"Today's science fiction is tomorrow's science fact.."

Isaac Asimov

Abstract

The demand for satellite access for efficient data communication coverage is rapidly increasing, driven by the growing need for global connectivity. As a result, the number of satellites launched into low Earth orbit has significantly increased, leading to overpopulation of these orbits. This overpopulation poses a risk to data communication safety, as the presence of flying debris in orbit can threaten the functionality of existing satellites and future launches. To address this issue, we propose to use Machine Learning techniques, specifically clustering and reinforcement learning algorithms, to optimize the coverage of Earth with the current satellites in orbit. We will leverage the publicly available UCS Satellite Database, which provides detailed information on over 5,465 satellites currently orbiting Earth (as of May 1, 2022) including their country of origin, purpose, and other operational details. The Machine Learning algorithms will be implemented using the PyTorch library in our experiments. We aim to evaluate the performance of our proposed method against state-of-the-art baseline methods to demonstrate its effectiveness in optimizing satellite coverage for data communication. Through this research, we anticipate that our approach will contribute to the development of more sustainable and efficient satellite communication systems, mitigating the challenges associated with overpopulation of low Earth orbits and ensuring the safety and reliability of data communication services in the ever-evolving satellite industry.

Keywords: Machine Learning – Clustering – Reinforcement Learning – Satellite Coverage – Optimisation

Contents

Declaration of Authorship **i**

Abstract **iii**

Acknowledgements **iv**

Contents **v**

List of Figures **vii**

List of Tables **viii**

Abbreviations **ix**

1 Introduction **1**

2 Literature Review **3**
2.1 Satellite Coverage 3
2.1.1 Different Kind of Orbits 3
2.1.2 Footprint of a Satellite on Earth 4
2.1.3 Satellite Constellations 6
2.1.3.1 Coverage of a constellation 6
2.1.3.2 Coverage of a multi-layer satellite constellation 6
2.2 Machine Learning, Learning to Optimise 7
2.2.1 Clustering 7
2.2.1.1 K-Means 7
2.2.1.2 DBSCAN 8
2.2.1.3 Fuzzy c-Means 8
2.2.1.4 Limitations 9
2.2.2 Reinforcement Learning 9
2.2.2.1 Deep Q-Networks 10
2.2.2.2 Trust Region Policy Optimisation 11
2.2.2.3 Proximal Policy Optimisation 12
2.2.2.4 Limitations 12
2.3 Related Work 13
2.3.1 Other Optimising Algorithm 13

2.3.1.1 Particle Swarm Opimisation 13
2.3.1.2 Genetic Algorithm 14
2.3.1.3 Limitations 15
2.3.2 Similar Problems 16

3 Requirements Analysis 18
3.1 Features 18
3.2 Evaluation 18

4 Methodology 20
4.1 PyTorch 20
4.2 UCS Dataset 20

5 Professional, Legal, Ethical, and Social issues 22
5.1 Professional issues 22
5.2 Legal issues 22
5.3 Ethical issues 23
5.4 Social issues 23

6 Project Plan 24
6.1 Gantt Chart 24
6.2 Risk Management 25

List of Figures

2.1 Geometry of the satellite coverage area. 5
2.2 Crossover Representation 15
2.3 Mutation Representation 15

6.1 Research Gantt Chart 24
6.2 Dissertation Gantt Chart 25

List of Tables

3.1 Requirements and Prioritisation of Project's Features 18
4.1 Extract of some data of 15 satellites from the database 21
6.1 Risk Assessment 26

Abbreviations

GEO	**G**ostationary **E**arth **O**rbit
LEO	**L**ow **E**arth **O**rbit
MEO	**M**edium **E**arth **O**rbit
SSO	**S**un-**S**ynchronous **O**rbit
DBSCAN	**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise
RL	**R**einforcement **L**earning
DRL	**D**eep **R**einforcement **L**earning
DL	**D**eep **L**earning
DQN	**D**eep **Q**-**N**etworks
TRPO	**T**rust **R**egion **P**olicy **O**ptimisation
PS	**P**olicy **S**earch
KL	**K**ullback-**L**eibler
PPO	**P**roximal **P**olicy **O**ptimisation
PSO	**P**article **S**warm **O**ptimisation
GA	**G**enetic **A**lgorithm
AI	**A**rtificial **I**ntelligence
SVM	**S**upport **V**ector **M**achine

Chapter 1

Introduction

From facilitating necessary communication services to assisting in scientific research and weather forecasting, satellites play a crucial part in our daily lives. As the demand for satellite services increases, it is essential to ensure that there is sufficient satellite coverage over the Earth's surface. Optimising satellite coverage can improve the efficiency and effectiveness of satellite-based services and reduce the costs associated with launching additional satellites as well as helping to limit the number of orbiting debits by limiting the number of space missions.

Machine learning techniques have shown great potential in solving optimization problems in various fields [Yakoubi et al., 2023]. With the satellites now in orbit, our project aims to apply machine learning methods to maximize Earth's satellite coverage. We will investigate how to ensure maximize satellite footprints with the current orbiting satellites by using clustering and reinforcement learning methods.

The main objective of this project is to develop a machine learning-based system that can optimize satellite coverage, taking into account various factors such as the number of satellites, their orbit parameters, and the geographical locations to be covered. The system ought to be flexible enough to adjust to shifting conditions and demands, making it appropriate for use in real-world scenarios.

In this thesis, we will review the existing literature on satellite coverage optimization, satellite footprint, reinforcement learning, and clustering. We will explore the related work in these areas, highlighting the strengths and limitations of each approach. We will then present our proposed machine learning-based approach and evaluate its performance against existing state-of-the-art baseline methods.

We will use the publicly accessible UCS Satellite Database, which provides valuable in-depth information such as orbit data (eccentricity, perigee, etc.), as well as detailed information on each satellite, such as ownership and purpose. This database was last updated on May 1, 2022.

Our aim with this research is to address the challenges arising from the overpopulation of low Earth orbits by developing advanced solutions for satellite communication systems. Our goal is to improve efficiency, effectiveness, and security of data communication services in the fast-growing satellite industry, while mitigating the problems associated with overpopulation of orbits.

Chapter 2

Literature Review

Satellite coverage is a complex topic that involves different orbit types, individual satellite footprints, and constellation footprints. Understanding how satellites are positioned in space and their coverage areas is essential. To better understand how satellite communication services are created and managed, we will examine the fundamentals of satellite coverage in this section, including orbit types, how to determine the area that a satellite covers, and how satellites are organized in constellations.

2.1 Satellite Coverage

Satellite coverage refers to the geographical area that a satellite can serve with its communication or observation capabilities. Understanding satellite coverage, which is determined by several factors, is crucial for designing and optimising satellite communication systems.

2.1.1 Different Kind of Orbits

Satellites can be placed into different orbits depending on their altitude and inclination, which determines the shape and orientation of their trajectory around the Earth.
The Geostationary Earth Orbit (GEO) represents a circle orbit around Earth above the equator from west to east, following the Earth rotation. A satellite on this orbit has an altitude of 35 786 km, a velocity of 3 km per second, making it looks 'stationary' over a fixed position on Earth. Thus, an antenna on Earth can be fixed to always point toward

a satellite.
Low Earth Orbit (LEO), is an orbit relatively close to Earth's surface. The altitude of a satellite on this orbit is less than 1000 km and can be as low as 160 km above Earth's ground. A LEO orbit plane can be tilted, which makes it a very commonly used orbit. Satellites in this orbit travel at a speed of around 7.8 km per second, giving it an orbital period of approximately 90 minutes. Communication satellites in LEO are generally working as part of large constellation to compensate for their rapid displacement.
Medium Earth Orbit (MEO) is comprised between GEO and LEO. It is a similar orbit to LEO in that it doesn't need to take a specific path around Earth.
Polar Orbit is usually traveled from north to south. A satellite on this orbit will approach the poles, even with a deviation of 20 or 30 degrees the orbit is still classified as a polar orbit. The orbit is located at a low altitude between 200 and 1000 km.
Sun-Synchronous Orbit (SSO) is a derivate from polar orbit. Satellites on SSO are synchronous with the Sun, meaning the satellites will always be over the same spot at the same local time.

2.1.2 Footprint of a Satellite on Earth

Now that we have explored the various types of orbits, let's take a closer look at another important aspect of satellite technology: satellite footprints. The satellite's footprint refers to the area on the Earth's surface that is covered by a satellite's signals or transmissions. A satellite on orbit can only cover a defined area on Earth's surface and this area depends of the altitude of the orbit

According to [Gavish and Kalvenes, 1998] the coverage angle of a satellite is 2θ with θ defined as

$$\theta = \arcsin\left(\frac{R_E}{R_E + h}\sin\varphi\right) \tag{2.1}$$

where R_E is the radius of the earth, h is the satellite altitude and φ is $\pi/2$ plus ε (the elevation angle).
By applying fundamental principles of trigonometry, we can modify (2.1)

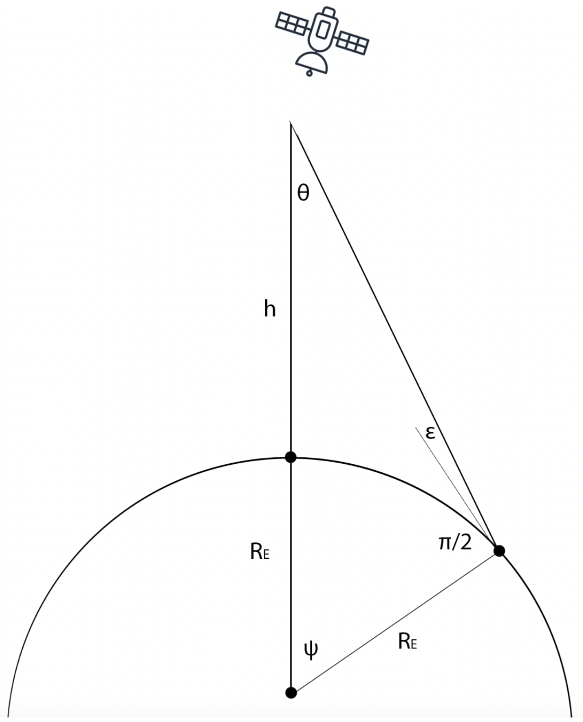

FIGURE 2.1: Geometry of the satellite coverage area.

$$\sin\theta = \frac{R_E}{R_E + h}\cos\epsilon \tag{2.2}$$

Maximal coverage is obtained for $\varepsilon = 0$.

From Figure 2.1 we can deduce geometrically $\theta + \psi + \varepsilon = 90$. Depending on the elevation we want, we can compute the value of θ and finally find ψ. The Earth's surface covered by a satellite on a GEO depends on ψ angle and in [Curtis, 2014] is described as

$$S_{Satellite} = 2\pi R_E^2(1 - \cos\psi) \tag{2.3}$$

We express the previous equation in a percentage of Earth's coverage by dividing S by Earth's surface:

$$Coverage = \frac{S_{Satellite}}{S_{Earth}} \tag{2.4}$$

$$Coverage = \frac{1}{2}(1 - \cos\psi) \tag{2.5}$$

GEO are generally circular orbits which mean they have an eccentricity of 0. At any given moment during its flight, a satellite in elliptical orbit (eccentricity $0 \leq e \leq 1$) is fundamentally similar to a satellite in circular orbit ($e = 0$). Therefore, we can extend equations (2.3) and (2.5) for elliptical orbits ($0 \leq e \leq 1$).

It has to be noted that the speed and altitude of the satellite may vary at different points in the elliptical orbit, whereas these values are constant in a circular orbit. This difference is only relevant when, for example, we try to compute the Earth's surface covered by the satellite during its entire orbit (due to changing altitude).

2.1.3 Satellite Constellations

2.1.3.1 Coverage of a constellation

According to [Dai et al., 2017], by subdividing the target area into several adjacent convex polygons, the constellation coverage problem can be decomposed into a set of single satellite coverage problems.

Those adjacent convex polygons are called convex hull. In geometry, a convex hull is the smallest convex polygon or polyhedron that contains a given set of points in a Euclidean space. A good analogy to construct a convex hull, is to imagine stretching a rubber band around the set of points such that the rubber band conforms to the shape of the set while remaining taut.

In Python, the algorithm *QHull* from the PyHull library provides convex hull computation.

Thus, the problem to solve is finding the angle ψ according to the coverage angle θ of the satellite, with an elevation degree $\varepsilon = 0$. The coverage angle θ of the satellite, with an elevation degree $\varepsilon = 0$ correspond to the maximal coverage radius of the satellite at altitude h. Hence we need to find the minimal altitude enabling the satellite to covers the area. From equation (2.2) we can deduce the minimal altitude.

2.1.3.2 Coverage of a multi-layer satellite constellation

Some interesting facts about satellite constellations include the possibility of using multi-layer constellations for coverage, which is similar to having two separate satellite constellations working for the same network. Thus the covered area can be summed. By utilizing both LEO and GEO constellations, it may be possible to reduce latency in transmitting information [Zhu et al., 2020b]. Instead of passing information from LEO to Ground Station (GS), which results in LEO-GS-LEO communication, a multi-layer constellation could enable LEO-GEO-LEO or GS-GEO-GS communication for transmitting information. However, this paper does not discuss the concept of incorporating additional multi-layer satellite constellations.

2.2 Machine Learning, Learning to Optimise

In this section we will discuss about Machine Learning techniques that are used to solve optimisation problems.

2.2.1 Clustering

Clustering is a widely used technique in machine learning and data analysis, which involves grouping similar data points together into clusters based on their underlying characteristics. Clustering has numerous applications across a range of domains, including image and speech recognition, bio-informatics, and social network analysis. The primary goal of clustering is to identify patterns and structure in large databases that might not be immediately apparent through visual inspection or traditional statistical methods. Let's explore some of these algorithms.

2.2.1.1 K-Means

Given a set of points $(x_1, x_2, \ldots, x_n)$ K-means algorithm will assigns each point to one of the k clusters $C = (C_1, C_2, \ldots, C_k)$. It does so by iteratively finding the minimal distance between a points and a centroid (center of cluster):

$$\arg\min_C \sum_{i=1}^{k} \sum_{x_j \in C} \|x_j - \mu_i\|^2 \tag{2.6}$$

Where μ_i represent the centroid of cluster C_i.

One advantage of K-means is its ability to handle large datasets efficiently [Pauletic et al., 2019]. It has low computational complexity and is known for its computational speed, making it suitable for large datasets [Govender and Sivakumar, 2020]. This is because the algorithm's time complexity is linear with the number of data points, making it scalable for large datasets.

2.2.1.2 DBSCAN

The algorithm DBSCAN is designed to discover clusters through the noise in a spatial database. DBSCAN stands for *Density-Based Spatial Clustering of Applications with Noise*. The fundamental concept of this algorithm is decribed in [Zhu et al., 2020a] as that in order for a data object to be considered part of a cluster, it must have a minimum number of other objects (MinPts) within its given radius (Eps) neighborhood.

DBSCAN is described by two definitions from [Ester et al., 1996]. The first definition (cluster) can be summarized as said above. The second definition describes noise as the points in the dataset which aren't in any of the clusters. According to the same paper DBSCAN is efficient even for large database. Which makes the algorithm noise resistant and able to perform on large dataset.

2.2.1.3 Fuzzy c-Means

To write about Fuzzy c-Means clustering we first need to introduce the notion of soft clustering. Where hard clustering means assigning a point to a cluster or not, in soft clustering each point belongs to each cluster to a certain degree. The degree of membership is represented by a value between 0 and 1, where a value of 1 indicates that the point belongs fully to a cluster, and a value of 0 indicates that the point does not belong to the cluster at all.

As in 2.2.1.1, given a set of points $(x_1, x_2, \ldots, x_n)$ the algorithm will assigns each point to one of the c fuzzy clusters $C = (c_1, c_2, \ldots, c_c)$. From [Bezdek, 1981] this clustering algorithm minimize the following objective function:

$$J_m(W, C) = \sum_{k=1}^{n} \sum_{i=1}^{c} w_{k,i}^m \|x_k - c_i\|^2 \tag{2.7}$$

Where $W = w_{k,i} \in [0, 1]$ represent the degree to which element, x_k, belongs to cluster c_j and m is a weighting component $m \in [1, \infty)$.

Fuzzy c-means offers advantages such as cluster assignment flexibility and the ability to provide a more realistic representation of the probability of belonging to a cluster [Govender and Sivakumar, 2020].

2.2.1.4 Limitations

K-means clustering has several limitations, including the need to specify the number of clusters in advance, sensitivity to outliers, and an inability to handle non-convex clusters of varying size and density [Govender and Sivakumar, 2020]. Additionally, the algorithm is sensitive to the scale of the data set and different initial centroid selections may produce different clustering results. These limitations can affect the accuracy and generalizability of the algorithm in certain data sets and applications.

DBSCAN has limitations related to parameter sensitivity and scalability. Results can be impacted by the values of its parameters (Eps and MinPts). Additionally, DBSCAN may face challenges with scalability when dealing with large datasets, as the computational cost of calculations may become prohibitive [Mungekar et al., 2021].

Fuzzy c-Means clustering also has its own set of disadvantages, including its high complexity, which can lead to slow convergence and difficulties in interpreting the results [Govender and Sivakumar, 2020]. In addition, the number of clusters needs to be specified in advance, similar to K-means, and there is a risk of converging to local optima. These limitations should be taken into consideration when choosing a clustering algorithm for a specific task.

2.2.2 Reinforcement Learning

Reinforcement learning (RL) is a subfield of machine learning that has recently gained much attention due to its potential to enable autonomous agents to learn and improve decision-making abilities in dynamic environments. RL algorithms learn through interaction with an environment, receiving feedback in the form of rewards or penalties for their actions. This learning process enables the agent to develop policies that maximize the cumulative rewards over time, leading to optimal decision-making. In this literature review, we explore the latest advancements in RL algorithms, their strengths and limitations, and their potential applications in various domains.

Deep Reinforcement Learning (DRL) – a subfield of RL – has been one of the most important directions in the area of artificial intelligence. It integrates the understanding of Deep Learning (DL) with the decision-making capability of RL and directly regulates the actions of agents by high-dimensional perceptual input learning [Rajendran and Geetha, 2021].

2.2.2.1 Deep Q-Networks

Deep Q-Networks (DQN) combines RL and DL. The DL part is utilized to learn and estimate the maximum discounted future reward associated with taking specific actions in response to particular states [Li et al., 2016]. This allows the DQN to implicitly learn the optimal actions to take in different situations based on the expected rewards.

The DL is made of artificial neurons organized in interconnected layers. The network learns by adjusting the weights of connections between neurons. A model of the update of a neuron can be written as:

$$Y(X) = f(WX + b) \tag{2.8}$$

With, X: Input Vector, W: Weight Matrix, b Bias Vector, f: Activation Function, Y: Output Vector.

The weight and bias parameters are learned during training to optimise the network's performance on a specific task. A typical ANN consists of an input layer which receipt the data, one or more hidden layers, and an output layer.

A DQN is an critic-only methodology dealing with only a Q-function for solving a RL problem [Lee et al., 2020]. The use of a neural network in the DQN algorithm allows for greater flexibility in defining the state, eliminating the need for specialized policies like epsilon-greedy to achieve convergence, as required in Q-learning [Sutisna et al., 2022].

To update the Q-value based on the result of the agent we need the Bellman equation which is described in [Sutisna et al., 2022]:

$$Q_{new}(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_a Q(s_t + 1, a_t)) \tag{2.9}$$

Several parameters are used:

s_t: Current State,

a_t: Action taken by the agent on the current state,

α: Learning Rate, represents how fast the information is updated,

γ: Discount Factor, determines the importance of future rewards,

r_t: Current Reward value.

2.2.2.2 Trust Region Policy Optimisation

The Trust Region Policy Optimisation (TRPO) algorithm is a widely used and effective method for policy search (PS) [Schulman et al., 2015].

TRPO's aim is to enhance policy gradually while maintaining stability in the learning process. From [Zhang et al., 2019]:

$$\begin{aligned} &\underset{\theta}{\text{maximize}} \quad \mathbb{E}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A(s,a)\right] \\ &\text{subject to} \quad \mathbb{E}\left[\mathbf{KL}\left[\pi_{\theta_{old}}(\cdot|s),\ \pi_\theta(\cdot, s)\right]\right] \leq \delta \end{aligned} \tag{2.10}$$

With: $\pi_{\theta_{old}}$: Old Policy, π_θ: Policy to Optimise, δ: Bound for the **KL** divergence, A: Advantage Function, a: Action from the Action Set, s: State from the State Set.

The maximum deviation of the new policy is limited from the previous one, as defined by the δ bound which restricts the Kullback-Leibler (KL) divergence between the two policies, thus ensuring that the optimisation process is stable and does not lead to significant policy changes. The advantage function A is used by the algorithm to figure out which actions are better in a given situation. It helps TRPO learn which actions are more likely to lead to good outcomes.

2.2.2.3 Proximal Policy Optimisation

Proximal Policy Optimisation (PPO) is a reinforcement learning algorithm that uses a clipping mechanism to simplify the optimisation of the policy network. It can be treated as a simpler approximation of TRPO. This clipping mechanism constrains the size of policy updates to a certain range, which makes the optimisation more stable and effective. By using this clipping mechanism, PPO can use first-order optimisation methods to simplify the optimisation process. This makes it more efficient and easier to implement compared to TRPO [Chen et al., 2018].
According to [Tuan et al., 2018], PPO uses the following clipped objective to heuristically constrain the KL-divergence:

$$\max_{\theta} \mathbb{E}\left[\min\left(\rho A(s,a),\ clip(\rho, 1-\varepsilon, 1+\varepsilon)A(s,a)\right)\right] \tag{2.11}$$

With $\rho = \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}$ and ε an hyper-parameter.

The *clip* function takes ρ and limits it to be within the range (1-ε) and (1+ε). ε represents a small margin of tolerance, which is added or subtracted from the value 1 (i.e. if $\varepsilon = 0.1$ then $0.9 \leq \rho \leq 1.1$). Each element of A is limited to be within the specified range (1-ε) and (1+ε), with any values outside that range being set to the closest limit.

2.2.2.4 Limitations

The limitations of the DQN algorithm include that it is inefficient in high-dimensional environments because it requires a large amount of training data, which can be computationally expensive to gather [Lillicrap et al., 2016]. Additionally, it is not suitable for continuous action spaces and struggles with partially observable environments where it assumes the agent has complete information about the environment. These limitations can prevent the algorithm from learning the optimal policy or slow down its convergence.

The computational inefficiency and scalability challenges of TRPO arise from the complex second-order optimisation involved, which can pose difficulties when extending the algorithm to large-scale problems or complex network architectures [Wang et al., 2019].

There is a need for further study on the optimisation behavior of PPO, despite its success. The proximal property of PPO, which governs how the likelihood ratio is constrained, has been a topic of concern among some researchers. More research is required to better understand and analyze the optimisation behavior of PPO in order to address these concerns [Wang et al., 2019].

2.3 Related Work

2.3.1 Other Optimising Algorithm

In the field of artificial intelligence, optimisation algorithms play a critical role in solving complex problems. Among these algorithms, Particle Swarm Optimisation (PSO) and Genetic Algorithms (GA) are two widely used and well-studied techniques.

PSO is a population-based optimisation algorithm that simulates the social behavior of swarms or flocks. The algorithm consists of a set of particles, each representing a candidate solution, which move in a high-dimensional search space to find the optimal solution.

GA is a class of evolutionary algorithms that are based on the principles of natural selection and genetics. This mimics the process of natural selection, where the fittest individuals are more likely to survive and reproduce.

2.3.1.1 Particle Swarm Opimisation

PSO is a heuristic search algorithm due to its utilization of a population of particles that move randomly in the search space to find the optimal solution. This algorithm was introduced by [Kennedy and Eberhart, 1995].

From the authors of PSO, the variables pbest (personal best) and gbest (global best), along with their increments, are both essential in optimisation algorithms. The pbest variable is similar to memory where each individual remember its own experience and best fitness and adjust its velocity accordingly. In contrast, individuals strive to attain gbest as it represents the global best fitness (how good a solution is) point of the swarm.

PSO is a stochastic algorithm which allow it to efficiently explore complex and high-dimensional search spaces, where traditional deterministic optimisation algorithms might get stuck in local optima. PSO is a scalable optimisation algorithm that can be easily applied to systems with any number of units [Rajashree and Upadhyay, 2016].

The simplified version of this algorithm obtained by [Kennedy and Eberhart, 1995] multiplies the stochastic factors "by 2 to give it a mean of 1". According to their results, this version outperform their previous version. The velocities are adjusted by the following formula:

$$\vec{V_i} = \vec{V_i} + 2rand() \times \left(\overrightarrow{pbest_i} - \overrightarrow{present_i}\right) + 2rand() \times \left(\overrightarrow{pbest_g} - \overrightarrow{present_i}\right) \tag{2.12}$$

Where: V_i is the velocity vector of particle i, $pbest_i$ is the personal best vector of particle i, $present_i$ is the position vector of particle i, $pbest_g$ is the personal best vector of the particle with the best fitness (thus is gbest).

The position of each particles is then updated:

$$present_i = present_i + V_i \tag{2.13}$$

2.3.1.2 Genetic Algorithm

GA starts with a random population of chromosomes and evaluates their fitness. The chromosomes with the highest fitness have the highest chances of reproduction. The algorithm evolves iteratively the populations through generations toward better solutions by applying genetic operators such as crossover and mutation [Whitley, 1998].

Each generation a breeding process takes place it consist of 3 steps, 1) select parents, 2) apply crossover and mutation to create new chromosomes from the parents and 3) replace the old individuals by the new ones. From [Sivanandam and Deepa, 2008] there are different selection methods and usually they are based on the fitness of the chromosomes. According to them, selection happens crossover which consists of selecting crossover site(s), and swapping genetic material between the parents at the selected site(s) (there are several techniques such as Single Point Crossover or N-Point crossover). Crossover probability is a crucial parameter as it set the chances of creating new combinations of genetic material in the chromosomes. Figure 2.2 can help to understand the crossover process.

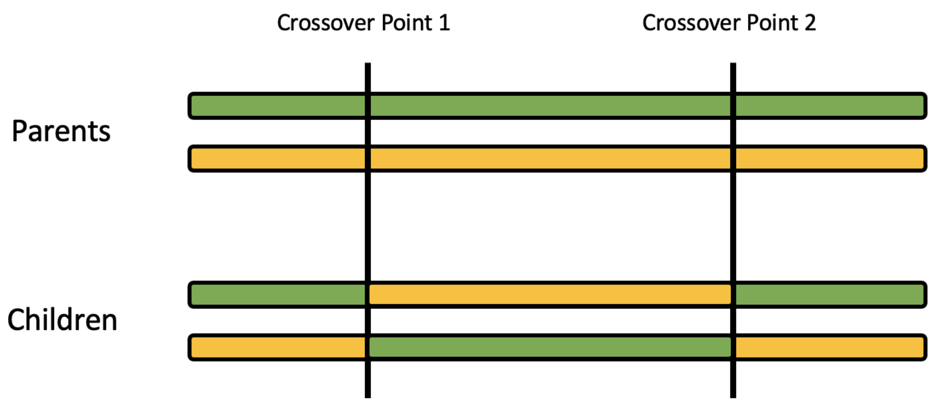

FIGURE 2.2: Crossover Representation

Mutation randomly modifies the new chromosomes, helping to explore the entire search space, avoid convergence to local optima and maintaining genetic diversity. The probability of mutation is usually set to $1/L$, where L is the length of the chromosome [Sivanandam and Deepa, 2008]. Figure 2.3 can help to understand the mutation process.

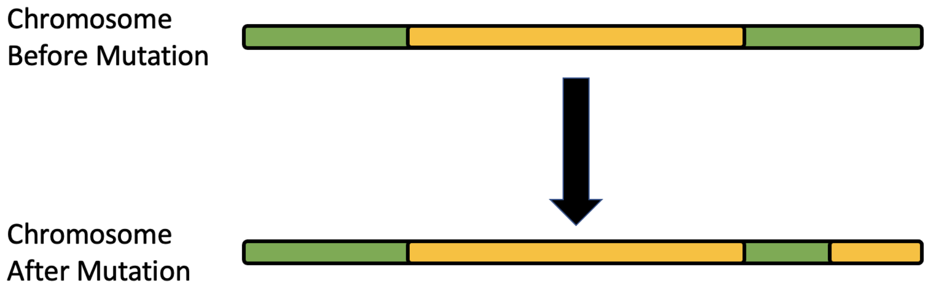

FIGURE 2.3: Mutation Representation

2.3.1.3 Limitations

PSO is sensitive to the choice of its parameters, and improper parameter values may lead to poor performance [Obayyanahatti and Shi, 1998]. PSO may have limited exploration capability, which can cause the algorithm to miss potentially optimal solutions in the search space.

Choosing parameters in GA like population size, mutation and crossover rates, selection method and its strength requires coupling with a local search technique. However, GAs may have trouble finding the exact global optimum [Sivanandam and Deepa, 2008].

2.3.2 Similar Problems

In their study [Peng and Bai, 2019], H. Peng, and X. Bai propose a machine learning-based approach for predicting satellite coverage in order to optimize satellite orbits and improve coverage efficiency. The authors use a Support Vector Machine (SVM) based on a range of input parameters, including the satellite's position, velocity, and orientation. The results of their experiments show that the proposed approach significantly improves the accuracy of orbit prediction for various components and is effective for most position and velocity components in both SSO and LEO.
Overall, this study highlights the potential of machine learning techniques for improving the performance of satellite systems and enhance the current physical-based prediction frameworks. However, the authors note that further research is needed to refine the proposed approach and address some of the limitations and challenges associated with satellite orbit prediction, such as atmospheric interference and changing environmental conditions.

The paper [Kopacz et al., 2021] presents a new approach to satellite constellation management and replacement using PPO2, a DRL algorithm, on a custom spacecraft build and loss model. The authors created a custom environment to simulate spacecraft behavior, revenue generation, and decay. The reinforcement learning agent successfully learned an optimal policy for two models: a Simplified Model where financial cost is ignored, and an Advanced Model where financial cost is considered.
The study found that the PPO2 algorithm was able to converge to an optimal solution for the Simplified Model after about 200,000 simulations, while the Advanced Model, which included financial impacts and satellite health, took approximately 25,000,000 simulations to converge to an optimal policy. The Advanced Model demonstrated the first step in applying AI for the engineering application of constellation management and autonomous replacement, as the policy not only learned to deploy a satellite constellation but also evaluated the health of all satellites to maximize reward by maintaining the constellation and generating revenue.

Overall, this study shows promising initial research developments towards a real-world tool and an AI application that can aid various Aerospace businesses in managing LEO constellations. The use of DRL algorithms for satellite constellation management and replacement has the potential to become imperative for deploying and maintaining small satellite mega-constellations.

Chapter 3

Requirements Analysis

3.1 Features

N°	Feature Description	MoSCoW	Priority
1	Implement a machine learning model using PyTorch to optimize satellite coverage	M	H
2	Use Reinforcement Learning and/or Clustering algorithms to solve the problem	M	H
3	Evaluate the effectiveness of the solution against state-of-the-art baseline models	M	H
4	Provide better non collision orbit	S	H
5	Implement a user-friendly interface to visualize the satellite coverage optimization results	S	M
6	Improve the accuracy of the model by incorporating additional features or data sources	C	L
7	Implement a real-time satellite coverage optimization system	C	L
8	Explore the feasibility of scaling up the solution to larger datasets and more complex optimization problems	W	L
9	Explore the use of techniques or algorithms that are beyond the scope of the project or not feasible given the resources available	W	L

Table 3.1: Requirements and Prioritisation of Project's Features

3.2 Evaluation

The effectiveness of the thesis project will be evaluated using several criteria, including accuracy, efficiency, fuel usage (if possible to determine), and repeatability. Accuracy will be assessed by comparing the predicted satellite coverage generated by the machine learning model with actual satellite coverage data, quantifying the percentage increase in coverage compared to state-of-the-art baseline or existing methods. Efficiency will be measured by evaluating the computational time taken and the number of iterations required to achieve optimal satellite coverage. The impact of the model on fuel consumption will be assessed if feasible to determine. Lastly, repeatability will be evaluated by documenting the settings, parameters, and techniques used in the model to ensure that others can replicate the experiments and obtain similar results. These criteria will provide a comprehensive evaluation of the performance and viability of the proposed solution in optimizing Earth's satellite coverage with current satellites on orbit.

Chapter 4

Methodology

4.1 PyTorch

We will implement our clustering and reinforcement learning algorithms using PyTorch, a popular open-source machine learning framework. PyTorch provides a flexible and intuitive platform for building and training complex models, making it well-suited for our task of optimizing Earth's satellite coverage.

4.2 UCS Dataset

For our analysis, we used the publicly available UCS Satellite Database, which contains information about the current satellites on orbit around Earth. The database includes details such as the satellite's name, owner, purpose, launch date, and orbit parameters, as well as the satellite's footprint, which describes the area of the Earth's surface that the satellite can "see" and transmit signals to.

The dataset consists of over 5,000 satellite records, each of which contains a range of attributes and measurements. We will preprocess the dataset by filtering out any records with missing or invalid data, and converted it into a format suitable for analysis using Python's Pandas library.

ADD TABLE WITH 15 LINES OF EXAMPLES FROM THE DATASET

Name of Satellite	**Users**	**Purpose**	**Class of Orbit**	**Perigee (km)**	**Eccentricity**
Aalto-1	Civil	Technology Development	LEO	497	$1,45 \times 10^{-03}$
AAUSat-4	Civil	Earth Observation	LEO	442	$1,77 \times 10^{-02}$
ABS-2	Commercial	Communications	GEO	35 778	$1,78 \times 10^{-04}$
ABS-2A	Commercial	Communications	GEO	35 700	$0,00 \times 10^{+00}$
ABS-3A	Commercial	Communications	GEO	35 788	$1,78 \times 10^{-04}$
ABS-4	Commercial	Communications	GEO	35 780	$1,54 \times 10^{-04}$
ABS-6	Commercial	Communications	GEO	35 777	$2,02 \times 10^{-04}$
Adelis-Sampson 1	Commercial	Technology Development	LEO	538	$1,66 \times 10^{-03}$
Adelis-Sampson 2	Government	Technology Development	LEO	539	$4,32 \times 10^{-03}$
Adelis-Sampson 3	Government	Technology Development	LEO	537	$1,73 \times 10^{-03}$
Advanced Orion 10	Government	Earth Observation	GEO	35 700	$1,19 \times 10^{-03}$
Advanced Orion 4	Military	Earth Observation	GEO	35 560	$5,37 \times 10^{-03}$
Advanced Orion 5	Military	Earth Observation	GEO	35 589	$4,68 \times 10^{-03}$
Advanced Orion 6	Military	Earth Observation	GEO	35 714	$2,64 \times 10^{-03}$
Advanced Orion 7	Military	Earth Observation	GEO	35 500	$0,00 \times 10^{+00}$

TABLE 4.1: Extract of some data of 15 satellites from the database

Chapter 5

Professional, Legal, Ethical, and Social issues

5.1 Professional issues

This research will follow professional standards and practices in the field of study. All resources will be referenced and used in compliance with their respective licenses. The produced software will be tested and documented according to professional software engineering practices. TThe software developed for this research project will be released under the MIT License, which is a permissive open source license that allows for free use, modification, distribution, and sharing of the software. This license provides a framework that promotes collaboration and encourages further research and development by the community. By adopting the MIT License, I aim to make the software widely accessible and encourage its use and contribution by other researchers, while also providing a disclaimer of liability., allowing any user to modify, share, and distribute the code under the same license.

5.2 Legal issues

This research will comply with all relevant laws and regulations. Data used is obtained from public sources or with the appropriate permissions, and will comply with applicable data protection laws and policies. Any copyrighted materials will be used with the necessary permissions and licenses.

5.3 Ethical issues

This is a research project that does not involve any sensitive data or human subjects. Therefore, there is no risk of violating any ethical guidelines or codes of conduct.

5.4 Social issues

This research aims to improve Earth's satellite coverage. It will contribute to the advancement of technology and knowledge in this area, potentially benefiting society as a whole.

Chapter 6

Project Plan

6.1 Gantt Chart

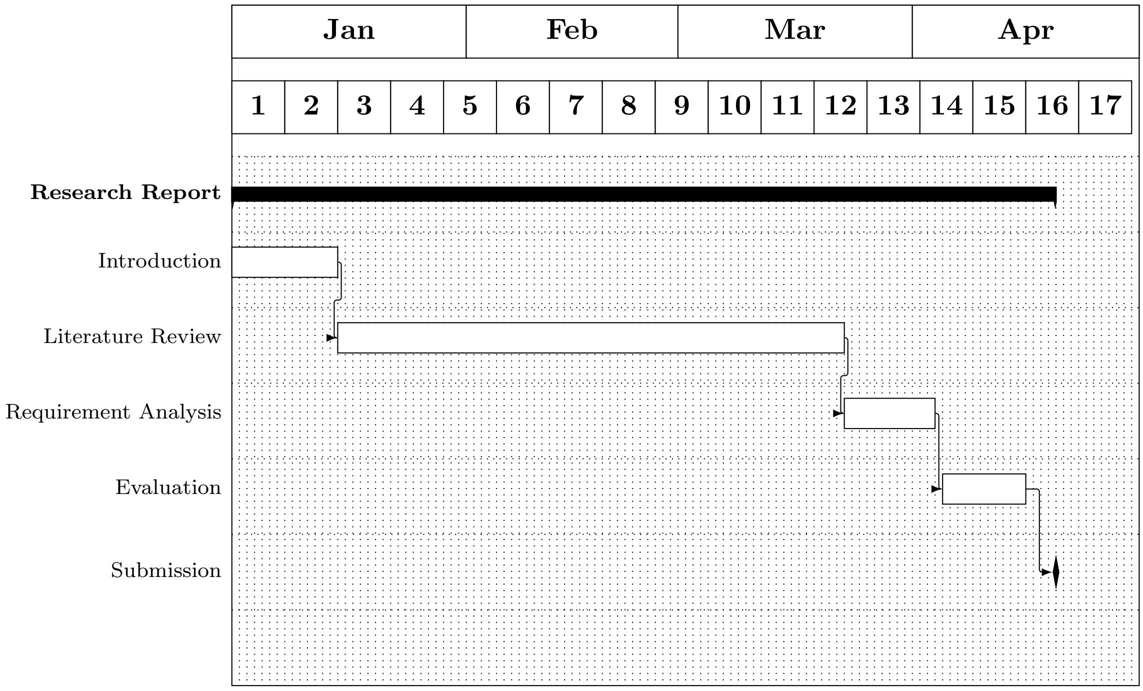

FIGURE 6.1: Research Gantt Chart

Figure 6.1 and 6.2 show the project plan for this project. Research report takes place from January to April. After writing this first part, the project implementation will start in the beginning of May. The implementation will consist of two prototypes, one using clustering algorithm and the second using reinforcement learning. Both prototypes will be tuned and evaluated and compared against state-of-the-art baseline method. Dissertation writing will take place simultaneously with evaluation and the poster will be done in august.

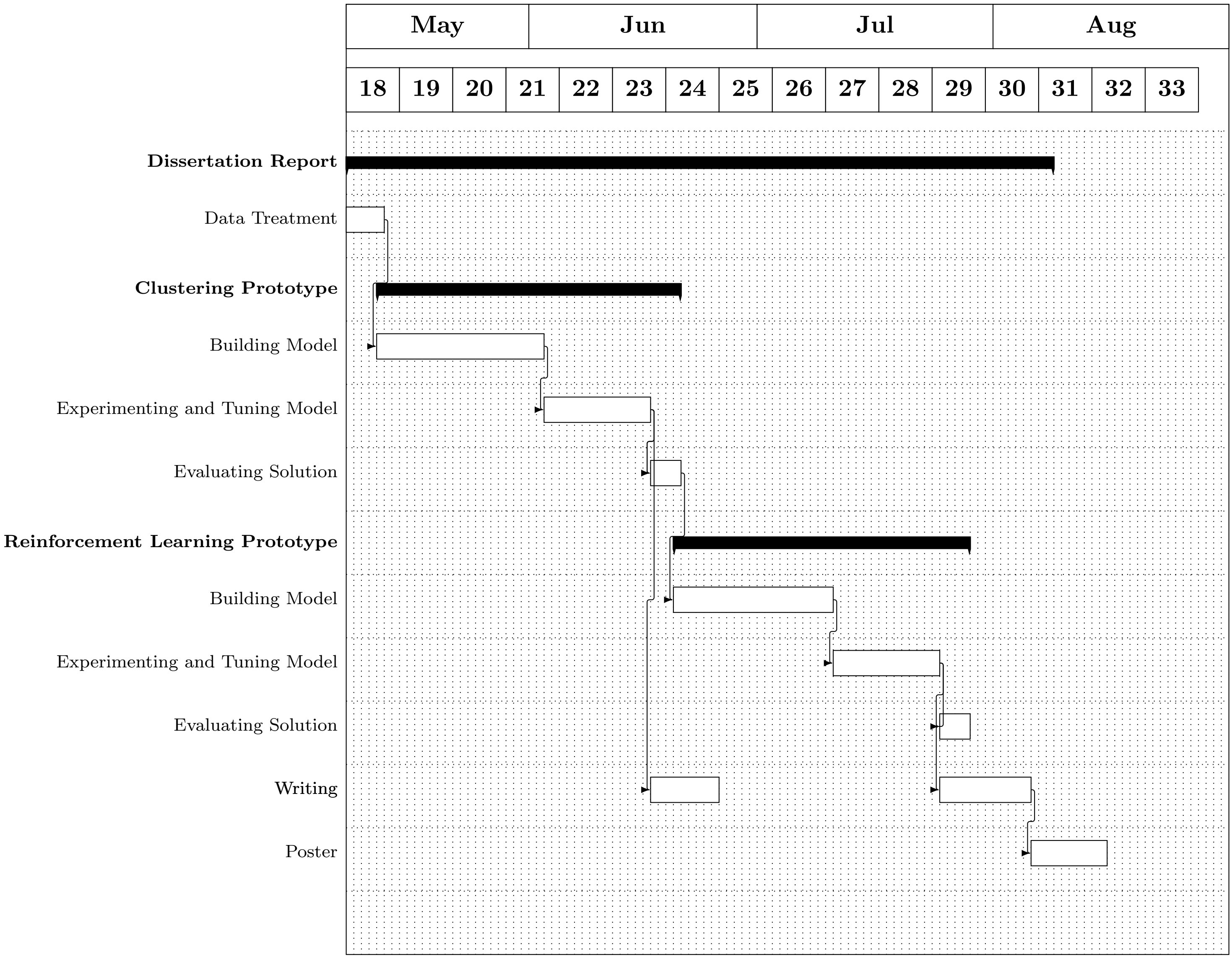

FIGURE 6.2: Dissertation Gantt Chart

6.2 Risk Management

Risk assessment's role is to identify, analyze, and evaluate potential risks and their impacts on a particular project. Inadequate risk management can lead to increased likelihood of risks occurring, higher impact of risks or reduced project success. A proactive approach to risk management involves identifying and addressing potential uncertainty issues before they materialize, by anticipating and mitigating risks in advance to minimize their impacts. Established standards [1], provides systematic and structured approach to risk management such as:

- Identify all potential risks that could affect the project

[1] Such as ISO 31000:2018 Risk Management - Guidelines

- Analyze each identified risk in terms of its likelihood of occurrence, potential consequences, and severity of impact.
- Prioritize risks based on their severity and impact.
- Develop strategies to mitigate or manage each identified risk.
- Continuously monitor and review risks

The risks associated with this project are outlined in Table 6.1.

Risk	**Likelihood of Occurrence**	**Impact**	**Avoidance Strategy**
Technical challenge	M	H	Use simpler algorithms and talk to supervisor/professor
Difficulties to get good results	M	H	Use more complex tuning algorithms
Time constraints	M	M	Focus on one prototype and delay the other one for future studies
Feedback delays	M	M	Ask for online meetings and/or find a professor to give a feedback
Insufficient dataset	L	H	Find other complementary datasets
Student is ill	L	L	Work remotely

TABLE 6.1: Risk Assessment

Bibliography

Bezdek, J. C. (1981). *Objective Function Clustering*, pages 43–93. Springer US, Boston, MA.

Chen, G., Peng, Y., and Zhang, M. (2018). An adaptive clipping approach for proximal policy optimization. *CoRR*, abs/1804.06461.

Curtis, H. D. (2014). Chapter 2 - the two-body problem. In Curtis, H. D., editor, *Orbital Mechanics for Engineering Students (Third Edition)*, pages 59–144. Butterworth-Heinemann, Boston, third edition edition.

Dai, G., Chen, X., Wang, M., Fernández, E., Nguyen, T. N., and Reinelt, G. (2017). Analysis of satellite constellations for the continuous coverage of ground regions. *Journal of Spacecraft and Rockets*, 54(6):1294–1303.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

Gavish, B. and Kalvenes, J. (1998). The impact of satellite altitude on the performance of leos based communication systems. *Wireless Networks*, 4:199–213.

Govender, P. and Sivakumar, V. (2020). Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). *Atmospheric Pollution Research*, 11(1):40–56.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4.

Kopacz, J., Roney, J., and Herschitz, R. (2021). Deep replacement: Reinforcement learning based constellation management and autonomous replacement. *Engineering Applications of Artificial Intelligence*, 104:104316.

Lee, J., Chung, J., and Sohn, K. (2020). Reinforcement learning for joint control of traffic signals in a transportation network. *IEEE Transactions on Vehicular Technology*, 69(2):1375–1387.

Li, L., Lv, Y., and Wang, F.-Y. (2016). Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica*, 3(3):247–254.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. *ICLR*, abs/1509.02971.

Mungekar, D., Joshi, H., Kankekar, A., Nair, P., and Das, P. (2021). Crime analysis using dbscan algorithm. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 628–635.

Obayyanahatti, B. and Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization. volume 1447, pages 611–616.

Pauletic, I., Prskalo, L. N., and Bakaric, M. B. (2019). An overview of clustering models with an application to document clustering. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1659–1664.

Peng, H. and Bai, X. (2019). Machine learning approach to improve satellite orbit prediction accuracy using publicly available data. *The Journal of the Astronautical Sciences*, 67.

Rajashree, B. and Upadhyay, P. (2016). Pso approach for eld problem: A review. In *2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, pages 225–228.

Rajendran, P. and Geetha, A. (2021). Optimization of hospital bed occupancy in hospitals using double deep q network (ddqn). In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pages 1029–1033.

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and Abbeel, P. (2015). Trust region policy optimization. *CoRR*, abs/1502.05477.

Sivanandam, S. and Deepa, S. (2008). *Terminologies and Operators of GA*, pages 39–81. Springer Berlin Heidelberg, Berlin, Heidelberg.

Sutisna, N., Ilmy, A. M. R., Arifuzzaki, Z., Syafalni, I., Maulana, D., Mulyawan, R., and Adiono, T. (2022). Deep q-network model for intelligent traffic light. In *2022 International Symposium on Electronics and Smart Devices (ISESD)*, pages 1–5.

Tuan, Y., Zhang, J., Li, Y., and Lee, H. (2018). Proximal policy optimization and its dynamic version for sequence generation. *CoRR*, abs/1808.07982.

Wang, Y., He, H., and Tan, X. (2019). Truly proximal policy optimization. *CoRR*, abs/1903.07940.

Whitley, D. (1998). A genetic algorithm tutorial. *Statistics and Computing*, 4.

Yakoubi, S., Kobayashi, I., Uemura, K., Nakajima, M., Hiroko, I., and Neves, M. A. (2023). Recent advances in delivery systems optimization using machine learning approaches. *Chemical Engineering and Processing - Process Intensification*, 188:109352.

Zhang, Z., Luo, X., Xie, S., Wang, J., Wang, W., and Li, Y. (2019). Proximal policy optimization with mixed distributed training. *CoRR*, abs/1907.06479.

Zhu, Q., Tang, X., and Liu, Z. (2020a). Revised dbscan clustering algorithm based on dual grid. In *2020 Chinese Control And Decision Conference (CCDC)*, pages 3461–3466.

Zhu, X., Jiang, C., Kuang, L., Dong, M., and Zhao, Z. (2020b). Capacity analysis of multi-layer satellite networks. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 251–256.