

**Machine Learning Based Route Optimization for The Travelling
Salesman Problem with Pickup and Delivery**

ONG ZHI YING

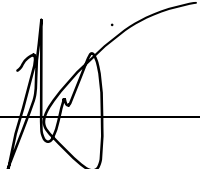
**A project report submitted in partial fulfilment of the
requirements for the award of Bachelor of Science
(Honours) Software Engineering**

**Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman**

May 2023

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 
Name : ONG ZHI YING
ID No. : 011219-14-0408
Date : 23/4/2023

APPROVAL FOR SUBMISSION

I certify that this project report entitled “**TITLE TO BE THE SAME AS FRONT COVER, CAPITAL LETTER, BOLD**” was prepared by **STUDENT’S NAME** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of XXX (Honours) XXXXXXXX at Universiti Tunku Abdul Rahman.

Approved by,

Signature

:



Supervisor

:

Ir Ts Dr Tham Mau Luen

Date

:

15/5/2023

Signature

:

keckhor

Co-Supervisor

:

Dr Khor Kok Chin

Date

:

15/5/2023

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2023, ONGZHIYING. All right reserved.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to everyone who contributed to this project's successful completion. Firstly, I would like to thank my supervisor Ir Ts Dr Tham Mau Luen, for providing me with invaluable insights and support throughout this journey. I am also deeply grateful to my co-supervisor, Dr Khor Kok Chin, for assisting me in furnishing my report with his expertise and resource. Without them, I would not be able to complete this work. Furthermore, I am also thankful to my family and friends for their unwavering encouragement. Their words of motivation and positive energy kept me going during the challenging times. Once again, I am truly honoured to have them throughout this project.

ABSTRACT

The booming of online consumers has resulted in the strong demand of e-commerce postal delivery services. To gain the competitive advantages among other couriers, most couriers try their best to offer their customers effective pickup and delivery services with short delivery time. The customers refer to retailers or purchasers or both. In this context, the travelling salesman problem can be applied. This project aims to achieve the shortest Estimated Time of Arrival (ETA) that allows couriers to collect goods from every customer's location exactly once and returns to the original travelling point. However, there is a high possibility for a courier to visit a customer's location multiple times if the customer happens to be the seller and the buyer at the same time. Consequently, the courier cost will increase, which leads to in low customer satisfaction due to long pickup and delivery time, particularly during peak hours. The goal of this project is to deliver an optimal route for pickup and delivery using Reinforcement Learning (RL) and Genetic Algorithm (GA). 16 locations in Klang Valley are chosen randomly and later ETAs between them are retrieved for testing purpose. It is found that GA is better than RL in finding optimal route.

TABLE OF CONTENTS

Table of Contents

CHAPTER 1

INTRODUCTION	1
1.1 General Introduction	1
1.2 Background of The Problem	1
1.3 Problem Statement	2
1.3.1 Lack of Pickup and Delivery Route Optimisation	3
1.3.2 Inefficiency in Handling Pickup and Delivery Request	4
1.4 Aim and Objectives	4
1.5 Proposed Solution	5
1.6 Proposed Approach	6
1.7 Project Scope	8

CHAPTER 2

LITERATURE REVIEW	9
2.1 Technique for Route Optimisation	9
2.2 Solving Optimisation Problem Using Reinforcement Learning	11
2.3 Solving Optimisation Problem Using Genetic Algorithm	13
2.4 Google Maps API	14
2.5 Performance Evaluation Based On The Shortest Estimated Time of Arrival (ETA)	15

CHAPTER 3

METHODOLOGY AND WORK PLAN	17
3.1 Introduction	17
3.2 Research Methodology	17
3.2.1 Goal Understanding	18
3.2.2 Data Preparation	18

	8
3.2.3 Data Pre-processing	19
3.2.4 Data Transformation	19
3.2.5 Machine Learning	19
3.2.6 Evaluation	34
3.2.7 Consolidation	34
3.3 Research Tool and Technology Used	34
3.3.1 Visual Studio Code	34
3.3.2 Tensorflow	35
3.3.3 Google Colab	35
3.3.4 GitHub	35
3.3.5 Google Maps-Distance Matrix API	35
3.4 Work Plan	36
3.4.1 Work Breakdown Structure	36
3.5 Gantt Chart	38
3.5.1 Gantt Chart for FYP1	38
3.5.2 Gantt Chart for FYP2	39
CHAPTER 4	
RESULTS AND DISCUSSION	40
4.1 Datasets Used	40
4.2 Evaluation Criteria	41
4.3 Experiments	41
4.3.1. Experiment on RL	41
4.3.2. Experiment on GA	43
4.4 Compare RL and GA Results	44
4.5 Discussion	46
CHAPTER 5	
CONCLUSION AND RECOMMENDATIONS	50
REFERENCES	52

LIST OF TABLES

Table 3.1:	Comparison of Reinforcement Learning Algorithms	27
Table 4.1:	Experiment Results of RL and GA	45
Table 4.2:	Comparison between RL and GA	48

LIST OF FIGURES

Figure 1.1	Workflow of Proposed Solution	6
Figure 1.2	Overview of KDD Process for Data Mining	7
Figure 3.1	Overview of KDD Process for RL and GA	17
Figure 3.2	Sample Response from Distance Matrix API	18
Figure 3.3	16 Node's Location in Klang Valley for Training Dataset	20
Figure 3.4	ETA Matrix for Training Dataset	20
Figure 3.5	16 Node's Location in Klang Valley for Testing Dataset	20
Figure 3.6	ETA Matrix for Testing Dataset	21
Figure 3.7	RL Model	22
Figure 3.8	Flowchart of Implemented RL	23
Figure 3.9	Observation Space	25
Figure 3.10	Initial State	25
Figure 3.11	Node 1 is Visited	25
Figure 3.12	Comparison between Q-learning and Deep Q-learning	27
Figure 3.13	Flowchart of Implemented GA	30
Figure 3.14	Pseudocode of A Complete GA Loop	33
Figure 4.1	Initial Route without Route Optimization Technique	40
Figure 4.2	Learning Curve of DQN	42
Figure 4.3	A Graph of Fitness Function (Best Fitness) Convergence of GA	44
Figure 4.4	Optimal Route Provided by RL	45
Figure 4.5	Optimal Route Provided by GA	46

LIST OF SYMBOLS / ABBREVIATIONS

AI	Artificial Intelligence
ANOVA	Analysis of Variance
API	Application Programming Interface
CNN	Convolutional Neural Network
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
ETA	Estimated Time of Arrival
GA	Genetic Algorithm
JSON	JavaScript Object Notation
KDD	Knowledge Discovery in Database
MDP	Markov Decision Process
MLDRL	Meta-Learning Based Deep Reinforcement Learning
RL	Reinforcement Learning
SARSA	State-Action-Reward-State-Action
TSP	Travelling Salesman Problem
TSSPPD	Travelling Salesman Problem with Pickup and Delivery
VRP	Vehicle Routing Problem
WBS	Work Breakdown Structure

CHAPTER 1

INTRODUCTION

1.1 General Introduction

This paper investigates a solution for Travelling Salesman Problem (TSP) with pickup and delivery (TSPPD) by proposing an optimal route to satisfy all customer requests and minimise transportation-related expenses. A request has to be a pickup node or delivery node, or both. TSP is one of the famous combinatorial optimisation problems frequently found in various domains for decades, including transportation and logistics. In an ordinary TSP, the overall goal is to find an optimal route between a set of nodes. Meanwhile, each node has to be visited just once by the agent. Moreover, the starting and ending points should be the same (Otoni et al. 2021). Unlike traditional TSP, the problem is reformulated with an additional precedence constraint, where each pickup node must be visited before its corresponding delivery node (Bai et al. 2021). With the evolution of Artificial Intelligence (AI), transportation route optimisation is becoming a streamlined process - AI will dynamically learn from the gathered data and provide the best route. To minimise the scope, this research paper only focuses on Reinforcement Learning (RL) and Genetic Algorithm (GA).

This chapter outlines a brief introduction to the background of the problem, problem statement, aim and project objectives, proposed solutions, approaches, as well as the project scope.

1.2 Background of The Problem

GlobalData's E-Commerce Analytics, an outstanding data and analytics organisation, reported that e-commerce sales in Malaysia rose at a CAGR of 22.4% from 2017 to 2021 (GlobalData, 2022). The sales were as much as 31.9 billion Malaysian Ringgit in 2021. According to International Trade Administration, 80% of Malaysian are active Internet users, whereas 84.2 % of them are mobile phone users. In the meantime, nine out of ten users are experienced in online shopping. There is an estimated revenue of US\$10523.7

million for the sales of online retailing in Malaysia by 2023, provided by EcommerceDB. With the shift from physical retail stores to e-commerce, the logistics industry is significantly affected in a positive way. In short, the booming of online consumers has resulted in strong demand for e-commerce postal delivery services. To gain competitive advantages over other couriers, most of them try their best to offer their customers effective pickup and delivery services with short delivery times. The customers refer to retailers or purchasers, or both. However, there is a high possibility for a vehicle to visit a customer multiple times if one of the customers happens to be the seller and the buyer in a single travelling tour. To elaborate, visiting a customer more than once is very inefficient and inconvenient for both the courier and the customer. This is because the courier carries out the delivery request before its corresponding pickup request. Consequently, the courier cost will increase, resulting in low customer satisfaction due to extended pickup and delivery time, especially during peak hours. This can be seen in a survey performed by Parcel Platform and iPrice Group within five countries: Singapore, Indonesia, Vietnam, Thailand, and Malaysia. Despite the expansion of Malaysia's e-commerce market, the survey reported that Malaysian consumers expressed the most dissatisfaction with our country's courier services across Southeast Asia (Editoron, 2019).

1.3 Problem Statement

This paper addresses the TSPPD and its application in courier services involving a distribution centre and a set of customers. Only a single vehicle is considered for each optimal pickup and delivery route. First of all, the courier will receive a set of pickup and delivery requests. Then, the courier departs from the distribution centre with goods to be delivered to a set of customers. There are some basic assumptions for this project:

- i. The courier should depart and return to the same distribution centre in a single travelling tour.
- ii. Each customer should be visited only once in a single travelling tour.
- iii. The capacity should be sufficient.

- iv. The courier should travel during working hours; hence, traffic condition is considered.
- v. The precedence constraint is predetermined.

Solving the TSPPD is important in reducing the total pickup and delivery route cost and increasing customer satisfaction with a short delivery time. Over 90% of customers complain about delayed deliveries and comment negatively on the inefficient route and poor schedule (Editoron, 2019). A study revealed that improving on-time delivery can effectively lead to higher customer satisfaction and loyalty (DÜNDAR & ÖZTÜRK,2020). In order to rebuild the customer's confidence in the courier services, many problems of pickup and delivery route planning need to be addressed.

The common challenges can be viewed from two main areas. The first would be a lack of pickup and delivery route optimisation, which led to an inefficiently planned route. The second would be integrating the precedence constraint into the pickup and delivery route.

1.3.1 Lack of Pickup and Delivery Route Optimisation

From a business perspective, the most significant impact of conventional pickup and delivery routes is that the courier has to spend extra money on petrol and waste time on inefficient routing. Moreover, the conventional route obviously does not take other conditions into account before performing route planning. Hence, the courier will stop at the nearest customer without considering whether it is efficient and will the traffic jam delay the schedule or not. Therefore, external factors, including traffic congestion and precedence constraint, should be considered for an ideal pickup and delivery route. Lack of route optimisation does not maximise the profit but requires additional time. Other than that, it is not cost-effective. According to Paragon (2017), the Frozen Food Express (FFE) has improved on-time delivery and reduced the delivery cost by 12% after adopting Paragon's route optimisation software. Route optimisation shall be adopted to solve this problem above.

1.3.2 Inefficiency in Handling Pickup and Delivery Request

As for the traditional vehicle routing problem, the routing problem is often viewed as a pure delivery problem or pickup problem. Nevertheless, the same customer can have both pickup and delivery requests in practical logistic distribution. Therefore, the courier is required to fulfil both pickup and delivery requirements, and thus this logistic service can be reformulated into a combinatorial problem (Min, 1989). Furthermore, same-day delivery can provide a competitive advantage for businesses to attract more customers who value the convenience and immediacy of same-day delivery. Same-day Delivery Services Global Market Report 2022 pointed out that the worldwide market for same-day delivery services is anticipated to expand from \$5.14 billion in 2021 to \$6.43 billion in 2022, representing a compound annual growth rate (CAGR) of 25.1%. On the other hand, customer satisfaction is closely tied to the efficiency of pickup and delivery services. When customers make pickup and delivery requests, they expect the courier to handle their requests in a timely and reliable manner. From the customer's perspective, the primary objectives behind the route optimisation are to make them happy while also potentially leading to cost reductions for the pickup and delivery service. Research by Chen and Ngwe (2018) emphasised the significance of shipping fees for online retailers and purchasers and the importance of effective pricing strategies in the e-commerce industry. To resolve such a real-world problem, offering simultaneous pickup and delivery service for customers based on the precedence constraint becomes a promising alternative to minimise the costs of satisfying all customer's requests and, in the meantime, improve transportation efficiency (Wang, 2016).

1.4 Aim and Objectives

This project aims to deliver an optimal route for TSSPD using RL and compare it with GA.

Objectives:

- i. To determine the optimal route for TSSPD using RL and GA considering traffic congestion and precedence constraint.
- ii. To implement the optimal pickup and delivery route solution on Google Maps.

- iii. To evaluate the effectiveness of route optimisation by comparing RL and GA.

1.5 Proposed Solution

The challenges mentioned above can be tackled by implementing a route optimisation technique into the current logistic system. Different from manual pickup and delivery route planning, the optimal route will minimise the courier operational cost depending on more than just distance. To do so, the optimal pickup and delivery route will prioritise the customer with the nearest distance from the distribution centre and so on. Besides, the optimal route is generated based on the latest traffic data during peak hours. Hence, time and efficiency can be maximised. This can benefit couriers by saving the petrol fee and enhancing the customer's satisfaction by providing the right route at the right time. Both RL and GA are used to formulate the best route-taking customer request and the latest traffic data as parameters. After learning from the data gathered, they will generate the optimal pickup and delivery route with maximum reward or fitness value. Later, the solution is implemented on Google Maps as it offers reliable directions around the world. A comparison between the optimal routes of both methods is conducted to evaluate the efficiency of RL and GA. The best route should contain minimum travelling duration and cost. From the customer's perspective, their satisfaction is directly influenced by their pickup and delivery requests, which can be settled at once in a pickup and delivery travelling tour. Figure 1.1 illustrates the workflow of the proposed solution.

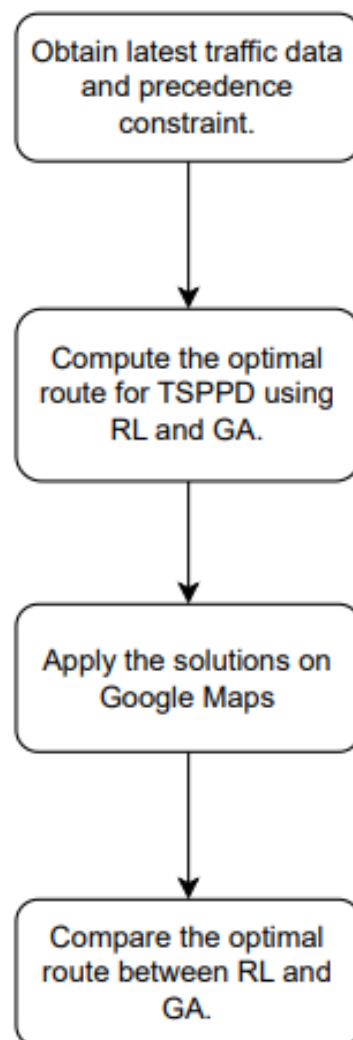


Figure 1.1 Workflow of Proposed Solution

1.6 Proposed Approach

The research methodology applied in this project is Knowledge Discovery in Database (KDD). However, this project replaces the data mining stage with RL and GA. Figure 1.2 shows the overview of the KDD process for data mining. The KDD process aimed to acquire meaningful knowledge from the raw and extensive database and apply it to various project domains. In this project, The KDD process mainly consists of seven stages which are goal setting and domain understanding, data selection, data pre-processing, data transformation, modelling (RL and GA), result evaluation/interpretation, and consolidation of discovered knowledge (Marbn, Mariscal and Segovi, 2009).

Firstly, the process requires a clear understanding of the project objective, as the wrong goal can result in false interaction. After the goals and objectives are defined, the target dataset needed for the KDD process is chosen, followed by data cleaning involving the removal of unwanted, redundant info. The next step is converting the extracted data into a suitable form and prepare to be fed into the algorithm. Later, generate an optimal route or near-optimal route with the transformed data. Once trends and desired outcomes have been obtained, the effectiveness of the RL model and GA will be evaluated in the view of the domain. Lastly, the discovered “knowledge” from the previous stages is ready to be applied in another domain or future activity.

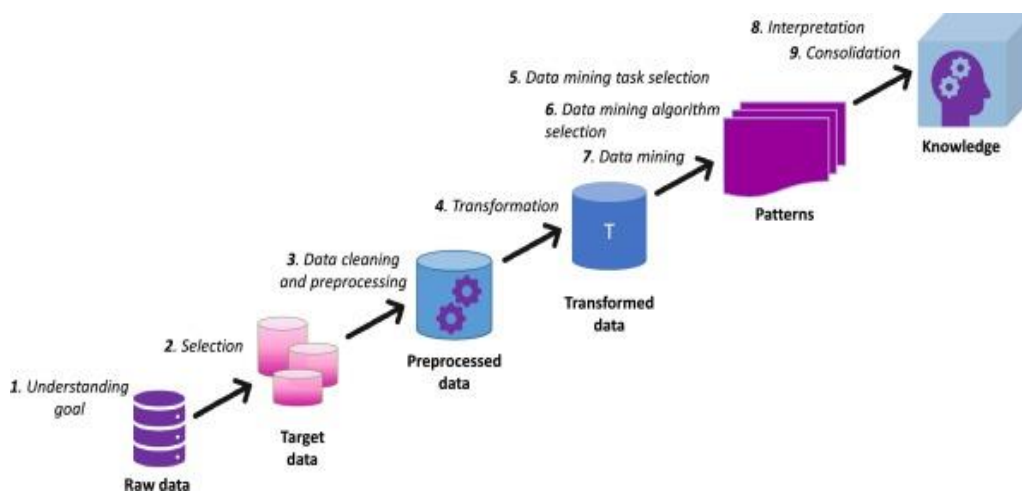


Figure 1.2 Overview of KDD Process for Data Mining (Kotu and Deshpande, 2019)

1.7 Project Scope

This project mainly focuses on finding the optimal route for TSPPD using RL and GA. In other words, route optimisation in this project is used to minimise the travelling duration of a single vehicle in its pickup and delivery route by considering external factors, including traffic congestion and precedence constraint. The traffic data is retrieved from Distance Matrix API powered by Google, whereas the precedence constraint is simulated. Both RL and GA are written in Python. The Deep Q-Network (DQN) will be implemented in RL to compute the optimal pickup and delivery routes using TensorFlow. TensorFlow is a Python-friendly open-source library that supports RL. The optimal pickup and delivery route solutions are implemented on Google Maps. After that, the Python Script is executed on the Jupyter Notebook, which provides a web-based interface that supports interactive computing. The device specifications are 8.00GB RAM, 64-bit operating system, x64-based processor, and Intel(R) Core (TM) i5-10210U CPU @ 1.60GHz 2.11 GHz. Lastly, the results of the RL model and GA are compared to evaluate the effectiveness of both techniques in route optimisation.

CHAPTER 2

LITERATURE REVIEW

2.1 Technique for Route Optimisation

This section introduces the significant findings and approaches for route optimisation problems by reviewing several prior studies.

Over recent years, a large variety of existing research has emphasised the importance of heuristic algorithms to solve route optimisation problems. Nayeem, Islam and Yao (2019) proposed that Transit Network Design Problem should be restructured into a many-objective optimisation problem in order to solve real-world problems. After that, the problem-specific genetic operator is implemented and eventually applied to recent evolutionary algorithms. However, there are two limitations to this approach. First, this approach generates a group of near-optimal solutions instead of the optimal solution. Hence, the transport operator must select the optimal route based on domain knowledge. Besides, this study does not consider traffic congestion while calculating travelling and waiting times. Wang, Ye and Wang (2020) suggested multi-level optimisation and hybrid heuristic techniques for transit route networks. To evaluate the effectiveness of the proposed technique, it was tested and compared with other algorithms using Mandl's Swiss road network as a benchmark. This approach has proven to satisfy more passengers with less travelling time.

Zhong et al. (2018) pioneered the Particle Swarm Optimization (PSO) technique with the acceptance criterion of simulated annealing to solve the TSP. The main purpose of acceptance criteria is to avoid premature convergence. Thus, the algorithm is able to escape from the local optima and search for the global optimum in the search space. Wang and Han (2021) have researched on optimising the TSP with Ant Colony Algorithm. The ACO parameters, including evaporation rate and exploration-exploitation parameter, are set at the initial stage. By doing so, it is proven that the SOS-ACO algorithm is capable of finding competitive solutions with fewer iterations and higher convergence

compared to the traditional ACO algorithm. The improved version K-Means clustering technique with an optimisation algorithm is proposed by Anantathanavit and Munlin (2016) to solve the TSP. This technique aims to find the optimal solution by dividing the large search space into subproblems and utilising the optimisation algorithm to obtain the optimal route in each cluster.

With the advancement of deep neural networks (DNNs), Zhang, Prokhorchuk and Dauwels (2020) mentioned that research on supervised DNNs for learning had been carried out and produced a satisfactory result. Nevertheless, it is necessary to have a predetermined set of solutions in order to train the supervised DNNs. Hence, more recent attention has focused on the provision of RL in addressing various types of combinatorial optimisation problems, including TSPs and Vehicle Routing Problems (VRP) (Bello et al., 2017; Nazari et al., 2018; Kool, Hoof, and Welling, 2019). Although considerable research has been conducted on tackling the typical TSPs, the study on solving the constrained TSPs with RL is still limited. One example is Ottoni et al. (2022) presented the Q-learning and State-Action-Reward-State-Action (SARSA) in addressing the TSP with refuelling. Prokhorchuk and Dauwels (2020) also conducted experiment on the TSP with time windows using RL.

There is a large number of published studies described the effectiveness of GA in solving the route optimisation problem (Purusoatham et al., 2022; Ha et al., 2020). Riazi (2019) has pioneered a double-chromosome method with GA to solve the TSPs. The proposed method showed a high convergence rate towards the shortest travelling tours. Nevertheless, the author suggested enhancing the result with improved operators. Hariyadi et al. (2019) provided that GA with natural selection could generate a promising result for TSPs regardless of the number of cities. Chen, Zhang and Du (2022) concluded that the GA could outperform the Ant Colony Algorithm and Particle Swarm Algorithm with better global optimisation ability and faster operating speed. The paper also suggested smaller population size should be considered in the early stage.

In conclusion, deep reinforcement learning (DRL), and GA are the best approaches for route optimisation to the best of current knowledge. RL can learn through interacting with the environment and solve the route optimisation problem without heuristics. It is significant to mention that most of the previous work used RL to interact with dynamically changing environments. Besides, GA provides relatively better performance in solving route optimisation problems too. Thus, RL and GA are adopted for experimental comparison in this project. Other than that, the previous method also highlighted that premature convergence should be carefully addressed. From the studies we surveyed above, several limitations exist in generating a well-suited optimal solution. For instance, the proposed mathematical model does not incorporate traffic congestion and precedence constraint. Also, hand-crafted heuristics are needed to optimise the route.

2.2 Solving Optimisation Problem Using Reinforcement Learning

In recent years, RL has gradually replaced traditional methods in solving optimisation problems. Some research has been done to study RL and evaluate its effectiveness in handling optimisation problems.

According to Xing and Cai (2020), the heuristic method can enhance DRL. The RL method outperformed the tabu search algorithm that requires a substantial domain with Markov Decision Process (MDP). Similar to the objectives of this project, the studies indicated that travelling duration had decreased tremendously, and customer satisfaction was uplifting with the implementation of RL. In short, the optimal route will directly determine the service quality (Abhyankar et al., 2018).

Zhang et al. (2022) have proposed Meta-Learning Based Deep Reinforcement Learning (MLDRL) for the multi-objective optimisation problem. Reptile-first-order gradient-based meta-learning algorithm will be applied as the meta-learning algorithm for complex combinatorial problems. The authors used Solomon's dataset as test instances. Moreover, the MRDRL is suitable for priori or posteriori schemes as well as provides better generalisation. Long training time for multiple sub-models incurred more cost and reduced the

model efficiency. With an improved model, the fine-tuning process can significantly lower down the number of gradient update steps and eventually shorten the training process.

TSP is a popular combinatorial optimisation. Miki, Yamamoto and Ebara (2018) have introduced heuristics using the CNN algorithm and RL to the 2D Euclidean TSP Model with geometric spatial. The training dataset is generated randomly according to uniform random. Furthermore, the Good-Edge Distribution is used to produce an optimal path. Although the error rate of this solution has reduced compared with the traditional method after being tested with supervised learning, the author suggested that stabilising algorithm (experience replay) must be adapted to overcome the learning collapse problem.

Miglani et al. (2021) have studied the effectiveness of Q-learning in shortening the total travel time by reducing the transfer stations of the direct passenger. Q-learning is one of the reinforcement algorithms. The proposed algorithm can be divided into four steps: training the Q-matrix, updating the Q-matrix, normalising Q-matrix, and producing an optimal route. The Q-learning will learn and update the Q-values for all-state action pairs. The result has shown that Q-learning can find the optimal metro route 7-9 times faster than the conventional path-searching algorithm. Nevertheless, the proposed approach is unsuitable for larger environments and becomes infeasible if the number of policies increases.

RL has become increasingly popular for solving optimisation problems, aiming to maximise the cumulative reward in a dynamic environment based on operant conditioning. The studies above outlined the effectiveness of RL (Xing and Cai, 2020), the need for fine-tuning hyperparameter process (Zhang et al., 2022), and experience replay while implementing RL (Miki, Yamamoto and Ebara, 2018). In addition, the last paper studied above has provided important information that the Q-learning algorithm will not be considered in this project as it is more challenging to learn multiple policies by computation of Q-values for all possible state-action pairs. One of the limitations of RL is the curse of dimensionality. Thus, this project would ensure the consistency of the

experimental nodes for both GA and RL. Overall, there is no doubt that RL can be viewed as one of the best approaches for optimisation problems after future improvement based on its advantages and limitations mentioned above.

2.3 Solving Optimisation Problem Using Genetic Algorithm

In recent years, numerous scholars have conducted extensive research in tackling route optimisation problems with GA. Most studies have found that GA generally outperformed existing route optimisation techniques and gained promising results.

Ha et al. (2020) have introduced a new hybrid GA with an improved version of the crossover method, a penalisation and restore mechanism to tackle the TSP with Drone. The restore method aims to enhance the algorithm's convergence, whereas the penalisation mechanism optimises the search among feasible and infeasible routes. They figured out that the proposed technique performs better than the existing method, such as Greedy Randomized Adaptive Search Procedure (GRASP), in finding optimal solutions after conducting extensive computational experiments.

Among different GA-based techniques often investigated in research, one famous algorithm is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Cai, Gao, and Yin, 2018; Deb et al., 2002). Deb et al. (2002) mentioned that the primary idea behind the NSGA-II is to search for a set of optimal solutions that are not dominated by other solutions and later perform sorting according to non-domination rank and crowding distance. It is undeniable that NSGA-II has a competitive advantage over other optimisation techniques in terms of generating multiple non-dominated solutions and convergence speed in different test scenarios (Yuan et al., 2018; Wang et al., 2018). Nevertheless, CHEN et al. (2019) argue that the NSGA-II does not guarantee a good exploration and exploitation trade-off strategy and might suffer from premature convergence while dealing with the bi-objective TSP (BTSP).

A series of recent studies have established that parameters such as mutation rate, population size, crossover rate and maximum generation are closely related to the generated fitness value (Han and Xiao, 2022; Herdiana et al., 2022). Besides parameters, Hameed and Kanbar (2017) and Vashisht et al. (2013) also pointed out that the choice mutation operator and crossover operator can affect the performance of the overall GA to a certain level. According to Han and Xiao (2022), it is necessary to adopt an adaptive GA as the fixed parameters fail to meet individual dynamic requirements will ultimately lead to a drop in the performance and efficiency of the GA. Han and Xiao (2022) also proved that adaptively improving the mutation probability can effectively increase the convergence speed and operational efficiency.

In short, GA is one of the more representative techniques in solving the multi-objective optimisation problem. However, the mutation operator, crossover operator and its parameters should be taken into account while designing the GA to increase the convergence speed and operational efficiency. Since the effectiveness of the crossover operator and mutation operator is highly dependent on the problem context, this project adopts Practically Mapped Crossover (PMX), adaptive mutation, and elitism techniques to avoid premature convergence. In addition, population size and the number of generations are carefully selected to obtain an optimal or near-optimal result.

2.4 Google Maps API

Recently, an increasing number of developers have embedded Google Maps into their applications or website. For instance, web mapping platforms such as Mango Map, Mapbox, and Google Maps are used by transport service providers in the decision-making process (Ullah et al., 2020). Real-time and accurate data is crucial for measuring the traffic congestion in the area as the road conditions change continuously (García-Albertos et al., 2019). Google has offered a variety of APIs to fulfil different needs. The services can be categorised into three major fields, which are Maps, Places, and Route (Muñoz-Villamizar et al., 2021). In 2018, García-Albertos et al. claimed that Google Maps API could help to evaluate the dynamic accessibility of different areas and their travelling time.

Fu et al. (2010) have summarised three main advantages of Google Maps: low development cost, up-to-date data, and analysis. Firstly, Google Maps API is free to use if the usage does not exceed \$200 per month with a valid API Key. Moreover, the user can acquire the latest geographical and traffic information updates from map service with vector maps and high-resolution street view. Lastly, Google Maps support spatial analysis functions such as measurement and path analysis (Jinhui, 2007).

The primary focus of this project is to integrate the Google Maps API into the machine learning techniques. Thus, an optimal pickup and delivery route capable of the real-world environment can be generated considering the latest traffic data and precise geographical information.

2.5 Performance Evaluation Based On The Shortest Estimated Time of Arrival (ETA)

Customer satisfaction in the context of route optimisation is usually based on the shortest ETA they can experience. Parasuraman et al. (1985) introduced the SERVQUAL model, which serves as a tool for assessing service quality. The model consists of five dimensions, namely tangibility, reliability, assurance, empathy, and responsiveness. Kersten and Koch (2010) highlighted the significance of the reliability dimension in terms of timely delivery, resolving customer issues and ensuring accuracy in the first pickup or delivery attempt. In short, this dimension proposed that service quality is highly associated with pickup and delivery time. Besides, Saad (2020) reviewed over 45 articles related to online purchasing and found that delivery time positively influences customers' adoption and use of online purchasing. There is plenty of research and proposed methods to investigate the best way to evaluate passengers' satisfaction regarding pickup and delivery service domain for Vehicle Routing Problems. Niu et al. (2018) claimed that the traffic actual traffic data must be considered while examining customer satisfaction. Thus, the experiments were conducted in 120 areas, including congested areas, with data obtained from actual geographical passenger data from Beijing. Tang et al. (2009) proposed a solution to a vehicle routing problem with fuzzy time windows, where the service time may deviate from the customer-specific time window. The degree

of deviation between the service time and the time window was considered a measure of customer satisfaction. Barkaoui (2015) utilised the Bayesian formula to update customer satisfaction for multiple visits to customer points. Pan et al. (2020) claimed that serving customer requests before the customer's expected time while adhering to the transportation cost constraint can maximise customer satisfaction.

In a nutshell, even though every author has a different perspective and evaluation method on customer satisfaction, we can commonly agree that pickup and delivery time are highly associated with customer satisfaction rates in this project. According to Fan (2011), the shorter the waiting time, the higher the satisfaction rate. Hence, the multi-objective function is proposed to lower the overall transportation cost and increase total passenger satisfaction. Most of the research has put extra effort into performing computational analysis to compare the proposed solution with some baseline algorithms for evaluation. For this project, a comparison will be conducted to evaluate the performance of our proposed model. Nambisan et al. (2016) assumed that the passenger is completely satisfied with the optimal route if the travelling duration and the waiting duration for all interactions between the customer and the service provider are shorter in the adopted route optimisation method.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

This chapter describes each phase of this project's methodology and work plan. The research-based method is KDD, which involves data handling, modelling, and extracting applicable “knowledge” from a large database. Moreover, this chapter also explored adopted development tools and technologies. Lastly, Work Plan, including Work Breakdown Structure (WBS) and Gantt Chart is generated to smoothen the project planning and scheduling.

3.2 Research Methodology

The chosen research-based development methodology for this project is Knowledge Discovery in Database (KDD). The general definition of the KDD process and its stages have been mentioned in Chapter 1 (Section 1.5). Hence, the section focuses on how this project's workflow is associated with each stage of the KDD process in a specific manner. These steps can be performed iteratively if necessary. Figure 3.1 below illustrates the overview of the KDD process for RL and GA.

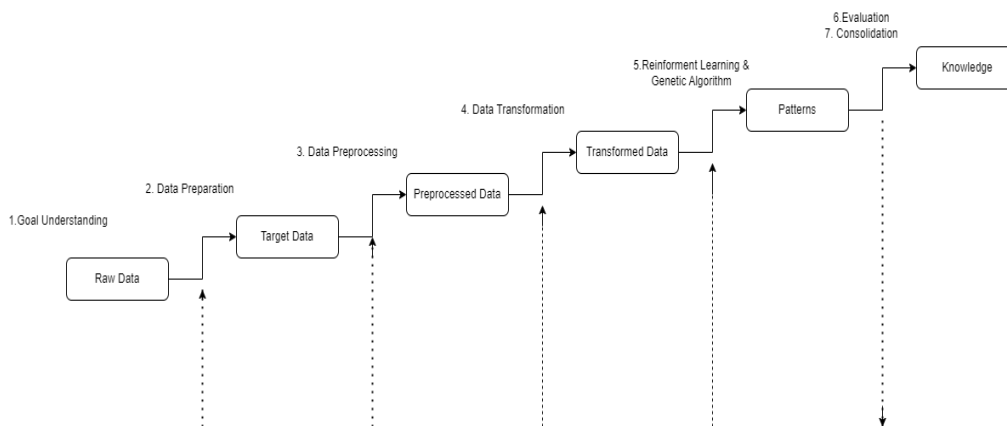


Figure 3.1 Overview of KDD Process for RL and GA

3.2.1 Goal Understanding

To kick start this project, the project domains and goals must be identified, and related prior knowledge needs to be studied. Thus, problem statements (Section 1.2) and project objectives (Section 1.3) are described clearly and listed accordingly in Chapter 1 as a guideline for this project during the research process. Besides, the literature review is performed in Chapter 2 to ensure the project's feasibility and gain more relevant knowledge.

3.2.2 Data Preparation

Once the goal is defined, quality and meaningful datasets that are required is selected and gathered. One of the objectives of this project is to determine the optimal pickup and delivery route with RL and GA, considering traffic congestion and precedence constraint. Thus, ETAs travelling to 16 locations in Klang Valley at the current time are collected in this project. The precedence constraint is simulated. Furthermore, the traffic congestion data is requested through Google Maps-Distance Matrix API. Figure 3.2 below shows the sample response from Google Maps-Distance Matrix API.

```
{
  "destination_addresses" : [ "Cheras 11 Miles, 43200 Cheras, Selangor, Malaysia" ],
  "origin_addresses" : [
    "Jalan Sungai Long, Bandar Sungai Long, 43000 Kajang, Selangor, Malaysia"
  ],
  "rows" : [
    {
      "elements" : [
        {
          "distance" : {
            "text" : "4.8 km",
            "value" : 4797
          },
          "duration" : {
            "text" : "10 mins",
            "value" : 579
          },
          "duration_in_traffic" : {
            "text" : "10 mins",
            "value" : 591
          },
          "status" : "OK"
        }
      ]
    }
  ],
  "status" : "OK"
}
```

Figure 3.2 Sample Response from Distance Matrix API

3.2.3 Data Pre-processing

During this pre-processing process, only necessary information from the response of the Distance Matrix API is collected for the RL model. The critical information that is needed for further modelling is the “duration_in_traffic” data. The “value” in “duration_in_traffic” represents the duration from origin to destination, considering traffic congestion in seconds. To ensure the accuracy of the result, the “value” is used as the data parameter to be fed into the model. It is declared as either reward or punishment during the training process in RL, whereas it is the fitness value for the GA as well. Besides, noisy data handling is carried out during this phase to ensure better data quality and data accuracy. If the retrieved ETA is not zero when both the origin and destination are the same, that ETA will be changed to zero.

3.2.4 Data Transformation

Data transformation can uplift efficiency during the decision-making process. In this project, the “duration_in_traffic” data is extracted in JavaScript Object Notation (JSON) format. To make the data easier to use by the machine for RL, the data in JSON format is converted into a Python dictionary with `json.loads()`.

3.2.5 Machine Learning

Datasets Used

To present the proposed machine learning technique in a simpler and clearer way, a study conducted by Filip, (1970) regarding the TSP and its feasibility in logistic field within specific constraints is referred. Thus, a total of 16 nodes, including one distribution centre (node 0) and 15 customers (node 1-15), are selected in the proposed solutions below. Since this project has considered traffic congestion, the ETAs of these 16 nodes are retrieved within working hours, which is 2:30pm. Each training and testing dataset contains 16 nodes and their ETA matrixes, the node’s locations are randomly selected around the Klang Valley area in Malaysia. Figure 3.3 and Figure 3.4 below provide the details of the training dataset, whereas Figure 3.5 and Figure 3.6 illustrates the details of the testing dataset. The ETA matrix is symmetric. For instance, if the

courier moves from node 0 to node 1 in the training dataset, the ETA will be 614 seconds. Also, the data is read from the Excel file, and their ETAs are saved in the dataframe format.

```
Training=['7 eleven, TAMAN CHERAS PERDANA, 43200 CHERAS, SELANGOR.',
'Cheras Vista, Bandar Mahkota Cheras, 18, Jalan Vista 3, Bandar Sungai Long, 43200 Kajang, Selangor',
'Jalan Ridgeview, Taman Bukit Permai, 43000 Kajang, Selangor',
'2, Jalan Nusaputra 4/2c, Bandar Nusa Putra, 47100 Puchong, Selangor',
'7, Jalan Ptp 1/2, Taman Perindustrian Tasik Perdana, 47120 Puchong, Selangor',
'Taman Putra Prima, 28, Jalan PP 2/3, Taman Putra Prima, 47130 Puchong, Selangor',
'Sunway Lagoon, 3, Jalan PJS 11/11, Bandar Sunway, 47500 Subang Jaya, Selangor',
'Cuci Station Selangor, Mentari Court, 22, Jalan PJS 8/12, Bandar Sunway, 46150 Petaling Jaya, Selangor',
'Sri Petaling, KL 2027 Dewan Seberguna Taman Sri Indah, , Jalan 5, 149B, Taman Sri Endah, 57000 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur',
'124, Jln Sb Indah 3/1, Taman Sungai Besi Indah, 43300 Seri Kembangan, Selangor',
'Kelana Sentral, 71, Jalan SS 6/12, Ss 7, 47301 Petaling Jaya, Selangor',
'1, Jalan Petaling Utama 11, Taman Petaling Utama, 58200 Petaling Jaya, Selangor',
'Vila Vista Condominium, Vila Vista, Jalan Selar 4, Taman Pertama, 56100 Kuala Lumpur, Federal Territory of Kuala Lumpur',
'22a, Jalan Bahagia 15a, Taman Seri Bahagia, 56000 Kuala Lumpur, Selangor',
'PV3 (Platinum Victory), 53100 Kuala Lumpur, Selangor',
'Residensi Kuchaimas, Jalan Indrahana 3, Taman Indrahana, 58100 Kuala Lumpur, Federal Territory of Kuala Lumpur']
```

Figure 3.3 16 Node’s Location in Klang Valley for Training Dataset

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Node	0	701	1278	2249	2266	2356	2011	2033	1644	858	2417	1694	1146	640	1982	1557
1	614	0	1383	2415	2360	2456	2126	2144	1757	1308	2488	1805	1230	1111	2125	1665
2	1251	1392	0	2064	2053	2165	2553	2454	1964	1527	2757	2248	1803	1698	2700	2157
3	2120	2292	1813	0	452	1002	2146	2036	1456	1776	2409	1806	2056	2490	2975	1842
4	2135	2312	1842	565	0	630	1814	1738	1649	1821	2088	1987	2286	2542	3184	2015
5	2335	2487	2052	1151	774	0	1644	1560	1882	1996	1915	1788	2497	2789	3530	2175
6	2162	2276	2226	1836	1859	1681	0	581	1227	1895	1085	834	1844	2108	2770	1328
7	2019	2120	2197	2063	1665	1512	540	0	1191	1771	728	700	1705	2016	2632	1210
8	1546	1667	1492	1531	1696	2015	1279	1203	0	1166	1562	1090	1137	1534	2062	826
9	749	1012	1171	1780	1778	1889	1659	1647	1256	0	1993	1309	1311	1101	2203	1177
10	2172	2304	2282	2151	1776	1608	802	688	1283	1976	0	838	1926	2157	2345	1340
11	1672	1804	1905	1944	1793	1636	601	602	977	1454	989	0	1387	1661	2176	797
12	1025	1145	1740	2120	2307	2689	1583	1618	1239	1508	2055	1270	0	1024	1491	1141
13	572	1017	1670	2439	2622	2741	1941	1975	1581	1030	2331	1636	1048	0	2025	1501
14	2205	2337	2925	3073	3165	3702	2577	2774	2183	2514	2344	2242	1683	2234	0	2160
15	1397	1532	1673	1680	1828	2117	1056	1064	795	1036	1435	714	988	1405	1794	0

Figure 3.4 ETA Matrix for Training Dataset

```
Testing=['Best View Hotel, SS2 Petaling Jaya, SS 2, Petaling Jaya, Selangor, 47300 Petaling Jaya, Selangor',
'Sushi Mentai, PA-S, Pearl Avenue, 23(G, Jalan Pasir Emas, Sungai Chua, 43000 Kajang, Selangor',
'Serika Residences, Jalan TKS 1, Taman Kajang Sentral, 43000 Kajang, Selangor',
'Kuchai Entrepreneurs Park, Kuala Lumpur, Federal Territory of Kuala Lumpur',
'Sultan Abdul Aziz Shah Airport (Subang Airport) (SZB)',
'Residensi The Trees, Damansara, 55, Jalan Bukit Lanjan, Sungai Penchala, 60000 Kuala Lumpur, Federal Territory of Kuala Lumpur',
'Shah Alam Stadium, lot no g43, Level 1, Kuadran AB, Stadium, 1, Jln Akuatik 13/64, Seksyen 13, 40100 Shah Alam, Selangor',
'Belly and the Chef Cafe (PJ), 625, 1st floor, Jalan 17/8, Seksyen 17, 46400 Petaling Jaya, Selangor',
'Ioi Puchong Jaya, Bandar Puchong Jaya, 47100 Puchong, Selangor',
'Balakang Badminton Sports Center, 2012, Jalan Besar, Kampung Baru Balakong, 43300 Seri Kembangan, Selangor',
'National Stadium Bukit Jalil, Jalan Barat, Bukit Jalil, 57000 Kuala Lumpur, Federal Territory of Kuala Lumpur',
'Kota Kemuning, 45, Jalan Tukul P15/P, Seksyen 15, 40200 Shah Alam, Selangor',
'Damai Apartment, Jln PJS 8/9, Bandar Sunway, 46150 Petaling Jaya, Selangor',
'7-11 Icon City, Jalan SS 3/39, Sungai Way Free Trade Industrial Zone, 47300 Petaling Jaya, Selangor',
'Taman Hulu Langat Jaya, 2, Jalan Hulu Langat Jaya 2/2, Taman Hulu Langat Jaya, 43200 Cheras, Selangor',
'FamilyMart Gombak, Ground Floor, 13G, Jalan Prima SG 1, Prima Seri Gombak, 68100 Batu Caves, Selangor' ]
```

Figure 3.5 16 Node’s Location in Klang Valley for Testing Dataset

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
State																
0	0	2522	2620	1555	1336	931	1678	607	1340	2463	1661	1543	1057	747	2403	1619
1	2552	0	689	1478	2741	2802	2898	2363	2005	1067	1374	2867	2021	2157	1329	2631
2	2911	744	0	1735	3055	3143	3159	2651	2225	1276	1676	3113	2298	2426	1735	2949
3	1454	1535	1567	0	1676	1724	1844	1325	1419	1066	828	1719	1054	1201	1558	1926
4	1214	2556	2617	1632	0	1384	1354	1367	1387	2459	1631	1283	1007	980	3064	2159
5	728	2621	2668	1688	1465	0	1991	990	1696	2548	2052	1769	1391	1098	2490	1187
6	1495	2780	2787	1617	1185	1893	0	1643	1513	2416	1878	430	1017	997	3038	2453
7	463	2609	2659	1514	1311	999	1737	0	1619	2319	1863	1617	1306	912	2367	1715
8	1178	1758	1786	1076	1292	1697	1436	1335	0	1476	866	1350	550	707	2076	2394
9	2252	1038	1289	1114	2499	2486	2604	2052	1855	0	1211	2539	1839	1971	1136	2409
10	1777	1326	1388	727	1852	2147	2006	1695	953	967	0	1895	1090	1235	1823	2166
11	1703	2770	2819	1844	1371	2243	579	1843	1626	2653	1945	0	1207	1192	3151	2656
12	992	2073	2118	1069	969	1514	1151	1085	861	1850	1185	1043	0	489	2224	2209
13	748	1996	2066	937	927	1416	1114	942	806	1703	1144	997	454	0	2038	2135
14	2260	1146	1418	1339	3065	2488	2800	2044	2375	1254	1626	2741	2076	2408	0	2282
15	1658	2578	2735	1766	1992	1100	2501	1541	2641	2523	1886	2476	2328	2024	2318	0

Figure 3.6 ETA Matrix for Training Dataset

Precedence Constraint

For the experimental purpose, only one precedence constraint (3,2) is considered in this project. This means that node 3 should be visited before node 2 in any circumstance, regardless of the ETA.

Reinforcement Learning

In general, RL is a subset of machine learning applied in the decision-making process through trial and error. It trains the agent to perform the action in a given environment to obtain a maximum reward. The primary aim of RL is to maximise the total rewards. To do so, the agent is designed based on reward and punishment mechanisms. In other words, the agent is rewarded for positive behaviours and penalised for negative behaviours. For example, the AI starts the play without prior knowledge and gradually improves over time by obtaining higher scores. Figure 3.7 below illustrates the general RL model. Figure 3.8 shows the flowchart of implemented RL in this project.

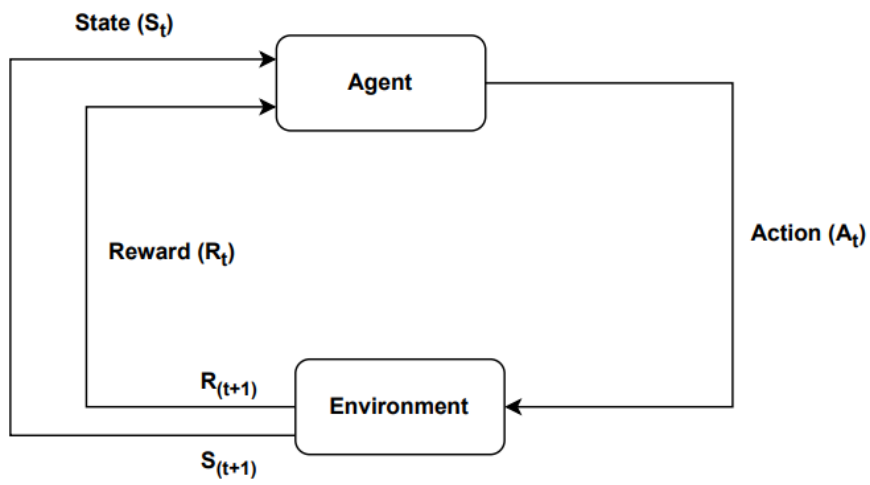


Figure 3.7 RL Model (adopted from Spiceworks, 2022).

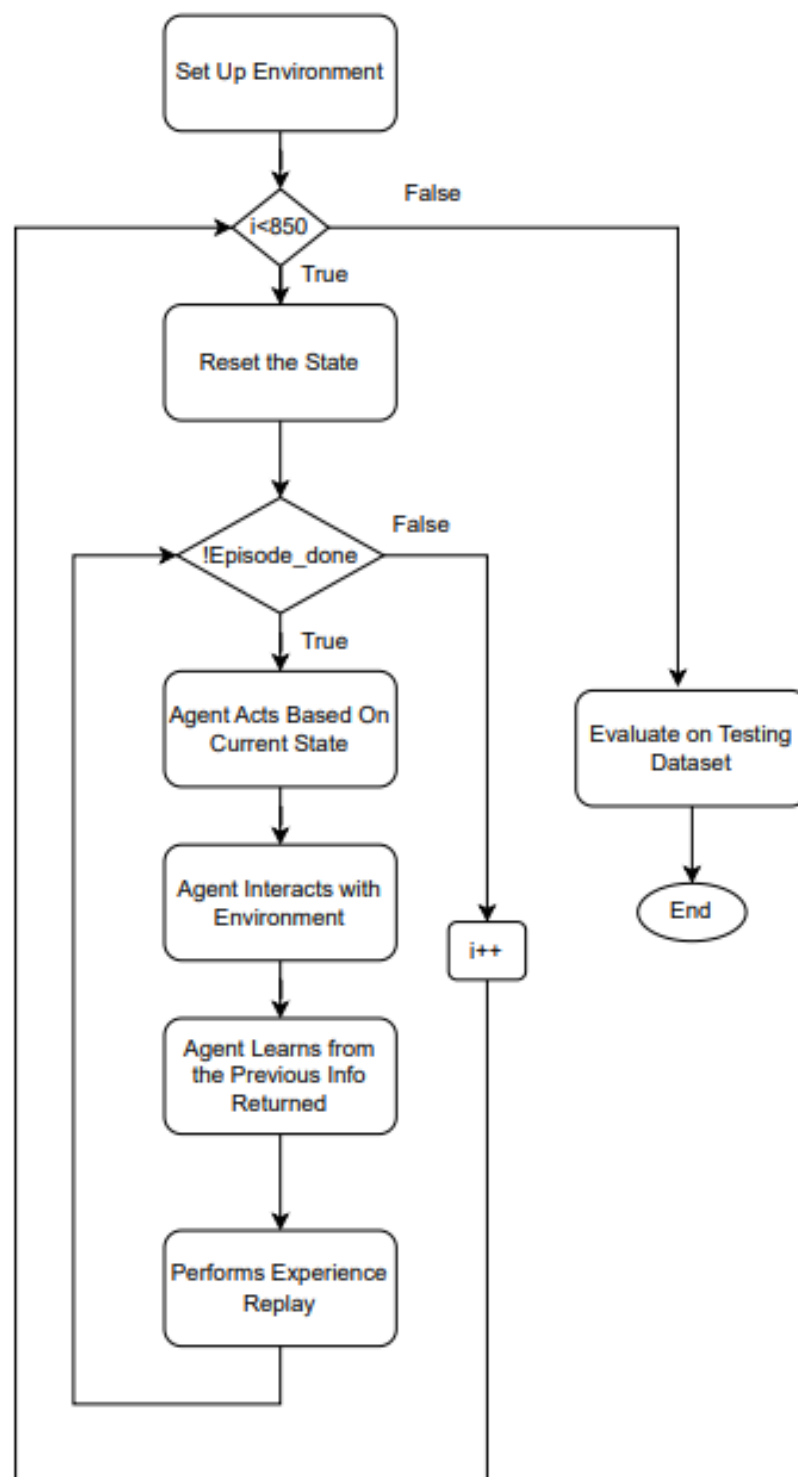


Figure 3.8 Flowchart of Implemented RL

I. Markov Decision Process (MDP)

The problem in RL is commonly formulated based on MDP. The MDP adopted from Dechter (2018) is shown below. There are five components in MDP: a set of states (state space), a set of actions (action space) transition probability matrix, a reward function, and a discount factor. In our case, the states are the different customers, the actions are moving from one customer location to another, and the reward function is created based on travel duration in traffic and precedence constraint. The policy is the action that the agent takes in a given state, also a solution to the MDP. Later, the expected return based on policy is evaluated by the state-value function and action-value function.

Policy Function:

$$\pi(a|s) = P[A_t = a|S_t = s] \quad (1)$$

State-Value Function:

$$v_\pi(s) = E_\pi[G_t|S_t = s] \quad (2)$$

Action-Value Function:

$$q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a] \quad (3)$$

Markov Decision Process is a tuple (S, A, P, R, γ) :

- *S is a finite set of states*
- *A is a finite set of actions*
- *P is a state transition probability matrix, $m P(s'|s, a)$*
- *R is a reward function, $R(s, a, s')$*
- *γ is a discount factor*

II. Environment Setup

To begin the experiment, the environment is defined using the OpenAI Gym library. An array with 16 possible nodes environment that is represented as a binary vector from node 0 to node 15 is set up as shown in figure 3.9.

The main components of the environment are:

- i. Learning Agent:

In the initial state, the learning agent is located at the starting point, node 0, as shown in Figure 3.10. It is ready to move to the next state when it is activated. The vector will have a 1 in the position corresponding to the visited nodes. For instance, if node 1 is visited in the next state, the one-hot encoding is illustrated in Figure 3.11. Once the agent is terminated in an episode, the environment will be reset to its initial state.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Figure 3.9: Observation Space

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Figure 3.10: Initial State

1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Figure 3.11: Node 1 is Visited

ii. Goal State:

The goal state is reached when the agent successfully visits all unvisited states while obeying the precedence constraint.

iii. Termination State:

The agent will be forced to terminate if it does not follow the precedence constraint or visit the visited node again.

iv. Agent Action

A set of 16 possible allowed actions from integer 1 to 15 inclusive for the agent, which represents the node that the agent can move to from its current node. Node 0 is excluded as it is fixed as the starting and ending point.

v. Reward

The reward function is designed to encourage an agent to find the optimal or near-optimal solution. Hence, the environment will give a positive reward if the agent visits the unvisited node while obeying the precedence constraint, whereas a negative reward is awarded for an undesirable situation as mentioned in the Termination State section above. To find the optimal route with the shortest ETA, the reward is set to be inversely proportional to the ETA among the nodes. The pre-defined reward functions are clearly showed as below:

Positive Reward Function:

$$Reward_{positive} = (1/ETA_{(S,S_{t+1})}) * 2000 \quad (4)$$

Negative Reward Function:

$$Reward_{negative} = -20 \quad (5)$$

There are various algorithms in RL. This project has adopted Deep Q-Network (DQN) approach. Figure 3.12 illustrates the Q-learning and DQN architecture. Unlike Q-learning, the most common model-free algorithm in RL, the agent's brain of DQN is a fully connected neural network instead of a Q-

value table. DQN can outperform traditional Q-learning by reducing the correlation between consecutive experiences and improving generalisation. To do so, the neural network in DQN will learn from the experience replay. Experience replay happens by storing a collection of past agents' experiences in a buffer and later randomly sampling a batch of experiences during each training iteration. Thus, the optimal policy can be learned faster without a huge number of interactions with the environment.

Table 3.1: Comparison of RL Algorithms

Algorithm	Agent's Brain	Target, Y_t^Q
Q-Learning	Q-table	$Y_t^Q = R_{t+1} + \gamma \max Q(S_{t+1}, a)$
Deep Q-Network (DQN)	One deep neural network model	$Y_t^{DQN} = R_{t+1} + \gamma \max Q(S_{t+1}, a; \theta_t^-)$

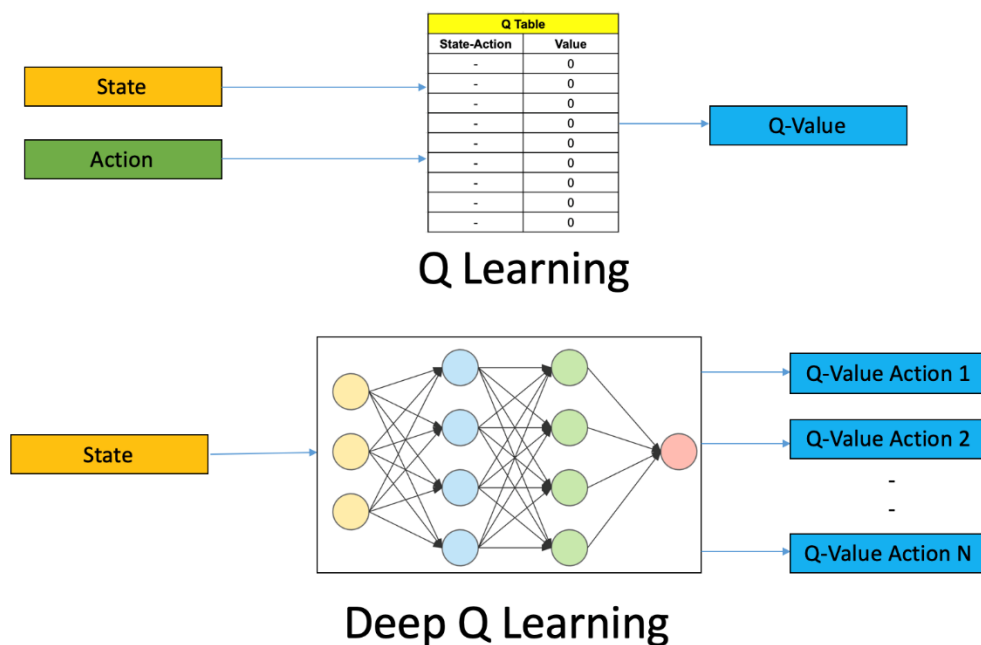


Figure 3.12 Comparison between Q-learning and Deep Q-learning
(Choudhary, 2020)

III. Deep Q-Network

i. Design

Firstly, the DQN model architecture is designed to take in the current observation and output the Q-value for each possible action. In this experiment, the neural network model is created with Keras Sequential API for deep RL. It contains two ‘Dense’ layers with 24 neurons. The activation function being used is Rectified Linear Unit (ReLU), which is commonly used in deep neural networks. In brief, the DQN model architecture is a simple feedforward neural network with two hidden layers and a linear output layer that allows the training agent to learn the optimal policy.

ii. Implementation

Next, the DQN model is implemented with the Tensorflow DRL framework.

iii. Train Model

The DQN model is trained with a well-balanced exploration and exploitation rate in the agent’s decision process. The highest Q-value is chosen during exploitation, whereas random action is taken during exploration. Besides, the experience replay is carried out to train the model using a randomly sampled batch of past experiences stored in the agent's memory. For each experience, the function computes the target Q-value using the Bellman equation. The training process is repeated iteratively until the agent learns the optimal policy updating the Q-values based on past experiences.

Bellman Equation for DQN:

$$Y_t^{\text{DQN}} = R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta_t^-) \quad (6)$$

Other than that, the hyperparameters, such as learning rate and epsilon probability, are carefully tuned. After trying out various combinations of hyperparameters, the finest solution is produced with :

MEMORY_SIZE = deque([], maxlen=2500)

GAMMA = 0.95

EPSILON = 1.0

EPSILON_DECAY = 0.995

EPSILON_MIN= 0.01

LEARNING_RATE = 0.001

iv. Evaluation

Once the DQN model is trained, it is applied to the testing dataset. The performance is evaluated based on the majority optimal route it generated within 60 testing episodes.

Genetic Algorithm

This project has separated the GA separated into seven main stages: Fitness Value Calculation, Population Initialization, Selection, Crossover, Mutation, Repeat, and Termination. Figure 3.13 below illustrates the overview of the GA method flow in this project.

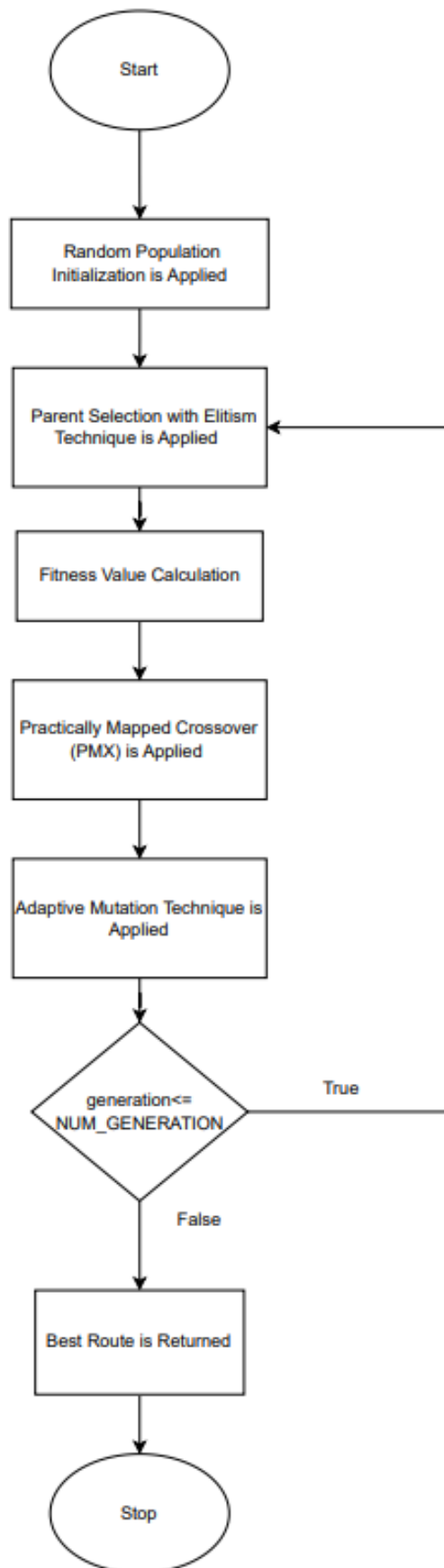


Figure 3.13 Flowchart of Implemented GA

I. Initial Population

The initial population is generated through random shuffling of potential solutions. As the POPULATION_SIZE is set to 500, 500 individuals are created for every loop. Each individual contains a list of nodes from 1-15. Node 0 is excluded as it is mandatory to be visited at the beginning and the end of the travelling tour.

II. Parent Selection with Elitism Technique

Random parent selection is required to choose the parents from the existing population. This stage aims to create children individuals for the crossover and mutation processes later. The Elitism strategy is applied to preserve a list of best-performance individuals. Hence, good diversity in the population can be assured and at the same time, pertaining individuals with optimal fitness value.

III. Fitness Value Calculation

The fitness value is the key measure in selecting the optimal solution. In this project, the individual's fitness function is computed by summing up the total ETA of every single route in the population. Thus, the lower the fitness value, the better the solution. However, if the generated does not follow the precedence constraint stated above, that particular individual will be penalised with a huge fitness value.

$$Fitness\ Value_{individual} = \sum_{n=1}^{nodes\ count} n_i \quad (7)$$

IV. Partially Mapped Crossover

The Crossover operator selects more than one parent individual to perform genetic crossover and produce new individuals. A partially Mapped Crossover technique is implemented in this project. First, all unvisited nodes of a child individual with a length similar to the parent individual are marked with -1. It will randomly select the “start” and “end” indices from the parent1, representing the segment to be copied to the corresponding child individual later. In contrast, the crossover operator will loop through every node of the parent2 to check

where it has existed in the child individual. If the node is not found, the crossover operator will copy the corresponding node from parent1 to the child individual in the corresponding index of parent2.

V. Adaptive Mutation Technique

Adaptive Mutation is a technique to adjust the mutation probability throughout every generation based on the total individual's fitness value in the current population. The main idea is to selectively increase the mutation rate for those individuals with a low fitness value; thus, the exploration rate and the solution diversity can be increased. In short, the higher the individuals' fitness value, the lower the mutation probability. As the exploration and exploitation rates are well-balanced, a better convergence and optimisation performance for GA can be achieved. In this project, the mutation probability is calculated as below:

$$Mutation\ Probability = \frac{\sum Fitness\ Value}{(Population\ Size * Best\ Fitness\ Value)} \quad (8)$$

VI. Repeat

The GA loops through the stages mentioned above until a termination condition is reached. The pseudocode of a complete GA loop is shown in the Figure 3.14 below.

```

BEGIN GA
  #Initial Population
  Generate initial population
  #Repeat
  Repeat until termination condition is met
    BEGIN REPEAT
      #Parent Selection with Elitism Technique
      Select parent population for reproduction with elitism technique
      #Fitness Value Calculation
      FOR each individual in population
        Calculate fitness value using fitness function
      END FOR
      #Partially Mapped Crossover
      Select 2 individuals randomly from population
      Perform partially mapped crossover to create new individual
      #Adaptive Mutation Technique
      FOR each gene in the individual
        IF a random number is less than the mutation rate
          Perform adaptive mutation to mutate the individual
        END IF
      END FOR
      #Termination
      IF termination condition is met
        EXIT loop
      END IF
    END REPEAT
    RETURN the individual with the best fitness value
  END GA

```

Figure 3.14 Pseudocode of A Complete GA Loop

VII. Termination

The repeated GA loop is terminated when it reaches an absolute number of generations. After trying out various combinations of `POPULATION_SIZE`, `ELITE_SIZE`, and `ELITE_SIZE`, the finest solution is produced with :

POPULATION_SIZE = 500

ELITE_SIZE = 100

NUM_GENERATIONS = 500

3.2.6 Evaluation

In this project, the effectiveness of the route optimisation model is evaluated by the result comparison of the RL and GA. The research done in Chapter 2 (Section 2.5) proved that the lesser the travel duration while fulfilling the precedence constraint, the better the route optimisation model, and the higher the user's satisfaction. Both training results are visualised to make them more understandable. Since the reward calculation in RL and the fitness value evaluation in the GA are two distinct processes, the final scores of the optimal routes are converted into total ETA in seconds and compared. By doing so, the effectiveness of both methods can be evaluated in a more standardised and comprehensive way.

3.2.7 Consolidation

In the last stage, the discovered knowledge of generating a route optimisation model for TSPPD using RL and GA is well-documented in a report form. The generated report can be used as a reference for future work as it has explored the possibility of implementing optimal pickup and delivery routes for courier services in Malaysia with machine learning techniques.

3.3 Research Tool and Technology Used

3.3.1 Visual Studio Code

Visual Studio Code is a streamlined code editor that supports many programming languages, such as Java, C++, and Python. It is free and available for cross-platform, including Microsoft Windows, Mac OS X, and Linux. VS Code provides a variety of features, such as running, debugging, and testing code. Besides, VS Code is more convenient for the user to switch between Python environments, including the virtual and Conda environment. For this project, we used Python as our main programming language. Thus, Microsoft Python extension is installed for better project development.

3.3.2 TensorFlow

TensorFlow is an open-source library for machine learning provided by Google. It can develop and train machine learning models faster and more effectively with high-level APIs such as Keras. Keras is a Python-based deep learning API aimed at optimising experimentation. Also, TensorFlow can easily deploy in the cloud or external browsers. Besides, TensorFlow is used to integrate the neural network in our project.

3.3.3 Google Colab

Google Colab is a free Jupyter Notebook environment that Google Research built. The purpose of Google Colab is to allow developers to produce and execute Python code through Google's cloud servers. As it runs on a cloud environment, the user can share their Colab notebook with the public, allowing them to comment or modify the code. After finished execution, the Colab notebook can be saved in the personal Google Drive account to serve as a backup for this project.

3.3.4 GitHub

GitHub is a web-based hosting service that helps to manage open-source project repositories better. GitHub overcomes the distance challenge and fosters communication. Hence, users around the world are able to work collaboratively on the same project and invent new project versions without affecting the current version. Furthermore, GitHub allows users to host their projects in various programming languages, including C++, Java, and Python. The developers can also access other developers' repositories if it is made "public" and store remote copies of repositories. To develop this project successfully, several similar projects in GitHub are reviewed and studied.

3.3.5 Google Maps-Distance Matrix API

Distance Matrix API allows users to request the travel distance and duration for a matrix of origins and destinations. However, an API key must be generated before accessing the Distance Matrix API functions as well as enabled billing in the Cloud Console. Users can customise the requested info by specifying

departure time, preferred transportation mode, and traffic model. Moreover, user can get the traffic data and uses location modifiers at a higher bill rate. This project has used Distance Matrix API to retrieve predicted travel duration in traffic information for training and real-time data for the testing process in RL.

3.4 Work Plan

3.4.1 Work Breakdown Structure

Work Breakdown Structure (WBS) is used for project management by dividing and conquering large projects. It contains all the significant tasks of the project. The WBS of this project is listed as follows:

0.0 Public Transport Route Optimization with Reinforcement Learning

1.0 Planning and Analysis

- 1.1 Register Project Title
- 1.2 Study Project Background
- 1.3 Identify Problem Statement
- 1.4 Define Aim and Project Objectives
- 1.5 Propose Project Solution
- 1.6 Propose Project Approach
- 1.7 Identify Project Scope
- 1.8 Conduct Literature Review
 - 1.8.1 Review of Technique for Route Optimization
 - 1.8.2 Review on Solving Optimisation Problem Using Reinforcement Learning
 - 1.8.3 Review on Solving Optimisation Problem Using Genetic Algorithm
 - 1.8.4 Review Google Maps API
 - 1.8.5 Review Performance Evaluation Based On Estimated Time of Arrival
- 1.9 Define Methodology
 - 1.9.1 Describe Research Methodology
 - 1.9.2 Decide Research Tools and Technology
 - 1.9.3 Prepare Work Plan

1.9.3.1 Create Work Breakdown Structure (WBS)

1.9.3.2 Create Gantt Chart

2.0 Data Handling

2.1 Data Preparation

2.2 Data Pre-processing

2.3 Data Transformation

3.0 Machine Learning

3.1 Prepare Datasets

3.2 Reinforcement Learning

3.2.1 Formulate Problems with Markov Decision Process (MDP)

3.2.2 Conduct Planning Phase

3.2.2.1 Environment Setup

3.2.3 Apply Deep Q-Network Model

3.2.3.1 Design

3.2.3.2 Implementation

3.2.3.3 Train model

3.2.3.4 Evaluation

3.3 Genetic Algorithm

3.3.1 Apply Fitness Value Calculation

3.3.2 Define Initial Population

3.3.3 Apply Random Parent Selection

3.3.4 Apply Partially Mapped Crossover

3.3.5 Apply Adaptive Mutation Technique

3.3.6 Repeat until Maximum Iteration is Reached

3.3.7 Terminate the Genetic Algorithm

4.0 Evaluation

4.1 Run Both Machine Learning on Testing Dataset

4.2 Evaluate Performance of Both Methods

4.2.1 Compare the Performance in terms of Total ETA

4.2.2 Identify Strengths and Weaknesses of Each Method

4.2.3 Provide Potential Improvements for Each Method

5.0 Closing

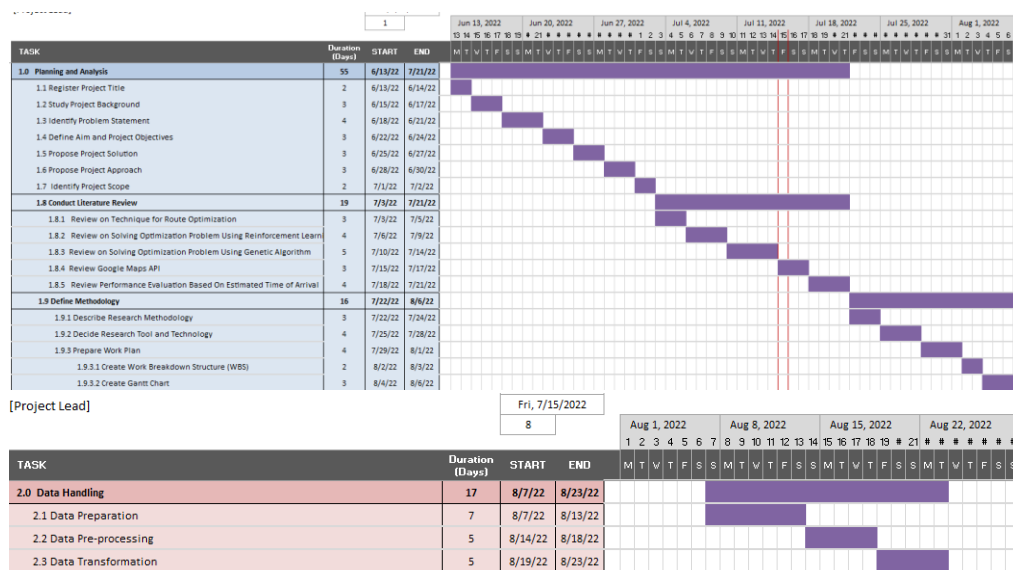
5.1 Complete Documentation

5.2 Prepare Presentation Slides

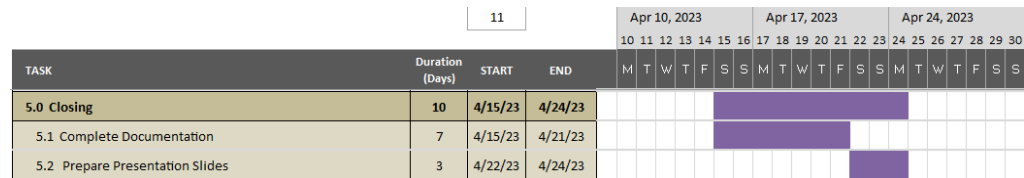
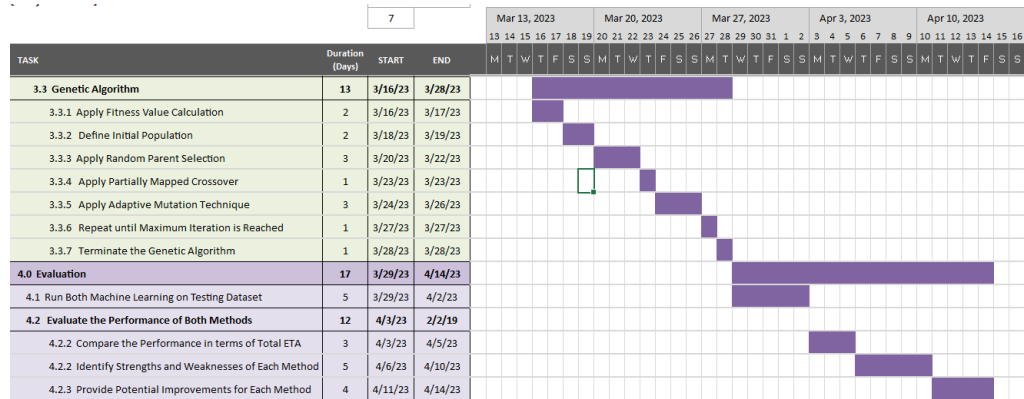
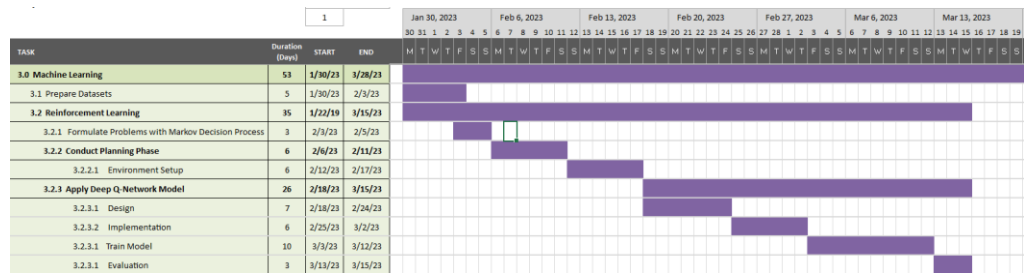
3.5 Gantt Chart

Gantt Chart is used for project scheduling and resource allocation. It is a graphical representation of the project's progress. The Gantt Chart for this project is attached below.

3.5.1 Gantt Chart for FYP1



3.5.2 Gantt Chart for FYP2



CHAPTER 4

RESULTS AND DISCUSSION

4.1 Datasets Used

The details of the datasets used in this project have been clearly described in Chapter 3 (Section 3.2.5). For evaluation purposes, both RL and GA are applied for the testing dataset with 16 nodes. Figure 4.1 demonstrates the initial route of the testing datasets on Google Maps without implementing the route optimisation technique. The starting point is marked in red.

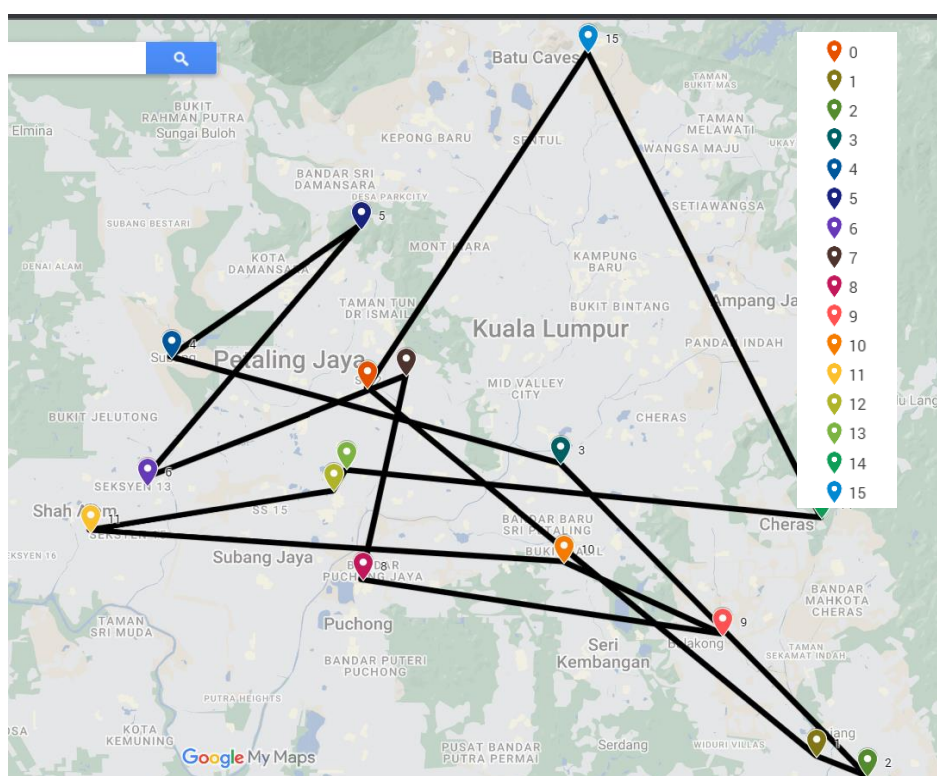


Figure 4.1 Initial Route without Route Optimization Technique

4.2 Evaluation Criteria

According to the objectives mentioned above, this paper aims to generate an optimal route while considering traffic congestion and precedence constraint stated in Chapter 3 (Section 3.2.5). The ETA matrix contains the traffic data in advance. Hence, this project assumes that the shorter the ‘total ETA’, the lower the courier cost, and the higher the customer satisfaction. ‘Total ETA’ refers to the minimum estimated time required for a courier to visit all the customers and return to the starting point.

4.3 Experiments

After conducted literature review in Chapter 2, RL (with DQN) and GA (with Adaptive Mutation Technique, Partially Mapped Crossover, and Elitism Technique) are implemented to solve the TSPPD proposed in Chapter 1. The solution process curve and running time are captured. To ensure impartiality in the experiments, both methods are fine-tuned with different combinations of hyperparameters. Lastly, the optimal solutions are recorded.

4.3.1. Experiment on RL

Figure 4.2 presents the learning curve of the DQN agent. It shows the progression of the agent’s performance over multiple episodes after interacting with the environment. At the beginning of the training process, the DQN’s scores are mostly negative, indicating that it does not learn the optimal Q-values for the actions in each state. This is expected as the agent has not accumulated sufficient experience to determine the optimal policy yet.

As the DQN’s agent interacts with the environment and receives feedback in the form of rewards and penalties throughout the training process, its Q-values are iteratively updated. It will begin to select the action with higher rewards. This is reflected in the learning curve, which shows an uplift in the scores from episode 300 onwards. The DQN’s agent started learning effective policies and is progressing toward finding the optimal route. In addition, a replay buffer is adopted to enhance the learning process and prevent the agent from getting stuck in local minima. Thus, the agent can learn from the previous

experience, even those with fewer rewards. This helps to improve the agent's performance and speed up the learning progress.

Furthermore, fluctuations in the learning curve indicate that the DQN is fine-tuning its policy and exploring different possible solutions. This is important for RL as it ensures the agent does not overfit to a particular set of states and actions. In contrast, the agent is exploring a wide range of possible solutions to find the optimal policy.

The DQN model is assumed well-trained as the plateau is achieved in the learning curve. A plateau indicates that the agent has learned an effective policy and consistently obtains a high score over time. Nevertheless, judging whether the DQN model has a good enough convergence is difficult, as it could perform better with minimal fluctuations.

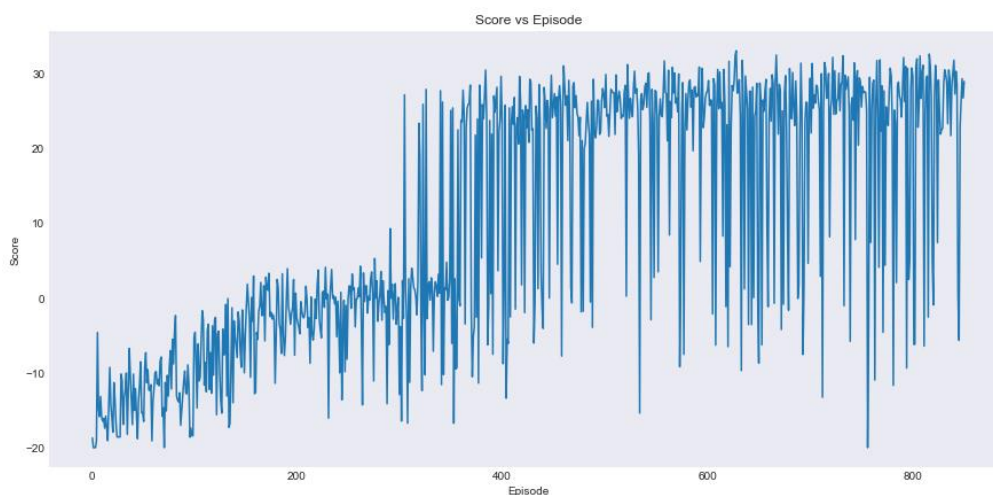


Figure 4.2 Learning Curve of DQN

4.3.2. Experiment on GA

The GA's evolution is visualised in Figure 4.3 below, showing the changes in the best fitness score over the generations. The fitness function is designed to be the total ETA of a travelling tour. Hence, the lesser the fitness value, the better. Besides, the graph is trending downward throughout the evolution, indicating that the GA continually improves and optimises the solutions. The rapid decrease of the fitness value in the early generation in the graph proved that this algorithm could explore the search space effectively and find the best solutions efficiently.

One key factor contributing to the GA's effectiveness is the implementation of the elitism technique. It helps retain the marvellous individual in each generation and directly transfers them to the next. Thus, premature convergence can be successfully avoided. This is because excellent individuals are allowed to continue to evolve, rather than being replaced by new individuals with potentially less fitness values. Besides, the crossover and mutation techniques in GA are introduced to further balance the exploration and exploitation trade-off. In other words, the current good individuals have pertained while trying to produce better individuals at the same time. The algorithms will have the chance to improve the quality of the overall solution and prevent the algorithm from getting stuck in the local optima.

The graph also indicates a good convergence speed and clear convergence toward the optimal solution. It can be seen from a little fluctuation in the fitness value before 150 generations, suggesting the GA is exploring various solutions in the initial stages. This is a desirable behaviour as it leads to a more thorough exploration of the search space. After 150 generations, the GA's solutions become more stable and optimised without fluctuation in the fitness value. Hence, it can be assumed that the GA has found an optimal solution through the optimisation process. However, there is still a possibility it might get stuck in local optima, where it cannot find a better solution.

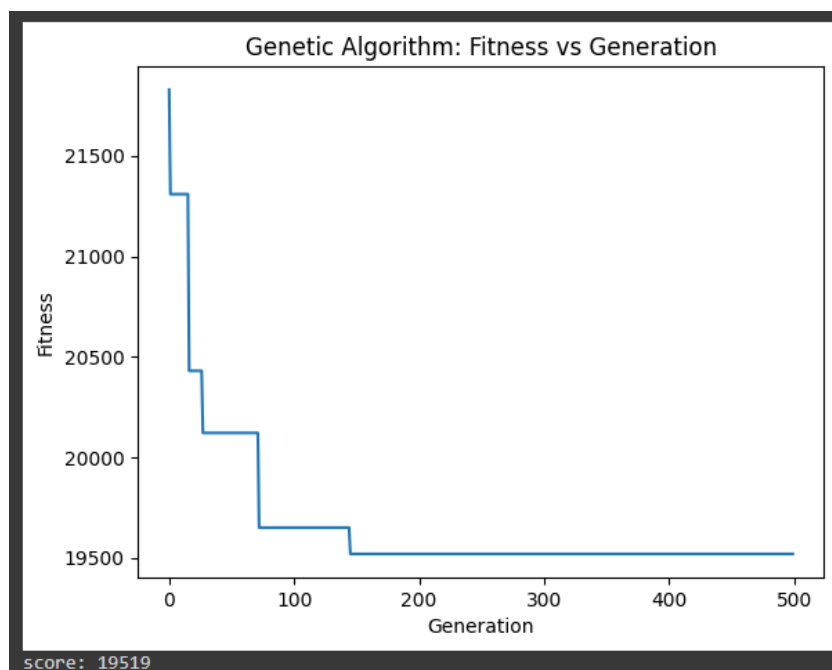


Figure 4.3 A Graph of Fitness Function (Best Fitness) Convergence of GA

4.4 Compare RL and GA Results

After conducting experiments with both GA and RL, it is found that both can satisfy the evaluation criteria stated above in Section 4.2. The experimental results are shown in Table 4.1. Using the testing dataset, the total ETA computed by RL and GA are 26,889 seconds and 19,519 seconds, respectively. The generated optimal routes are visualised on Google Maps in Figure 4.4 and Figure 4.5 for better interpretation. Besides, both RL and GA manage to fulfil the precedence constraint, indicating that node 3 is visited before node 2. The results show that GA can give a lower ETA than RL, thus saving the courier cost and improving customer satisfaction. Regarding the running time, RL requires a longer time to train than GA. However, it needs a relatively shorter time to generate optimal solutions when applying the trained model to a new dataset.

Table 4.1: Experiment Results of RL and GA

Algorithm	Optimal Route	Total ETA (seconds)	Operation Time (seconds)
RL	[0, 9, 1, 13, 8, 7, 6, 11, 5, 3, 10, 14, 4, 15, 2, 12, 0]	26,889	Training: 30600 Testing: 44
GA	[0, 5, 15, 3, 14, 2, 10, 1, 9, 12, 4, 13, 8, 11, 6, 7, 0]	19,519	342

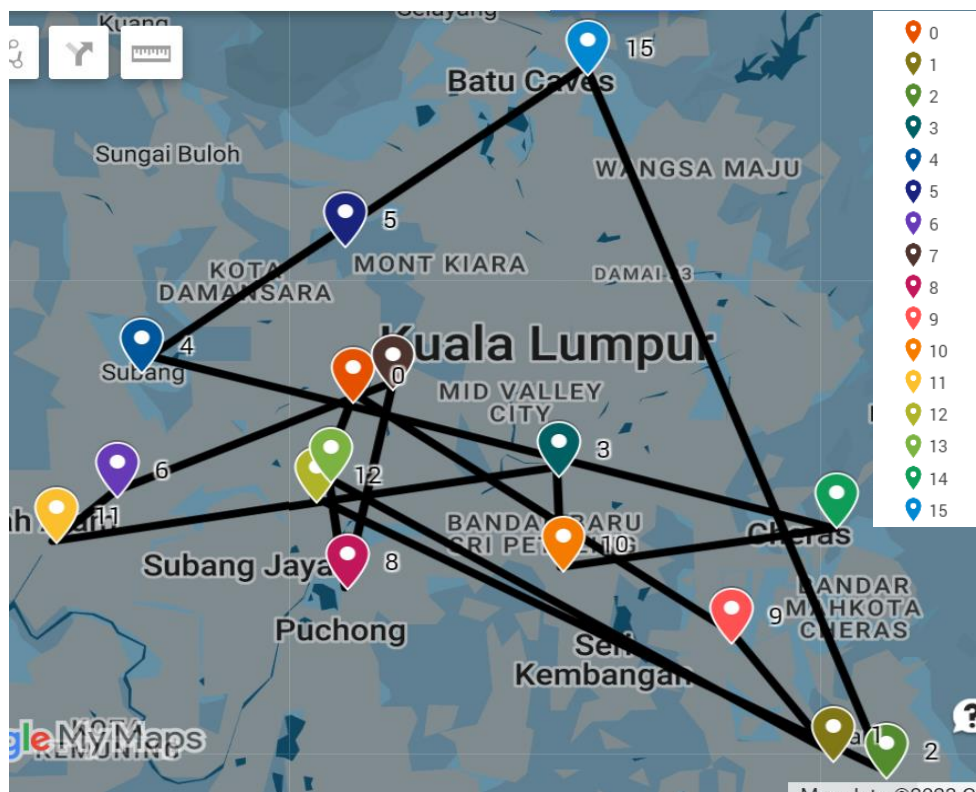


Figure 4.4 Optimal Route Provided by RL

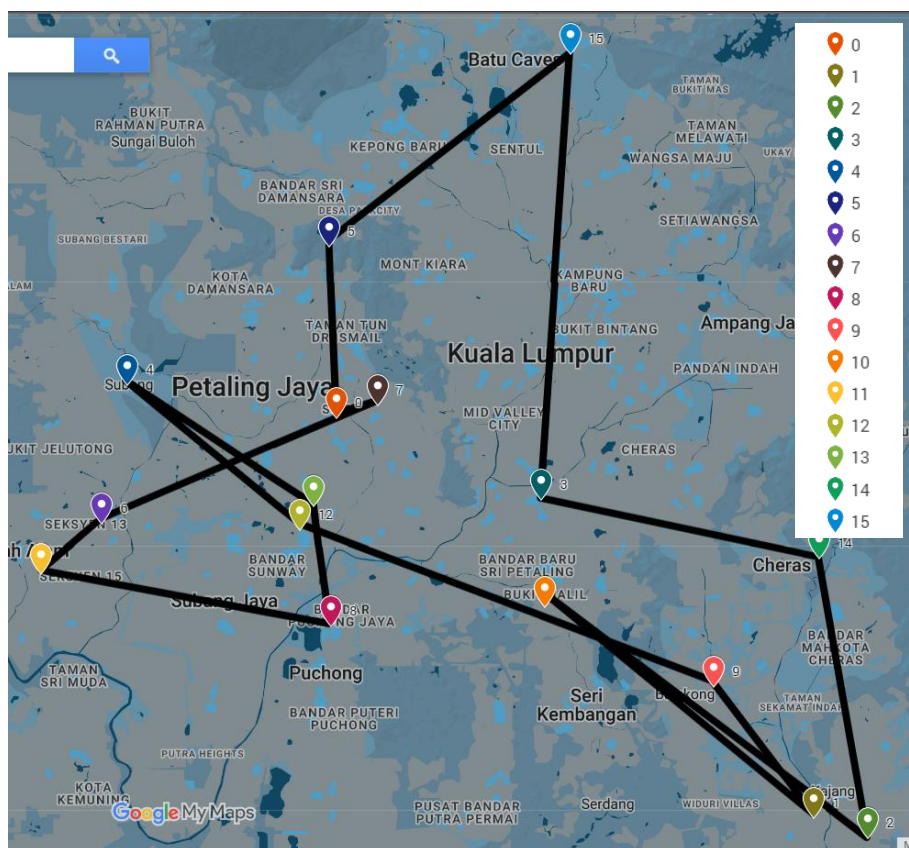


Figure 4.5 Optimal Route Provided by GA

4.5 Discussion

One significant advantage of GA and RL is they do not require any predetermined solutions to the proposed problem. RL agent interacts with the environment iteratively and learns the quality solution from the received reward signal after each action. In contrast, GA converges to an optimal solution by generating an initial population randomly and evolving them over time. This makes them well-suited for solving complex problems like TSPPD, where the optimal or near-optimal solutions may not be observed in the early stage. Besides, both methods tend to avoid premature convergence and getting stuck in local minima. DQN with experience replay encourages the exploration of different solutions, and the target network helps to stabilise the learning process while preventing the DQN model from overfitting the training data. As for the GA, the elitism technique, partially mapped crossover, and adaptive mutation technique effectively preserved the good solutions while maintaining the diversity in the population.

From the perspective of operation speed, GA has a shorter computational time than RL while generating optimal solutions. This is because GA can perform parallel evaluations of different candidate solutions concurrently, whereas RL involves a sequential learning process. Although RL requires a long training time that incurs more cost, it is capable of generating the optimal route after the RL model is fully trained. The trained DQN model can produce an optimal route approximately eight times shorter than GA.

The conducted experiments outlined that the GA could outperform the RL regarding the solutions' quality. From the RL learning curve presented in Section 4.3.1, the DQN agent is assumed to converge to a local minimum rather than find a globally optimal solution. It may cause by the inadequate tuning of RL's hyperparameter and poor generalisation that resulted in overfitting during training (Kalakanti et al., 2019). Overfitting occurs when the agent performs well on the training data but fails to generalise well on new data. The reason for overfitting might be that the RL has been trained without any regularisation component and the chosen hyperparameter through manual tuning is infeasible.

Other than that, this result may be due to the deterministic nature of the problem in this project. In other words, GA is more suitable for the problem with a deterministic environment than RL. The deterministic environment refers to the environment with no uncertainty or randomness. This is because the GA's ultimate goal is to search for an optimal solution in the complex search space in the end. Hence, GA can effectively generate the optimal route with the population-based approach. Unlike GA, RL more focus on action selection in the dynamic condition, learning policies, and the intermediate rewards obtained from each step taken, rather than prioritising the final reward only. However, it is undeniable that RL has the advantage of adapting to the stochastic environment, which is impossible for GA.

In terms of policy learning, GA does not learn the policies in the same way that RL does. Unlike RL, GA directly encodes the precedence constraint in the fitness function, allowing the algorithm to easily prioritise the feasible

solutions over the infeasible ones. Thus, it may not be suitable for problems that require dynamic adaption or exploration of new strategies. Conversely, RL involves a separate implementation of precedence constraints and a more intricate reward function that incorporates penalties for violating the constraint. This can increase the problem's complexity and the challenge of developing an effective RL algorithm.

In brief, both RL and GA have their strengths and weaknesses in solving the proposed problem. Table 4.2 summarises the strength and limitations of RL and GA. Although RL does not perform well in the experiment conducted above, it is worth identifying potential improvements in the related future work.

Table 4:2 Comparison between RL and GA

Algorithm	Strengths	Limitations
RL	<ul style="list-style-type: none"> • Can learn multiple policies for decision-making in sequential problems • Capable of adapting to stochastic environment • Contains experience replay and target network to improve the stability and efficiency of the learning process. • Requires less testing time with the trained DQN model 	<ul style="list-style-type: none"> • Long training time • Tends to converge to local optimal instead of global optima due to incorrect selection of hyperparameter • Requires complex reward function design and policy implementation
GA	<ul style="list-style-type: none"> • Fast computational time • Effective in finding globally optimal solution after being improved with 	<ul style="list-style-type: none"> • Incapable of adapting to changing environment • Does not learn the policies

	<p>elitism technique, Partially Mapped Crossover and adaptive mutation technique.</p> <ul style="list-style-type: none">• Relatively easier fitness function design compare to RL• Suitable for problems without dynamic changes	
--	---	--

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

This paper proposes RL and GA frameworks for the Travelling Salesman Problem with pickup and delivery. In short, this paper's contribution relative to the latest study can be summarised as follow: (i) Formulation of TSPPD for logistic application, (ii) Implementing RL and GA to solve the TSPPD problem. (iii) Prepare instances based on real-world traffic data from Google Maps-Distance Matrix API (iv) Examine the effectiveness of both RL and GA methods

Besides, it is significant to emphasise that all objectives in this project are met. The produced optimal routes are visualised in Google Maps for better understanding. The result shows that GA is able to outperform RL by 27.41% in ETA. The reason for RL's poor performance might be due to the deterministic nature of the problem or insufficient adjustment of RL's hyperparameter that leads to the algorithm falling into local optima. Their opportunities and limitations have been clearly listed to provide a comprehensive understanding of the RL and GA methods in addressing the proposed problem in Chapter 4 (Section 4.5). It can be a reference for future researchers who wish to tackle similar problems. As for future work, real-time scheduling shall be included in the application development phase. Despite the fact that RL does not produce outstanding results in the conducted experiment, further research is still worthwhile for solving dynamic problems in real-world application development.

Potential Improvements and Future Works

In this project, GA has proven effective in solving the proposed TSSPD problem in Chapter 1. However, one limitation of GA and RL is they are both problem-specific. Hence, the same GA or RL algorithm that works well currently problem may not necessarily work well for another. Real-world problems

should consider more than 15 customers and multiple precedence constraints. The ability of the current GA to solve the proposed problem with large instances remains to be discovered. As the number of customers increases, the number of possible solutions grows exponentially, making the problem more complex. Thus, further research should be carried out.

Besides, real-time scheduling should be considered when it comes to web application development in future work. In this case, a hybrid approach combining GA and RL can be explored to solve such a problem. Zheng et al. (2022) had introduced a technique named the Reinforced Hybrid Genetic Algorithm to solve the TSP, which can be further modified for this purpose. For instance, the GA can be used to explore the search space and generate candidate solutions. At the same time, the RL can be employed to learn the optimal policies for selecting the best solution based on the dynamic state of the system.

Lastly, the performance of both GA and RL is highly dependent on their hyperparameter. In this project, the hyperparameter selection is done by choosing a range of hyperparameter values and training the model with different hyperparameter combinations. It is suggested that statistical methods such as ANOVA and Turkey Test should be selected to tune the RL parameters carefully. With the statistical methodology mentioned above, Ottoni et al. (2021) highlighted the tuning of 4 hyperparameters in RL: discount factor, learning rate, reinforcement function, and Epsilon-Greedy algorithm in solving TSP. Even with the optimal hyperparameters, the optimisation algorithms may produce a satisfactory solution instead of discovering the optimal solution. The reason is that the domain knowledge is unknown at the beginning of the cases. Thus, it is important to consider domain-specific knowledge and expertise when solving complex scheduling problems.

REFERENCES

- A. K. Kalakanti, S. Verma, T. Paul and T. Yoshida, (2019) "RL SolVeR Pro: Reinforcement Learning for Solving Vehicle Routing Problem," 2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS), Ipoh, Malaysia, pp. 94-99, doi: 10.1109/AiDAS47888.2019.8970890.
- Anantathanavit M., Munlin M. (2016). Using k-means radius particle swarm optimization for the travelling salesman problem. IETE Technical Review, 33(2), 172–180.
- B. Chen, H. Zhang and J. Du, (2022). "An Algorithm for Solving The Traveling Salesman Problem," IEEE/ACIS 22nd International Conference on Computer and Information Science (ICIS), Zhuhai, China, 2022, pp. 43-47, doi: 10.1109/ICIS54925.2022.9882346.
- Chen, C. and Ngwe, D. (2018) Shipping fees and product assortment in online retail. Available at: https://www.hbs.edu/ris/Publication%20Files/19-034_b2382177-a462-447e-86f8-690d1ea7af18.pdf
- CHEN, X. et al. (2019) A New Evolutionary Multiobjective Model for Traveling Salesman Problem, IEEE Xplore Full-text PDF: Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8718296> (Accessed: April 5, 2023).
- Choudhary, A. (2020) A hands-on introduction to deep Q-learning using openai gym in python, Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/> (Accessed: April 3, 2023).
- D. Cai, Y. Gao, and M. Yin, (2018) "NSGAI with local search based heavy perturbation for bi-objective weighted clique problem," IEEE Access, vol. 6, pp. 51253–51261, . doi: 10.1109/ACCESS.2018.2869732.

DÜNDAR, A. O. and ÖZTÜRK, R. (2020) “THE EFFECT OF ON-TIME DELIVERY ON CUSTOMER SATISFACTION AND LOYALTY IN CHANNEL INTEGRATION”, *Business & Management Studies: An International Journal*, 8(3), pp. 2675–2693.

ecommerceDB (no date) ECommerce market in Malaysia, ecommerceDB. Available at: <https://ecommercedb.com/markets/my/all> (Accessed: February 17, 2023).

Editoron (2019) Most Malaysian consumers unhappy in their e-commerce delivery experience across Southeast Asia, Borneo Post Online. Borneo Post Online. Available at: <https://www.theborneopost.com/2019/06/30/most-malaysian-consumers-unhappy-in-their-e-commerce-delivery-experience-across-southeast-asia/> (Accessed: February 17, 2023).

Fan, J. The vehicle routing problem with simultaneous pickup and delivery based on customer satisfaction. *Oper. Res. Manag. Sci.* 2011, 15, 5284–5289.

Filip, E. (1970) [PDF] the travelling salesman problem and its application in logistic practice: Semantic scholar, [PDF] The Travelling Salesman Problem and its Application in Logistic Practice | Semantic Scholar. Available at: <https://www.semanticscholar.org/paper/The-Travelling-Salesman-Problem-and-its-Application-Filip/a72e799885c5dbe39b9144d2484be37a20516e44> (Accessed: April 6, 2023).

Fu, C. et al. (2010) “The logistics network system based on the Google Maps API,” in 2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM). IEEE, pp. 1486–1489.

García-Albertos, P., Picornell, M., Salas-Olmedo, M., Gutiérrez, J. (2019). Exploring the potential of mobile phone records and online route planners

for dynamic accessibility analysis. *Transportation Research Part A: Policy and Practice*, 125, 294-307.

Global Data (2022). ShieldSquare Captcha. [online] Available at: <https://www.globaldata.com/media/banking/malaysia-e-commerce-market-grow-19-9-2022-estimates-globaldata/#:~:text=GlobalData%27s%20E%2DCommerce%20Analytics%20reveals.>

GlobalData UK Ltd. (2022) Malaysia e-commerce market to grow by 19.9% in 2022, estimates GlobalData, GlobalData. GlobalData UK Ltd. Available at: <https://www.globaldata.com/media/banking/malaysia-e-commerce-market-grow-19-9-2022-estimates-globaldata/#:~:text=GlobalData's%20E%2DCommerce%20Analytics%20reveals,22.4%25%20between%202017%20%E2%80%93%202021> (Accessed: February 17, 2023).

Ha, Q. M., Deville, Y., Pham, Q. D., & Hà, M. H. (2020). A hybrid genetic algorithm for the traveling salesman problem with drone. *Journal of Heuristics*, 26(2), 219-247.

Ha, Q.M., Deville, Y., Pham, Q.D. et al. (2020). A hybrid genetic algorithm for the traveling salesman problem with drone. *J Heuristics* 26, 219–247 . Available at: <https://doi.org/10.1007/s10732-019-09431-y>

Hameed, W.M. and Kanbar, A.B. (2017) A comparative study of crossover operators for genetic algorithms to solve travelling salesman problem, Zenodo. Available at: <https://zenodo.org/record/345734#.ZC1N2ntBw2w> (Accessed: April 5, 2023).

Han, S. and Xiao, L. (2022) An improved adaptive genetic algorithm, SHS Web of Conferences. EDP Sciences. Available at: https://www.shs-conferences.org/articles/shsconf/abs/2022/10/shsconf_iteme2022_01044/shsconf_iteme2022_01044.html (Accessed: April 5, 2023).

Hariyadi, Putri,M., Phong T. N , Iswanto. I, Dadang. S (2019) ‘Traveling Salesman Problem Solution using Genetic Algorithm’ Available at: <https://www.jcreview.com/admin/Uploads/Files/61a476421548d4.79397681.pdf> (Accessed: April 20, 2023).

Herdiana, I.K., Candiasa, I.M. and Indrawan, G. (2022). Optimization of adaptive genetic algorithm parameters in traveling salesman problem, *Journal of Computer Networks, Architecture and High Performance Computing*. Available at: <https://doi.org/10.47709/cnahpc.v4i2.1581> (Accessed: April 5, 2023).

I,Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, (2017) “Neural combinatorial optimization with reinforcement learning,” in 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, Workshop Track Proceedings, 20

Ibrahim, et al. (2021). An Improved Genetic Algorithm for Vehicle Routing Problem Pick-upand Delivery with Time Windows. *Jurnal Teknik Industri*, 22(1), 1-17.
doi:<https://doi.org/10.22219/JTIUMM.Vol22.No1.1-17>

International Trade Administration (2022) Malaysia - ecommerce, International Trade Administration | Trade.gov. International Trade Administration, U.S. Department of Commerce, . Available at: <https://www.trade.gov/country-commercial-guides/malaysia-ecommerce> (Accessed: February 17, 2023).

J. Pan, M. Huang, Q. Zhang and Y. Yu, (2020) "Dynamic Vehicle Routing Problem Considering Customer Satisfaction," 2020 39th Chinese Control Conference (CCC), Shenyang, China, 2, pp. 5602-5606, doi: 10.23919/CCC50068.2020.9188538.

J.Tang, Z. Pan ,RYK. Fung, H. Lau. Vehicle routing problem with fuzzy time windows. *fuzzy sets and systems*, 160(5): 683-695

- Jihene, K. & Youssef, H. (2019) Permutation rules and genetic algorithm to solve the traveling salesman problem, Arab Journal of Basic and Applied Sciences, 26:1, 283-291, DOI: 10.1080/25765299.2019.1615172
- Jinhui Wang.(2007) Recommender of Green Logistics Management for Chinese Enterprises. Hebei University of Technology (Social Science Edition), Vol.7, No.4.
- K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, (2002).“A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197.
- Kersten, W. & Koch, J. (2010). The effect of quality management on the service quality and business success of logistics service providers. International Journal of Quality & Reliability Management, 27(2), 185 – 200. Available at: <http://dx.doi.org/10.1108/02656711011014302>
- Kotu, V. and Deshpande, B. (2019) “Introduction,” in Data Science. Elsevier, pp. 1–18.
- ltd, R.and M. (2022) Same-day Delivery Services Global Market Report 2022 by , type, service type, mode of transportation, application, Research and Markets - Market Research Reports - Welcome. Available at: https://www.researchandmarkets.com/reports/5546259/same-day-delivery-services-global-market-report?utm_source=CI (Accessed: February 24, 2023).
- M. Barkaoui, J. Berger, A. Boukhtouta. Customer satisfaction in dynamic vehicle routing problem with time windows. Applied Soft Computing, 35: 423-432
- M. Nazari, A. Oroojlooy, L. Snyder, and M. Takac, (2018) “Reinforcement learning for solving the vehicle routing problem,” in Advances in Neural Information Processing Systems 31,pp. 9839–9849.

- Marbn, S., Mariscal, G. and Segovi, J. (2009) "A data mining & knowledge discovery process model," in *Data Mining and Knowledge Discovery in Real Life Applications*. I-Tech Education and Publishing.
- Miglani, P. et al. (2021) "Optimal metro route identification using Q-Learning," in *2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*. IEEE, pp. 698–702.
- Miki, S., Yamamoto, D. and Ebara, H. (2018) "Applying deep learning and reinforcement learning to traveling salesman problem," in *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. IEEE, pp. 65–70.
- Min, H. (1989), "The multiple vehicle routing problem with simultaneous delivery and pick-up points", *Transportation Research Part A General*, Vol. 23 No. 5, pp. 377-386.
- Muñoz-Villamizar, A. et al. (2021) "Study of urban-traffic congestion based on Google Maps API: the case of Boston," *IFAC-PapersOnLine*, 54(1), pp. 211–216. doi: 10.1016/j.ifacol.2021.08.079.
- Nambisan, P., Gustafson, D. H., Hawkins, R., and Pingree, S. (2016). Social support and responsiveness in online patient communities: impact on service quality perceptions. *Health Expect.* 19, 87–97. doi: 10.1111/hex.12332
- Nayem, M. A., Islam, M. M. and Yao, X. (2019) "Solving transit network design problem using many-objective evolutionary approach," *IEEE transactions on intelligent transportation systems: a publication of the IEEE Intelligent Transportation Systems Council*, 20(10), pp. 3952–3963. doi: 10.1109/tits.2018.2883511.
- Niu, Y. et al. (2018) "Optimising the green open vehicle routing problem with time windows by minimising comprehensive routing cost," *Journal of cleaner production*, 171, pp. 962–971. doi: 10.1016/j.jclepro.2017.10.001.

- Otoni, A.L. et al. (2021) "Reinforcement learning for the traveling salesman problem with refueling," *Complex & Intelligent Systems*, 8(3), pp. 2001–2015. Available at: <https://doi.org/10.1007/s40747-021-00444-4>.
- Paragon transforms transport planning for Frozen Food Express (no date) Paragon Routing. Available at: <https://www.paragonrouting.com/en-us/press-releases/post/paragon-transforms-transportation-planning-frozen-food-express-2/> (Accessed: February 18, 2023).
- Parasuraman, A., Zeithaml, V. A. & Berry, L. L. (1985), A Conceptual Model of ServiceQuality and Its Implications for Future Research. *Journal of Marketing*, 49(4), 41 – 50. Available at:<http://dx.doi.org/10.2307/1251430>
- Purusotham, S., Jayanth, T., Vimala, T & Ghanshyam, K. (2022). An efficient hybrid genetic algorithm for solving truncated travelling salesman problem. *Decision Science Letters* , 11(4), 473-484.
- R. Zhang, A. Prokhorchuk and J. Dauwels, (2020) "Deep Reinforcement Learning for Traveling Salesman Problem with Time Windows and Rejections," 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9207026.
- Riazi, A. (2019). Genetic algorithm and a double-chromosome implementation to the traveling salesman problem. *SN Applied Sciences*, 1(11). doi:<https://doi.org/10.1007/s42452-019-1469-1>.
- S. Wang, S. Ali, T. Yue, and M. Liaen, (2018). "Integrating weight assignment strategies with NSGA-II for supporting user preference multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 378–393.
- S., A.P., Vashisht, V. and Choudhury, T. (2013) Comparison of various mutation operators of genetic algorithm ... - *IJERT*. Available at: <https://www.ijert.org/research/comparison-of-various-mutation->

operators-of-genetic-algorithm-to-resolve-travelling-salesman-problem-IJERTV2IS60404.pdf (Accessed: April 5, 2023).

Spiceworks. (2022). What Is Reinforcement Learning? Working, Algorithms, and Uses. [online] Available at: <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-reinforcement-learning/>.

Ullah, Zaib, Fadi Al-Turjman, Leonardo Mostarda, and Roberto Gagliardi. (2020). "Applications of Artificial Intelligence and Machine Learning in Smart Cities." *Computer Communications*, 154, 313–23.

W. Kool, H. van Hoof, and M. Welling, (2019) "Attention, learn to solve routing problems!" in *International Conference on Learning Representations*.

Wang Y., Han Z. (2021). Ant colony optimization for traveling salesman problem based on parameters optimization. *Applied Soft Computing*, 107, 107439.

Wang, C., Ye, Z. and Wang, W. (2020) "A multi-objective optimisation and hybrid heuristic approach for urban bus route network design," *IEEE access: practical innovations, open solutions*, 8, pp. 12154–12167. doi: 10.1109/access.2020.2966008.

Wang, J. (2016), "Multi-objective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances, and algorithms", *IEEE Transactions on Cybernetics*, Vol. 46 No. 3, pp. 582-594.

X. Bai, M. Cao, W. Yan, S. S. Ge and X. Zhang, "Efficient Heuristic Algorithms for Single-Vehicle Task Planning With Precedence Constraints," in *IEEE Transactions on Cybernetics*, vol. 51, no. 12, pp. 6274-6283, Dec. 2021, doi: 10.1109/TCYB.2020.2974832.

Xing, E. and Cai, B. (2020) "Delivery route optimisation based on deep reinforcement learning," in *2020 2nd International Conference on*

Machine Learning, Big Data and Business Intelligence (MLBDBI). IEEE, pp. 334–338.

Y. Yuan, Y.-S. Ong, A. Gupta, and H. Xu,(2018). “Objective reduction in many-objective optimization: Evolutionary multiobjective approaches and comprehensive analysis,” IEEE Trans. Evol. Comput., vol. 22, no. 2, pp. 189–210.

Zhang, Z. et al. (2022) “Meta-learning-based deep reinforcement learning for multi-objective optimisation problems," IEEE transactions on neural networks and learning systems, PP, pp. 1–14. doi: 10.1109/TNNLS.2022.3148435.

Zheng, J., Zhong, J., Chen, M. and He, K. (2022). Reinforced Hybrid Genetic Algorithm for the Traveling Salesman Problem. arXiv:2107.06870 [cs]. [online] Available at: <https://arxiv.org/abs/2107.06870>.

This page intentionally left blank.