de funções sujeitas a restrições [32]. No VL-ALPSO, as posições dos obstáculos são tratadas como restrições sob a função objetivo.

A movimentação dos robôs é guiada pelo conjunto PSO+AL no qual os robôs são atraídos para as áreas que tiveram os melhores valores de *fitness* e repelido por áreas onde existem alvos ou outros robôs. Caso o robô chegue muito próximo de um obstáculo, entra em ação uma estratégia de limitação de velocidade. Essa estratégia consiste em parar o robô, fazer um giro de 360 graus e se mover para uma posição livre de obstáculos. Caso o obstáculo seja um outro robô, a estratégia é ficar parado até que o outro robô se distancie, nesse caso podem ocorrer situações em que os robôs fiquem parados esperando a movimentação do outro o que pode ser um problema.

Embora os resultados apresentados neste trabalho mostrem taxas de sucesso acima de 95%, vale salientar que o ambiente modelado é relativamente simples, sem áreas com um único acesso onde um robô pudesse ficar preso. A distância entre os

obstáculos grande, de modo que os robôs podem facilmente passar por entre os obstáculos. Além disso, considera-se que os robôs não sabem a posição exata do alvo, porém ele pode ser detectado em qualquer parte do ambiente, como se o alvo emitisse um sinal que pudesse ser captado em qualquer parte do ambiente e fica mais forte conforme o robô se aproxima do alvo.

1.4.2 Otimização por Enxame de Partículas Dinâmico (Dynamic Particle Swarm Optimization – DPSO)

Assim como na subseção anterior, o trabalho de Shoutao [33] utiliza uma combinação de algoritmos para executar a tarefa de busca. Nesse caso a combinação é do algoritmo de busca aleatória e uma versão do dinâmica do PSO (DPSO).

A ideia por trás do DPSO é permitir que uma partícula dinamicamente obtenha informações da partícula local, global ou mesmo de grupos de partículas. Essa mudança diminui a probabilidade de que o enxame convirja rapidamente para extremos locais, a principal causa disso no PSO tradicional, é a retardo no compartilhamento das informações de *gBest* e *lBest*.

A solução empregada nesse trabalho funciona em duas etapas, enquanto o robô não detecta sinais do alvo ele continua realizando uma busca aleatória que funciona como uma busca em largura. Quando sinais do alvo são detectados em uma região, entra em ação o DPSO, que executa uma busca em profundidade na região. Caso não sejam encontrados alvos após um certo período de tempo, o robô abandona a região e volta a fazer uma busca aleatória.

Os autores citam que a solução possui módulos de detecção e desvio de obstáculos, porém não dá detalhes sobre como eles foram implementados. Nas simulações descritas no artigo, os ambientes são livres de obstáculos que não sejam os outros robôs, por isso não é possível avaliar como essa solução se comportaria em cenários mais próximos ao mundo real.

Capítulo 4

Experimentos

Para que fosse possível a execução dos experimentos, a ferramenta V-REP foi utilizada para modelar um ambiente que pode ser visto na Figura 5. O modelo desenvolvido simula um ambiente de 5 metros de comprimento por 5 metros de largura, totalizando 25 metros quadrados. Paredes de 10 centímetros de espessura, 1 metro de altura e comprimento variável, foram introduzidas no ambiente servindo como obstáculos para o enxame. Os alvos são representados por cubos vermelhos com 10 centímetros de aresta e são inicializados sequencialmente do 1 ao 10 nas posições que podem ser vistas na Figura 5.

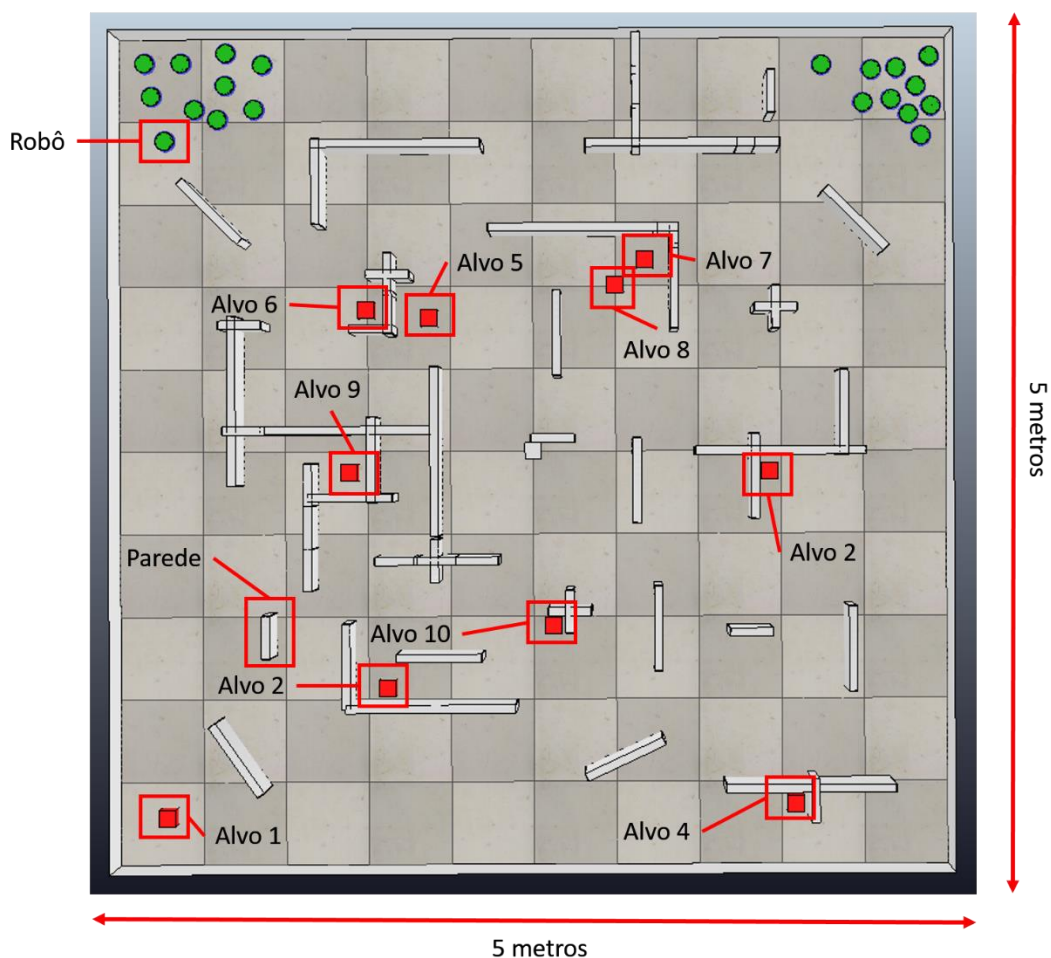


Figura 5. Ambiente e componentes criados no V-REP

O robô selecionado para compor o exame é um modelo do robô K-Junior V1 [34], um robô para fins educacionais desenvolvido pela *K-Team S.A* (Figura 6). Devido a sua simplicidade, tamanho, características e por existir um modelo desenvolvido pela fabricante para o V-REP, O K-Junior foi utilizado neste estudo. Na Tabela 3, encontram-se as especificações técnicas dele. Como pode ser visto na Figura 6, o robô possui um conjunto de sensores infravermelho em sua lateral. A distribuição dos sensores produz uma ótima área de detecção do alvos e obstáculos, permitindo que alvos e obstáculos sejam rapidamente detectados em posições ao redor do robô. O diâmetro de 125 milímetros permite que o K-Junior transite por áreas reduzidas e o conjunto formado pelas rodas diferenciais proporcionam uma maior liberdade de movimento, como rotação ao redor do seu eixo, curvas fechadas, etc.

Tabela 3. Especificações do K-Junior

Característica	Informações Técnicas
Locomoção	<ul style="list-style-type: none"> • Duas rodas diferenciais alimentadas por motores de corrente contínua. • Velocidade máxima de 0,16 m/s
Processador	<ul style="list-style-type: none"> • Microcontrolador PIC16F887 8MHz
Sensores Infravermelhos	<ul style="list-style-type: none"> • Seis sensores frontais e laterais de proximidade • Quatro sensores na parte inferior para seguir linhas e evitar quedas • Raio de detecção de até 30cm
Bluetooth	<ul style="list-style-type: none"> • Comunicação sem fio com um raio de até 20 metros
Dimensões	<ul style="list-style-type: none"> • Diâmetro de 125 milímetros • Altura de 40 milímetros

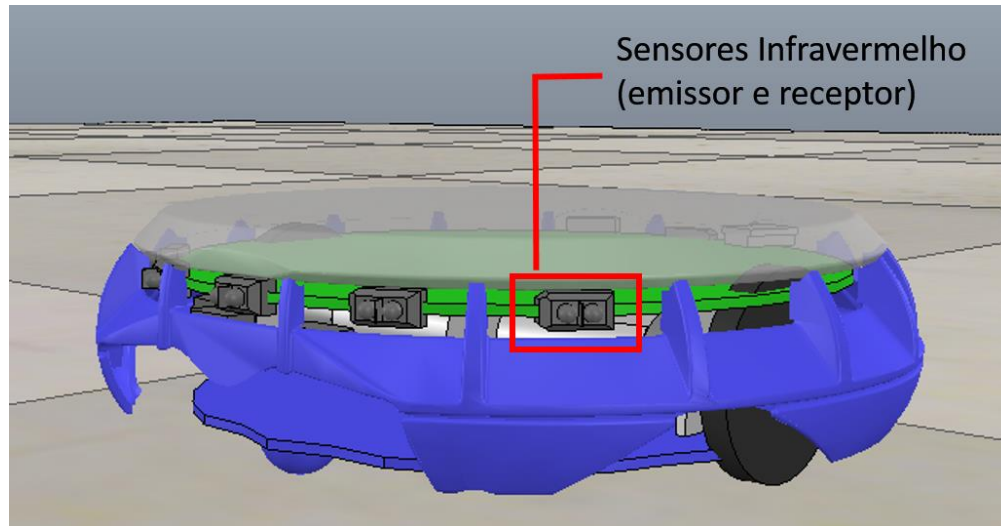


Figura 6. Robô K-Junior Utilizado nas Simulações

Como pode ser visto na Figura 5, os robôs são divididos em dois grupos, cada grupo com 10 robôs e eles são inicializados em posições próximas a lateral direita superior e esquerda superior. Essas posições são consideradas os pontos de acesso ao ambiente (portas). Optou-se por dividir os robôs dois grupos pois em simulações de teste, quando o enxame era inicializado em uma única posição, ocorriam pontos de congestionamento que comprometiam a capacidade de deslocamento dos robôs.

Os experimentos consistiram em executar 30 simulações para cada cenário de teste, onde cada cenário é uma combinação entre um número 'x' de robôs e um número 'y' de alvos, onde x pode ser 1, 5, 10, 15 ou 20 robôs e y pode ser 1, 2, 4, 6, 8 ou 10 alvos. Definiu-se o número máximo de alvos como sendo 10, para se obter uma melhor distribuição de alvos pelo ambiente, sendo esses inicializados nas posições para todas as execuções. Já o número máximo de robôs, foi determinado como 20 robôs para limitar o número de cenários, devido à restrição de tempo assim como o número de iterações. Os grupos de robôs eram formados seguindo a seguinte regra:

- Se o tamanho total do enxame é divisível por 2, então metade dos robôs são inicializados no grupo da direita e outra metade no grupo da esquerda;
- Caso contrário, o enxame é dividido de forma que o grupo da direita tenha 1 robô a mais do que o grupo da esquerda.

Para poder comparar os resultados, foram adotadas duas métricas de avaliação de desempenho do enxame. A primeira é a área de cobertura alcançada pelo enxame. Essa área é definida pela relação entre a área visitada pelos robôs, e a área total do ambiente. A segunda métrica é a taxa de sucesso do enxame. Essa taxa é obtida dividindo-se o número de alvos encontrados pelo número de alvos existentes. A Tabela 4 apresenta os valores utilizados nos experimentos.

Tabela 4. Parâmetros utilizados nas execuções

Parâmetro	Valor
Número de Robôs	1,5,10,15 ou 20
Número de Alvos	1,2,4,6,8 ou 10
Número de Iterações	1000
Número de Execução por Cenário	30
Raio de Comunicação dos Robôs	1 m
Alcance de Detecção de Alvos e Obstáculos	25 cm

O PSO foi configurado tomando como base os valores padrão, difundidos e frequentemente utilizado em trabalhos científicos envolvendo esse algoritmo. Esses valores foram descritos na Tabela 1.

Foram tomadas três funções de avaliação:

- Função 1: *fitness* inversamente proporcional à distância Euclidiana entre a partícula e o alvo mais próximo.
- Função 2: *fitness* diretamente proporcional à distância Euclidiana entre a partícula e sua posição inicial.
- Função 3: Enquanto os sensores da partícula não detectam alvos, o *fitness* é inversamente proporcional à distância Euclidiana entre partícula e um falso alvo gerando em uma posição aleatória do ambiente.

Capítulo 5

Resultados

A Figura 7, a Figura 8 e a Figura 9 ilustram os resultados de área de cobertura obtidos por cada função objetivo.

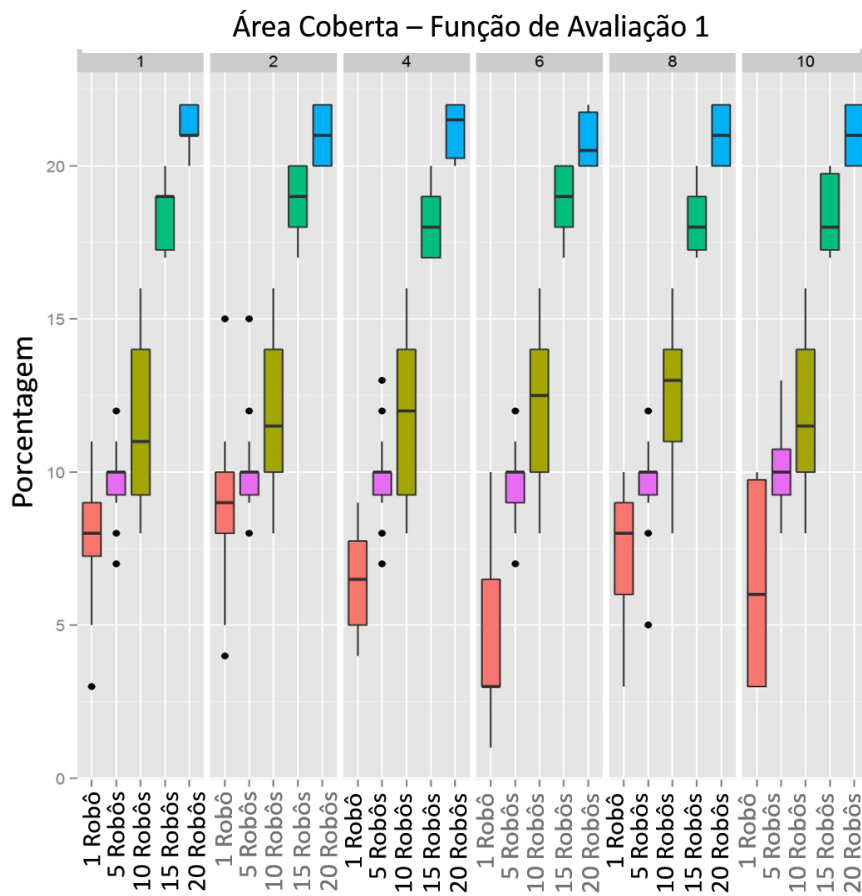


Figura 7. Área de Cobertura para Função de Avaliação 1

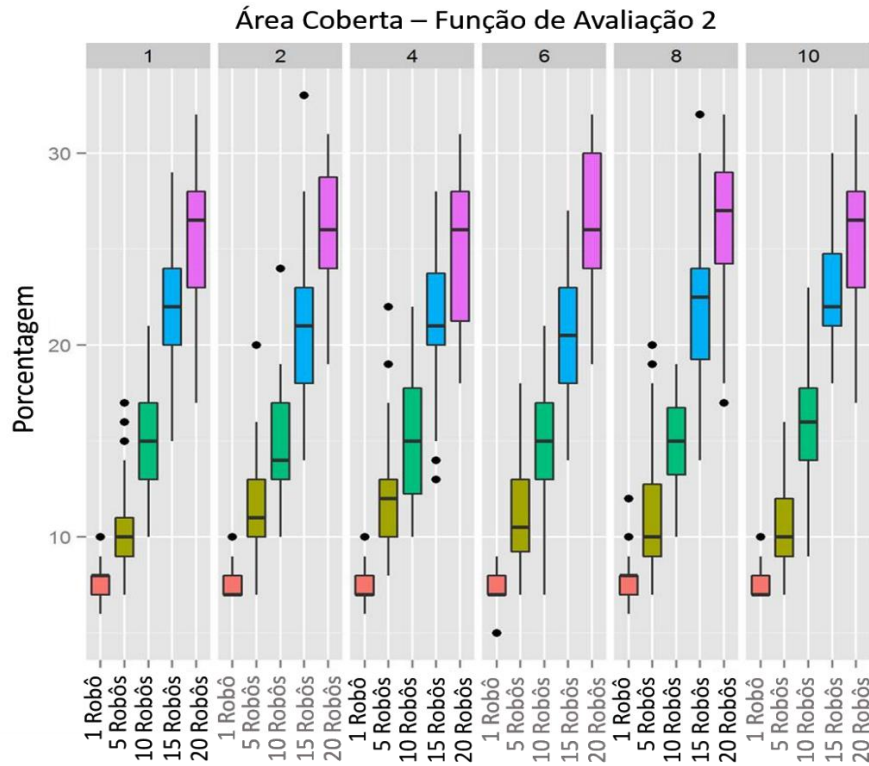


Figura 8. Área de Cobertura para Função de Avaliação 2

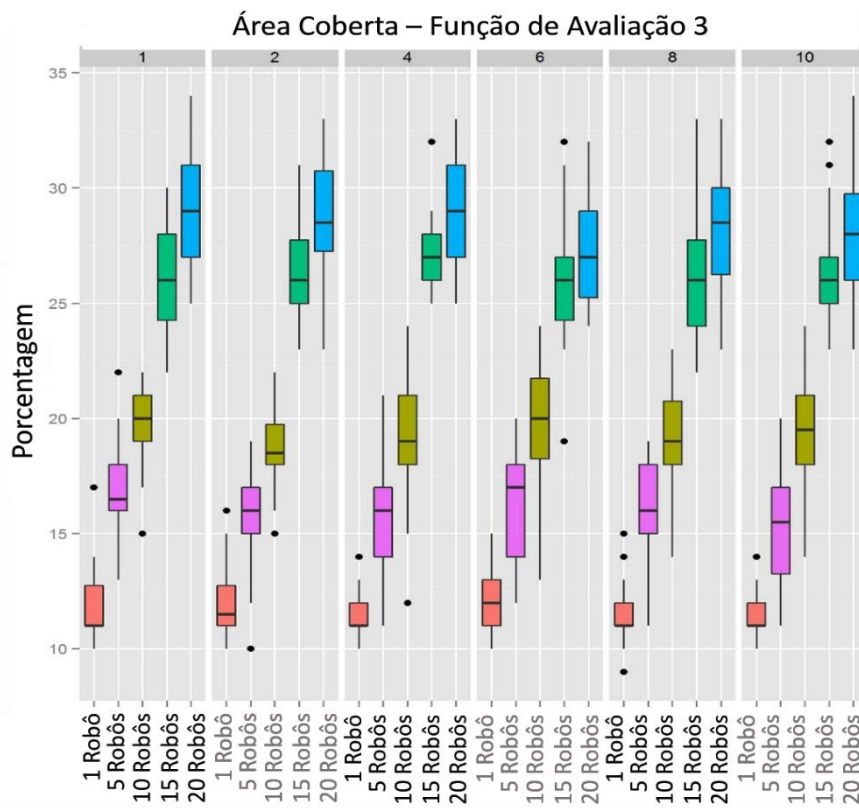


Figura 9. Área de Cobertura para Função de Avaliação 3

Os resultados apresentados pela Figura 7, Figura 8 e Figura 9 mostram que, assim como esperado, a área de cobertura aumenta conforme o número de robôs no enxame aumenta. Nota-se também que, dentre as funções de avaliação estudadas, a função 3 foi a que alcançou melhores resultados, ficando à frente das demais quase todos os cenários. A Função 1 obteve desempenho abaixo das demais funções, isso se deve em parte ao fato de que nessa função, o robô tem uma noção da localização dos alvos, e isso os permite traçar trajetetas menores que o leve aos alvos. Já nas outras estratégias, os robôs não têm essa indicação, por isso acabam explorando mais o ambiente em busca dos alvos.

A Figura 10, a Figura 11 e a Figura 12 apresentam os resultados relativos a taxa de sucesso do enxame.

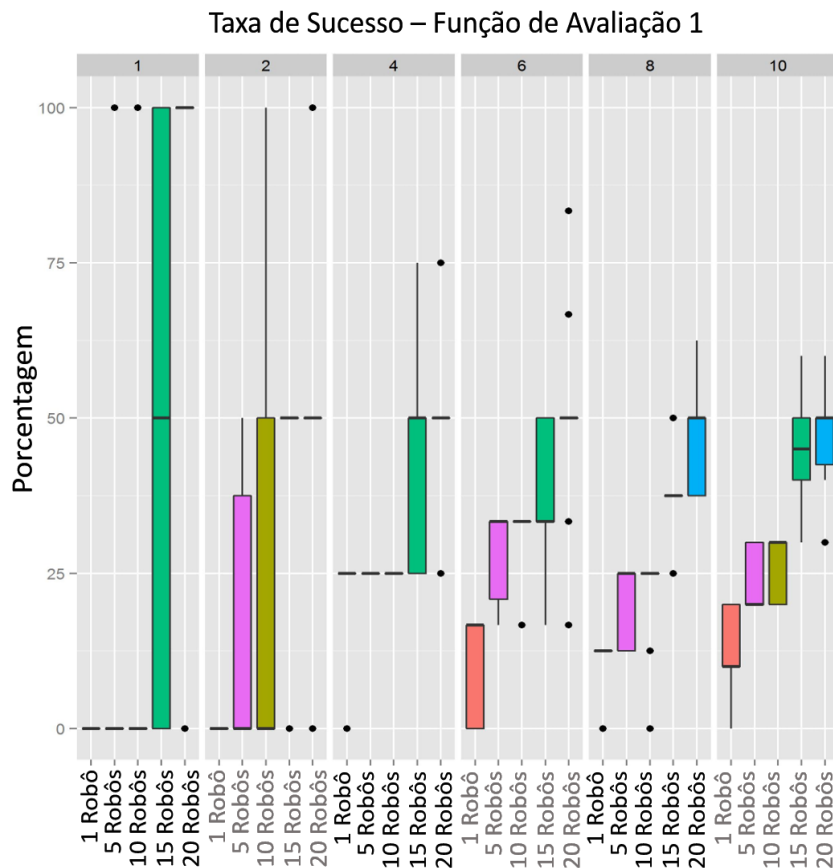


Figura 10. Taxa de Sucesso para Função de Avaliação 1

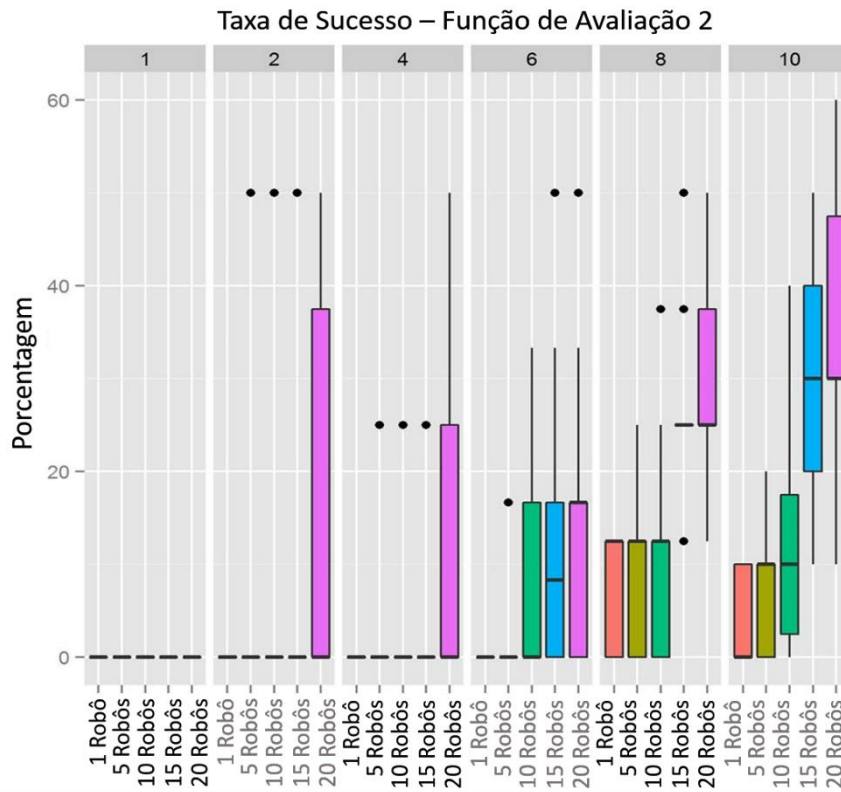


Figura 11. Taxa de Sucesso para Função de Avaliação 2

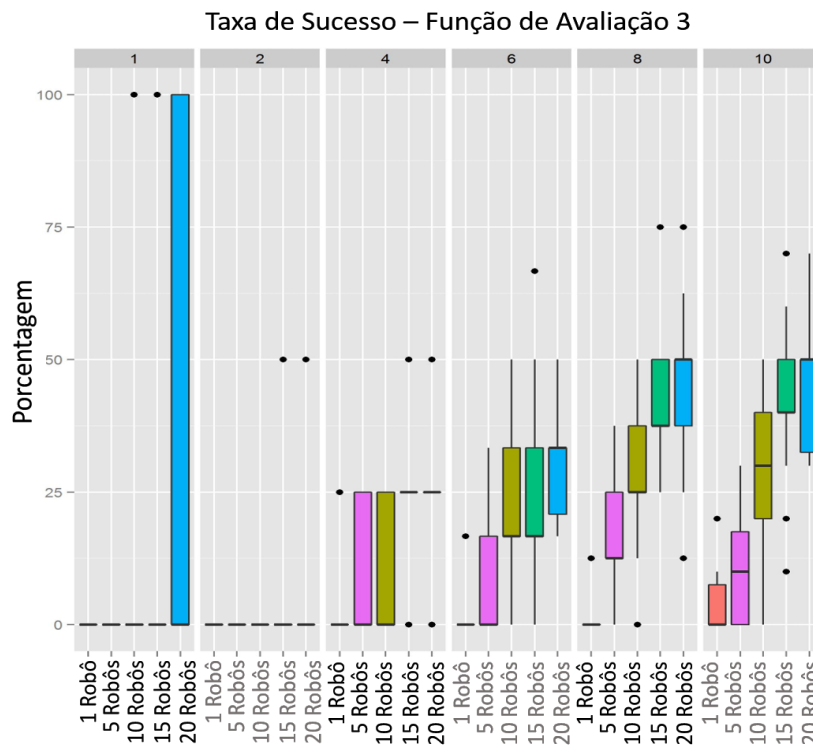


Figura 12. Taxa de Sucesso para Função de Avaliação 3

Observando os gráficos da taxa de sucesso, nota-se que a função de avaliação 1 teve em média resultados melhores, o que já era esperado já que os robôs têm noção da posição do alvo mais próximo através da distância Euclidiana e por isso necessitam de menos tempo para achar os alvos. No entanto, vale salientar que a função de avaliação 2 obteve resultados similares, e que, para cenário com mais de 4 alvos, a função 3 também alcançou resultados similares as demais funções. Isso pode ser explicado pelo fato de que nessa função, o robô é estimulado a fazer a exploração do ambiente e o enxame fica mais disperso o que aumenta as chances de localização de um maior número de alvos.

A baixa área de cobertura, em torno de 35% nos melhores casos, pode ser explicada pela restrição nos números de iterações. Durante os experimentos foi observado que mesmo após as 1000 (mil) iterações, o enxame não se encontrava estagnado. Para as três funções de avaliação, os robôs são programados para continuar buscando por novos alvos mesmo quando eles já encontraram algum, logo o enxame está em constante exploração do ambiente. No entanto, as simulações têm um alto custo computacional, devido principalmente a processamentos dos motores gráfico e de simulação física. Além disso, o tempo de simulação aumenta proporcionalmente com o número de robôs no enxame. Para a execução de mil simulação com mil iterações, o tempo necessário variava de 30 minutos para cenários com 1 robô e 3 horas para o cenário com 20 robôs. Tendo em vista que cada cenário foi executado 30 vezes, elevar o número de iterações acima 1000 não seria viável devido a limitação do tempo disponível para a realização deste trabalho. Por isso, acredita-se que área de cobertura poderia alcançar resultados ainda melhores, caso houvesse um número maior de iterações por cenário.

A taxa de sucesso também foi influenciada pela limitação no número de iterações e poderia ter alcançado resultado melhores caso o número de iterações fosse maior. Por outro lado, para a função de avaliação 1, notou-se que o enxame tendia a estar mais agrupado e vários robôs iam na direção do mesmo alvo o que também afetou o desempenho do enxame, já que os robôs descreviam trajetórias

similares no ambiente reduzindo a área de cobertura como pode ser visto na Figura 13.

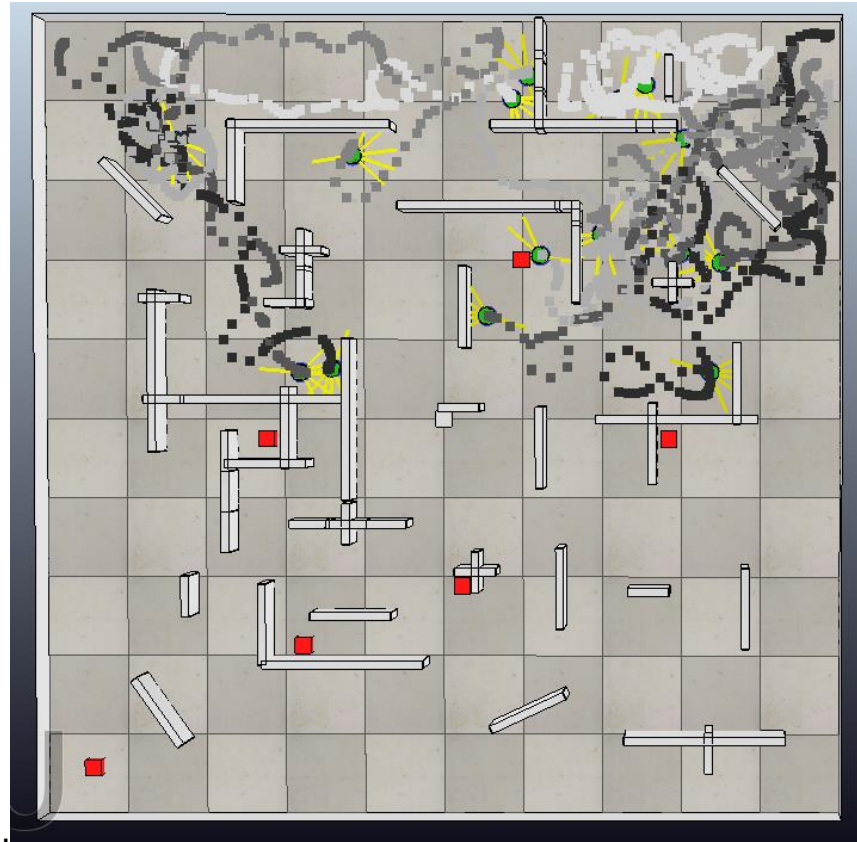


Figura 13. Trajetórias traçadas pelos robôs

A forma adotada para detectar e evitar obstáculos também influenciou o desempenho do enxame. Nesse caso, para grandes grupos de robôs em áreas reduzidas, para evitar colisões os robôs se moviam extremamente devagar e em alguns casos passava muito tempo parado calculando a uma forma de evitar as colisões. Nesses casos, várias iterações se passavam até que o enxame pudesse se mover de forma mais ágil.

Outro fator que também pode ter influenciado o desempenho do enxame, é que para as funções de avaliação 2 e 3, o enxame tem um caráter exploratório mais forte o que fez com que o enxame se dispersasse mais e, em alguns casos, levou robôs a perderem comunicação com o resto do enxame.

Capítulo 6

Conclusões

Analisando os resultados obtidos a partir da execução das simulações, foi possível observar que o número de robôs no enxame influencia o desempenho do enxame de modo que, um número muito baixo de robôs resulta em uma baixa área de cobertura e baixa quantidade de alvos detectados. Por outro lado, observou-se que enxames muito populosos apresentam problemas de mobilidade, formando pontos de congestionamento nas áreas de acesso ao ambiente e em áreas que simulam corredores. Nesses casos, poderia ser desenvolvida uma estratégia de coordenação que auxilie o enxame a se movimentar de forma mais ágil em ambientes com área reduzida, eliminando ou reduzindo congestionamentos.

Para o cenário utilizado neste trabalho, pode-se concluir de que um enxame de 15 robôs é suficiente para executar a busca de maneira eficiente. Esse número foi alcançado levando-se em consideração que não houve um ganho significativo em desempenho para enxames com 20 robôs. Além disso, devido à dificuldade que o enxame enfrenta em se deslocar em regiões como corredores, aumentar o número de robôs resultaria em mais congestionamento, dado que a estratégia de coordenação do enxame não tem um bom desempenho em ambientes congestionados.

No que diz respeito as funções objetivo comparadas, observou-se que, no cenário estudado, os robôs que através da função objetivo obtinha indicações da posição dos alvos, chegam ao alvo de forma mais rápida e com um trajeto menor do que os robôs que não recebem informações da localização dos alvos. Porém, os robôs do segundo caso alcançam uma cobertura maior do cenário, mantendo a sua taxa de sucesso similar as obtidas pelos robôs do outro caso. Vale salientar que esses resultados foram observados no cenário desenvolvido para as simulações e que, os mesmos resultados podem não ser obtidos em cenários reais. Isso se deve as simplificações que foram feitas no modelo, como por exemplo a ausência de desníveis no solo, a desconsideração da interferência das paredes no sinal de comunicação dos

robôs e a desconsideração de falhas no *hardware* dos robôs que poderiam resultar no mal funcionamento deles.

Os experimentos ainda permitiram observar situações em que um robô se distanciava demais do enxame, perdendo a capacidade de comunicação com os demais robôs. Em outros casos, a ausência de um protocolo mais avançado que permitisse a difusão de informações, resultou em robôs se concentrando em alvos que tinham sido descobertos previamente. Isso poderia ser resolvido com a utilização de um protocolo de comunicação mais robusto, que fosse capaz de identificar quando se perde a comunicação com o enxame e nesses casos cria uma força que atrai os robôs para uma posição em que ele ainda tenha comunicação com o enxame.

De maneira geral, os experimentos realizados foram bem-sucedidos, os resultados obtidos foram satisfatórios e as análises feitas permitiram verificar a viabilidade do emprego do PSO aplicado a coordenação de robôs realização de busca por alvos fixos em cenários desconhecidos e com a presença de vários obstáculos.

Como trabalhos futuros, sugere-se:

1. Novas funções objetivo e efetuar um estudo comparativo com as que foram utilizadas neste trabalho;
2. Desenvolver diferentes tipos de cenário que possibilitem retratar ambientes próximos aos encontrados no mundo real;
3. Verificar a viabilidade de utilização de outros algoritmos para a coordenação do enxame, como o ACO, ou mesmo produzir algoritmos híbridos resultantes de combinações de dois ou mais algoritmos de enxames;
4. Desenvolver um protocolo de comunicação mais robusto, visando estender as possibilidades de mensagens que podem ser trocadas pelos robôs e otimizando o processo de difusão de informações;
5. Verificar o desempenho do enxame para cenários com alvos móveis;
6. Propor novas métricas de medição do desempenho do enxame.

Referências Bibliográficas

9. ATAEIL, N.; ZIARATI, K.; EGHTEHAD, M. A BSO-Based Algorithm for Multi-robot and Multi-Target Search, p. 312-321, 2013.
28. BEHERA, ; SASIDHARAN ,. **Ant Colony Optimization for Co-operation in Robotic Swarms**. Advances in Applied Science Research. [S.l.]: [s.n.]. 2011. p. 476-482.
18. BENI,. **From swarm intelligence to swarm robotics**. SAB'04 Proceedings of the 2004 international conference on Swarm Robotics. Berlin: Springer. 2005. p. 1-9.
5. CASPER , J.; MURPHY , R. Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)** , v. 33, p. 367 - 385, Junho 2003.
16. COPPELIA ROBOTIC. coppeliarobotic. **V-REP Virtual Robot Experimentation Platform**. Disponível em: <<http://coppeliarobotics.com/>>. Acesso em: 21 Maio 2016.
14. DORIGO , ; STÜTZLE,. **Ant Colony Optimization**. Cambridge: The MIT Press , 2004.
19. DORIGO, et al. Evolving Self-Organizing Behaviors for a Swarm-Bot. **Autonomous Robots** , Hingham, Setembro 2004. 223-245.
20. DORIGO, M. SWARM-BOT: AN EXPERIMENT IN SWARM ROBOTICS, Bruxelas, 2005.
37. EBERHART, R.; SHI, Y. **Computational Intelligence**. Burlington: Elsevier Inc, 2007.

22. FREESE, et al. Virtual Robot Experimentation Platform V-REP: A Versatile 3D Robot Simulator. **Simulation, Modeling, and Programming for Autonomous Robots**, Darmstadt, Novembro 2010. 51-62.
7. GUO , Y.; BAO, ; SONG ,. Designed and implementation of a semi-autonomous search robot. **International Conference on Mechatronics and Automation** , Changchun, p. 4621 - 4626 , Agosto 2009.
35. HOLLAND, J. **Adaptation in Natural and Artificial Systems**. Cambridge: MIT Press , 1975. 211 p.
10. HOLLINGE, G.; SINGH, S.; DJUGASH, J. Efficient Multi-Robot Search for a Moving Target. **The International Journal of Robotics Research**, 2009.
6. JENNINGS , J. S.; WHELAN, G.; EVANS , W. Cooperative search and rescue with a team of mobile robots. **International Conference on Advanced Robotics**, Monterey, p. 193 - 200 , Julho 1997.
25. JEVTEC, A. et al. **Building a swarm of robotic bees**. World Automation Congress. Kobe: IEEE. 2010. p. 1-6.
24. KARABOGA,. AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION, Kayseri, utubro 2005.
15. KENNEDY , ; EBERHART,. Particle Swarm Optimization. **Proceedings of IEEE International Conference on Neural Networks**., Perth, 27 Novembro 1995. 1942–1948.
1. KHALDI, ; CHERIF ,. An Overview of Swarm Robotics: Swarm Intelligence Applied to Multi-robotics. **International Journal of Computer Applications** , Setembro 2015. 31-37.

33. LI, ; LI, ; ZHANG,. A Hybrid Search Algorithm for Swarm Robots Searching in an Unknown Environment. **Applied Mechanics and Materials**, Novembro 2014. 553-860.
17. MARINI, ; WALCZAKB,. Particle swarm optimization (PSO). A tutorial. **Chemometrics and Intelligent Laboratory Systems**, v. 149, p. 153-165, 4 Dezembro 2015.
2. MARJOVI, ; MARQUES, ; PENDERS, . **Guardians Robot Swarm Exploration and Firefighter Assistance**. Int. Conf. on Intelligent Robots and Systems (IROS). St Louis: [s.n.]. 2009.
27. MENG, ; KAZEEM, ; MULLER, . **A Hybrid ACO/PSO Control Algorithm for Distributed Swarm Robots**. 2007 IEEE Swarm Intelligence Symposium (SIS 2007). Honolulu: IEE. 2007.
12. NAVARRO, ; MATÍA,. An Introduction to Swarm Robotics. **ISRN Robotics**, v. 2013, p. 10, 2013.
29. PASSINO, K. Bacterial Foraging Optimization. **International Journal of Swarm Intelligence Research**, Janeiro 2010. 1-16.
26. PHAM, D. et al. **The Bees Algorithm – A Novel Tool for Complex Optimisation Problems**. 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2006). [S.l.]: [s.n.]. 2006. p. In Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2006) (2006), pp. 454-459 Key: citeulike:8377514.
34. ROBOTICS, K.-T. M. K-Team Mobile Robotics. **K-JJunior**. Disponível em: <<http://www.k-team.com/mobile-robotics-products/k-junior>>. Acesso em: 20 Junho 2016.

23. ROHMER, ; SINGH , ; FREESE,. **V-REP: a Versatile and Scalable Robot Simulation Framework**. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. Tokyo: IEEE. 7 Novembro 2013. p. 1321 - 1326.
32. SANT'ANA, C. Uma nova proposta utilizando métodos de lagrangeano aumentado com penalidades modernas na resolução de problemas de contato. **Universidade Federal do Paraná**, Curitiba, 2005.
3. SCHENKER, P. et al. **Robotic automation for space: planetary surface exploration, terrain-adaptive mobility, and multi-robot cooperative tasks**. Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision. Boston: [s.n.]. 2001.
36. SHEHATA , ; SCHLATTMANN ,. **Non-Dominated Sorting Genetic Algorithm for Smooth Path Planning in Unknown Environments**. IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). Espinho: IEEE. 2014. p. 14-21.
21. TAKIZAWA , ; MITA ,. Secure group communication protocol for distributed systems. **Computer Software and Applications Conference**, Phoenix, 5 Novembro 1993. 159 - 165.
13. TAN, ; ZHENG, Z.-Y. Research Advance in Swarm Robotics. **Defence Technology**, 9, n. 1, 2 Março 2013. 18-39.
31. TANG , ; EBERHARD ,. A PSO-based algorithm designed for a swarm of mobile robots. **Structural and Multidisciplinary Optimization** , New York Outubro 2011. 483-498.

11. TANG, Q.; EBERHARD, P. A PSO-based Algorithm Designed for a Swarm of Mobile Robots. **Stuttgart Research Centre for Simulation Technology (SRC SimTech)**, 2011.
8. UNLUTURK, ; AYDOGDU , O. Design and implementation of a mobile robot used in bomb research and setup disposal. **International Conference on Electronics, Computers and Artificial Intelligence (ECAI)**, p. 1-5, 2013.
30. YANG, ; DING, ; HAO,. Target searching and trapping for swarm robots with modified bacterial foraging optimization algorithm. **11th World Congress on Intelligent Control and Automation Shenyang**, Shenyang, 26 Junho 2014. 1348-1353.
4. ZHENG , X.-Z. et al. Development of human-machine interface in disaster-purposed search robot systems that serve as surrogates for human. **IEEE International Conference on Robotics and Automation**, v. 1, p. 225-230, 2004.