



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

PATRÍCIA DE SOUSA PAULA

**UTILIZAÇÃO DE INTELIGÊNCIA DE ENXAME EM UAVS PARA BUSCA DE ALVO
FIXO DE LOCALIZAÇÃO DESCONHECIDA**

FORTALEZA

2019

PATRÍCIA DE SOUSA PAULA

UTILIZAÇÃO DE INTELIGÊNCIA DE ENXAME EM UAVS PARA BUSCA DE ALVO
FIXO DE LOCALIZAÇÃO DESCONHECIDA

Dissertação apresentada ao Curso de do
Programa de Pós-Graduação em Computação
do Centro de Ciências da Universidade Federal
do Ceará, como requisito parcial à obtenção do
título de mestre em Ciência da Computação.
Área de Concentração: Sistemas de Informação

Orientador: Prof. Dr. Miguel Franklin
de Castro

Coorientador: Prof. Dr. Gabriel Antoine
Louis Paillard

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

P348u Paula, Patricia de Sousa.

Utilização de Inteligência de Enxame em UAVs para Busca de Alvo Fixo de Localização Desconhecida /
Patricia de Sousa Paula. – 2019.
58 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação
em Ciência da Computação, Fortaleza, 2019.

Orientação: Prof. Dr. Miguel Franklin de Castro.

Coorientação: Prof. Dr. Gabriel Antoine Louis Paillard.

1. Inteligência de Enxame. 2. Unmanned Aerial Vehicle. 3. Algoritmos Bioinspirados. I. Título.

CDD 005

PATRÍCIA DE SOUSA PAULA

UTILIZAÇÃO DE INTELIGÊNCIA DE ENXAME EM UAVS PARA BUSCA DE ALVO
FIXO DE LOCALIZAÇÃO DESCONHECIDA

Dissertação apresentada ao Curso de do
Programa de Pós-Graduação em Computação
do Centro de Ciências da Universidade Federal
do Ceará, como requisito parcial à obtenção do
título de mestre em Ciência da Computação.
Área de Concentração: Sistemas de Informação

Aprovada em: 28 de novembro de 2019

BANCA EXAMINADORA

Prof. Dr. Miguel Franklin de Castro (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Gabriel Antoine Louis Paillard (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Arthur de Castro Callado
Universidade Federal do Ceará (UFC)

A minha mãe, cujo exemplo sempre me fez seguir em frente. A meu marido, Wellington, cujo apoio durante o mestrado foi fundamental para conseguir vê-lo finalizado. E a meu filho, Matheus, cujo amor será sempre minha força.

AGRADECIMENTOS

Aos professores Miguel Franklin e Gabriel Paillard, cujo apoio e orientação tornaram este trabalho possível. Suas contribuições, ponderações e paciência foram inestimáveis.

Ao bom amigo Paulo de Tarso, por providenciar que eu tivesse um computador para as simulações.

Agradeço a minha família, Matheus e Wellington Sarmiento, por se manterem ao meu lado durante essa jornada.

Por fim, agradeço a minha mãe, Marilene, que sempre me encorajou.

“Toda a nossa ciência, comparada com a realidade, é primitiva e infantil – e, no entanto, é a coisa mais preciosa que temos.”

(Albert Einstein)

RESUMO

O contexto desta pesquisa é o uso de algoritmos bioinspirados aplicados a veículos aéreos não tripulados (UAV) para busca de um alvo fixo, de localização desconhecida. Um alvo pode ser um ser humano perdido ou um veículo quebrado, por exemplo. Mostra-se que algoritmos de enxame usados com UAVs podem ser adaptados para apresentar melhor desempenho do que um algoritmo de varredura simples, como o *Parallel Path Finder*. Comparam-se os algoritmos *Particle Swarm Optimization* e *Bat Algorithm*, usando restrições como tempo de vida das baterias dos UAVs e o tamanho da área de busca. Assim, é mostrada a melhor solução para este problema, dentre as adaptadas, considerando as restrições aplicadas.

Palavras-chave: Inteligência de Enxame. UAV. Algoritmos Bioinspirados.

ABSTRACT

The context of this research is the use of bioinspired algorithms applied to unmanned aerial vehicles (UAV) to search for a fixed target of unknown location. A target can be a lost human being or a broken vehicle, for example. Swarm algorithms used with UAVs can be adapted to perform better than a simple scanning algorithm such as Parallel Path Finder. The Particle Swarm Optimization and Bat Algorithm algorithms are compared using constraints such as UAV battery life and the size of the search area. Thus, the best solution to this problem is shown, among the adapted ones, considering the applied restrictions.

Keywords: Swarm intelligence. UAV. Bioinspired algorithms.

LISTA DE FIGURAS

Figura 1 – UAVs atuando em conjunto para busca	15
Figura 2 – Imagem de um UAV real	18
Figura 3 – Exemplo de nós em uma rede formada por UAVs	19
Figura 4 – WSN com nós sensores (<i>motés</i>) e <i>gateway</i>	20
Figura 5 – <i>Autonomous Control Level</i> (ACL) e tendência da autonomia em UAVs	24
Figura 6 – Ilustração sobre experimento de coleta de alimento entre formigas e escolha do menor caminho	25
Figura 7 – Comportamento das partículas no PSO	38
Figura 8 – Comportamento no BA	39
Figura 9 – Funcionamento da varredura em coluna utilizada	40
Figura 10 – Visualização da câmera	42
Figura 11 – Percentual de sucesso do cenário 1	43
Figura 12 – Percentual de sucesso do cenário 2	43
Figura 13 – Percentual de sucesso do cenário 3	44
Figura 14 – Percentual de sucesso do cenário 4	44
Figura 15 – Percentual de sucesso do cenário 4 incluindo o BA original	44
Figura 16 – Percentual de sucesso do cenário 5	45
Figura 17 – Percentual de sucesso do cenário 6	45
Figura 18 – Percentual de sucesso por tamanho da grid	46
Figura 19 – Percentual de sucesso por quantidade de UAVs	46
Figura 20 – Distância média percorrida por tamanho da grid	48
Figura 21 – Distância média percorrida por quantidade de UAVs	48
Figura 22 – Tempo calculado para o cenário 1	49
Figura 23 – Tempo calculado para o cenário 2	49
Figura 24 – Tempo calculado para o cenário 3	50
Figura 25 – Tempo calculado para o cenário 4	50
Figura 26 – Tempo calculado para o cenário 5	50
Figura 27 – Tempo calculado para o cenário 6	50

LISTA DE TABELAS

Tabela 1 – Categorização de fenômenos biológicos e algoritmos de rede que imitam esses conceitos	25
Tabela 2 – Levantamento bibliográfico	32
Tabela 3 – DJI Phantom 4 Pro	35
Tabela 4 – Configuração do computador utilizado para simulação	41
Tabela 5 – Percentual de sucesso cenário 1	43
Tabela 6 – Percentual de sucesso cenário 2	43
Tabela 7 – Percentual de sucesso cenário 3	43
Tabela 8 – Percentual de sucesso cenário 4	43
Tabela 9 – Percentual de sucesso do cenário 5	45
Tabela 10 – Percentual de sucesso do cenário 6	45
Tabela 11 – Margem de erro do cenário 4	46
Tabela 12 – Margem de erro do cenário 5	46
Tabela 13 – Margem de erro do cenário 6	46
Tabela 14 – Distância média percorrida no cenário 1	47
Tabela 15 – Distância média percorrida no cenário 2	47
Tabela 16 – Distância média percorrida no cenário 3	47
Tabela 17 – Distância média percorrida no cenário 4	47
Tabela 18 – Distância média percorrida no cenário 5	47
Tabela 19 – Distância média percorrida no cenário 6	47
Tabela 20 – Velocidade média no cenário 1	48
Tabela 21 – Velocidade média no cenário 2	48
Tabela 22 – Velocidade média no cenário 3	48
Tabela 23 – Velocidade média no cenário 4	48
Tabela 24 – Velocidade média no cenário 5	49
Tabela 25 – Velocidade média no cenário 6	49
Tabela 26 – Tempo calculado para o cenário 1	49
Tabela 27 – Tempo calculado para o cenário 2	49
Tabela 28 – Tempo calculado para o cenário 3	50
Tabela 29 – Tempo calculado para o cenário 4	50
Tabela 30 – Tempo calculado para o cenário 5	51

Tabela 31 – Tempo calculado para o cenário 6	51
--	----

LISTA DE ABREVIATURAS E SIGLAS

ABC	<i>Artificial Bee Colony</i>
ACO	<i>Ant Colony Optimization</i>
APSO	<i>Accelerated PSO</i>
AS	<i>Ant System Algorithm</i>
BA	<i>Bat Algorithm</i>
BC	<i>Bee Colony</i>
FA	<i>Firefly Algorithm</i>
GA	<i>Genetic Algorithm</i>
MANET	<i>Mobile Ad hoc Network</i>
MAV	<i>Micro Aerial Vehicle</i>
PP	<i>Parallel Path</i>
PSO	<i>Particle Swarm Optimization</i>
QoS	Qualidade de Serviço
RSSF	Redes de Sensores Sem Fio
UAV	<i>Unmanned Aerial Vehicle</i>
UGV	<i>Unmanned Ground Vehicle</i>
VANT	Veículo Aéreo Não Tripulado
WSA	<i>Wolf Search Algorithm</i>
WSN	<i>Wireless Sensor Network</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Visão Geral	15
1.2	Pergunta Motivadora	16
1.3	Hipótese	16
1.4	Objetivos	16
1.4.1	<i>Geral</i>	17
1.4.2	<i>Específicos</i>	17
1.4.3	<i>Contribuição Acadêmica</i>	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	<i>Unmanned Aerial Vehicle (UAV)</i>	18
2.2	<i>Wireless Sensor Network (WSN)</i>	20
2.3	Computação Autônoma	22
2.4	Computação Bioinspirada e Comportamento de Enxame	24
2.4.1	<i>Algoritmo de otimização baseado em colônia de formigas (no inglês, Ant Colony Optimization)</i>	27
2.4.1.1	<i>Ant System (AS)</i>	27
2.4.2	<i>Algoritmo de otimização baseado em enxame de partículas (no inglês, Particle Swarm Optimization) - PSO</i>	28
2.4.3	<i>Algoritmo de Morcegos (no inglês, Bat Algorithm - BA)</i>	29
2.4.4	<i>Outros algoritmos bioinspirados</i>	30
2.4.4.1	<i>Algoritmo baseado em colônia de abelhas (no inglês, Bee Colony - BC)</i>	30
2.4.4.2	<i>Algoritmo de busca dos lobos (no inglês, Wolf Search Algorithm) - WSA</i>	30
3	TRABALHOS RELACIONADOS	32
4	PROPOSTA	35
5	LEVANTAMENTO E ANÁLISE DE DADOS	40
5.1	Especificações de hardware e software utilizados para a simulação	40
5.2	Cenários utilizados para simulação	41
5.3	Análise de resultados	42
6	CONCLUSÕES E TRABALHOS FUTUROS	52
6.1	Trabalhos futuros	53

REFERÊNCIAS	54
APÊNDICE A – <i>IEEE EATIS 2016</i>	58

1 INTRODUÇÃO

Neste capítulo será apresentado um resumo do contexto e das necessidades que motivaram o desenvolvimento do presente trabalho de pesquisa, justificando sua criação e apresentando seus objetivos, tanto principal quanto específicos. Por fim, será apresentada a estrutura geral do texto com um rápido resumo sobre cada um dos assuntos abordados em suas seções.

1.1 Visão Geral

O avanço do uso de veículos autônomos para aplicações fora de laboratórios de pesquisa vem aumentando consideravelmente nos últimos anos. Assim, os veículos autônomos estão se inserindo aos poucos em nosso dia a dia, sendo chamados *Unmanned Ground Vehicle* (UGV), *Unmanned Aerial Vehicle* (UAV) e *Micro Aerial Vehicle* (MAV), dependendo do meio de movimentação e tamanho. Os UGVs, UAVs e MAVs tem relação com a computação autônoma¹, podendo ser parcial ou completamente autônomos, de acordo com as características levantadas no referencial teórico a ser visto no capítulo 2.

Figura 1 – UAVs atuando em conjunto para busca



Fonte: adaptado de <http://cabbuavphd.blogspot.com.br/2012/11/project-progress-search-patterns-and.html>.

Aparece como desafio a localização de alvos sem posição conhecida, tais como pessoas desaparecidas ou objetos perdidos. Já é corrente o uso de *drones* para localização de alvos em terrenos difíceis, mas estes são guiados por uma pessoa, o que pode ser um empecilho

¹ Área da computação que visa o desenvolvimento de sistemas computacionais complexos capazes de autogerenciamento e adaptação a mudanças não previstas (RIBEIRO; CASTRO, 2010)

ao fazer uma varredura em áreas extensas, como 100 hectares ou mais. Os UAVs podem facilitar este processo, possuindo autonomia e, assim, podendo ser utilizados em grupos. Estes UAVs, juntamente com algoritmos de busca e reconhecimento permitiriam a localização de forma eficiente e eficaz. No entanto, qual seria o melhor algoritmo ou abordagem algorítmica para se otimizar esta buscas? E qual seria a tecnologia de rede para que se possa melhor comunicar os nós UAVs?

Um comportamento bioinspirado bastante utilizado com UAVs, principalmente para buscas, é o comportamento de enxame, como veremos no capítulo 2. Quanto à tecnologia de redes adequada, pode-se utilizar a abordagem de Redes de Sensores sem Fio (em inglês, WSN).

É neste contexto que se encontra o presente trabalho, o qual propõe uma solução bioinspirada para a distribuição e comportamento de UAVs com WSN para a identificação de pessoas ou veículos em ambientes abertos e/ou de difícil acesso (e.g. florestas, rede de cavernas e montanhas).

1.2 Pergunta Motivadora

Tem-se a seguinte questão como geradora do trabalho:

É possível utilizar UAVs com comportamento bioinspirado, respeitando certas restrições reais dos UAVs, para solucionar um problema de busca e identificação de um alvo de posição desconhecida em uma dada área?

1.3 Hipótese

Como se trata de um problema de busca de um alvo cuja posição é desconhecida, a utilização de UAVs com algoritmos bioinspirados e comportamento de enxame para explorar o ambiente e localizar o alvo pode apresentar uma solução viável.

1.4 Objetivos

Os objetivos que guiam o presente trabalho estão divididos em geral e específicos, descritos a seguir.

1.4.1 Geral

Implementar algoritmos bioinspirados simulando sua aplicação em um grupo de UAVs para verificar o comportamento destes no processo de busca de um alvo fixo de localização desconhecida, em uma dada área.

1.4.2 Específicos

Temos como objetivos específicos:

- (a) Comparar algoritmos bioinspirados para a busca do alvo realizada por UAVs;
- (b) Verificar a possibilidade de alterar um algoritmo bioinspirado para um melhor resultado da busca e localização;
- (c) Analisar dados, colhidos a partir das simulações, que levem a uma solução viável para a melhoria do uso de UAVs para sensoriamento e busca em ambientes abertos ou de difícil acesso, para validação da solução proposta.

1.4.3 Contribuição Acadêmica

Este trabalho propiciou a publicação de um artigo no EATIS 2016, contemplando a primeira parte do desenvolvimento. Este artigo foi citado por quatro outros trabalhos, conforme pode ser visto na comunidade digital *Research Gate*².

Por fim, este trabalho está dividido em seções. A primeira seção sendo a Introdução, a qual se encerra nesse ponto, seguida pela fundamentação teórica, trabalhos relacionados, proposta, análise de resultados e, por último, a conclusão e as referências bibliográficas.

² https://www.researchgate.net/publication/305649652_A_swarm_solution_for_a_cooperative_and_self-organized_team_of_UAVs_to_search_targets/citations

2 FUNDAMENTAÇÃO TEÓRICA

Para o presente trabalho foi feito um levantamento bibliográfico acerca dos temas relacionados na literatura.

A seguir, são colocadas algumas definições e tecnologias que contextualizam esta pesquisa.

2.1 *Unmanned Aerial Vehicle (UAV)*

Os UAVs (ver figura 2) – ou, no português, Veículo Aéreo Não Tripulado (VANT) –, são aeronaves sem a presença de pilotos que podem ser controladas remotamente ou voar de forma autônoma de acordo com planos de voo pré-programados (CRUZ *et al.*, 2008; CHEN *et al.*, 2009). Suas principais formas de controle são via rádio remoto ou pela pré-programação do voo.

Figura 2 – Imagem de um UAV real



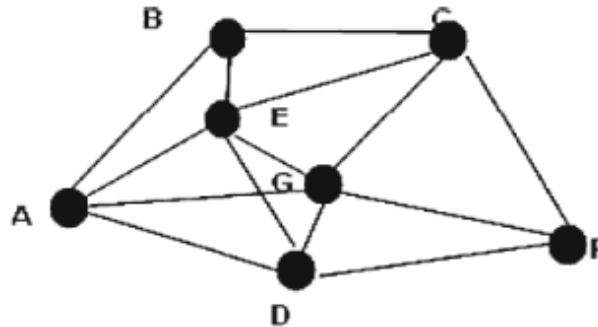
Fonte: <http://www.dji.com/br/phantom-4-pro-v2>.

Sua versão em tamanho reduzido é chamada MAV. A versão terrestre é chamada UGV, que normalmente assume a forma de carros ou robôs com deslocamento terrestre. Os UAVs levam vantagem sobre os UGVs no que diz respeito ao terreno, que pode ser íngreme e/ou com muitos obstáculos, e no maior alcance de área, lembrando que missões de maior duração vão requerer que os UAVs retornem periodicamente para reabastecimento (NIGAM *et al.*, 2012).

Os UAVs têm capacidade de sensoriamento limitada e a cooperação entre um grupo de UAVs depende fortemente da comunicação entre os UAVs próximos. Como exemplo, pode-se ver na figura 3 que os nós A e F não possuem um canal direto de comunicação. Para que a informação seja enviada, deve-se seguir um caminho intermediário de comunicação, por

exemplo, $A > B > C > F$. No caso dos nós UAVs estarem acoplados a nós de uma rede WSN, essa comunicação pode ser feita pela rede WSN, utilizando protocolo de comunicação próprio desse tipo de rede. Um maior detalhamento desse ponto será visto na seção 2.2, sobre WSN.

Figura 3 – Exemplo de nós em uma rede formada por UAVs



Fonte: (SATHYARAJ *et al.*, 2008)

Podem-se utilizar metáforas biológicas e algoritmos bioinspirados para que um grupo de UAVs atue de forma cooperativa e com comportamento relacionado em prol de um objetivo comum. Um comportamento bioinspirado bastante utilizado com UAVs é o comportamento de enxame (MCCUNE; MADEY, 2013; WEI; WEI, 2009; LI *et al.*, 2006), principalmente no tocante a busca, reconhecimento e roteamento (NIGAM *et al.*, 2012). No comportamento de enxame, cada elemento possui um comportamento simples, individualmente, mas juntos são capazes de executar tarefas complexas.

Os enxames se comportam de forma descentralizada e com alta tolerância a falha, pois a falha de um elemento não compromete o todo. Por outro lado, temos elementos com comportamento relativamente simples que, juntos, são capazes de resolver problemas complexos (MCCUNE; MADEY, 2013). Assim, temos na colaboração e coordenação de grupos de UAVs as vantagens de completar missões em um período de tempo mais curto e poder cumprir várias metas simultaneamente. Além disso, pequenas aeronaves são de difícil detecção quando comparadas a um veículo grande, como um carro, por exemplo e, como foi colocado antes, se um UAV é danificado a missão continua sendo mantida pelos demais (SATHYARAJ *et al.*, 2008).

Segundo (WEI; WEI, 2009; BASILICO; CARPIN, 2012), um problema de busca com enxames de UAVs é exatamente como alocar e controlar de forma eficiente os UAVs para realizar buscas exaustivas com o menor custo energético.

No planejamento da sua distribuição em uma área, um UAV pode ser afetado por muitos fatores de risco, os quais devem ser considerados no planejamento. Dois fatores importan-

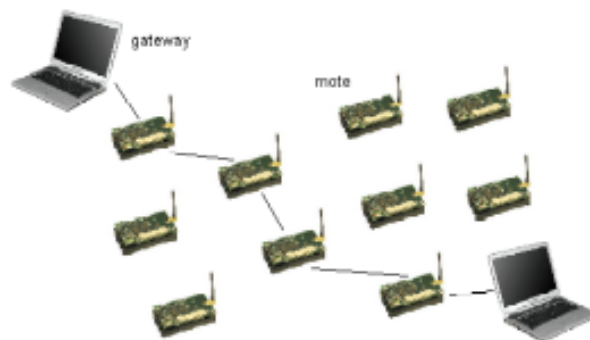
tes nesse sentido são a habilidade de resistência da aeronave, que inclui a capacidade de manobra do UAV, velocidade de voo e velocidade de voo em altitude, e as condições ambientais da área a ser varrida pelos UAVs, como visibilidade, pressão do ar e velocidade do vento. No caso de aplicações militares, incluem-se como fatores de risco os radares, mísseis e artilharia (LI *et al.*, 2006).

Temos inúmeros exemplos de uso de UAVs, tanto civis quanto militares, que envolvem reconhecimento de superfícies, monitoramento de sítios arqueológicos, assistência a desastres, vigilância de fronteira, monitoramento de linhas de energia entre outros (CHEN *et al.*, 2009).

2.2 *Wireless Sensor Network (WSN)*

Wireless Sensor Network (WSN) ou, no português, Redes de Sensores Sem Fio (RSSF), tem aplicações em várias áreas, como monitoramento ambiental, prédios inteligentes, prevenção de desastres, monitoramento de tráfego e vigilância militar (GUERRERO *et al.*, 2009). Uma WSN é uma rede (figura 4) normalmente com um grande número de nós sensores que podem coletar, armazenar e processar os dados coletados, assim como comunicar com nós próximos para prover um sistema de observação ambiental. Cada nó individualmente possui limitada velocidade de processamento, capacidade de armazenamento e largura de banda para comunicação (NAN; LI, 2008).

Figura 4 – WSN com nós sensores (*motes*) e *gateway*



Fonte: http://recolog.blogspot.com.br/2011_07_01_archive.html.

Segundo Sarmiento *et al.* (2012), as WSNs têm como principais elementos os nós sensores (*sensor nodes* ou *motes*) - compostos, normalmente, por um microcontrolador, um ou mais sensores, bateria e um *transceiver*, um ou mais nós sorvedouros (*sink nodes*) - responsáveis pela coleta de informações dos sensores e, em alguns casos, pela fusão ou agrupamento de

dados para serem enviados a um sistema monitor, e os equipamentos de interligação com outras redes, chamados *gateway*. Outro ponto importante é que as WSNs são orientadas a dados e sua topologia é voltada para permitir que os dados coletados pelos sensores cheguem aos *sink nodes* com o menor gasto energético.

Temos vários protocolos de comunicação sem-fio estabelecidos e que podem ser usados na comunicação das redes WSN, como *Bluetooth*, Wi-Fi (IEEE 802.11), UWB ¹, Zigbee, 6LoWPAN ², dentre outros (YANG, 2014).

O *Bluetooth* apresenta um alcance baixo e queda na comunicação (emparelhamento) e o Wi-Fi (IEEE 802.11) apresenta alto gasto de energia, sendo mais susceptível a interferência, o que os torna menos confiáveis para esse trabalho.

O Zigbee é um padrão baseado no IEEE 802.15.4, desenvolvido para atender as necessidades específicas da maioria das aplicações *wireless* de sensoriamento e controle. Possui baixo custo, baixo consumo de energia, baixa taxa de transmissão, alta confiabilidade, alta segurança em aplicações *wireless* de automação e controle remoto (CHENGBO *et al.*, 2009). Segundo (CHENGBO *et al.*, 2009), em comparação com *Bluetooth*, Wi-Fi, UWB e outros, o Zigbee se mostra com excelente transmissão em baixas taxas de transmissão.

Segundo (TOSCANO; BELLO, 2012), os protocolos ZibBee e 6LoWPAN apresentam vantagens aproximadas, tendo o Zigbee um menor delay na comunicação fim a fim. Outra diferença é que o 6LoWPAN está sempre na escuta de pacotes não dando suporte a *sleeping nodes*, isso acaba interferindo no gasto energético dos nós.

Por essas características, a tecnologia de comunicação considerada nesse trabalho será o ZigBee. O uso de rede WSN em conjunto com UAVs geralmente se dá com os *sensor nodes* fixos e os UAVs em trânsito coletando dados desses nós (BOUKERCHE *et al.*, 2013) (JAWHA *et al.*, 2013) (FAIÇAL *et al.*, 2014) (VILLAS *et al.*, 2013) (GUERRERO *et al.*, 2009).

Neste trabalho, a intenção é considerar o acoplamento dos *sensor nodes* da rede WSN aos nós do grupo de UAVs, juntando as capacidades de mobilidade dos UAVs ao sensoriamento e comunicação providos pela WSN. Algo similar foi proposto por Grado-Caffaro *et al.* (2011), através do acoplamento de uma rede WSN a *mobile robots* para sensoriamento pós desastres naturais. Outro trabalho propõe o uso de *sensor nodes* transportados por MAVs que podem coletar diversos tipos de informações ambientais (ALLRED *et al.*, 2007).

¹ UWB é um acrônimo para *UltraWideBand*. Usa tecnologia de rádio principalmente para comunicar aparelhos sem fio como teclado, mouse, impressora.

² IPv6over Lowpower Wireless PersonalArea Networks, utiliza IPv6 sobre o protocolo IEEE 802.15.4.

2.3 Computação Autônômica

Com o aumento da complexidade dos sistemas computacionais, com arquiteturas orientadas a serviços e compostas por elementos de *hardware* e *software* distribuídos geograficamente, busca-se transferir responsabilidades de gerenciamento para o próprio sistema.

Nesse cenário, novas características foram se tornando cada vez mais importantes, como a dinamicidade, para que as aplicações possam responder a necessidades variáveis de recursos da aplicação ao longo do tempo; complexidade, em grande parte associada a uma arquitetura de *software* relacionada a serviços; Qualidade de Serviço (QoS), envolvendo fatores como corretude e desempenho; emergência, que busca tratar comportamentos não antecipados, como falha de recursos e concorrência. Dessas necessidades, a IBM produziu um manifesto em 2001 onde propôs o conceito de Computação Autônômica, ou Autônoma, referente a sistemas computacionais capazes de autogerenciamento a partir de um conjunto de objetivos definidos (CORREA; CERQUEIRA, 2009).

Assim, temos que a computação autônômica objetiva o desenvolvimento de sistemas computacionais complexos capazes de autogerenciamento e adaptação a mudanças não previstas (RIBEIRO; CASTRO, 2010).

O autogerenciamento é o centro de um sistema autônômico e é caracterizado por quatro propriedades, chamadas *self-* properties*, que possibilitam esse autogerenciamento (RIBEIRO, 2009).

Essas propriedades são (IBM, 2005; RIBEIRO, 2009; CORREA; CERQUEIRA, 2009):

- (a) Autoconfiguração (*self-configuring*) - o sistema deve ser capaz de realizar as autoconfigurações necessárias ao seu funcionamento. Essas configurações podem ser necessárias devido a novas circunstâncias relacionadas ao seu funcionamento ou como apoio aos processos de autorreparação, auto-otimização ou autoproteção;
- (b) Auto-otimização (*self-optimizing*) - o sistema autônômico deve buscar otimizar seu próprio funcionamento, buscando melhorar e aprimorar seu desempenho. Em síntese, o próprio sistema deve buscar seu aprimoramento;
- (c) Autorreparação (*self-healing*) - o sistema deve ser capaz de identificar falhas, assim como suas causas, e traçar soluções, que devem ser aplicadas e testadas sem necessitar de intervenção externa. Requer a identificação do problema e seu reparo imediato, sem interrupção do serviço; e

(d) Autoproteção (*self-protecting*) - o sistema deve decidir quando e como se defender de falhas ou ataques maliciosos, assim como, antecipar possíveis falhas e se prevenir para elas antecipadamente.

Um ponto importante é que sistemas autogerenciáveis estão aptos a realizar ajustes apenas dentro de seu próprio escopo. Então, dado um gerenciamento autônomo de um servidor, por exemplo, apenas esse servidor pode ter seu desempenho otimizado.

Assim, temos que, sistemas de controle autônomos são projetados para funcionar bem sob incertezas significativas no sistema e ambiente por longos períodos de tempo, e eles devem ser capazes de compensar as falhas do sistema, sem intervenção externa.

Esses sistemas, muitas vezes, utilizam técnicas do campo da Inteligência Artificial (IA) para alcançar essa autonomia (CHEN *et al.*, 2009). Segundo Nigam *et al.* (2012), o interesse nas pesquisas para controle e coordenação de veículos autônomos aumentou nos campos da inteligência artificial, programação dinâmica, programação linear inteira. Técnicas tradicionais de IA, como A* (*A star*) e suas variantes, estariam sendo aplicadas, mas sem abranger cooperação entre múltiplos veículos. Da mesma forma, algoritmos de cobertura de *spanning tree* trabalham bem com robô único, mas não múltiplos.

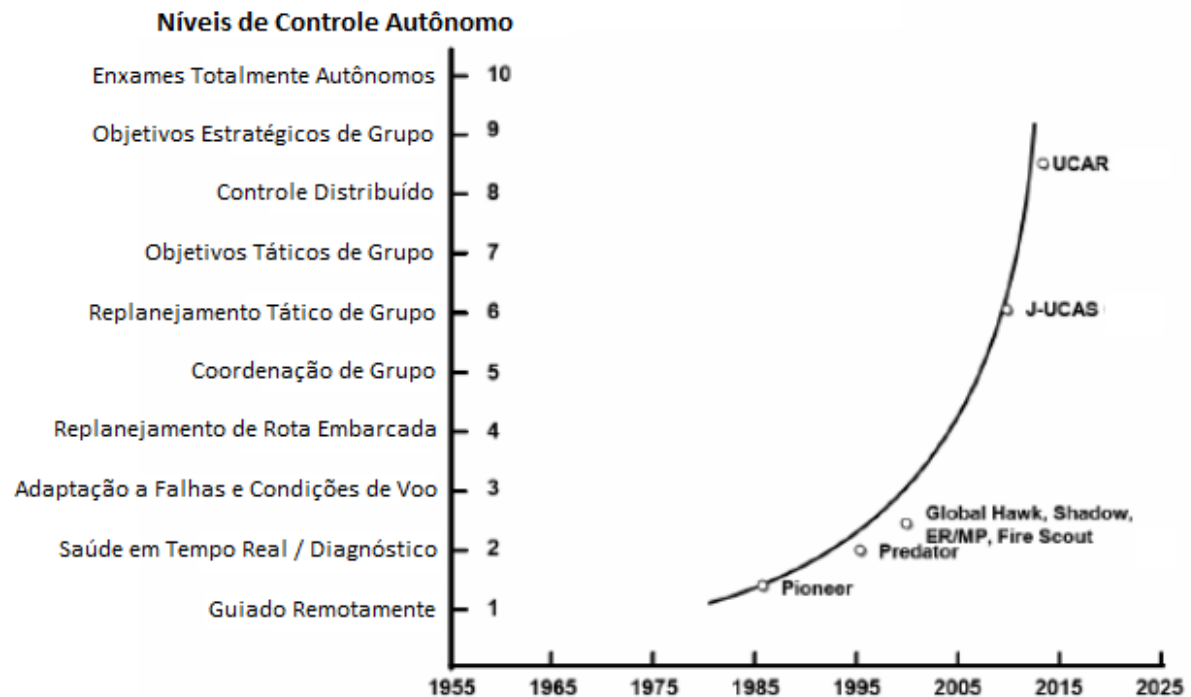
O contexto dos UAVs, mais fortemente quando em ambientes externos, está inserido em um ambiente de incerteza que muda rapidamente. Nessa área, alguns estudos têm tratado sobre o controle autônomo dos UAVs, como em (CHEN *et al.*, 2009) e (SHOLES, 2007).

É apresentada por Sholes (2007) uma técnica de avaliação que combina tecnologias de simulação para criar um método de avaliação de autonomia de um veículo aéreo não tripulado que aproveita ao máximo os ambientes virtuais visuais e simulações construtivas. Seu trabalho também apresenta nove comportamentos colaborativos de grupo que foram definidos em 2003 pela AATD³. São eles, em tradução livre: “Ver e Evitar” Voo Cooperativo com Prevenção de Obstáculos e Colisão, Vingar Morte/Proteção do Time, Reconhecimento/Segurança Cooperativa, Ajuste do Time a Falha de Componente, Adaptação da Rede para Assegurar Comunicações, Manter Vigilância Sobre Vários Alvos Móveis em Terreno Urbano, Estabelecer Vários Pontos Ótimos de Observação, Dupla Elevação e Suportar Abastecimento Não-Tripulado.

Uma métrica apresentada em Sholes (2007) e Chen *et al.* (2009) e citada como amplamente aceita para comparar a autonomia em UAVs é a ACL (*Autonomous Control Level*), desenvolvida pela força aérea norte-americana.

³ AATD (*Applied Aviation Technology Directorate*), órgão ligado às forças armadas norte-americanas responsável por sua tecnologia de aviação.

Figura 5 – *Autonomous Control Level (ACL)* e tendência da autonomia em UAVs



Fonte: Adaptado de (CHEN *et al.*, 2009)

Na Figura 5, tem-se um gráfico que relaciona os dez níveis de autonomia em UAV, descritos pela ACL, com uma tendência de evolução ao longo dos anos. Os níveis começam com Remotamente Guiado, como menor grau de autonomia, indo até o nível mais complexo que é o dos Enxames Plenamente Autônomos.

Neste trabalho, a relevância da computação autônoma está intimamente ligada à auto-otimização da distribuição em enxame dos UAVs, assim como a autoconfiguração dessa distribuição para busca no caso de falha de um ou mais UAVs.

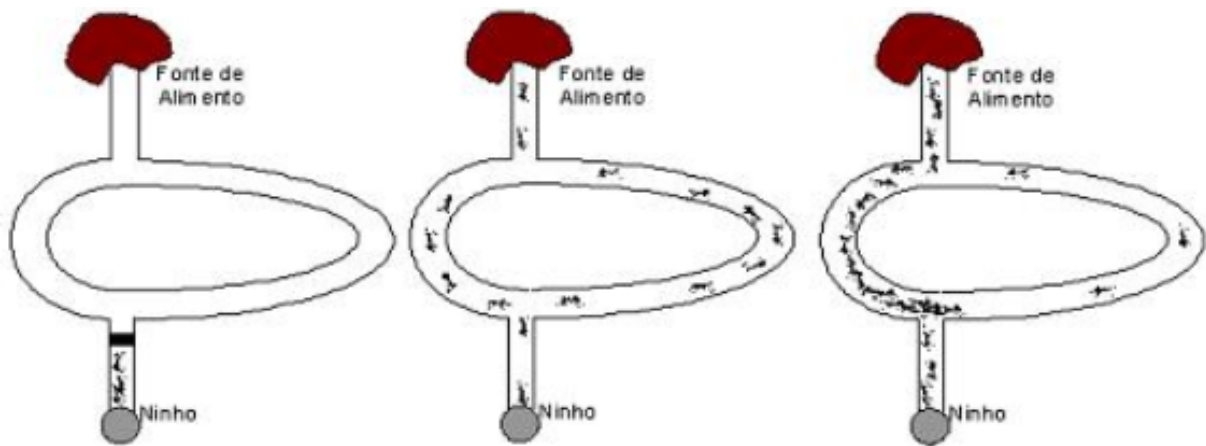
2.4 Computação Bioinspirada e Comportamento de Enxame

Encontra-se na natureza uma série de exemplos de funcionamento de sistemas, compostos por elementos menores que cooperam em prol do todo, e de como solucionar problemas, dos mais simples aos mais complexos.

No caso das colônias de formigas (CASTRO, 2010), elas exploram inicialmente as fontes de comida de melhor qualidade ou que estão mais próximas da colônia para, só depois, buscar as fontes mais distantes e de menor qualidade. Na imagem seguinte, pode-se verificar um experimento onde um grupo de formigas está inicialmente preso e após a saída existe uma bifurcação, com um caminho longo e um caminho curto, ambos levando à mesma fonte de

comida. Quando as formigas são liberadas, algumas seguem pelo caminho mais curto e outras pelo caminho mais longo. Com o tempo, quanto mais formigas seguirem pelo caminho mais curto, maior a liberação de feromônios nesse caminho e maior a quantidade de formigas que tenderá a segui-lo. Esse comportamento pode ser utilizado para estimar a menor rota a ser seguida entre dois pontos (figura 6).

Figura 6 – Ilustração sobre experimento de coleta de alimento entre formigas e escolha do menor caminho



Fonte: <http://www.lvcon.computacaonatural.com.br/tema?tema=4>

O trabalho de Dressler e Akan (2009) nos fornece um resumo que relaciona os princípios biológicos com as suas aplicações na área de rede de computadores. A tabela 1 oferece uma visão geral dessas aplicações.

Tabela 1 – Categorização de fenômenos biológicos e algoritmos de rede que imitam esses conceitos

Princípio biológico	Campo de aplicação em redes
Inteligência de enxame e insetos sociais	Busca distribuída e otimização; roteamento em redes de computadores, especialmente MANETs, WSNs e redes <i>overlay</i> ⁴ ; alocação de tarefas e recursos.
Sincronização de vagalumes	Sincronização de <i>clock</i> totalmente distribuída e robusta.
Sistema ativador-inibidor	Auto-organização de sistemas autônomos; coordenação distribuída; adaptação contínua de parâmetros de sistemas em ambientes altamente dinâmicos.
Sistema imunológico artificial	Segurança de rede; detecção de anomalias e mau comportamento.
Disseminação epidêmica	Distribuição de conteúdo em redes de computadores (por exemplo, nas DTNs ⁵); redes <i>overlay</i> ; análise de propagação de <i>worms</i> ⁶ e vírus na internet.
Redes de sinalização celular	Controle e coordenação em sistemas massivamente distribuídos; programação de redes centralizadas de sensores e atuadores.

Fonte: Traduzido e adaptado de (DRESSLER; AKAN, 2009).

A *Mobile Ad hoc Network* (MANET) citada nas aplicações da tabela 1 é uma rede móvel que se estabelece conforme a necessidade. Tem uma relação direta com as WSN, pois alguns autores consideram as WSN como tipos especiais de MANETs (SARMENTO *et al.*, 2012).

Algoritmos bioinspirados podem ser usados tanto para problemas de otimização, quanto para exploração, mapeamento ou reconhecimento de padrões (DRESSLER; AKAN, 2009). Uma das abordagens de algoritmos bioinspirados aplicados a redes de computadores, também vista na tabela 1, é a inteligência de enxame. Ela é baseada na observação do comportamento coletivo de sistemas descentralizados e auto-organizados, como colônias de formigas ou enxames de abelhas (DRESSLER; AKAN, 2009) (BITAM *et al.*, 2010). As principais características, que são ao menos parcialmente partilhadas pelos membros dessa classe de algoritmos, são o uso de uma metáfora natural, inerente paralelismo, natureza estocástica, adaptabilidade, e utilização de realimentação positiva (DORIGO *et al.*, 1996).

Os algoritmos que utilizam inteligência de enxame basicamente imitam métodos existentes na natureza para direcionar a busca de uma solução ótima e usam um conjunto de soluções a cada iteração, ao invés de apenas uma solução (PHAM *et al.*, 2006).

Segundo Kennedy e Eberhart (1995), a inteligência de enxame tem cinco princípios básicos. O primeiro é o princípio da proximidade, onde a população deve ser capaz de executar cálculos de tempo e espaço simples. O segundo é princípio da qualidade, onde a população deve ser capaz de responder satisfatoriamente a elementos do ambiente. O terceiro é o princípio da resposta diversificada, onde a população não deve enviar suas atividades por canais excessivamente estreitos. O quarto é o princípio da estabilidade, onde a população não deve mudar seu comportamento toda vez que o ambiente mudar. O quinto é o princípio da adaptabilidade, pelo qual a população deve ser capaz de mudar seu comportamento quando o custo computacional valer a pena.

Se um problema de otimização possuir apenas uma solução ótima, as populações geradas pelos algoritmos baseados em inteligência de enxame tenderão a convergir para a

⁴ Rede *Overlay* é uma rede de computadores construída sobre a infraestrutura de outra rede. Os nós desta rede podem ser conectados logicamente, cada um a um caminho próprio, podendo haver vários destes caminhos em um enlace físico. Um exemplo de uso são as redes *peer to peer* (DING *et al.*, 2009)

⁵ DTN é uma rede tolerante a atraso cuja arquitetura original foi projetada para ser uma internet interplanetária. A DTN é um sistema de comunicação usado para prover serviços de internet em distâncias interplanetárias para suporte de exploração do espaço profundo (PURI; SINGH, 2013).

⁶ *Worms* são códigos de autopropagação, semelhantes a vírus. Diferente do vírus, o *worm* não precisa de um programa hospedeiro para se propagar. Ex: Code Red (SELLKE *et al.*, 2008).

solução ótima. Caso tenham múltiplas soluções ótimas, esse tipo de algoritmo pode ser usado para capturá-las em sua população final (PHAM *et al.*, 2006).

Algumas soluções podem trabalhar em conjunto com Algoritmos Genéticos (no inglês, *Genetic Algorithm* (GA)) com heurísticas que dependem do domínio para melhorar os resultados. Os Algoritmos Genéticos são baseados em seleção natural e recombinação genética e funcionam escolhendo soluções da população corrente e aplicando operações genéticas, como mutações e cruzamentos, para criar uma nova população (PHAM *et al.*, 2006).

Uma característica que pode se mostrar difícil ao trabalhar com inteligência de enxame é que, pelo seu comportamento probabilístico, uma solução executada uma segunda vez não terá exatamente o mesmo comportamento que a primeira execução, não sendo possível, portanto, reproduzir igualmente cada simulação.

Segundo (SOUISSI *et al.*, 2013), as soluções mais usadas para ambientes em tempo real são os métodos probabilísticos, como *Ant Colony Optimization* (ACO) e *Particle Swarm Optimization* (PSO), vistos nos itens 2.4.1 e 2.4.2. Nos tópicos a seguir, detalho cada um deles e apresento um pouco sobre outros algoritmos mais recentes, como o *Bat Algorithm* (BA), visto no item 2.4.3.

2.4.1 Algoritmo de otimização baseado em colônia de formigas (no inglês, *Ant Colony Optimization*)

Foi proposto por Dorigo *et al.* (1996) e utiliza o princípio de forrageamento das formigas. Nele, formigas artificiais se baseiam em formigueiros reais e sua busca por alimentos para encontrar boas soluções de problemas de melhor caminho, utilizando, para tanto, o paradigma da trilha de feromônio das formigas reais. Alguns casos especiais de meta-heurísticas do ACO foram propostos na literatura. Aqui detalho um dos mais conhecidos, o *Ant System* (AS).

2.4.1.1 *Ant System* (AS)

Animais como formigas podem estabelecer rotas com caminhos mais curtos da sua colônia às fontes de alimento e, então, de volta. A forma que elas utilizam para compartilhar a informação entre os outros elementos da colônia é através de trilhas de feromônios. Uma formiga segue um caminho aleatório para buscar alimento e quando volta ao ninho deposita no caminho algum feromônio, em quantidade variável, que serve como uma trilha para buscas futuras (DRESSLER; AKAN, 2009; DORIGO *et al.*, 1996). Quando uma formiga encontra uma

trilha de feromônio já existente, ela pode detectar e decidir seguir por esse caminho com uma maior probabilidade, reforçando, assim, a trilha com o seu próprio feromônio. Dessa forma, quanto mais formigas seguirem por um caminho, mais atraente às demais ele se torna (DORIGO *et al.*, 1996).

Voltando à figura 6, quando a primeira formiga chega à bifurcação, a probabilidade dela seguir pela direita ou pela esquerda é a mesma, já que nenhum feromônio foi ainda depositado. À medida que as formigas seguem em busca do alimento, a probabilidade do menor caminho ser escolhido é maior, pois cada vez que uma formiga volta ela deposita feromônio e cada vez que uma formiga deposita feromônio a probabilidade de escolha do caminho pelas demais muda, aumentando a probabilidade do caminho com mais feromônio. No caminho menor esse acúmulo vai acontecer mais rapidamente, fazendo com que mais formigas tendam a escolher esse caminho. A ideia é que no final a maioria das formigas terá escolhido o menor caminho.

Segundo Dorigo *et al.* (1996), as principais diferenças entre a colônia do *Ant System* e a colônia de formigas reais são que as formigas artificiais terão memória, não andarão totalmente às escuras e o tempo será discreto.

2.4.2 Algoritmo de otimização baseado em enxame de partículas (no inglês, *Particle Swarm Optimization*) - PSO

Foi proposto por Kennedy e Eberhart (1995). No PSO, tem-se um conjunto de indivíduos, chamados partículas e se leva em consideração o melhor desempenho que cada um desses indivíduos teve até o momento, assim como o melhor desempenho dos indivíduos vizinhos para decidir os próximos passos (CASTRO, 2010).

O PSO tem bases no conceito de vida artificial em geral, englobando bandos de aves, cardumes de peixes e teoria de enxame. O compartilhamento de informações entre elementos de uma espécie oferece uma vantagem evolucionária e, segundo (KENNEDY; EBERHART, 1995), essa hipótese foi fundamental para o desenvolvimento do PSO.

No Quadro 1 é mostrado um algoritmo para PSO. Segundo (BROWNLEE, 2011), conceitualmente, P_{best} é similar à memória individual, onde cada indivíduo lembra sua própria experiência e P_{gbest} é similar ao conhecimento do grupo que cada indivíduo procura alcançar. Um alto valor incremental de p_{best} relativo a g_{best} resulta em um excesso de indivíduos isolados no espaço do problema, enquanto o contrário (incremento de g_{best} relativamente alto) resulta em

Quadro 1 – Algoritmo para o PSO

	Entrada: TamanhoProblema, TamanhoPopulacao
	Saída: Pgbest
1	Populacao $\leftarrow \emptyset$
2	Pgbest $\leftarrow \emptyset$
3	Para i=1 até TamanhoPopulacao faça
4	Pvelocidade $\leftarrow VelocidadeAleatoria()$
5	Pposicao $\leftarrow PosicaoAleatoria(TamanhoPopulacao)$
6	Ppbest $\leftarrow Pposicao$
7	Se Custo (Ppbest) \leq Custo(Pgbest) ento
8	Pgbest $\leftarrow Ppbest$
9	Fim
10	Fim
11	Enquanto \neg CondicaoParada() faa
12	Para cada P \in Populacao faa
13	Pvelocidade $\leftarrow AtualizaVelocidade(Pvelocidade, Pgbest, Ppbest)$
14	Pposicao $\leftarrow AtualizaPosicao(Pposicao, Pvelocidade)$
15	Se Custo(Pposicao) \leq Custo(Ppbest) ento
16	Ppbest $\leftarrow Pposicao$
17	Se Custo (Ppbest) \leq Custo(Pgbest) ento
18	Pgbest $\leftarrow Ppbest$
19	Fim
20	Fim
21	Fim
22	Fim
23	Retorna Pgbest

Fonte: (BROWNLEE, 2011) - tradução livre.

um bando correndo prematuramente em direção a um mínimo local ⁷. A população responde aos fatores de qualidade Ppbest e Pgbest e muda seu status apenas quando Pgbest muda (a população é adaptativa e respeita o princípio de estabilidade).

2.4.3 Algoritmo de Morcegos (no inglês, *Bat Algorithm - BA*)

Foi proposto por Yang (2010) e tem como base o comportamento de morcegos em seu processo de busca por alimento. Cada morcego voa aleatoriamente em busca de alimento e, quando encontra sua presa, modifica frequência, taxa de emissão do pulso (guincho) e amplitude da onda sonora; este processo continua até o objetivo ser atingido.

De forma simplificada, o algoritmo faz a busca dos elementos através de eco-localização. A velocidade dos morcegos é variável e a frequência das ondas fixa, tendo uma variação no comprimento de onda dependendo da proximidade com seus alvos.

⁷ Nessa situação, o mínimo local induziria ao erro, pois o valor esperado seria o mínimo global.

2.4.4 Outros algoritmos bioinspirados

2.4.4.1 Algoritmo baseado em colônia de abelhas (no inglês, *Bee Colony - BC*)

O algoritmo baseado no forrageamento das colônias de abelhas, *Bee Colony* (BC), é inspirado no comportamento das abelhas coletoras de mel (forrageiras) em busca de alimento. Nele, abelhas batedoras saem e exploram a região em busca de uma fonte de alimento. Quando acham, elas retornam à colmeia, depositam o néctar e pólen e seguem para a "pista de dança" (*dance floor*) para transmitir e compartilhar a descoberta com os demais membros da colmeia através de uma dança (*waggle dance*). Essa dança informa a direção do alimento, sua distância da colmeia e sua qualidade (ou *fitness*). Algumas abelhas são, então, recrutadas e se tornam abelhas forrageiras. O seu número será proporcional à quantidade de alimento informado pelas abelhas batedoras - trechos que possuem maiores quantidades de néctar ou pólen que podem ser coletados com menos esforço devem ser visitados por mais abelhas e trechos com menos néctar ou pólen devem receber poucas abelhas. Nessa nova fase de exploração, a abelha coleta o alimento e calcula sua quantidade para decidir se continua coletando e memorizando essa localização ou se abandona essa fonte e retorna à colmeia como uma abelha simples (CASTRO, 2010; PHAM *et al.*, 2006).

As abelhas podem se espalhar por longas distância (acima de 10km) e em múltiplas direções, podendo explorar simultaneamente várias fontes de alimento. Durante toda a estação de coleta, a colmeia mantém um percentual de abelhas batedoras que continuamente se movem de um trecho a outro procurando canteiros de flores promissoras para a colmeia. A dança executada quando uma abelha batidora retorna vai definir, de acordo com a qualidade do alimento encontrado e com a energia necessária para coletá-lo, a quantidade de abelhas que retornarão para realizar a coleta. Enquanto uma fonte é viável, mais abelhas são recrutadas para a coleta (PHAM *et al.*, 2006).

2.4.4.2 Algoritmo de busca dos lobos (no inglês, *Wolf Search Algorithm*) - WSA

O *Wolf Search Algorithm* (WSA) imita a busca dos lobos por alimentos e sua sobrevivência evitando ameaças. Os lobos caçam juntos em matilhas, formando um único núcleo familiar, mas permanecem em silêncio e furtivos quando caçando juntos. Cada lobo caça de forma independente, lembrando seu percurso, e apenas se une a outro da matilha se esse se encontra em melhor posição quanto à presa. Dessa forma, a comunicação a longo alcance é

praticamente eliminada entre eles. Outra característica é que, à medida que os lobos caçam, eles evitam potenciais ameaças, como humanos ou outros predadores animais (TANG *et al.*, 2012), quando uma ameaça é detectada, o lobo vai buscar uma nova posição saindo do seu campo visual.

A seguir serão vistos alguns trabalhos correlatos ao estudo na presente pesquisa.

3 TRABALHOS RELACIONADOS

Para o presente trabalho foi feito um levantamento bibliográfico acerca dos trabalhos relacionados na literatura. Os mecanismos de busca e as palavras-chave utilizadas nesse levantamento são mostrados no quadro 2.

Tabela 2 – Levantamento bibliográfico

Mecanismos de busca	Palavras-chave
1. ACM Digital Library	1. Bioinspired target search
2. IEEE Xplore	2. UAV target search
3. Periódicos CAPES	3. Swarm target search
	4. Swarm uav target search
	5. Swarm uav search rescue

Alguns dos trabalhos identificados durante o levantamento bibliográfico foram selecionados por apresentar maior relação com a presente pesquisa.

O trabalho de (PALMIERI *et al.*, 2019) trata do gasto de energia de um sistema autônomo de robôs para localização de alvo cuja posição não é previamente conhecida. Segundo os autores, isso passa por duas fases, são elas o reconhecimento da área onde o alvo estaria e o recrutamento dos robôs em relação ao alvo. Uma premissa importante para o funcionamento correto do paradigma proposto é que deve haver mais robôs do que alvos no espaço de procura. Os autores levam em consideração a colaboração dos robôs, porém com baixo custo de comunicação, processamento e sensoriamento; usam uma abordagem não centralizada de coordenação dos robôs. Os algoritmos utilizados para a resolução destas duas fases foram o *Ant System Algorithm* (AS), o *Firefly Algorithm* (FA), o PSO e o *Artificial Bee Colony* (ABC). Destes o que obteve melhor relação de gasto por desempenho na fase de recrutamento foi o FA. Os robôs são colocados em um ambiente cujo conhecimento não é prévio. Quando um robô identifica o alvo, transmite esta informação para os demais a fim de haver um “recrutamento” - convergência dos robôs para região do alvo. Cada robô decide se quer ou não aceitar o convite do enviado. Esta estratégia é boa para múltiplos alvos em um mesmo espaço de busca. O principal desafio, segundo os autores, foi determinar o número de robôs para o processo de busca. Os autores utilizam o AS para o estágio de busca e o FA, ABC e PSO para recrutamento (convergência dos UAVs). Este trabalho diverge dessa pesquisa pois foca no desempenho e gasto energético na chamada fase de recrutamento, quando o alvo já teve sua posição identificada, e os métodos bioinspirados têm sua eficácia testada durante o processo de aproximação do alvo.

O trabalho de (INNOCENTE; GRASSO, 2019) apresenta a utilização de Inteligência de Enxame aplicada em UAVs com o propósito de monitorar e identificar focos de incêndio em floresta; mapear ponto de risco para início de incêndios em florestas; e auxiliar os bombeiros no suporte para operações de incêndio florestal. Os autores escolheram os UAV autônomos e que se comportam como enxame, utilizando para tanto o PSO. O artigo faz análise dos dados obtidos através de diversas simulações feitas a partir de um conjunto de modelos matemáticos de propagação de incêndios. Foi detectado pelos autores que quanto maior o número de partículas maior será a eficiência para encontrar focos ou possíveis focos de incêndio. A área total utilizada na simulação foi de 6529 m². foram usados 30, 40, 60,70 e 100 drones. Infelizmente os gráficos analisados pelos autores não mostram a quantidade de sucessos (identificação do foco) para valores abaixo de 70 drones. Por fim, o trabalho conclui que o uso de drones para auxiliar bombeiros em incêndios florestais pode chegar a um nível de 90% de eficiência. Como o trabalho utilizou apenas um algoritmo bioinspirado, é possível que outros algoritmos bioinspirados possam alcançar melhores resultados. No entanto, os modelos matemáticos usados e o algoritmo de PSO implementado mostrou-se bem eficiente.

Já o artigo de (MCCUNE; MADEY, 2013) propõe um novo comportamento de busca onde se divide a área de busca, e quando combinado com soluções de enxame anteriores, constitui um problema de otimização de como melhor atribuir soluções de enxame para uma topologia complexa. Ele aborda dois algoritmos de busca cooperativa descentralizada que utilizam inteligência de enxame, o *Parallel Path Search* e o *SWEEP Protocol* e propõe que a área de busca seja dividida em sub-regiões de forma a usar esses dois algoritmos em pedaços separados do mapa original de busca de acordo com o melhor em cada divisão do mapa. Devido às limitações desses dois algoritmos, com relação a terreno e custo computacional, é proposta a ideia de adicionar UAVs sentinela, o que gera um novo problema de otimização visto que se deve buscar a melhor forma de particionar o mapa considerando os UAVs disponíveis, tempo de busca, e custos de comunicação, coordenação e processamento. Essa busca melhorada baseada em sentinela ainda estava em desenvolvimento quando da publicação do artigo.

O artigo de (GERKE *et al.*, 2011) aborda dois pontos bioinspirados com *Micro Air Vehicles* (MAVs), o voo de insetos e a detecção de borda e reconhecimento de forma usado por abelhas e outros insetos. Esse trabalho apresenta um repertório de módulos comportamentais que podem ser controlados por meio de máquinas de estados finitos (no inglês, *Finite State Machines* - FSM); e detectores de movimentos básicos (no inglês, *Elementary Movement Detectors* - EMD)

combinados com um algoritmo de detecção de borda centralizada ou espaçada para produzir melhor informação na detecção de objetos. O artigo utiliza máquinas de estados finitos para projetar o comportamento dos MAVs em ambiente *indoor*, onde os status descrevem o tipo de comportamento (e.g. *Left Turn*) e as transições descrevem as condições necessárias para mover de um estado a outro (e.g. *Turn Complete*). Outro ponto do artigo é combinar detecção de borda com informação de movimento captada por EMDs, de forma que as informações de EMD possam ser usadas como filtro para remover o ruído na identificação das bordas. O foco do artigo é o voo autônomo de MAVs em ambientes *indoor* buscando autonomia no trajeto e desvio de obstáculos.

O trabalho de (ZHANG *et al.*, 2010) apresenta uma proposta de determinação de rota para UAV utilizando o *Ant Colony Optimization* (ACO). A ideia principal é dividir a área em grids e, tendo um ponto inicial e um ponto final definidos, utilizar o ACO para determinar a menor rota. Este trabalho não se aplica à presente pesquisa por trabalhar com destino predeterminado.

Já o artigo de (BAO *et al.*, 2010) apresenta uma proposta para determinar rotas de reconhecimento para UAVs, considerando existência dos alvos e pontos de ameaças aos UAVs, e utilizando o *Particle Swarm Optimization* (PSO) como base para seu algoritmo. O algoritmo foi alterado para considerar como fatores relevantes a importância do alvo que se deseja localizar/monitorar, onde cada alvo tem um valor associado; a eficácia do caminho de reconhecimento, onde o UAV deve se aproximar o suficiente para o alvo estar ao alcance do seu equipamento de reconhecimento; e a sobrevivência do UAV, colocada como a capacidade de se manter escondido, evitar mísseis inimigos entre outros. O algoritmo determina então o caminho a ser seguido de forma a evitar que os UAVs se aproximem das ameaças e se aproximem mais dos alvos mais importantes. Esse trabalho também se mostrou bastante próximo ao objetivo da presente pesquisa.

No próximo capítulo será discutida a proposta para resolução do problema abordado.

4 PROPOSTA

A presente pesquisa tem como proposta identificar se é viável utilizar UAVs com comportamento bioinspirado, respeitando certas restrições, para solucionar um problema de busca de um alvo de posição desconhecida em uma dada área.

Para tanto, foram seguidos os seguintes passos:

- (a) Seleção dos algoritmos de enxame com base na literatura;
- (b) Definição das restrições aplicadas, a fim de verificar a eficácia dos algoritmos em diferentes cenários;
- (c) Alteração dos algoritmos selecionados para aplicação ao problema de busca;
- (d) Comparação dos resultados obtidos nas simulações em relação aos objetivos alcançados.

As restrições consideradas na simulação foram: quantidade de UAVs, tamanho da área de busca, velocidade de deslocamento e tempo de voo.

Quanto à quantidade de UAVs, encontram-se na literatura pesquisas utilizando entre 20 e 40 UAVs, como é visto em Palmieri *et al.* (2019). No entanto, costuma-se extrapolar essa quantidade nas simulações, podendo usar valores próximos ou superiores a 100 UAVs, como usado por Innocente e Grasso (2019). Para esse estudo, utilizou-se a quantidade de 5, 10 e 40 UAVs, com o intuito de mostrar o impacto no resultado final.

Para definição do tamanho da área de busca, estudos utilizam, normalmente, valores com grid de 100m x 100m ($10.000m^2$), como é o caso de Innocente e Grasso (2019). No entanto, uma área de busca desse tamanho não seria condizente com a necessidade real do uso de *drones* em uma busca. Assim, temos o cenário inicial com grid de 200m x 200m ($40.000m^2$), um valor apenas de efeito comparativo, e esse tamanho é ampliado até 1000m x 1000m ($1.000.000m^2$).

Para restrição de velocidade de deslocamento e tempo de voo, foi considerado o modelo comercial DJI Phantom 4 Pro, cujas especificações aparecem na tabela 3.

Tabela 3 – DJI Phantom 4 Pro

Velocidade de deslocamento	até 72km/h = 20m/s
Tempo de bateria	até 30min

Fonte: <https://www.dji.com/br/phantom-4-pro>

No algoritmo BA e PSO, os UAVs são inicialmente posicionados aleatoriamente no centro da grid e de lá seguem para a busca. No caso do *Parallel Path* (PP), os UAVs são inicialmente colocados no início da grid.

Quanto aos algoritmos escolhidos, inicialmente foi trabalhado o AS, porém sua característica de depositar feromônio ao voltar da localização do alvo teria que ser desconsiderada para melhor adequação ao problema, tirando uma característica importante desse algoritmo. Da mesma forma, o BC foi descartado devido a sua característica de volta da abelha, para aviso e recrutamento das demais. Assim, um outro algoritmo dentre os mais presentes na literatura de inteligência de enxame, o PSO, foi o primeiro selecionado para teste, tendo seus resultados iniciais publicados em (PAULA *et al.*, 2016).

Como segunda opção, dentre os estudados, haviam o WSA e o BA. É característica do WSA os lobos manterem uma certa proximidade com o alfa, o que torna esse algoritmo, sob certo aspecto, similar ao PSO. Assim, o BA foi selecionado como segundo algoritmo para teste por não apresentar tal restrição.

A intenção deste trabalho é mostrar a aplicabilidade dos algoritmos e verificar sua eficiência. Dessa forma, as preocupações sobre os algoritmos de comunicação entre os UAVs e de reconhecimento de padrões para identificação do alvo não são o foco da pesquisa. Portanto, parte-se princípio de que os UAVs serão usados como nós sensores em uma rede de sensores sem fio. Para suprir a comunicação entre estes, sugere-se usar o protocolo Zigbee, já estabelecido na literatura. O algoritmo de reconhecimento adotado será um algoritmo padrão, sem ser abordado seu funcionamento neste trabalho. Desta forma, os esforços algorítmicos serão centrados no comportamento dos grupos de busca. Outro fator a ser colocado é que não se está considerando tempos de atraso que poderiam afetar a velocidade de deslocamento, como *delay* de comunicação entre os UAVs, *delay* no reconhecimento do alvo e *delay* causado pelo desvio de obstáculos.

Os algoritmos bioinspirados utilizados, PSO e BA, foram adaptados para o presente problema. Nos exemplos encontrados, é usual que haja valores de referência para que se possa comparar, durante as iterações, a solução encontrada no momento com a solução ótima até então. No problema de busca de alvo desconhecido, não há como haver comparação de que solução é a melhor até que o alvo seja encontrado por um dos UAVs. Sendo assim, os dois algoritmos buscam o alvo a cada iteração de forma aleatória. A partir do momento que o alvo é encontrado por um deles, há convergência dos demais para o, então, ponto ótimo.

O BA também teve uma alteração na sua movimentação, foi aplicado *random walk* até o primeiro UAV localizar o alvo, quando então passa a utilizar a convergência utilizada pelo APSO. Isso permitiu uma melhoria na taxa de localização do alvo.

Tanto o PSO quanto o BA, tiveram alterada sua função objetivo para atender o

problema da busca e obedeceram os mesmos parâmetros nas simulações. A função objetivo de onde se calcula o valor de comparação para o *fitness* e o *best*¹ passa a considerar valores padrão até que o alvo seja localizado, quando, então, passa a considerar a posição do UAV que localizou o alvo.

No quadro 2, está o algoritmo simplificado do PSO. Um ponto chave é o cálculo da posição das partículas que, no PSO, mantém uma proximidade das demais durante o movimento. Para a movimentação das partículas, foi usado o cálculo do *Accelerated PSO* (APSO), desenvolvido por (YANG, 2010) em contraposição à movimentação original proposta por (KENNEDY; EBERHART, 1995).

Quadro 2 – Algoritmo adaptado do *Accelerated PSO*

1	Inicializa partículas com solução randômica
2	Inicia contagem de Tempo
3	Enquanto critério de parada não for encontrado
4	Para cada partícula Faça
5	Avalia solução
6	Verifica alvo
7	Se alvo encontrado Então
8	Atualiza <i>best</i>
9	Fim-Se
10	Atualiza posição da partícula
11	Verifica posição dentro dos limites da grid
12	Fim-Para
13	Fim-Enquanto
14	Retorna Tempo

Fonte: Adaptado de (KENNEDY; EBERHART, 1995), (BROWNLIE, 2011) e (YANG, 2010)

A etapa de movimentação considera os modelos propostos por Kennedy e Eberhart (1995) e Yang (2010). Nela, as posições *x* e *y* das partículas são sorteadas para se movimentar aleatoriamente em grupo, conforme pode ser visto na figura 7, onde o bolinha vermelha é o alvo e os asteriscos são os UAVs.

A partir do momento que o alvo é localizado, a movimentação segue um cálculo que as faz convergir para a sua localização. Essa movimentação de convergência é calculada no Matlab através das seguintes expressões:

$$x_n = \text{round}(x_n \cdot (1-b) + x_{best} \cdot b + a \cdot \text{randi}([1,n]) - 0.5);$$

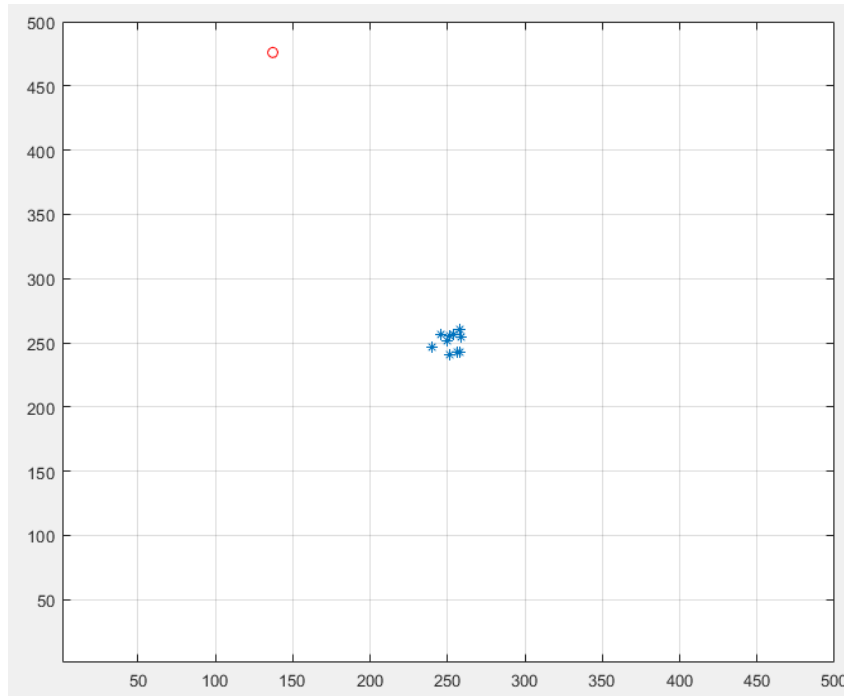
$$y_n = \text{round}(y_n \cdot (1-b) + y_{best} \cdot b + a \cdot \text{randi}([1,n]) - 0.5);$$

Onde, a cada iteração, um valor α (o *a* do cálculo de posição) é recalculado de acordo com γ . Esses valores, juntamente com o β (*b*) vão influenciar na velocidade de convergência

¹ O *fitness* é calculado a partir da função objetivo e é a partir do seu valor que são definidos os valores de *best*, a melhor solução apresentada, a ser seguida pelos demais

das partículas. O valor de b é arbitrado entre 0 e 1, da convergência mais lenta à mais rápida. O valor utilizado para γ foi o de 0.7, sugerido por (YANG, 2010).

Figura 7 – Comportamento das partículas no PSO



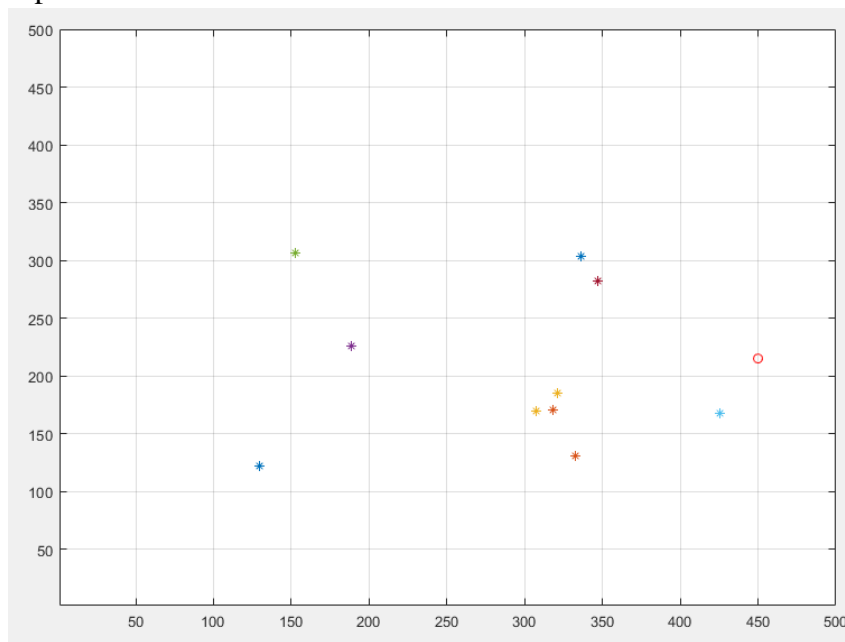
No *Bat Algorithm*, visto de forma simplificada no quadro 3, os morcegos se movimentam como ondas que vão se expandindo, diferente do PSO onde a movimentação mantém um agrupamento das partículas. Com a alteração na movimentação do BA, os morcegos passam a se movimentar individualmente até que um deles localize o alvo. A figura 8 ilustra a movimentação no BA.

No BA, a função objetivo, como dito anteriormente, também foi alterada para considerar se o alvo foi ou não encontrado. Até o alvo ser encontrado, os morcegos atuam de forma individual, sem comunicação, até que um deles encontre o alvo e passe sua posição como a melhor até o momento. Com isso, os morcegos passam a convergir até a posição do alvo.

Para a etapa de movimentação do BA, calcula-se para cada elemento uma frequência $Q(i) = Q_{min} + (Q_{max} - Q_{min}) * rand$ (código Matlab), com valores Q_{min} e Q_{max} sugeridos entre 0 e 2, por (MIRJALILI *et al.*, 2014). A partir do valor de Q , é calculada a posição do morcego. Nesse ponto, houve uma modificação do algoritmo original. Depois de alguns testes, foi preferível adaptar o cálculo de convergência de posição para usar o do APSO, utilizando a frequência do morcego como parte do seu cálculo de posição.

O critério de parada adotado nos dois algoritmos foi o mesmo. A simulação encerra

Figura 8 – Comportamento no BA

Quadro 3 – *Bat Algorithm* alterado

-
- 1 Inicializa partículas com solução randômica e *fitness* global
 - 2 Inicia contagem de Tempo
 - 3 Enquanto critério de parada não for encontrado
 - 4 Para cada elemento Faça
 - 5 Avalia *fitness*
 - 6 Atualiza frequência, velocidade e posição do elemento
 - 7 Verifica posição dentro dos limites da grid
 - 8 Atualiza (*loudness*) e (*pulse rate*)
 - 9 Verifica alvo com valores atualizados
 - 10 Se alvo encontrado Então
 - 11 Atualiza best
 - 12 Fim-Se
 - 13 Atualiza *fitness*
 - 14 Fim-Para
 - 15 Fim-Enquanto
 - 16 Retorna Tempo
-

Fonte: Adaptado de (YANG, 2010) e (MIRJALILI *et al.*, 2014)

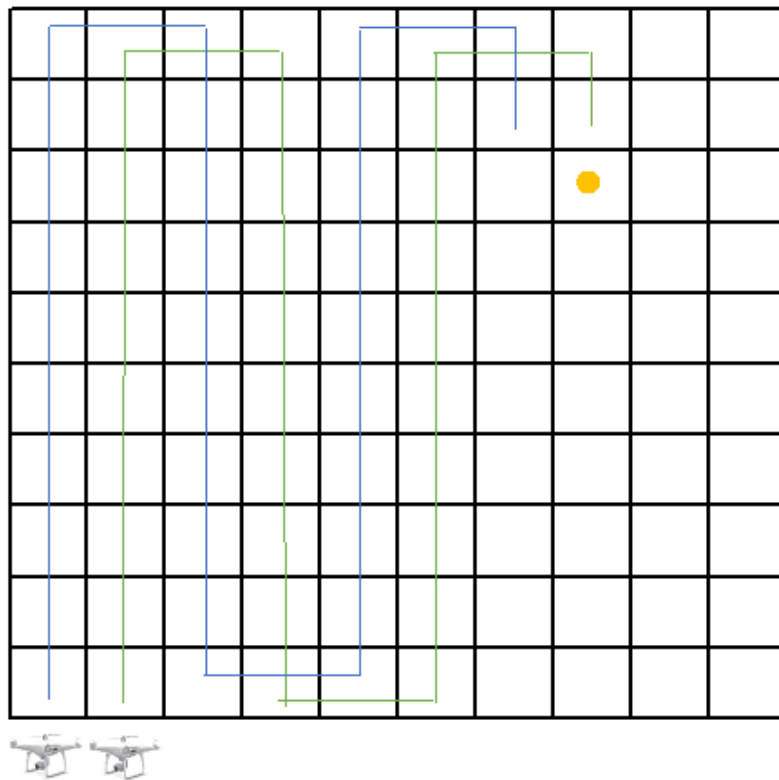
ao atingir o máximo de iterações definido ou quando todos os UAVs localizam o alvo.

Os algoritmos descritos aqui foram testados em 6 (seis) cenários a fim de que seu comportamento pudesse ser analisado em termos de percentual de sucesso na localização e tempo estimado da busca, considerando o espaço percorrido e a velocidade média. O próximo capítulo descreve os cenários e analisa seus resultados.

5 LEVANTAMENTO E ANÁLISE DE DADOS

O presente capítulo trata do levantamento de dados a partir de simulações feitas com os algoritmos PSO e BA, tendo como parâmetro o tempo de processamento de cada um deles na busca de um alvo fixo de localização desconhecida. Além dos algoritmos bioinspirados, foi utilizado, para efeito de comparação, o algoritmo de *Parallel Path Finder* (PP), que faz uma varredura em coluna/linha do caminho, similar ao mencionado em (WEI; WEI, 2009) e (MCCUNE; MADEY, 2013). Na figura 9, tem-se um exemplo de seu funcionamento.

Figura 9 – Funcionamento da varredura em coluna utilizada



A comparação dos resultados foi feita através de um conjunto de cenários para que se pudesse fazer um melhor estudo do comportamento de cada algoritmo, bem como, suas potencialidades e limitações.

5.1 Especificações de *hardware* e *software* utilizados para a simulação

O *software* Matlab foi escolhido em virtude de ser utilizado pelos pesquisadores estudados no referencial teórico e nos trabalhos relacionados utilizados nesta pesquisa. Ele teve como plataforma de sistema operacional o Microsoft Windows 10.

O computador utilizado para simulação teve a configuração vista na tabela 4.

Tabela 4 – Configuração do computador utilizado para simulação

CPU	Intel Xeon CPU ES-2640 v4 @ 2.40GHz
Memória	32GB DDR4

Os códigos utilizados estão disponíveis no repositório:

<https://bitbucket.org/patriciaspaula/dissertacao/src>.

5.2 Cenários utilizados para simulação

Cada cenário possui um conjunto de parâmetros, comum a todos, que permite a comparação dos algoritmos. São eles:

- (a) Espaço de busca em forma de grid, onde cada célula equivale a um espaço de 1m x 1m;
- (b) Alvo com posição fixa iniciada aleatoriamente, dentro de um espaço de grid com dimensões variáveis de acordo com o cenário de simulação;
- (c) Quantidade de 10 UAVs nos primeiros quatro cenários, variando para 5 (cinco) UAVs no cenário cinco e 40 UAVs no cenário seis;
- (d) Distância de visão para identificação do alvo em até 80m;
- (e) Velocidade de deslocamento dos UAVs em até 20m/s;
- (f) O máximo de iterações foi definido em 1500.

Foram definidos os seguintes cenários para as simulações:

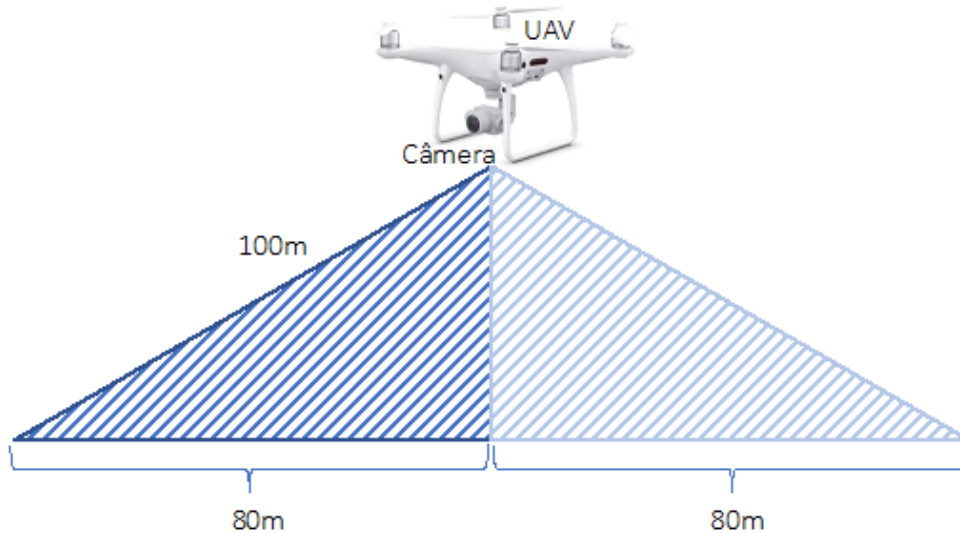
1. Tamanho da grid em 200x200, com 10 UAVs;
2. Tamanho da grid em 500x500, com 10 UAVs;
3. Tamanho da grid em 750x750, com 10 UAVs;
4. Tamanho da grid em 1000x1000, com 10 UAVs;
5. Tamanho da grid em 1000x1000, com 5 UAVs;
6. Tamanho da grid em 1000x1000, com 40 UAVs.

A dimensão máxima, 1000x1000, equivale a uma área real de 100 hectares.

A distância de visão adotada para a identificação do alvo é de até 80m, da câmera do UAV ao alvo. Segundo (YOKOYAMA *et al.*, 2014), há mais de 99% de precisão de validação para distâncias de até 100m em câmeras com resolução de 1920x1080. A câmera do modelo considerado (DJI Phantom 4 Pro) é de 4K, o que supre com folga esse requisito. A área de visualização fica similar ao visto na da figura 10. Sendo 100m a distância da câmera até o ponto

máximo de visão e 80m a distância do raio da projeção do UAV no solo ao ponto máximo de visão. Dessa forma, o alvo deve estar dentro dessa região para ser considerado encontrado por um UAV.

Figura 10 – Visualização da câmera



O máximo de iterações definido em 1500, respeita o tempo de vida da bateria do modelo usado como referência, o DJI Phantom 4 Pro, o qual percorre até 20m/s, durante até 30 minutos, totalizando uma distância percorrida máxima de 36000m. Assim, as 1500 iterações, usadas na simulação, geram um valor de distância percorrida de até 30000m, o que é inferior ao limite de bateria do UAV.

5.3 Análise de resultados

Foram feitas 30 (trinta) simulações para cada cenário. Nos primeiros 4 (quatro) cenários, foram executados o PP, o BA e o PSO. Nos cenários 5 (cinco) e 6 (seis) foram executados o BA e o PSO; nesses cenários, variou-se a quantidade de UAVs utilizada para 5 (cinco) e 40 (quarenta), respectivamente. No cenário 4 (quatro), para efeito de comparação, foi executado também o BA em sua versão original, sem a alteração de movimentação.

A seguir são exibidos os resultados das simulações, valores e gráficos comparativos.

Nas tabelas 5 e 6, tem-se a taxa de sucesso da simulação dos cenários 1 e 2. Seus gráficos podem ser visto nas figuras 11 e 12. Os gráficos relacionados a taxa de sucesso de localização indicam o algoritmo no eixo x e o percentual de sucesso no eixo y.

Tabela 5 – Percentual de sucesso cenário 1

PP	100,00
BA	100,00
PSO	100,00

Tabela 6 – Percentual de sucesso cenário 2

PP	100,00
BA	100,00
PSO	90,00

No cenário 1, os alvos foram localizados em 100% dos casos.

Figura 11 – Percentual de sucesso do cenário 1

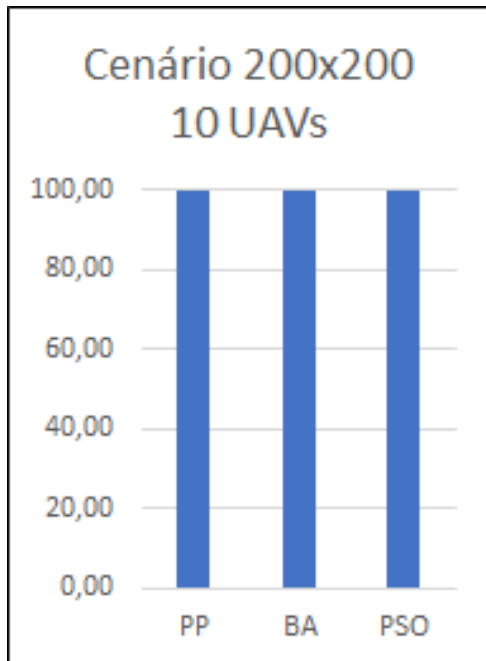
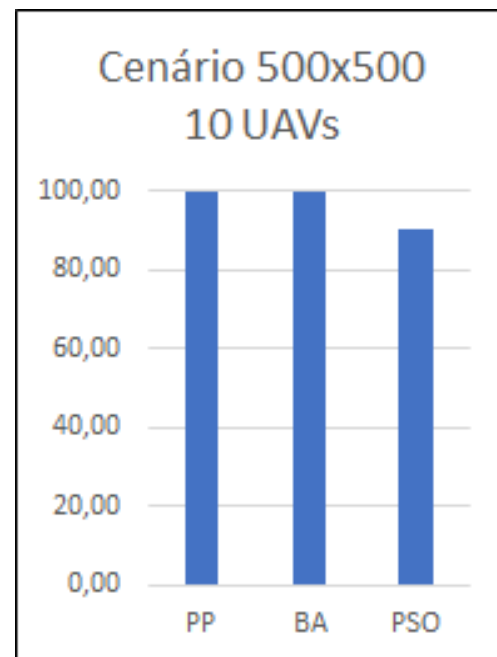


Figura 12 – Percentual de sucesso do cenário 2



No segundo cenário, o BA, junto com o PP, apresentam o melhor resultado, com 100% de sucesso. O PSO teve 90% dos alvos localizados.

Nas tabelas 7 e 8, tem-se a taxa de sucesso da simulação dos cenários 3 e 4. Seus gráficos podem ser visto nas figuras 13 e 14.

Tabela 7 – Percentual de sucesso cenário 3

PP	83,33
BA	100,00
PSO	46,67

Tabela 8 – Percentual de sucesso cenário 4

PP	43,33
BA	90,00
PSO	36,67

No terceiro cenário, o único a apresentar 100% de sucesso foi o BA, seguido pelo PP e pelo PSO. No quarto cenário, com a maior grid, nenhum teve sucesso em 100% dos casos, no entanto, o BA ainda se apresenta como melhor opção dentre os 3 (três), com uma taxa de sucesso de 90% na localização dos alvos.

Figura 13 – Percentual de sucesso do cenário 3

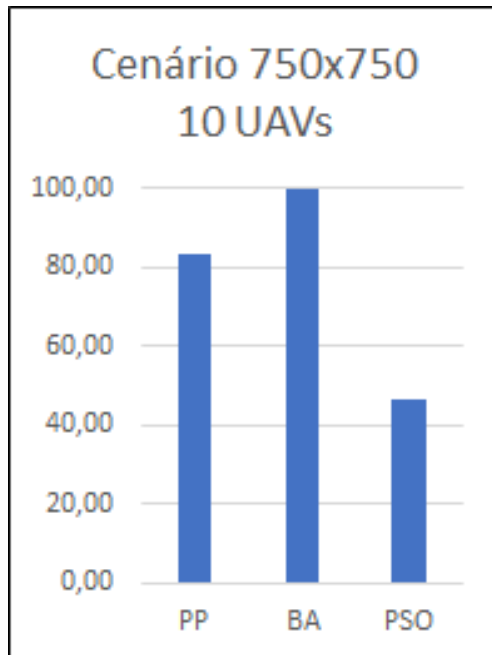
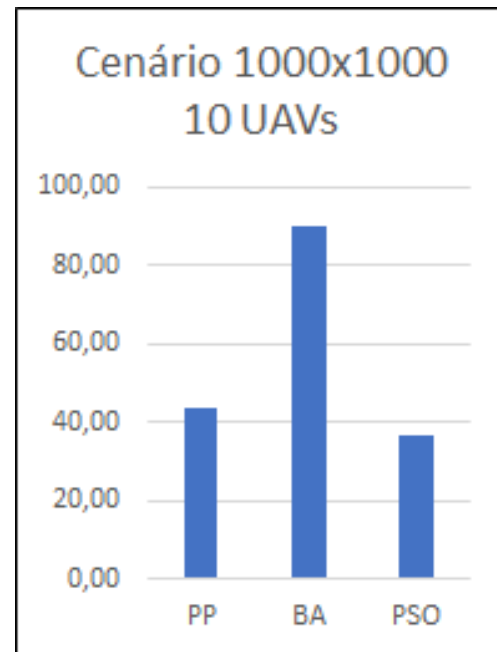
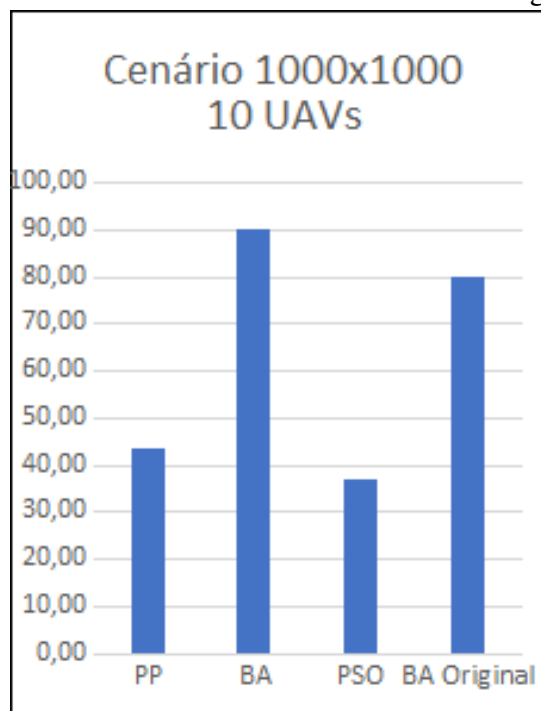


Figura 14 – Percentual de sucesso do cenário 4



No cenário 4 (quatro), foi comparado o algoritmo BA utilizado nesse trabalho, com alteração na movimentação, com o BA original. Sua versão original apresentou um percentual de sucesso de 80%, conforme pode ser visto na figura 15, sendo menos eficaz que a versão com movimentação alterada utilizada neste trabalho.

Figura 15 – Percentual de sucesso do cenário 4 incluindo o BA original



Nas tabelas 9 e 10, tem-se o percentual de sucesso da simulação dos cenários 5 e 6, com o BA e o PSO. Seus gráficos podem ser visto nas figuras 16 e 17.

Tabela 9 – Percentual de sucesso do cenário 5

BA	83,33
PSO	36,67

Tabela 10 – Percentual de sucesso do cenário 6

BA	100,00
PSO	50,00

Pode-se perceber que a alteração na quantidade de UAVs utilizada na simulação foi mais significativa quando aplicada ao BA, que voltou a alcançar 100% de eficácia.

Figura 16 – Percentual de sucesso do cenário 5

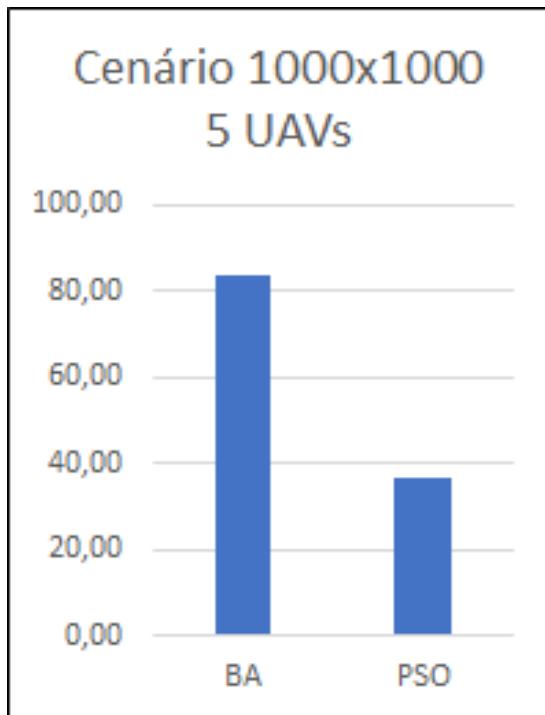
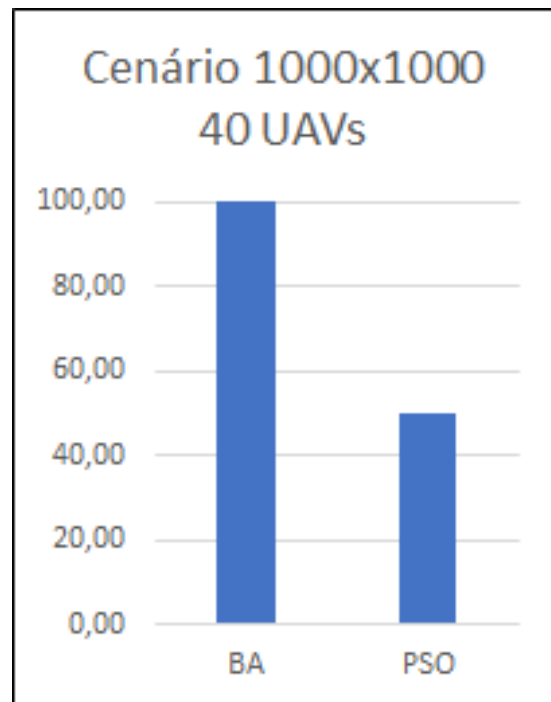


Figura 17 – Percentual de sucesso do cenário 6



Nas figuras 18 e 19, pode-se visualizar os gráficos resumidos sobre o percentual de sucesso da busca quanto ao tamanho da grid e quanto à variação na quantidade de UAVs.

Na figura 18, temos o gráfico comparativo referente aos cenários 1 (um) a 3 (três). Na figura 19, temos o gráfico comparativo dos cenários 4 (quatro) a 6 (seis). No eixo y, tem-se o percentual de sucesso e no eixo x o tamanho da grid percorrida e a quantidade de UAVs.

Analisando comparativamente os resultados, pode-se perceber que o algoritmo BA tende a se sobressair à medida que a área percorrida aumenta, sendo possível atingir 100% de eficácia aumentando a quantidade de UAVs.

Figura 18 – Percentual de sucesso por tamanho da grid

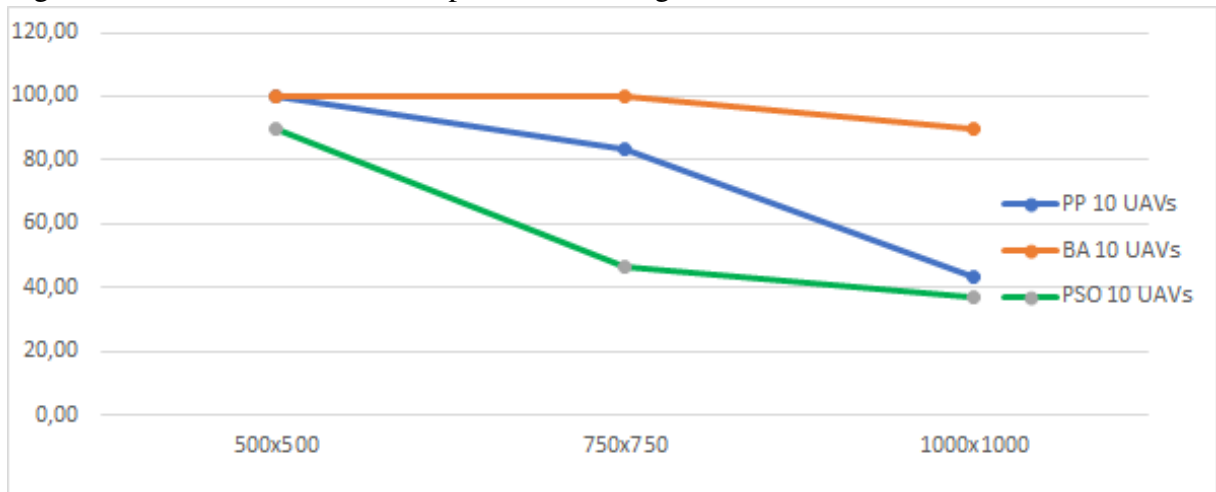
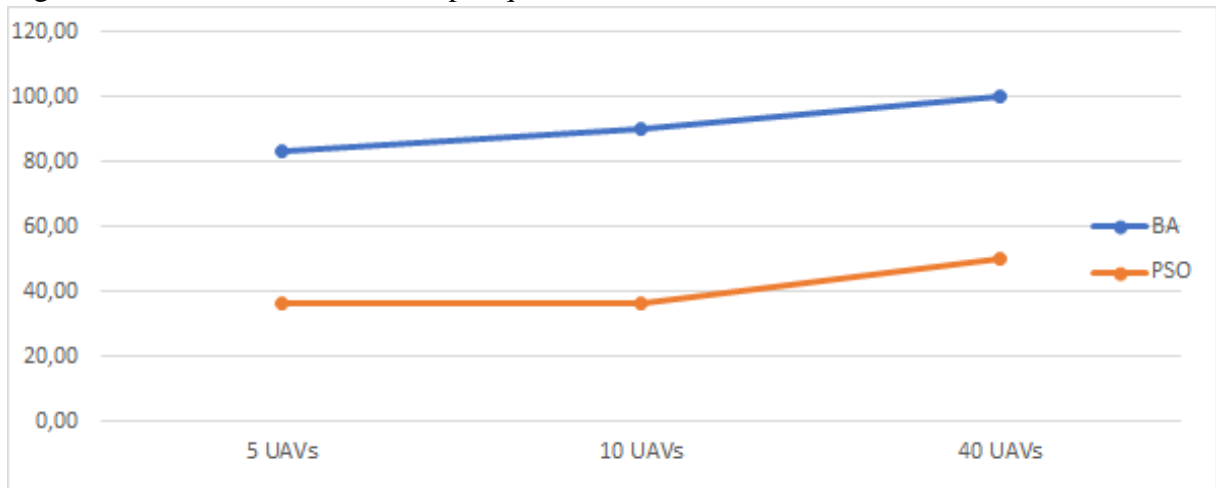


Figura 19 – Percentual de sucesso por quantidade de UAVs



A margem de erro calculada para os cenários 4 (quatro), 5 (cinco) e 6 (seis), pode ser vista nas tabelas 11, 12 e 13.

Tabela 11 – Margem de erro do cenário 4

PP	8,52%
BA	3,58%
PSO	9,01%

Tabela 12 – Margem de erro do cenário 5

BA	4,62%
PSO	9,01%

Tabela 13 – Margem de erro do cenário 6

BA	0,00%
PSO	8,00%

O valor da margem de erro foi calculado considerando um nível de confiabilidade de 95%. Margem de erro é o intervalo de valores abaixo e acima da estatística da amostra com um certo nível de confiabilidade. Ela permite identificar qual a incerteza de uma pesquisa.

Além do percentual de sucesso na localização dos alvos, tem-se nas tabelas e figuras a seguir a informação acerca do espaço médio percorrido pelos UAVs em cada simulação até a

localização do alvo ou até as 1500 iterações serem realizadas, o que acontecer primeiro. Temos também a velocidade média. Com isso, é possível calcular o equivalente ao tempo real de busca, como será visto adiante.

Nas tabelas 14 e 15 temos as informações de espaço médio dos cenários 1 e 2.

Tabela 14 – Distância média percorrida no cenário 1

PP	630,33
BA	67,40
PSO	111,11

Tabela 15 – Distância média percorrida no cenário 2

PP	9973,00
BA	400,77
PSO	8807,74

Nas tabelas 16 e 17 temos as informações de espaço médio percorrido nos cenários 3 e 4.

Tabela 16 – Distância média percorrida no cenário 3

PP	13376,00
BA	1009,44
PSO	21134,38

Tabela 17 – Distância média percorrida no cenário 4

PP	22851,67
BA	3894,43
PSO	22018,02

Conforme dito anteriormente, no cenário 4 foi comparado o algoritmo BA utilizado nesse trabalho, com alteração na movimentação, com o BA original. Em sua versão original, a distância média percorrida até os UAVs localizarem o alvo foi de 12557,79m, contra 3894,43m da versão com movimentação alterada utilizada neste trabalho.

Nas tabelas 18 e 19 temos as informações de espaço médio percorrido referente aos cenários 5 e 6.

Tabela 18 – Distância média percorrida no cenário 5

BA	6398,63
PSO	22004,54

Tabela 19 – Distância média percorrida no cenário 6

BA	671,40
PSO	18818,17

Nas figuras 20 e 21, temos, respectivamente, o comparativo da distância média percorrida por tamanho da grid e por quantidade de UAVs utilizada na simulação.

Pode-se perceber nos dois gráficos, que o BA percorre uma menor distância até atingir seu objetivo.

Figura 20 – Distância média percorrida por tamanho da grid

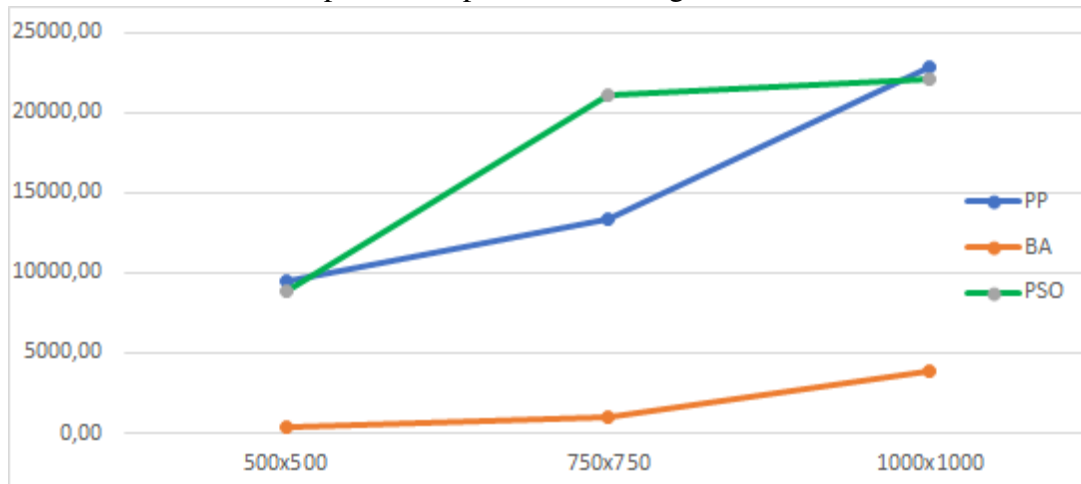
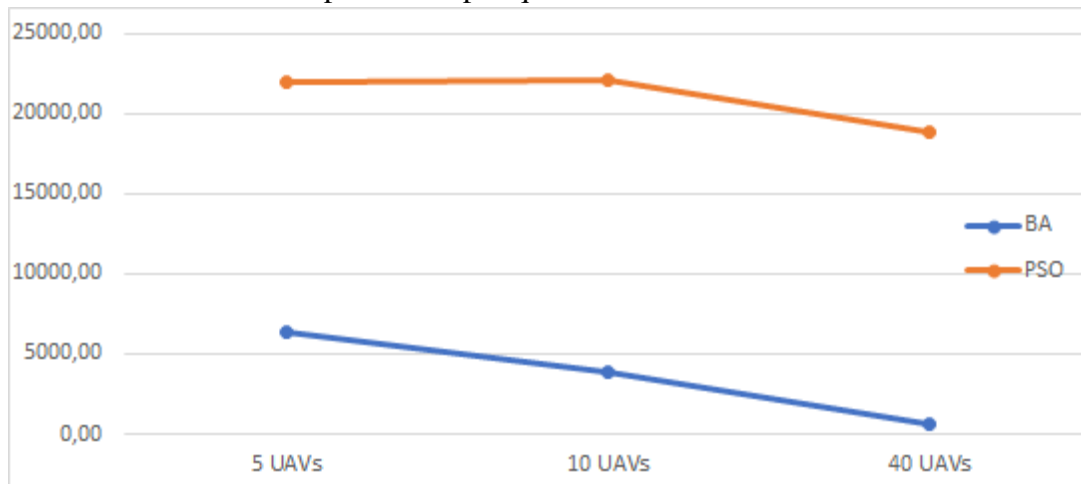


Figura 21 – Distância média percorrida por quantidade de UAVs



Nas tabelas seguintes, 20, 21, 22, 23, 24 e 25, tem-se a informação de velocidade média em cada cenário e algoritmo.

Tabela 20 – Velocidade média no cenário 1

PP	19,16
BA	16,18
PSO	19,49

Tabela 21 – Velocidade média no cenário 2

PP	19,63
BA	16,01
PSO	19,65

Tabela 22 – Velocidade média no cenário 3

PP	19,50
BA	16,63
PSO	19,70

Tabela 23 – Velocidade média no cenário 4

PP	19,81
BA	18,59
PSO	19,74
BA Original	19,97

Tabela 24 – Velocidade média no cenário 5

BA	19,36
PSO	19,71

Tabela 25 – Velocidade média no cenário 6

BA	14,73
PSO	19,75

Vê-se que as velocidades ficaram dentro da restrição estabelecida, que é de, no máximo, 20m/s. Ela não é uniforme pois muda devido ao cálculo feito em cada algoritmo, sendo calculada dentro de uma margem e restrita a um valor máximo.

Tendo em mãos a distância média percorrida e a velocidade média dos UAVs nas simulações, podemos calcular o equivalente ao tempo de busca em cada cenário para ter uma ideia da eficiência de cada algoritmo em cada um dos cenários propostos. As tabelas 26, 27, 28, 29, 30 e 31 mostram o tempo médio calculado, em segundos, para cada cenário.

Tabela 26 – Tempo calculado para o cenário 1

PP	32,90
BA	4,17
PSO	5,70

Tabela 27 – Tempo calculado para o cenário 2

PP	482,63
BA	25,03
PSO	448,23

Figura 22 – Tempo calculado para o cenário 1

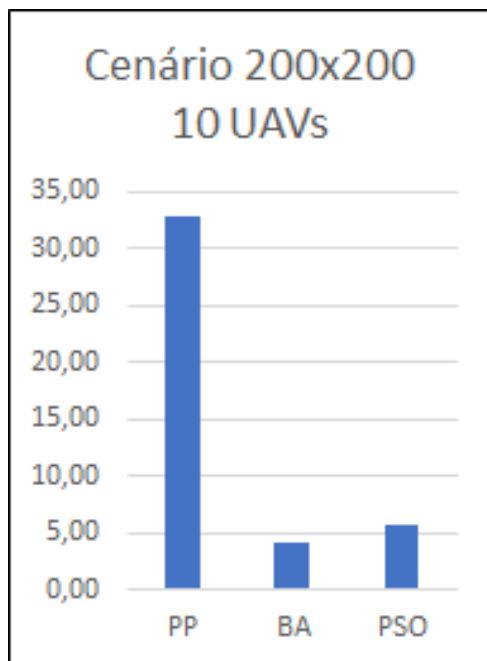
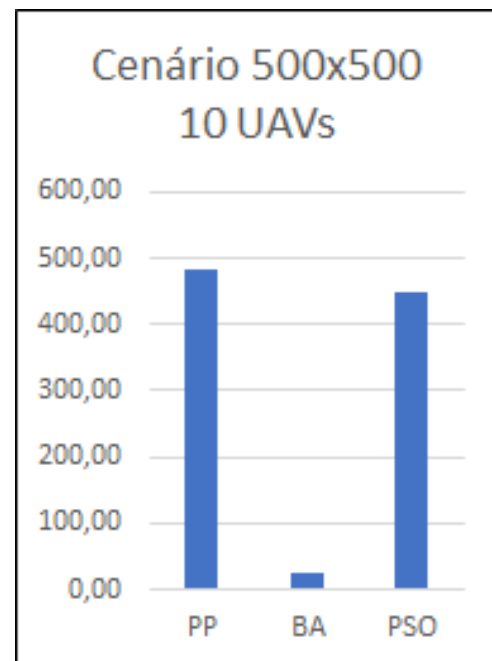


Figura 23 – Tempo calculado para o cenário 2



Observando os gráficos 22, 23, 24, 25, 26 e 27, fica mais clara a diferença do tempo entre cada um.

Tabela 28 – Tempo calculado para o cenário 3

PP	685,93
BA	60,70
PSO	1072,73

Tabela 29 – Tempo calculado para o cenário 4

PP	1153,67
BA	206,17
PSO	1115,47
BA Original	628,93

Figura 24 – Tempo calculado para o cenário 3

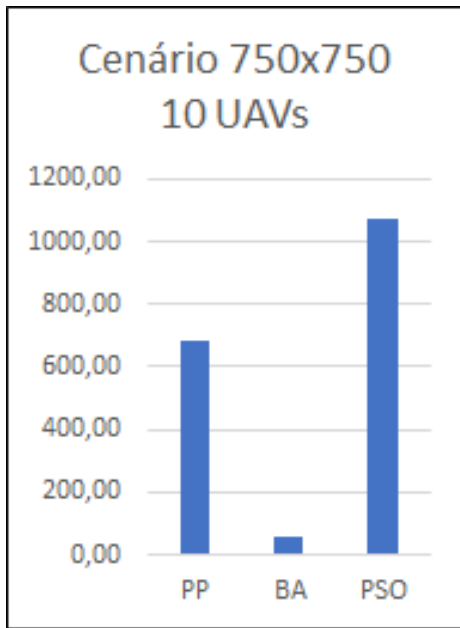


Figura 25 – Tempo calculado para o cenário 4

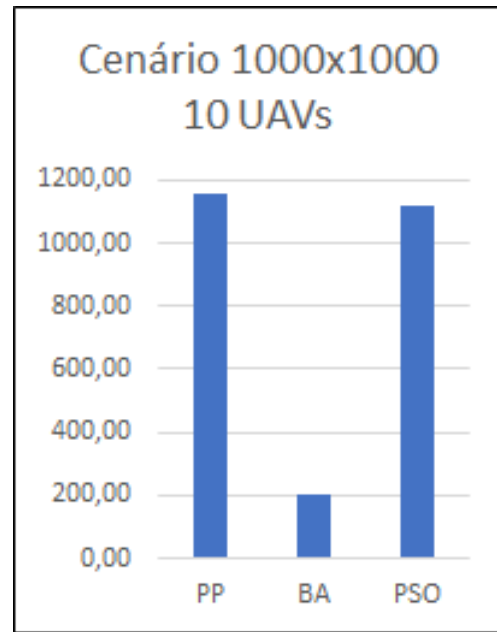


Figura 26 – Tempo calculado para o cenário 5

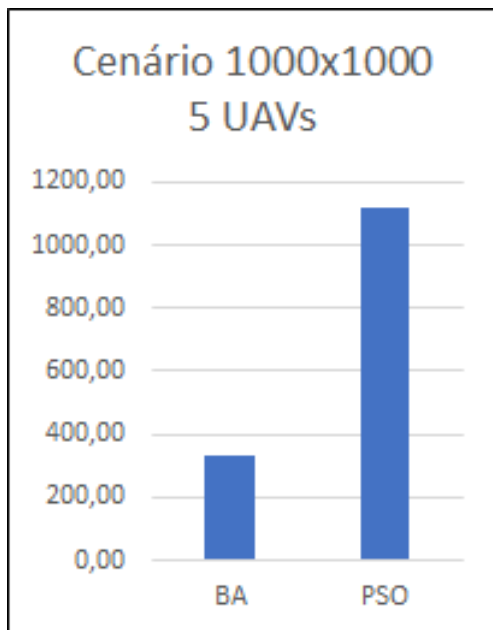


Figura 27 – Tempo calculado para o cenário 6

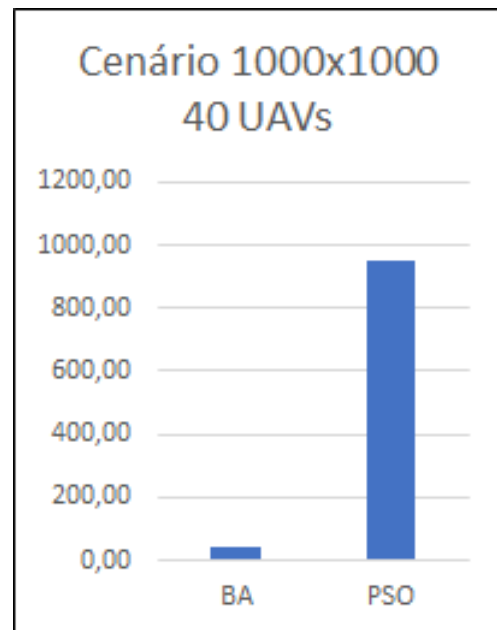


Tabela 30 – Tempo calculado para o cenário 5

BA	330,50
PSO	1116,18

Tabela 31 – Tempo calculado para o cenário 6

BA	45,59
PSO	952,83

É importante lembrar aqui que cada UAV se move próximo dos 20m/s, tendo um raio de visão de até 80m para realizar a localização dos alvos.

Com base nas informações mostradas até aqui, o BA com movimentação alterada apresentou uma melhor eficiência na busca.

No capítulo seguinte, serão discutidas as conclusões finais deste trabalho.

6 CONCLUSÕES E TRABALHOS FUTUROS

A proposta do presente trabalho vai além dos objetivos traçados nas simulações dos algoritmos utilizados. Ela parte da questão se é possível utilizar UAVs com comportamento bioinspirado, respeitando certas restrições reais dos UAVs, para solucionar um problema de busca e identificação de um alvo de posição desconhecida em uma dada área. Note que esse alvo poderia ser, em uma aplicação no mundo real, um objeto ou pessoa o qual se precisa localizar. Sendo assim, a contribuição do presente trabalho visa ajudar a melhorar o processo de buscas que hoje são feitas por pessoas ou drones guiados remotamente.

Em relação aos objetivos definidos no presente trabalho, pode-se afirmar que se mostrou a eficácia dos algoritmos simulados para aplicação em UAVs na busca de um alvo fixo, de localização desconhecida, em uma dada área.

Da mesma forma, os objetivos específicos obtiveram sucesso, haja vista que os seguintes objetivos foram alcançados:

- Comparar algoritmos bioinspirados para a busca do alvo realizada por UAVs;
- Verificar a possibilidade de alterar um algoritmo bioinspirado para um melhor resultado da busca e localização;
- Analisar dados, colhidos a partir das simulações, que levem a uma solução viável para a melhoria do uso de UAVs para sensoriamento e busca em ambientes abertos ou de difícil acesso, para validação da solução proposta.

Por fim, é importante ressaltar que as simulações partiram de parâmetros de um drone real, o que possibilitou ter cenários mais próximos da realidade.

No decorrer do trabalho, foram identificados alguns problemas que interferiram diretamente nos resultados analisados. O algoritmo inicialmente trabalhado foi apenas o PSO, o número de UAVS era menor (5 e não 10 no cenário inicial) e o máximo de iterações também era menor. Com isso, a taxa de localização do alvo estava em cerca de 65% de acerto, contra 90% do PSO no cenário equivalente, isso é, o cenário 2. Isso pode ser visto no artigo (PAULA *et al.*, 2016). Utilizando o BA, a taxa de sucesso na localização subiu para 100% nesse cenário, ficou em 90% com tamanho da grid ampliado e conseguiu, novamente, 100% quando o número de UAVs subiu para 40, como pode ser visto no cenário 6.

6.1 Trabalhos futuros

A fim de verificar o tempo de processamento em relação a outros algoritmos bioinspirados, seria interessante aplicar os testes no Algoritmo *Firefly* e no *Wolf Search*, ambos com certa similaridade com os estudados no presente trabalho.

Além disso, poderia ser útil realizar uma medição de gasto energético, considerando a comunicação entre os UAVs, tanto de forma distribuída quanto localizada, com um nó coordenador. Isso iria ao encontro do trabalho apresentado por (PALMIERI *et al.*, 2019).

Outra possibilidade de ampliação dessa pesquisa seria desenvolver a busca por múltiplos alvos, podendo ser alvos em movimento.

REFERÊNCIAS

- ALLRED, J.; HASAN, A. B.; PANICHSAKUL, S.; PISANO, W.; GRAY, P.; HUANG, J.; HAN, R.; LAWRENCE, D.; MOHSENI, K. Sensorflock: An airborne wireless sensor network of micro-air vehicles. In: **Proceedings of the 5th International Conference on Embedded Networked Sensor Systems**. New York, NY, USA: ACM, 2007. (SenSys '07), p. 117–129. ISBN 978-1-59593-763-6.
- BAO, Y.; FU, X.; GAO, X. Path planning for reconnaissance uav based on particle swarm optimization. In: **International Conference on Computational Intelligence and Natural Computing (CINe)**. Piscataway, NJ, USA: IEEE, 2010. p. 28–32.
- BASILICO, N.; CARPIN, S. Online patrolling using hierarchical spatial representations. In: **IEEE International Conference on Robotics and Automation, 2012**. Piscataway, NJ, USA: IEEE, 2012.
- BITAM, S.; BATOUCHE, M.; TALBI. A survey on bee colony algorithms. In: **IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW)**. Piscataway, NJ, USA: IEEE, 2010.
- BOUKERCHE, A.; VILLAS, L. A.; GUIDONI, D. L.; MAIA, G.; CUNHA, F. D.; UEYAMA, J.; LOUREIRO, A. A. A new solution for the time-space localization problem in wireless sensor network using uav. In: **DIVANet '13 Proceedings of the third ACM international symposium on Design and analysis of intelligent vehicular networks and applications**. New York, NY, USA: ACM, 2013.
- BROWNLEE, J. **Clever Algorithms: Nature-Inspired Programming Recipes**. [S.l.]: Creative Commons, 2011.
- CASTRO, L. **Computação Natural - Uma Jornada Ilustrada**. [S.l.]: São Paulo: Livraria da Física, 2010.
- CHEN, H.; WANG, X.; LI, Y. A survey of autonomous control for uav. In: **International Conference on Artificial Intelligence and Computational Intelligence, 2009**. [S.l.: s.n.], 2009.
- CHENGBO, Y.; YANZHE, C.; LIAN, Z.; SHUQIANG, Y. Zigbee wireless sensor network in environmental monitoring applications. In: **5th International Conference on Wireless Communications, Networking and Mobile Computing**. Piscataway, NJ, USA: IEEE, 2009.
- CORREA, S.; CERQUEIRA, R. Computação autônoma: Conceitos, infra-estruturas e soluções em sistemas distribuído. In: **27o SBRC - Livro Texto dos Minicursos**. [S.l.: s.n.], 2009. p. 151–198.
- CRUZ, J. M. de la; BESADA-PORTAS, E.; TORRE-CUBILLO, L.; ANDRES-TORO, B.; LOPEZ-OROZCO, J. A. Evolutionary path planner for uavs in realistic environments. In: **10th annual conference on Genetic and evolutionary computation, 2008**. New York, NY, USA: ACM, 2008. p. 1477–1484.
- DING, J.; BALASINGHAM, I.; BOUVRY, P. Management of overlay networks: A survey. In: **Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM '09**. Piscataway, NJ, USA: IEEE, 2009.

- DORIGO, M.; MANIEZZO, V.; COLORNI. Ant system: Optimization by a colony of cooperating agents. In: **IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B CYBERNETICS**. Piscataway, NJ, USA: IEEE, 1996. p. 29–41.
- DRESSLER, F.; AKAN, O. A survey on bio-inspired networking. *Computer Networks*, 2009.
- FAIÇAL, B. S.; COSTA, F. G.; PESSIN, G.; UEYAMA, J.; FREITAS, H.; COLOMBO, A.; FINI, P. H.; VILLAS, L.; OSÓRIO, F. S.; VARGAS, P. A.; BRAUN, T. The use of unmanned aerial vehicles and wireless sensor networks for spraying pesticides. **Journal of Systems Architecture**, v. 60, p. 393–404, 2014.
- GERKE, P.; LANGEVOORT, J.; LAGARDE, S.; BAX, L.; GROOTSWAGERS, T.; DRENTH, R.; SLIEKER, V.; VUURPIJL, L.; HASELAGER, W.; SPRINKHUIZEN-KUYPER, I.; OTTERLO, M. v.; CROON, G. d. **BioMAV: bio-inspired intelligence for autonomous flight**. 2011.
- GRADO-CAFFARO, M. A.; GRADO-CAFFARO, M.; EBRAHIMI, M. S. Some comments on wireless sensor networks for natural hazards. In: **1st International Conference on Wireless Technologies for Humanitarian Relief**. [S.l.: s.n.], 2011. p. 315–322.
- GUERRERO, E.; XIONG, H.; GAO, Q. A distributed range-free localization algorithm for wireless sensor networks based on a mobile robot. In: **International Conference on Advanced Technologies for Communications, 2009**. [S.l.: s.n.], 2009.
- IBM. **An architectural blueprint for autonomic computing**. 2005.
- INNOCENTE, M. S.; GRASSO, P. Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems. **Journal of Computational Science**, 2019.
- JAWHA, I.; MOHAMED, N.; AL-JAROUDI; ZHANG, J. . Data communication in linear wireless sensor networks using unmanned aerial vehicles. In: **International Conference on Unmanned Aircraft Systems (ICUAS)**. [S.l.: s.n.], 2013. p. 492–499.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: **IEEE International Conference on Neural Networks**. Piscataway, NJ, USA: IEEE, 1995. p. 1942–1948.
- LI, S.; SUN, X.; XU, Y. Particle swarm optimization for route planning of unmanned aerial vehicles. In: **IEEE International Conference on Information Acquisition, 2006**. [S.l.: s.n.], 2006.
- MCCUNE, R.; MADEY, G. Agent-based simulation of cooperative hunting with uavs. In: **Symposium on Agent Directed Simulation, 2013**. [S.l.: s.n.], 2013.
- MIRJALILI, S.; MIRJALILI, S. M.; YANG, X.-S. Binary bat algorithm. **Neural Comput. Appl.**, Springer-Verlag, London, UK, UK, v. 25, n. 3-4, p. 663–681, set. 2014. ISSN 0941-0643.
- NAN, G.; LI, M. Evolutionary based approaches in wireless sensor networks: A survey. In: **Fourth International Conference on Natural Computation, 2008**. [S.l.: s.n.], 2008.
- NIGAM, N.; BIENIAWSKI, S.; KROO, I.; VIAN, J. Control of multiple uavs for persistent surveillance: Algorithm and flight test results. **IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY**, v. 20, p. 1236–1251, 2012.

- PALMIERI, N.; YANG, X.; RANGO, F.; MARANO, S. Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption. **Neural Computing and Applications**, 2019.
- PAULA, P. de S.; CASTRO, M. F. de; PAILLARD, G. A. L.; SARMENTO, W. W. F. A swarm solution for a cooperative and self-organized team of uavs to search targets. In: **Proceedings of the 2016 8th Euro American Conference on Telematics and Information Systems (EATIS)**. Washington, DC, USA: IEEE Computer Society, 2016. p. 24–. ISBN 978-1-5090-2436-0.
- PHAM, D. T.; GHANBARZADEH, A.; KOÇ, E.; OTRI, S.; RAHIM, S.; ZAIDI, M. The bees algorithm – a novel tool for complex optimisation. In: **2nd International Virtual Conference on Intelligent Production Machines and Systems**. [S.l.: s.n.], 2006. p. 454–459.
- PURI, P.; SINGH, M. Survey paper on routing in delay-tolerant networks. In: **International Conference on Information Systems and Computer Networks (ISCON)**. [S.l.: s.n.], 2013.
- RIBEIRO, L. **BiO4SeL: Uma Abordagem Baseada em Colônia de Formigas para a Otimização do Tempo de Vida de Redes de Sensores Sem Fio**. 2009. Dissertação (Mestrado em Ciência da Computação) — Centro de Tecnologia, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal do Ceará, Fortaleza, 2009.
- RIBEIRO, L.; CASTRO, M. Bio4sel: A bio-inspired routing algorithm for sensor network lifetime optimization. In: **17th International Conference on Telecommunications**. Piscataway, NJ, USA: IEEE, 2010.
- SARMENTO, W. W. F.; PAULA, P. d. S.; FILHO, P. T. C. P.; PAILLARD, G. A. L.; NEUMAN, J.; BARROSO, G. C.; PEQUENO, M. C. u-lab: A ubiquitous computing based architecture to labs works using wireless sensor network and radio-frequency identification. In: **6th Euro American Conference on Telematics and Information Systems (EATIS)**. New York, NY, USA: ACM, 2012.
- SATHYARAJ, B.; JAIN, L.; FINN, A.; DRAKE, S. Multiple uavs path planning algorithms: a comparative study. **Fuzzy Optimization and Decision Making**, v. 7, p. 257–267, 2008.
- SELLKE, S.; SHROFF, N.; BAGCHI, S. Modeling and automated containment of worms. **IEEE Transactions on Dependable and Secure Computing**, 2008.
- SHOLES, E. Evolution of a uav autonomy classification taxonomy. In: **Aerospace Conference**. Piscataway, NJ, USA: IEEE, 2007.
- SOUISSI, O.; BENATITALLAH, R.; DUVIVIER, D.; ARTIBA, A.; BELANGER, N.; FEYZEAU, P. Path planning: A 2013 survey. In: **5th IESM Conference**. [S.l.: s.n.], 2013.
- TANG, R.; FONG, S.; YANG, X.-S.; DEB, S. Wolf search algorithm with efemeral memory. In: **Seventh International Conference on Digital Information Management (ICDIM)**. Piscataway, NJ, USA: IEEE, 2012. p. 165–172.
- TOSCANO, E.; BELLO, L. L. Comparative assessments of ieee 802.15.4/zigbee and 6lowpan for low-power industrial wsns in realistic scenarios. In: **9th IEEE International Workshop on Factory Communication Systems (WFCS)**. [S.l.: s.n.], 2012.
- VILLAS, L. A.; GUIDONI, D. L.; UEYAMA, J. 3d localization in wireless sensor networks using unmanned aerial vehicle. In: **12th International Symposium on Network Computing and Applications**. Piscataway, NJ, USA: IEEE, 2013. p. 135–142.

- WEI, L.; WEI, Z. Path planning of uavs swarm using ant colony system. In: **International Conference on Natural Computation, 2009**. Piscataway, NJ, USA: IEEE, 2009.
- YANG, S. **Wireless Sensor Networks: Principles, Design and Applications**. [S.l.]: Springer, 2014.
- YANG, X.-S. **Nature-Inspired Metaheuristic Algorithm**. [S.l.]: Luniver Press, 2010.
- YOKOYAMA, R. S.; KIMURA, B. Y. L.; MOREIRA, E. dos S. Secure positioning in a uav swarm using on-board stereo cameras. In: **Proceedings of the 29th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2014. (SAC '14), p. 769–774. ISBN 978-1-4503-2469-4.
- ZHANG, C.; ZHEN, Z.; WANG, D.; LI, M. Uav path planning method based on ant colony optimization. In: **Chinese Control and Decision Conference**. [S.l.: s.n.], 2010. p. 3790–3792.

APÊNDICE A – IEEE EATIS 2016

Artigo publicado no *8th Euro American Conference on Telematics and Information Systems (EATIS)*, Cartagena de Indias, Bolivar - April 27–29, 2016, COL.

A swarm solution for a cooperative and self-organized team of UAVs to search targets

Patrícia de Sousa
Paula¹

Miguel Franklin de
Castro²

Gabriel A. Louis
Paillard³

Wellington W. F.
Sarmiento⁴

Universidade Federal do Ceará, UFC
Fortaleza, CE, Brasil

¹patricia@virtual.ufc.br, ²miguel@ufc.br, ³gabriel@virtual.ufc.br, ⁴wwagner@virtual.ufc.br

ABSTRACT

The context of this research is the use of Unmanned Aerial Vehicles (UAV) to search and localize a target. The problem to be solved is the cooperative behavior of the UAV group to optimize the search. The focus of this research is the use of bioinspired algorithms to spread the aerial vehicles and the optimization of the time of search, comparing the application of Particle Swarm Optimization with others found in literature.

Keywords

Unmanned Aerial Vehicle; UAV; swarm intelligence; UAV swarm; bio-inspired.

1. INTRODUÇÃO

O avanço do uso de veículos autônomos para aplicações fora de laboratórios de pesquisa vem aumentando consideravelmente nos últimos anos. Assim, os veículos autônomos estão se inserindo aos poucos em nosso dia a dia, sendo chamados UGV (Unmanned Ground Vehicle), UAV (Unmanned Aerial Vehicle) e MAV (Micro Aerial Vehicle), dependendo da movimentação e tamanho. Os UGVs, UAVs e MAV tem relação com a computação autônoma, podendo ser parcial ou completamente autônomos, de acordo com as características levantadas no referencial teórico a ser visto na seção 2.

Uma questão fundamental no uso de UAVs com WSN (*Wireless Sensor Networks*) é o comportamento e distribuição dos UAVs (ver Figura 1) para varredura de uma área. Tal comportamento pode ser modelado por meios mais tradicionais, como Máquinas de Estado Finito (no inglês Finite State Machine - FSM) ou por abordagens inspiradas em comportamentos de seres vivos, o que é chamado de Computação Bioinspirada [1][2].

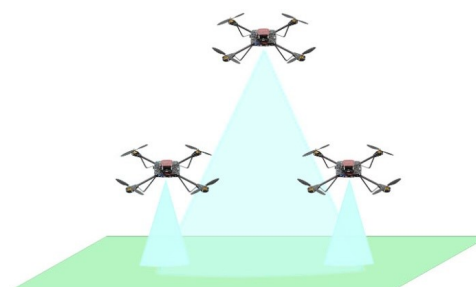


Figura 1 Ilustração de UAVs atuando em conjunto
[Fonte:<http://cabbuavphd.blogspot.com.br/2012/11/project-progress-search-patterns-and.html>]

É neste contexto que se encontra o presente trabalho, o qual propõe uma solução bioinspirada para a distribuição e comportamento de UAVs para a identificação de pessoas ou veículos em ambientes abertos e/ou de difícil acesso (e.g. florestas, rede de cavernas e montanhas).

2. FUNDAMENTAÇÃO TEÓRICA

Algumas definições e tecnologias que contextualizam a presente pesquisa são colocadas a seguir.

2.1. Veículos Aéreos Não Tripulados

Os UAVs (Unmanned Aerial Vehicles) – ou, no português, Veículos Aéreos Não Tripulados (VANTs) –, são aeronaves sem a presença de pilotos que podem ser controladas remotamente ou voar de forma autônoma de acordo com planos de voo pré-programados [3][4]. Suas principais formas de controle são via rádio remoto ou pela pré-programação do voo.

Sua versão em tamanho reduzido é chamada MAV, Micro Aerial Vehicle. A versão terrestre é chamada Unmanned Ground Vehicle, UGV, que normalmente assume a forma de carros ou robôs com deslocamento terrestre. Os UAVs levam vantagem sobre os UGVs no que diz respeito ao terreno, que podem ser íngremes e/ou com muitos obstáculos, e no maior alcance de área, lembrando que missões de maior duração vão requerer que os UAVs retornem periodicamente para reabastecimento [5].

Os UAVs tem capacidade de sensoriamento limitada e a cooperação entre um grupo de UAVs depende fortemente da comunicação entre UAVs próximos. Como exemplo, pode-se ver na Figura 2 que os nós A e F não possuem um canal direto de comunicação. Para que a informação seja enviada, deve-se seguir um caminho intermediário de comunicação, por exemplo, $A > B > C > F$. No caso dos nós