

UNIVERSIDADE FEDERAL DE GOIÁS  
INSTITUTO DE INFORMÁTICA

LUCAS SOARES RODRIGUES

**Plano de Voo Autônomo e Arquitetura  
*Drone-as-a-Service* para Cobertura de  
Redes IoT**

Goiânia  
2020



UNIVERSIDADE FEDERAL DE GOIÁS  
INSTITUTO DE INFORMÁTICA

## TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a [Lei 9.610/98](#), o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

### 1. Identificação do material bibliográfico

Dissertação      Tese

### 2. Nome completo do autor

Lucas Soares Rodrigues

### 3. Título do trabalho

Plano de Voo Autônomo e Arquitetura Drone-as-a-Service para Cobertura de Redes IoT

### 4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento  SIM      NÃO<sup>1</sup>

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:

- a) consulta ao(à) autor(a) e ao(à) orientador(a);
- b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.

O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

**Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.**



Documento assinado eletronicamente por **Antonio Carlos De Oliveira Junior, Professor do Magistério Superior**, em 02/09/2020, às 13:52, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **LUCAS SOARES RODRIGUES, Discente**, em 02/09/2020, às 17:00, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://sei.ufg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1527306** e o código CRC **4A713A5D**.



LUCAS SOARES RODRIGUES

# **Plano de Voo Autônomo e Arquitetura *Drone-as-a-Service* para Cobertura de Redes IoT**

Dissertação apresentada ao Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

**Área de concentração:** Sistemas de Computação

**Orientador:** Prof. Antonio Carlos de Oliveira Júnior

**Co-Orientador:** Prof. Kleber Vieira Cardoso

Goiânia  
2020

Ficha de identificação da obra elaborada pelo autor, através do  
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Soares Rodrigues, Lucas

Plano de Voo Autônomo e Arquitetura Drone-as-a-Service para  
Cobertura de Redes IoT [manuscrito] / Lucas Soares Rodrigues. -  
2020.

XCII, 92 f.

Orientador: Prof. Dr. Antonio Carlos de Oliveira Júnior; co  
orientador Dr. Kleber Vieira Cardoso.

Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto  
de Informática (INF), Programa de Pós-Graduação em Ciência da  
Computação, Goiânia, 2020.

Bibliografia.

1. Drones. 2. VANTs. 3. Problema de Roteamento de Veículos. 4.  
Mula de Dados. 5. IoT. I. Carlos de Oliveira Júnior, Antonio, orient. II.  
Título.

CDU 004



UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

**ATA DE DEFESA DE DISSERTAÇÃO**

Ata nº **15/2020** da sessão de Defesa de Dissertação de **Lucas Soares Rodrigues**, que confere o título de Mestre em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos quatro dias do mês de agosto de dois mil e vinte, a partir das dez horas, via sistema de webconferência da RNP, realizou-se a sessão pública de Defesa de Dissertação intitulada “Plano de Voo Autônomo e Arquitetura Drone-as-a-Service para Cobertura de Redes IoT”. Os trabalhos foram instalados pelo Orientador, Professor Doutor Antonio Carlos de Oliveira Júnior (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Kleber Vieira Cardoso (INF/UFG), coorientador; Professor Doutor Cristiano Bonato Both (UNISINOS), membro titular externo; e Professor Doutor Waldir Aranha Moreira Júnior (Fraunhofer Portugal AICOS), membro titular externo. A realização da banca ocorreu per meio de videoconferência, em atendimento à recomendação de suspensão das atividades presenciais na UFG emitida pelo Comitê UFG para o Gerenciamento da Crise COVID-19, bem como à recomendação de isolamento social da Organização Mundial de Saúde e do Ministério da Saúde para enfrentamento da emergência de saúde pública decorrente do novo coronavírus. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Antonio Carlos de Oliveira Júnior, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos quatro dias do mês de agosto de dois mil e vinte.

## TÍTULO SUGERIDO PELA BANCA



Documento assinado eletronicamente por **CRISTIANO BONATO BOTH, Usuário Externo**, em 04/08/2020, às 12:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Antonio Carlos De Oliveira Junior, Professor do Magistério Superior**, em 04/08/2020, às 12:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Kleber Vieira Cardoso, Professor do Magistério Superior**, em 04/08/2020, às 12:26, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **LUCAS SOARES RODRIGUES, Discente**, em 04/08/2020, às 12:27, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).

Documento assinado eletronicamente por **Waldir Aranha Moreira Junior, Usuário Externo**, em 04/08/2020, às 12:27, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do



[Decreto nº 8.539, de 8 de outubro de 2015.](#)

---



A autenticidade deste documento pode ser conferida no site [https://sei.ufg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1470335** e o código CRC **A10311DC**.

---

**Referência:** Processo nº 23070.030525/2020-01

SEI nº 1470335

---

## Resumo

---

Rodrigues, Lucas Soares. **Plano de Voo Autônomo e Arquitetura *Drone-as-a-Service* para Cobertura de Redes IoT**. Goiânia, 2020. 92p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

Este trabalho é uma abordagem para o problema do roteamento de veículos com foco em *drones* como coletores de dados numa rede de sensores IoT. A dissertação apresenta um modelo para apenas uma aeronave bem como para múltiplas. Também é abordada uma técnica de clusterização que amplia a abrangência do número de sensores visitados por VANTs. O plano de voo gerado a partir do modelo tem foco em prevenção de panes por falta de carga na bateria. Dessa forma com objetivo de maximizar o número de nós que serão visitados várias restrições diferem deste modelo dos modelos convencionais de roteamento de veículos. Além da autonomia de voo do *drone* outro aspecto de limitação é considerado: o armazenamento de dados. São utilizadas fórmulas de cálculo do consumo energético dos *drones* baseadas nas características de aerodinâmica desta aeronave. São replicadas simulações para observar o comportamento do algoritmo na geração das rotas e é realizada avaliação do modelo utilizando uma métrica de confiabilidade. O trabalho também possui a proposta de uma arquitetura com propósito de compartilhamento dessa função de coleta de dados entre várias aplicações externas, esta arquitetura baseada em serviços e componentes na nuvem aplica o conceito de *Drone-as-a-Service*. Também é apresentado um ambiente de simulação para visualizar os voos gerados por algoritmos de roteamento com a conversão do plano de voo em comandos MAVlink, e também é definido uma ferramenta para a geração dos planos de voos e gerenciamento dos sensores de uma determinada área.

### Palavras-chave

<Drones, VANTs, Problema de Roteamento de Veículos, Mula de Dados, IoT >

---

## Abstract

---

Rodrigues, Lucas Soares. <**Autonomous Flight Plan and Architecture Drone-as-a-Service for IoT Network Coverage**>. Goiânia, 2020. 92p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

This work is an approach to the vehicle routing problem with a focus on drones as data collectors in a network of IoT sensors. The dissertation presents a model for only one aircraft as well as for multiple ones. A clustering technique that extends the scope of the number of sensors visited by UAVs is also addressed. The flight plan generated from the model focuses on the prevention of breakdowns due to lack of battery charge. Thus, in order to maximize the number of nodes that will be visited, various restrictions differ from this model from conventional vehicle routing models. In addition to the drone flight autonomy, another limitation aspect is considered: data storage. Math Formulas for calculating the energy consumption of drones based on the aerodynamic characteristics of this aircraft are used. Simulations are replicated to observe the behavior of the algorithm in the generation of routes and the model is evaluated using a reliability metric. The work also proposes an architecture with the purpose of sharing this data collection function between various external applications, this architecture based on services and components in the cloud also applies the concept of Drone-as-a-Service. A simulation environment is also presented to visualize the flights generated by routing algorithms with the conversion of the flight plan into MAVlink commands, and a tool is also defined for the generation of flight plans and management of sensors in a given area.

### Keywords

<Drones, UAVs, Vehicle Routing Problem, Data Mule, IoT >

---

# Sumário

---

<b>1</b>	<b>Introdução</b>	<b>7</b>
1.1	Contextualização do Problema de Pesquisa	7
1.2	Objetivos	10
1.2.1	Objetivo Geral	10
1.2.2	Objetivos Específicos	11
1.3	Contribuições	12
1.4	Estrutura do Documento	12
<b>2</b>	<b>Background e Trabalhos Relacionados</b>	<b>14</b>
2.1	IoT ( <i>Internet of Things</i> )	14
2.1.1	Coleta de Dados	15
2.2	Cenários Inteligentes	16
2.3	Veículos Aéreos Não Tripulados (VANTs)	16
2.4	Problema do Caixeiro Viajante	18
2.5	Problema de Roteamento de Veículos	19
2.6	Trabalhos Relacionados	20
2.6.1	Aplicações Com VANTs nas Redes de Sensores Sem Fio e no Contexto IoT	20
2.6.2	Serviços de Compartilhamento de Drone na Nuvem	22
2.6.3	Problema do Roteamento de Veículos com VANTs	23
<b>3</b>	<b>Modelos de Otimização Propostos e Avaliação de Resultados</b>	<b>27</b>
3.1	Plano de voo com único VANT e autonomia limitada	28
3.1.1	Abordagem com Localizações 2D e Gasto Energético Simplificado	28
	Rota resultante do Plano de Voo em 2D	31
3.1.2	Abordagem com Localizações 3D	37
	Cálculo da Autonomia e do Gasto Energético do Drone	38
	Formulação do Problema do PCV com Localizações 3D	41
	Rota resultante do Plano de Voo em 3D	41
3.2	Clusterização dos Dispositivos IoT	44
3.3	Plano de voo com múltiplos VANTs com capacidade de armazenamento	47
3.3.1	Formulação do problema	47
	Cálculo do <i>hovering</i> do VANT	49
	Eficiência Energética	51
3.3.2	Avaliação	51
	Desempenho	57

4	Arquitetura <i>Drone-as-a-Service</i> e Plataforma de Serviços	<b>61</b>
4.1	Cenário	61
4.2	Arquitetura	62
4.2.1	Data Offloading	64
4.2.2	Diagrama de Classes	65
4.3	Implementação e Avaliação	67
4.3.1	Funções Básicas no Componente <i>Services</i>	67
4.3.2	Interação <i>Services</i> x <i>Optimization</i>	67
5	Ferramentas de Apoio a Voos Reais e Simulados com gerenciamento de dispositivos IoT	<b>70</b>
5.1	Motivações	70
5.2	Ambiente de Simulação e Ferramentas	72
5.3	Web Tool	74
5.3.1	Estação de controle de solo (ECS)	75
	<i>Mission planner</i> para VANT como mula de dados	75
5.3.2	Protocolo MAVLink	76
5.3.3	Uso de Algoritmos de Meta-Heurística	77
	Busca Tabu	78
	Algoritmo Genético	79
5.3.4	Interface de Gerenciamento	81
6	Conclusão	<b>84</b>
6.1	Trabalhos Futuros	85
	Referências Bibliográficas	<b>86</b>

## Introdução

---

Embora já seja amplamente utilizado em uma série de segmentos da economia, como agronegócios, segurança e infraestrutura, o mercado de *drones* no Brasil está em estágio incipiente [66]. Os preços elevados continuam sendo um dos grandes obstáculos para a adesão a este recurso na indústria. Um *drone* para fins comerciais, segundo uma pesquisa do Pitt Street Research, em parceria com a consultoria Gartner, costuma exigir um maior investimento - em torno de US\$ 10 mil - por conta de seu tamanho e especificações técnicas. Enquanto para uso doméstico, em média, US\$ 1,2 mil, valor 88% menor. [65]

Para o Brasil há ainda outro grande entrave: regulamentação. A Anac (Agência Nacional de Aviação Civil), que rege o setor, reconhece que dentre os estimados 100 mil *drones* existentes no país, há apenas 70 mil aeronaves portáteis cadastradas em seu sistema, e destas, 35% são registradas para uso profissional. Ou seja, pelo menos 30 mil *drones* voando pelo país sem a chancela da entidade não sendo permitidos, de forma ainda mais restritiva, seu uso comercial. [66]

No contexto regional e nacional em que este trabalho é inserido é evidente que a tecnologia dos veículos aéreos não tripulados ainda é emergente quanto a totalidade das formas e aspectos possíveis de serem explorados [65]. Devido ao pouco conhecimento acerca da tecnologia bem como de seus benefícios, estas aeronaves permanecem um ativo escasso em disponibilidade na maioria das organizações nacionais. Muitas empresas ou instituições de pesquisa ainda não fizeram um investimento massivo na compra destas aeronaves, devido às dificuldades acima elencadas, fazendo com que haja uma quantidade limitada deste veículos disponíveis para uso, seja para pesquisa ou operação.

### 1.1 Contextualização do Problema de Pesquisa

Ao se tratar de infraestrutura de comunicação os veículos aéreos não tripulados (VANTs) podem ser úteis para realizar coleta de dados em uma rede de dispositivos IoT (*Internet of Things*). A coleta de dados de nós sensores estáticos é uma aplicação típica para redes de sensores da Internet das Coisas. Dado que os sensores estão espalhados

aleatoriamente uma área abrangente, não é fácil obter informações de todos sensores, se não estiverem totalmente conectados. Os sensores utilizados numa rede IoT geralmente possui baixa autonomia energética, portanto alguns não possuem conectividade móvel, fazendo com que a coleta de dados seja realizada de forma alternativa à estação celular, seja por veículo móvel terrestre ou aéreo. Os VANTs são promissores veículos para coleta de dados em redes de sensores devido à sua habilidades de comunicação direta entre o VANT e o sensor [75].

A coleta de dados pode ser aplicada numa área específica, como um Campus Inteligente, que pode possuir um parque de dispositivos IoT implantados com a função de coletar os mais variados tipos de dados possíveis relativos ao funcionamento deste. Considerando um campus universitário, por exemplo, pode-se implantar dispositivos IoT para auxiliar a coleta de lixo, verificar vagas de estacionamento disponíveis no campus e o funcionamento de recursos do campus, como ar condicionado em salas de aula, dentre outros. Um exemplo deste cenário é demonstrado na Figura 1.1, porém neste a trajetória não da aeronave foi necessariamente a mais eficiente dentro das possibilidades disponíveis.

Além destes veículos terem autonomia limitada em si, as mesmas unidades são compartilhadas entre muitos usuários, fazendo com que recorrentemente suas baterias sejam encontradas com pouca carga restante. Um VANT não pode correr risco de queda durante o voo, o que uma eventual falta de bateria provocaria, acarretando no possível prejuízo da perda deste equipamento. O uso amplo e compartilhado por vários usuários, advindos de diversas origens, também acarreta no preenchimento da capacidade de memória interna do veículo, quando é executada, por exemplo, a instalação de diversos *softwares*, armazenamento de arquivos coletados de imagem como fotos e vídeos, entre outros usos da memória com variados propósitos.

Em um bom número de exemplares a memória fornecida nestes veículos utiliza a tecnologia *flash*, que possui um peso ideal para ser carregado por este tipo de aeronave. Porém, esta tecnologia possui alto valor para aquisição e por isso é fornecida com pequeno espaço, facilmente preenchido em sua totalidade. Para viabilizar a adoção em larga escala desta tecnologia, nos mais variados serviços e atividades, uma boa prática adotada é o planejamento dos voos destes veículos considerando a limitação de autonomia existente no momento da realização do voo, e também a memória disponível para armazenamento de dados considerando o uso dos VANTs como coletor de dados gerados por sensores. Ao realizar o planejamento do voo a ser executado pela aeronave é possível inserir um plano de missão no veículo que realiza o plano de forma autônoma total ou supervisionada.

A consideração dos atributos da autonomia de múltiplos VANTs bem como *buffer* dos sensores de maneira integrada torna o problema muito mais desafiador e diferente da literatura clássica de roteamento de veículos com capacidade aplicado à VANTs. É

considerada também a capacidade de armazenamento de dados nos *drones* como um fator limitante, pois considerando que estes veículos muitas vezes são compartilhados para variadas atividades, o total de armazenamento disponível pode inviabilizar uma missão. Além desses fatores limitantes também é necessário repensar o modelo já existente na literatura para que se adapte ao novo objetivo de maximizar os pontos visitados.

Muitos aspectos que acabam derivando variáveis para serem consideradas em modelos de otimização se relacionam diretamente com a necessidade de eficiência também derivada da escassez de recursos. A notável necessidade de compartilhamento dos recursos que os *drones* possuem também se desdobra no estabelecimento de um meio único, virtualizado de consumo dos recursos de coleta de dados providos pela aeronave. Desse modo, um arquitetura de compartilhamento, se mostra também não só viável como fundamental, para virtualizar, dar transparência e gerenciamento das funções de comunicação que o VANT oferece.

De modo a aferir o comportamento da aeronave numa situação de voo autônomo é necessário projetar um cenário onde seja possível realizar voos experimentais utilizando os planos de voos gerados pelo modelo e outros da literatura. Portanto, é possível identificar o ganho do uso dessa tecnologia em cenários reais da indústria como por exemplo na Agricultura de Precisão (AP). Sondas (*probes*) e sensores são usados para identificar a qualidade do solo e do ambiente no cultivo agrícola a partir de indicadores como da umidade, salinidade, temperatura e nutrientes como nitrogênio, fósforo e potássio. Considerando que várias áreas rurais permanecem sem conectividade pois não estão ao alcance de estações base celular os VANTs podem ser úteis para coletar os dados gerados por esses dispositivos de aplicação na agricultura. Portanto, para a coordenação desses voos e o gerenciamento dos dispositivos alocados num campo agrícola uma série de ferramentas são necessárias para apoiar a operação desses voos.

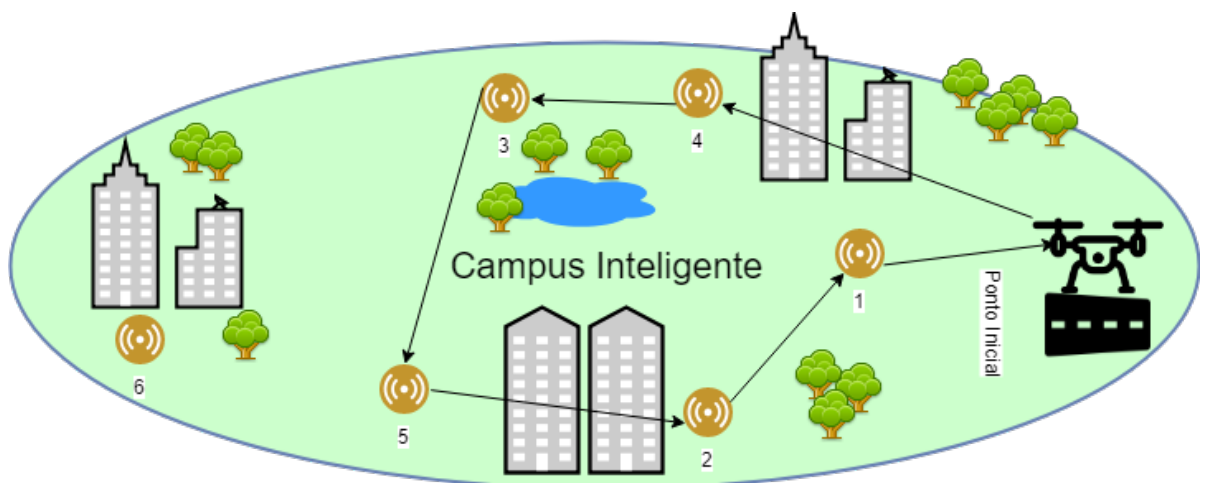


Figura 1.1: Campus Inteligente.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

O objetivo principal do trabalho é adaptação do modelo de otimização do Problema do Roteamento de veículos com foco à prevenção de acidentes, evitando panes técnicas por falta de energia. O modelo gerado com essas características gera por sua vez um plano de voo para ser executado de forma autônoma pela aeronave. O objetivo é atingido ao especificar que a aeronave tem autonomia limitada e que o resultado da otimização é o máximo de pontos possíveis de serem visitados.

Neste trabalho é abordado o planejamento de voo autônomo de VANTs, fazendo com que ao invés de realizar uma atualização da trajetória de voo enquanto o *drone* estiver pairando no ar, esta atualização é feita antes do início de cada voo, realizando o percurso focando apenas na sincronização dos dados com cada um dos dispositivos IoT ou *clusters* previamente definidos. São apresentadas as restrições que devem ser consideradas para que o *drone* sirva ao máximo de dispositivos IoT possíveis dentro de uma área, informando quais são os nós a serem sobrevoados e visitados.

São apresentados dois modelos de otimização, um que restringe a viagem ser executada por apenas um veículo bem como um segundo que atende um plano de voo para vários veículos, que visitam o máximo de pontos de interesse para sincronização de dados considerando a limitação de energia para realizar este voo. O uso dessa otimização fornece uma maneira de prevenir uma pane brusca no equipamento que possa acarretar acidentes ou prejuízos materiais. O modelo leva em consideração a aerodinâmica heterogênea das aeronaves para calcular os custos energéticos, também o tamanho do *buffer* armazenamento nos sensores, também a capacidade de armazenamento no VANT em si, bem como o gasto energético necessário para pairar sobre ponto de visitação enquanto executa a transferência de dados.

É proposta uma arquitetura orientada à serviços para o compartilhamento de recursos de VANT via infraestrutura *cloud*, dessa forma a plataforma recebe solicitação de diversas aplicações externas para que possam consumir dados dos sensores que estão dispostos em uma determinada área, a partir disso os dados coletados são disponibilizados. Integrado a esta arquitetura existe uma ferramenta, apresentada no último capítulo, para gerenciamento do conjunto de sensores abarcados pela infraestrutura bem como um gerador de planos de voo implementando comandos do protocolo *MAVLINK*. A plataforma usa o algoritmo de otimização para realizar o planejamento do voo, e também disponibiliza alternativas de duas meta-heurísticas para também gerarem os planos de voo.

## 1.2.2 Objetivos Específicos

- (1) Modelo de otimização baseado no problema de roteamento de veículos com a finalidade de maximizar os pontos visitados e para esta quantidade máxima de pontos a serem visitados encontrar também a quantidade mínima de consumo energético.
- (2) Arquitetura de micro serviços que compartilhe via cloud os recursos do drone, permitindo a diversas aplicações fazerem adesão à assinatura dos dados que o VANT coletará para que os dados atendem às finalidades próprias de cada aplicação.
- (3) Plataforma integrada à arquitetura apresentada no item anterior, que permite gerenciamento dos dispositivos IoT alocados em uma determinada área e seus tipos de dados relativos. A mesma plataforma oferece através do uso do modelo de otimização apresentado no trabalho um gerador de planos de voos implementado comandos que são executados pela maior parte dos *firmwares* de VANTs.

São apresentados modelos de otimização baseados no típico problema do caixeiro viajante e também no problema do roteamento de veículos aplicados para determinar a sequência de nós de dispositivos IoT ou *clusters* que serão visitados para coletar dados gerados por estes dispositivos. Os VANTs são usados como mula de dados. O VANT provê conectividade *uplink* para dispositivos IoT fixos numa localização em um ambiente inteligente, por exemplo uma cidade ou campus. O VANT é empregado como uma mula de dados que se associa com tais dispositivos durante um voo programável e autônomo para sincronização dos dados coletados por estes sensores. A sequência de sensores e dispositivos IoT que serão visitados, e conseqüentemente, terão os dados sincronizados, é realizado através de vários critérios e restrições, como a autonomia energética (capacidade de bateria) dos VANTs, a capacidade de armazenamento de dados do drone e o buffer limite que cada sensor ou um grupo de dispositivos IoT (*clusters*) possui.

O caminho percorrido pelos *drones*, que executam essa função de mula de dados, é algo muito importante de ser avaliado, pois está relacionado diretamente com a autonomia e o gasto energético do *drone* durante o desempenho deste papel. Ao visualizar-se um conjunto de dispositivos IoT espalhados por uma determinada área geográfica, sabe-se que em muitos casos não é possível servir todos os dispositivos devido a uma limitada carga de bateria encontrada em tais dispositivos.

Para a aplicação no cenário indicado, o *drone* é utilizado como um coletor de dados que percorre os pontos de localização dos sensores de um conjunto pré-estabelecido anteriormente. Este voo é executado com objetivo de sincronizar com o *drone* os dados coletados pelos sensores, que permanecem sempre ativos. O conjunto de sensores somente mudará na realização de um novo voo. Os requisitos para a modelagem da otimização

realizada neste trabalho definem que o modelo atenda exclusivamente a realização de um único voo.

A necessidade de que os dispositivos atendidos pelo *drone* estejam sempre ativos tem por objetivo garantir que os dados coletados no momento do voo sejam o mais atualizados naquele momento, pretendendo não realizar outro voo logo em seguida para realizar atualização dos dados que foram coletados por último. Para situações em que se percebe que algum dos sensores não será mais utilizado por determinado período de tempo que valha a desativação, o *drone* pode realizar um voo de levantamento para aferir quais são os sensores ativos, atualizando assim, o conjunto de sensores para o próximo voo. A cada novo conjunto de dispositivos ativos um novo processamento da otimização é realizado.

## 1.3 Contribuições

Abaixo estão elencados as principais contribuições do trabalho:

- Modelo de otimização de roteamento para *drone* como Mula de Dados que considera: o momento de parada para transferência de dados, a aerodinâmica do veículo, a autonomia do *drone*, o tamanho do buffer dos dispositivos IoT e a capacidade de armazenamento do VANT. O modelo diferente da literatura pois tem foco de prevenção do *panes* por falta de bateria durante o voo, isso acarreta uma série de modificações no modelo em comparação com a estrutura tradicional do problema de roteamento de veículos.
- Plataforma de geração de planos de voo e gerenciamento de dispositivos IoT alocados em uma área, e que pode ser integrada com a arquitetura de micro serviços apresentada, que por sua vez permite que os recursos de coleta de dados do VANT sejam vistos de forma transparente e compartilhada, aplicando o conceito de *Drone as a Service*. Esta contribuição é importante por que conecta o modelo de otimização gerado à arquitetura de compartilhamento, criando uma ferramenta de geração de planos de voo, juntas essas contribuições permitem evoluir na literatura em relação ao conceito de virtualização dessas aeronaves, já que integra a programação linear à arquitetura de serviços.

## 1.4 Estrutura do Documento

O restante deste trabalho está organizado da seguinte maneira: o segundo capítulo trata dos fundamentos conceituais acerca do contexto da pesquisa e segue com a

seção de trabalhos relacionados que apresenta na literatura pesquisas recentes que trataram do mesmo problema e utilizaram abordagens semelhantes referentes a cada um dos fragmentos da solução proposta. O terceiro capítulo se divide em três partes a primeira é apresentado um modelo de roteamento de VANTs a partir de uma abordagem muito semelhante ao caixeiro viajante o que se aplica muito bem para o cenário quando é utilizado apenas uma aeronave, na segunda parte é abordada a técnica de clusterização com a finalidade de ampliar abrangência do número de pontos atendidos por uma aeronave, em seguida definindo um cenário mais próximo ao problema de roteamento de veículos é apresentado uma formalização que leva em consideração múltiplos veículos e é acrescentada a perspectiva da capacidade de armazenamento e *buffer* de *clusters*.

No quarto capítulo é modelada uma arquitetura de serviços com objetivo de compartilhamento de recursos de VANT na *cloud*, com propósito de aplicar o conceito de *drone* como serviço. No quinto capítulo é apresentado um estudo aplicado desenvolvido no período de intercâmbio em Portugal e é mostrada a ferramenta de geração de plano de voos desenvolvida para ser utilizada principalmente num contexto de IoT na agricultura de precisão. O sexto e último capítulo compreende os trabalhos futuros, as considerações finais e conclusão do trabalho.

## ***Background e Trabalhos Relacionados***

---

Este capítulo se inicia abordando os principais fundamentos conceituais em que se baseia o desenvolvimento deste trabalho, isto é feito com objetivo de trazer algumas das origens tecnológicas e científicas que dão impulso para a exploração dos VANTs como instrumento presente na infraestrutura de rede e derivam o surgimento de novas soluções utilizando este que tem estado mais presente nas pesquisas em redes sem fio. Por fim são apresentados os trabalhos relacionados com toda a estrutura do trabalho e os problemas investigados que coincidem com a temática da atual pesquisa.

### **2.1 IoT (*Internet of Things*)**

A importância da IoT, também chamada de Internet das Coisas, é praticamente inquestionável, em partes devido à enorme quantidade de aplicações disponíveis e em desenvolvimento. Devido à implantação e do uso de grandes números de dispositivos de IoT já ser uma realidade, o tema ainda é considerado extremamente importante tanto para a indústria quanto para a academia [62].

A IoT possui um grande potencial para melhorar eficiência na indústria e na sociedade, permitindo a conectividade de dispositivos físicos. Como uma aplicação promissora da IoT, o monitoramento de uma área geográfica, seus aspectos ambientais e sociais, a coleta de dados é viabilizada em tempo real para rastrear e entender as mudanças no ambiente [39]. Pode-se citar como exemplo o fato de que o governo federal do Brasil reconheceu a importância do tema através da iniciativa nomeada como Plano Nacional de Internet das Coisas [42]. Convencionalmente, os nós dos sensores são implantados para coletar dados em áreas de interesse. No entanto, essa abordagem não é econômica ou pode ser difícil em determinadas circunstâncias, por exemplo, um volume grande de nós de sensores são necessários para cobrir uma área grande ou sensores precisam ser implantados em locais perigosos ou áreas de difícil acesso [39].

### 2.1.1 Coleta de Dados

As redes de sensores sem fio são amplamente usadas para coletar diferentes parâmetros detectáveis em um campo. Estas redes propiciam a coleta, o armazenamento e a comunicação com outros sensores encaminhando dados para fins de coleta. o custo de comunicação pode atingir patamares elevados a medida que o encaminhamento de dados também aumenta, o que reduz a vida útil da rede. A vida útil da rede é um dos principais parâmetros de desempenho e um dos requisitos ao implantar uma rede de sensores, garantir que a rede realizou as atividades de coleta de dados durante mais tempo, caso contrário, os custos de implantação serão maiores [64]. A coleta de dados é uma das mais estudadas funcionalidades nas redes de sensores sem fio, que pode, ao ser otimizada, aumentar a vida útil da rede.

As pesquisas envolvendo métodos de coleta de dados em redes de sensores sem fio e nas redes IoT são divididas em três categorias:

O primeiro método é a coleta de dados estática, acontece quando o nó coletor é fixo e os nós comuns fazem upload de dados usando redes por meio de roteamento *one-hop* ou *multi-hop*. Esse tipo de coleta de dados é simples e fácil de implementar, dada a extensa pesquisa realizada na literatura.

Os dois métodos seguintes são coletores móveis também chamados mulas de dados. o uso de nós *sink* para auxiliar na coleta de dados tornou-se uma maneira popular em pesquisas em redes sem fio e IoT (*Internet of Things*). Os elementos móveis podem aumentar a conectividade e a confiabilidade, reduzir custos e diminuir o consumo de energia em nós individuais.

Portanto, o segundo método é a coleta de dados com base em um nó *sink* móvel que trafega em meio terrestre. Esse coletor de dados, geralmente um veículo, é instalado com o nó coletor para visitar a rede através do transporte terrestre e coleta dados de todos os nós no rede. Por exemplo, em uma rede de sensores sem fio urbana, o transporte público, como ônibus e táxi, pode ser usado como coletor de dados móvel.

O terceiro método é a coleta de dados com nós aéreos móveis. Eles coletam dados de sensores implantados em terra. Esse tipo de coleta de dados não possui apenas a flexibilidade da coleta de dados móvel para quando o ambiente de implantação da rede varia, ou seja podem serem levantados novos prédios, ruas podem ser alteradas, bem como locais onde não é possível o alcance humano na região. Comparados com a coleta móvel de dados via terrestre o uso de um coletor aéreo provê uma maior agilidade para se reunir os dados. Uma outra vantagem é o fato que ele tem uma latência mais baixa e maior largura de banda. A coleta de dados aéreos geralmente apresenta menos obstáculos e maior cobertura do sinal sem fio que pode diminuir a latência de comunicação e aumentar a largura de banda [71].

Como a abordagem de coletor móvel elimina o roteamento de vários saltos e o

encaminhamento de rede, espera-se melhorar significativamente a vida útil da rede e a taxa de coleta de dados que justifiquem as vantagens do uso de coletores móveis em vez do encaminhamento de rede. No entanto, ainda existem vários problemas desafiadores que precisam ser mais investigados no contexto de várias aplicações IoT.

Por causa dos obstáculos frequentes no ambiente terrestre e devido a longas distâncias que as mulas de dados precisam se deslocar é necessário usar um coletor móvel mais rápido para reduzir o atraso da coleta física. Nesse sentido, um veículo aéreo não tripulado (UAV) seria um coletor móvel ideal, pois ele pode visitar rapidamente os nós da rede e retornar à estação base com os dados coletados [74].

## 2.2 Cenários Inteligentes

Dentre as aplicações para IoT, aparecem como destaques os Cenários inteligentes, e.g., Cidades e Campus Inteligentes [76]. Um Campus Inteligente pode ser visto como uma versão reduzida da cidade, colocando sua menor extensão como facilitadora de uma gestão mais simples, acelerando a implantação e os testes de solução devido a possibilidade de acesso a toda sua infraestrutura física. Além disso, esta possibilidade de modelagem e testes mais velozes faz com que o campus possa ser considerado um laboratório adequado para testar soluções para ambientes inteligentes baseadas em IoT.

Inúmeros trabalhos aliando Cenários inteligentes e IoT vêm sendo desenvolvidos, com a autonomia energética se apresentando como um dos principais desafios encontrados acerca dos dispositivos. Por causa de sua baixa autonomia, estes possuem pouca mobilidade de conexão, demonstrando uma limitação quanto à transmissão de longo alcance [46].

## 2.3 Veículos Aéreos Não Tripulados (VANTs)

Os VANTs são aeronaves que não precisam de um piloto humano a bordo. Em geral, esses veículos são controlados por um computador incorporado ou por um piloto que opera um controle remoto. Drones, helicópteros com controle remoto e planadores não tripulados são exemplos de VANTs. Os planadores diferem dos outros tipos devido à falta de propulsão a bordo (por exemplo, um motor elétrico ou de combustão). Os VANT modernos foram desenvolvidos pela primeira vez na década de 1920 para apoiar operações militares nas quais a presença de pilotos humanos era impossível ou muito perigosa.

Este tipo de veículo recentemente atraiu um interesse significativo na comunicação sem fio devido à sua alta capacidade de manobra, implantação flexível e baixo custo.

Isso também fornece uma nova ideia para aumentar a vida útil de uma rede de dispositivos IoT [38].

Para coletar dados com eficiência, VANTs podem ser empregados para coletar informação. O rápido desenvolvimento de UAVs, tem permitido uma ampla gama de aplicações, como sensoriamento remoto, fotografia, entrega de carga, inspeção e comunicação. Devido ao alcance operacional limitado de UAVs individuais, pode ser ineficiente para os UAVs detectarem informações em grandes áreas de monitoramento. Com a tecnologia IoT, e o espalhamento de nós IoT ao redor de uma área, um ou vários UAVs podem se conectar a estes sensores e coletar dados de forma colaborativa de maneira eficiente. A capacidade da bateria e o consumo de energia são os fatores mais importantes que afetam o desempenho de UAVs [39]. Geralmente, existem atualmente duas direções principais de pesquisa: otimização da trajetória do UAV ou design da distribuição. [38]

Os VANTs, também chamados *Drones*, podem ser empregados para prover maior mobilidade à uma infraestrutura IoT, pois conseguem se deslocar a uma posição com maior proximidade aos dispositivos IoT, mitigando as deficiências que estes possuem em relação ao seu alcance de transmissão.

Os VANTs são previstos como importantes componentes de um avançado ecossistema ciber-físico da Internet das Coisas [69], devido ao fato que estes possuem características únicas como [60]:

- Facilidade de implantação em diferentes localizações, fazendo com que seu uso seja dinâmico, adaptando às necessidades sazonais e de localidade
- Capacidade de medir qualquer coisa em qualquer lugar, já que há uma enorme abrangência de sensores possíveis de serem acoplados e assim utilizados para capturar dados
- Capacidade de entrega de carregamentos flexíveis, usada com inúmeros propósitos como militar, salvamento e entrega de produtos.
- Reprogramáveis em missão, ou seja no decorrer de um voo pré definido, é possível alterar o percurso estabelecido pelo plano de voo.

A tecnologia de sistemas aéreos não tripulados se espalhou amplamente e se tornou popular em vários campos militares e civis. Algumas das aplicações populares são localização de radar [52], gerenciamento de incêndios florestais, suporte de observação [20], monitoramento agrícola, vigilância e monitoramento de fronteiras, monitoramento ambiental e meteorológico e fotografia aérea [56] além de missões de busca e salvamento [19] Os benefícios mais notáveis em comparação com veículos tripulados convencionais são baixo custo, maior segurança para humanos e fácil implantação [28].

Dentre as aplicações dos VANTs em um ambiente de RSSF ou IoT, o drone é utilizado para estender o alcance da comunicação [7], maximizar a capacidade de

comunicação de dados da rede usando veículos como nós de retransmissão (*relays*) [27], auxiliar na localização do nó em uma rede móvel [63] e por fim, a aplicação que é pesquisada neste trabalho, coletar dados de uma rede de área ampla em locais remotos ou ambientes de difícil acesso [30] [75].

A coleta de dados de nós sensores é uma funcionalidade típica para redes de sensores da Internet das Coisas. Dado que os sensores estão espalhados aleatoriamente uma vasta área, não é fácil obter informações de todos sensores, se não estiverem totalmente conectados. Drones são promissores veículos para coleta de dados em redes de sensores devido à sua habilidades de comunicação direta entre o VANT e o sensor. Em relação à qualidade da comunicação entre os sensores redes e UAVs, estudos de [78] mostraram que uma menor taxa de erro de bit (BER) é detectada se o UAV se comunicar com menos nós de sensores em um determinado momento [75].

## 2.4 Problema do Caixeiro Viajante

Focado na otimização, o PCV é frequentemente usado em ciência da computação para realizar roteamento, seja de dados, veículos, ou pessoas entre nós. Em uma formulação básica do PCV, proposta por Dantzig–Fulkerson–Johnson, atribui-se um índice para cada uma das cidades que o vendedor irá visitar,  $1, \dots, n$ , sendo  $n$  o número de cidades que o problema trata. Tome o custo por aresta percorrida  $c_{ij} > 0$ , a partir disto, define-se:

$$x_{ij} \in \{0, 1\}, \forall i, j = 1, \dots, n \quad (2-1)$$

$$\text{Minimize} \sum_{i=1}^n \sum_{j \neq i, j=1}^n c_{ij} x_{ij} \quad (2-2)$$

$$\sum_{i=1, i \neq j}^n x_{i,j} = 1, \forall j = 1, \dots, n \quad (2-3)$$

$$\sum_{j=1, j \neq i}^n x_{i,j} = 1, \forall i = 1, \dots, n \quad (2-4)$$

$$\sum_{i \in Q} \sum_{j \neq i, j \in Q} x_{i,j} \leq |Q| - 1, \forall Q \not\subseteq \{1, \dots, n\}, |Q| \geq 2 \quad (2-5)$$

Em 2-1, tem-se a variável de decisão binária que atribui valor 1 caso o vendedor tenha percorrido um trajeto saindo de  $i$  para  $j$ . O objetivo 2-2 é minimizar o custo de toda a viagem. As restrições 2-3 e 2-4 garantem que o vendedor irá chegar bem como sair apenas uma vez de cada cidade. A última restrição da formulação garante que não haja

sub-rotas entre os vértices não iniciais, portanto a solução retornada é um único roteiro e não a união de roteiros menores.

## 2.5 Problema de Roteamento de Veículos

O problema de roteamento de veículos (PRV) é um problema muito conhecido em pesquisa operacional e otimização combinatória. No PRV, as rotas deve ser atribuído a um conjunto de veículos que deve atender a um conjunto de clientes de modo que o custo total da operação seja minimizado.

A partir do modelo básico de PCV, deriva-se a formulação do PRV. Seja  $K$  o número de veículos disponíveis para servir o roteamento, e o conjunto de vértices  $V = \{1, \dots, n\}$ , sendo  $n$  o número de nós que deverão ser visitados pelos veículos. Nessa formulação o  $c_{ij}$  representa o custo para que um veículo saia do nó  $i$  para o nó  $j$ , e a variável  $x_{ij}$  é atribuído o valor 1 para quando um veículo ter percorrido a mesma aresta, ao contrário disto o valor será 0. Desse modo temos as a função objetivo bem como as restrições seguintes:

$$x_{ij} \in \{0, 1\}, \forall i, j \in V \quad (2-6)$$

$$\text{Minimize } \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2-7)$$

$$\sum_{i \in V} x_{i,j} = 1, \forall j \in V \quad (2-8)$$

$$\sum_{j \in V} x_{i,j} = 1, \forall i \in V \quad (2-9)$$

$$\sum_{i \in V} x_{i,0} = K \quad (2-10)$$

$$\sum_{j \in V} x_{0,j} = K \quad (2-11)$$

$$\sum_{i \in S} \sum_{j \in S} x_{i,j} \geq |S| - r(s) \quad (2-12)$$

O objetivo 2-7 é minimizar o custo de toda a viagem que os veículos realizarão. As restrições 2-8 e 2-9 garantem que o vendedor irá chegar, bem como sair apenas uma vez de cada cidade. Já em 2-10 e 2-11 é certificado que a quantidade de arcos saindo do nó inicial para qualquer um dos outros nós é igual ao número de veículos disponíveis,

tal qual a quantidade de arcos saindo de qualquer um dos nós e chegando no nó inicial. A última restrição impõe que pelo menos  $r(s)$  arcos deixem cada conjunto de clientes definido como  $S$

## 2.6 Trabalhos Relacionados

Nesta seção são apresentados os trabalhos que se relacionam com os principais pontos abordados nessa pesquisa. As subseções são Aplicações Com VANTs nas Redes de Sensores Sem Fio e no Contexto IoT, onde é elencado trabalhos que utilizam VANTs com finalidade prover vários tipos de comunicação diferentes como *relay*, estação base e também coleta de dados. Em Serviços de Compartilhamento de Drone na Nuvem apresentamos trabalhos que desenvolvem aplicações com finalidade de compartilhamento da aeronave com vários usuários. Na seção com título Clusterização é mostrado exemplos em que a clusterização foi usada para agrupar pontos que serão visitados por *drones*. Por último na seção Problema do Roteamento de Veículos com VANTs são apresentados os trabalhos utilizaram a mesma abordagem para geração de voos que foi usada nesta pesquisa.

### 2.6.1 Aplicações Com VANTs nas Redes de Sensores Sem Fio e no Contexto IoT

Para possibilitar um sistema de transportes inteligente totalmente automatizado em um contexto de Cidade Inteligente, Menouar et al. [43] usa VANTs como dispositivos IoT conectados através de DSRC (*Dedicated Short Range Communications*) aos carros, provendo uma comunicação V2V (*vehicle-to-vehicle*) e V2X (*vehicle-to-infrastructure*). Dados são capturados diretamente dos veículos e transmitidos até seu destinatário interessado na estação base via comunicação *multihop* (*relays* de rádios), em uma abordagem D2D (*device-to-device*). O trabalho conclui que a operação verdadeiramente autônoma dos VANTs é um real desafio, pois este demanda a observação de seres humanos e outros obstáculos quaisquer.

Seiber et al. [58] utilizam *drones* como sensores IoT, com a finalidade de detectar áreas perigosas contaminadas com agentes químicos, sem que seja necessário o uso de homens para fazer esta detecção. A tecnologia de comunicação utilizada entre os *drones* foi o Wi-Fi, e conexão entre *drone* e a estação base foi BLE (*Bluetooth Low Energy*). As limitações encontradas nos testes realizados foram em relação ao voo programado dos *drones*, que se limitou por estabelecer um alinhamento horizontal no enxame de VANTs, o que não aconteceria numa situação real.

Kirichek [35] desenvolve um modelo de entrega de dados partindo de sensores terrestres, sendo transmitidos aos *drones* até a sua chegada em um servidor na nuvem. O modelo proposto possibilita o cálculo do número de VANTs necessários para coleta e entrega dos dados, levando em consideração a intensidade e volume do tráfego na rede, o número de unidades e o raio de sua conexão.

Os trabalhos acima mapearam os elementos técnicos de comunicação que foram eficazes para compor o todo o sistema proposto em operação. Alguns aspectos de modelagem de otimização também foram endereçados para se descobrir qual a melhor configuração em termos de quantidade de VANTs e saltos na repetição das mensagens enviadas dentro do sistema. Embora os trabalhos trazem à tona importantes aspectos referentes ao uso das aeronaves não tripuladas dentro de um sistema de comunicação deve-se enfatizar que a trajetória autônoma desses veículos é um outro importante aspecto que pode ser levada em consideração também quando se for elaborar um modelo de comunicação mesmo que não tenha foco na coleta de dados.

Jiang e Swindlehurst [33] investigaram a trajetória ótima de VANTs equipados com múltiplas antenas para maximizar a somatória da taxa de transmissão em comunicações *uplink*. Zeng, Zhang e Lim [77] maximizam a taxa de transferência de um sistema de relay baseado em VANT através da otimização da trajetória do VANT juntamente com a taxa de transmissão entre a fonte e o *relay*. Mozaffari et al. [45] investigam a implantação ótima e o movimento de um único VANT para apoiar comunicações sem fio *downlink*. Apesar destes trabalhos apresentarem considerações próprias da localização e trajetória das aeronaves, e a melhor implantação de acordo com aspectos do alcance de transmissão e tecnologias de comunicação empregadas, eles ainda não consideram a função de coleta de dados dado uma rede de sensores IoT, por isso podem ser trabalhos que agregam com relação a outras funcionalidades com exceção daquela que leva em consideração o VANT como mula de dados.

Para mitigar a interferência, Mozaffari et al. [44], ajustam apropriadamente a altitude dos VANTs baseando-se no *beamwidth* da antena direcional, bem como nos requisitos de cobertura. Além disso, o número mínimo de VANTs necessários para garantir a probabilidade de cobertura do alvo para uma área geográfica é determinado. Mozaffari et al. [46] pretenderam encontrar uma implantação ótima para que os *drones* servissem como estação base, conseguindo estabelecer uma associação com os dispositivos ativos naquele instante. A quantidade de dispositivos ativos se altera no decorrer do tempo de acordo com que novos dispositivos são ativados e desativados, com este padrão de ativação sendo conhecido pelo servidor na nuvem.

Para fazer esta atualização da trajetória de mobilidade dos VANTs, é realizada uma parada aérea estacionária para propiciar uma pequena economia de energia antes que se continue o movimento do VANT, sem saber qual será a nova trajetória. Este cenário

prevê o uso de um VANT de média autonomia energética, pois considera uma manutenção do voo do *drone* por um período extenso, fazendo com que ele paire no ar durante este período com o objetivo de servir como uma estação base para os dispositivos.

O propósito do modelo deste trabalho não é considerar a mudança da quantidades de dispositivos ativos em tempo real, como acontece na referência [46] do trabalho que foi citado. Esse detalhe foi mencionado apenas para descrever as características daquele, reforçando sua consideração à uma autonomia extensa. Na verdade, pretende-se enfatizar a forma como a autonomia do drone foi usada no referido trabalho, e que aquele caso requer uma quantidade de energia bem maior, já que o drone é usado como estação base e fica durante um longo tempo em voo. A proposta deste trabalho tem como objetivo aplicar o uso da otimização dado um conjunto de sensores previamente estabelecidos a cada voo, fazendo com que o drone não permaneça muito tempo no ar, diferentemente do trabalho [46].

Para o cenário determinado em [46], são consideradas situações com autonomia limitada à uma bateria sem recarga, porém, não há mudanças em quais dispositivos estão ativos no decorrer do tempo, já que é utilizado o VANT como mula de dados, diferentemente do trabalho anterior que utilizou o *drone* como estação base. Quando os *drones* possuem baixa autonomia restante para voo, é necessário utilizar estas para sobrevoo e coleta de dados do máximo de dispositivos IoT possíveis. Em alguns casos pode-se sobrevoar todos os dispositivos IoT existentes em um campus inteligente, caso autonomia dos *drones* for utilizada eficientemente.

Yang e Yoo [75] primeiro adquirem pontos para sensoriamento de dados a partir de todo o campo de sensores, no qual o VANT se comunica com os sensores para obter dados, então determinam a melhor trajetória para percorrer os pontos de aquisição vizinhos. É usado um algoritmo genético para realizar a otimização.

Sujit et al [64] o VANT voa sobre sensores recuperando dados. Para minimizar o tempo de voo do VANT e maximizar o tempo de vida da rede uma rota otimizada conjuntamente para o VANT e a rede de sensores é executada. A principal técnica utilizada para tornar isso possível é a clusterização. Ambos trabalhos apesar de ter atingido ótimos resultados não estabelecem esta trajetória ótima que demonstre um modelo de gasto energético baseado na aerodinâmica do *drone*, nem mesmo adaptam essa trajetória de acordo com aplicações externas que fizeram previamente a assinatura a determinados grupos de sensores o que pode alterar a definição da rota otimizada.

### 2.6.2 Serviços de Compartilhamento de Drone na Nuvem

Choi et al [10] utilizam um esquema de comunicação baseado em padrões que é útil em realizar um gerenciamento e controle eficiente de operações baseadas em VANTs.

Considerando essa questão, fazem uso de um padrão global denominado oneM2M, que é usado para prover comunicação máquina à máquina e uma camada de serviços para interoperar soluções M2M/IoT.

Koubâa et al [36] contribuem para a IoD (*Internet of Drones*) e para a implantação de VANTs sobre a nuvem. Um sistema inovador orientado a serviço e baseado na nuvem juntamente com um sistema de gestão que provê acesso aos VANTs através de *WebServices*, agendando missões e promovendo colaborações entre *drones*.

Nos trabalhos acima os serviços desenvolvidos não contemplam a disponibilização dos dados adquiridos pelo drone ao sobrevoar uma rede de sensores IoT. Este serviço atende a inúmeros usuários ou aplicações clientes que definem quais tipos de sensores elas necessitam consumir os dados. A partir da solicitação por parte destas aplicações a plataforma é provocada a otimizar os recursos do VANT em relação a sua autonomia e a disponibilidade para voos e o próprio serviço otimiza o voo do drone. Esta abordagem conciliando a otimização do plano de voo com uma interface para usuários compartilham os recursos de um drone mula de dados ainda não tem precedente na literatura.

### 2.6.3 Problema do Roteamento de Veículos com VANTs

Isaacs explorou a mobilidade do coletor de dados para mitigar o consumo de energia dos nós estacionários e permitir implantações esparsas ao custo de latência adicional de dados [31]. Também buscaram estratégias eficientes para aliviar essa latência de dados. Trabalharam com o posicionamento ideal do sensor e seleção ideal do sensor para ajustar adaptativamente a rota da mula de dados para minimizar o tempo para localizar eventos de interesse. Quando a mula de dados é um VANT de asa fixa, esse algoritmo pode ser aprimorado incluindo as restrições cinemáticas do VANT e explorar a capacidade do VANT para se comunicar com sensores distantes dentro de sua linha de visão. Por fim, forneceram políticas de roteamento de mulas de dados para localização de fontes acústicas em um grande área envolvendo muitas fontes e sensores.

Fügenschuh et Müllenstedt formulam o problema de planejamento da missão para uma frota de VANTs como um problema de Programação Não Linear Inteira Mista (PNLIM) [21]. O problema pede uma seleção de alvos de uma lista para os VANTs e trajetórias que visitam os alvos escolhidos. Para ser viável, uma trajetória deve passar por cada alvo a uma distância desejada e, dentro de um determinado período, janelas, obstáculos ou regiões de alto risco devem ser evitados e as limitações de combustível devem ser observadas. Uma trajetória ideal maximiza a soma dos valores de todos os alvos que podem ser visitados e, como objetivo secundário, conduz a missão no menor tempo possível.

Semiz desenvolveu um método algorítmico para resolver o problema de rotea-

mento de veículos aplicado a VANTs [59]. Este método usa o método de dividir e conquistar para a resolução. Dessa forma, o problema é transformado em uma combinação de vários subproblemas. É projetado um método para converter esses pequenos problemas em problemas de transporte. Cada problema de transporte é resolvido com o algoritmo simplex. Neste problema, uma frota de VANTs deve atravessar um conjunto de áreas-alvo dentro de um área. Os destinos estão disponíveis apenas dentro de certas janelas de tempo e precisam ser atravessado prontamente. Além disso, para algumas áreas-alvo grandes, são necessários vários VANTs para executar a tarefa. O objetivo é encontrar uma programação completa e atribuição de meta ao VANT, de forma que se minimize o consumo total de combustível.

O Problema de Roteamento de Veículos (PRV) e suas diferentes variantes foram estudado por muitas décadas, encontrando sempre aplicações no mundo real. Porém, a exemplo dos trabalhos acima, pouco foi feito no desenvolvimento de problemas de roteamento de veículos (PRVs) especificamente para cenários de *drones* como mula de dados. Os PRVs existentes são insuficientes para o planejamento de coletas de dados com drones, levando a possibilidade da perda do veículo com quedas repentinas devido ao esgotamento da bateria, que na maioria dos casos é bastante limitada. Neste trabalho os modelos propostos visam primordialmente a prevenção de acidentes decorrentes da falta de energia, garantindo que o *drone* visite somente a quantidade de pontos que ele tem autonomia para concluir.

Sistemas aéreos de robótica podem ser muito úteis para executar tarefas complexas de maneira distribuída e cooperativa, como localização de alvos e busca de pontos de interesse (POIs). Guerriero et al propõe um sistema distribuído de veículos aéreos não tripulados autônomos, capaz de se coordenar e cooperar para garantir a cobertura espacial e temporal de tempo específico e POIs espaciais variáveis [23]. É considerado que o sistema de VANTs é capaz de resolver problemas de agendamento dinâmico distribuído, sendo necessário que cada dispositivo mova em direção a uma determinada posição em um determinado momento. É dado uma formulação matemática do problema como modelo de otimização de múltiplos critérios, no qual as distâncias totais percorridas pelos VANTs (a serem minimizados), a satisfação do cliente (a ser maximizada) e o número dos VANTs usados (a serem minimizados) são considerados simultaneamente. É apresentada uma variante dinâmica do modelo básico de otimização, considerando o conceito de *rolling horizon*. É introduzido um estudo de caso como cenário de aplicação, em que as ações esportivas de uma bola de futebol são filmadas através de um sistema de VANTs distribuídos.

O problema de roteamento de veículos com *drones* (PRVD) é uma extensão do clássico problema de roteamento de veículos capacitados, onde não apenas caminhões, mas *drones* são usados para realizar entregas a clientes. Uma característica distintiva do

PRVD é que um drone pode viajar com um caminhão, levar fora de sua parada para atender os clientes e pousar em um centro de serviços para viajar com outro caminhão desde que as limitações de alcance e capacidade de carga sejam atendidas. Wang et Sheu propõe um modelo de programação inteiro misto, e desenvolvem um algoritmo que utilizam o método *branch-and-price* [72].

Adbelhafiz et al analisam brevemente as instâncias do PRV (Problema do Roteamento de Veículos) para operações de VANT com vários objetivos [2]. Também focam em problemas de planejamento de missão multi-objetivo e multi-VANT, e tentam tirar proveito da literatura do PRV e suas variantes. É mostrado que cada missão militar multi-VANT possui sua variante PRV correspondente. É apresentado um novo algoritmo que se baseia em um aprimoramento da pesquisa em árvore para resolver problemas complexos de planejamento de missões multi-VANT com restrições complexas.

A consideração dos atributos da autonomia de múltiplos VANTs bem como *buffer* dos sensores de maneira integrada torna o problema muito mais desafiador e diferente da literatura clássica de roteamento de veículos com capacidade. Estes aspectos não são abordados dessa forma nos trabalhos anteriores. É considerada neste trabalho, também a capacidade de armazenamento de dados nos *drones* como um fator limitante, pois considerando que estes veículos muitas vezes são compartilhados para variadas atividades, o total de armazenamento disponível pode inviabilizar uma missão. O *buffer* de dados contido nos sensores ou até mesmo em clusters de sensores é considerado para identificar o tempo de *hovering* (planação), e a partir disso o consumo energético gerado durante esse momento de transferência de dados é acrescido ao consumo do deslocamento do drone entre os clusters.

Dorling et al propõem dois PRVs multi viagem para *drones* de entrega que abordam várias viagens ao depósito e o efeito do peso da bateria e da carga útil no consumo de energia [15]. Um minimiza custos restrito até um prazo de entrega, enquanto o outro minimiza o prazo de entrega sujeito a uma restrição orçamentária. Propõem um modelo de consumo de energia para *drones* multi rotores, demonstrando que o consumo de energia varia aproximadamente linearmente com a carga útil e o peso da bateria. É usado essa aproximação para derivar programas lineares inteiros mistos para nossos VRPs. É proposta uma função de custo que considera o modelo de consumo de energia e reuso de *drones* e é aplicada em uma heurística de *simulated annealing* (SA) para encontrar soluções subótimas para cenários práticos.

Coutinho et al fornecem uma definição do Problema de Otimização de Roteamento e Trajetória de VANT [11]. É introduzida uma taxonomia e revisaram contribuições recentes na otimização da trajetória do VANT, roteamento do VANT e abordagem de artigos para esses problemas e suas variantes. Concluem que a complexidade do roteamento de VANTs não foi totalmente investigado na literatura, e apresentam oportunidades

futuras de pesquisa.

A utilização de apenas uma classe de algoritmo para resolver problemas complexos como o roteamento de mula de dados tem se mostrado não ser uma boa estratégia, por isso os trabalhos acima fazem uso do encadeamento de algoritmos com a finalidade de se ter a melhor performance de processamento bem como a melhor solução ótima. No entanto, eles não fazem uso de clusterização nesse encadeamento com objetivo de atingir o resultado ótimo. O MILP pode acomodar variáveis de decisão contínuas e binárias, bem como uma variedade de restrições e funções objetivas, no entanto, a natureza NP-Completo implica um aumento dramático na complexidade de computação à medida que o número de variáveis e restrições aumenta. Esse problema é altamente crítico e precisa ser resolvido quase em tempo real. Portanto, métodos que fazem pesquisas exaustivas podem se tornar inviáveis a medida que o espaço amostral aumenta. Para lidar com uma quantidade elevada de sensores e realizar o agrupamento dos dispositivos, é proposto o uso de técnicas de *clustering* entre o método *linkage complete*, subtipo da clusterização hierárquica. Este agrupamento é útil para que a aeronave paire num mesmo ponto e assim realizar a sincronização com vários sensores simultâneos. Os modelos finais proposto tem dois objetivos diferentes e complementares, sendo que um visa maximizar o total dos sensores visitados e o outro atingir o mínimo de consumo de energia para aquela quantidade de sensores máxima. Esta proposta é endereçada no capítulo 3.

## Modelos de Otimização Propostos e Avaliação de Resultados

---

Neste capítulo é apresentado modelos derivados dos problemas clássicos do caixeiro viajante (PCV) e do roteamento de veículos (PRV), ambos aplicados no uso de *drones* como coletores de dados de dispositivos IoT espalhados numa dada área. Entre as propostas estão um modelo que atende apenas uma única aeronave e outro modelo que atenda múltiplos VANTs para que sejam comparados em termos eficiência energética, além disso o modelo para uma única aeronave é apresentado como estudo preliminar onde não são mapeadas todas as características limitantes dos *drones* são endereçadas neste trabalho, o que passa a acontecer no segundo modelo proposto. VANTs é As soluções dadas são modeladas e implementadas usando PLIM (Programação Linear Inteira Mista) [47]. A escolha da técnica Programação Linear se dá pois as funções objetivo e as restrições são funções lineares. Também é apresentado no capítulo a abordagem utilizada para agrupamento de dispositivos próximos. Esse aglutinamento torna os modelos ainda mais eficientes pois possibilitam a conexão em um mesmo ponto central com vários dispositivos ao mesmo tempo.

Com relação aos modelos derivados do PCV foram apresentados dois modelos de otimização visando reduzir o consumo de energia através do uso otimizado da autonomia de apenas um *drone*, fazendo com que o máximo possível de dispositivos IoT sejam visitados dentro dos nós alocados. Os modelos apresentados na seção relacionada com o PCV possuem publicações referentes a cada abordagem. O PLIM apresentado na seção (3.1.1) possui sua versão publicada na ERI-GO (Escola Regional de Informática de Goiás) [54]. Quanto a seção (3.1.2) também gerou uma publicação na Revista de Sistemas de Informação da Faculdade Salesiana Maria Auxiliadora [55], há uma evolução em pontos muito importantes em relação a primeira publicação, como por exemplo a localização 3D dos VANTs e o aperfeiçoamento do modelo de consumo energético que será detalhado a seguir.

Na segunda sessão é determinado o método de clusterização utilizado. É descrito a técnica de clusterização hierárquica aglomerativo através do método *Linkage Complete*,

para se chegar a quantos *clusters* são possíveis de serem gerados, bem como qual é o *centroid*, ou seja, o ponto central em que a aeronave irá permanecer enquanto sincroniza dados com o grupo de sensores daquele *cluster*.

Na terceira sessão do capítulo, referente aos modelos derivados do PRV, acrescentamos alguns aspectos que tornaram o modelo mais abrangente em relação ao cenário já descrito. Foi introduzida a possibilidade de se trabalhar com mais do que um *drone*, e em nosso caso a simulação é feita com apenas dois veículos para que seja planejados os voos simultâneos dos dois para uma dada missão. Dessa forma o considerando as limitações de energia próprias e capacidades próprias de cada aeronave as rotas que cada uma deverá percorrer são determinadas.

Após a definição do modelo matemático usado para tratar as variáveis existentes dentro do contexto dos problemas na primeira seção, ele é processado através da ferramenta de otimização CPLEX [12], e, para a implementação, é usada a linguagem de otimização OPL (*Optimization Programming Language*). Após a coleta de dados extraídos das simulações e soluções encontradas no CPLEX, é utilizado o MATLAB [24] para realizar a plotagem dos rotas resultantes das soluções ótimas encontradas. Os modelos derivados de PRV bem como os gráficos de avaliação do modelo são implementados a linguagem Python. Para o modelo de PLIM é utilizado o *framework* PuLP que é uma modelador de Programação Linear escrito em Python. E para os gráficos é utilizado a biblioteca *mpl\_toolkits* que fornece algumas ferramentas básicas de plotagem 3D.

## 3.1 Plano de voo com único VANT e autonomia limitada

Nesta seção é aplicado o Problema do Caixeiro Viajante (PCV) ao caso de uso de VANTs como mula de dados. o PCV é um problema algorítmico encarregado de encontrar a rota mais curta entre um conjunto de pontos e locais que devem ser visitados. Neste trabalho, os pontos são as dispositivos que uma aeronave não tripulada pode visitar. O objetivo do vendedor, neste caso o *drone*, é manter os custos de viagem o mais baixo possível.

### 3.1.1 Abordagem com Localizações 2D e Gasto Energético Simplificado

Para otimização da trajetória de *drones* é aplicado a técnica de Programação Linear. Para cenários onde *drones* possuem baixa autonomia restante para voo, é necessário utilizar esta autonomia para sobrevoo e coleta de dados do máximo possível de sensores. Em alguns casos pode-se sobrevoar todos os sensores existentes em um campus inteligente, se a autonomia dos *drones* for utilizada otimadamente.

Considere um conjunto de sensores  $S = \{1, 2, \dots, n\}$  estes sensores estão espalhados em um campus de forma aleatória, estabelecendo entre cada um desses sensores um arco de distância. O conjunto de arestas de distâncias entre os pontos de localização dos sensores é dado por  $D = \{(1, 1), (1, 2), \dots, (n - 1, n)\}$ , onde os pares distância entre o mesmo sensor, por exemplo  $(1, 1)$ , é zero. A distância entre um par de sensores, por exemplo  $(1, 2)$ , é igual ao seu inverso  $(2, 1)$ . Este conjunto de arestas dá origem uma matriz de distâncias entre todos os sensores do conjunto  $S$ , que por sua vez gera um grafo completo bidirecional (Figura 3.1). Para fins de modelagem, o conjunto  $S$  possui o número de elementos igual a: Número de Sensores somado o Ponto Inicial. O grafo demonstra todas as possibilidades de percurso a ser sobrevoado por um *drone*, com finalidade de coletar dados em todos esses sensores. O *drone* que executa o voo nessa trajetória possui autonomia limitada dada pela variável  $A$ . Os grafos não ilustram o tamanho das arestas real de acordo com os valores atribuídos a cada uma delas. As figuras apenas irão demonstrar a sequência de vértices visitados para cada conjunto de sensores e autonomia especificada.

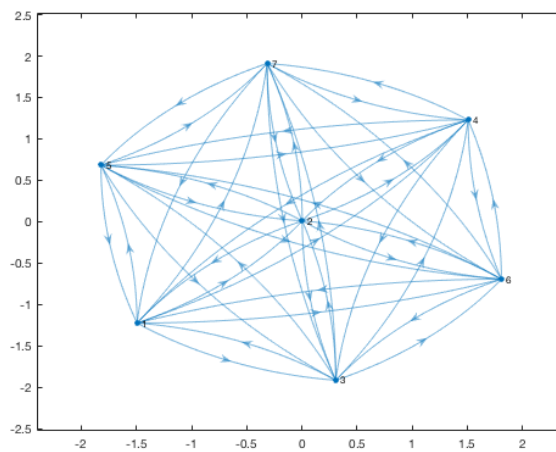


Figura 3.1: Grafo Completo Bidirecional

A partir da energia dada, em watt-hora, a autonomia é calculada pelo tempo restante  $T = E \times P$ , ( $P =$  Potência, em watt). Considerando uma velocidade constante  $V$ , em  $m/s$ , do *drone* se tem a distância máxima possível de ser sobrevoada, considerando a autonomia inicial, dada por  $A = T \div V$ . É importante frisar que para fins de simplificação não é considerada um outro vetor de localização que determinaria a altitude do drone num dado instante do percurso, também não são considerados outros fatores de influência na autonomia do *drone* como peso total do *drone*, empuxo líquido etc. Porém, para compreensão e solução de parte importante do problema as informações consideradas são suficientes para demonstração da otimização realizada e das simulações feitas.

Para cada uma das arestas que é computada como trajetória contida no percurso do voo é necessário determinar uma variável booleana que especifica se a aresta é

percorrida ou não para um par  $(i,j)$  sendo o  $i, j \in S$  e  $(i,j) \in E$ . Esta variável de decisão é dada em (3-1):

$$X_{i,j} \in \{0,1\}, \forall (i,j) \in E \quad (3-1)$$

Num cenário onde se tenha um conjunto de dispositivos IoT espalhados por determinada área, serviriam ao propósito de produção de dados, que por sua vez serão utilizados para prover algum serviço ou aplicação. Portanto, para este contexto o maior interesse é que seja visitado o maior número de dispositivos IoT possíveis. Ainda, temos para este cenário, que o VANT deverá sair de um Ponto Inicial, padronizado como tendo coordenadas  $(0,0)$  e deverá voltar para o mesmo ponto, ou seja, esta coordenada é definida também como a posição de retorno do VANT. Dado um conjunto  $A_r$  de arestas percorridas, o número de dispositivos IoT visitados é igual a  $A_r - 1$ , pois a última aresta  $(i,j)$  é relativa à volta do *drone* para o Ponto Inicial  $(0,0)$ . O objetivo da otimização modelada é maximizar o número de dispositivos IoT visitados, dado pelo somatório de arestas percorridas. Este objetivo é demonstrado em (3-2):

$$\text{Maximizar} \left( \sum_{(i,j) \in E} X_{i,j} \right) - 1 \quad (3-2)$$

Para atender o objetivo especificado existem várias restrições de comportamento para se atender em relação à trajetória. O *drone* deve obrigatoriamente sair do Ponto Inicial e se dirigir para algum dos dispositivos IoT somente uma vez. Para isso, têm-se que o somatório das variáveis de decisão das arestas do Ponto Inicial para todos os  $N$  dispositivos IoT deve ser igual a um, tal como demonstrado em (3-3). Isso significa que caso haver, por exemplo, 5 sensores espalhados pelo campus, o VANT irá sair do Ponto Inicial e ir sair de lá para somente um dos 5 sensores, ou seja, não poderá uma vez que retornou ao Ponto Inicial sair novamente para um outro sensor, considerando que o VANT realizará apenas um voo. Tal padrão vale para o retorno ao Ponto Inicial, onde o somatório das variáveis de decisão também vale um (3-4).

$$\sum_{(l,j) \in E} X_{l,j} = 1 \quad (3-3)$$

$$\sum_{(l,j) \in E} X_{l,l} = 1 \quad (3-4)$$

A restrição (3-5) garante que o somatório das variáveis de decisão de chegada em um sensor deve ser igual ao somatório das saídas no mesmo sensor. Essa restrição é importante pois em cada sensor que o *drone* entre sobrevoando, percorrendo a trajetória de chegada pelo sensor, o *drone* deve, para este, percorrer a trajetória de saída. Entende-se

por trajetória de saída a aresta que vai do último ponto visitado pelo drone até o próximo ponto de sensor que será visitado.

$$\sum_{(i,h) \in E} X_{i,h} = \sum_{(h,j) \in E} X_{h,j}, \forall h \in S \quad (3-5)$$

Dado um subconjunto  $R$  de dispositivos IoT dentro do conjunto total, neste, o total de arestas percorridas deve ser igual o total de vértices (dispositivos IoT), menos um, para que não se feche uma sub trajetória dentro daquele conjunto de dispositivos IoT (3-24). Esta restrição atua para evitar qualquer desperdício na autonomia restante do *drone* durante o voo, uma vez que se um sensor já tiver sido visitado anteriormente não é necessário que se retorne ao mesmo.

$$\sum_{i \in R} \sum_{j \in R} X_{i,j} \leq |R| - 1, R \subset S, 2 \leq |R| \leq n \quad (3-6)$$

Para especificar que o *drone* não possa percorrer uma distância maior do que a autonomia que ele possui, deve-se multiplicar cada variável de decisão, zero ou um, pela distância daquela aresta, e o somatório de todas distâncias percorridas deve ser menor ou igual à autonomia total do *drone* (3-7).

$$\sum_{(i,j) \in E} E_{i,j} X_{i,j} \leq A \quad (3-7)$$

### Rota resultante do Plano de Voo em 2D

É adotada uma estratégia nas simulações a seguir que consiste em variar a autonomia da aeronave considerando o mesmo grupo de sensores alocados na mesma área, mantendo portanto as distâncias entre os pontos. Foram simuladas dois conjuntos de sensores alocados, o primeiro com 6 sensores e o segundo com 14, cada conjunto com três instâncias de autonomies diferentes. Os sensores são alocados em uma área num raio de no máximo 100 metros de distância entre cada um dos sensores. As distâncias entre os sensores são estáticas, isto é, não mudam a cada vôo. O nó inicial, que é o ponto de saída do *drone* deve ser levado em consideração para compor o total de nós do conjunto de sensores. O Ponto Inicial é um local onde será guardado o *drone* e eventualmente pode-se efetuar recargas, preferencialmente no período onde não há vôos programados, por isso neste cenário é considerado vários vôos executados sequencialmente por exemplo num campus universitário ao longo de um período que não houveram recargas no VANT.

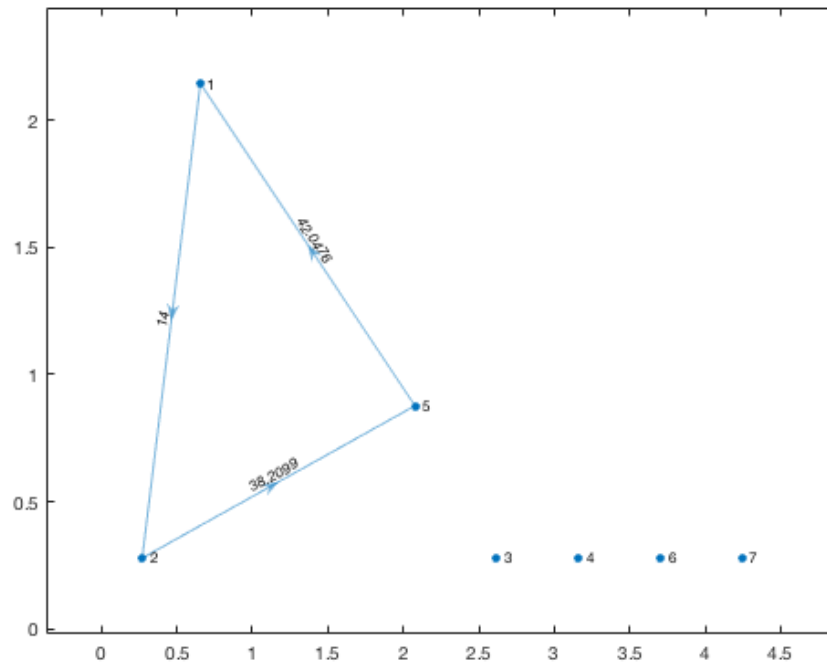


Figura 3.2: Trajetória gerada para o *drone* com autonomia de 150 metros

Na primeira instância (Figura 3) o *drone* tem uma autonomia total de 150 metros para ser percorrido, e considerando os 6 sensores alocados na área, o *drone* só consegue visitar 2 sensores, fazendo o percurso de três arestas para visitar estes sensores, sendo eles: (1,5), (5,2) e (2,1). A distância total percorrida pelo *drone* é de 94,24 metros portanto restante 55,76 metros de autonomia que ainda pode ser percorridos pelo *drone*, porém não podem ser aproveitados pelo *drone* por que não seria suficiente para visitar mais sensores considerando não somente a distância para chegada no sensor como também a distância de volta até o ponto inicial. As distâncias entre os sensores utilizada na primeira instância é a mesma entre todas as demais instâncias simuladas dentro deste conjunto de sensores, mas as distâncias são diferentes no outro conjunto de sensores. Nos grafos ilustrados nas figuras é mostrado a sequência de pontos visitados porém não é marcada a localização geográfica dos pontos.

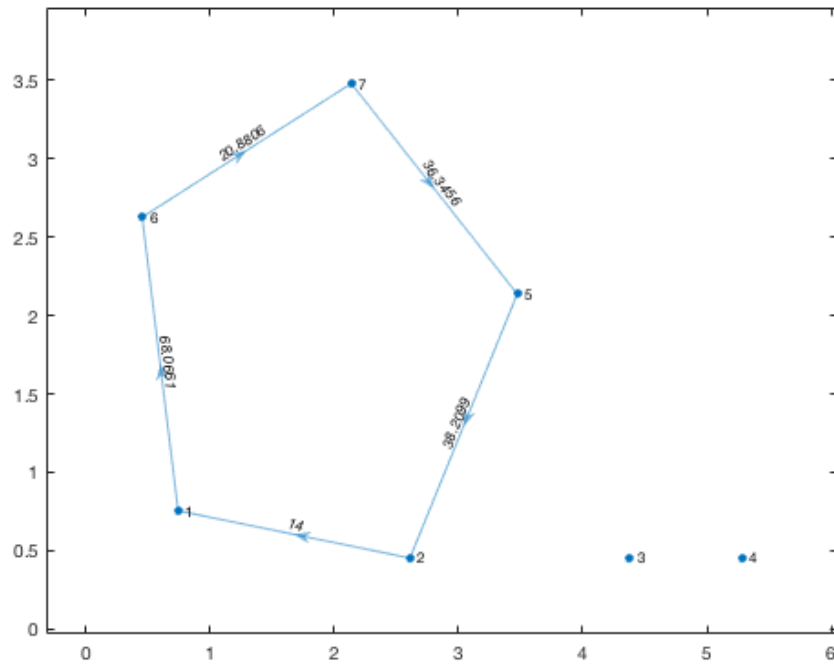


Figura 3.3: Trajetória gerada para o *drone* com autonomia de 200 metros

Quanto à segunda instância (Figura 4) analisada foi atribuído uma autonomia total do *drone* de 200 metros. Fazendo um aproveitamento desses 200 metros para o total de sensores alocados, o maior número possível de sensores visitados foi 4. O total da distância sobrevoada é de 125,28 metros com restante de autonomia de 74,72 metros. Veja que apesar da autonomia restante parecer alta a otimização realizada concentra-se na quantidade de sensores visitados, por isso é certo que independente da quantidade de autonomia restante o número de sensores visitados é o máximo que se conseguiria sobrevoar com a autonomia considerada. É possível que haja várias instâncias de distâncias totais percorridas para a mesma quantidade de sensores alcançadas com a autonomia limitada.

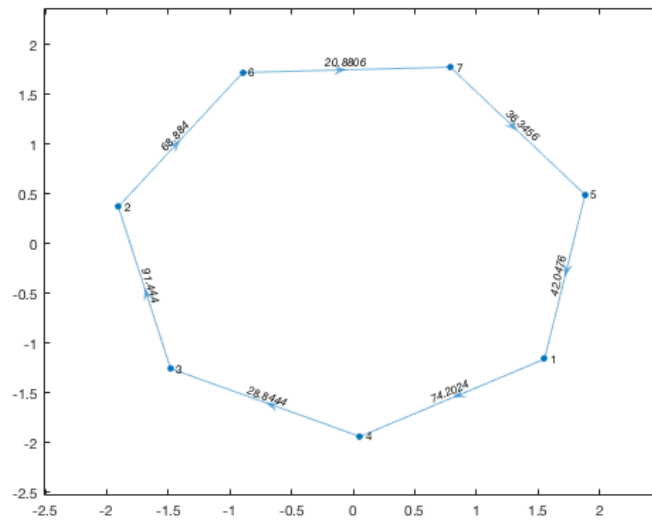


Figura 3.4: Trajetória gerada para o *drone* com autonomia de 500 metros

Uma terceira e última instância de autonomia é analisada, atribuindo 500 metros de autonomia para o *drone* (Figura 5). Para essa instância o *drone* consegue sobrevoar todos os 6 nós de sensores dentro da alocação estabelecida. Para efetuar toda essa trajetória ele gasta 362,62 metros de autonomia com restante de 137,38 metros. Nota-se que com apenas um pouco mais de carga no *drone*, já seria possível efetuar a trajetória da primeira instância. É importante enfatizar que não se pode afirmar que 362,62 metros é o mínimo de autonomia que o *drone* deve ter para sobrevoar todos os sensores. Por que as restrições não garantem que ele sobrevoe o mínimo de distância para sobrevoar o máximo de sensores. Essa distância mínima não é garantida pelo modelo de otimização proposto.

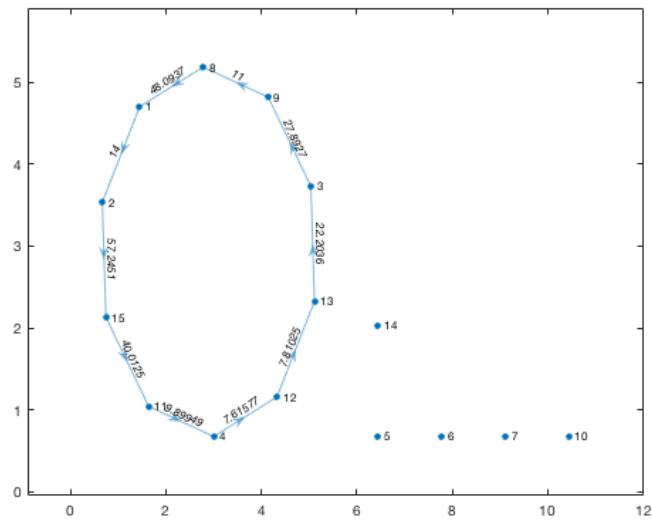


Figura 3.5: Trajetória possível para 15 sensores alocados e autonomia de 250 metros para o *drone*

Também é simulado um campus onde haja 14 sensores alocados, ou seja, mais que o dobro de sensores nos testes realizados no cenário anterior. Para este conjunto de sensores é executado três instâncias de testes. Na primeira instância, (Figura 6) para uma autonomia de 250 metros encontrada no VANT é percorrida a distância de 245,63 metros, e consegue-se sobrevoar 9 sensores, sem contar o Ponto Inicial de onde parte o *drone*. Na segunda instância (Figura 7) é alocada uma autonomia de 350 metros para o veículo, que assim percorre uma distância de 348,73 metros para se atingir 13 sensores. Quanto a terceira e última instância (Figura 8) com autonomia de 450 metros, são sobrevoados todos os 14 sensores e é percorrido 448.38 metros para chegar a este fim.

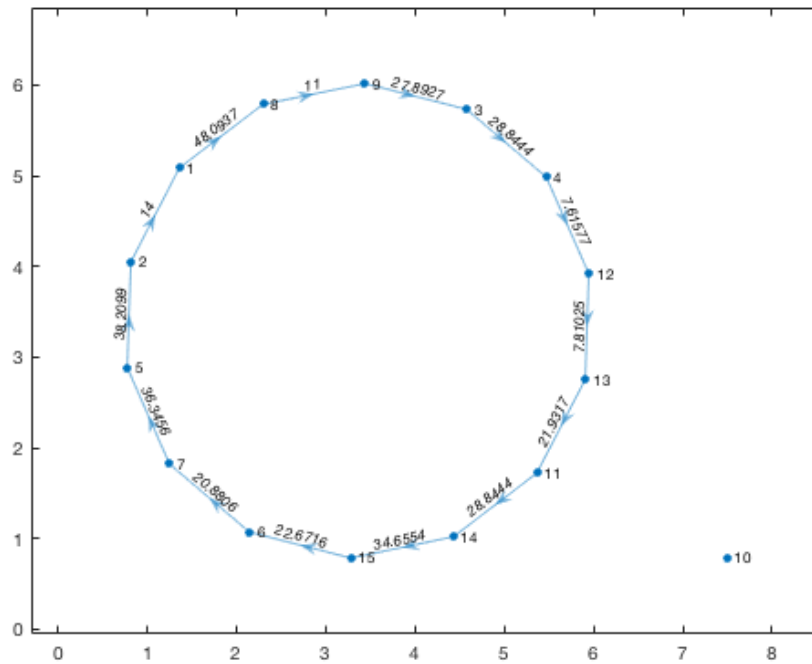


Figura 3.6: Trajetória possível para 15 sensores alocados e autonomia de 350 metros para o *drone*

Algo que pode ser destacado quanto a essas simulações se refere ao grande aproveitamento das autonomias encontradas no VANT, fazendo com que a autonomia restante seja muito pouca. Considerando que o limite de distância entre os sensores é o mesmo usado no conjunto anterior alocado (100 metros) nota-se que uma densidade maior de sensores num dado espaço é encontrado, e por isso há um maior aproveitamento das rotas, pois é possível fazer dentro dos mesmos deslocamentos referentes ao conjunto de sensores anteriores porém visitando um maior número de sensores.

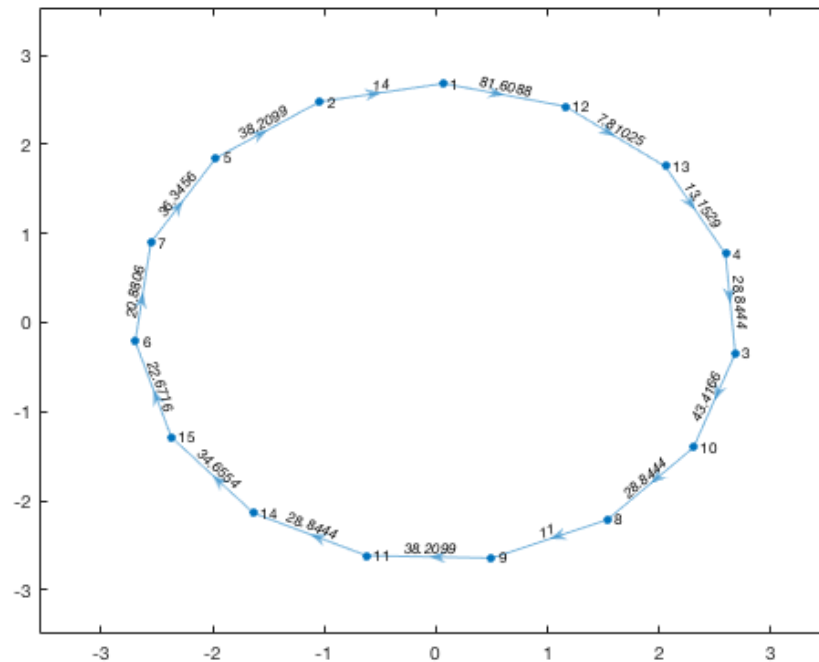


Figura 3.7: Trajetória possível para 15 sensores alocados e autonomia de 450 metros para o *drone*

Este modelo tem uma proposta muito semelhante ao problema do caixeiro viajante, porém, várias especificidades são adequadas neste cenário com um *drone* sobrevoando sensores IoT alocados numa determinada área. O modelo nesta atinge com um *drone* de autonomia limitada, o máximo de visitas nos nós de sensores alocados. Este modelo foi elaborado considerando algumas simplificações quanto ao padrão de consumo dos *drones*, ignorando peso total do *drone* bem como o empuxo, fatores que são mostrados na próxima seção.

Outra simplificação empregada foi em relação a altitude do *drone* durante o sobrevôo. Esta localização do *drone* em três dimensões propiciaria um acompanhamento do VANT de forma que poderia ser calculado precisamente a distância do *drone* para o sensor e com isso determinar quais os alcances possíveis dentro das tecnologias de comunicação utilizadas entre o *drone* e os sensores.

### 3.1.2 Abordagem com Localizações 3D

Considerando um conjunto de dispositivos IoT  $S = \{1, 2, \dots, N\}$  espalhados em um campus de forma aleatória, estabelecendo entre cada um desses sensores IoT um arco de distância, ou seja a linha da trajetória entre cada um dos pontos de localização destes dispositivos. A partir desse conjunto de dispositivos são gerados outros

3 conjuntos para as coordenadas tridimensionais da localização de cada um dos IoT:  $X_S = \{X_{S_1}, X_{S_2}, \dots, X_{S_N}\}$ ,  $Y_S = \{Y_{S_1}, Y_{S_2}, \dots, Y_{S_N}\}$  e  $Z_S = \{Z_{S_1}, Z_{S_2}, \dots, Z_{S_N}\}$ .

Na abordagem feita na subseção anterior, desconsiderava-se a altura do VANT, por este motivo, o cálculo energético é feito de forma simplificada, baseando-se apenas em dois eixos de localização, utilizando assim a fórmula básica de velocidade para calcular a autonomia. Neste trabalho o cálculo energético leva em consideração outras variáveis que são apresentadas na próxima seção.

O conjunto de arestas entre os pontos de localização dos dispositivos IoT é dado por  $A_r = \{(1, 1), (1, 2), \dots, (N - 1, N)\}$ . Este conjunto dá origem a uma matriz de energias consumidas para o trajeto entre os nós da aresta, referentes a todos os dispositivos IoT do conjunto S entre si quando a energia consumida dos pares com o mesmo sensor, por exemplo, (1,1), é zero. A matriz de energias juntamente com todo o conjunto  $A_r$  geram um grafo completo, ponderado e bidirecional, representado na Figura (3.8), usada para mostrar todos os percursos possíveis a serem executados pelo VANT. Na figura também são especificadas as localizações (x, y e z) dos 9 dispositivos IoT e o Ponto Inicial nessa instância.

O *drone* que executa o voo nessa trajetória possui autonomia limitada dada pela variável  $A$ , que terá seu cálculo, bem como o do Gasto Energético do *drone*, demonstrados na subseção a seguir. Para fins de modelagem, o conjunto S possui o número de elementos igual a: Número de dispositivos IoT somados ao Ponto Inicial. O Ponto Inicial é o local de decolagem e aterrissagem do *drone*.

É importante observar que o problema aqui abordado, em relação à modelagem de Programação Linear, possui similaridades à vários aspectos no Problema do Caixeiro Viajante [14]. O PCV tem por objetivo visitar todos os nós distribuídos em uma determinada área atingindo o menor custo para se realizar isso. Note que neste trabalho é tratado a possibilidade de não ser viável visitar todos os nós, eventualmente por não se ter a autonomia necessária para tal. Com isso difere-se especialmente o objetivo da otimização, que neste caso é saber o máximo possível de pontos visitados dada a restrição de autonomia energética que é oferecida.

### **Cálculo da Autonomia e do Gasto Energético do Drone**

É efetuado o cálculo da energia total armazenada em cada bateria que o drone pode utilizar. Todo o cálculo explicado nesta subseção é utilizado para gerar entradas para a matriz de energias especificada na subção anterior. Primeiramente, é necessário calcular a Potência da bateria ( $P$ ). Este cálculo se inicia a partir dos parâmetros de Corrente Elétrica ( $I$ ) e Tensão ( $V$ ) (3-8):

$$P = I \times V \quad (3-8)$$

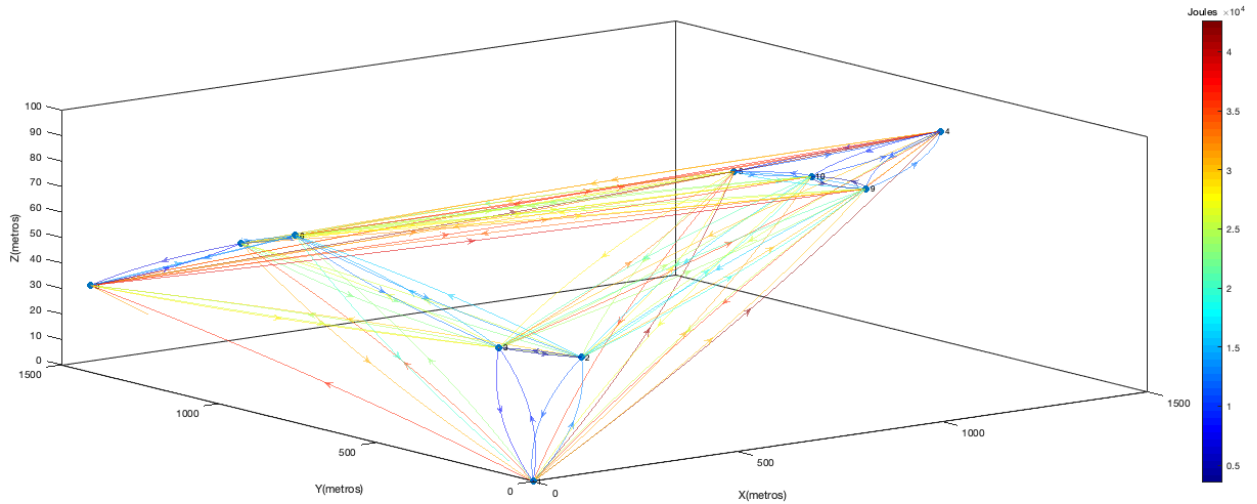


Figura 3.8: Grafo completo, ponderado e bidirecional.

Logo em seguida, com o dado de Potência auferido, pode-se calcular também o total de Energia, uma vez que a Potência nos informa sobre a quantidade de energia consumida por unidade de tempo. Esta quantidade de energia será utilizada para determinar a autonomia ( $A$ ) do drone no momento anterior ao vôo. Neste caso, é indicada a quantidade de energia armazenada na bateria em Watts-hora.

Para compatibilizar os cálculos do gasto energético do *drone* durante o deslocamento, feitos utilizando o Sistema Internacional de Medidas, converte-se a energia armazenada para Joules, multiplicando o valor em Watt-hora por 3600. Os dados utilizados em relação à autonomia das baterias, durante as simulações realizadas, se encontram na Tabela 3.1.

No cenário tratado pela modelagem em Programação Linear é definido que o *drone* irá visitar o ponto exato de localização tridimensional do sensor. Observa-se que os dispositivos IoT estão em alturas diferentes, e a cada altura foi atribuído um valor entre o intervalo de 30 e 100 metros. Este intervalo é definido pela legislação brasileira, para que os VANTs transitem sem que haja riscos de colisão com aeronaves comerciais ou ameaça às pessoas. Esse aspecto do cenário pode ser visto de uma forma como se esta localização indicasse um ponto de visita para cobrir um grupo de sensores clusterizados o que justifica a determinação de que o VANT visite o exato ponto também na coordenada ( $Z$ ). Porém esta etapa de clusterização não foi o foco deste trabalho.

Considerando que o *drone* irá percorrer um trajeto realizando subidas e descidas, assim como deslocamentos horizontais, é necessário calcular o gasto de energia que há de acordo com a natureza do movimento que a aeronave executa. Em (3-9) tem-se a equação geral que define o cálculo do consumo de energia ( $E_{traj}$ ) para que o *drone* efetue um trajeto específico. A Distância ( $D$ ) é resultado da distância euclidiana tridimensional entre dois pontos de localização de dois dispositivos IoT diferentes, portanto, a fórmula

(3-9) se refere a cada trajeto de deslocamento entre apenas dois dispositivos, e será usada para alimentar o conjunto ( $E$ ).  $P_v$  é a potência de consumo para o deslocamento vertical, e  $P_h$  é a potência de consumo para o deslocamento horizontal. Se houver diferença de altura  $\Delta h$  entre os dois pontos de movimentação, a velocidade correspondente ao movimento vertical e ao movimento horizontal serão respectivamente  $V_v = v \text{ seno } \phi$  e  $V_h = v \text{ cos } \phi$ , com  $\phi = \text{seno}^{-1} \left( \frac{\Delta h}{D} \right)$ .  $P_h$  é composto pela Força Parasita ( $P_p$ ) (3-10) e a Força Induzida ( $P_i$ ) (3-11), respectivamente para superarem o arrasto parasita e o arrasto induzido. O Arrasto Parasita é o arrasto, ou seja a resistência de atrito ao ar, provocado por todas as superfícies do avião que não produzem sustentação, já o Arrasto Induzido é provocado pela tendência que o ar tem em escoar da parte inferior para a parte superior das superfícies aerodinâmicas, devido a diferença de pressão existente entre elas, formando pequenos redemoinhos ou turbilhonamentos, que se não forem minimizados, afetarão a sustentação destas superfícies. Por isso o  $P_h$ , ou seja a força necessária para o VANT se locomover horizontalmente deve ser equivalente a ação dessas duas outras forças somadas, e dessa forma a Força Parasita ( $P_p$ ) e a Força Induzida ( $P_i$ ) se relacionam ao  $P_h$ . Estas fórmulas, bem como suas demonstrações, podem ser encontrados em [18].

$$E_{traj} = \frac{D}{v}(P_v + P_h) \quad (3-9)$$

A Força Parasita é dada por:

$$P_p = \frac{1}{2} \rho C_{D0} A_e v_h^3 \quad (3-10)$$

Onde  $v_h$  é a velocidade horizontal efetiva,  $C_{D0}$  é o coeficiente de arrasto,  $\rho$  é a densidade do ar,  $A_e$  é a área de referência, neste caso específico, se tratando da área frontal do VANT, esta formula não leva em consideração a força aplicada por ventanias contra o VANT, o que pode ser um fator complicador para se auferir os resultados práticos.

A Força Induzida é calculada através de:

$$P_i = \omega R_{ad} W \times \lambda \quad (3-11)$$

A equação é formada por  $\omega$  que é a velocidade angular,  $R_{ad}$  é o radio do disco rotor,  $W$  é o peso do *drone* e o  $\lambda$ , caracterizado por ser o fator da velocidade induzida, que será encontrado através da resolução da seguinte equação:

$$g(\lambda) = 2\rho\pi\omega^2 R_{ad}^4 \lambda \sqrt{\frac{v_h^2}{\omega^2 R_{ad}^2} + \lambda^2} - W = 0 \quad (3-12)$$

A potência de consumo vertical devido à ascensão e às descidas (assumindo descidas rápidas) é dada [18] pelas equações:

$$P_v = \frac{W}{2} v_v + \frac{W}{2} \sqrt{v_v^2 + \frac{2W}{\rho R_{ad}^2}}, \text{ascendendo.} \quad (3-13)$$

$$P_v = \frac{W}{2} v_v - \frac{W}{2} \sqrt{v_v^2 - \frac{2W}{\rho R_{ad}^2}}, \text{descendo.} \quad (3-14)$$

### Formulação do Problema do PCV com Localizações 3D

A formulação do problema de otimização permanece a mesma, ainda que se passe a considerar as localizações em três dimensões. Porém, o conjunto de arestas tem características diferentes pois começa a englobar pontos de três eixos, na forma (i, j, l). Além disso a matriz de distâncias é considerada, depois disso, uma matriz de custos energéticos dos trechos que o VANT pode percorrer. As arestas que, ao considerar pontos em 2D, possuíam mesmo valor pela equivalência das distâncias, por exemplo (i,j) e (j,i), são conferidos valores diferentes pois têm alturas diferentes, sendo assim, o valor do consumo de energia subindo será divergente do valor do consumo de energia descendo.

### Rota resultante do Plano de Voo em 3D

Para realizar as simulações são mantidas as distâncias entre os pontos e a área de alocação dos dispositivos, variando portanto somente as autonomias da aeronave para cada teste. É simulado um conjunto de 10 dispositivos IoT distribuídos numa área de  $1500m^2$  e três instâncias de autonomias diferentes, cada uma destas referindo -se à um modelo de bateria com especificações técnicas diferentes, demonstradas na Tabela 3.1. Os dispositivos IoT são alocados em uma área num raio de no máximo 1500 metros de distância, distribuídos a partir da definição de dos eixos x e y. Para os valores do eixo z, referente altura do sensor, foram definidos números no à intervalo entre 30 e 100 metros, atendendo aos requisitos de altura para voo do VANT na legislação brasileira. As distâncias entre os dispositivos IoT são estáticas, isto é, não mudam a cada voo. O nó inicial, que é o ponto de saída do *drone*, deve ser levado em consideração para compor o total de nós do conjunto de dispositivos IoT. O Ponto Inicial é um local onde será guardado o *drone*, e eventualmente pode-se efetuar recargas, preferencialmente no período onde não há voos programado, por isso neste cenário se é considerado vários voos executados sequencialmente, como por exemplo, num campus universitário ao longo de um período que não houveram recargas no VANT. Para o cálculo das potências de consumo vertical e horizontal, bem como o custo energético em cada uma dos caminhos do drone, são utilizados os parâmetros constantes da Tabela 3.2, sendo empregado as especificações do *drone* Intel Aero Ready to Fly Drone [29], inclusive para o requerimento das baterias utilizadas, sendo estas: Li-Po, 3S ou 4S e conector XT60.

Carga Elétrica	Tensão	Energia Convertida em Joules
1500Ah	11,1V	59940J
2500Ah	11,1V	99900J
4000Ah	11,1V	159840J

Tabela 3.1: Especificações das baterias LiPO simuladas

Parâmetro	Valor
$\rho$	$1,225kg/m^{-3}$
$C_{D0}$	0,0225
$A_e$	$0,07992m^2$
$\omega$	20rad/s
$R_{ad}$	0,5cm
$V$	15m/s
$W$	19N

Tabela 3.2: Parâmetros para cálculo do gasto energético do Drone.

Como foi especificado nas formulações do modelo de consumo energético, os cálculos de potência para movimento vertical assumem descidas rápidas. Porém, em alguns casos os cálculos padrões atribuem velocidade muito baixa na descida devido a grande proximidade horizontal entre os dois dispositivos IoT por onde o drone realiza seu deslocamento. Como a formulação não contempla esse caso de descida devagar, é atribuída a velocidade padrão do VANT utilizado, que é de 15m/s.

Na primeira instância (Figura 3.9), o *drone* tem uma autonomia total de 59940 Joules para serem usados no percurso, e considerando os 10 dispositivos IoT alocados na área, o *drone* só consegue visitar 3 destes, fazendo um percurso de 4 arestas para visitar tais dispositivos, sendo elas: (1,2), (2,3), (3,6) e (6,1). O gasto total da trajetória percorrida pelo *drone* é de 52713 Joules, restando 7227 Joules de autonomia que ainda poderiam ser usada pelo *drone*, porém, não é aproveitada pois não é suficiente para visitar mais dispositivos IoT, considerando não somente o gasto para chegada no sensor como também o gasto de volta até o Ponto Inicial.

Apesar da autonomia restante parecer alta, a otimização realizada concentra-se na quantidade de dispositivos IoT visitados, isto é, independente da quantidade de autonomia restante o número de dispositivos IoT visitados é o máximo que se conseguiria sobrevoar com a autonomia considerada. É possível que haja várias instâncias de consumos energéticos para a mesma quantidade de dispositivos IoT visitados considerando a mesma autonomia. Os custos energéticos entre os dispositivos IoT utilizados na primeira instância são os mesmos entre todas as demais instâncias simuladas dentro deste conjunto. Na trajetória ilustrada nas figuras é exibida a sequência de pontos visitados com a

sua localização especificada em relação ao Ponto Inicial e aos eixos do gráfico.

Quanto a segunda instância analisada (Figura 3.10), foi atribuído uma autonomia total do *drone* de 99900 Joules. Fazendo um aproveitamento dessa autonomia, o maior número possível de dispositivos IoT visitados foram 6. O total do gasto energético é de 94696 Joules com restante de autonomia de 5204 Joules.

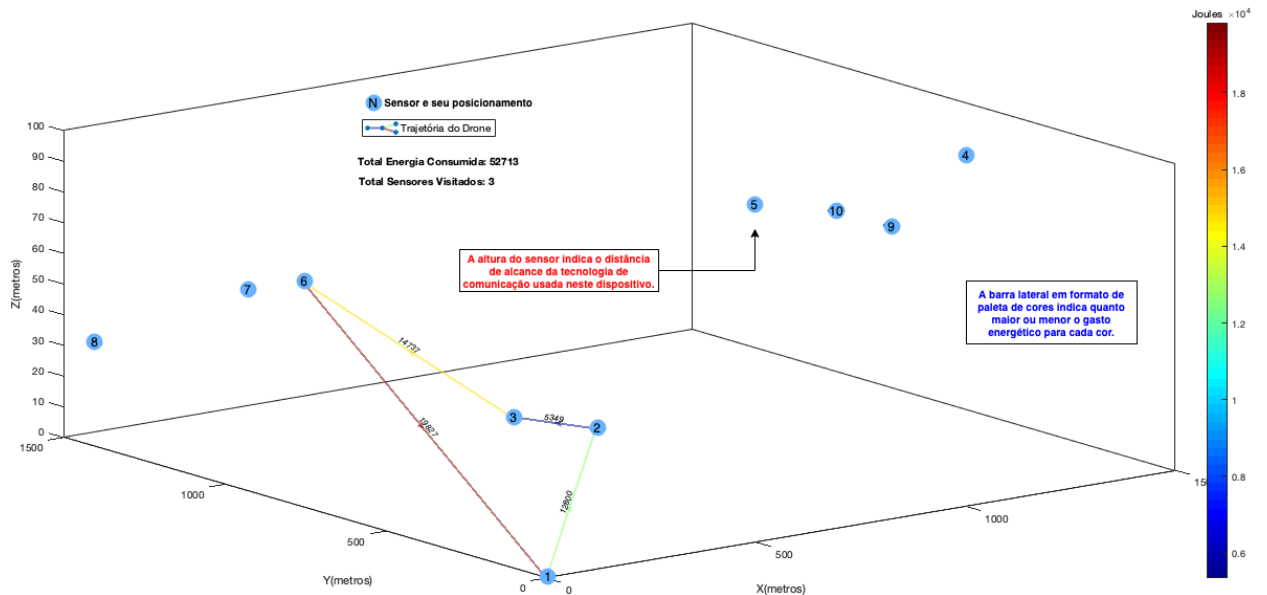


Figura 3.9: Trajetória do Drone para a bateria de 1500 mAh.

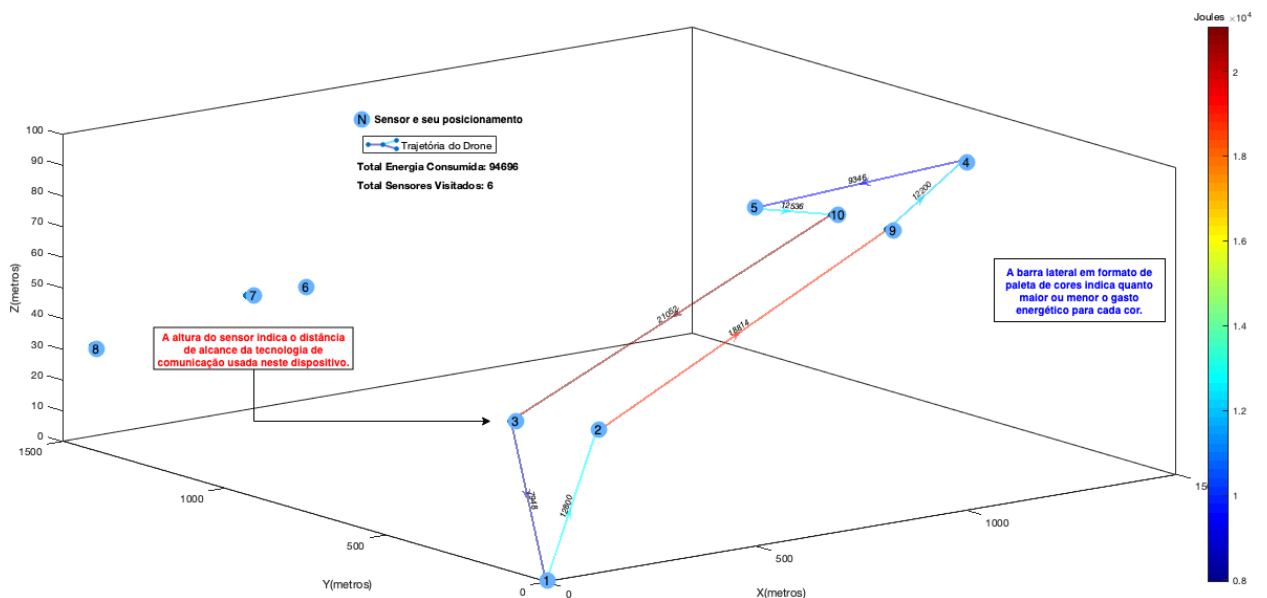


Figura 3.10: Trajetória do Drone para a bateria de 2500 mAh.

Para a terceira e última instância de autonomia analisada foram atribuídos 159840 Joules de autonomia para o *drone* (Figura 3.11). Nessa instância, o *drone* consegue sobrevoar 9 nós de dispositivos IoT dentro da alocação estabelecida, ou seja, todos

os dispositivos IoT alocados e o Ponto Inicial de saída do drone, totalizando 10 nós. Para efetuar toda a trajetória ele gasta 128202 Joules de autonomia com o restante de 31638 Joules. Não se pode afirmar que o custo do sobrevôo empregado equivale ao mínimo gasto para que o *drone* sobrevoe os dispositivos IoT dessa instância, pois as restrições não garantem que ele use o mínimo de energia para sobrevoar o máximo de dispositivos IoT. Esse gasto energético mínimo não é garantido pelo modelo de otimização proposto, pois caso fosse, se constituiria uma função multiobjetivo, o que não é o caso deste modelo.

Distribuindo os sensores por uma área de 1,5 km, esta abrangência de espaço passa a retratar mais fidedignamente as áreas encontradas nos campus universitários, assim como o campus da UFG. Uma vez que os cálculos de gasto energético do drone são feitos com detalhes específicos do VANT que o INF possui, toda simulação para a retratar mais realmente o cenário que onde voo pode ser realizado. O cálculo da energia gasta em cada trecho do trajeto é possível ser identificado nos gráficos, apresentando quais movimentos têm um custo energético maior para o VANT.

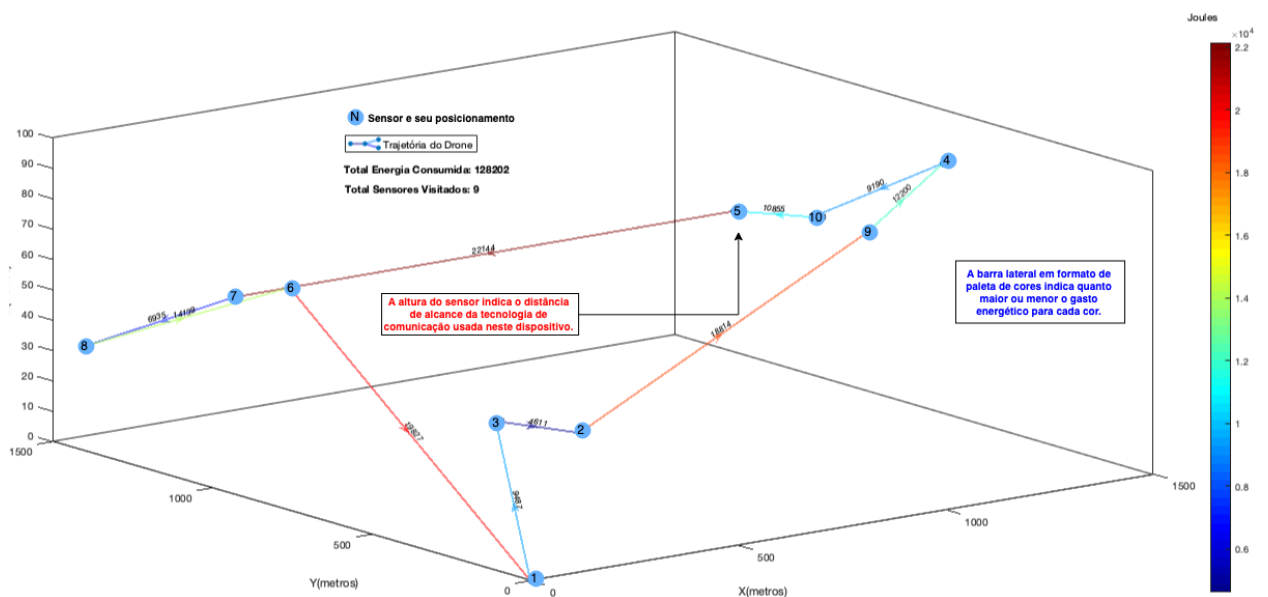


Figura 3.11: Trajetória do Drone para a bateria de 4000 mAh.

## 3.2 Clusterização dos Dispositivos IoT

Neste trabalho é utilizado o método de clusterização aglomerativo através do método *linkage complete* para se chegar a quantos *clusters* são possíveis dado um *threshold*, ou seja, um limite de distância entre os dois pontos mais longes de dois clusters prévios, que são anteriores à geração de um novo *cluster*. A partir deste parâmetro é possível construir um *cluster* com raio que atenda os limites de alcance de uma aeronave até todos os dispositivos daquele grupo a partir do centro do *cluster*, pois a distância entre

esses dos pontos será equivalente ao diâmetro ( $D$ ) do *cluster*, conseqüentemente sendo  $D = 2 \times r$  temos o raio ( $r$ ) de alcance a partir do centro do *cluster*.

A análise de *cluster* organiza um conjunto finito de objetos ou dados em subconjuntos que têm significado específico para um determinado cenário. Existem vários métodos e técnicas de computação que agrupam dados de maneiras diferentes. A escolha de um método preferido pode depender da cenário ou pode basear-se apenas na eficiência computacional do algoritmo. Antes de escolher um método de armazenamento em *cluster*, um método de medição da distância entre os objetos deve primeiro ser estabelecido. Considerando um ambiente que não apresente obstáculos na altura que o VANT voa, nem mesmo seja necessário realizar amaciamentos da trajetória criando curvas ou *flips* será utilizado o método euclidiano. Embora simplista, este cenário é suficiente para entender métodos hierárquicos de clusterização que são usados, neste exemplo, para ampliar as limitações no tempo de computação existentes na PLIM.

Os métodos hierárquicos de clusterização incluem ambos os métodos divisivos e aglomerativos. Métodos aglomerativos iniciam o processo de organizar os dados atribuindo cada objeto ao seu próprio *cluster* e continua mesclando *clusters* de acordo com a matriz de proximidade [73]. Os métodos divisivos são semelhantes aos aglomerativos, mas funcionam ao contrário, começando atribuindo todos os objetos ao mesmo *cluster* e separar os *clusters* até que cada objeto seja único *cluster*. O lado esquerdo do dendrograma (Figura 3.12), gerado pelo algoritmo de clusterização aglomerativo, fornece a proximidade em quais os destinos foram agrupados. Para produzir esse dendrograma, a matriz de proximidade é iterativamente alterada até que todos os destinos sejam combinados em um *cluster* (usando um método aglomerativo).

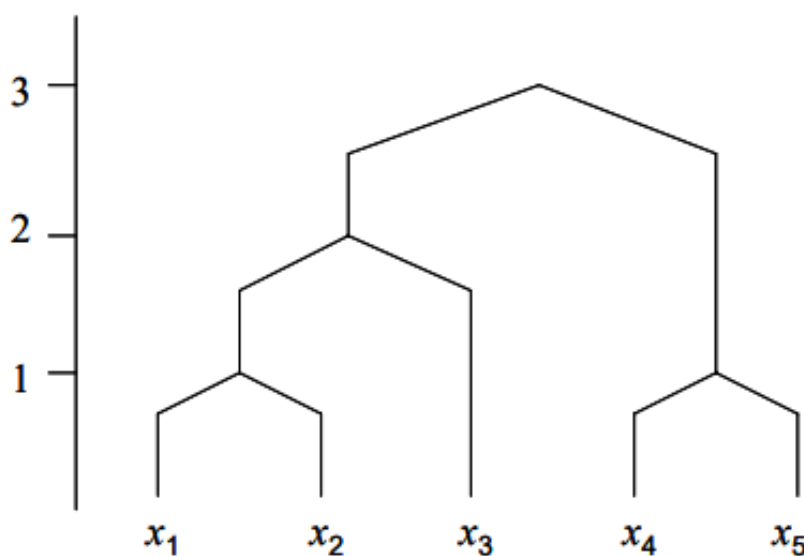


Figura 3.12: Dendrograma de exemplo em uma clusterização aglomerativa.

Existem vários métodos para combinar *clusters* na abordagem aglomerativa,

entre eles o *single*, *complete*, *Average Link* e o *Ward's Method*. O mais apropriado para nessa aplicação é o *complete* pois produz um diâmetro consistente em cada *cluster*, dessa é possível limitar também o raio a partir do centro do grupo de dispositivos. O raio é considerado dentro do limite de alcance da comunicação dos dispositivos com a aeronave. No caso da tecnologia de comunicação de exemplo temos o *Bluetooth Low Energy* com um alcance em média de 60 metros, por tanto o limite de diâmetro atribuído ao algoritmo é de 120 metros. O método é usado para calcular a distância  $d(s, t)$  entre dois *clusters*  $s$  e  $t$ . O algoritmo começa com uma floresta de *clusters* que ainda precisam de ser usados na hierarquia que está sendo formada. Quando dois *clusters*  $s$  e  $t$  desta floresta são combinados em um único *cluster*  $u$ ,  $s$  e  $t$  são removidos da floresta e  $u$  é adicionado. Quando apenas um *cluster* permanece na floresta, o algoritmo para e esse *cluster* se torna a raiz. A seguir, é apresentado como o método *complete* para calcular a distância entre o *cluster* recém-formado  $u$  e cada  $v$ . Para todos os pontos  $i$  no *cluster*  $u$  e  $j$  no *cluster*  $v$ . Ele também é conhecido como *Farthest Point Algorithm* (Algoritmo de Ponto mais Distante):

$$d(u, v) = \max(\text{dist}(u[i], v[j])) \quad (3-15)$$

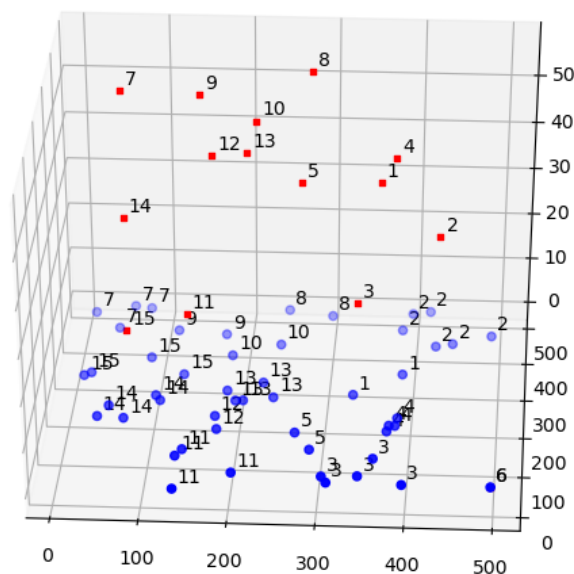


Figura 3.13: Clusterização com pontos alocados em azul e os *centroids* de cada grupo em vermelho.

A partir da dos grupos de dispositivos estabelecidos pelo método de clusterização dentro do diâmetro dado como entrada ao algoritmo é extraído o diâmetro de fato, na maioria das vezes menor do que o que foi colocado como limite (*threshold*). Portanto, para se chegar à altura ideal que a aeronave deve pousar quando realizar a sincronização de dados é usada a fórmula  $h = \sqrt{60^2 - r^2}$  sendo a raio  $r$  o valor real do raio daquele

*cluster* específico. A fórmula que é deduzida a partir do Teorema de Pitágoras, considera que a hipotenusa será a distância máxima permitida, dado o alcance da tecnologia de comunicação, para que a aeronave possa receber a transmissão de dados. A clusterização mostrada na figura 3.13, foi gerada a partir da alocação de 50 pontos distribuídos num espaço de 500 metros resultando em 15 grupos de sensores.

### 3.3 Plano de voo com múltiplos VANTs com capacidade de armazenamento

#### 3.3.1 Formulação do problema

Seja o conjunto  $K = \{1, 2, \dots, n\}$  o qual temos o total dos drones utilizados para realizar a missão a ser otimizada. Considere um conjunto de *clusters*  $S = \{0, 1, 2, \dots, n\}$ , estes sensores estão espalhados em um campus de forma aleatória, estabelecendo entre cada um desses sensores um arco de distância. Para fins de modelagem, o conjunto  $S$  possui o número de elementos igual ao número de sensores somado o ponto inicial, ou seja o elemento 0 é um nó virtual que identifica o local de onde a aeronave decola. Considerando que os *drones* podem ter características heterogêneas, o consumo energético entre os sensores é representado neste caso pelo conjunto  $E = \{E_{0,0}^1, E_{0,1}^1, \dots, E_{n-1,n}^k\}$ , neste conjunto, um elemento  $E_{i,j}^k$ , tem um  $i, j \in S$  e  $k \in K$ . Nos casos de  $i = j$  o custo energético é zero. Cada *drone* que executa o voo nessa trajetória possui uma autonomia específica no momento anterior ao voo. Para isso apresentamos o conjunto  $C = \{C_1, C_2, \dots, C_n\}$ , cada elemento  $C_k$  representa a autonomia específica do VANT  $k$ . O conjunto apresentado anteriormente é adicional ao que é descrito convencionalmente no PRVC, pois no modelo clássico o objetivo é minimizar a distância percorrida pelos veículos no total, porém, neste trabalho como forma de prevenção, o objetivo é maximizar a quantidade de sensores visitados. Dito isto, é necessário restringir o gasto energético total de cada veículo à autonomia que cada um possui. O atributo capacidade do enunciado do problema neste caso é tratado como a capacidade de armazenamento de dados. A maior parte dos *drones* comerciais que estão no mercado é vendido com memória flash acoplada, estas memórias possuem custo elevado e são cedidas com espaço de armazenamento pequeno, porém são utilizadas pois seu peso, sendo pequeno, não influenciam tanto no peso total do *drone*, portanto auxiliando que o veículo mantenha sua autonomia total razoável. Os sensores ou *clusters* de sensores têm um *buffer* máximo de armazenamento dos dados que serão coletados. Para isso apresentamos o conjunto  $q = \{q_1, q_1, \dots, q_n\}$ , cada elemento  $q_s$  representa o *buffer* total em cada *cluster* específico. Para identificar a capacidade de armazenamento da memória flash que o VANT carrega tem-se o conjunto  $Q = \{Q_1, Q_1, \dots, Q_n\}$ , para cada uma das aeronaves.

Para cada um dos trechos que é computada no roteamento do voo é necessário determinar uma variável binária que especifica se a aresta  $(i, j)$  será percorrida, ou não, pelo drone  $k$ . Esta variável de decisão é dada em (3-16):

$$X_{ij}^k \in \{0, 1\}, \forall i, j \in S, \forall k \in K \quad (3-16)$$

Para este contexto o maior interesse é que seja visitado o maior número de sensores ou *clusters* possíveis, considerando que os veículos possuem limitação de bateria. Ainda, temos para este cenário, que os VANTs deverão decolar do primeiro ponto e encerrar o voo voltando para o mesmo ponto, ou seja, esta coordenada é definida também como a posição de retorno dos VANTs. Dado um conjunto  $A_r$  de arestas percorridas, para se saber o número de pontos visitados é necessário a subtração de  $A_r - k$ , pois as últimas arestas percorridas são relativas à volta dos *drones* para o primeiro ponto. O objetivo da otimização modelada é maximizar o número de pontos visitados através da maximização do somatório de arestas percorridas. Este objetivo é demonstrado em (3-17):

$$\text{Maximizar} \quad \sum_{i, j \in S, k \in K} X_{ij}^k \quad (3-17)$$

A restrição (3-18) garante que o somatório das vezes que um determinado VANT chega em um ponto através de uma aresta, por exemplo  $(i, j)$  deve ser igual ao somatório das arestas de saídas do mesmo ponto, ou seja  $(j, i)$ , sendo esta saída executada pelo mesmo *drone* que executou a entrada. Essa restrição é importante pois em cada sensor ou *cluster* que um *drone* entre sobrevoando, percorrendo a trajetória de chegada naquele ponto, o *drone* deve, para este, percorrer a trajetória de saída, e não permite que outro veículo diferente daquele que visitou inicialmente aquele ponto, possa, por ventura realizar uma trajetória saindo do mesmo local. Entende-se por trajetória de saída a aresta que vai do último ponto visitado por um *drone* até o próximo ponto de sensor que será visitado.

$$\sum_{i \in S, i \neq j} X_{ij}^k = \sum_{i \in S, i \neq j} X_{ji}^k, \forall j \in S, \forall k \in K \quad (3-18)$$

As restrições (3-19) e (3-20) determinam que cada veículo deve sair exatamente uma vez do ponto inicial e voltar também uma vez.

$$\sum_{i \in S, i \neq 0} X_{0i}^k = 1, \forall k \in K \quad (3-19)$$

$$\sum_{i \in S, i \neq 0} X_{i0}^k = 1, \forall k \in K \quad (3-20)$$

A restrição (3-21) dá a possibilidade de que um ponto seja sobrevoado ou não, pois neste cenário é permitido que nem todos os pontos sejam visitados, considerando que

o veículo não tenha autonomia para isso. Porém há o limite de visitação de no máximo uma vez. Esta restrição não seria colocada no problema clássico do PRV, que considera obrigatório todos os pontos serem visitados.

$$\sum_{k \in K} X_{ij}^k \leq 1, \forall i, j \in S, i \neq j \quad (3-21)$$

Para que o *drone* armazene os dados coletados de todos os *clusters* visitados, a quantidade total de dados coletados por um VANT específico deve ser menor ou igual ao armazenamento disponível naquele *drone* tal como formaliza a restrição (3-22). O consumo energético total ao visitar os pontos e pairar para realizar a sincronização deve ser menor que a autonomia no veículo específico (3-23).

$$\sum_{i, j \in S, i \neq j} q_j X_{ij}^k \leq Q_k, \forall k \in K \quad (3-22)$$

$$\sum_{i, j \in S, i \neq j} e_{ij}^k X_{ij}^k \leq C_k, \forall k \in K \quad (3-23)$$

Dado um subconjunto  $R$  clusters dentro do conjunto total, neste, o total de arestas percorridas deve ser igual o total de vértices menos um, para que não se feche uma sub trajetória dentro daquele conjunto de clusters (3-24). Esta restrição atua para evitar qualquer desperdício na autonomia restante do *drone* durante o voo, uma vez que se um sensor já tiver sido visitado anteriormente não é necessário que se retorne ao mesmo.

$$\sum_{k \in K} \sum_{i, j \in S, k \in K} X_{ij}^k \leq |R| - 1, R \subseteq K \quad (3-24)$$

### Cálculo do *hovering* do VANT

A tecnologia considerada para conexão entre o VANT e os sensores é a BLE (*Bluetooth Low Energy*) que possui *Air Data Rate* igual a 1Mb/s [61]. Seja o conjunto  $H = \{H_1, H_1, \dots, H_n\}$  que traz os valores referentes ao consumo energético do drone durante a planação em que é transmitido os dados contidos no buffer dos sensores ou clusters. O cálculo da potência de consumo energético durante o hover específico do drone é mostrado a seguir. Para saber quanto tempo necessário para transmitir todo o buffer coletado pelos sensores é feita a divisão:  $Buffer/DataRate$ . Logo em seguida para aferir o consumo energético da planação é utilizada a fórmula clássica de energia:  $E = P \times t$ . Esse conjunto de valores resultantes do consumo energético a partir do momento de transmissão de dados são acrescidos aos valores de consumo de dados para visita em um determinado ponto, fazendo com que o modelo de otimização considere todo o gasto energético como um conjunto apenas.

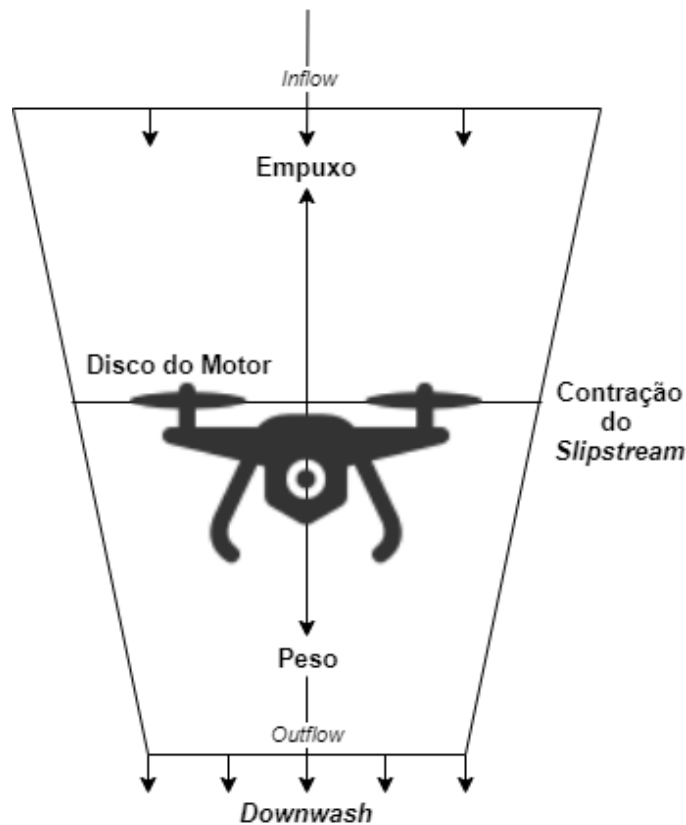


Figura 3.14: Condição de *hover*, com tubo de fluxo ao redor do disco do rotor e estrutura.

O método mais simples para obter uma estimativa de primeira ordem da potência necessária para o *hover* é a Teoria Unidimensional do Momento Axial. O *Slipstream* tem um eixo vertical através do centro do rotor. O *inflow* é do topo da figura. A contração do *Slipstream* é explicada com os mesmos conceitos usados para a hélice. Quando levamos em conta essas alterações para aplicação ao helicóptero em suspensão, a Teoria Unidimensional do Momento Axial fornece a velocidade induzida no motor em *hover*:

$$V_h = \sqrt{\frac{T}{2\rho A}} = \sqrt{\frac{W}{2\rho A}} \quad (3-25)$$

A potência ideal necessária para pairar é dada pelo produto entre o empuxo e a velocidade induzida, ou seja:

$$P_h = T \times V_h = W \sqrt{\frac{W}{2\rho A}} = \frac{W^{\frac{3}{2}}}{\sqrt{2\rho A}} = \sqrt{\frac{W}{2\rho}} \sqrt{\frac{W}{A}} \quad (3-26)$$

Onde a área do disco é  $A = \pi R^2$ . Concluimos que a potência induzida para pairar aumenta com o peso da aeronave e diminui com o aumento do raio do rotor (todos os outros parâmetros sendo constantes). Isso implicaria que o aumento do diâmetro seria benéfico considerando o peso constante, o que não é sempre o caso.

### Eficiência Energética

É proposto um segundo modelo com objetivo na equação 3-27. Com o resultado do valor ótimo de quantidade máxima de sensores que podem ser visitados é possível se chegar a um valor mínimo de gasto energético mantendo o mesmo número de sensores visitados. A abordagem é feita em duas fases conforme demonstra o algoritmo (1), a restrição 3-28 só é adicionada a partir do momento em que o primeiro passo na função *maximoSensores* é concluído, ou seja, o segundo modelo depende do resultado extraído da primeira função objetivo. Logo, com o valor conhecido referente ao máximo de sensores possíveis de serem visitados de acordo com as autonomias dos veículos, é possível também se chegar ao mínimo consumo energético para aquela quantidade de sensores.

$$\text{Minimizar } \sum_{(i,j,k) \in E} E_{ji}^k X_{ji}^k \quad (3-27)$$

$$\left( \sum_{(i,j,k) \in E} X_{ij} \right) = \text{maxSensores} \quad (3-28)$$

---

**Algorithm 1:** Abordagem para resolução do problema da eficiência energética

---

```

1 Function maxSensoresMinEnergia (sensores, drones) :
2   max = maxSensores(drones, sensores)
3   min = minEnergia(drones, sensores, max)
4   return max, min
5
```

---

### 3.3.2 Avaliação

Na simulação apresentada nos gráficos a seguir foram considerados 2 VANTs e a capacidade de armazenamento igual para cada um dos veículos, ambos com 4000mb cada. Foram simulados os valores de 5, 7, 9 e 11 de pontos. Também foi considerado que cada veículo possuía uma autonomia total de 30000J de energia para ser gasta. As constantes e indicadores da aerodinâmica da aeronave são os mesmos utilizados na simulação do modelo anterior.

É possível notar entre os gráficos nas Figuras (3.15) e (3.16), que o consumo de energia sofre alteração, e é diminuído entre os passos já descritos na seção anterior, conforme é o objetivo da otimização da eficiência energética minimizando o consumo no máximo de sensores possíveis de serem visitados. No último ponto dos gráficos é possível ver que há uma queda no consumo similar nas duas figuras, essa queda acontece

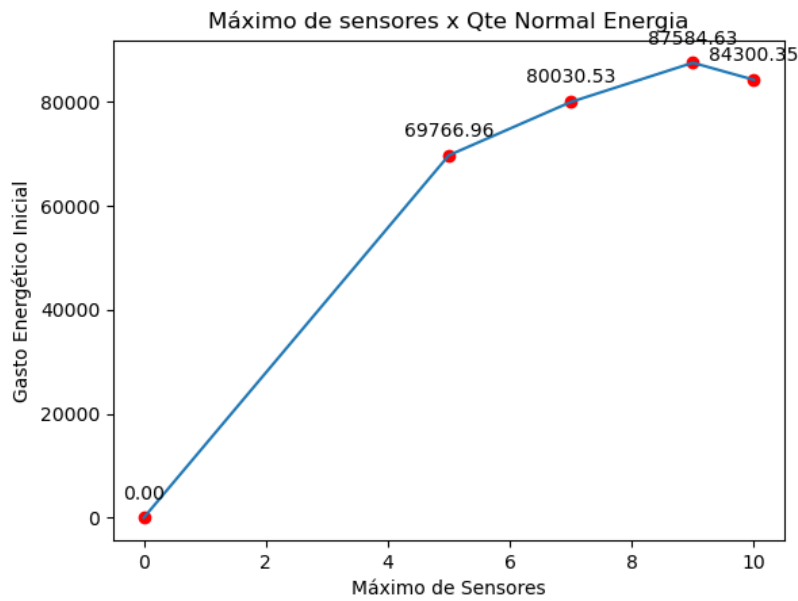


Figura 3.15: Máximo de sensores x Quantidade Normal Energia

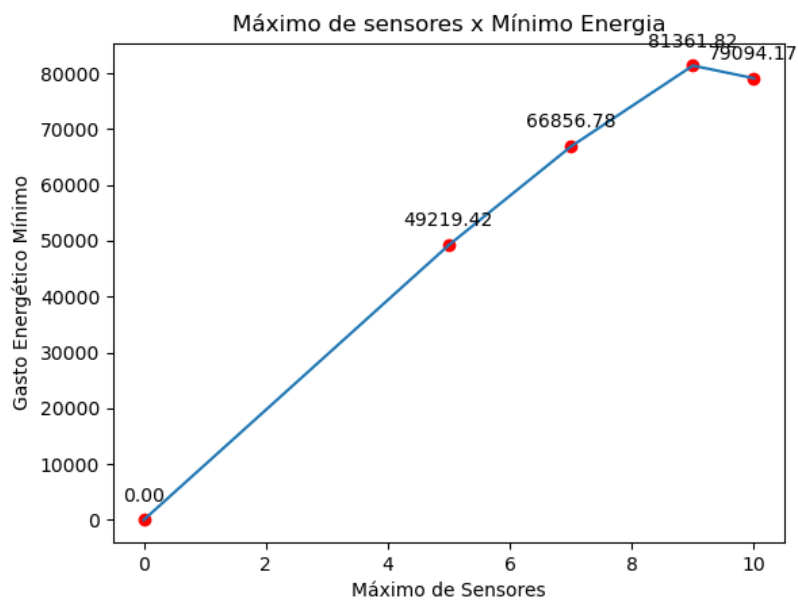


Figura 3.16: Máximo de sensores x Mínimo Energia

por que mesmo que a quantidade de sensores visitada é maior que no conjunto simulado anterior a partir do momento que o espaço amostral vai aumentando mais possibilidades de minimizar o consumo de energia aparecem.

O que vemos nas Figuras (3.17) e (3.18) é que a quantidade de dados coletados tem um crescimento quase linear e na verdade tem uma distribuição aproximadamente igualitária entre os dois veículos, isso se dá por que tanto a carga quanto a capacidade de

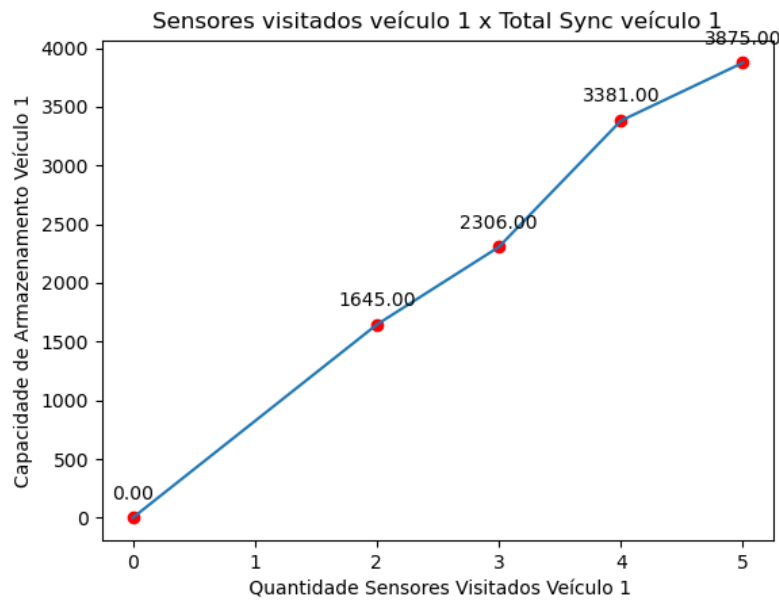


Figura 3.17: Sensores visitados veículo 1 x Total Sync veículo 1

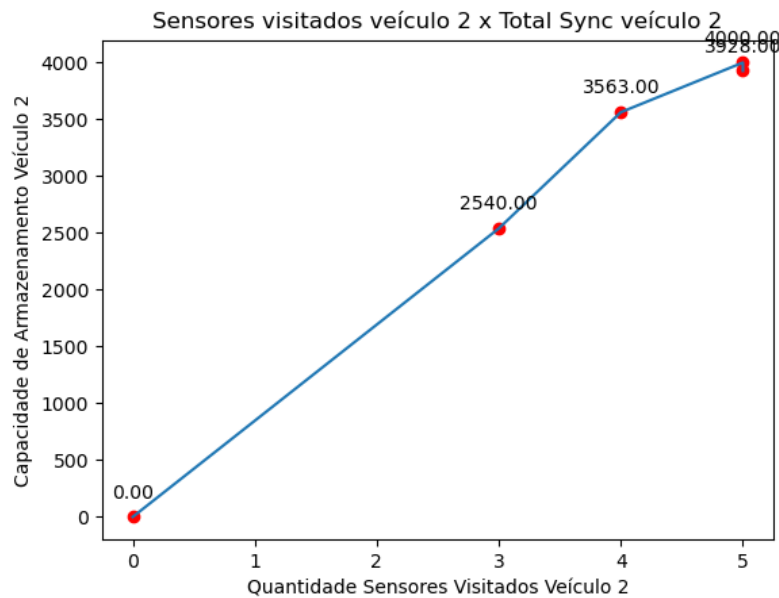


Figura 3.18: Sensores visitados veículo 2 x Total Sync veículo 2

armazenamento dos dois é setada com valores iguais para ambas aeronaves.

Nas Figuras (3.17) e (3.18) é comparado o total de dados coletados por cada um dos veículos, e à medida que o número de nós simulados aumenta o volume de dados naturalmente também se eleva. Apesar de o VANT 2 apresentar maior volume de dados coletados do que o VANT 1 não há nenhum ponderamento no modelo que provoca isso, e a capacidade de armazenamento dos dois veículos é igual. Uma nova simulação é feita,

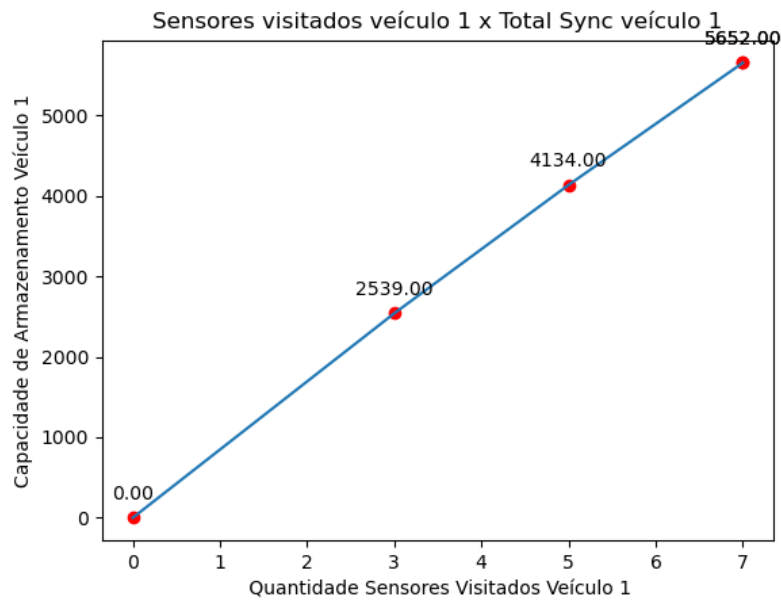


Figura 3.19: Sensores visitados veículo 1 x Total Sync veículo 1

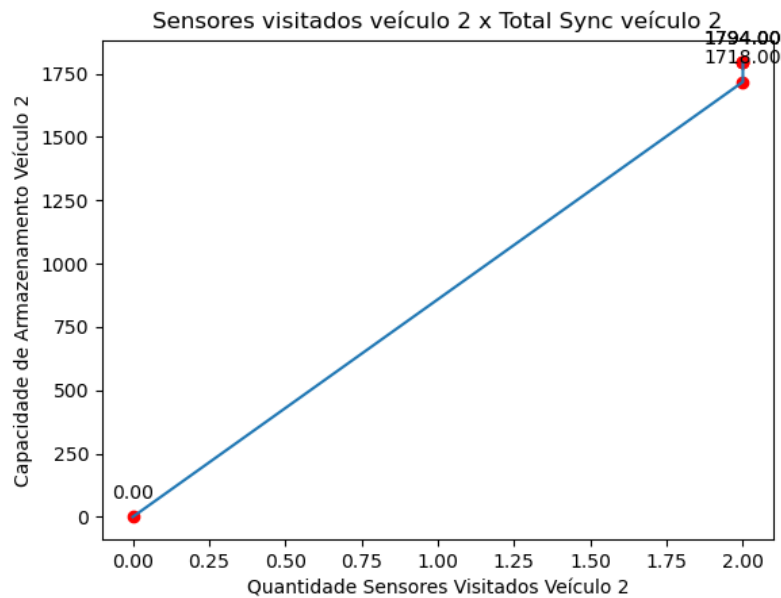


Figura 3.20: Sensores visitados veículo 2 x Total Sync veículo 2

conforme mostra as Figuras (3.19) e (3.20) mantendo todos os dados anteriores, porém alterando apenas a capacidade total de armazenamento, neste caso o VANT 1 passa a ter 6000mb de espaço, e já o VANT 2 passa a ter 2000 mb de espaço, a partir disso já notamos que o modelo faz a devida distribuição no roteamento para que cada veículo não sobrecarrega seu espaço de armazenamento.

Nas Figuras (3.21) e (3.22) tem-se as rotas geradas pelos casos anteriores quando

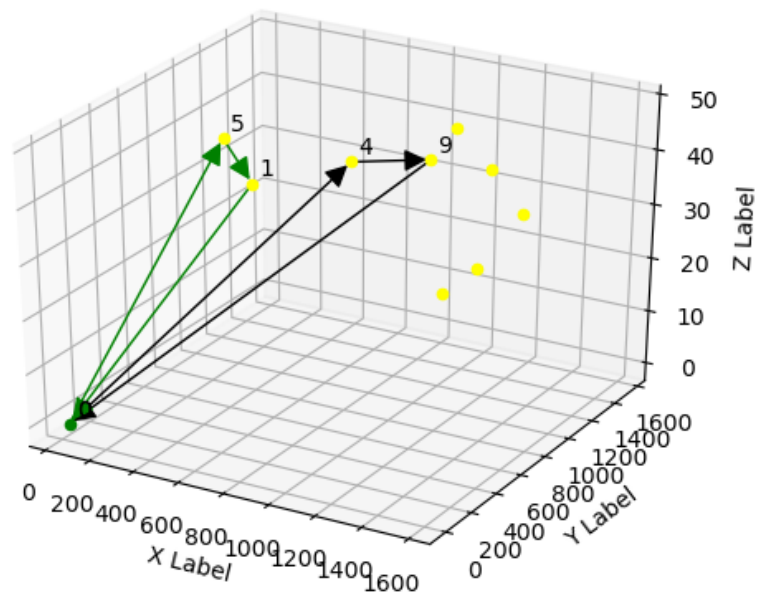


Figura 3.21: Rota com 9 pontos para visitaç o com capacidade de armazenamento igual.

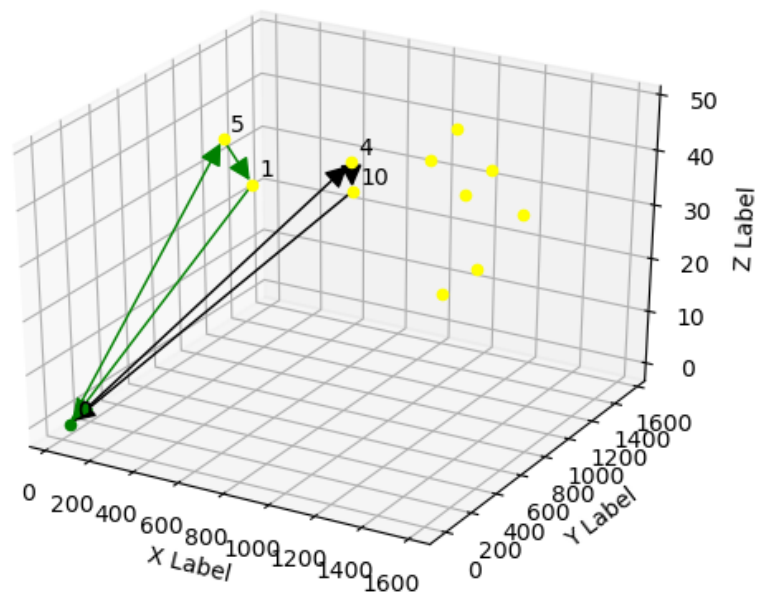


Figura 3.22: Rota com 11 pontos para visitaç o com capacidade de armazenamento igual.

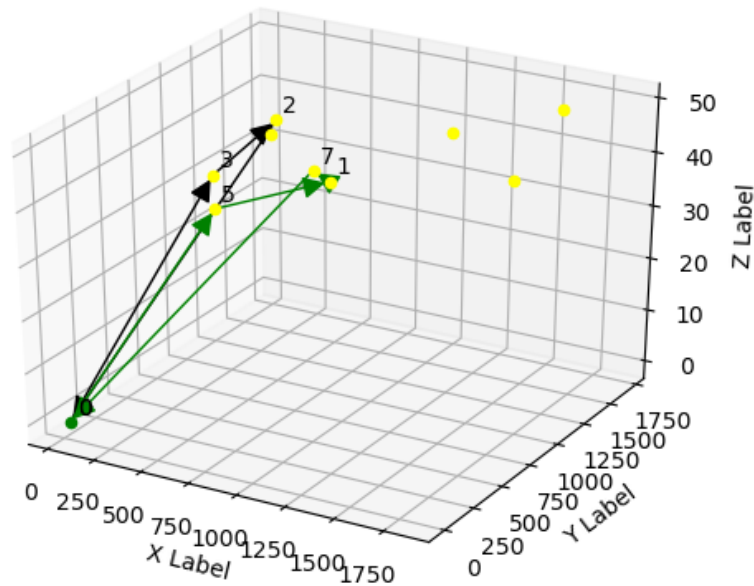


Figura 3.23: Rota com 9 pontos para visitaç o com diferena de capacidade de armazenamento.

a capacidade de armazenamento era semelhante observa-se que a quantidade de sensores visitados tamb m permanece a mesma entre os dois ve culos. No entanto nas Figuras (3.23) e (3.27)   notado que apesar da variao grande de capacidade de armazenamento a quantidade de sensores entre os dois n o se diferente tanto, o que acontece   que o modelo apenas seleciona uma soluo  tima que em que passe por n s onde h  maior tamanho de *buffer*.

Nas Figuras (3.25) foram simulados 4 inst ncias alocando 5, 7, 9 e 11 sensores. Cada inst ncia   rodada ao mesmo tempo no modelo com multi ve culos e tamb m a mesma no modelo com  nico ve culo. Nessa comparao   atribu da a mesma autonomia de 80000 joules para cada ve culo.   poss vel observar as constataes naturais de que no modelo com m ltiplos ve culos considerando tamb m que a inst ncia utiliza duas aeronaves   poss vel sobrevoar maior n mero de sensores, no entanto o gasto energ tico fica obrigatoriamente maior, neste caso, por que o modelo obriga que as duas aeronaves saiam e retornem pelo menos uma vez, uma melhoria seria a possibilidade de que o modelo atribua as tarefas para apenas um VANT caso identificasse que dessa forma gastaria um total menor de energia.

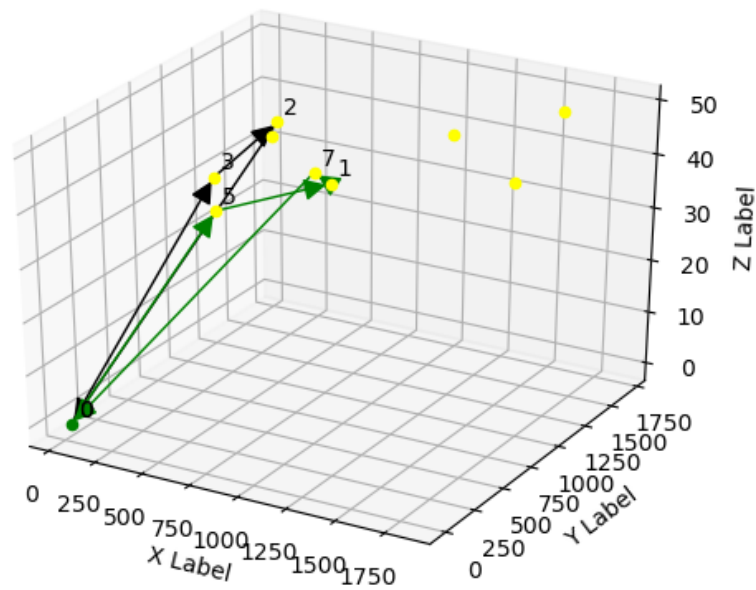


Figura 3.24: Rota com 11 pontos para visitação com diferença de capacidade de armazenamento.

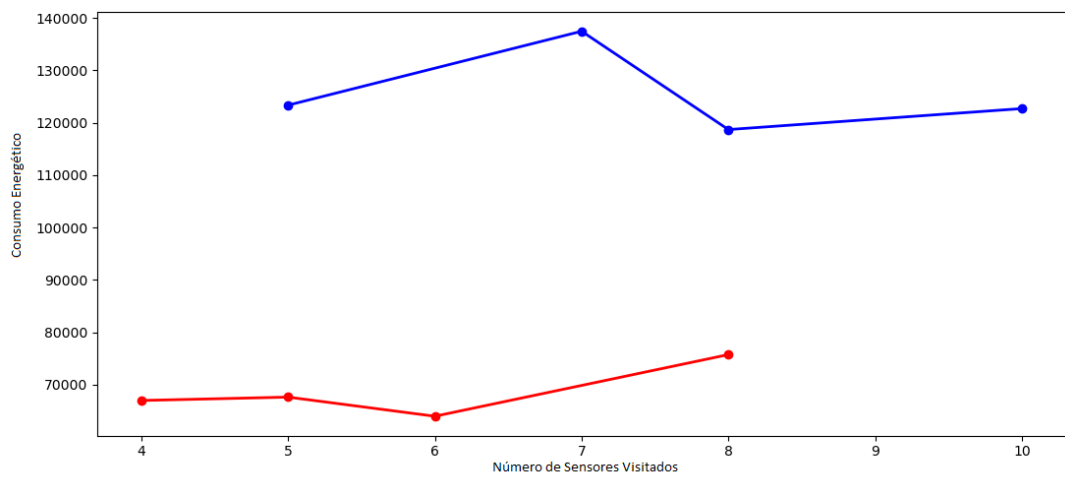


Figura 3.25: Comparação entre o modelo com único veículo e multi veículos.

## Desempenho

Para extrair dados para analisar o desempenho do modelo no decorrer das simulações foram utilizadas três métricas dos resultados: tempo de resposta, taxa de confiabilidade (com variação de autonomia e de pontos de visitação) e também a quantidade de

consumo energético. Quanto a taxa de confiabilidade nos dois casos usados ela é dada conforme é mostrado na equação 3-29 [32] onde  $R$  é a quantidade de replicações da simulação, e  $n$  o número de falhas, ou seja o modelo não possui solução ótima para os dados de entrada. Esta situação pode acontecer pois a premissa básica é que cada um dos veículos deve ao menos uma vez sair e retornar ao ponto inicial, no entanto pode haver situações em que o veículo não terá autonomia suficiente nem mesmo para executar tais tarefas, portanto a simulação não irá apresentar nenhum resultado factível.

$$r = \frac{R - n}{R} \quad (3-29)$$

São extraídas 50 amostras de valores médios a partir de outras replicações das simulações. Cada uma das amostras é retirada de uma simulação executada a partir de uma instância diferente, considerando novos pontos de localização dos sensores, em outros casos variações de autonomia dos veículos também. Para serem aferidas as métricas de tempo de resposta e consumo energético é mantido uma autonomia constante que permita a todas as instâncias retornar solução ótima. Para as simulações com variação da autonomia é gerado um valor randômico entre 35000 e 60000 joules. É utilizada a fórmula do intervalo de confiança para a média das amostras conforme mostra a equação 3-30, onde o  $\mu$  é a média aritmética das amostras, o  $t$  é a função *t-distribution* que depende do  $\alpha$ , chamado nível de significância, e do  $df$  que é o nível de liberdade. O elemento  $s$  se trata da variância da amostra e o  $n$  é o tamanho do conjunto de amostras.

$$I.C. media : \mu \pm (t_{(1-\frac{\alpha}{2}, df)}) \times \frac{s}{\sqrt{n}} \quad (3-30)$$

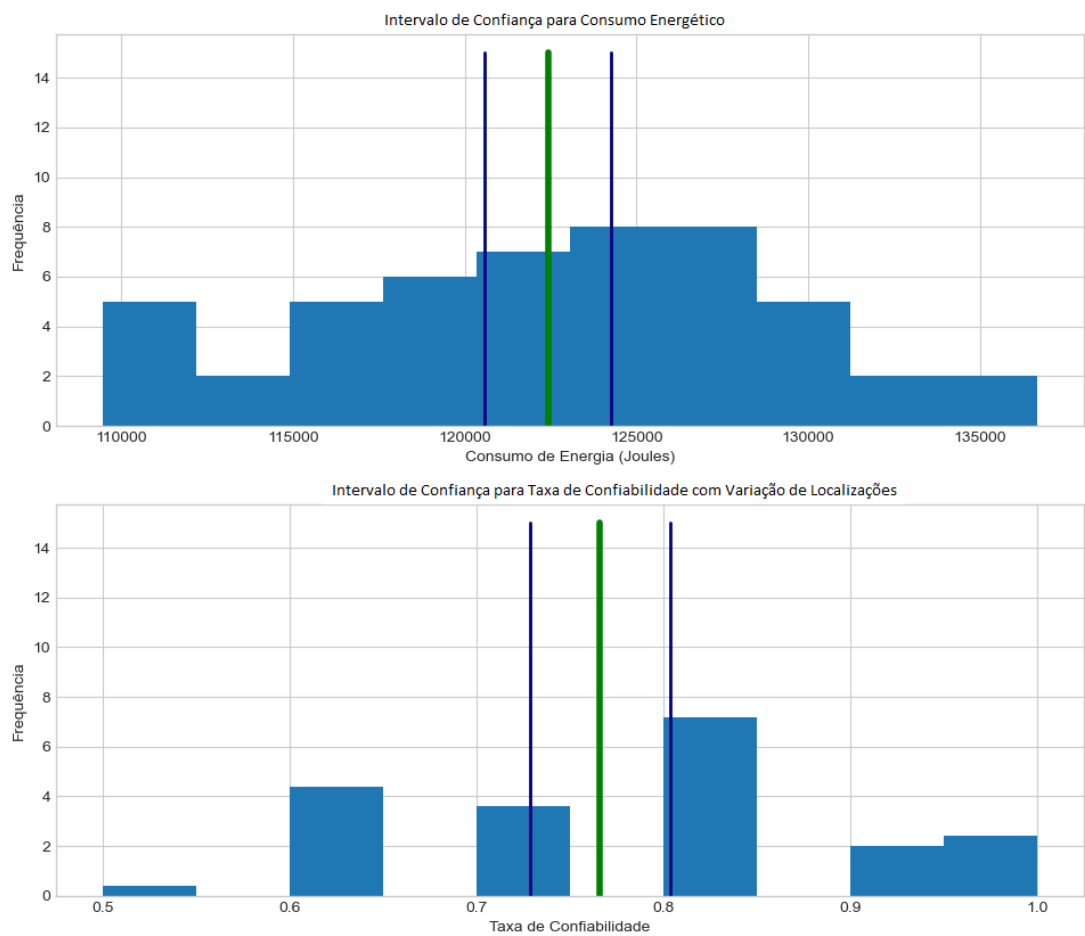


Figura 3.26: Histograma com Consumo Energético e Taxa de Confiabilidade Variando Localizações.

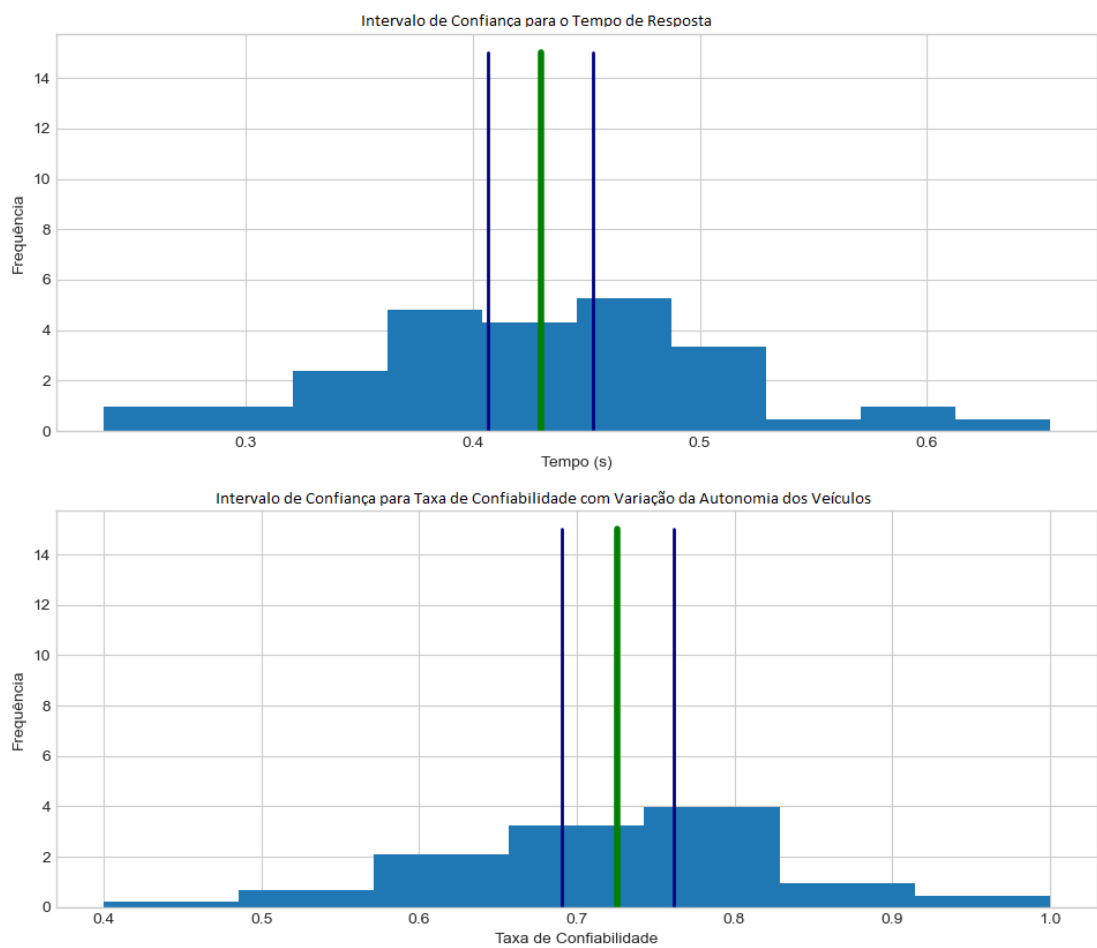


Figura 3.27: Histogramas com Tempo de Resposta e Taxa de Confiabilidade Variando Autonomias .

## Arquitetura *Drone-as-a-Service* e Plataforma de Serviços

---

Neste capítulo é apresentada uma arquitetura orientada à serviço para controle e gerenciamento de VANTs usados como mula de dados com finalidade de prover estes dados coletados à aplicações clientes. No contexto da arquitetura, um componente fundamental é a otimização da trajetória necessária para atender às requisições realizadas por aplicações externas que queiram utilizar os drones para coletar dados do grupo de sensores controlados pelo serviço. O objetivo é que a partir da arquitetura possa ser desenvolvido uma nova plataforma IoT que seja disponibilizada pela infraestrutura de *cloud computing*, e que haja troca de mensagens utilizando protocolo HTTP, e que através desses serviços sejam compartilhadas as funções de comunicação de um ou mais VANTs com sensores IoT.

### 4.1 Cenário

Dentre as várias aplicações dos VANTs, uma em especial pode ser definitivamente importante para o contexto de ambientes inteligentes: a coleta de dados. Esta função trata da captura dos dados gerados por sensores IoT espalhados por exemplo por uma área geográfica tal como cidade ou campus universitário. Estes dispositivos que não possuem conectividade móvel ou alcance ao (*gateway*) terrestre. O uso de um VANT para se conectar com cada um dos sensores com o propósito de coletar seus dados é denominado *Data MULE (Mobile Ubiquitous LAN Extension)*, ou ainda, Mula de Dados. À medida que diversos usuários e aplicações consomem os dados coletados por este VANT, é necessário oferecer uma interface para controlar estes clientes e também gerenciar os recursos do VANT que é compartilhado.

A estrutura que permite o uso desses recursos compartilhados é chamada DaaS (*Drone-as-a-Service*). Um DaaS é uma interface que permite que um ou mais usuários solicitem espontaneamente um ou mais serviços. O serviço solicitado é recebido e um ou mais drones são selecionados automaticamente no lado do serviço do drone, com base na

solicitação de serviço do usuário. Após a conclusão do serviço, os resultados são entregues de forma automática e conveniente ao usuário [22].

A IoD (*Internet of Drones*) pode ser definida como uma arquitetura para prover controle e acesso entre VANTs e usuários através da internet. A IoD provê uma maneira, através da infraestrutura *cloud*, para escalar o processamento e armazenamento remoto de dados descarregados pelos VANTs [36].

Pode-se enfatizar dois principais benefícios da integração de drones com infraestrutura IoT e cloud[36]:

- (1) Virtualização: a infraestrutura *cloud* ajuda a virtualização dos recursos dos VANTs através de interfaces abstratas.
- (2) Offloading computacional: a *cloud* executa o papel de um cérebro remoto para os VANTs provendo serviços de otimização, armazenamento e processamento.

## 4.2 Arquitetura

Apoiado no contexto descrito acima, este capítulo visa apresentar uma plataforma de acesso remoto a dados de dispositivos IoT, aplicando o conceito de DaaS. A arquitetura é baseada em várias camadas portanto sendo do tipo *n-Tier*. A comunicação com a plataforma se dá através da subscrição de clientes à interface de serviços conforme mostrado na figura pelo componente *Services*. Outro componente da arquitetura é responsável por realizar um plano de vôo (*Flight Plan*) a partir da otimização da trajetória de um VANT que realiza voo como mula de dados o motor de otimização é representado na figura pelo o elemento *Optimization*.

O caminho percorrido pelos *drones*, que executam essa função de mula de dados, é algo muito importante de ser avaliado, pois está relacionado diretamente com a autonomia e o gasto energético do *drone* durante o desempenho deste papel, considerando que estas aeronaves serão compartilhadas, a eficiência energética é um aspecto importante de ser otimizado. Ao visualizar-se um conjunto de dispositivos IoT espalhados por uma determinada área geográfica, sabe-se que em muitos casos não é possível servir todos os dispositivos devido a uma limitada carga de bateria encontrada em tais dispositivos, por isso a otimização se torna um importante componente da arquitetura, visando, entre outros objetivos, prevenir que seja dada ao veículo uma missão inviável.

Portanto, a partir da adesão de diversas aplicações, que se comportam como clientes da interface de serviços, e que têm a intenção de consumir dados coletados pelo VANT, o conjunto de componentes da arquitetura permite que haja interação desde às aplicações até os dispositivos IoT que não possuem conectividade. Considerando o algoritmo de roteamento proposto neste trabalho, a abordagem dele pode ser utilizada

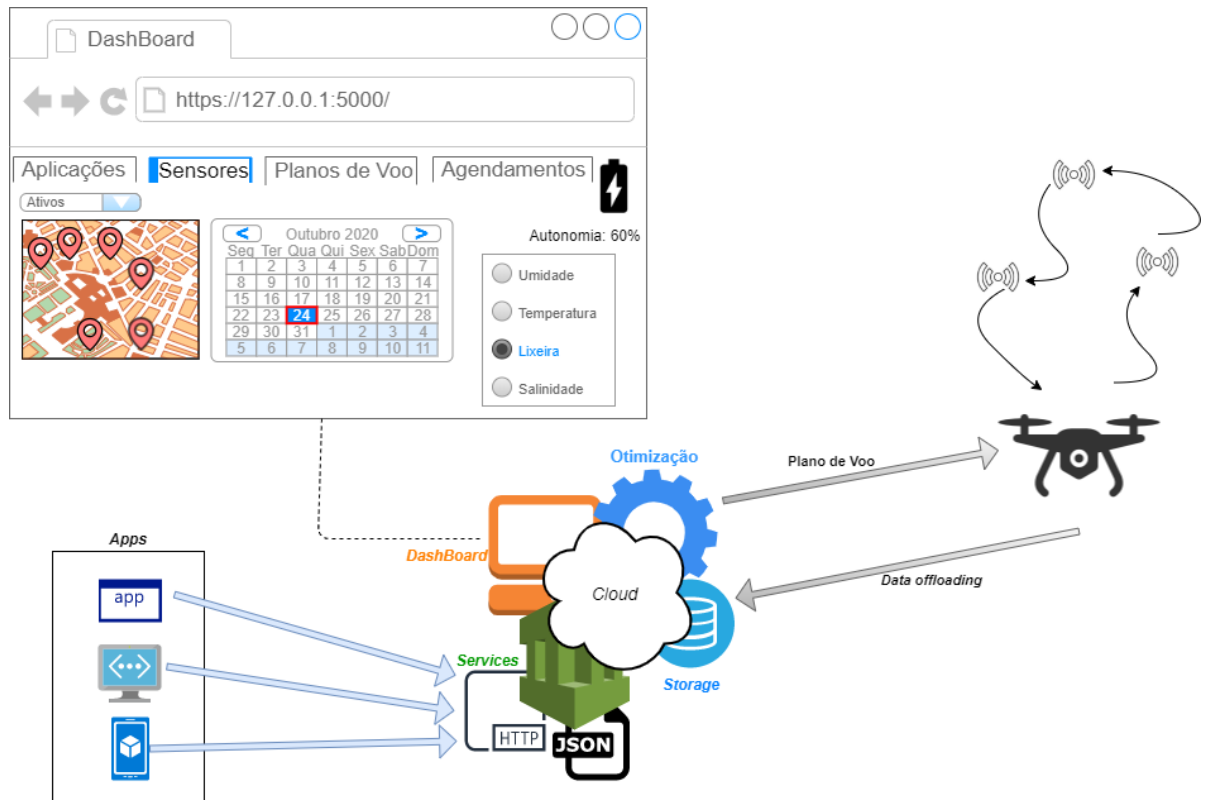


Figura 4.1: Visão geral da arquitetura *Drone as a Service*

para identificar qual a quantidade máxima de dispositivos pode ser visitada considerando as limitações das aeronaves disponíveis.

Após concluir o voo uma parte dos dados solicitados pode não ter sido transmitida, neste caso um novo voo com os dispositivos restantes é programado. A medida que dos dispositivos IoT transmitirem seus dados para o veículo aéreo esses dados são descarregados e disponibilizados no componente *Storage*, conforme mostra a Figura (4.1). Toda a gestão das aplicações, aeronaves, e dispositivos IoT que interagem na plataforma, acontece no componente *DashBoard*. Por ele é visualizado os indicadores de desempenho, o consumo energético, também é necessário um controle e aprovação de aplicações novas que solicitam subscrição à ferramenta.

Entre as principais funcionalidades encontradas nos componentes da arquitetura estão:

- Gerenciamento de dispositivos IoT: localização, tipo e status (ativado/desativado)
- Otimização da trajetória dos VANTs alocados na plataforma
- Geração de plano de voos baseado em protocolo de comandos compreendido pela maioria dos VANTs
- Interface para aplicações externas
- Data Offloading
- Gerenciamento da autonomia do drone

### 4.2.1 Data Offloading

Os serviços que as aeronaves não tripuladas podem oferecer apoiam muitas aplicações em parte de suas soluções, como por exemplo: implantação de provedores de comunicação, *streamers* de mídia, entrega de produtos e coleta de dados. Uma alternativa viável para implementação desses serviços é o compartilhamento dos recursos proporcionados pelos VANTs entre diferentes usuários através de uma mesma interface, tornando possível a virtualização das funções da aeronave por um meio de acesso único, por exemplo um serviço HTTP na web.

A estrutura *cloud* pode proporcionar uma forma de virtualizar o processamento, armazenamento e transmissão dos dados gerados por sensores IoT. Após serem cadastradas na plataforma de serviços e serem autorizadas a consumir os dados que a aeronave coleta, estas aplicações clientes fazem solicitações para ter a disponibilidade do dados e essas solicitações são empilhadas na plataforma para que se possam ser utilizadas como entrada para definir os pontos que devam ser visitados no mínimo para atender a necessidade das aplicações. A partir das subscrições feitas pelas aplicações consumidoras de dados, a plataforma agenda voos e toma decisão de quando realizar os voos para realizar a coleta de dados.

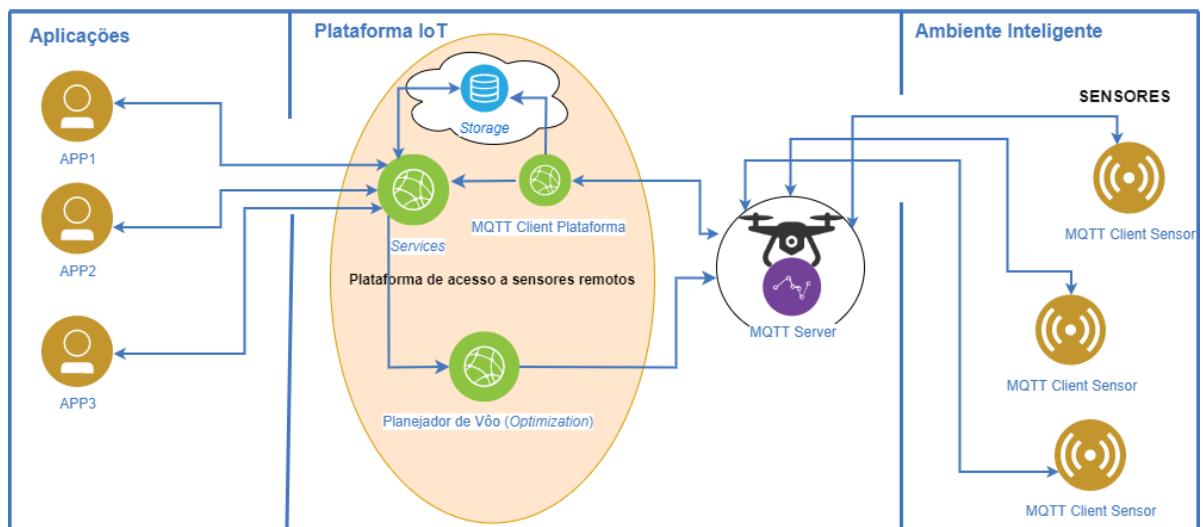


Figura 4.2: Esquema de funcionamento do *Data Offloading*.

O funcionamento do *DataOffloading* que acontece na plataforma IoT compreende os elementos:

- (1) *Service*, o que seria o *Middeware* no funcionamento da plataforma, pois é uma porta de entrada para aplicações que provocam a programação de voos que cumpre a demanda de dados requisitado por estes clientes e também é responsável por enviar uma mensagem de aviso da disponibilidade de dados ou entregar os arquivos decorrentes dos voos para as aplicações que estavam aguardando.

- (2) MQTT Client: a plataforma IoT possui um cliente que receberá os dados trazidos pela aeronave. O componente *Service* se comunica com Aplicações que queiram consumir dados de sensores IoT (servidos pelo *drone*, pois não possuem conectividade móvel). o *Service* envia para o Planejador de Voo qual grupo de sensores as aplicações precisam consumir dados (umidade, temperatura, profundidade de lixeiras etc).
- (3) MQTT Server que está implantado no sincronizador de dados (*drone*).
- (4) Planejador de Voo, componente de otimização da trajetória a ser percorrida pelo *drone*, utilizando o algoritmo de otimização implementado que garante que o máximo de sensores sejam visitados a cada voo, dado uma autonomia inicial. Ao ser constatado que a demanda existente equivale a um voo que irá consumir toda energia do *drone* ou algum percentual da bateria pré-definido. o Planejador de Voo envia um comando para iniciar voo baseado na trajetória definida pelo algoritmo de otimização.

### 4.2.2 Diagrama de Classes

Para projetar a solução que atenderá as especificações descritas no parágrafo anterior é desenhado um diagrama de classes (Figura 4.3). A classe *WebService* define a interface pela qual farão requisições as aplicações que irão consumir dados e portanto fazem assinaturas a determinados grupo de sensors, por exemplo a um grupo de sensores de umidade, ou um grupo de sensores de temperatura etc. Esta interface web também é utilizada para realizar o gerenciamento por parte do administrador do *drone* que está disponível, por exemplo para setar a autonomia que este drone possui.

Ao decorrer do tempo em que o sistema for recebendo mais assinaturas por parte das aplicações e o número de sensores a serem sobrevoados aumentar, a cada nova assinatura a classe *WebService* faz uma requisição a outra classe *Otimizacao* que é responsável por realizar o processamento da otimização referente ao mínimo de energia gasta para percorrer todos os sensores, o máximo de sensores possíveis de percorridos com a autonomia atual do *drone* e uma função que executa os dois modelos em sequência fazendo com que o drone visite o máximo de sensores gastando o mínimo de energia.

Dado cada grupo de sensores assinados pelas aplicações, o mínimo de energia para que drone percorrer todos os sensores é auferido, este critério é utilizado como exemplo dentro do fluxo de execução das tarefas na plataforma. Se este mínimo for pelo menos 80% da autonomia do drone uma *Thread* é disparada executando a classe *DroneMQTT* que é responsável por conectar ao *middleware* utilizado neste caso, chamado *mosquitto*, se tornado um *subscriber* no tópico de troca de mensagens especificado pela própria arquitetura. Este *middleware* está executando no computador à bordo do *drone*,

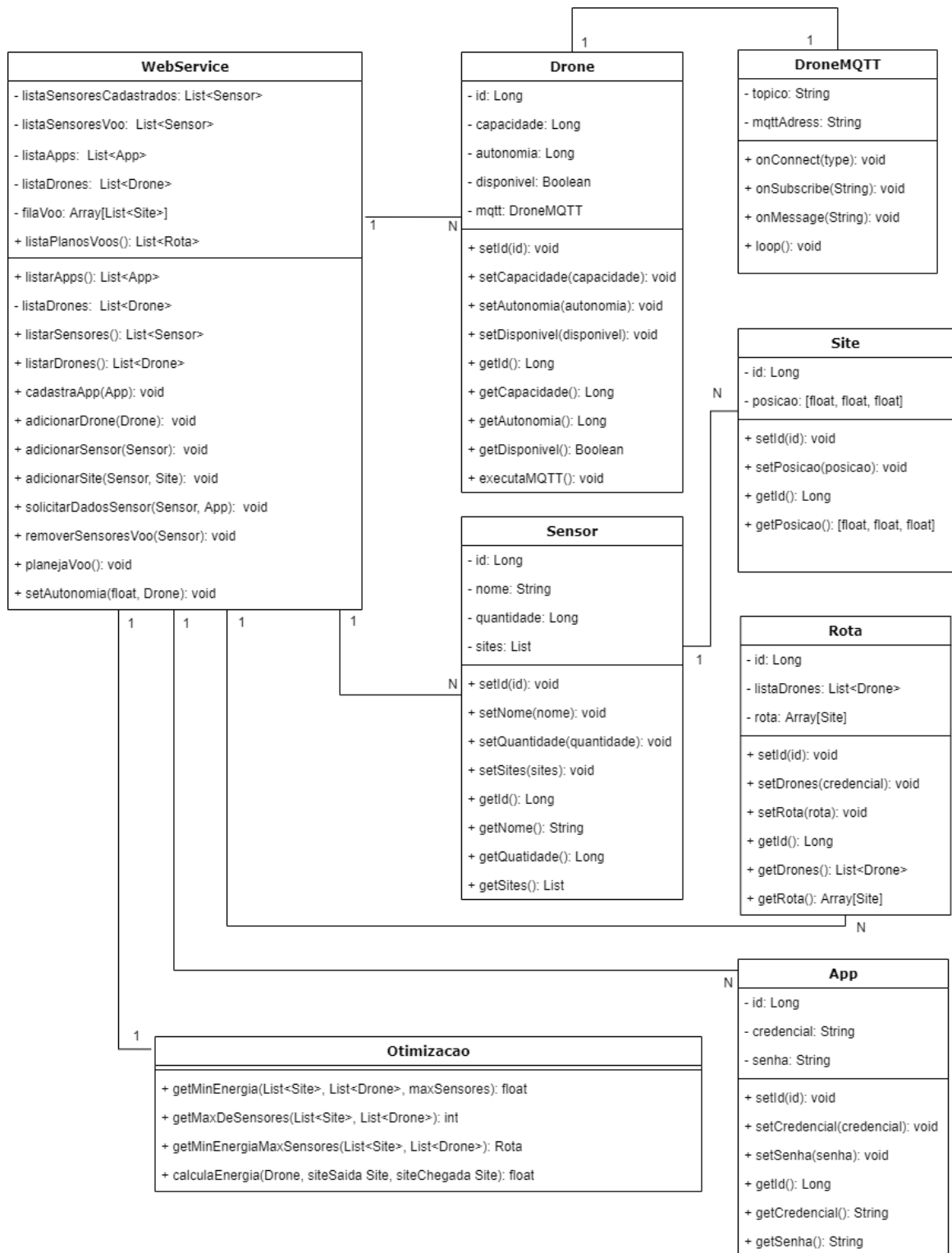


Figura 4.3: Diagrama de Classes.

para que assim que o *drone* faça sincronização com os sensores, estes escrevam no tópico já subscrito pela plataforma onde os sensores são *publishers*. Dessa forma é feita a transmissão de dados dos sensores até a plataforma onde inúmeras aplicações podem ter assinado para consumir estes dados transmitidos.

## 4.3 Implementação e Avaliação

### 4.3.1 Funções Básicas no Componente *Services*

É apresentado na Figura (4.3), a classe *WebService* que possui os serviços que o componente *Services* oferece, entre eles a lista de quais tipos de sensores existem na plataforma, a lista dos sensores específicos com informações de localização, se está alocado no próximo voo, também são mostradas funções básicas como adicionar ou remover sensores no cadastro, estes são exemplos de serviços que são utilizados pela própria plataforma com finalidade de gerenciamento dos sensores alocados, por exemplo.

### 4.3.2 Interação *Services* x *Optimization*

São implementados os principais serviços para oferecer uma interface entre usuários e o VANT, considerando que os recursos utilizados do veículo neste cenário são compartilhados. Para gerenciar este uso compartilhado é necessário basicamente se cadastrar para se identificar no uso desta plataforma, e identificar cada um dos tipos de sensores que o usuário (aplicação cliente) quer consumir. A medida que essa interação dos clientes do serviço for acontecendo são executados três algoritmos de otimização diferentes: (1) Identifica o mínimo de gasto energético necessário para visitar todos os sensores existentes (Problema do roteamento de veículos), (2) Verifica o máximo de sensores possíveis de serem visitados utilizando a autonomia existente no VANT naquele momento, (3) Sabendo qual é o máximo de sensores a serem visitados, é definido o mínimo de gasto energético possível para visitar este número de sensores.

No algoritmo (2) estão os principais serviços implementados para uso das aplicações clientes, *cadastraApp()* primeiro passo necessário antes de iniciar o consumo de dados, pois será requisitado nos demais serviços que a aplicação cliente esteja cadastrada na plataforma. O serviço implementado na função *solicitaDadosSensor()* implementa a ação que a aplicação cliente deverá requisitar para assinar um determinado grupo de sensores, quando a aplicação queira aquele tipo de dados por parte da plataforma. O método *planejaVoo()* faz integração com a parte de otimização da plataforma e depois do primeiro algoritmo de otimização executado verifica se o gasto total é maior do que 80% da

autonomia do VANT, caso seja são executados os algoritmos de otimização restantes.

---

**Algorithm 2:** Principais Serviços Oferecidos pela plataforma DaaS  
através da classe *WebService*

---

```

1 Method cadastraApp(credencialApp):
2   if (!isCadastrado(credencialApp)) then
3     └ this.listaAppsCadastrados.push(credencialApp)
4
5 Method verificaVoo(listaSensoresVoo):
6   listaSitesVoo = new List<Sites>
7   for x in listaSensoresVoos do
8     sites = x.getSites()
9     for y in sites do
10    └ listaSitesVoo.push(y)
11   minEnergia = MinimoEnergiaVoo(listaSitesVoo)
12   percentual = (minEnergia/autonomia) * 100
13   /* Este critério é utilizado como exemplo */
14   if (percentual > 80) then
15     └ this.filaVoos.push(listaSitesVoo)
16     return true
17   else
18     └ return false
19
20 Method planejaVoo(listaSites):
21   rota = Otimizacao.getMinEnergiaMaxSensores(listaSites,
22     this.listaDrones)
23   this.listaPlanosVoos.push(rota)
24
25 Method solicitaDadosSensor(credencialApp, tipoSensor):
26   if (isCadastrado(credencialApp)) then
27     └ this.listaSensoresVoo.push(tipoSensor)
28   if (VerificaVoo(this.listaSensoresVoo)) then
29     └ if (DroneIsDisponivel()) then
30       └ planejaVoo(this.filaVoos.pop())

```

---

Na Figura (4.4) temos um exemplo de utilização da plataforma, através de um

diagrama de sequência, demonstrando os passos possíveis no decorrer da interação com a plataforma. Na primeira ação são solicitados dados de status da aplicação quanto aos sensores que estão alocados e quais deles tiveram seus dados solicitados, e quais estão cadastrados. Em seguida é exemplificado o cadastro de uma aplicação. Quanto à terceira interação vista diagrama vê-se a possibilidade de se alterar a autonomia existente em um *drone* da frota naquele momento. Nos passos seguintes apresentam a interação em que são utilizados os serviços mostrados no algoritmo (2).

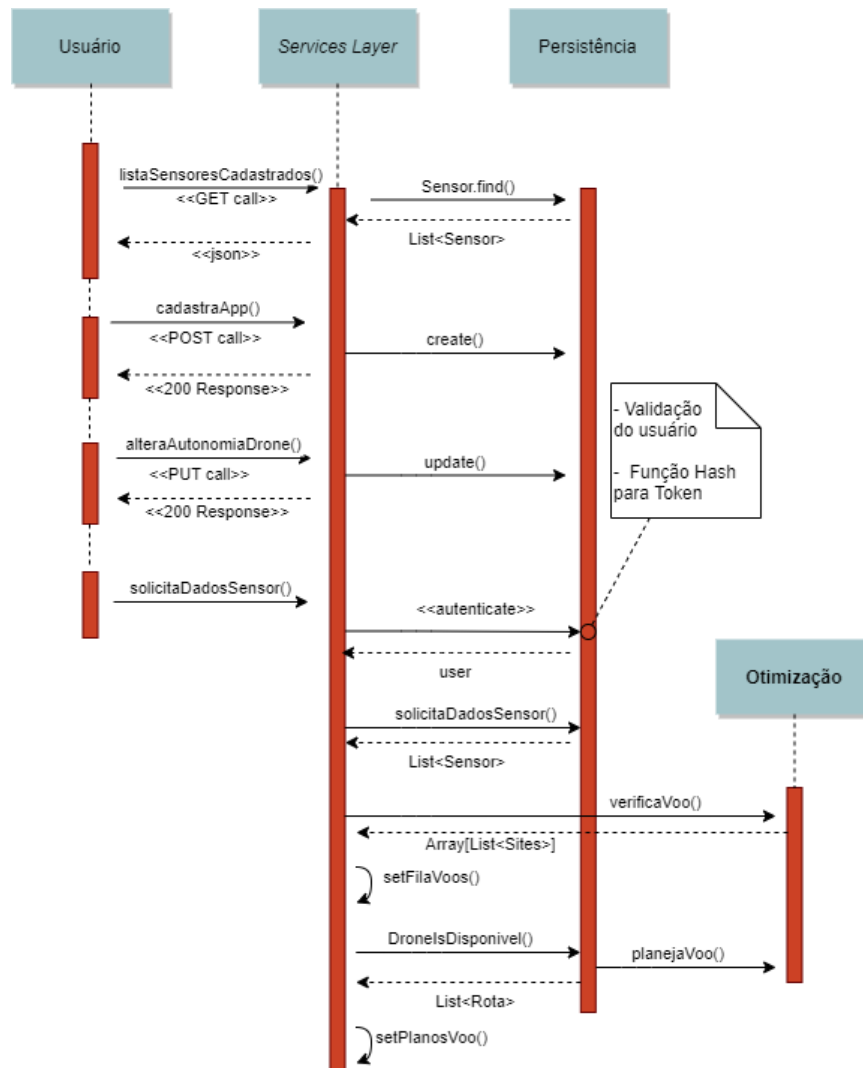


Figura 4.4: Exemplo de uso da interface de serviços web através de um diagrama de sequência.

## Ferramentas de Apoio a Voos Reais e Simulados com gerenciamento de dispositivos IoT

---

Este capítulo se concentra na possibilidade de testar um conjunto de ferramentas próprias para dar apoio ao *drone* durante a execução de missões para atender a um campo de cultivo, e que haja para tanto um planejamento de voo otimizado, dando maior autonomia e eficiência energética ao *drone*. É configurado um ambiente de simulação utilizando software adaptado ao bebop2 da Parrot. Também é desenvolvida uma *Web tool* para fazer o planejamento das missões, utilizando três tipos de algoritmos possíveis para a otimização da trajetória do voo.

O uso de *drones* para servir como coletor dos dados captados por sensores também é vantajoso. Entre outros pontos positivos destacam-se o baixo custo para instalar uma infraestrutura de comunicação, ao invés da compra de antenas e cabos de rede utilizados para o mesmo fim. A coleta dos dados é feita através de voo autônomo, ou seja, uma vez determinado o plano de voo, não é necessário que um piloto controle a execução da missão.

### 5.1 Motivações

As preocupações com a segurança alimentar aumentaram bastante nos últimos anos. A crescente população humana, juntamente com os recursos agrícolas cada vez menores, levam muitos governos e conglomerados internacionais ao redor do mundo a procurar novas maneiras de melhorar a eficiência da agricultura. Organizações de todo o mundo estão experimentando robôs móveis, manipuladores, humanoides e *drones* em várias tarefas agrícolas [25].

A chamada agricultura de precisão tem alavancado a produtividade e aprimorado a qualidade na produção agrícola mundial. Uma das abordagens é o uso das tecnologias da informação disponíveis na forma de máquinas inteligentes para reduzir os esforços humanos de maneira eficaz do que no passado [37].

Agricultura de precisão (AP) é a capacidade de lidar com variações de produtividade em um campo e maximizar o retorno financeiro, reduzir o desperdício e minimizar o impacto do meio ambiente, usando coleta automatizada de dados, documentação e utilização dessas informações para decisões estratégicas de gerenciamento de fazendas por meio de tecnologia de detecção e comunicação. O passo mais importante na AP é a geração de mapas do solo com suas características. Isso incluiu amostragem do solo, monitoramento de produtividade e observação de culturas.

O advento das arquiteturas de sistemas autônomos nos dá a oportunidade de desenvolver uma nova gama completa de equipamentos agrícolas baseados em pequenas máquinas inteligentes que podem fazer a coisa certa, no lugar certo, na hora certa e da maneira certa. O desenvolvimento de sistemas autônomos para a indústria agrícola está se tornando uma tarefa cada vez mais importante, especialmente devido à crescente demanda por produtos agrícolas de qualidade e à contínua diminuição da disponibilidade de mão-de-obra nas áreas rurais. Por fim, o planejamento e o controle de trajetória de veículos articulados e máquinas agrícolas continuam sendo uma questão central na vantagem da estrutura da dinâmica subjacente dos veículos para o planejamento e controle de trajetória [16].

O IoT em um contexto agrícola refere-se ao uso de sensores e outros dispositivos para transformar todos os elementos e ações envolvidos na agricultura em dados. Foi relatado que uma estimativa de 10% a 15% dos agricultores dos EUA estão usando soluções de IoT na fazenda em 1200 milhões de hectares e 250.000 explorações agrícolas [3]. A IoT impulsiona a Agricultura 4.0 [51], de fato, as tecnologias de IoT são uma das razões pelas quais a agricultura pode gerar uma quantidade tão grande de informações valiosas, e o setor agrícola é espera-se que seja altamente influenciado pelos avanços nessas tecnologias [67].

Estima-se que, com com novas técnicas, a IoT tem o potencial de aumentar a produtividade agrícola em 70% até 2050 [40], o que é positivo porque, segundo Myklevy et al., o mundo precisa aumentar a produção global de alimentos produção em 60% até 2050, devido a um crescimento populacional superior a nove bilhões [57]. As principais vantagens do uso da IoT estão em atingir maiores rendimentos e menor custo. Por exemplo, estudos da OnFarm constataram que, para uma fazenda média que usa a Internet das coisas, o rendimento aumenta 1,75% e os custos de energia caem para 32 dólares por hectare, enquanto o uso da água para irrigação cai 8% [51] [57].

As redes de sensores são cada vez mais consideradas pela comunidade científica como o futuro do monitoramento ambiental: oferecendo a baixo custo a possibilidade de reunir e processar todos os tipos de dados com uma quantidade de tempo inimaginável antes [17]. Em particular, o uso de *drones* permite a oportunidade de obter uma pesquisa geral da área e fazer um melhor uso do tempo do agricultor, evitando que seja necessária

intervenção humana em áreas problemáticas e perigosas que possam estar no campo, tornando também o cultivo mais seguro ao mitigar a possibilidade de acidentes.

## 5.2 Ambiente de Simulação e Ferramentas

Considerando os *drones* com *setups* parecidos, neste caso que não precisam o acoplamento de novos componentes para poder realizar voos, e pesando as vantagens e desvantagens de cada um, o Parrot O Bebop 2 é uma opção confiável para este projeto. Ele apresenta um preço equilibrado e bom desempenho conforme análise feita em [9]. Na tabela 5.1 é possível ver os principais *drones ready-to-fly* examinados, comparados lado a lado.

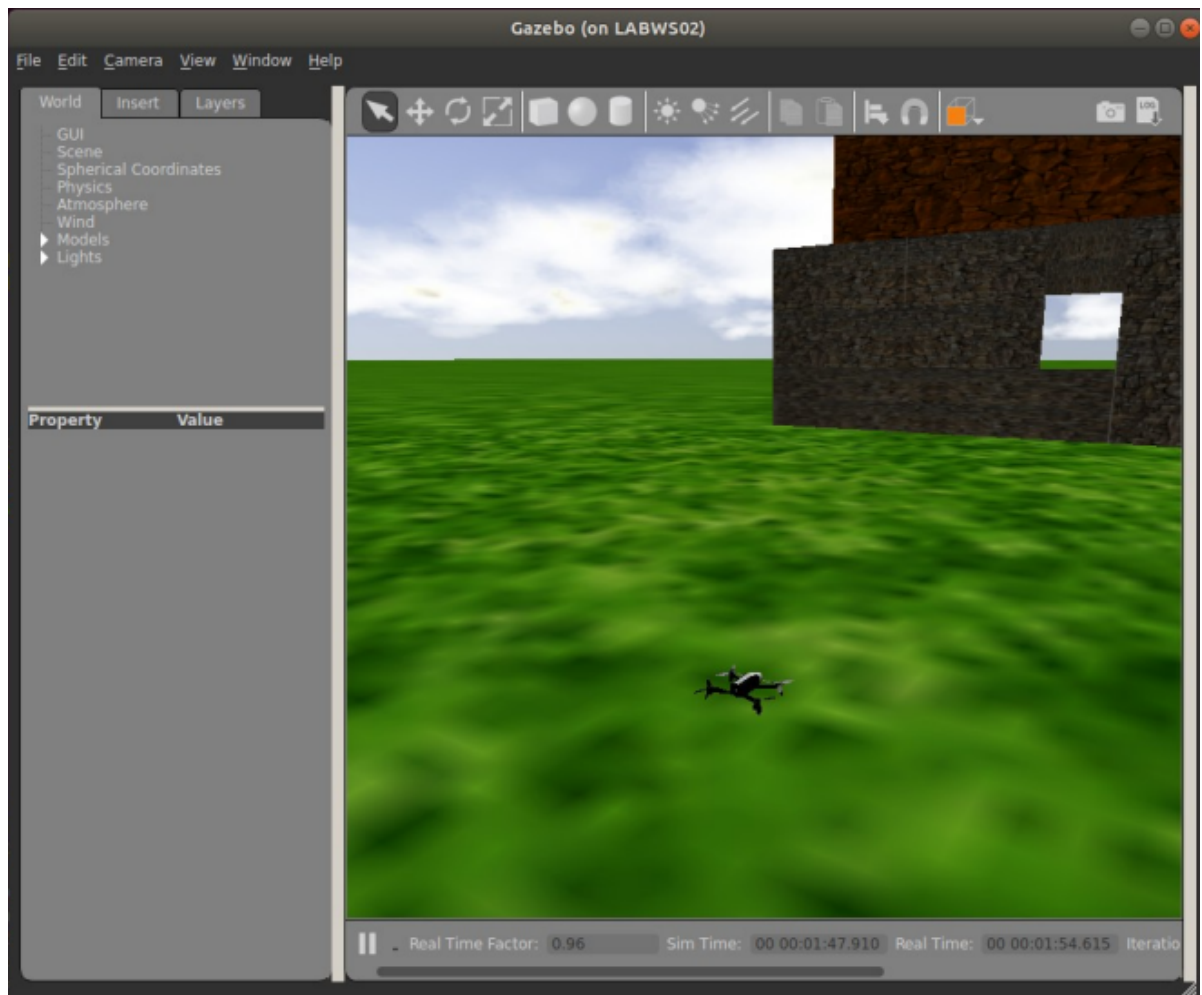


Figura 5.1: Simulação de voo executada na plataforma Gazebo.

O *drone* utilizado está disponível em laboratório nas dependências da Fraunhofer AICO-Portugal [9], o que possibilita a verificação, em momento posterior, do comportamento dos softwares em situação real de voo, aferindo o grau de precisão dos resultados mostrados ao decorrer das simulações. O Bebop2 é programável e o fabricante também

	Intel Aero RTF	AR Drone	Bebop 2	Mavic Pro
Entradas reconfiguráveis	✓	✓	✓	✗
Altera firmware	✓	✓	✓	✗
Tempo de Voo	Customizável	≈ 12min	≈ 25min	≈ 27min
GPS	✓	✗	✓	✓
Módulo de Odometria	✓	✓	✓	✓
Preço	985,50	219,00	652,25	1194,96

Tabela 5.1: Comparativo resumo entre *drones Ready-to-Fly*

oferece um kit de desenvolvimento de software. O Bebop2 não exige um *Mission Planner*, mas usa o mesmo computador como estação terrestre [34].

Experimentos reais usando VANTs são caros e arriscados devido a ferir seres humanos e danificar o hardware do VANT. Portanto, o desempenho dos VANTs deve ser analisado antes de sua implantação. Por esse motivo, muitos pesquisadores utilizam a simulação *Software in the loop* (SIL). Quando um teste no SIL for concluído com êxito, o próximo estágio poderia ser o teste no mundo real. A simulação SIL[48] pode ser executada no Gazebo [1] com o uso do ROS. A simulação é construída nas ferramentas Parrot-Sphinx. O Sphinx assume a interface *wifi* e a fornece ao firmware simulado. O *firmware* executa uma instância do *host daemon* de ponto de acesso (*hostapd*), que cria um ponto de acesso *wifi* [49]. Parrot-Sphinx é uma ferramenta de simulação inicialmente pensada para cobrir as necessidades dos engenheiros da Parrot que desenvolvem software para drones. O conceito principal é executar o *firmware* do *drone* Parrot em um PC, em um ambiente isolado e bem separado do sistema *host*.

O Gazebo é amplamente utilizado para simular o ambiente físico e visual do *drone* [50]. O Gazebo executa dois processos: servidor e cliente. O primeiro é o núcleo do Gazebo e executa o *loop* físico, podendo ser usado independentemente de qualquer interface gráfica. O último é a interconexão com o usuário e visualização da simulação. No Gazebo, é possível fazer o *upload* de todos os tipos de modelos, já disponíveis ou criados pelo usuário, o que permite ao desenvolvedor criar um cenário dinâmico simulado realista para uma enorme variedade de simulações. Comparado a outros softwares, o Gazebo oferece vantagens em termos de complexidade de software, plataforma de código aberto e integração com o ROS. Além disso, o Gazebo é o software mais compatível com o ROS [13]. Há muitas vantagens em usar o Gazebo com o *middleware* ROS. Eles fornecem as interfaces necessárias para simular um robô geral no Gazebo usando mensagens, serviços ROS e reconfiguração dinâmica [4].

O ROS é um sistema/*middleware* meta-operacional de código aberto para robôs. Suporta os mesmos serviços de um sistema operacional, incluindo abstração de hardware, controle de dispositivo de baixo nível, implementação de funcionalidade, passagem de



Figura 5.2: Parrot Bebop 2.

mensagens entre processos e gerenciamento de pacotes. Ele também fornece ferramentas e bibliotecas para reunindo, construindo, escrevendo e executando código em vários computadores. A principal vantagem de usar o ROS é que a comunicação entre diferentes processos e máquinas são facilmente alcançados [53].

Neste ambiente de simulação é usado o *driver* Bebop Autonomy para ROS [5]. O Bebop autonomy é um *driver* ROS para os drones Parrot Bebop 1.0 e 2.0 (quadcopters), com base no ARDroneSDK3 oficial da Parrot, que funciona como um núcleo ROS. Este *driver* foi desenvolvido para aplicações de VANTs avançadas, possui um modo de voo externo onde o veículo obedece a um ponto de ajuste de posição, velocidade ou atitude fornecido pelo protocolo de comunicação MAVLink (*Micro Air Vehicle Link*) [6].

## 5.3 Web Tool

É proposta uma ferramenta com interface web que seja utilizada para planejamento da trajetória de voos de *drones* a partir da especificação dos tipos de sensores, autonomia do *drone* e localização dos sensores a serem visitados. O plano de voo feito é utilizado para coleta dos dados capturados pelos sensores espalhados em um campo agrícola.

A ferramenta desenvolvida tem como principal objetivo visualizar os sensores alocados no campo de cultivo, dessa forma auxiliar a geração de um plano de voo que atenda aos requisitos de interesse do administrador. Portanto é permitido adicionar sensores e sondas e suas localizações, e indicar se os sensores estão ativos ou inativos com

o passar o tempo. A partir da geração de plano de voo é permitido salvá-los e nomeá-los para identificar o objetivo de criação daquele plano de voo em específico. As principais funcionalidades da ferramenta web para planejamento de missão são:

- Gerenciamento de sensores e sondas: localização, tipo e status (ativado/desativado).
- Otimização do plano de voo utilizando 3 diferentes algoritmos (MILP, Busca Tabu e Algoritmo Genético).
- Geração de plano de voos baseado em protocolo de comandos MAVLink.

### 5.3.1 Estação de controle de solo (ECS)

Os VANTs geralmente são controlados por vários operadores de estações de controle de solo (pilotos internos) dentro de uma ou mais estações de controle de solo, dependendo do tamanho da missão. Um software de Estação de Controle de Solo possui vários recursos [26]:

- Planejamento da missão: o ECS planeja os caminhos da missão para o VANTs, portanto, define o caminho a ser percorrido através do estabelecimento de pontos onde VANTs deve atingir para completar a missão, considerando, de tal modo, a trajetória planejada.
- Navegação: durante a missão, os VANTs são colocados em várias posições em diferentes altitudes (configurações de *waypoints*), a fim de verificar a área monitorada. Para isso, o GCS deve exibir e controlar os movimentos dos VANTs para ter sucesso na missão.
- Carga útil: os VANTs são montados por dispositivos como câmeras, sensores etc. O ECS deve lidar com os parâmetros de carga útil durante a missão, fornecendo *feedback* como saídas.
- Comunicação e troca de dados: ECS e VANTs devem ter comunicação direta e bidirecional entre eles. O ECS envia comandos e ordens para o VANT em relação à missão e o VANT transmite telemetria e dados (imagens, vídeos, etc.) para a estação de controle em terra.

#### ***Mission planner para VANT como mula de dados***

O planejamento de missão é um recurso existente no ECS, onde são preparados os planos da missão e caminhos para o VANT, de acordo com o ambiente e os requisitos da missão, então, o VANT tem que concluir a missão dependendo da trajetória planejada [26].

Geralmente o planejamento de missão tem muitos papéis, tais como exibir status de um VANT, enviar comandos de movimento para o veículos através, por exemplo,

do protocolo MAVLink, e definir um plano de voo para ser executado. O planejador de missão pode também gravar e reproduzir os dados de registro de um voo de *drone*.

As mensagens de erro associadas ao status e ao plano de navegação, incluindo ruído ambiental, como interferência magnética, podem ser registradas e então analisadas por um usuário. Um planejador de missão é usado para atribuir um plano de navegação. Esses parâmetros podem ser definidos pelo planejador de missão usando o protocolo MAVLink [34].

### 5.3.2 Protocolo MAVLink

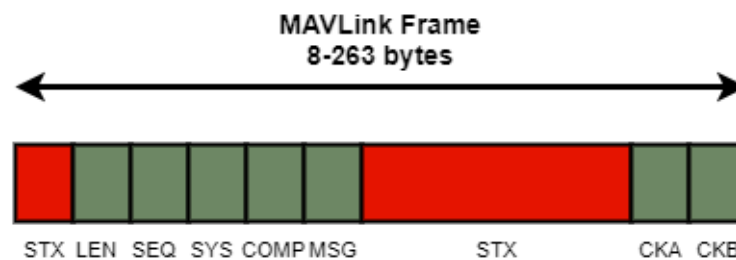


Figura 5.3: Pacote MAVLink.

O protocolo de comunicação MAVLink permite que as entidades se comuniquem através de um canal sem fio. Quando usado em *drones*, é usado para as comunicações bidirecionais entre o *drone* e o ECS. O ECS envia comandos e controles para o *drone* enquanto o *drone* envia telemetria e informações de status [68]. Este protocolo foi extensivamente testado em várias plataformas VANT, e numerosos aplicativos de software ECS executados no Microsoft Windows, Sistemas operacionais Mac e Linux. Existem 18 tipos de mensagens no Protocolo MAVLink e 91 mensagens definidas na Documentação do MAVLink.

Os tipos de mensagem estabelecem parâmetros de codificação, como tipos de arquitetura, configuração e tipo de mensagem de comando (por exemplo, hardware do *autopilot*, tipo de veículo, comando para o servidor etc). O tipo de mensagem é o primeiro campo da mensagem MAVLink. O MAVLink As mensagens carregam os comandos transmitidos do ECS para VANT e fornecem *feedback* (por exemplo, telemetria e status do sistema) do VANT ao ECS, permitindo ao piloto manter o controle da aeronave. Essas mensagens são enviadas como pacotes de dados entre a estação terrestre e o piloto automático para que pilote corretamente o VANT [8]. A Figura (5.3) mostra a estrutura do pacote MAVLink.

O MAVLink é apenas o cabeçalho, biblioteca de distribuição de mensagens usada como protocolo de comunicação entre a estação terrestre e a aeronave [41]. O principal componentes de uma mensagem MAVLink são o cabeçalho, ID do sistema, ID da mensagem e *payload*. O cabeçalho é usado para classificar a mensagem como um pacote

MAV_CMD_NAV_WAYPOINT (16)	MAV_CMD_NAV_LAND (21)	MAV_CMD_NAV_TAKEOFF (22)	MAV_CMD_DO_CHANGE_SPEED (178)
<b>Parâmetro:(Rótulo)</b>	<b>Parâmetro:(Rótulo)</b>	<b>Parâmetro:(Rótulo)</b>	<b>Parâmetro:(Rótulo)</b>
1: Hold	1: Abort Alt	1: Pitch	1: Speed Type
2: Accept Radius	2: Land Mode	2	2: Speed
3: Pass Radius	3	3	3: Throttle
4: Yaw	4: Yaw Angle	4: Yaw	4: Relative
5: Latitude	5: Latitude	5: Latitude	5
6: Longitude	6: Longitude	6: Longitude	6
7: Altitude	7: Altitude	7: Altitude	7

Tabela 5.2: Exemplos de comandos MAVLink.

MAVLink. O ID do sistema identifica o sistema que está enviando a mensagem enquanto o ID da mensagem identifica o tipo de mensagem que está sendo enviada. Por exemplo, a mensagem mais comum a ser enviada é a pulsação ID = 0, que é constantemente enviado para garantir que o avião e a estação terrestre estão conectados e se comunicando adequadamente. O *payload* da mensagem é o conteúdo dentro dela. O *payload* pode conter campos como o tipo de veículo, modo de voo, dados de posicionamento ou comandos para executar. A Tabela (5.2) mostra quatro exemplos de comandos que são utilizados na ferramenta desenvolvida, com que a aeronave vá para até um ponto determinado ponto considerando latitude e longitude que foi passada, envia mensagem para pousar, é um comando para decolagem, muda a velocidade do veículo. O primeiro comando *MAV\_CMD\_NAV\_WAYPOINT* significa navegar para um *waypoint*, seguindo a latitude e longitude passadas como parâmetros. O comando *MAV\_CMD\_NAV\_LAND* envia a informação para que a aeronave pouse. O comando *MAV\_CMD\_NAV\_TAKEOFF* faz com que o VANT decole. A última mensagem mostrada é a *MAV\_CMD\_DO\_CHANGE\_SPEED* e é utilizada para mudar a velocidade de voo da aeronave.

### 5.3.3 Uso de Algoritmos de Meta-Heurística

Para gerar o arquivo de plano de voo baseado no protocolo MAVLink, são oferecidos três opções de algoritmos possíveis de serem utilizados o MILP, a busca tabu e o algoritmo genético. O algoritmo MILP usado é mostrado no terceiro capítulo na seção do problema do caixeiro viajante, e é a mesma descrição do problema utilizada para implementar as demais heurísticas. O plano de voo é formatado num arquivo de extensão *.mavlink* para ser utilizado conforme mostrado no ambiente de simulação controlado pelo *driver* ROS.

Baseados em Vizinhança	Freq.	Baseados em População	Freq.
Busca Tabu	17	Algoritmos Genéticos	16
Busca Local Iterada	7	Colônia de Formigas	4
Busca em Vizinhança Variável	5	Busca Dispersa	2
Busca Adaptativa em Grandes Vizinhanças	4	<i>Path Relinking</i>	2
Recozimento Simulado	3	Enxame de Partículas	1
		Outros	4

Tabela 5.3: Tabela com algoritmos mais utilizados para resolução do problema de roteamento de veículos [70].

O termo "meta-heurística" designa uma ampla classe de métodos heurísticos que continuam a pesquisar além do primeiro local encontrado ideal. Um pouco definição bruta, mas reveladora, caracteriza as meta-heurísticas como heurísticas que orientam outras heurísticas.

Em [70] são agrupados 14 diferentes versões do problema de roteamento de veículos no mesmo grupo, o problema de roteamento possui características derivadas do problema clássico do caixeiro viajante. Todos juntos, em suas abordagens, constituem um conjunto de 64 algoritmos bem-sucedidos para as 14 descrições do problema relacionado. A análise é apoiada por observações quantitativas sobre o frequência de aparecimento de elementos da metodologia nas abordagens bem-sucedidas [70].

A Tabela 5.3 agrupa os métodos de meta-heurística mais utilizados para o problema. Esses métodos são visivelmente de várias naturezas, métodos de vizinhança e de base populacional [70]. Os dois algoritmos mais implementados para a resolução do problema de roteamento de veículos são o Busca Tabu e o Algoritmo Genético, portanto foram escolhidos estes dois algoritmos para serem utilizados na web *tool*, como recurso de comparação e uso de um cada deles no caso mais apropriado quando necessário gerar um plano de voo.

### **Busca Tabu**

A pesquisa por tabu associa uma pesquisa centrada na escolha do melhor vizinho da solução estabelecida, com recursos de aprendizado, geralmente representados como memórias de curto, médio e longo prazo nos elementos da solução, que substituem ou complementam significativamente a randomização usada em outras meta-heurísticas. O método está, assim, escapando das ótimas locais ao aceitar movimentos que degradam a solução, desde que sejam os melhores da vizinhança explorada. Esse processo de decisão é aprimorado por dois mecanismos, o primeiro evitando pedalar e confiar em memórias de curto prazo para rejeitar soluções que contêm elementos tabu examinados recentemente, a segunda que aceita soluções que atendem a alguns critérios como “a melhor solução em valor ou que contenha um determinado elemento da solução”. O aspecto de central importância são as memórias de médio e longo prazo usadas para gerenciar procedimentos como intensificação, por exemplo, concentrado à pesquisa de soluções de elite, promovendo elementos de alta qualidade, e diversificação, quando pesquisa em uma área pouco explorada do espaço de busca, promovendo elementos pouco frequentes, e assim por diante [70]. Na figura 5.4 encontra-se uma representação do funcionamento do algoritmo.

Para construção de um algoritmo baseado em Busca Tabu siga a formulação a seguir. Primeiro selecione uma solução inicial  $s_0 \in S$ , em seguida inicialize a Lista de Tabu  $L_0 = \emptyset$  e selecione um tamanho para a lista tabu. Estabeleça  $k = 0$ . Determine

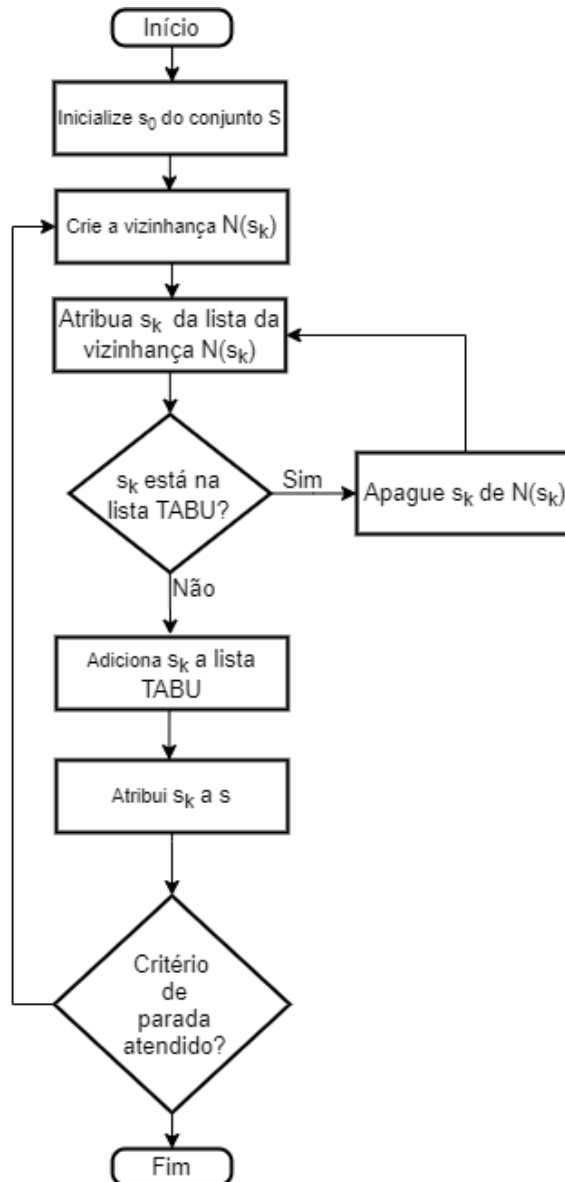


Figura 5.4: Meta-heurística Busca Tabu.

a viabilidade da vizinhança  $N(s_k)$  que exclui membros inferiores da lista de tabus  $L_k$ . Seleciona o próximo movimento  $s_{k+1}$  de  $N(s_k)$  ou  $L_k$  se houver uma solução melhor e atualiza  $L_{k+1}$ . Interrompa a execução se a condição de término for atingida, caso contrário,  $k = k + 1$  e retorne para 1.

### Algoritmo Genético

Os métodos baseados em população são frequentemente inspirados em mecanismos naturais. Os Algoritmos Genéticos (AG) foram introduzidos no final da década de 1950. Esses algoritmos interpretam leis genéticas e seleção natural para evoluir uma população de indivíduos assimilados a soluções, por meio de seleção elitista, cruzamento e operadores de mutação [70].

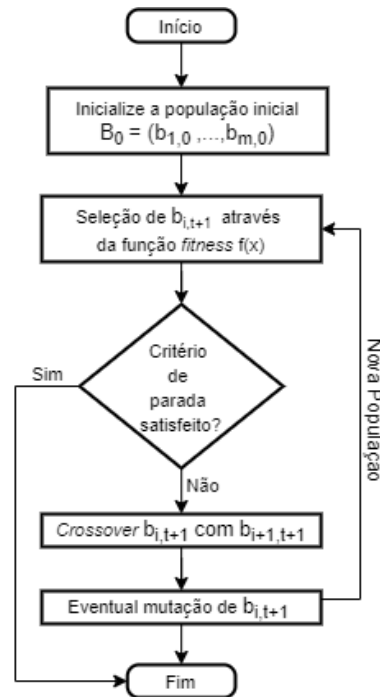


Figura 5.5: Meta-heurística Algoritmo Genético.

Os AGs não operam nos valores em um espaço de pesquisa  $X$ , por exemplo, mas em algumas versões codificadas deles (*strings* utilizando simplificação). Assuma o  $S$  um conjunto de *strings*, por exemplo,  $S = \{0, 1\}$ , exemplificando com caracteres binários. O tamanho da população é representado por  $m$ , portanto a geração de descendentes da população inicial em um dado tempo  $t$  é denotado por  $B_t = \{b_{1,t}, b_{2,t}, \dots, b_{m,t}\}$ . O algoritmo genético é mostrado de forma genérica na Figura (5.5).

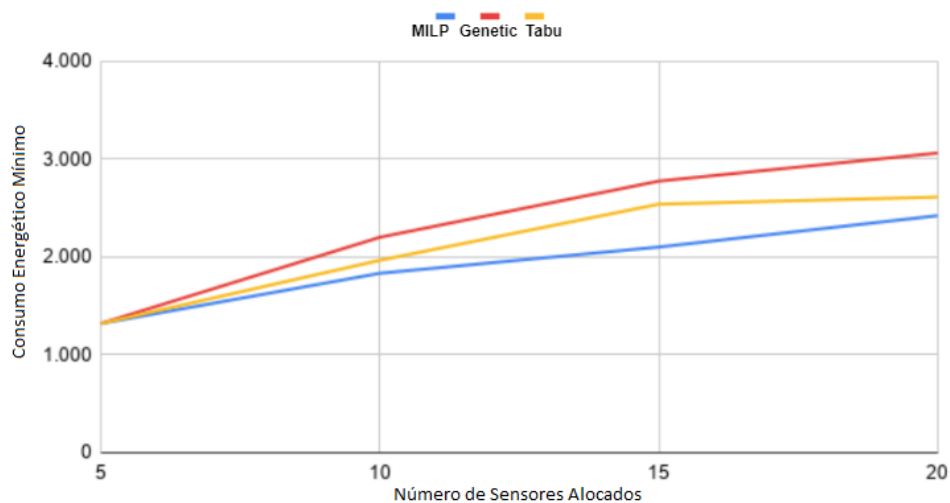


Figura 5.6: Comparação entre os três algoritmos disponibilizados na ferramenta web, executando a otimização para o mesmo conjunto de sensores.

O que é notado na comparação entre os três algoritmos conforme apresenta na Figura 5.6 é que dentre todos, conforme era esperado, o MILP é que apresenta a solução ótima, no entanto durante a simulação o Algoritmo Genético, obteve melhor tempo de processamento, porém assim como mostra a figura teve o pior resultado em termos de eficiência energética da aeronave. A Busca Tabu apesar de não ter conseguido chegar no valor ótimo, se mostrou extremamente melhor no tempo de processamento da otimização em relação ao MILP, o que poderia fazer com que seja escolhido na maioria dos casos, já que oferece melhor custo benefício na solução.

### 5.3.4 Interface de Gerenciamento

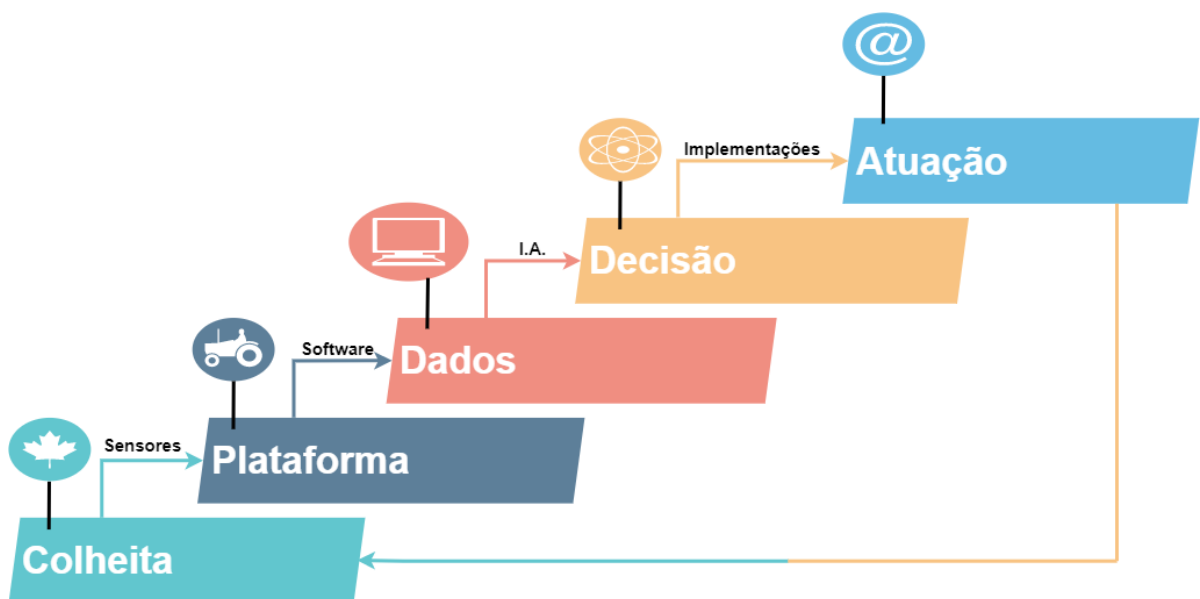


Figura 5.7: Ciclo de gerenciamento baseado em informações para agricultura avançada [57].

A Figura 5.7 explica o ciclo que incorpora um sistema geral de gerenciamento orientado a dados para agricultura avançada. Este trabalho se concentra nos estágios I (cultivo) e II (Plataforma), por que é onde acontece a coleta de dados e onde é necessário um gerenciamento da infraestrutura de comunicação para que os dados cheguem até às aplicações. Para tal é preciso uma ferramenta que centralize as informações dos sensores que estão espalhados em uma área agrícola, bem como neste caso da forma que os dados serão coletados (*drone* como mula de dados), o plano de voo.

A ferramenta possui duas abas: Simulação e Manual. A aba simulação é utilizada para gerar dados randômicos para que seja possível compreender o comportamento dos algoritmos e identificar previamente qual se adapta melhor para o número de sensores e nível de distribuição no campo. Na figura 5.8 é mostrada a ferramenta em aba de simulação.

Random Simulation Manual

### Locations

Mapa Satélite

Google

### Sensors List

New Random Points

#	Sensor Type	Amount	Add to Flight Plan?	Options
3	Temperatura	25	false	+ -
4	NPK	10	false	+ -
2	Umidade	15	false	+ -
1	Ph	5	true	+ -

Nothing yet...

Change Takeoff Point

### Generate Flight Plan

MILP Generate Flight Plan

0 -> 4 -> 2 -> 1 -> 3 -> 5 -> 0 390.81271425076153

Figura 5.8: Exemplo do uso da aba de simulação para testar o algoritmo.

Random Simulation Manual

### Locations

Mapa Satélite

Google

### Sites List

New Point

#	Location	Added to the Flight Plan?	Sensor Type	Active?	Options
0	41.287083,-8.639556,0	true	0	true	+ -
1	41.287050,-8.639622,0	true	Temperatura	true	+ -
2	41.287108,-8.639496,0	true	Ph	true	+ -
3	41.287087,-8.639453,0	true	Umidade	true	+ -
4	41.287139,-8.639533,0	true	Umidade	false	+ -

### New Sensor

Temperatura

Marker dropped: Current Lat: 41.287059 Current Lng: -8.639646

-8.639646 41.287059 Add Point

### Sensors Group List

#	Sensor Type	Amount	Add to Flight Plan?	Options
1	Temperatura	1	true	+ -
2	Ph	1	true	+ -
3	Umidade	2	true	+ -

### Generate Flight Plan

MILP Generate Flight Plan

0 -> 1 -> 3 -> 2 -> 0 429.57946726156814

Figura 5.9: Exemplo de uso de manual para inclusão de novos pontos de localização de sensores, e geração de plano de voo.

Na figura 5.9 vemos um caso em que são adicionados sensores manualmente onde é possível também especificar se o sensor está ativo ou não, caso o sensor não esteja ativo não será incluso no plano de voo. Para inclusão no plano de voo pode-se optar por adicionar através do tipo de sensor, ou adicionar individualmente o sensor em si, bem como retirá-lo. Na figura 5.10 observamos uma tela para cadastro de novos tipos de sensor que será usado na área.

## Sensors Type

#	Sensor Type	Options
1	Ph	+ -
2	Umidade	+ -
3	Temperatura	+ -
4	NPK	+ -
5	XPTO	+ -

test

New Sensor Type

Figura 5.10: Menu de inclusão de novos tipos de sensores.

---

## Conclusão

---

O trabalho visou aprofundar principalmente sua pesquisa no roteamento do voo de veículos aéreos não tripulados. A eficiência energética e a prevenção de acidentes com riscos de dados materiais e humanos foram um norte que balizou o desenvolvimento dos modelos de otimização, trazendo a formalização do problema de roteamento com características diferentes do convencional. A preocupação com a capacidade de armazenamento é também um importante diferencial no modelo pois os *drones* permanecem sendo um recurso escasso e limitado nas organizações, e os voos autônomos e programados devem se preocupar com a capacidade do espaço de alocação de dados pois este recurso será utilizado por diversos usuários e ao sobrevoar *clusters* os VANTs podem se deparar com uma massa de dados pesada. Outro aspecto fundamental foi o cálculo do gasto energético durante a transmissão dos dados a cada parada.

Ao longo do período de pesquisa e engajamento no Programa de Mestrado vários passando pelos estágios de frequência nas disciplinas, início e desenvolvimento da pesquisa, foram gerados diversos resultados consideráveis para serem mencionados. Foram publicados dois artigos, o primeiro no evento regional ERI-GO[54], onde ocorreu também a apresentação deste trabalho na ocasião, e o segundo artigo na revista de Sistemas de Informação da Faculdade Salesiana Maria Auxiliadora [55] quando contamos com a participação do doutorando Ciro Macedo que integrou o trabalho com a implementação comparativa da heurística da Colônia de Formigas .

Ao final de 2019, entre os meses de novembro de 2019 e janeiro de 2020 ocorreu a experiência de intercâmbio internacional em Portugal, na instituição de pesquisa Fraunhofer Portugal AICOS, e através desta participação foi possível um bom contato com a tecnologia dos VANTs de forma prática realizando voos controlados e autônomos, através de plataformas de simulação bem como *drone* físico. Através deste período de intercâmbio foi desenvolvido uma plataforma para gerenciamento de sensores e geração do plano de voo que poderia ser executado tanto de forma simulada ou real, através de *firmware* virtual ou software de controle de veículos autônomos. Nesta experiência foram realizados testes de voos reais e simulados executados durante período de intercâmbio, com finalidade de compreensão da dinâmica prática de manuseio dos veículos aéreos e

controle de voos autônomos através de um plano prévio enviado ao veículo. Por todo o período de pesquisa o grande esforço foi em modelar o máximo de aspectos inerentes ao voo dos VANTs para poder prever custo energético e abrangência do uso desta ferramenta enquanto nó de comunicação.

Neste trabalho foram abordados três aspectos de um sistema envolvendo a coleta de dados através de uma aeronave não tripulada: o roteamento do voo, uma arquitetura para compartilhamento do VANT entre aplicações externas, e a configuração de um ambiente de apoio para simulação e geração de planos de voos de *drone*.

## 6.1 Trabalhos Futuros

Para o prosseguimento desta pesquisa considera-se que se deva calcular o *bit rate* para a transmissão de dados a cada para a partir de um modelo dinâmico. Existem alguns elementos que podem trazer a degradação do sinal e que podem impactar nesta taxa de transmissão tais como o *path loss*, a interferência das conexões simultâneas entre si etc. O veículo aéreo pode se deparar com um grupo de sensores com tecnologias de comunicação heterogênea bem como uma quantidade de nós que ele não tenha capacidade para efetuar todas as conexões. Esses fatores também deverão ser levados em consideração para a geração de *clusters*. Outros aspectos do voo com característica dinâmica podem vir a serem modelados como o atrito e a influência do vento como variante e o surgimento de obstáculos de forma aleatória. Para além do sistema já apresentado no trabalho também poderemos incluir aspectos referentes a novas tecnologias emergentes na indústria e na academia, por exemplo a arquitetura e conectividade da rede 5G. Os VANTs coletam informações em tempo real tais como dados topográficos de superfície tridimensionais de alta precisão, imagens em tempo real, vídeo em tempo real etc. Essas informações são transmitidas ao sistema de IA pela rede 5G por que isso exige alta taxa de dados. A trajetória de voo da aeronave da rota fixa (trajetória ótima gerada pelo modelo construído em Programação Linear) é processada pelo sistema de IA e o comando de julgamento de decisões críticas ao voo é calculado e efetuado. Essas instruções são enviadas ao VANT através da rede 5G, o que requer baixo atraso e alta confiabilidade.

---

## Referências Bibliográficas

---

- [1] **Gazebo**. <http://gazebosim.org/>. Data acesso: março 2020.
- [2] ADBELHAFIZ, M.; MOSTAFA, A.; GIRARD, A. **Vehicle routing problem instances: Application to multi-uav mission planning**. In: *AIAA Guidance, Navigation, and Control Conference*, p. 8435, 2010.
- [3] ALPHA BROWN. **What is iot in agriculture? farmers aren't quite sure despite \$4bn us opportunity**. <https://agfundernews.com/iot-agriculture-farmers-arent-quite-sure-despite-4bn-us-opportunity.html>. Data acesso: março 2020.
- [4] APACHE. **Gazebo - ros overview**. [http://gazebosim.org/tutorials?tut=ros\\_overview](http://gazebosim.org/tutorials?tut=ros_overview). Data acesso: março 2020.
- [5] AUTONOMYLAB, SIMON FRASER UNIVERSITY. **bebop\_autonomy - ros driver for parrot bebop drone (quadrocopter) 1.0 2.0**. <https://bebop-autonomy.readthedocs.io/en/latest/>, 2015. Data acesso: março 2020.
- [6] AUTONOMYLAB, SIMON FRASER UNIVERSITY. **Sending commands to bebop**. <https://bebop-autonomy.readthedocs.io/en/latest/piloting.html>, 2015. Data acesso: março 2020.
- [7] BROWN, T.; ARGROW, B.; DIXON, C.; DOSHI, S.; THEKKEKUNNEL, R.-G.; HENKEL, D. **Ad hoc uav ground network (augnet)**. In: *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, p. 6321, 2004.
- [8] BUTCHER, N.; STEWART, A.; BIAZ, S. **Securing the mavlink communication protocol for unmanned aircraft systems**. *Appalachian State University, Auburn University, USA*, 2013.
- [9] CARNEIRO, V. T. **Opportunistic data collection using drones**, 2019.
- [10] CHOI, S.-C.; SUNG, N.-M.; PARK, J.-H.; AHN, I.-Y.; KIM, J. **Enabling drone as a service: Onem2m-based uav/drone management system**. In: *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, p. 18–20. IEEE, 2017.

- [11] COUTINHO, W. P.; BATTARRA, M.; FLIEGE, J. **The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review.** *Computers & Industrial Engineering*, 120:116–128, 2018.
- [12] CPLEX, I. I. **V12. 1: User's manual for cplex.** *International Business Machines Corporation*, 46(53):157, 2009.
- [13] CRAIGHEAD, J.; MURPHY, R.; BURKE, J.; GOLDIEZ, B. **A survey of commercial & open source unmanned vehicle simulators.** In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*, p. 852–857. IEEE, 2007.
- [14] DANTZIG, G. B.; FULKERSON, D. R.; JOHNSON, S. M. **On a linear-programming, combinatorial approach to the traveling-salesman problem.** *Operations Research*, 7(1):58–66, 1959.
- [15] DORLING, K.; HEINRICHS, J.; MESSIER, G. G.; MAGIEROWSKI, S. **Vehicle routing problems for drone delivery.** *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, 2016.
- [16] EATON, R.; KATUPITIYA, J.; SIEW, K. W.; HOWARTH, B. **Autonomous farming: Modeling and control of agricultural machinery in a unified framework.** In: *2008 15th International Conference on Mechatronics and Machine Vision in Practice*, p. 499–504. IEEE, 2008.
- [17] EL-KADER, S. M. A.; EL-BASIONI, B. M. M. **Precision farming solution in egypt using the wireless sensor network technology.** *Egyptian Informatics Journal*, 14(3):221–233, 2013.
- [18] FILIPPONE, A. **Flight performance of fixed and rotary wing aircraft.** Elsevier, 2006.
- [19] FLUSHING, E. F.; GAMBARELLA, L.; DI CARO, G. **Search and rescue using mixed swarms of heterogeneous agents: modeling, simulation, and planning.** *IDSIA, Lugano, Switzerland, Tech. Rep*, p. 05–12, 2012.
- [20] FREW, E. W.; BROWN, T. X. **Networking issues for small unmanned aircraft systems.** *Journal of Intelligent and Robotic Systems*, 54(1-3):21–37, 2009.
- [21] FÜGENSCHUH, A.; MÜLLENSTEDT, D. **Flight planning for unmanned aerial vehicles.** 10 2015.
- [22] GORDON, M. S.; KOZLOSKI, J. R.; MALKIN, P. K.; PICKOVER, C. A. **Autonomous drone service system,** Oct. 20 2016. US Patent App. 14/687,306.

- [23] GUERRIERO, F.; SURACE, R.; LOSCRI, V.; NATALIZIO, E. **A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows constraints.** *Applied Mathematical Modelling*, 38(3):839–852, 2014.
- [24] GUIDE, M. U. **The mathworks.** *Inc., Natick, MA*, 5:333, 1998.
- [25] HAJJAJ, S. S. H.; SAHARI, K. S. M. **Review of agriculture robotics: Practicality and feasibility.** In: *2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, p. 194–198. IEEE, 2016.
- [26] HENTATI, A. I.; KRICHEN, L.; FOURATI, M.; FOURATI, L. C. **Simulation tools, environments and frameworks for uav systems performance analysis.** In: *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, p. 1495–1500. IEEE, 2018.
- [27] HO, D.-T.; GRØTLI, E. I.; SHIMAMOTO, S.; JOHANSEN, T. A. **Optimal relay path selection and cooperative communication protocol for a swarm of uavs.** In: *2012 IEEE Globecom Workshops*, p. 1585–1590. IEEE, 2012.
- [28] HO, D.-T.; GRØTLI, E. I.; SUJIT, P.; JOHANSEN, T. A.; SOUSA, J. B. **Optimization of wireless sensor network and uav data acquisition.** *Journal of Intelligent & Robotic Systems*, 78(1):159–179, 2015.
- [29] INTEL. **Intel aero ready to fly drone.** <https://click.intel.com/intel-aero-ready-to-fly-drone-2534.html>. Data acesso: Julho 2018.
- [30] ISAACS, J. T.; VENKATESWARAN, S.; HESPANHA, J.; MADHOW, U.; BURMAN, J.; PHAM, T. **Multiple event localization in a sparse acoustic sensor network using uavs as data mules.** In: *2012 IEEE Globecom Workshops*, p. 1562–1567. IEEE, 2012.
- [31] ISAACS, J. T. **Uav data mule vehicle routing problems in sparse sensor networks.** University of California, Santa Barbara, 2012.
- [32] JANJARASSUK, U.; MASUCHUN, R. **Cost and reliability analyses for the vehicle routing problem.** In: *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, p. 338–341. IEEE, 2017.
- [33] JIANG, F.; SWINDLEHURST, A. L. **Optimization of uav heading for the ground-to-air uplink.** *IEEE Journal on Selected Areas in Communications*, 30(5):993–1005, 2012.

- [34] KANG, D.; CHA, Y.-J. **Autonomous uavs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging.** *Computer-Aided Civil and Infrastructure Engineering*, 33(10):885–902, 2018.
- [35] KIRICHEK, R. **The model of data delivery from the wireless body area network to the cloud server with the use of unmanned aerial vehicles.** In: *ECMS*, p. 603–606, 2016.
- [36] KOUBÂA, A.; QURESHI, B.; SRITI, M.-F.; JAVED, Y.; TOVAR, E. **A service-oriented cloud-based management system for the internet-of-drones.** In: *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, p. 329–335. IEEE, 2017.
- [37] KUMAR, G. A.; RAMYA, M.; OTHERS. **Wedding of robots with agriculture.** In: *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*, p. 1–4. IEEE, 2012.
- [38] LI, J.; GUO, L.; DONG, C. **Constructing small worlds in wsns with uav trajectory optimization.** In: *Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering*, p. 309–313, 2018.
- [39] LIU, X.; LIU, Y.; ZHANG, N.; WU, W.; LIU, A. **Optimizing trajectory of unmanned aerial vehicles for efficient data acquisition: A matrix completion approach.** *IEEE Internet of Things Journal*, 6(2):1829–1840, 2019.
- [40] MARIANI, JOE AND KAJI, JUNKO . **From dirt to data: The second green revolution and the internet of things.** <https://www2.deloitte.com/us/en/insights/deloitte-review/issue-18/second-green-revolution-and-internet-of-things.html#note-sup-9>. Data acesso: março 2020.
- [41] MARTY, J. A. **Vulnerability analysis of the mavlink protocol for command and control of unmanned aircraft.** Technical report, AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF ... , 2013.
- [42] MCTIC. **Plano nacional de iot.** <http://www.mctic.gov.br/mctic/opencms/salaImprensa/noticias/?titulo=IoT>. Data acesso: Julho 2018.
- [43] MENOVAR, H.; GUVENC, I.; AKKAYA, K.; ULUAGAC, A. S.; KADRI, A.; TUNCER, A. **Uav-enabled intelligent transportation systems for the smart city: Applications and challenges.** *IEEE Communications Magazine*, 55(3):22–28, 2017.

- [44] MOZAFFARI, M.; SAAD, W.; BENNIS, M.; DEBBAH, M. **Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage.** *IEEE Communications Letters*, 20(8):1647–1650, 2016.
- [45] MOZAFFARI, M.; SAAD, W.; BENNIS, M.; DEBBAH, M. **Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs.** *IEEE Transactions on Wireless Communications*, 15(6):3949–3963, 2016.
- [46] MOZAFFARI, M.; SAAD, W.; BENNIS, M.; DEBBAH, M. **Mobile unmanned aerial vehicles (uavs) for energy-efficient internet of things communications.** *IEEE Transactions on Wireless Communications*, 16(11):7574–7589, 2017.
- [47] OUAHOUAH, S.; TALEB, T.; SONG, J.; BENZAID, C. **Efficient offloading mechanism for uavs-based value added services.** In: *Communications (ICC), 2017 IEEE International Conference on*, p. 1–6. IEEE, 2017.
- [48] PARROT. **Bebop 2 firmware.** [https://www.parrot.com/files/s3fs-public/firmware/bebop2\\_update.plf](https://www.parrot.com/files/s3fs-public/firmware/bebop2_update.plf). Data acesso: março 2020.
- [49] PARROT. **Take off and hints, drone by drone.** <https://developer.parrot.com/docs/sphinx/drone-requirements.html>. Data acesso: março 2020.
- [50] PARROT. **What is parrot-sphinx.** <https://developer.parrot.com/docs/sphinx/whatisphinx.html>. Data acesso: março 2020.
- [51] PRESTON GRALLA. **Precision agriculture yields higher profits, lower risks.** <https://www.hpe.com/us/en/insights/articles/precision-agriculture-yields-higher-profits-lower-risks-1806.html>. Data acesso: março 2020.
- [52] PURVIS, K. B.; ASTROM, K. J.; KHAMMASH, M. **Estimation and optimal configurations for localization using cooperative uavs.** *IEEE Transactions on Control Systems Technology*, 16(5):947–958, 2008.
- [53] ROBOT SOURCE. **Download the ros robot programming book for free.** <https://community.robotsource.org/t/download-the-ros-robotprogramming-book-for-free/51>. Data acesso: março 2020.
- [54] RODRIGUES, L. S.; CARDOSO, K. V.; DA CM BORGES, V.; DE OLIVEIRA JR, A. C. **Otimização da trajetória de drones como gateway de comunicação para sensores iot.** In: *Anais da VI Escola Regional de Informática de Goiás*, p. 67–78. SBC, 2018.

- [55] RODRIGUES, L. S., M. C. J. A. B. V. C. M. E. A. **Otimização da trajetória com adaptação à autonomia de drones como gateway de comunicação para dispositivos iot.** *Revista de Sistemas de Informação da FSMA*, 23:10–23, 2019.
- [56] RUZ, J. J.; PAJARES, G.; JESUS, M.; AREVALO, O. **UAV trajectory planning for static and dynamic environments.** INTECH Open Access Publisher, 2009.
- [57] SAIZ-RUBIO, V.; ROVIRA-MÁS, F. **From smart farming towards agriculture 5.0: A review on crop data management.** *Agronomy*, 10(2):207, 2020.
- [58] SEIBER, C.; NOWLIN, D.; LANDOWSKI, B.; TOLENTINO, M. E. **Tracking hazardous aerial plumes using iot-enabled drone swarms.** In: *Internet of Things (WF-IoT), 2018 IEEE 4th World Forum on*, p. 377–382. IEEE, 2018.
- [59] SEMIZ, F. **Task assignment and scheduling in uav mission planning with multiple constraints.** *Middle East Technical University, Ankara*, 2015.
- [60] SNOW. **Why drones are the future of the internet of things?** <http://droneanalyst.com/2014/12/01/drones-are-the-future-of-iot/>, 2014. Data acesso: Julho 2018.
- [61] SOUSA, A. M. B. M. D. **Easy peripherals for the internet of things**, 2016.
- [62] STANKOVIC, J. A. **Research directions for the internet of things.** *IEEE Internet of Things Journal*, 1(1):3–9, 2014.
- [63] SUJIT, P.; LUCANI, D. E.; SOUSA, J. B. **Bridging cooperative sensing and route planning of autonomous vehicles.** *IEEE Journal on Selected Areas in Communications*, 30(5):912–922, 2012.
- [64] SUJIT, P.; LUCANI, D. E.; SOUSA, J. **Joint route planning for uav and sensor network for data retrieval.** In: *2013 IEEE International Systems Conference (Sys-Con)*, p. 688–692. IEEE, 2013.
- [65] TERRA. **Apesar de potencial, mercado de drones ainda é incipiente.** <https://www.terra.com.br/noticias/tecnologia/apesar-de-potencial-mercado-de-drones-ainda-e-incipiente,167909f0662618abddb150cdd8bd39f6k86l9pk1.html>. Data acesso: abril 2020.
- [66] TUDO CELULAR. **Mercado de drones no brasil tem potencial de crescimento, mas esbarra em regulamentação.** Data acesso: abril 2020.
- [67] TZOUNIS, A.; KATSOULAS, N.; BARTZANAS, T.; KITTAS, C. **Internet of things in agriculture, recent advances and future challenges.** *Biosystems Engineering*, 164:31–48, 2017.

- [68] VEENA, S.; VAITHEESWARAN, S.; LOKESHA, H. **Towards the development of secure mavs**. In: *ICRAMAV-2014 (3rd International Conference)*, 2014.
- [69] VERMESAN, O.; FRIESS, P. **Internet of things: converging technologies for smart environments and integrated ecosystems**. River Publishers, 2013.
- [70] VIDAL, T.; CRAINIC, T. G.; GENDREAU, M.; PRINS, C. **Heuristics for multi-attribute vehicle routing problems: A survey and synthesis**. *European Journal of Operational Research*, 231(1):1–21, 2013.
- [71] WANG, C.; MA, F.; YAN, J.; DE, D.; DAS, S. K. **Efficient aerial data collection with uav in large-scale wireless sensor networks**. *International Journal of Distributed Sensor Networks*, 11(11):286080, 2015.
- [72] WANG, Z.; SHEU, J.-B. **Vehicle routing problem with drones**. *Transportation research part B: methodological*, 122:350–364, 2019.
- [73] WEINSTEIN, A.; SCHUMACHER, C. **Uav scheduling via the vehicle routing problem with time windows**. In: *AIAA Infotech@ Aerospace 2007 Conference and Exhibit*, p. 2839, 2007.
- [74] WICHMANN, A.; KORKMAZ, T. **Smooth path construction and adjustment for multiple mobile sinks in wireless sensor networks**. *Computer Communications*, 72:93–106, 2015.
- [75] YANG, Q.; YOO, S.-J. **Optimal uav path planning: Sensing data acquisition over iot sensor networks using multi-objective bio-inspired algorithms**. *IEEE Access*, 6:13671–13684, 2018.
- [76] ZANELLA, A.; BUI, N.; CASTELLANI, A.; VANGELISTA, L.; ZORZI, M. **Internet of things for smart cities**. *IEEE Internet of Things journal*, 1(1):22–32, 2014.
- [77] ZENG, Y.; ZHANG, R.; LIM, T. J. **Throughput maximization for uav-enabled mobile relaying systems**. *IEEE Transactions on Communications*, 64(12):4983–4996, 2016.
- [78] ZHAO, Q.; TONG, L. **Quality-of-service specific information retrieval for densely deployed sensor networks**. In: *IEEE Military Communications Conference, 2003. MILCOM 2003*, volume 1, p. 591–596. IEEE, 2003.