INSTITUTO TECNOLÓGICO DE AERONÁUTICA

Isabela Matos Gomes

OTIMIZAÇÃO DE ROTAS DE DRONES USANDO REINFORCEMENT LEARNING

Trabalho de Graduação
2024

Curso de Engenharia Aeroespacial

CDU 629.78

Isabela Matos Gomes

OTIMIZAÇÃO DE ROTAS DE DRONES USANDO REINFORCEMENT LEARNING

Orientador

Prof. Dr. Christopher Shneider Cerqueira (ITA)

Coorientador

Maj. Leonan Entringer Falqueto (ITA)

ENGENHARIA AEROESPACIAL

São José dos Campos
Instituto Tecnológico de Aeronáutica

2024

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão de Informação e Documentação

Matos Gomes, Isabela
Otimização de Rotas de Drones Usando Reinforcement Learning / Isabela Matos Gomes.
São José dos Campos, 2024.
46f.

Trabalho de Graduação – Curso de Engenharia Aeroespacial– Instituto Tecnológico de Aeronáutica, 2024. Orientador: Prof. Dr. Christopher Shneider Cerqueira . Coorientador: Maj. Leonan Entringer Falqueto.

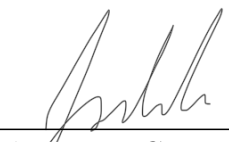
1. Veículos não-tripulado. 2. Aprendizado por Reforço. 3. Aprendizagem (inteligência artificial). 4. Rotas. 5. Logística. 6. Engenharia aeroespacial. I. Instituto Tecnológico de Aeronáutica. II. Título.

REFERÊNCIA BIBLIOGRÁFICA

MATOS GOMES, Isabela. **Otimização de Rotas de Drones Usando Reinforcement Learning**. 2024. 46f. Trabalho de Conclusão de Curso (Graduação) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSÃO DE DIREITOS

NOME DA AUTORA: Isabela Matos Gomes
TITULO DO TRABALHO: Otimização de Rotas de Drones Usando Reinforcement Learning.
TIPO DO TRABALHO/ANO: Trabalho de Conclusão de Curso (Graduação) / 2024



Isabela Matos Gomes
Rua H8A, Ap. 106
12.228-460 – São José dos Campos–SP

OTIMIZAÇÃO DE ROTAS DE DRONES USANDO REINFORCEMENT LEARNING

Essa publicação foi aceita como Relatório Final de Trabalho de Graduação

Isabela Matos Gomes

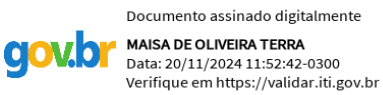
Autora

Prof. Dr. Christopher Shneider Cerqueira (ITA)

Orientador

Maj. Leonan Entringer Falqueto (ITA)

Coorientador



Prof. Dra. Maisa de Oliveira Terra

Coordenadora do Curso de Engenharia Aeroespacial

São José dos Campos, 10 de novembro de 2024.

Aos meus pais por sempre acreditarem em mim e me incentivarem a seguir meus sonhos. E aos meus amigos da graduação que assim como eu encerram uma difícil etapa da vida acadêmica.

Agradecimentos

Agradeço primeiramente aos meus pais, minhas irmãs e ao meu namorado pelo amor, apoio constante e encorajamento ao longo de toda minha trajetória acadêmica. Sem vocês, nada disso seria possível.

Ao meu orientador, Professor Christopher Shneider Cerqueira, e ao meu coorientador, Leonan Entringer Falqueto, agradeço pela orientação, paciência e valiosas contribuições que foram essenciais para o desenvolvimento deste trabalho.

Por fim, agradeço aos amigos que fiz durante a graduação. A amizade e o apoio de vocês foram uma fonte constante de inspiração e motivação.

"Courage is not having the strength to go on;
it's going on when you don't have the strength."
— Theodore Roosevelt

Resumo

Este trabalho apresenta uma metodologia para a otimização de rotas de entrega de drones utilizando técnicas de aprendizado por reforço (RL). Motivado por desastres naturais recentes no Brasil, como as inundações no Rio Grande do Sul, o estudo visa desenvolver soluções que permitam a entrega rápida e eficiente de suprimentos em áreas inacessíveis por meios de transporte convencionais. A pesquisa explora algoritmo de Proximal Policy Optimization (PPO) para resolver problemas de roteamento e de otimização. A aplicação dessa técnica visa reduzir o tempo de entrega e aumentar a eficiência dos drones, melhorando a capacidade de resposta em situações de emergência e contribuindo para a gestão global de desastres.

Abstract

This work presents a methodology for optimizing drone delivery routes using reinforcement learning (RL) techniques. Motivated by recent natural disasters in Brazil, such as the floods in Rio Grande do Sul, the study aims to develop solutions that allow the quick and efficient delivery of supplies to areas inaccessible by conventional means of transport. The research explores the Proximal Policy Optimization (PPO) algorithm to solve routing problems and optimization. The application of these techniques aims to reduce delivery time and increase the efficiency of drones, improving response capacity in emergency situations and contributing to global disaster management.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

RL	Reinforcement Learning
DQN	Deep Q-Network
PPO	Proximal Policy Optimization
TSP	Problema do Caixeiro Viajante
VRP	Problema de Roteamento de Veículos
MDP	Markov Decision Process
AI	Artificial Intelligence

Lista de Símbolos

Q(s, a)	Valor de ação no estado s para a ação a
α	Taxa de aprendizado
γ	Fator de desconto
θ	Parâmetros da rede neural
θ^-	Parâmetros da rede de destino
L(θ)	Função de perda da rede neural
π_θ	Nova política
R	Recompensa
P(s'|s, a)	Função de transição de estado
S	Conjunto de estados
A	Conjunto de ações
β	Parâmetro de controle de regularização

Sumário

1 Introdução

1.1 Contexto e motivação

Nos últimos anos, o Brasil enfrentou uma série de desastres naturais que afetaram inúmeras vidas e causaram uma significativa destruição. Recentemente, o estado do Rio Grande do Sul sofreu com uma tragédia natural devastadora, em que o nível de água de vários rios subiu causando o alagamento de diversas cidades. Com isso destacou-se a necessidade urgente de soluções eficientes para resposta a emergências. Nessas situações de desastre, a entrega rápida de suprimentos críticos, como alimentos, água, medicamentos e equipamentos de resgate, é essencial para salvar vidas e mitigar qualquer sofrimento humano.

FIGURA 1.1 – Mapa da região afetada pelas enchentes no Rio Grande do Sul, ilustrando as áreas impactadas e os principais pontos de referência para a análise logística no estudo. Fonte:(ESCOLA, 2024).

Nesses casos, muitas vezes as infraestruturas, como estradas e pontes, são danificadas ou destruídas, tornando o acesso terrestre impossível. O atual caso no sul do país evidenciou essa possibilidade uma vez que diversas áreas se tornaram completamente inacessíveis por meios de transporte convencionais. Desse modo, destaca-se a possibilidade

da utilização de drones graças à sua capacidade de acessar áreas inacessíveis ou perigosas para veículos terrestres e aeronaves tripuladas podendo assim entregar suprimentos diretamente às pessoas necessitadas em praticamente qualquer lugar. No entanto, esse estudo também foca que para maximizar a eficácia dos drones, é crucial otimizar suas rotas e a relação combustível/carga útil.

Dentre os possíveis benefícios deste estudo, destaca-se sua eficiência e eficacia em respostas de emergência. Em situações de desastre, cada minuto é crucial. Desse modo, otimizar as rotas dos drones pode levar a redução significativa do tempo de entrega de suprimentos críticos, garantindo que a ajuda chegue mais rapidamente às áreas afetadas.

Além disso, tem-se que a otimização da relação combustível/carga útil permite que os drones transportem mais recursos em cada voo, reduzindo a necessidade de múltiplas viagens e aumentando a eficiência. A gestão do consumo de combustível é de extrema importância para a sustentabilidade das operações de resposta. Um menor consumo de combustível significa que os drones podem assim operar por mais tempo sem necessidade de reabastecimento, o que é especialmente importante em áreas remotas ou isoladas. Já a otimização da carga útil garante que os drones não apenas não só maximizem a quantidade de suprimentos transportados, mas também operem de maneira segura, prevenindo falhas devido ao excesso de peso.

Faz se importante ressaltar que, embora este estudo seja motivado pela recente tragédia no Rio Grande do Sul, o objetivo do estudo feito é identificar e desenvolver soluções que possam ser aplicadas e adaptadas para um cenário qualquer de desastre natural. Em resumo, o estudo da otimização de rotas de drones e da relação combustível/carga útil não só tem o potencial de aliviar o sofrimento humano em situações de emergência, mas também contribui para o desenvolvimento de estratégias de gestão de desastres em um contexto global.

1.2 Objetivo

Neste contexto de catástrofes imprevisíveis, esse trabalho tem como objetivo aplicar o Reinforcement Learning para gerar soluções rápidas e eficazes baseadas em um principio: sanar as demandas de diferentes locais o mais rápido possível.

Além disso, uma grande parte do problema é garantir que ele seja facilmente adaptável para diferentes cenários. Assim, a implementação levou em conta a necessidade de fácil mudança das configurações inciais do problema.

1.3 Literatura

A pesquisa de aprendizagem por reforço (RL) fez progressos significativos, especialmente para aplicações em sistemas autônomos e robótica móvel, como navegação de drones. Sutton e Barto (2018) forneceram uma estrutura para a aprendizagem colaborativa, explicando conceitos-chave como processos de decisão de Markov (MDPs) e propostas de valor que são essenciais para a criação de problemas de tomada de decisão em ambientes dinâmicos. Essas estruturas teóricas foram aplicadas com sucesso em muitos campos, incluindo controle de robôs autônomos(SUTTON; BARTO, 2018)..

Recentemente, Goodfellow, Bengio e Courville (2016) focaram no papel das redes neurais profundas na aprendizagem ativa, levando ao conceito de aprendizagem profunda (Deep RL), que combina redes neurais com técnicas de RL para enfrentar os desafios regulatórios contínuos. Os avanços levaram ao desenvolvimento de agentes capazes de resolver problemas altamente complexos, como o AlphaGo(GOODFELLOW *et al.*, 2016), e como drones e veículos autônomos.

No campo da robótica em particular, estudos como Puterman (2014) e Bellman (1957) discutem como melhorar os pontos fracos e a utilidade do modelo MDP na concepção de políticas de controle melhor (PUTERMAN, 2014; BELLMAN, 1957). Esses primeiros artigos apoiam o desenvolvimento de algoritmos modernos de otimização de políticas, como Proximal Policy Optimization (PPO), que é amplamente utilizado em RL devido à sua eficiência e precisão, especialmente na área de robôs móveis, (SCHULMAN *et al.*, 2017a).

Para uma compreensão geral dos métodos de controle e otimização, o trabalho de Bishop (2006) é importante porque fornece métodos de aprendizado de máquina que complementam a RL e uma base para a construção de modelos preditivos (BISHOP, 2006). Além disso, o uso de redes neurais em robótica é revisado por Haykin (2009), que discute tarefas de construção e operação utilizadas em sistemas como drones, onde a capacidade de adaptação e aprendizagem em tempo real é importante(HAYKIN, 2009).

1.4 Contribuição

A maior contribuição alcançada por esse trabalho é a adaptação de um problema real onde se utilizam drones para a distribuição de recursos em um problema de Reinforcement Learning. Assim, permitindo o advento de novas soluções que levariam a uma resposta mais rápida e eficiente nesses casos de tragedias naturais.

1.5 Estrutura do trabalho

O restante desta tese é organizado da seguinte forma:

- Capítulo 2 explica a teoria de Machine Learning e Redes Neurais.
- Capítulo 3 descreve a delimitação do problema e a metodologia utilizada.
- Capítulo 4 expõe os resultados obtidos.
- Capítulo 5 mostra as conclusões e possíveis melhorias futuras.

2 Revisão Bibliográfica

2.1 Machine Learning

O machine learning é um subcampo da inteligência artificial que lida com a criação de algoritmos e modelos que podem aprender com os dados, sem serem programados explicitamente para fazer uma tarefa específica. Ou seja, ao invés de seguir regras estritas, os sistemas de ML identificam padrões e fazem previsões ou decisões em função dos dados (MURPHY, 2012).

Existem três principais tipos de aprendizado em machine learning (HASTIE *et al.*, 2009):

1. Aprendizado supervisionado: quando o modelo é treinado baseado uma solução já desejada.
2. Aprendizado não supervisionado: quando o modelo é treinado sem solução desejada.
3. Aprendizado por reforço (Reinforcement Learning): quando o modelo toma decisões através de tentativa e erro, recebendo recompensas ou penalidades baseadas nas suas ações.

Neste trabalho foi utilizado o Reinforcement Learning, mas é importante ressaltar que a recompensa final não é a saída desejada, ela é somente uma métrica utilizada para avaliar o algorítimo.

Nas próximas seções desse capítulo serão descritas os conceitos e técnicas utilizadas, dentre eles o de redes neurais, os fundamentos de Reinforcement Learning e o algorítimo Proximal Policy Optimization.

2.2 Redes Neurais

As redes neurais são modelos computacionais inspirados pela estrutura do cérebro humano. Elas consistem em unidades, chamadas ”neurons”, que se juntam em camadas.

Os neurônios são interconectados uns com os outros e processam informações de forma um tanto similar à forma como o sistema nervoso biológico opera. Cada conexão entre os neurônios carrega um ”peso”ajustável que define a intensidade na qual um neurônio influencia o próximo (HAYKIN, 2009).

Um único neuron não é capaz de completar tarefas complexas, no entanto uma rede neural composta por múltiplos neurons organizados em camadas é. A Figura 2.1 mostra a estrutura de uma rede neural, evidenciando as camadas de entrada, as camadas intermediarias e as camadas de saída.

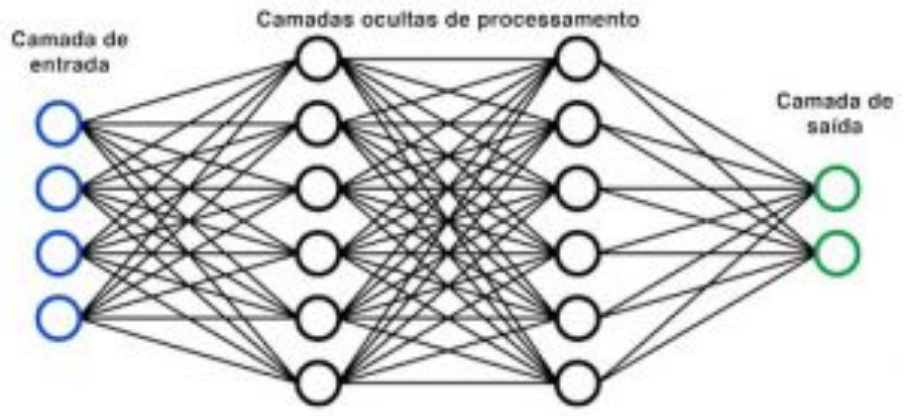


FIGURA 2.1 – Representação de uma rede neural. Fonte:(OPENCADD, 2024).

As redes neurais são essenciais para problemas complexos como o roteamento de drones, pois permitem que o modelo lide com uma grande quantidade de variáveis e aprenda relações não lineares entre elas, algo fundamental para capturar as complexidades de cenários reais (GOODFELLOW *et al.*, 2016).

2.3 Reinforcement Learning

O Aprendizado por Reforço (Reinforcement Learning - RL) é uma subárea do aprendizado de máquina onde um agente interage com o ambiente através de ações, e cada ação gera como resposta um novo estado e uma recompensa. O agente explora o ambiente e vai recebendo feedbacks de cada ação até atingir seu objetivo final de maximizar a recompensa cumulativa. (SUTTON; BARTO, 2018)

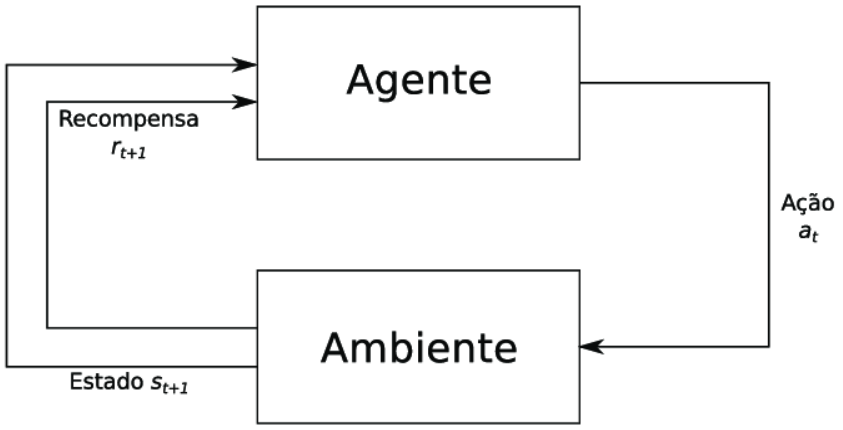


FIGURA 2.2 – Mapa da região afetada pelas enchentes no Rio Grande do Sul, ilustrando as áreas impactadas e os principais pontos de referência para a análise logística no estudo. Fonte:(SANTOS, 2019).

2.4 Markov Decision Process

O Markov Decision Process é um modo muito usado de estruturar problemas de Reinforcement Learning. É um modelo matemático que representa e resolve problemas de decisão sequenciais em que o agente interage com um ambiente com o objetivo de maximizar uma recompensa cumulativa ao longo do tempo (PUTERMAN, 2014) (BELLMAN, 1957). Um MDP é formalmente definido por cinco elementos principais:

1. Estados (S): representa as possíveis situações ou configurações do ambiente em que o agente poderá estar. Cada estado possui algumas informações, que descrevem o ambiente e que são observados pelo agente para que este faça a sua escolha.

2. Ações (A): O conjunto de ações descreve quais escolhas o agente pode fazer a partir de um estado específico. Em cada estado, o agente tem uma ou mais escolhas para selecionar.

3. Função de Transição de Estado (P): fornece a probabilidade de transições entre estados após a execução de uma determinada ação. Esta função define a probabilidade $P(s'\backslash s, a)$ de um agente chegar ao estado s' executando a ação a no estado s.

4. Função de Recompensa (R): associa uma recompensa numérica a cada transição entre estados. É o que o agente recebe por executar uma ação em um determinado estado. Cada recompensa é definida de modo a ajudar e orientar o agente a tomar ações que são favoráveis ao objetivo final.

5. Fator de Desconto (γ): representa a importância das recompensas futuras em comparação com as recompensas imediatas. Definido entre 0 e 1, é utilizado para calcular

o valor presente das recompensas futuras. Quando γ é próximo de 1, as recompensas futuras são quase tão importantes quanto as recompensas imediatas. Já quando γ é próximo de 0, o agente tende a priorizar recompensas de curto prazo.

2.5 Proximal Policy Optimization (PPO)

O Proximal Policy Optimization (PPO) é um algoritmo de aprendizado por reforço baseado em uma política que busca a otimizaçao diretamente na política de decisão do agente. O grande beneficio desse método é que ele melhora a estabilidade e a eficiência do treinamento, uma vez que ele vai ajustando a política sem grandes mudanças abruptas. (SCHULMAN *et al.*, 2017b)

O PPO utiliza uma função de perda baseada na razão de probabilidade entre a nova política π_θ e a política antiga $\pi_{\theta_{\text{old}}}$ dada por:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{2.1}$$

onde:

- $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ é a razão de probabilidade.
- $\hat{A}_t$ é a vantagem estimada.
- ϵ é um hiperparâmetro que controla o quanto a nova política pode diferir da antiga.

A função de perda $L^{\text{CLIP}}(\theta)$ promove atualizações estáveis, o que limita a grandeza das mudanças na política, assim ajudando a evitar oscilações e divergências durante o treinamento.

2.6 Aplicações de RL em Problemas de Otimização de Rotas

O uso de RL para resolver problemas de otimização de rotas tem se mostrado promissor, especialmente em cenários onde os métodos tradicionais enfrentam limitações. Alguns exemplos notáveis incluem:

2.6.1 Problema do Caixeiro Viajante (TSP)

O Problema do Caixeiro Viajante (Traveling Salesman Problem - TSP) é um problema de otimização combinatória clássico em que se tenta descobrir a rota mais curta que um indivíduo pode viajar de uma cidade para outra, e então retorna para a cidade de origem, como esauematizado na Figura 2.3. Este é um problema NP-difícil, o que significa que não há algoritmo eficiente comprovado que resolva todos os casos em tempo polinomial, então, para um número muito grande de cidades, soluções ótimas são basicamente impossíveis.

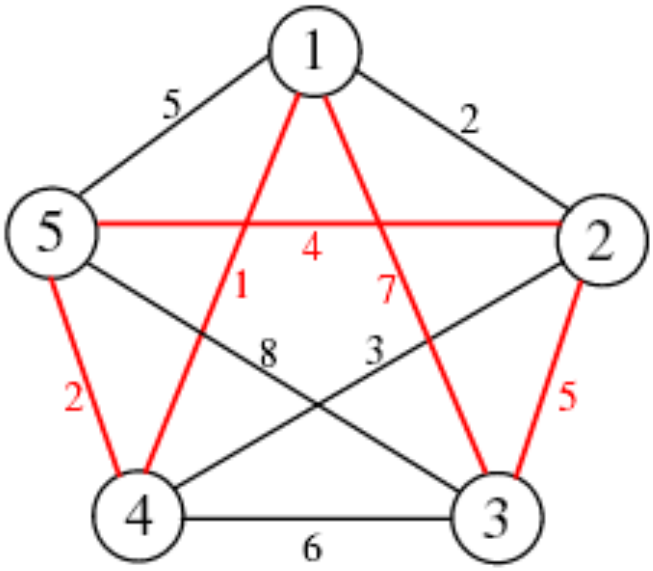


FIGURA 2.3 – Esquematização do Problema do Caixeiro Viajante (TSP). Fonte: (MORO *et al.*, 2018)

(KOOL *et al.*, 2019) propuseram uma abordagem de aprendizado baseada em RL com uma rede neural de atenção que aprende a construir soluções para o TSP. O modelo é treinado de forma iterativa com uma combinação de RL e busca local para gerar soluções competitivas, quando comparadas técnicas heurísticas clássicas. Uma vantagem deste método é que ele pode ser generalizado para execuções de tamanho variado do TSP com pouco re-treinamento.

2.6.2 Problema de Roteamento de Veículos (VRP)

O Problema de Roteamento de Veículos (Vehicle Routing Problem - VRP) é uma generalização do TSP, no sentido de que, em vez de um veículo, uma frota de veículos com capacidades finitas deve ser roteirizada para servir a múltiplos clientes. Ou, de outra forma, o problema é encontrar a rota ideal para cada veículo, de tal maneira que o custo global para encontrar todas as restrições de capacidade e tempo seja minimizado. Um esquema desse problema pode ser visto na Figura 2.4.

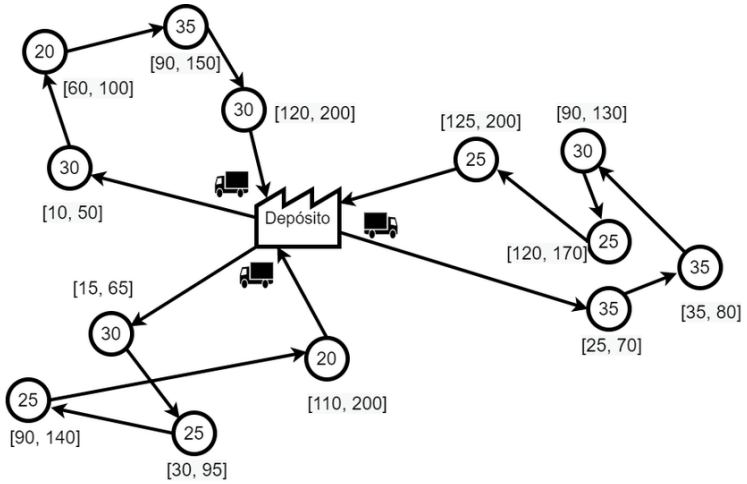


FIGURA 2.4 – Esquema do Problema de Roteamento de Veículos (VRP). Fonte: (MAIA; BESSANI, 2020)

(KILBY *et al.*, 2002) apresentam o uso de RL para resolver o VRP com janelas de tempo. O esquema depende de RL, com um método baseado em busca local guiada, a fim de melhorar a qualidade das soluções produzidas. O RL é empregado para a aprendizagem de políticas que, por sua vez, melhoram a busca local, de modo que a técnica é mais eficiente em termos de custo computacional e tempo do que as heurísticas clássicas.

3 Metodologia

3.1 Delimitação do problema

O projeto é delimitado como um sistema de roteamento otimizado para drones que precisam atender a demandas em múltiplos pontos de entrega, considerando restrições como consumo de bateria, tempo de missão e capacidade de carga. Para isso, temos uma base que funciona como um centro de distribuição e pontos que possuem demandas. Esse centro de distribuição serve como um ponto estratégico onde os drones podem recarregar suas baterias e carregar os recursos necessários para atender às demandas dos pontos de entrega. O objetivo final é minimizar o tempo total das missões e o consumo de bateria, atendendo de forma eficiente as demandas de cada ponto.

Um exemplo de como isso poderia ser aplicado é mostrado na Figura 3.1. Nas enchentes citadas na Seção 1, um dos grandes problemas foi a distribuição de medicamentos entre hospitais e centros médicos da região. Assim, a Base Aérea de Canoas (pino roxo da Figura 3.1) que se tornou um dos principais centros de distribuição poderia ser usada como a base do código, e os hospitais (restante dos pinos) seriam os diferentes pontos, cada um com uma determinada demanda de artigos médicos.

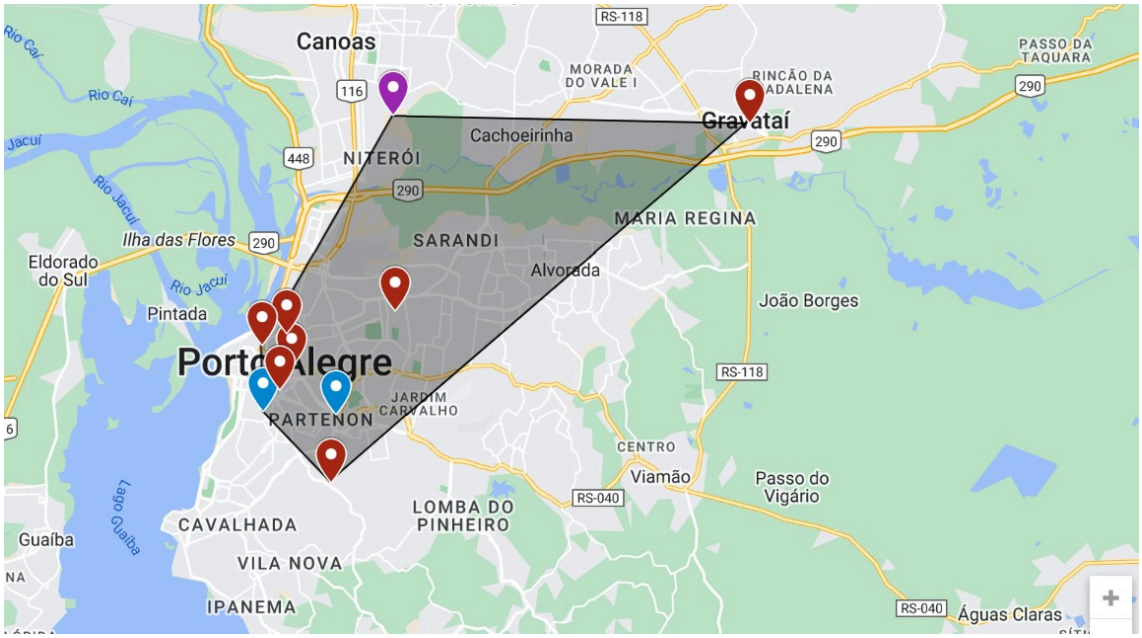


FIGURA 3.1 – Mapa da região afetada pelas enchentes do Rio Grande do Sul. Esta figura mostra a Base Aérea de Canoas e os principais hospitais de Porto Alegre.

Além disso, é importante definir também as condições especificas de trabalho dos drones. Para o estudo foi considerado um drone genérico uma vez que um grande objetivo é implementar uma solução facilmente adaptável que poderia ser aplicada a diversas situações de emergência.

3.2 Principais bibliotecas utilizadas

Para implementar e treinar o modelo, foram utilizadas várias bibliotecas fundamentais, dentre elas se destacam:

1. Stable-Baselines3: Utilizada para o treinamento do modelo de Reinforcement Learning, com o algoritmo Proximal Policy Optimization (PPO). Essa biblioteca facilita a implementação de algoritmos de aprendizado por reforço, permitindo ajustar e refinar o comportamento do agente em um ambiente simulado.

2. Gymnasium: Usada para a criação e configuração do ambiente. Essa biblioteca permite a integração do drone como agente no ambiente, definindo estados, ações e recompensas no processo de aprendizagem.

3. NumPy: Utilizada para qualquer manipulação matemática de dados e cálculos essenciais à simulação, como o cálculo de distâncias entre pontos e o consumo de bateria do drone.

3.3 Implementação do Código

A implementaçao do codigo foi dividida em três partes:

1. Configurações: onde é feita toda a definição do inputs do problema.
2. Ambiente: onde é feita a definição dos possíveis estados, ações e das recompensas.
3. Treinamento: onde o algorítimo PPO é aplicado para o treinamento.

3.3.1 Configurações

Nessa parte são definidos os inputs necessários para o código mostrados na Tabela 3.1. Os valores podem ser escolhido de modo a melhor simular o ambiente real.

Inputs
Número de pontos
Coordenadas dos pontos
Demandas de cada ponto
Bateria máxima
Bateria inicial
Peso máximo
Velocidade máxima
Tempo máximo
Coeficientes do consumo de bateria: $[\alpha, \beta, \gamma]$
Peso do drone

TABELA 3.1 – Parâmetros e configurações iniciais necessários que são utilizados como base para a análise e simulação.

O tempo de cada interação do drone é calculado pela Equação 3.1, e os coeficientes $[\alpha, \beta, \gamma]$ são os coeficientes utilizados para o calculo do consumo de bateria expresso pela Equação 3.2.

$$time = \frac{\text{distância entre os pontos}}{\text{velocidade escolhida pelo agente}} \tag{3.1}$$

$$\begin{aligned} \text{consumo de bateria} = \alpha * (\text{velocidade}) \\ + \beta * (\text{tempo de trajeto}) \\ + \gamma * (\text{peso carregado} + \text{peso do drone} \end{aligned} \tag{3.2}$$

3.3.2 Ambiente

Nessa parte são descritos os elementos principais do Markov Decision Process.

3.3.2.1 MDP Estado

Como descrito na Seção 2.4, o estado é uma representação do ambiente no momento atual do drone, e para o cado deste projeto, o estado atual é caracterizado por 4 variáveis:

1. Posição Atual: Coordenadas do drone no mapa (x, y).
2. Bateria Restante: Nível de bateria atual do drone.
3. Carga Carregada: Quantidade de peso que o drone está transportando.
4. Demanda dos Pontos: Quantidade de demanda ainda não atendida em cada ponto.

Assim, ao observar essas variáveis, o algorítimo toma as decisões que ele acredita que leva a maior recompensa cumulativa.

3.3.2.2 MDP Ações

Como descrito na Seção 2.4, as ações representam o que o algorítimo escolhe de modo a completar o seu objetivo. No caso deste projeto, cada ação é descrita por 4 escolhas:

1. Selecionar um ponto de entrega para visitar: Representa o próximo ponto de destino do drone.
2. Escolher a velocidade de deslocamento: Controla o consumo de bateria (velocidades mais altas consomem mais energia).
3. Decidir a quantidade de recarga (ao retornar à base): Controla o tempo gasto para recarregar a bateria.
4. Escolher o peso de carga (ao retornar à base): Quantidade de carga que o drone coleta para sua próxima rota.

3.3.2.3 MDP Recompensas

Como descrito na Seção 2.4, as recompensas funcionam de modo a gratificar ou penalizar ações que são ou não favoráveis ao resultado desejado. Na implementação, foram escolhidas as recompensas mostradas na Tabela 3.2.

Recompensas
Recompensa proporcional à demanda atendida
Penalidade proporcional ao consumo de bateria
Penalidade para retornos à base com carga não utilizada
Penalidade por revisitas e recompensa para novos pontos
Penalidade por tempo de missão e recompensa por rapidez
Bônus pela Conclusão da Missão
Penalidade tempo maior que tempo máximo

TABELA 3.2 – Diferentes recompensas ou penalidades definidas na implementação do problema.

3.3.2.4 MDP Política de Transição de Estado

A política de transição de estado define como o agente se desloca entre diferentes estados ao tomar uma ação. Neste projeto, a função de transição é determinística: uma vez que o agente executa uma ação em um estado inicial s, ele necessariamente alcança o estado de destino s', sem influência de fatores estocásticos ou aleatórios. Isso significa que, ao tomar uma ação específica, o drone sempre chega ao mesmo próximo estado, simplificando o processo de tomada de decisão ao eliminar incertezas na transição. Assim a função é sempre

$$P(s's, a).$$

3.3.2.5 MDP Fator de Desconto

O fator de desconto γ é um parâmetro essencial no Reinforcement Learning que determina a importância das recompensas futuras em comparação com as recompensas imediatas. Neste projeto, o fator de desconto foi pré-definido pela biblioteca Stable-Baselines3, utilizada na implementação do modelo.

3.3.3 Treinamento

Nessa parte o algoritmo de Proximal Policy Optimization é implementado. O PPO foi configurado para um total de 1.000.000 passos de treinamento. Durante o treinamento, o

drone foi instruído a tomar decisões sobre quais rotas seguir, quando recarregar e quanto peso transportar, buscando maximizar a eficiência das missões.

4 Resultados

Os resultados do projeto foram obtidos através da avaliação de diferentes versões do modelo, com ajustes progressivos para otimizar seu desempenho. Para cada versão, foram testados dois cenários com diferentes conjuntos de pontos, denominados Caso 1 e Caso 2. Para a analise desses resultados, eles foram avaliados com base nas seguintes quatro métricas de desempenho:

1. Distância total percorrida: Mede o quanto o drone viajou durante a missão.
2. Tempo Total de Missão: Tempo da total da solução final do algorítimo.
3. Consumo Total de Bateria: A quantidade de energia utilizada pelo drone ao longo da missão, considerando fatores como velocidade e carga transportada.
4. Atendimento de Demanda: Avalia se todas as demandas dos pontos de entrega foram atendidas.

4.1 Caso 1

O primeiro conjunto de pontos analisado, Caso 1, é mostrado na Figura 4.1 onde podemos ver os 8 pontos e a base localizada no ponto (0,0).

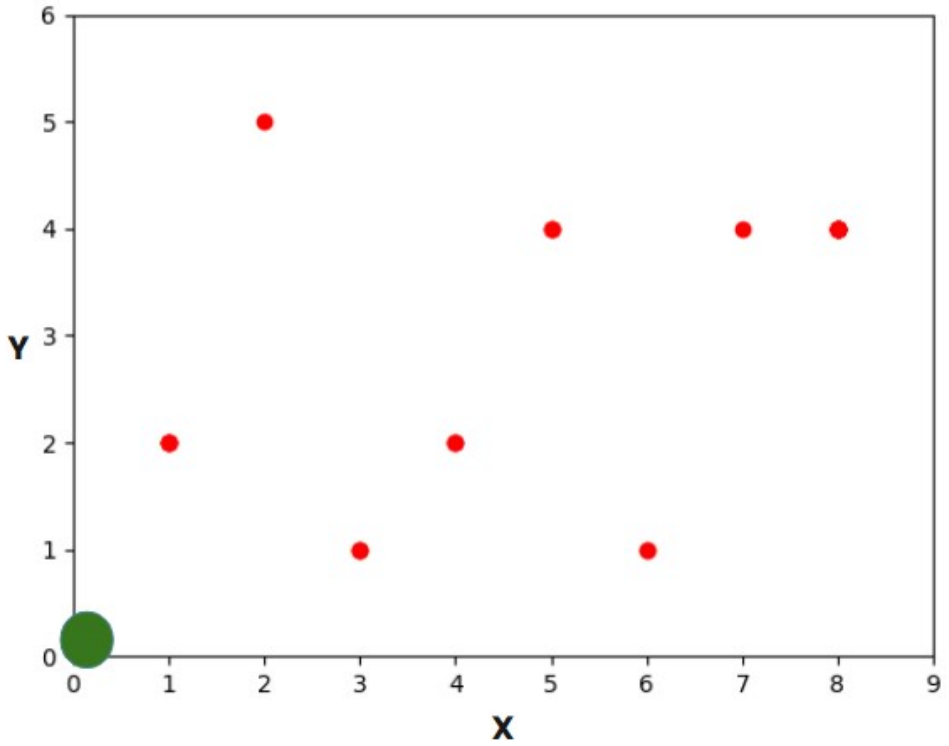


FIGURA 4.1 – Primeiro conjunto de pontos analisado. Os pontos vermelhos representam os diferentes pontos com demandas a ser atendidas e o ponto verde representa a base onde o drone pode recarregar seus suprimentos e a sua bateria.

Como dito na Seção 3.3.1, é necessária também a definição dos inputs do problema. A tabela 4.1 mostra os inputs desse caso.

Inputs	
Número de pontos	8
Coordenadas dos pontos	[1,2], [2,5], [3,1], [4,2], [5,4], [6,1], [7,2], [8,4]
Demandas de cada ponto	$[10, 15, 20, 25, 10, 15, 5, 10]$
Bateria máxima	100
Bateria inicial	100
Peso máximo	100
Velocidade máxima	10
Tempo máximo	10000
Coeficientes do consumo de bateria	$[\alpha, \beta, \gamma] = [0.1, 0.1, 0.1]$
Peso do drone	1

TABELA 4.1 – Parâmetros e configurações iniciais definidos para o estudo do Caso 1, mostrando os valores específicos utilizados como base para a análise e simulação deste cenário.

É importante ressaltar que essa primeiras analises foram feitas com valores normalizados, ou seja, nenhum desses inputs possui dimensão. Para uma aplicação mais realista em

que se deseja determinadas dimensões, se faz necessário apenas que o restante dos inputs sejam definidos com dimensões coerentes.

4.1.1 Versão 1 do Código

As primeiras análises do problema foram feitas com a versão inicial do código. As Tabelas 4.2 e 4.3 mostram respectivamente a demilitação das possíveis ações e a definição do valor das recompensas e penalidades dessa primeiro modelo.

Ações	Valores
Ponto de entrega para visitar	Todos os pontos
Velocidade de deslocamento	0 a 10
Quantidade de recarga da bateria	0 a 100
Escolher o peso de carga	0 a 100

TABELA 4.2 – Delimitação das possíveis ações do agente de RL para a versão 1 do código.

Recompensas	
Recompensa proporcional à demanda atendida	10*demanda atendida
Penalidade proporcional ao consumo de bateria	-10 (consumo bateria / 100)
Penalidade para retornos à base carga não utilizada	-20
Penalidade por revisitas e recompensa para novos pontos	-10 ou +5
Penalidade por tempo de missão e recompensa por rapidez	-2*tempo ou +50
Bônus pela Conclusão da Missão	+1000
Penalidade tempo maior que tempo máximo	-100

TABELA 4.3 – Definição do valor das recompensas e penalidades para a versão 1 do código.

Esses resultados obtidos são expressos na Tabela 4.4.

Resultados	
Distância total	31.61
Consumo total de bateria	14.40
Tempo de missão	4.46
Todas demandas atendidas	NÃO

TABELA 4.4 – Resultados obtidos das métricas de desempenho para a versão 1 do código.

Os resultados obtidos revelaram-se significativamente abaixo das expectativas. Assim, observa-se que o algoritmo demonstrou limitações consideráveis em relação à otimização

das rotas. Isso se da devido ao fato que a política de ações permitia que drone escolhesse uma velocidade de deslocamento entre 0 e 10, o que levou o agente a selecionar a velocidade zero em muitos momentos, resultando na estagnação do drone em uma posição fixa. Além disso, ao selecionar a velocidade zero, o custo energético calculado era extremamente elevado, uma vez que as penalidades e recompensas dependiam fortemente do tempo de missão e do consumo de bateria, dependentes da velocidade.

4.1.2 Versão 2 do Código

Na segunda versão, ajustes foram realizados nas ações do agente, restringindo a velocidade de deslocamento entre 5 e 10. Essa modificação foi feita visando uma redução na frequência de estagnação, pois o drone agora teria que se mover entre os pontos. A Tabela 4.5 mostra a nova demilitação das ações. Nessa versão não houve mudanças nos valores das recompensas, foram utilizadas as mesmas da versão 1 mostradas na Tabela 4.3.

Ações	Valores
Ponto de entrega para visitar	Todos os pontos
Velocidade de deslocamento	5 a 10
Quantidade de recarga da bateria	0 a 100
Escolher o peso de carga	0 a 100

TABELA 4.5 – Delimitação das possíveis ações do agente de RL para a versão 2 do código.

Os resultados obtidos com essa nova versão são expressos na Tabela 4.6.

Resultados	
Distância total	33.45
Consumo total de bateria	291.50
Tempo de missão	2137.12
Todas demandas atendidas	NÃO

TABELA 4.6 – Resultados obtidos das métricas de desempenho para a versão 2 do código.

Os dados finais ainda não alcançaram um padrão satisfatório para a analise pretendida. Os ajustes permitiram uma melhoria em relação à versão anterior uma vez que o drone passou a percorrer uma distância maior para atender mais pontos. No entanto, o modelo ainda apresentava problemas de eficiência, como revisitas frequentes aos mesmos pontos e retornos desnecessários à base.

4.1.3 Versão 3 do Código

Em vista os problemas anteriores, na versão final a política de recompensas foi ajustada para priorizar o atendimento das demandas e penalizar retornos desnecessários à base, além de incentivar o drone a visitar novos pontos. Os novos valores das recompensas são expressos na Tabela 4.7.

Recompensas	
Recompensa proporcional à demanda atendida	50*demanda atendida
Penalidade proporcional ao consumo de bateria	-10 (consumo bateria / 100)
Penalidade para retornos à base carga não utilizada	-100
Penalidade por revisitas e recompensa para novos pontos	-100 ou +5
Penalidade por tempo de missão e recompensa por rapidez	-2*tempo ou +50
Bônus pela Conclusão da Missão	+1000
Penalidade tempo maior que tempo máximo	-100

TABELA 4.7 – Definição do valor das recompensas e penalidades para a versão 3 do código.

Os resultados obtidos nesta ultima versão sao mostrados na Tabela 4.8

Resultados	
Distância total	129.29
Consumo total de bateria	733.01
Tempo de missão	76.94
Todas demandas atendidas	SIM

TABELA 4.8 – Resultados obtidos das métricas de desempenho para a versão 2 do código.

A versão 3 demonstrou que, ao ajustar as recompensas e penalidades, o drone conseguiu atender todas as demandas dos pontos de entrega de forma eficiente, com distâncias e tempos de missão significativamente melhorados. A política de recompensas ajudou a evitar revisitas e a reduzir os retornos à base, garantindo que o agente priorizasse novas entregas e concluísse a missão de forma mais rápida e econômica.

Desse modo, a versão final apresentou resultados superiores, com uma rota otimizada e menor consumo de recursos.

O resulto da rota final é: (0, 0), (6, 1), (0, 0), (0, 0), (5, 4), (0, 0), (4, 2), (8, 4), (0, 0), (0, 0), (0, 0), (0, 0), (4, 2), (5, 4), (0, 0), (1, 2), (0, 0), (8, 4), (1, 2), (0, 0), (0, 0), (3, 1), (8, 4), (3, 1), (0, 0), (1, 2), (0, 0), (4, 2), (8, 4), (0, 0), (0, 0), (0, 0), (8, 4), (0, 0), (1, 2), (8, 4), (5, 4), (0, 0), (8, 4), (0, 0), (0, 0), (0, 0), (0, 0), (8, 4), (0, 0), (0, 0), (0, 0), (2, 5), (8, 4), (0, 0), (0, 0), (0, 0), (0, 0), (0, 0), (1, 2), (0, 0), (0, 0), (7, 4), (0, 0), (0, 0), (0, 0), (3, 1), (0, 0), (0, 0), (0, 0), (0, 0), (0, 0), (8, 4), (0, 0), (0, 0), (0, 0), (0, 0), (6, 1).

Além disso, a Figura 4.2 mostra todos os trajetos feitos pela solução final do drone.

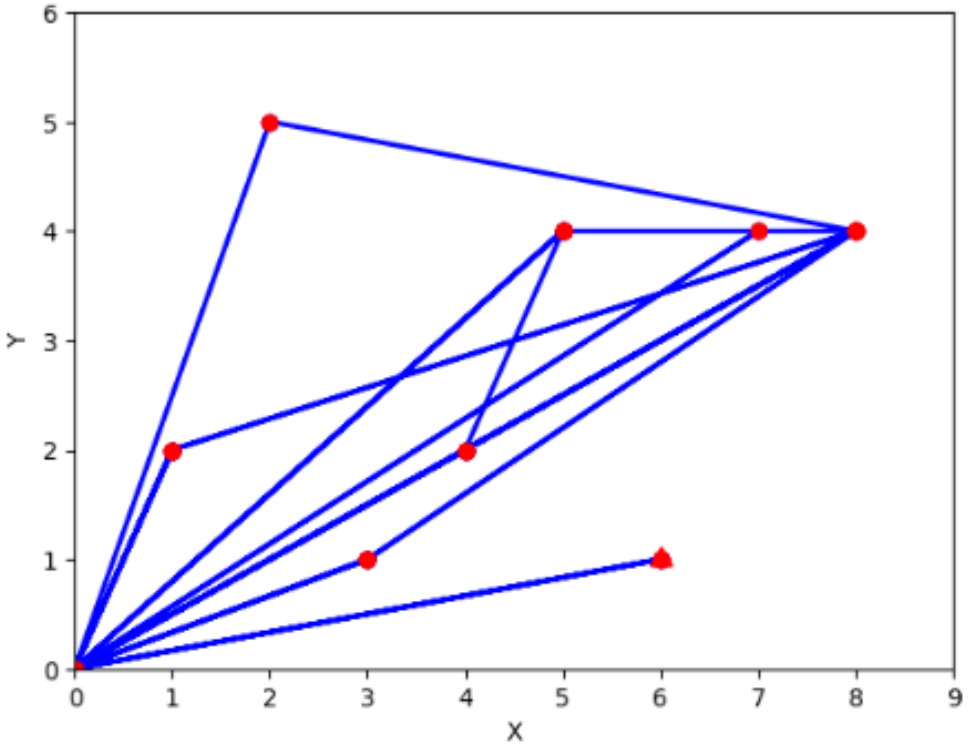


FIGURA 4.2 – Todos os trajetos feitos pela solução final do caso 1, usando a versão 3 do código. Os pontos vermelhos representam os pontos com demandas e a base, e os traços azuis representam os caminhos feitos.

4.1.4 Análise Comparativa

A Tabela 4.9 resume os principais indicadores de desempenho para cada versão.

	Versão 1	Versão 2	Versão 3
Distância total	31.61	33.45	129.29
Consumo total de bateria	14.40	291.50	733.01
Tempo de missão	4.46	2137.12	76.94
Todas demandas atendidas	NÃO	NÃO	SIM

TABELA 4.9 – Análise comparativa dos resultados obtidos das métricas de desempenho para as 3 versões do código para o primeiro conjunto de pontos, caso 1.

Esses resultados indicam que os ajustes progressivos nas ações e recompensas, especialmente na versão 3, foram fundamentais para alcançar uma solução eficiente para ambos os casos. Isso demonstra que o uso de algoritmos de Reinforcement Learning, como o PPO, com configurações de recompensa adequadas, é uma abordagem eficaz para otimização de problemas logísticos complexos.

Esses resultados destacam que, por meio da adaptação progressiva de ações e recompensas por meio das várias versões, especialmente a versão 3, foi possivel a construção de uma solução eficiente no cenário analisado. Foi a partir desse ajuste contínuo que o modelo de Reinforcement Learning pôde se adaptar precisamente aos detalhes do problema e alcançar a melhoria de seu desempenho. Isso também justifica por que algoritmos de aprendizado por reforço, como o PPO, podem ser aplicados a problemas logísticos muito complexos. Usando uma função de recompensa bem projetada, o PPO incentivará o modelo a alcançar soluções muito melhores, evitando obstáculos típicos com esses tipos de problemas e tendo grande potencial para aplicações logísticas e operacionais em larga escala.

4.2 Caso 2

Para um estudo mais abrangente foi analisado também um segundo conjunto de pontos, o Caso 2, na Figura 4.1 onde podemos ver os novos 8 pontos e a base localizada no ponto (0,0).

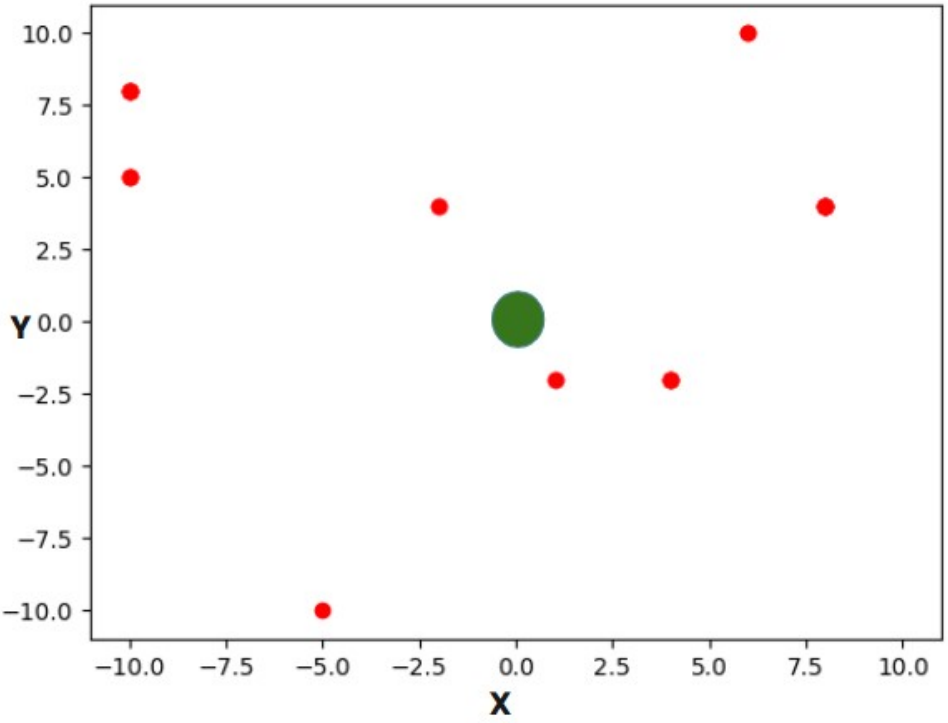


FIGURA 4.3 – Segundo conjunto de pontos analisado. Os pontos vermelhos representam os diferentes pontos com demandas a ser atendidas e o ponto verde representa a base onde o drone pode recarregar seus suprimentos e a sua bateria.

Assim, os novos inputs do problema sao expressos na Tabela 4.10.

Inputs	
Número de pontos	8
Coordenadas dos pontos	[-10,8], [-10,5], [-5,-10], [4,-2], [-2,4], [6,10], [1,-2], [8,4]
Demandas de cada ponto	$[10, 15, 20, 25, 10, 15, 5, 10]$
Bateria máxima	100
Bateria inicial	100
Peso máximo	100
Velocidade máxima	10
Tempo máximo	10000
Coeficientes do consumo de bateria	$[\alpha, \beta, \gamma] = [0.1, 0.1, 0.1]$
Peso do drone	1

TABELA 4.10 – Parâmetros e configurações iniciais definidos para o estudo do Caso 2, mostrando os valores específicos utilizados como base para a análise e simulação deste cenário.

Esse novo caso foi aplicado as 3 versões do código, os resultados são expressos na Tabela 4.11.

	Versão 1	Versão 2	Versão 3
Distância total	12.80	59.38	199.33
Consumo total de bateria	13039.36	2129.27	626.51
Tempo de missão	130386.60	313.0	77.01
Todas demandas atendidas	NÃO	NÃO	SIM

TABELA 4.11 – Análise comparativa dos resultados obtidos das métricas de desempenho para as 3 versões do código para o primeiro conjunto de pontos, caso 1.

No Caso 2, observou-se o mesmo padrão de melhoria que ocorreu no Caso 1: as versões iniciais do código apresentaram limitações na cobertura e na eficiência das demandas, enquanto a versão 3, com ajustes nas recompensas e penalidades, foi capaz de atender completamente todas as demandas dos pontos de entrega. Essa consistência nos resultados entre os dois casos é um forte indicador de que o código desenvolvido é robusto e funcional, sugerindo que ele pode gerar bons resultados quando aplicado a diferentes cenários. Assim, o algoritmo demonstra potencial para ser uma solução eficiente e adaptável em diversos contextos de otimização logística, comprovando sua aplicabilidade em cenários reais.

Assim como mostrado para o caso 1, a solução final da rota é: (0, 0), (6, 10), (0, 0), (0, 0), (6, 10), (0, 0), (4, -2), (8, 4), (0, 0), (0, 0), (-10, 8), (0, 0), (-2, 4), (1, -2), (0, 0), (-10, 5), (0, 0), (8, 4), (-10, 8), (0, 0), (0, 0), (4, -2), (8, 4), (4, -2), (0, 0), (-10, 5), (0, 0), (4, -2), (8, 4), (0, 0), (-10, 8), (0, 0), (8, 4), (0, 0), (-10, 8), (8, 4), (-2, 4), (0, 0), (8, 4), (0, 0), (0, 0), (0, 0), (0, 0), (8, 4), (0, 0), (0, 0), (0, 0), (-10, 5), (8, 4), (0, 0), (0, 0), (0, 0), (0, 0), (0, 0), (-10, 8), (0, 0), (0, 0), (1, -2), (0, 0), (0, 0), (0, 0), (-5, -10)

E a Figura 4.2 mostra todos os trajetos feitos pela solução final do drone.

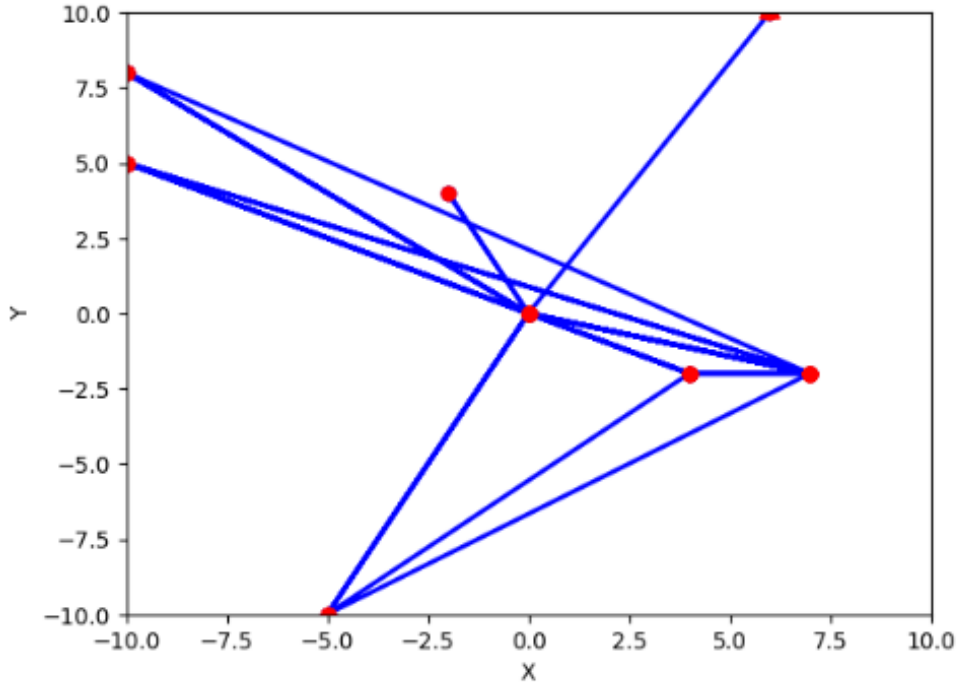


FIGURA 4.4 – Todos os trajetos feitos pela solução final do caso 2, usando a versão 3 do código. Os pontos vermelhos representam os pontos com demandas e a base, e os traços azuis representam os caminhos feitos.

5 Conclusões

5.1 Conclusões

O presente trabalho pretende propor um método para otimização de drones de entrega através de algoritmos de Reinforcement Learning. Foi dada uma maior ênfase na forma como a eficiência pode ser maximizada em casos de utilização que envolvem múltiplos pontos de entrega com diferentes demandas. O desenvolvimento das três versões do modelo evoluiu progressivamente as políticas de ação e recompensa, alcançando um aprimoramento mais significativo nos indicadores de desempenho: distância total percorrida, tempo total de missão, consumo de bateria e atendimento da demanda.

Os resultados mostraram que, ao ajustar as recompensas e penalidades, o drone poderia fazer um trabalho melhor ao priorizar suas decisões e não realizar revisitas desnecessárias, reduzindo a quantidade de retornos à base. Esta versão final do modelo conseguiu atender com eficiência todas as demandas nos dois cenários testados, validando assim a eficiência do algoritmo de Proximal Policy Optimization para resolver problemas de roteamento com muitas restrições.

Embora uma melhoria tenha sido feita, ainda existem certas limitações no modelo. O modelo foi criado apenas para um ambiente bidimensional e não considerou fatores do ambiente operacional, incluindo, entre outros, condições climáticas, obstáculos físicos ou variações na topografia que afetam as rotas reais percorridas pelos drones. Além disso, o modelo deverá fazer uso de técnicas de simulação mais sofisticadas, a fim de replicar com precisão as condições reais de voo dos drones de uma forma muito mais fiel.

5.2 Futuros Trabalhos

Para aprimorar e expandir o trabalho atual, algumas direções podem ser exploradas:

1. Simulação em Ambientes Tridimensionais: Implementar um ambiente tridimensional que considere outras variáveis como altitude e obstáculos físicos, permitindo que o modelo represente melhor as condições de voo reais dos drones.

2. Consideração de Condições Ambientais Dinâmicas: Incorporar variáveis climáticas, como vento e temperatura, que possam afetar a rota e o consumo de energia do drone. Esse aspecto tornaria o modelo mais robusto para operações em ambientes externos.

3. Aprimoramento da Política de Recompensa: Realizar uma busca por hiperparâmetros (por exemplo, através de random search ou grid search) para identificar as combinações de recompensas e penalidades que maximizem a eficiência do drone em diferentes cenários.

4. Benchmark com Outros Algoritmos de Otimização: Comparar o desempenho do PPO com outros algoritmos de Reinforcement Learning, como Deep Q-Networks (DQN) e Advantage Actor-Critic (A2C), bem como com métodos de otimização clássicos para avaliar a eficácia relativa de cada abordagem em resolver problemas de roteamento de drones.

5. Testes em Ambientes Reais: Implementar e testar o modelo em drones reais em ambientes controlados, como campos de teste, para validar a aplicabilidade prática da solução desenvolvida e identificar ajustes necessários para operação em condições reais.

Essas extensões não apenas aumentariam a complexidade e realismo do modelo, mas também potencializariam sua aplicabilidade em cenários industriais e logísticos. O desenvolvimento dessas melhorias futuras contribuiria significativamente para a evolução do uso de drones como uma solução prática e eficiente para problemas de distribuição e logística.

Referências

BELLMAN, R. **Dynamic Programming**. [*S.l.*]: Princeton University Press, 1957.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [*S.l.*]: Springer, 2006.

ESCOLA, B. **Enchentes no Rio Grande do Sul**. 2024. `https:https://brasilescola.uol.com.br/brasil/enchentes-no-rio-grande-do-sul.htm` [Accessed: 12/10/24].

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [*S.l.*]: MIT Press, 2016.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. 2nd. ed. [*S.l.*]: Springer Science & Business Media, 2009.

HAYKIN, S. **Neural Networks and Learning Machines**. 3rd. ed. [*S.l.*]: Prentice Hall, 2009.

KILBY, P.; PROSSER, P.; SHAW, P. Guided local search for the vehicle routing problem. 2002.

KOOL, W.; HOOF, H. van; WELLING, M. Attention, learn to solve routing problems! 2019.

MAIA, P. D. M.; BESSANI, M. AnÁlise do efeito de incertezas de tempo aplicadas ao problema de roteamento de veÍculos com janelas de atendimento. *In*: . **Proceedings** [...]. [*S.l.: s.n.*], 2020.

MORO, M.; WEISE, A.; REIS, C.; FLORES, S. d. Técnicas de pesquisa operacional aplicadas na otimização de rotas de uma rede de lojas de materiais de construção. v. 8, 09 2018.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. [*S.l.*]: The MIT Press, 2012.

OPENCADD. **O que são Redes Neurais? Importância e Como Funciona?** 2024. `https://www.opencadd.com.br/blog/o-que-sao-redes-neurais` [Accessed: 12/10/24].

PUTERMAN, M. L. **Markov Decision Processes: Discrete Stochastic Dynamic Programming**. [*S.l.*]: John Wiley & Sons, 2014.

SANTOS, E. **Alocação de Recursos baseada em Clustering com Aprendizado de Características e Orientação a QoS em Redes LTE-Advanced**. Thesis (Doutorado), 06 2019.

SCHULMAN, J.; WOLSKI, F.; DHARIWAL, P.; RADFORD, A.; KLIMOV, O. Proximal policy optimization algorithms. **arXiv preprint arXiv:1707.06347**, 2017.

SCHULMAN, J.; WOLSKI, F.; DHARIWAL, P.; RADFORD, A.; KLIMOV, O. Proximal policy optimization algorithms. 2017.

SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. 2nd. ed. [*S.l.*]: The MIT Press, 2018.

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO	2. DATA	3. DOCUMENTO Nº	4. Nº DE PÁGINAS
TC	29 de outubro de 2024	DCTA/ITA/TC-053/2024	46

5. TÍTULO E SUBTÍTULO:

Otimização de Rotas de Drones Usando Reinforcement Learning

6. AUTORA(ES):

Isabela Matos Gomes

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica – ITA

8. PALAVRAS-CHAVE SUGERIDAS PELA AUTORA:

Drones, Aprendizado por Reforço, Otimização de Rotas, Logística

9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

Veículos não-tripulado, Aprendizado por reforço, Aprendizagem (inteligência artificial), Rotas, Logística, Engenharia aeroespacial

10. APRESENTAÇÃO: (**X**) **Nacional** ( ) **Internacional**

ITA, São José dos Campos. Curso Engenharia Aeroespacial. Orientador: Prof. Dr. Christopher Shneider Cerqueira. Coorientadora: Leonan Entringer Falqueto.

11. RESUMO:

Este trabalho apresenta uma metodologia para a otimização de rotas de entrega de drones utilizando técnicas de aprendizado por reforço (RL). Motivado por desastres naturais recentes no Brasil, como as inundações no Rio Grande do Sul, o estudo visa desenvolver soluções que permitam a entrega rápida e eficiente de suprimentos em áreas inacessíveis por meios de transporte convencionais. A pesquisa explora algoritmo de Proximal Policy Optimization (PPO) para resolver problemas de roteamento e de otimização. A aplicação dessa técnica visa reduzir o tempo de entrega e aumentar a eficiência dos drones, melhorando a capacidade de resposta em situações de emergência e contribuindo para a gestão global de desastres.

12. GRAU DE SIGILO:

(**X**) **OSTENSIVO** ( ) **RESERVADO** ( ) **SECRETO**