

UNIVERSITY OF BAHRAIN

College of Information Technology



A Hybrid Approach Based on Machine Learning and Optimization Techniques for Waste Management Collection Using Electric Vehicles

A Thesis Submitted in Partial Fulfilment of the Requirements for the
Doctor of Philosophy Degree in Computing and Information Sciences

Submitted By

Shaima Hejres

19990190

Supervised By

Prof. Nabil Hewahi

(Professor)

Co-Supervisor

Dr. Amine Mahjoub

(Assistant Professor)

University of Bahrain

January 2026



Dissertation Deposit Form

Form Code: UDNGS025

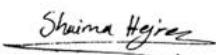
Author Name:	Shaima Ebrahim Mohamed Juman Hejres
Thesis Title:	A Hybrid Approach Based on Machine Learning and Optimization Techniques for Waste Management Collection Using Electric Vehicles
Year of Publication:	2025
Department:	Computer Science
College:	Information Technology
E-mail:	19990190@stu.uob.edu.bh
Contact No.	36434380

Author's Declaration: I agree to the following Conditions:

The above thesis will be available (in the University of Bahrain Library and externally) and reproduced as necessary at the discretion of the University of Bahrain. It may also be digitized by the University of Bahrain and made available via the university's repository or via other parties on the Internet.

This condition shall apply to all copies including electronic copies.

The above thesis has been provided on the understanding that it is copyright protected material and that no quotation from this thesis may be published without paper acknowledgement.

Sign. of the Author:		Date:	17 - Aug-2025
----------------------	---	-------	---------------

DECLARATION

I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it currently submitted for any degree.

Signed: Shaima Hejres Date: 25/6/2025
SHAIMA EBRAIMA MOHAMED HEJRES (Candidate)

The Thesis defense committee considers the thesis titled:

A Hybrid Approach Based on Machine Learning and Optimization Techniques for Waste Management Using Electric Vehicles

Submitted by SHAIMA EBRAIMA MOHAMED HEJRES to the College of Information Technology is satisfactory and acceptable for the Doctor of Philosophy in Computing and Information Sciences

The Defense Committee:

Nabil Hewahi
Prof. Nabil Hewahi
(Supervisor) Department of Computer Science
University of Bahrain

Amine Mahjoub
Dr. Amine Mahjoub
(co-supervisor) Department of Computer Science
University of Bahrain

Suresh Sankaranarayanan
Prof. Suresh Sankaranarayanan
(Member) King Faisal University
Saudi Arabia

Faisal Alkhateeb
Prof. Faisal Alkhateeb
(Member) Department of Computer Science
University of Bahrain

ABSTRACT

In this era, the growing demand for Electric Vehicles (EVs) depends on the high integration of this technology in many areas. Thus, an important area of research raises interest in finding the optimal path-planning solution for EVs. However, the Waste Management System (WMS) is an important economic sector that helps to support health and improve the quality of life in smart cities. Hence, this thesis discusses research for EV; specifically, Unmanned Aerial Vehicle (UAV), routing techniques to assist researchers in choosing the appropriate approach for developing a system based on optimization techniques, machine learning, or hybrid approaches in managing a waste collection system (WCS). Hybrid approach techniques that combine optimization/genetic algorithm (GA) and machine learning/reinforcement learning (RL) techniques can contribute to the production of effective solutions to problems for WCS, which is a vital part of WMS, by increasing the robustness in solving UAV routing problems. In this thesis, we address the problem of waste collection using UAVs, with the primary objective of minimizing both the collection time and energy consumption. Three different approaches are proposed. The first two focus on a single UAV, while the third involves the coordination of multiple drones. The first approach employs a multi-objective genetic algorithm. The second approach integrates GA with RL in a hybrid model. The third approach combines GA, RL, k-means clustering, and the A* algorithm to coordinate multiple UAVs. Across all approaches, several operational constraints are taken into account, including UAV capacity, flight time, and others. The multiobjective GA approach achieved a significant error reduction compared to previous research using the same experimental parameters: more than 81% for 100 instances, more than 59% for 500 instances, and over 44% for 1,000 instances. The second approach, which applies RL following genetic optimization, further improved performance by an average of 14.35% compared to the GA only approach. For the multi-UAV strategy (third approach), two scenarios were developed: one with UAV reassignment and one without. The results showed that the UAV reassignment significantly improved performance, reducing flight time by 32.11%, energy consumption by 63.12%, and the total distance traveled by 31.14% compared to the scenario without reassignment.

Contents

APPROVAL	A
DECLARATION	B
ABSTRACT	C
TABLE OF CONTENTS	D
LIST OF FIGURES	H
LIST OF TABLES	K
ACKNOWLEDGMENTS	M
DEDICATION	N
PUBLICATIONS	O
LIST OF ABBREVIATIONS	P
1 Introduction	1
1.1 Overview	1
1.1.1 Unmanned Aerial Vehicle for Waste Management System Using Genetic Algorithms	5
1.1.2 A Hybrid Approach of Genetic Algorithms and Re- inforcement Learning for Waste Collection Using Unmanned Aerial Vehicle	6
1.1.3 Multiple UAVs for Waste Management Using a Hy- brid Optimization and Machine Learning Approach	7
1.2 Research Problem	8
1.3 Research Objectives	8

1.4	Contribution to the Field	9
1.5	Research Questions	10
1.6	Thesis Outline	11
2	Literature Review	12
2.1	Literature Review	12
2.2	Routing Approaches Used for EVs System	13
2.2.1	Optimizations Techniques	13
2.2.1.1	Optimization Approaches in UAV	13
2.2.1.2	Optimizations Approaches in UGV	17
2.2.2	Machine Learning Approaches	33
2.2.2.1	Machine Learning Approaches in UAV	33
2.2.2.2	Machine Learning Approaches in UGV	35
2.2.3	Hybrid Approaches	44
2.2.3.1	Hybrid Approaches for UAV	44
2.2.3.2	Hybrid Approaches for UGV	46
2.3	The UAV Methods Used in the Routing Problem	58
2.3.1	UAV Individual Method used in Routing Problem	58
2.3.1.1	Genetic algorithm	58
2.3.1.2	Reinforcement Algorithm	62
2.3.2	UAV Hybrid Methods Used in Routing Problem	63
2.3.3	Approaches for Multiple Drones	66
2.3.4	Summary	71
3	Unmanned Aerial Vehicle for Waste Management System Using Genetic Algorithms	74
3.1	Introduction	74
3.2	Problem Definition and Description	75
3.3	The proposed Model	78
3.3.1	Genetic Algorithms	78

3.3.1.1	Initial Population	79
3.3.1.2	Fitness Function	80
3.3.1.3	Genetic Operators	88
3.3.1.4	Termination of GA	94
3.4	Experimental Results and Analysis	98
3.5	Summary	119

4 A Hybrid Approach of Genetic Algorithms and Reinforcement Learning for Waste Collection Using Unmanned Aerial Vehicle 123

4.1	Introduction	123
4.2	Proposed Solution	124
4.2.1	Genetic Algorithms	124
4.2.2	The Reinforcement Algorithm	125
4.2.2.1	Genetic Algorithm followed by Reinforcement Learning	126
4.2.2.2	Reinforcement Learning followed by Genetic Algorithm	133
4.3	Experimentations Results and Results Analysis	134
4.4	Summary	150

5 Multiple UAVs for Waste Management Using a Hybrid Optimization and Machine Learning Approach 152

5.1	Introduction	152
5.2	Problem Definition and Description	153
5.3	Proposed Solution	154
5.3.1	Genetic Algorithms	156
5.3.2	K-means Algorithm	157
5.3.3	A-star (A*) Algorithm	160
5.3.4	Reinforcement Learning Algorithm	162
5.3.5	The Dynamic Task Reassignment Approach	163

5.4	Experimental Results and Results Analysis	165
5.5	Summary	178
6	Conclusion and Recommendations	181
	References	188
	Abstract (in Arabic)	

List of Figures

3.1	Initial Population.	80
3.2	The Definitions of the Tour, Route, and Edge.	80
3.3	The Example of a Tournament in WMS.	89
3.4	Example of P1.	91
3.5	The Representation of P2.	91
3.6	The Subset Randomly Located from both P1 and P2 to Re- produce Offspring.	91
3.7	The Full Remaining WCLs as the Rule of the Crossover Order.	92
3.8	Example of Exchange Mutation.	93
3.9	Example of Local Search Space (Swap-based Hill Climbing Local Search).	94
3.10	GA System Model.	97
3.11	A Partial Example of Generated Data.	99
3.12	Example of a Typical Waste Disposal and Recharging Case.	102
3.13	Example of a Typical Waste Disposal Case.	102
3.14	Example of a Typical Recharging Case.	103
3.15	Example of a Typical Recharging and Waste Disposal Case.	103
3.16	WCS Diagrams for 100 WCLs Show the Relationships Be- tween Fitness Constraints and the Number of Routes.	105
3.17	WCS Diagrams for 500 WCLs Show the Relationships Be- tween Fitness Constraints and the Number of Routes.	105

3.18	WCS Diagrams for 1000 WCLs Show the Relationships Between Fitness Constraints and the Number of Routes.	106
3.19	Comparison of the Best Fitness Values Between Different WCLs.	106
3.20	Comparison of the Execution Time Between Different WCLs.	106
3.21	The Standard Deviations Diagram for a WMS with 100 WCLs.	107
3.22	The Standard Deviations Diagram for a WMS with 500 WCLs.	107
3.23	The Standard Deviations Diagram for a WMS with 1000 WCLs.	108
3.24	Percentage Reduction in Fitness Error of GA Compared with the First Comparative Research of Hybrid ACO.	115
3.25	Percentage Reduction in Fitness Error of GA Compared with the Second Comparative Research of MWMTT.	117
3.26	Percentage Reduction in Fitness Error of GA Compared with Both Comparative Researches in the Three Different Datasets.	119
4.1	Proposed Solutions with Two Hybrid Approaches.	124
4.2	The RL Process in the Model of Incorporating the GA with the RL (GARL)	127
4.3	The First Method of Incorporating the GA with the RL (GARL).	128
4.4	The RL Process in the Second Model That Incorporates the RL First then the GA (RLGA)	134
4.5	The Second Model That Incorporates the RL First, then the GA (RLGA).	137
4.6	Best Fitness Values.	140
4.7	Flight Time Values Among All Four Experiments.	140

4.8	Improvement Percentage Among All Algorithms Compared to the Individual Algorithm Performance (GA and RL). . . .	141
4.9	The Percentage Reduction in Fitness Error That Was Compared Between the Proposed Hybrid Approaches with Hybrid ACO.	149
4.10	The Percentage Reduction in Fitness Error That Was Compared Between the Proposed Hybrid Approaches with MWMTT.	149
5.1	The Proposed Hybrid Model for Multiple UAVs.	158
5.2	The Clusters of the Multiple Drones.	165
5.3	The Dynamic Task Reassignment Approach.	167
5.4	The Improvements from Before to After the Dynamic Reassignment Techniques.	175
5.5	The Improvements of Results Before Reassignment Compared to the Benchmark Single-Drone Model.	176
5.6	The Improvements of Results After Dynamic Reassignment Techniques Compared to the Benchmark Single Drone Model.	176

List of Tables

2.1	Summaries of Optimization Approaches	24
2.2	Summaries of Machine Learning Approaches	40
2.3	Summaries of Hybrid Approaches	50
2.4	Pros and Cons of Some Main Methods	54
3.1	The Mathematical Model Notation	77
3.2	The Mathematical Model Notation	96
3.3	Experiments Parameters.	100
3.4	Experimental Results.	101
3.5	The Comparison Between the First Research (Hybrid ACO) and Thesis Approach Within GA.	120
3.6	The Comparison Between the Second Research MWMTT and Thesis Approach Within GA.	121
4.1	RL Parameters in Experiments	138
4.2	Experiments Results	139
4.3	The Experiment's Percentage of Improvement	142
4.4	Experience Improvements Timing Result	142
4.5	The Comparison Between the Methods Research (hybrid ACO) and the Proposed Approaches Within GA.	147
4.6	The Comparison Between the Methods Research MWMTT and Proposed Approaches Within GA.	147
5.1	Experiments Parameters for RL Phase.	169

5.2	The Performance Metrics.	171
5.3	The Experimental Results Showing Percentage Improvements Between Before and After the Dynamic Reassignment Tech- niques.	174
5.4	Comparing the Three Experiment Results.	178

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to everyone who supported and assisted me throughout the process of completing this thesis.

DEDICATION

All praise, grace, and thanks be to You, O Lord, for granting me the ability and inspiring me with knowledge to reach this point, where I am completing a project, after which I will begin another one, with Your permission and will. My father, how I wished you could reach this stage with me where I enter with an academic title of 'Doctor', and how your happiness exceeded my own when I obtained my master's degree a few years ago in this prestigious institution... I miss your presence and pray to my Lord every day to place you in the highest paradise among the high with the prophets, the truthful, the martyrs, and the righteous... May God have mercy on you abundantly. My dear mother, words are not enough to tell you that you are everything to me... Here I stand thanks to you, and from here I continue to thrive with your support and strength... Thank you beyond what I can express. The soul that is within me, my dear husband Nawaf Al-Ghanem, your presence by my side is enough for me to be who I am... Thank you for everything. My dear aunt Dr. Mona Hejres... I was overwhelmed by your support for me until I thought that what I do is for you, not for myself. Thank you from the depths of my heart for all the support, review, advice, and observations you have provided. Professor Nabil Al-Huwahi, my supervisor, whom I saw in his severity as mercy and in his precision as refinement, enrichment, and development for what I have. Just as gold is polished and purified, you were the trustworthy hand that moved me from one state to another. Without your guidance, observations, and corrections, I would not have reached the end of this road... So, thank you as much as words cannot express. Thank you, Dr. Amin Mahgoub, for accepting me as your student. You have been an invaluable asset and benefit to me.

PUBLICATIONS

This thesis is based on previously published papers written jointly with my Supervisor, Prof. Nabil Hewahi, and my Co-Supervisor, Dr.Amine Mahjoub:

JOURNAL PAPERS

1. (Published) Hejres, S., Mahjoub, A. & Hewahi, N. (2024). *Routing approaches used for electrical vehicles navigation: A survey*. International Journal of Computing and Digital Systems, 15(1), 801–819.
URL: https://www.researchgate.net/publication/390328616_Routing_Approaches_used_for_Electrical_Vehicles_Navigation_A_Survey
2. (Published) Hejres, S., Mahjoub, A. & Hewahi, N. (2025). *Unmanned aerial vehicle for waste management system using genetic algorithms*. Arab Journal of Basic and Applied Sciences, 32(1), 57–80.
URL: <https://www.tandfonline.com/doi/full/10.1080/25765299.2025.2474798>
3. (Published) Hejres, S., Mahjoub, A & Hewahi, N. (2025). *A hybrid approach of genetic algorithms and reinforcement learning for waste collection using unmanned aerial vehicles *. IAENG International Journal of Computer Science, IJCS_250308Dz.
URL: https://www.iaeng.org/IJCS/issues_v52/issue_11/IJCS_52_11_42.pdf
4. (Submitted) Hejres, S., Hewahi, N., Mahjoub, A. & (2025). *Multiple UAVs For Waste Management Using a Hybrid Optimization and Machine Learning Approach*.

LIST OF ABBREVIATIONS

ACC	Accuracy
3BPP	Three-Dimensional Bin Packing Problems
3L-CVRP	3D Loading Capacitated Vehicle Routing Problem
A*	A-star search algorithm
ACVRP	Asymmetric Capacity Vehicle Routing Problem
AGA	Adaptive Genetic Algorithm
AGVs	Automated Guided Vehicles
AMP	Autonomous Motion Planning
AoI	Age of Information
AVI	Approximate Value Iteration
BHPF	Black Hole Potential Field
BMA	Bi-directional Martins Algorithm
CHs	Cluster Heads
CLP	Constraint Logic Programming
CNN	Convolutional Neural Network
CPs	Check Points
CVRP	Capacitated Vehicle Routing Problem

CVRPPAD Capacitated Vehicle Routing Problem with Pickup and Alternative Delivery

cA2C Contextual Advantage Actor-Critic

cDQN Contextual Deep QNetwork

DDS-F Drone Delivery Schedule model with drone Failures

DNN Deep Neural Networks

DP Disjoint Pricing

DRL Deep Reinforcement Learning

DRQN Deep Recurrent Q Network

DT Digital Twin

DTC Direct Torque Control

D3QN Dueling Double Deep Q-Networks

DDQN Dueling Deep Q-Network

DMPC Distributed Model Predictive Control

DWPT Dynamic Wireless Power Transfer

E-VRPTW Electric Vehicle Routing Problems with Time Windows and Charging Stations

EAPP Efficient AGV Path Planning

EC Energy Consumption

EHA Efficient Heuristic Allocation algorithm

ELOD	Expected Loss Of Demand
EV	Electronic Vehicles
FSTSP	Flying Companion Traveling Salesman Problem
FL	Fuzzy Algorithm
GACO	Gain-Based Ant Colony Optimization
GA	Genetic Algorithm
GARL	Genetic and Reinforcement Learning algorithm model
GCS	Ground Control Stations
GIS	Geographical Information System
GGAnt	Gain-based Green Ant
GPS	Global Positioning System
GTPAVP	Genetic Trajectory Planning Algorithm with Variable Population Size
GVD	Generalized Varying Domain
GWN	Gaussian White Noise
HO	Hierarchical Optimization
HBMO	Honeybees Mating Optimization
IEEE	IEEE Xplore database
IGA	Improved Multi-objective Genetic Algorithm
IoT	Internet of Things

IoTds	Internet of Things Devices
IRL	Inverse Reinforcement Learning
IRP	Inventory Routing Problem
ISE	Information Synergy Entropy
ISE-IITOPSIS	Interval Intuitionistic Fuzzy TOPSIS
KNN	K-Nearest Neighbor's
LB	Learning-Based methods
LOS	Line-Of-Sight
LQR	Linear Quadratic Regulator
LSTM	Long Short-Term Memory
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
MAPE	Mean Absolute Percentage Error
MCCPP	MultiCriteria Collaborative Path Planning
MCS	Mobile Crowd Sensing
MDP	Markov Decision Process
MEP	Maximum Entropy Principle
MEC	Mobile Edge Computing
MPWT	Maximum Waste in the Minimum Time during each Trip
MSA	Multiple Scenario Approach

MUTAPP	Multi-UAV Target Assignment and Path Planning
MVP	Multi Vehicle-assisted Multi UAV Path Planning Algorithm
MWMTT	Maximum Waste in the Minimum Time during each Trip
NGA	Neural Genetic Algorithm
NN	Neural Network
NP-Hard	Polynomial Non-Deterministic Hardness Problem
OVRP	Open Vehicle Steering Problem
PHEV	Plug-in Hybrid Electric Vehicles
PPO	Proximal Policy Optimization
PSO	Particle Swarm Optimization
RBFN-AFS	Radial Basis Function Network-Based Adaptive Fuzzy System
RF	Random Forest
RNN	Recurrent Neural Network
RL	Reinforcement Learning
RLGA	Reinforcement Learning and Genetic Algorithm Model
SA	Simulated Annealing
SCPSO	SineCosine Particle Swarm Optimization
SEAD	Suppression of Enemy Air Defense
SO	Social Optimum

SPs	Stopping Points
STAPP	Simultaneous Target Assignment and Path Planning
SVM	Support Vector Machine
SWM	Solid Waste Management
TD	Time Difference
TSP	Traveling Salesman Problem
2L-CVRP	Two-Dimensional Loading Vehicle Routing Problem
UE	User Equilibrium
UAV	Unmanned Aerial Vehicles
UGV	Unmanned Ground Vehicles
USV	Unmanned Surface Vehicle
V2G	Vehicle-to-Grid
VNS	Variable Neighborhood Search
VRP	Vehicle Route Problem
VRPMT	Vehicle Routing Problem with Multiple Trips
WCLs	Waste Collect Locations
WCM	Waste Collection Management
WCS	Waste Collection System
WCVRP	Waste Collection Vehicle Routing Problem
WSNs	Wireless Sensor Networks

Chapter 1

Introduction

1.1 Overview

In this era of artificial intelligence, there is a growing need to improve the waste collection system, a crucial component of the overall waste management system. WMS encompasses the entire waste life cycle, including waste collection systems and a variety of processes such as waste control, treatment, and disposal. This system is of great importance for the effective disposal of all types of waste, including solid, specialized, recyclable, organic, biological, and hazardous waste. WCS requires serving multiple objectives methods (Hou et al., 2021) addressing various constraints such as minimizing cost and path distance, conserving energy (Beigi et al., 2022), and maximizing capacity. This system relies on the logistical process of collecting waste from its locations, establishing a transportation network between these waste locations, and finally disposing of waste. The main infrastructure of the WCS includes waste vehicles, such as gas or electricity-powered garbage trucks. Many countries have introduced advanced technology to integrate electric vehicles with various AI methods. Japan, Luxembourg, Germany, Switzerland, Austria, New Zealand, the United States, the United Kingdom, Canada, South Korea, and Singapore have used AI methods in the waste management cycle industry (Fang et al., 2023). How-

ever, 40% of developing countries still do not have well-designed WMSs compared to 90% of developed countries.

EVs reduce greenhouse gas emissions that are emitted from traditional vehicles and reduce the effects of pollution. However, improving energy efficiency reduces the overall cost of the tour on the basis of energy-saving reactions through different methods. Some of these methods are microwaves, ultrasounds, and mechanochemical reactions, which are promising techniques that add high potential to energy-saving chemical reactions that can be combined with artificial intelligence to achieve more efficient and better environmentally sustainable industries. The UAV is used as a waste vehicle, ensuring that it effectively collects and connects all waste locations and disposes of them in a central landfill or a treatment or recycling station. The importance of qualifying the WCS (Kaabi et al., 2023) to assist, protect, and preserve the environment from pollution threats and mitigate the effects of climate change is evident. Hence, improved WCS provides an improved quality of life through the improvement of overall health by preventing the spread of rodents and diseases. In addition, it can be used to recycle and reuse materials. Hence, the overall importance of WCS in industries comes from the provision of job opportunities and economic profits.

Address challenges and analyze potential solutions to increase the efficiency of waste vehicles in collecting and delivering all waste from waste locations and then disposing of it in a central landfill, treatment plant, or recycling plant. The challenges in this research refer to the following points:

1. Environmental challenges that focus on providing increased quality of life and improving public health by preventing the spread of rodents and diseases resulting from the natural decomposition of waste that produces toxic gases such as carbon dioxide and methane. These gases can cause pollution and negative climate impacts (Z. Zhang et al., 2024).

2. **Economic challenges:** It is striking that during waste collection processes, transportation costs account for almost 70 to 80% of the total waste management costs. Furthermore, personnel constraints may lead to inadequacies, such as disorganization of collection plans and insufficient vehicles (Fang et al., 2023). Moreover, good facilities for WMS can be expensive; however, WCSs in industries can and should provide jobs and economic profits.
3. **Capacity challenges:** The electric vehicles (EV) and drones used must have a maximum waste container capacity to maximize the amount of accommodated waste, thus reducing the number of routes needed to be completed.
4. **Energy challenges:** Most of transportation operating costs go to energy costs, which can pose an enormous problem, especially to countries that lack fossil energy. Those can involve both developed and developing countries; henceforth, it is a major challenge that needs to be addressed and overcome.

Hence, solutions that address these challenges have been developed and implemented on the basis of AI to optimize waste transport and logistics processes (Abdallah et al., 2020). This encompasses optimizing waste logistics and transportation from four perspectives: transportation time, transportation distance, transportation cost, and effectiveness.

The optimization of techniques and methods employed to enhance WCS (Bihun & Lytvyn, 2022) are grounded in extensive research. These techniques and methods are designed to solve routing problems more efficiently and at lower costs than traditional methods, ensuring the system's effectiveness by finding the shortest paths. Notably, heuristic optimization methods such as GA, Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) are widely used due to their ability to deliver optimal so-

lutions quickly (Ruan et al., 2013)(Mesquita & Gaspar, 2021)(Khan et al., 2021).

Many studies involve the integration or modification of several methods to optimize performance, with genetic algorithms (Hejres et al., 2024) frequently highlighted as one of the most effective techniques. However, most studies face the lack of realistic dynamic methods based on the scaling effect of computation time and the failure to encounter many constraints with balancing priorities. Moreover, there is a great need for more research in this area as a significantly growing industry.

The genetic Algorithm is specifically tailored to meet the model's multi-objective methods/solutions addressing constraints (Erdelić et al., 2019), significantly enhancing the efficiency of the WCS and yielding promising results. On the other hand, RL can interact with the environment more appropriately by achieving an optimal path based on a complex dynamic environment, and handle problems that require multi-objective solutions with greater flexibility.

Moreover, a hybrid approach that integrates two algorithms to provide better performance than a single algorithm can achieve (Hejres et al., 2024). Optimization methods and machine learning (ML) techniques can make a very powerful combination that can solve complex problems such as routing problems (Hou et al., 2021). Integrated GA with RL can leverage and synergize the supremacy of both methods (Yoo et al., 2024).

This thesis presents an efficient model for addressing WCS challenges using UAVs, a specific type of EV. The primary goal is to identify the most effective approaches by applying various algorithms and methods to minimize operational costs while satisfying key constraints-namely, minimizing route length, maximizing waste collection capacity, and reducing energy consumption, all within a defined time frame to service all waste locations. The study follows a progressively complex experimental design to evaluate

the performance of the proposed models. This progression includes moving from single to hybrid algorithmic approaches, increasing the number of waste collection locations (WCLs), testing different sequences of algorithms integration in hybrid models, and evaluating performance as the number of UAVs scales from one to multiple. By comparing these configurations across multiple performance metrics, the research systematically demonstrates the effectiveness of optimization and machine learning techniques in solving the NP-hard WCS routing problem. The next three chapters discuss the three studies conducted by this thesis researchers summarized below.

1.1.1 Unmanned Aerial Vehicle for Waste Management System Using Genetic Algorithms

This study aims to find the most effective algorithm method to solve the WCS problem based on the constraints of our thesis, which are to minimize cost through finding the shortest paths, maximizing waste collection capacity, and minimizing energy consumption during serving all waste locations within a certain period of time. Based on in-depth research conducted by many researchers, GA is selected as one of the most superior algorithms that achieves high performance and can obtain optimal or near-optimal solutions within its features to find global optimal outcomes to find better paths in complex and dynamic environments. Furthermore, GA can handle noisy data, is more adaptive to change, and is more relevant to real-world problems. A robust fitness function can achieve extraordinary results, as proposed in this study. Hence, a GA with multi-objective fitness functions with a highly efficient structure addressing all the necessary constraints is proposed and used in the main model structure. The model also incorporates several hybrid approaches into the cross-GA operator, and this experiment of this optimization approach is demonstrated and discussed in Chapter 3,

and compared to other comparable previous studies, so that the outstanding results of the proposed model are substantiated.

1.1.2 A Hybrid Approach of Genetic Algorithms and Reinforcement Learning for Waste Collection Using Unmanned Aerial Vehicle

The study expands upon this model and proposes another approach that utilizes the operations of the GA in the first phase of the experiment and proceeds with its outcomes to be employed in the second phase using a RL. RL is also selected based on extensive research based on its features that ensure improvements over time, based on the concept of trial-and-error learning. Hence, RL can find the best route path within a complex, dynamic environment and a high-dimensional space through its highly adaptive feature set. However, RL requires more training time to find better results. Therefore, combining it with GA can be used to find the best starting populations, hence using the RL population as the trained outcomes experiment. In this study, four experiments are conducted based on two sets of methods, classified as individuals and hybrids. In this particular part, the experiments are expanded using the proposed two-phase model utilizing GA and RL. Initially, the model starts with GA operations, and the best path outcome is taken as the initial population for the second phase of RL, hence called GARL. The second experiment uses the RL algorithm in the first phase of the two phases and then uses the best population outcome in the RL part as the initial population for the GA part, which is the second phase; hence, this model is called RLGA. Thenceforth, the results of the two hybrid experiments are compared with two more experiments of both GA and RL individually. As a result, the GARL model achieved excellent results, outperforming all other models in all experiments.

1.1.3 Multiple UAVs for Waste Management Using a Hybrid Optimization and Machine Learning Approach

All of the above models are performed with a single UAV. In the real world, multiple UAVs can be more achievable in serving more waste locations in cities; thus, based on large-scale requirements, it is practical that multiple UAVs should be studied as a proposed model. Therefore, Chapter 5 discusses and illustrates multiple UAVs utilized within the GARL integrated model with the best proposed, and adds more algorithms that can achieve balanced task distributions between waste locations. Hence, adding and merging two more algorithms to both phases of the GARL model, K-means to do the clustering of WCLs among 5 drones, and A* algorithm, which finds the shortest distance after clusters have occurred. The model also adds another approach, the reassignment approach. After the drones are assigned groups and tasks, the re-assignment technique is activated based on the concept of achieving the balance of tasks and full UAV use all the time until all WCLs are serviced. Furthermore, the study compares the results of the application of the methods before and after the reassignment of tasks among multiple drones, and then also compares both models to a benchmark model using a single drone to compare the efficiency of the techniques between all models. This benchmarking helps in detecting incremental efficiency concerning two measures: the number of drones (from single to multiple), and the addition or absence of dynamic tasks assignment and reassignment techniques. Three criteria are used to evaluate the performance of the models by recording and comparing significant improvements in flight time, energy consumption, and total distance. The results show that the dynamic tasks reassignment technique model demonstrates better performance than the tasks pre-reassignment model. However, both models achieve superior performance to the single-drone model, i.e., the dynamic reassignment technique model outperforms the other two models, followed by the pre-reassignment

mission model, and the single-drone model came last. Overall, for all the above results, the methods proposed with these intense studies and analytics have proven to have high reliability with internal and external validity. These methods demonstrate the efficiency of algorithmic modification that can be used for real-world applications with any desired drone and that can be integrated into the system with high efficiency.

1.2 Research Problem

WCSs face a significant challenge in the form of high operational costs, which must be minimized to achieve efficient and sustainable waste management. The integration of UAVs presents a promising solution; however, it introduces complex constraints such as limited capacity, energy consumption, flight time, tour distance, and the requirement to service all WCLs. These constraints must be optimized simultaneously, which frames the problem as a multi-objective path planning task/ an NP- hard problem. The goal of this research is to develop efficient computational models that can solve the path planning and task allocation problems in WCS using UAVs. The thesis explores single and multiple drone scenarios and proposes solutions using advanced optimization and machine learning techniques.

1.3 Research Objectives

The research objectives in our thesis can be summarized as follows:

1. To propose a model that can improve the waste collection process proven through tests implementations and efficient validation of high-performance results.
2. To design and develop a highly efficient WCS model using a hybrid

approach to address all constraints, minimize cost, and improve the efficiency of the model by at least 20% compared to existing methods.

3. To conduct a series of experiments with increasing complexity to evaluate the scalability and robustness of the proposed model while being tested in different measures.
4. To evaluate the process efficiency through the calculation of the fitness function that provides the shortest path with the least time to finish the planned routes.
5. To develop an approach that supports the dynamic environment.
6. To address complex constraints of the system as a trade-off policy between these various constraints to minimize the distance and energy consumed with the flight time based on maximizing the amount of waste collected using the proposed model by optimizing the route planning and obtaining an improvement of at least 10% compared to the proposed superior model.

1.4 Contribution to the Field

This thesis contributes to the research field with several approaches in developing efficient models for single and multiple UAVs utilized in WCS, relying on optimization and machine learning techniques, as follows:

- Developing an efficient and high-accuracy model using a single optimization method, utilizing an innovative multi-objective function based on GA and hybrid GA operators for WCS with a single UAV.
- Proposing a high-accuracy hybrid approach for WCS with a single UAV that integrates optimization and machine learning algorithms based on a two-phase approach and alternates sequences between these algorithms. In other words, the developed approach starts with a GA

algorithm, followed by a second phase of RL, delimited as GARL. The other proposed approach starts inversely with an RL algorithm, followed by a second phase of GA, demarcated as RLGA.

- Improving the proposed GARL approach, which is optimized through managing WCS with multiple drones. This approach also integrates additional algorithms, namely K-means and A*, to efficiently aggregate data according to the drone's nearest locations and capacity and energy constraints. The A * algorithms are then used to manage the clustering tasks for each UAV based on the shortest distance.
- Design an approach to further optimize the proposed model for multiple drones, based on dynamic reassignment tasks, which ensures a further balanced and efficient task reassignments distribution among multiple UAVs, hence reducing the cost of WCS operations while increasing the efficiency of the target system.

1.5 Research Questions

The experiment process in this thesis depends on gradually evolving complexity to gradually substantiate the application efficiency of the algorithms while being tested on different measures. The experiment of gradually increasing complexity employs efficiency comparisons in different measures; i.e., from single algorithm to hybrid integrative algorithms; from lower to higher number of WCLs; comparing different sequences of algorithms employed in the hybrid models in their respective phases; and then exploring the UAVs' efficiency regarding the increase in number from single to multiple.

RQ1: What is the effect of GA application in WCS utilizing UAV when developing a robust fitness function that addresses multiple constraints based on trade-off techniques?

RQ2: Which approach performs better: genetic algorithms alone or a hybrid approach combining genetic algorithms and reinforcement learning?

RQ3: What is the effect of the proposed hybrid approaches using four algorithms: GA, RL, K-means, and A* to solve WCS cost problems using multiple drones?

RQ4: What is the impact of using reassignment techniques with multiple UAVs on improving the performance of the proposed model to reduce WCS operational costs?

1.6 Thesis Outline

This thesis is structured as follows. In Chapter 2, a literature review discusses several reviews and analyzes some of the constraints of WCS and the techniques used to improve these systems. GA, RL algorithms, and hybrid methods contribute to solving path planning problems and improving WCS. Chapter 3 proposes a multi-objective GA aiming at improving the waste collection process through identifying the best appropriate tours that can minimize the cost using a single drone, which answers research question 1 (RQ1). Chapter 4 presents the hybrid integrative models of multi-objective GA and RL to optimize the waste collection process by finding the best solution for routing problems with suitable tours in complex scenarios that minimizes WCS cost while addressing its constraints multi-objectively using a single drone, which addresses research question 2 (RQ2). Chapter 5 shows a hybrid integrative model that combines different algorithms multi-objectively, namely GA, RL with K-means and A* search algorithms, to cluster and ensure an efficiently balanced distribution of waste locations assignments among multiple drones; it answers research questions 3 and 4 (RQ3, RQ4). Chapter 6 concludes the thesis and presents limitations and future work.

Chapter 2

Literature Review

2.1 Literature Review

The literature review for this thesis discusses previous research and approaches to solving the routing problem for an EV system, including UAVs in WCS; hence, the outline of this chapter includes a search for the best methods and techniques in that area. In Section 2.2, the routing problem in EV systems is investigated. This section includes subsections that analyze these methods, such as optimization techniques, machine learning techniques, and hybrid techniques. Each subsection consists of approaches for certain types of electric vehicles, namely UAVs and unmanned ground vehicles (UGVs). Although this thesis focuses on one type of EV, namely, UAVs. Section 2.3 discusses the routing problem of UAVs. This section focuses on individual approaches to the routing problem, which includes research on GA and RL individually. The second subsection discusses hybrid methods for the routing problem, specifically for GA and RLs in a single UAV, and the third subsection discusses similar approaches for multiple UAVs.

2.2 Routing Approaches Used for EVs System

This section introduces and analyzes most of the methods and techniques for improving the performance of finding the optimum routes using various types of EV systems. The following subsections present several studies of EV routing that achieved comparably high performance using various techniques such as optimization, machine learning, or a hybrid approach.

2.2.1 Optimizations Techniques

Optimization techniques are methods that modify parameters and hyperparameters to find optimal solutions by maximizing or minimizing constraints using one of the various optimization techniques. Table 2.1 summarizes the optimization approaches in both types of EVs.

2.2.1.1 Optimization Approaches in UAV

Optimization techniques aim to control the drone path distance, obstacle avoidance, and flight time management which can overcome the path planning problem by finding the best design variables to discover the optimal solution for the UAV route system. Optimal path planning is one of the most important problems that drones encounter. It contains major problems such as solving Traveling Salesman Problem (TSP) (Sonmez et al., 2015) and the Vehicle Routing Problem (VRP) problems. The GA can solve the problem and find the optimal flight path for the drone in a 3D environment. The proposed method using the GA ensures that Check Points (CPs) are not visited twice, and redirection should occur at checkpoints outside the no-fly zone. Unfortunately, this approach does not support dynamic real-life scenarios. Mesquita et al. (Mesquita & Gaspar, 2021) proposed a novel path planning optimization algorithm based on PSO to find the optimal path which can reduce energy consumption, enhance flight time, and lower com-

putational costs. Some software was used to simulate the efficiency of the proposed algorithm via Ground Control Stations (GCS) and Mission Planner that autonomously control UAVs and visualize the stream in real-time. Their algorithm could not be compared with similar studies, so to examine the accuracy of the results, it was compared to three case studies. Two of the case studies applied simulation software to simulate the efficiency of the proposed algorithm via GCS and Mission Planner that autonomously control UAVs and visualize the flow in real-time. The third case study compared the real field recorded final errors for the entire path planning distance, which amounted to 10% resulting in an average of less than 7% when considering all three case studies. Although the final error is 3%, the average of all cases is 1%, but it is not practical in the simulation cases. In summary, the average execution time error for all cases is 5%. Similarly, a survey (Zhao et al., 2018) compares 231 various studies from 2008 to 2017. It has been found that the top-ranked algorithms which outperforms the other algorithms in terms of performance are GA, ACO, Learning-Based methods (LB), PSO, and fuzzy algorithm (FL). Platanitis et al. (Platanitis et al., 2020) study addresses a solution to the path planning problem in UAVs focusing on energy reduction and the creation of a safe flyable path, where an instantaneous ideal path as the shortest one is obtained by the GA method. GA generates different seeds to find the best flyable path corresponding to each gene for each chromosome. Crossover, mutation, and fitness processes have been used to find the shortest path and the least waypoint distance. The simulation map implementation of the test environment involves static sampling only, so using a variable sample increases the efficiency of experimenting with the results. An additional application that uses optimal path planning layout is cinematography. In this study, a novel formulation approach (Alcántara Marín et al., 2021) is used for a method based on nonlinear optimization that runs multiple drones to find the best paths and coordinates

collision avoidance in a dynamic environment that uses priority-based computation. This technique improves filming shots that becomes smooth with the mutual camera visibility, which also reduces jerky movements. Even so, this method needs to compare its results with other known methods such as RL to check the feasibility of its approach. For the purpose of waste management, a similar method is used also based on nonlinear optimization that runs multiple drones to find the best paths and coordinates collision avoidance in a dynamic environment that uses priority-based computation. WMS (Kaabi et al., 2022) is an important sector in which solving the VRP is paramount, hence a study is conducted that utilizes a two-phase approach to the Maximum Waste in the Minimum Time during each Trip (MWMTT). The approach in the first phase is to increase the capacity load of hazardous waste with the shortest path. Then in the second phase, it distributes tasks among multiple groups to avoid frequent charging of drones and to manage continuous flights to the target. This proposed approach contains two trials, the first involving small-sized instances of around 10 to 40 and medium-to large-scale instances of 48 cases within 41 to 981 sites. The results are compared with the lower bound and linear programs. Particularly MWMTT achieves a performance error of less than 10% in small cases. On the other hand, a linear program cannot solve more than 40 cases. The lower bound is used to evaluate the efficiency of the approach which satisfies small-scale cases, otherwise it requires further optimization to solve medium and large scales cases. In many WMSs (Aktemur et al., 2020)) the Internet of Things (IoT) technology is used to collect data continuously and to manage waste path planning. Like other systems GA is a valuable approach for WMS in a smart city. In a case study experience in the Bakirkoy district of Istanbul, a traditional city with its old infrastructure causes inconvenient outcomes, so a proposed approach for the routes planning to turn around the city and improve performance is considered. The study suggests solving the Ca-

capacitated Vehicle Routing Problem with Pickup and Alternative Delivery (CVRPPAD) through a hybrid approach of Constraint Logic Programming (CLP) (Sitek & Wikarek, 2019) to pre-solve the model and master planning (MP) with heuristic optimization creation inference. To generate an initial solution, another study uses the SA heuristic algorithm (Torabbeigi et al., 2021) based on the petal algorithm to find path selection via a binary integer programming model using a local neighborhood search algorithm. The main objective of the model is to reduce the Expected Loss Of Demand (ELOD) by utilizing the Drone Delivery Schedule model with drone Failures (DDS-F) which determines each drone's task to Subset customers with trips sequence and compares the results with the Makespan problem model to find a better path, flight time, and payload. S. I. Khan et al. (Khan et al., 2021) discuss the most widely used heuristic algorithms to determine the optimal path planning for medical emergencies using drones that can save lives and reduce cost and computational time as Capacitated Vehicle Routing Problem (CVRP) algorithm, GA, ACO, and PSO. The CVRP algorithm attained the highest performance compared to the other algorithms used in the proposed model. PSO can find exact local minima and global minima to locate the optimal path within the best target location. In addition, ACO and GA have the ability to get the shortest path, while ACO results perform better than GA. However, the algorithm's achievements require benchmark solutions that are optimally comparable in terms of accuracy. Additionally, it should be evaluated based on the number of patients that can be served, considering the urgency and level of risk in their situations. Another research proposed a real-time distributed path planning (Hu et al., 2021) that can track moving targets on the ground using fixed-wing drones. The proposed method aims to solve the constraints problems such as real speed and turning rate, control input saturation, collision avoidance, and obstacle avoidance. UAV speed and turning rate can be taken based on the change-

able movement of the target. The onboard sensors are used to avoid occluded paths by buildings. It sets the target coverage degree instead of the target distance and it includes control input cost and sensor energy consumption. Priority objectives are fuzzified and a satisfactory degree and the Generalized Varying Domain (GVD) is used to balance between the optimization and priority. The Distributed Model Predictive Control (DMPC) is combined with the system to create a distributed cooperative path-planning model for each drone. The simulation results show the effectiveness of the proposed method by comparing it with the Hierarchical Optimization (HO) method. Each UAV can achieve the optimum path at every selected time. On the contrary, the performance evaluation of building detection criterion sometimes obstructs the sensor's line of sight when the distance between the drone and the target cannot be captured. In addition, this model only uses the simulator and is not examined in real life.

2.2.1.2 Optimizations Approaches in UGV

Optimization methods improve autonomy, localization, and navigation, and avoid obstacles. Liu. et al. (Q. Liu et al., 2021) employ a GA in an EV path learning model to improve energy efficiency to increase mileage endurance. The charging state has an important impact on the travel route which illustrates the problem of finding the shortest path and reducing traffic volume as well. Some of the significant constraints that are essential to how the model works are the state of charge which defines battery capacity, along with the charging pile, queue length, and path direction depending on the nature of the selected route. For example; a route as the freeway has its own features; such as the starting origin of the journey, the destination, and finally the charging nodes. The proposed method reduces travel time by up to 6.5%. Furthermore, the research analyzes various parameters that affect the model such as the high volume of traffic that increases the path time, and because

the alternative path with a long distance can be chosen, thus the power consumption is increased as well. Moreover, the temperature has a greater effect on the driving path, and therefore when the temperature drops, a shorter vehicle endurance occurs. In addition, when EV speed increases, it reduces travel time and increases average power consumption. This approach can limit the vehicle paths obtained to only start from specific origins and reach specific destinations. It should be noted that the speed ranges are determined between 90 km/h and not exceeding 120 km/h. Therefore, it does not consider all user behavior and different vehicle speed in real life. Users may encounter special circumstances that require reducing vehicle speeds below 90 km/h, such as in the event of a traffic accident or traffic jam. In (Sangeetha et al., 2019), consuming Gain-Based Ant Colony Optimization (GACO) and GGAnt as modified colony optimization algorithms to solve path planning problems, are examined by a pseudo-code representation of modified ACO as the proposed algorithm. Optimum path planning not only finds the shortest route but also reduces the cost and time of the journey with less computation time. Clearly, GG-Ant is the energy computed for the best path and ignores all other paths. The sigmoid function is used in the GG-Ant which provides smooth rotations. The new approach to UGV demonstrates power reduction and has recorded lower computation time and efficient power saving. While it can end at the local optimum when the environment complexity increases, then attempts to solve it with the global optimal solutions. Automated guided vehicles (AGVs) (Z. Zhang et al., 2021) have been a critical factor in the efficiency of manufacturing workshops. The AGV employs a multi-objective optimization approach addressing problems with main key objectives such as transport distance, energy consumption (EC), and cost based on the energy-efficient AGV Path Planning (EAPP) model. It aims to find the best path using Dijkstra's algorithm which is a type of graph search algorithm that finds the shortest path between

any two nodes in a region, but it can't consider all the nodes in the entire path. Thus, some improvements have been done to compute all the shortest paths between nodes in an EV path using the Search Path algorithm. While comparing several algorithms, PSO, NSGA-II, and ACO, all registered the same best path though PSO takes longer run time created on priority-based indirect particle encoding. Additionally, PSO granted an efficient scheme that extracts the path from heuristic information based on priority nodes. Therefore, the EAPP problem is solved by a two-stage solution and PSO. The extraction method is improved to prevent backtracking, and path loops and can reduce driving time. This method needs improvement in the cases of several AGVs and multiple AGV loads that suffer from a shortage of the EC data acquisition system. Zhou. et.al. (Zhou et al., 2021) proposes a new intelligent path planning strategy based on the Floyd algorithm which is used to find the best shortest path in a given weight between two nodes. This system is integrated between the electrified transportation network and the power grid. In addition, Dynamic Wireless Power Transfer (DWPT) is integrated into this EV system as the primary charging method, which is expected to become the predominant method in the future. This is in contrast to battery swapping, which increases charging costs and time. The multi-stage two-layer optimization algorithm has been added to the framework to verify the charging power between the links of the nodes. In addition, the upper layer uses PSO to estimate the charging power decision, while the lower layer is employed to solve the optimal power flow problem by determining the voltage deviation rate. The proposed system respects travel time, user convenience, power charge, and cost. Moreover, a novel charging path planning algorithm (Hou et al., 2021) based on bidirectional Martins' algorithm is proposed to guide the user with an optimal path at a cost-effective charging station. The PSO is integrated with the proposed EV path planning framework cost-effective charge station. This framework is designed based

on multi-objective path plan addressing constraints such as driving distance, total route time, energy consumption, and energy charging cost. The multi-objective decision-making solution to the problem uses different methods to handle this kind of problem such as the Information Synergy Entropy (ISE) based interval intuitionistic fuzzy TOPSIS (ISE-IITOPSIS) method (Z. Chen et al., 2021), and the weighted method which is the simplest way to solve this problem by assigning an exceptional weight to each objective and normalizing all parameters. In addition, graph-based theory as a Yen algorithm is applied to reduce waiting time caused by traffic congestion. Furthermore, Dijkstra algorithm, Martin's algorithms, and various heuristic algorithms are used to determine the optimal paths for EVs. In particular, an important factor in operating EVs is the pricing mechanism at the charging station, which reflects the selection of the best charging cost and affects the route of the EV. Fixed electricity pricing, time-of-use pricing, and real-time pricing adjust the electricity pricing model to maximize charging station revenue more efficiently when the PSO algorithm is applied. Indeed, the new algorithms represent enhancements of Martins' Algorithm (MA), and the Bi-directional Martins Algorithm (BMA) (Demeyer et al., 2013) referred to as EHBMA are designed to address the EV charging path problem while accommodating heuristic constraints more effectively than traditional models. CSBAP is also used to raise the energy cost revenue of EVs by adjusting the more profit than the fixed cost due to increased lower prices demand, which affects the selection of charging stations and resolved routes. To create a more accurate charging lane planning model, it is essential to take into account the more complex road conditions and driving conditions for EVs. Without a doubt creating user avatars that execute a better fit is recommended for charging routes. Alizadeh. et al. (Alizadeh et al., 2014) discuss solving the travel path decision problem for EVs with two main constraints being the shortest path and charging locations across the social planner to

reach balanced electricity prices. The decision is made based on the traffic situation, electricity prices by location, and charging rate, which are the main targets of solving this problem objectives. Electric or intelligent transportation systems include the power system, transportation network, and EV owners. Importantly, their approach examines the effect of EV owners to solve the problem based on main policies as the social planner's solution is Social Optimum (SO) which reduces the cumulative path and time cost to the population in a centralized manner. In addition, the UE is an individual decision based on the lowest cost and Disjoint Pricing (DP) that examines the separation between intelligent transportation and energy systems. As a result, this approach reaches the lowest cost and UE records less iteration time, although the cost of electricity increases. Whereas the influence of diverse user categories based on travel time is not discussed in this approach, though most user behaviors are based on cost. The WMS approach can be a very promising application in smart cities. One study (Sarvari et al., 2020) combines three popular algorithms for building productivity and scalability systems: GA, ant colony, and tabu search. The Waste Collection Vehicle Routing Problem (WCVRP) is managed by combining these metaheuristic algorithms. The proposed model addresses the constraints of shortest path, time window, computation cost, and service priority over the needs of each customer. This study investigates the case study of Luxembourg which is a waste management company that deals with dynamic waste collection optimization models that are found in large organizations such as shopping malls, restaurants, etc. In addition, the database is collected from sensors based on the filling of customer bins, the number of trucks, locations, and availability of trucks based on customer need. All of these data can affect the working efficiency of the model. Despite the good results of ACO and TS, the computation time takes a lot and affects the dynamic VRP solution. As a result, GA outperforms the other two algorithms regarding achieving

effective cost and time on a priority basis. Unfortunately, fixed route planning solutions cannot be applicable to waste operating system management in real life. Based on their conclusion, when experimenting with the results of efficient solutions for only finite nodes, they are not necessarily applicable in real life. Similar to the previous study, another case study of Al-Awali district in Makkah (Alwabli & Kostanic, 2020), Saudi Arabia, GA is chosen to enhance the dynamic path VRP of the WMS. In summary, GA performs better than other metaheuristic algorithms regarding computational time reliability, shortest path, and lowest fuel consumption. In experiments, many path planning methods are not the best path solution between two nodes. Undoubtedly, if VRP solely concentrates on finding the shortest path without addressing other constraints, it fails to achieve the primary objective of reducing costs for enterprises. Solid Waste Management (SWM) as well (Bhargava et al., 2019) uses GA in its system, because it has the convergence ability, which can indicate that the performance of the GA is on the right track. Another study defines the smart clean city system (Dolinina et al., 2019) as a dynamic WCS with actual data that accepts any emergency from a predetermined schedule path based on the optimal path to be achieved. The proposed model uses a rerouting optimization algorithm that can improve constraints handling, such as path cost and additional capacity. Due to that, the dynamic path planning is proposed as an improvement of the ant colony algorithm (C. Wu et al., 2020) by optimizing its parameters along with the particle swarm algorithm. Two of the experiments are examined for regular traffic, and the third experiment is for heavy traffic. The proposed method achieves greater accuracy and reduces the congestion rate, especially when the method replaces the road length with the whole road condition features. Nevertheless, the optimization process itself requires computation time and increases computational complexity.

Previous research suffers from several limitations, such as dealing with a

simulated environment rather than a real-world environment. Furthermore, most studies deal only with static data and steady-state conditions. The proposed models do not address all the constraints that may be encountered in the real world. Some solutions do not generalize their results to more complex and demanding data. In other words, using a small dataset will result in poor generalization and the results may be affected; however, as the dataset increases, as is the case in the real world, the results decline. In addition, it lacks benchmark solutions that are ideally comparable in precision and can be evaluated for various problems, such as the problem of evaluating the number of patients that can be served based on the severity of their urgent risk conditions (Khan et al., 2021). Most models do not address complex cases and route conditions.

Table 2.1: Summaries of Optimization Approaches

Ref	Approach	Methodology	Application	Limitations	Type
Sonmez et al.2015	Finding the optimal flyable path for the UAV in a 3D environment.	GA	Multiple applications (Military and civilian missions).	Dynamic environment	UAV
Mesquita et al. 2021	Finding an optimal path that can reduce the limitations of energy efficiency and flight time, as well as reduce the computation and cost round controlling stations.	Novel path planning optimization algorithm based on PSO.	Agriculture (birds eat and damage fruit in orchards, making them susceptible to infection and reducing their quality, thus using drones to solve this problem).	Their algorithm cannot be compared with similar studies, so they examined the results from 3 case studies.	UAV

Continued on next page

Table 2.1 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Platanitis et al. 2020	Solve offline UAV path planning problem, through the principles of safe flyable candidate trajectories, and energy reduction.	GA	Real-world applications with real geospatial DTED.	Involves only static sampling in the simulation map.	UAV
Nguyen et al. 2021	Define paths as desired and directed paths and implement them independently.	Drone management system and its component systems.	Civil and commercial application.	Need more investigations and real tests.	UAV
Alcantara et al. 2021	Propose a method for planning optimal trajectories of UAVs performing autonomous cinematography.	Novel optimization non-linear formulation.	Autonomous cinematography.	Need to compare the results with other known methods as reinforcement learning.	UAV

Continued on next page

Table 2.1 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Sitek et al. 2019	Solve CVRP problem with CVRPPAD.	Merging CLP and MP	Multiple applications.	The model does not handle some constraints (route length, number of route points, prioritization of items, timeouts).	UAV
Torabbeig et al. 2021	Reduce the ELOD. The optimization model (DDS-F) was developed to determine the assignment of each drone to a subset of customers and the corresponding delivery sequence.	SA heuristic algorithm based on the Petal algorithm to find initial solution generation, a binary integer programming model for path selection, and a local neighborhood search algorithm to find better solutions.	Parcel delivery.	The exact method failed to find a solution for cases with more than 12 clients; it was impossible to compare SA performance in larger cases.	UAV
					Continued on next page

Table 2.1 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Khan et al. 2021	Achieve optimal route planning (vehicle routing) through a proposed algorithm.	PSO, ACO, and GA.	Medical domain.	Need benchmark solutions that are optimally comparable to the level of accuracy and can additionally be evaluated by the number of patients that can be served based on the intensity of their urgent situations of risk.	UAV
Continued on next page					

Table 2.1 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Hu et al. 2021	Cooperative path planning with different priorities via DMPC-GVD (Solve multiple goals with priorities).	Fuzzy multiobjective optimization via DMPC-GVD.	Tracking a ground moving target in urban.	The detection performance building evaluation standard sometimes obstructs the sensor's line of sight when the distance between the drone and the target cannot be captured. Using the simulator only.	UAV
Continued on next page					

Table 2.1 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Liu. et al.2021	EV Path planning model to develop the driving competence of electric vehicles on the highway.	GA	EV path planning.	limit the vehicle paths obtained to only start from specific origins and destinations/ speed analyses are determined between 90 km / h and not exceeding 120km/h.	UGV
Sangeetha et al.2019	Reduce energy (cost and time) during path detection.	GACO and GG-Ant.	New approaches to energy-based path planning.	It can end at the local optimum when the environment complexity increases.	UGV
Continued on next page					

Table 2.1 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Zhang. et al.2021	Solve the problem of AGV path planning and energy efficiency considering EC in a manufacturing work- shop environment (Time and cost).	PSO-NSGA-II and ACO-Dijkstra algorithm-the two-stage solution.	Pick-up and drop-off locations for every transport task.	Not building an EC data acquisition system/Not a multi-load AGV and multiple AGVs.	UGV
Zhou. et al. 2021	The proposed path planning strategy considers travel time, power charge, charging cost, and user convenience.	Floyd algorithm and multi-stage two-layer optimization algorithm (PSO-voltage deviation rate).	Urban electrified transportation network.	Did not mention	UGV
					Continued on next page

Table 2.1 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Hou. et al.2021	Guiding the user on the optimal path at a cost-effective charging station.	PSO algorithm is applied to find the optimal solution to the charging station and charging rate.EHBMA to solve the EV charging path problem, consider the heuristic constraints CSBAP is also used to increase the energy cost revenue of EVs. A novel freight-route planning algorithm based on the bidirectional Martins' algorithm.	Large-Scale EVs	Create user avatars to better fit recommended charging routes. More complex road conditions and driving conditions for electric vehicles must be considered to obtain a more accurate charging lane planning model.	UGV
					Continued on next page

Table 2.1 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
WU.et al.2020	Proposes a dynamic path planning method based. on the ACO algorithm in high traffic.	Enhance ACO based on optimizing its parameters via the PSO algorithm.	Dynamic path planning	The optimization process itself requires computation time and also increases computation complexity.	UGV
Alizadeh.et al.2021	Find travel routes including charging locations and associated fee amount under time-varying traffic conditions as well as dynamic location-based electricity pricing.	SO reduces the cumulative path and time cost to the population. UE is based on an individual decision based on lower cost. disjoint pricing.	Intelligent transporting system	The influence of heterogeneous user types is left to inconvenience travel time for future work.	UGV

2.2.2 Machine Learning Approaches

Machine learning (ML) techniques encompass a variety of efficient learning algorithms that enable the creation of models capable of interacting with static and dynamic environments to optimize system solutions to the highest degree. ML (Abcouwer et al., 2021) can find the best path with minimal computing for real-time actions, the shortest route with obstacle avoidance, acquisition of energy management, and reduction of the cost of the electric vehicle system. Table 2.2 summarizes the machine learning approaches in both types of EVs. The following subsections describe the methodologies and approaches used in the most recent research on UAVs and UGVs.

2.2.2.1 Machine Learning Approaches in UAV

UAV systems utilize ML models to forecast UAV trajectories, with the models being trained on data collected to predict optimal paths. ML algorithms are used incrementally for multi-objective Path efficiency. Sequential decision-making in problem-solving, based on intelligent mission planning methods (Yue et al., 2022), can be applied in the Suppression of Enemy Air Defense (SEAD). The experiments study Deep Reinforcement Learning (DRL) based on a Proximal Policy Optimization (PPO) algorithm that considers the basis of DRL to gain high performance and robustness. DRL is used as a trial-and-error approach with dynamic learning created on agent action within its environment. The experience needed further investigation of the complex large-scale combat and multi-agent cooperative planning problem. As in the previous study, the DRL is used to handle drone path planning in dynamic environments (Chao et al., 2020) for military applications while maintaining safety standards aimed at any opportunity for enemy threats. The information used for system training is collected from the radars needed to create a situation map of enemy threats. A modified Dueling Double deep Q-Networks (D3QN) algorithm is used to predict Q values

for all procedures. The chosen action policy is based on a combination of ϵ -greedy strategy and heuristic research. The real-time reaction has been achieved with good results, although it is not possible to collect data in all cases from just one drone. System optimization usually requires operating within several drones that can communicate with each other to perform better in the real world. Consequently, managing drone communications can robust the system. ML algorithms, particularly DRL, encounter challenges and considerations in drone applications, notably within the specific domain of the Green Vehicle Routing Problem (GVRP) (Sabet & Farooq, 2022). It is recommended to be used in further research due to its reliability, robustness, and problem-solving efficiency. Quadrotor UAVs are a specific type of UAV (X. Lin et al., 2019), and it is a proposed supplementary controller that uses RL to enhance the control of this type of UAV. Thus, this model suffers from uncertainty and external disturbances. RL has two main functions, the policy that takes action to control the value and the value function that evaluates effectiveness. In addition, this approach can also be applied to online drone controllers. Furthermore, Lyapunov's theory applies to check the convergence of RL. This model uses three simulations to demonstrate the accuracy of the methods. The simulation includes learning mode and test mode; where the learning mode means the drone starts the trajectory at a random point and ends at the target to have more range for exploration. Additionally, the test mode means starting from a stable node and aiming at their target. RL has QLearning that gives less variation to actor training and Time Difference (TD) that updates continuous steps that can be run online with experience to increase the rate of training. On the other hand, this approach works well in experiment, but is not tried in real life. DRL is used to solve Autonomous Motion Planning (AMP) problem and to solve the Multi- UAV Target Assignment and Path Planning (MUTAPP) problem for UAVs using a Multi-Agent Deep Deterministic Policy Gradient (MAD-

DPG) algorithm. Novel proposed algorithms are identified as Simultaneous Target Assignment and Path Planning (STAPP) (Qie et al., 2019) based on MADDPG. This approach needs to assign many tasks to solve problems required to be solved within discrete routes, manage a dynamic environment, and improve the quality of training. Despite the efficient outcomes, training convergence becomes more complex when agents increase, otherwise, agents increase do not affect the training time.

2.2.2.2 Machine Learning Approaches in UGV

ML algorithms have demonstrated significant promise in identifying optimal charging paths for EVs while simultaneously minimizing costs and emissions. DRL (C. Chen & Zhou, 2021) reduces the time interval of an uncertain logistic transport pathway and is suitable for a complex problem model when compared with traditional methods by 60.71%. Numerous researchers are striving to address previous challenges while developing solutions for common path problems. These include methods such as the Markov decision making (Anghinolfi et al., 2016) method, Monte Carlo algorithm, and Rollout algorithm which decrease runtime. Overall, the DRL has recorded very promising results in the case of optimizing uncertain logistics transportation systems. A study introducing innovative techniques uses the full convolutional model (Mazzia et al., 2020) of path design in a grid map as a DRL approach to global path planning. The deep neural network (DNN) trains a synthetic dataset to find the optimal waypoints and then removes unnecessary nodes to add the required nodes. The Density-Based Clustering Algorithm (DBSCAN) is used to group nearby nodes based on field geometry. Furthermore, the search algorithm A* is used to calculate the final global path to cover the crop area. In the field of agriculture UGV are used to increase crop quality, harvest yield, and reduce cost. Since synthetic datasets are used when the real dataset is difficult to find in the real

world, therefore, an algorithm is designed to generate an occupancy grid map, which dataset includes 1000 synthetic images that were evaluated after the training phase. Hence the method showed scalability and robustness. The system can be optimized by using remote sensing color space to integrate Deepway computing with a segmentation network that connects waypoints with the relevant occupancy grid map. Nonetheless, this approach skips various inadequate map regions that can affect the overall prediction negatively. Another study (Yao et al., 2020) suggested a new method based on DRL and the artificial potential field method. The artificial potential field can improve the effective path planning to solve the local stable-point problem which means the agent is entrapped in one point of the complex environment as a specially shaped obstacle or is trapped inside a barrier and cannot reach the target. The DRL is combined with a Black Hole Potential Field (BHPF) that is used as a DRL environment which can work under static or dynamic conditions. A trained agent can handle real-time obstacle via avoidance competently. On the contrary, optimizing the system requires enhancing the proposed algorithm to avoid collisions between more than one agent and to ensure agents cooperation with a multiagent RL algorithm. The system can achieve adaptation to different environments and smooth trajectories through real-time calculations. Even so, the model requires prior optimization of the BHPF domain to adapt to diverse environments. Obviously choosing a wide range can affect multiple gravitational superpositions that the model can fail to avoid obstacles, on the other hand, a small domain may miss agent detection. Shibl. et al. (Shibl et al., 2021) conducted an examination of six ML algorithms, including Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), Long Short-Term Memory (LSTM), DNN, and K-Nearest Neighbor (KNN), to address power loss, load variance, charging costs, and voltage fluctuations. Among these algorithms, LSTM achieved the highest accuracy, estimated at 95%. The system

achieves robustness, and high performance, and increases protection from data uncertainty to optimize the route charging of the EV to reach charging stations based on the charging speed. Through utilizing a Vehicle-to-Grid (V2G) technology it reduces power requirements, as well as adding Gaussian White Noise (GWN) techniques that support system preservation of data uncertainty. Thus, the results show only a slight improvement when this specific technology is used. However, these methods require a comparison of their results with other systems, especially due to lack of data. The multidisciplinary approach (Ramesh & Praveen, 2021) used in this system is based on the method of artificial intelligence, which uses deep learning as reinforcement learning to path learning and localize vehicles at the high and low levels with respect to the real-time decisions made by cloud facilities and advanced resources through transfer learning. Additionally, Recurrent Neural Network (RNN) is employed for power management to reduce cost and control power usage because it contains memory that is continuously learning using time series data. Several methods engage speed control as fuzzy logic, Radial Basis Function Network-Based Adaptive Fuzzy System (RBFN-AFS), and Direct Torque Control (DTC). SVM has a high potential to minimize CO₂ emissions within the satisfied travel time cost function using RL techniques (Zeng et al., 2017) solved via the KSP algorithm and resolves the environmental routing problem. Additionally, SVM exhibits durability for handling overfitting and high generalization problems, as it is extremely estimating for many emission problems. To validate this method, it compares several regression-based models for ML namely the EOPS model, SIDRA running model, VT-Micro model, and ANN model. EOPS model for mean correlation velocity, VT-Micro model for velocity, and acceleration estimation. The SIDRA model operates based on vehicle acceleration, deceleration, and a backpropagation neural network based on CO₂ emissions. In addition to using some statistical metrics to esti-

mate performance such as Mean Absolute Percentage Error (MAPE) and squared correlation coefficient. It found that the SVM model is gaining better than other classic models. These research results affect market penetration based on travel time and environment saving. This approach can be included within EVs, or with Plug-in Hybrid Electric Vehicles (PHEV) and autonomous vehicles to solve some of the constraints that are vital in this study as determining the shortest path problem. This model does not discuss the different emissions ratios that depend on the vehicle's driving condition at various speeds. This model should not only measure CO₂ emissions, but also needs to consider other toxic gases such as NO_x and HC. Furthermore, Dijkstra's algorithm is used to find the shortest path of WMSs for smart city systems (Reddy et al., 2019) that use priority based on needs combining IoT alerts and SMS. It has the ability to separate wet and dry waste by using a moisture sensor which achieves efficient garbage collection. Many solutions utilize well-known methods to start their new approaches such as applying the k-means algorithm which is used to solve waste collection problems. Its waste collection process is based on the amount of waste that is first collected. The k-means algorithm determines the nearest WCL point to be serviced. In addition, the improved k-means algorithm (Bihun & Lytvyn, 2022) finds the optimal path with certain constraints such as time with a controlled search interval. This algorithm can be improved when it combines up with other exact or heuristic algorithms that solve the traveling salesman problem.

For machine learning research, most approaches require further investigation into the complex, large-scale, combat, and multi-agent cooperative planning problem. Some constraints of previous research have not been addressed, such as real-world validation, dynamism-privacy, uncertainty, and interoperability. Further improvements in drone synchronization are needed to improve real-world implementation. Comparison of model results with

other real-world systems is required, especially due to limited data. Due to the lack of actual data, insufficient mapping areas can be overlooked, which could affect the overall prediction.

Table 2.2: Summaries of Machine Learning Approaches

Ref	Approach	Methodology	Application	Limitations	Type
Yue et al. 2022	Solve sequential decision-making problems with intelligent drone mission planning based on DRL.	DRL based on PPO algorithm.	SEAD mission planning.	Needed further investigation of the complex large-scale combat and multi-agent cooperative planning problem.	UAV
Rigas. et al. 2014	Intro Solve energy-efficient routing problem (Survey).	G2V or in V2G model and AI techniques.	Integration of EVs into the smart grid.	Some constraints not satisfied as (Uncertainty-Dynamism-Privacy-Real-world validation-Interoperability).	UGV
					Continued on next page

Table 2.2 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Yan et al. 2019	DRL method for path planning of drones in dynamic environments with potential enemy threats.	DRL based on D3QN algorithm.	Anti-terror missions.	Data can be collected from only a single UAV / Need to improve the system within several UAVs which can communicate with each other to perform better feasibility in the real world.	UAV
Yuan. et al. 2021	The path planning scheme is proposed to solve the uncertain logistics problem.	Reinforcement learning.	Transportation Scheduling (Package delivery).	The model requires prior optimization of the BHPF domain to adapt to diverse environments.	UGV
Continued on next page					

Table 2.2 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Shibl. et al.2021	Used to direct electric vehicles to charge stations to reduce load variation, power loss, voltage fluctuations, and charging cost while considering conventional and fast charging and V2G charging technologies.	ML create accurate future decisions based on historical data, which are DT, RF, SVM, KNN, LSTM, and DNN, the best result is LSTM.	Electric vehicle charge management systems.	Require a comparison of their results with other systems, especially due to a lack of data.	UGV
Mazzia et al.2020	Create an autonomous global path capable of covering the entire crop extension.	feature learning fully convolutional model.	Improving crop quality and seasonal productivity.	Lack of real data - skips various inadequate map regions that can affect the overall prediction.	UGV
Continued on next page					

Table 2.2 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Zeng. et al.2014	Define an eco-friendly route that produces minimal CO2 emissions while meeting a travel time budget (similar to the shortest path travel solution).	SVM has a high potential to minimize CO2 emissions, Heuristic KSP algorithm and resolves the environmental routing problem of shortest path travel.	PHEV and Autonomous Vehicles.	The model does not discuss the different emissions ratios that depend on the vehicle's driving condition at various speeds - Needs to consider other toxic gases such as NOx and HC.	UGV
YAO. et al.2016	Effective path planning method.	Artificial Potential Field method with DRL.	Effective Path planning method for automated vehicles.	Cooperating with different agents.	UGV

2.2.3 Hybrid Approaches

Many studies combine multiple methods and algorithms to achieve better performance than a single algorithm can achieve. Hence, modifying and combining multiple algorithms in optimization or ML techniques is an integration of both approaches. These hybrid methods can solve complex problems and overcome some problems, such as optimizing local and global search for collision-free and avoiding local minima problems. In addition, it designs an ideal space, improves convergence, and produces better routes. Table 2.3 summarizes the hybrid approaches in both types of EVs. The following subsections describe hybrid techniques used in some of the research on UAVs and UGVs.

2.2.3.1 Hybrid Approaches for UAV

Hybrid approaches tend to overcome the limitations of single or conventional technologies by combining efficient methods that can overcome the limitations in systems used mostly for dynamic environments or efficient drone navigation, etc. The main objective of VRP is to reduce the cost of the routing path, time, and the number of used and overloaded vehicles. Thus, meta-heuristics are applied to find fundamental solutions with less computation time and to enhance possible solutions. Similar to CVRPPAD, it allows only one trip with the possibility of alternate drop-off locations. In contrast, the model does not deal with constraints such as route length, number of route points, item prioritization, and timeout. Another study investigates the Flying Companion Traveling Salesman Problem (FSTSP) (Özoğlu et al., 2019) which is a new version of the traveling companion problem, that uses hybrid vehicles such as drones and regular trucks. Its main objective is to reduce the full delivery distance of trucks in the depot after loading all supplies. This method solves this problem via a hybrid approach including a GA and a well-recognized heuristic approach based on Clarke and

Wright's savings Algorithm. In turn, this approach may increase the truck waiting time for the drone, and there is no comparison with other similar studies. Ruan et al. (Ruan et al., 2013) proposed a hybrid approach integrating honeybee mating optimization and six-loading heuristics to solve a 3D Loading Capacitated Vehicle Routing (3L-CVRP) problem. This approach achieves the reduction of transport costs while serving all customer requirements, and ensures that the load-bearing capacity does not exceed the vehicle's loading space. These constraints, using Three-Dimensional Bin Packing Problems (3BPP), are the overall weight and loading space of items provided by customers that cannot exceed the vehicle's weight capacity and loading volume, respectively. Items have a fixed top and can be rotated 90 degrees on the horizontal plane, while inverted rotation is not allowed. Items are loaded orthogonally to the vehicle, with a fixed orientation for each item. The element stability requirements must also be respected, and the required support must be provided during the loading process. In addition, the Two-Dimensional Loading Vehicle Routing (2L-CVRP) Problem is utilized by meta-heuristic algorithms as a branch-and-cut algorithm, simulated annealing algorithm, tabu search algorithm, and memetic algorithm. The proposed solution solves VRP with Honeybees Mating Optimization (HBMO) and Three-dimensional Loading Problem (3LP) within six heuristics algorithms. The hybrid approach enhances its robustness and memory usage, as well as finding better solutions by 50% lower than others in computing time. Nonetheless, this study does not cover many constraints such as a heterogeneous fleet of vehicles, pickup and delivery, and time windows. Furthermore, on applying the Markov model (Nguyen et al., 2021) to control the reaction time and distance between UAVs, this method records more accurate metrics based on wind flow and turbulence than the SD method. It develops an automated system that constantly updates its data about location and timing via the Global Positioning System (GPS) along

with the Geographical Information System (GIS). It also improves trajectory tracking with various methods such as a multi-loop PID controller, sliding mode control, backstepping, Linear Quadratic Regulator (LQR), and Line-Of-Sight (LOS) angle constraint, which addresses initial large angle errors with acceptable system performance. In addition, both collision avoidance and navigation are important factors in this approach for the UAV system. Despite the exact method's failure to find a solution for cases with more than 12 clients, it was impossible to compare SA performance in larger cases. Drones are used in the agricultural field to defeat birds that damage crops and affect productivity. the Age of Information (AoI) (W. Li et al., 2019) is a critical object for finding the best UAV path planning model by reducing the amount of expired data for the optimum path solution for the Min-Max-AoI problem. Due to minimization of the number of expired data packets, the number of each carried packet from each sensor is maximized. Path planning is improved by RL-based approach and its efficiency is evaluated through two benchmark comparisons. As a consequence of comparing their approaches with the two benchmarks used, the GA is found to be the best method to find the optimal trajectories, while Q-Learning is better in time-consuming, and in contrast, the greedy strategy shows the worst-performing solution. In short, the certainty of the data is a very vital condition for selecting the best path according to the time and dynamic environment. It should be noted that this method is not experimented in real life.

2.2.3.2 Hybrid Approaches for UGV

UGV achieves a highly efficient system by using a hybrid approach based on ML and optimization methods in autonomous driving, battery saving, navigation, and object detection. Hybrid metaheuristic algorithms achieve high performance as a novel proposed algorithm (Schneider et al., 2014), such as integrating Variable Neighborhood Search (VNS) and Tabu Search

to solve Electric Vehicle Routing Problems along with Time Windows and charging stations (E-VRPTW). These algorithms can solve some constraints such as recharge times, capacity, customer time windows, number of EVs in use, and cumulative distance for all routes. Besides this search design, the performance measure includes two groups, one with small-scale instances through the CPLEX software, which enhances the results of the proposed algorithm, and one with large-scale instances. Evidence of the performance efficiency of this new algorithm is demonstrated compared to the performance results of algorithms designed to solve other similar problems, such as MDVRPI and G-VRP. Optimal route planning plays a crucial role in various applications, such as autonomous driving in EVs. These systems need to include some constraints that validate the operative approach. These constraints are energy management, cost reduction, path learning, and speed control. You. et al. (You et al., 2019) proposes a model that improves the communication between the electric autonomous vehicle and adjacent vehicles in traffic, which enhances the path planning of the driving strategy to find the optimal route based on some constraints. The Markov Decision Process (MDP) is used to create a policy addressing this problem to design a reward function using the RL approach. Furthermore, Inverse Reinforcement Learning (IRL) is used when the reward function is ambiguous, thus it uses data from previous driving experiences. Following that, the Maximum Entropy Principle (MEP) is used to match each reward with its feature to learn a parameterized feature reward function, and for unknown rewards, the Deep Neural Network (DNN) is used. The Q-learning algorithm is also used to observe typical driver behavior, such as tailgating and overtaking. The performance of this proposed method is demonstrated with simulated results. Moreover, the difference between RL and IRL is that RL maximizes the reward based on the sequences of current states to take an action. On the other hand, IRL learns from the behavior of the agent to take states and

actions, which are called demonstrations. Despite the qualities of the proposed model, it does not distinguish between different vehicle velocities nor experiment with real-world driving scenarios. To provide the optimal route, some research (Lu et al., 2019) has combined route planning algorithms to satisfy traffic and charging costs based on various trip objectives via an application in which user input has been entered. To reduce the total tour distance based on the allocation of the GRU, a neural network is used to set the time window for the price of electricity, and to predict the optimal path depending on the requirements. EV user journey objectives can be divided into in-situ charging, close-range travel, and multi-site travel.

The hybrid approach research has several major shortcomings, including its incompatibility with most real-world experiments. It also lacks comprehensive benchmark data, and most studies do not address complex situations and constraints as encountered in real life. Furthermore, some models do not take into account the more restrictive requirements of real-world systems, such as fleet coordination and multi-state energy consumption, which are coping with increasing carbon emissions. More consideration of the best approach to hybrid systems in both traditional UGVs and UAVs, without optimizing and choosing one over the other.

The first two stations are within their range and explain the condition if there is no charging, and the last is that the user must stop at more than one stop to complete the journey. As a result, the route planning algorithm has a better impact on travel time and predicts estimated charging rates in real time within user goals. Although spot pricing is not presented in detail in this thesis, and similar to the previous approach, there are no experiments with real-world driving scenarios. Traffic congestion is one of the vital factors that can affect the efficiency of optimal path planning. Although the shortest path does not mean the optimal path, this is due to other limitations, such as the speed of electric vehicles and the high traffic congestion.

In general, these routing approaches used in EV systems can disclose some advantages and disadvantages based on the goals of the systems and can determine which are the best approaches that can meet the desired needs. Table 2.4 shows the pros and cons of some main methods used in this survey.

Table 2.3: Summaries of Hybrid Approaches

Ref	Approach	Methodology	Application	Limitations	Type
Özoğlu et al. 2019	Solve FSTSP (Reduce the total distance of delivery).	A hybrid approach based on Clarke & Wright's savings algorithm and GA.	Customer service delivery.	Increased truck waiting time, which is the main drawback of this proposed hybrid algorithm, and the lack of comparison with similar research.	UAV
Ruan et al. 2013	Solve Three-Dimensional Loading CVRP (Meet customer demands).	Hybrid approach which combines Honeybee Mating Optimization and six loading heuristics.	Customer service delivery.	It does not cover many constraints such as a heterogeneous fleet of vehicles, pickup and delivery, and time windows.	UAV
Continued on next page					

Table 2.3 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Li et al. 2019	Propose a UAV trajectory planning model for data collection while minimizing expired data packets in the entire sensor system.	RL-based and among the three algorithms, GA-based, Q-learning, and greedy strategy.	UAV trajectory planning model.	This method has not been experimented with in real life.	UAV
Schneider et al. 2014	Solve the problem of EV steering with time windows and charging stations (E-VRPTW) - (Cost-optimal routes).	The hybrid heuristic combines a variable neighborhood search algorithm with a tabu search heuristic.	EV for last-mile deliveries.	It can enhance the accuracy of the steering model by considering the loads, degrees, and speeds of the vehicle.	UGV
Continued on next page					

Table 2.3 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
Ramesh et al. 2014	Autonomous driving has the potential for technical innovations in a multidisciplinary approach (Power management - speed and time control).	<i>DL</i> is used for localization of vehicles, path planning at a high level, and path planning at low levels, and RL and transfer learning to speed up the process of gaining real-time business intelligence - Recurrent Neural Network-Fuzzy logic.	EV with autonomous driving system.	The results need to be compared with other similar methods.	UGV
Continued on next page					

Table 2.3 – continued from previous page

Ref	Approach	Methodology	Application	Limitations	Type
You et al. 2019	A proposed model for the planning problem of autonomous vehicles in traffic.	MDP -RL and IRL techniques - DNN - MEP - Q-learning algorithm.	Autonomous vehicles in traffic.	Need to experiment on the real-world driving scenario- The model does not distinguish different vehicle velocities.	UGV
LU et al. 2019	Providing a route planning algorithm based on spot price and user Objectives.	Integrated route planning algorithm - SAA is combined with A* algorithm and Dijkstra algorithm GRU neural network.	Route planning algorithm.	The spot price is not provided in detail in this thesis Need to experiment on realworld driving scenario.	UGV

Table 2.4: Pros and Cons of Some Main Methods

Approaches	Advantages	Disadvantages
Dijkstra algorithm	• Reducing time consumption. • Low complexity.	• This approach gives the shortest path but is not guaranteed to be the optimal path. • Cannot manage negative weights. A* algorithm.
	• Higher efficiency. • Find the lowest cost. • Find the shortest path.	
PSO	• More computation time. • A continuous path cannot be achieved.	
	• Highly accurate. • Easy to implement. • Fast convergence completeness.	• In high-dimensional space or global search, it could have a local optimum. • low convergence speed.
GA	• Solving optimization problems with effective results. • Efficiency. • Robustness.	• Can be trapped at the local optimum when using an inappropriate fitness function that makes GA an expensive and incomplete solution.
Continued on next page		

Table 2.4: Continued on next page methods

Approaches	Advantages	Disadvantages
ACO	• Suitable for the predetermined problem and goal, which quickly finds the appropriate solution and prevents premature convergence. • Suitable for dynamic environments.	• Increase the search complexity. • The convergence time became uncertain; therefore, ACO has the longest simulation runtime.
ANN	• Find the optimal solution within the interconnected network based on learning and error convergence. • Optimize complex environments.	• More extensive datasets for training. • High computational power with long-time complexity.
Heuristic Techniques	• Solve problems in the fastest route and reduce cost while finding the shortest path to ensure system efficiency.	• The accuracy of the solution could not be optimal. • Bias can occur with the final solution.
MDP	• It performs actions at any time and at any location randomly to find the best path.	• Actions and rewards are affected only by the current situation, not by history, which affects the entire solution.
<i>Continued on next page</i>		

Table 2.4: Continued on next page methods

Approaches	Advantages	Disadvantages
TS	• The ability to overcome a local optimum problem. • Tabu list used to avoid reuse of old solutions.	• Tunable parameters can be challenging. • It can include high iteration.
RL	• Used for efficient end-to-end planning and complex data computation. • High performance and durability. • Balance exploration and exploitation.	• High computation. • It requires a lot of important data.
Floyd algorithm.	• Handling negative edge weights and graphs with cycles and a large number of nodes in the graph.	• High-time complexity. • Great memory requirements.
SA	• Finding the best path and fastest convergence. • Dealing with noisy and complex data.	• High-time complexity. • Tunable parameters can be challenging. Q-Learning algorithm
Continued on next page		

Table 2.4: Continued on next page methods

Approaches	Advantages	Disadvantages
• Effective for high-dimensional space and dynamic environments, without the need for prior data knowledge for the system environment.	• High computation. • Long time to train and find the optimal solution.	
SVM	• Shows robustness to handle overfitting and high generalization problems.	• It cannot handle large or noisy data.
	• Reduce the cost of travel time.	

2.3 The UAV Methods Used in the Routing Problem

The UAV methods used in the Routing problem section are divided into three subsections, demonstrating the use of the main proposed thesis models. The first section discusses individual approaches to the routing problem, which are GA and RLs algorithms. The second section is based on hybrid approaches to the routing problem using GA and RL across multiple scenario types, and the third section discusses multiple approaches using multiple UAVs.

2.3.1 UAV Individual Method used in Routing Problem

Routing problems can be solved efficiently using several methods. This subsection explains the algorithms used in the proposed model in the following subsections.

2.3.1.1 Genetic algorithm

The WCS has been gaining significant attention recently. Several researchers investigate the best solutions for efficiently finding a sequence of deployment locations that connects a starting node and a target node. The efficiency of this sequence in the optimal or near-optimal path is determined by its satisfaction of overcoming their constraints. Hence, using a GA as an optimization technique that can adjust parameters and hyperparameters to find optimal solutions that minimize constraints. GAs are types of heuristic optimization methods (Sabet & Farooq, 2022), which are defined as part of nature, as based on biology, a social behavior that mimics natural behaviors and can improve the efficiency of many systems that use these methods as paths planning. There are plethora of studies investigating WCS. A case

study showed a reduction in the total cost of a solid WCS (Assaf & Saleh, 2017) by finding the best routes with the least distance to solve VRP in Nablus City, Palestine. GA was used in this model and a distance reduction of 66% was observed compared to the conventional path. However, this study addresses the static solution only and does not account for any accident that could occur. Similar to this study other studies confirm GA's superiority when used in SWM (Bhargava et al., 2019). GA is recognized for its ability to provide coverage which can guide the system in the right direction. Moreover, GA can be used to support the dynamic WCS which can modify the path based on adding emergency locations, thus optimizing the path cost and waste capacity (Dolinina et al., 2019). A GA method has been improved that can solve the TSP (Sonmez et al., 2015) and the VRP. This method does not support dynamic methods in real life; however, it can determine the best path for the UAV. Alternative systemic review research (Zhao et al., 2018) examined 231 studies published from 2008 to 2017 indicated that the most efficient algorithms used in the field are GA, PSO (Zhou et al., 2021), ACO, FL and LB method. Furthermore, researchers have studied several constraints addressed by GA, such as energy consumption reduction (Zhao et al., 2018) based on safe paths using GA. It generates multiple seeds representing each gene to find the best and shortest journey path. Similarly, another study (Zhou et al., 2021) investigates the relationship between the shortest path constraint and the power state of an EV which affects the capacity endurance of the EV. Power status affects many factors such as waiting time and queue length. The proposed method reduces travel time by up to 6.5%. The research analyzes some of the factors that affect the model, noting that higher traffic volume will increase the route length because the alternative route with more distance can be selected, hence the energy consumption will also be increased. In addition, the temperature will have a greater impact on the driving path, for when the temperature decreases, the

vehicle's endurance capacity decreases, and vice versa. In addition, when the speed increases, the travel time is reduced and the average energy consumption is increased (Platanitis et al., 2020). Whereas implementing a simulation map for a test environment involves only constant sampling, on the other hand, using variable samples will increase the efficiency of experimenting with results. In another study, a GA was used to satisfy the multi-task assignment problem (Q. Liu et al., 2021) and multi-objective method addressing constraints such as time management and reliable paths. The study concluded that GA finds near-optimal solutions faster than other methods. As an Improved Multi-objective Genetic Algorithm (IGA) (D. Wang et al., 2023); this function adds a point between two adjacent networks to fill the path between two points to initialize the population. Moreover, it uses a multi-objective fitness function that includes path safety, energy consumption, and path distance. Comparison of IGA with traditional GA shows 17% better performance based on path length. However, this modified GA is required to further improve the smoothness of the planned security path because the algorithm prioritizes the best short path compared to other constraints which necessitates better fitness function design. Another approach has been studied using six 3D environment models (Zhao et al., 2018) and the best approach reduces up to 70% of the function cost requirements used in the cluster centers approach for collision locations. The type of selection model influences the generation of the population through a possible path to solving the VRP. However, accurate prediction in this method is crucial whereas based on cluster centers having higher waypoints may cause poor predictions. Hence, the role of cluster centers is vital. One study proposes a new selection method (Yusuf et al., 2014) and compares it with roulette wheel selection to solve the VRP problem. This method minimizes fuel consumption at maximum load while providing maximum port numbers. It shows that the proposed selection has 20.38% lower fuel consumption. It

uses two types of crossovers and mutations. Experiments show that roulette wheel selection gains are smaller than those of the new selection. Moreover, the single-cut point crossover is less effective than the multi-cut point crossover. Also, the pair exchange mutation is less than the shift neighbor mutation. This method is tested on simulations and not on real systems. This algorithm uses three local search methods, namely 2-opt, Insertion-move, and interexchange move, which lead to finding new optimal solutions and addressing different constraints simultaneously to aim for low cost and number of EVs used in the system. This method finds the optimal solution based on the appropriate new solutions. When calculating LINGO13, it records a higher cost even if the location is suitable. In research conducted in the city of Mecca, Saudi Arabia, GA was applied in WMSs and showed its capability to improve dynamic routes. The results showed that GA successfully reduced costs profoundly by addressing multiple objectives, such as consuming lower fuel, finding the shortest path, and spending less computation time. However, while GA does not always guarantee finding the optimal path, it constantly identifies the best or near-optimal routes (Alwabri & Kostanic, 2020). Similar research applied GA in SWM (Bhargava et al., 2019), taking into account factors such as vehicle capacity and time constraints to improve routing schedules. Furthermore, many WMSs leverage IoT tools within smart city systems to verify the best routes. A case study conducted in the Bakirkoy district of Istanbul examined GA's ability in path planning, revealing that it is superior to the traditional methods in optimizing routing solutions (Aktemur et al., 2020). A novel GA (Farahbakhsh & Kheirkhah, 2023) can solve the Inventory Routing Problem (IRP) to meet different customer requirements within a specified period of time based on incorporating a new chromosomal structure designed to represent solutions efficiently. Another case study was used to solve the Asymmetric Capacity Vehicle Routing Problem (ACVRP) in Istanbul delivery operations that de-

liver bread daily using GA to improve delivery process efficiency (Meniz & Tiryaki, 2024).

2.3.1.2 Reinforcement Algorithm

Routing problems can be effectively solved using several methods. One such method is the RL algorithm, which is well-compatible with dynamic environments, hence tolerating it to adjust and fine-tune its parameters during the training phase. This capability enhances the DRL algorithm; particularly for simple search tasks, due to its efficient operation of state-action relationships. RL is adept at multi-constraint problems providing multi-objective solutions and does not require prior data, making it adaptable for complex environments. In one study, DRL (Yan et al., 2020) was applied to address UAV path planning problems using a simulation model known as STAGE scenario software. This method handled the situation as a survival mode; granting the avoidance of enemy attacks in both static and dynamic modes. The RL framework is used exclusively with various types of algorithms, such as the D3QN algorithm. In a study that used RL-optimized quadcopter UAVs that consider specific types of UAV through a supplementary controller (X. Lin et al., 2019); it thus utilized the DRL algorithm to solve many problems, such as the Autonomous Motion Planning (AMP) problem. DRL is powerful in artificial fields for path planning, tackling static and dynamic situations, and overcoming complex environmental barriers by merging with the BHPF to allow agents to collaborate and avoid obstacles (Yao et al., 2020). Additionally, deep algorithms such as Double Deep Q-Networks (DDQN) and DRL optimize UAV routing with advanced solutions. To obtain information from the constrained and complex environment and to avoid obstacles; a Deep Recurrent Q Network (DRQN) (Xie et al., 2021) emerges with the Markov decision process and recurrent neural network and Convolutional Neural Network (CNN) for spatial feature

analysis. This method combines current rewards, states, and actions to decrease unnecessary exploration. To address the challenge of dynamic vehicle routing (Joe & Lau, 2020), a research study used DRL, which included neural network-based temporal difference learning along with the Markov decision process to solve customer routing issues. This method showed improvement in the value function and enabling better re-routing decisions involving the ability to switch and insert customer locations. The study introduced DRLSA, based on simulated annealing, and compared its performance with Approximate Value Iteration (AVI) and the Multiple Scenario Approach (MSA). These methodologies assist in calculating costs based on remaining locations and time window requirements; henceforth further expanding the efficiency of dynamic routing solutions.

2.3.2 UAV Hybrid Methods Used in Routing Problem

Many studies highlight the benefits of combining different approaches to synergize the advantages of each individual approach of them. One such study efficiently combines ML and optimization algorithms to improve outcomes in a garbage management system (Fathurrahman et al., 2021). This research applies a deep learning platform known as DenseNet121; operating it at an image database sourced from Google Sheets. It groups images into six discrete garbage sorts, hence achieving a remarkable accuracy of up to 99.6%. A GA is engaged to enhance the model via running through four generations to find the best solution. This model is implemented in mobile applications that not only detect the type of garbage but also determine its location using GPS. The system establishes exceptional performance even with a data transmission delay of less than one second. Building on the emphasis of synergy of different algorithms, another study investigates the enhancement of a ML-based model via the optimization of hyperparameters, which are essential for model performance (Quevedo et al., 2021). The

study suggests a model that merges RL and GA algorithms. In this setup, the GA takes actions from RL to generate new generations; that number of generations is determined by the rewards. The learning rate; a key hyperparameter, controls the current Q values. As the GA adjusts the learning rate, RL ensures that its impact on the learned Q values does not lead to a failure in fitness. The model is trained using a dataset from the CVRP benchmark. CVRP is a classical optimization routing problem that minimizes the total travel distance based on several constraints under the capacity condition. When this approach is compared to models with static parameters, it achieves up to 11% better results while enhancing population diversity. Li et al. (W. Li et al., 2019) compare two methods, the RL and GA specifically in the agricultural sector. The study focuses on improving drone path planning by minimizing the Age of Information (AoI), a vital factor in lowering the amount of outdated data. The findings depict that while the GA excels at identifying the highest effective paths, the RL method demonstrates to be more efficient in achieving the lowest time consumption. Concurrently, these studies prove the ability of integrated algorithms to solve complex routing problems successfully and more efficiently. Kaabi et al. (Kaabi et al., 2023) propose a novel VRP for hazardous WCM. The constraints addressed are the same as in our study, such as drone capacity, flight time, and distance. Conversely, the method is used inversely from our proposed method as it is a two-stage process with a new linear program called Maximum Waste in MWMTT. The first stage is used to create a tour with the maximum number of WCLs within the shortest distance, and the second stage is used to assign tasks to different groups. The data scale used in that experiment is split into small group cases between 10-40, medium group cases 41-48, and large group cases 49-89. The performance is evaluated based on a narrow minimum threshold. The research results revealed various routes that start and end at a depot and when the drone is almost out

of power or the waste capacity load is almost full, it travels to the disposal point and then to the depot, noting that it is unable to recharge until the group trip is complete. A new route is then started to complete the remaining location trips. Abdulsattar et al. (Abdulsattar et al., 2023) study has similar constraints to the previous study and its main objective is to increase the drone's payload capacity and reduce the flight path distance and time. The study uses a similar dataset as the previous study of MWMTT and has the same experimental problem. The study adds an improvement based on the previous study and redefines it as the Improved MWMTT (IMWMTT). These improvements are shown in two stages; the first stage is to optimize a group of flights of the UAVs that are recharged only after completing each group task without the need for frequent recharging. The second stage is to combine MWMTT with the ACO algorithm to find the shortest flight distance and find better routes. This approach ends the flight sooner if the drone's capacity load is almost full or when it needs to be recharged, hence it goes to the waste disposal and then to the depot and thenceforth continues to unvisited waste locations. Despite the promising results from previous studies, there are notable disadvantages associated with existing approaches, specifically regarding increased computational time. Additionally, many researchers rely heavily on simulation environments without validating their findings through real-world applications or adequately testing them in complex environments. It is also crucial to find a balance between exploration and exploitation within proposed models. In the same context, the exploration is to explore new solutions, and if it's explored exhaustively, it could increase training time without efficient outcomes. On the other hand, if the exploitation focuses on current knowledge, it could miss better solutions, since that of balancing exploration and exploitation could be a perfect solution.

2.3.3 Approaches for Multiple Drones

WCSs in smart cities require many intelligent systems that improve human health and enhances the quality of life and economy. Therefore, researchers investigate the path planning problems and seek improvements that are fundamental in WCSs. This literature explains some methods and techniques used for multiple UAVs tasks which are of interest to the realization of this thesis work such as RL, GA, and adding a combination of other algorithms such as k-Means and A* to the integrative hybrid model of GA and RL (GARL) for further tuning of UAVs tasks in its pursuit to solving the routing problem. Several studies investigate the multi-drone approach. A study explores (Nguyen et al., 2021) the drone control trajectory based on drone safe distance, autonomous control, and conflict detection with possible solutions in transportation in smart cities. Safety and security aspects of drones are discussed by setting speed limits and using predefined fixed routes to ensure that drones do not collide. Most of the methods uses sensors to detect objects to avoid obstacles. It is proposed to use a barometric, multiloop PID controller and an infrared (IR) camera to manage the altitude control. Moreover, the trajectory tracking control problem can be solved through backstepping, multiloop PID controller, LQR, and more methods. Swarn drones can help manage drone collaborations and set Save Distance (SD). The Markov model can control distance and reaction time. Wu et al. (Y. Wu et al., 2021) conducted an experiment using a comprehensive approach to collaborative path-planning problems using multiple UAVs in persistent surveillance. To plan a path for each UAV it is divided into a space grid forming cubes and the coverage area for each UAV is determined. The design formula provides safe space between UAVs and prevents collision and coverage area overlapping. The goal is to achieve persistent surveillance of the entire urban area; especially for the grid that has not been visited for a long time, thus requiring maximization of the sum of state values

and gaining more monitoring priority and consistency over time. Therefore, the Multi-Criteria Collaborative Path Planning (MCCPP) algorithm is proposed. Another solution is the emergency path planning algorithm that maximizes the coverage area and satisfies the timing and shortest path constraints. Another promising approach is RL algorithms that cooperate with the attention model used in multi-UAVs (T. Wang et al., 2021) which consider multiple factors such as survival probability, short path, capacity, and energy constraints while examining large amounts of data. The attention model is a framework created by Google in neural networks that uses drone reconnaissance strategy, transformer structure, and parallel operation which improves the speed and performance of the training process. A novel DRL (Zhu et al., 2021) method is also used to resolve multiple UAV trajectories for Wireless Sensor Networks (WSNs) to reduce energy consumption. The model is inspired based on the DRL method, achieving high performance against Neural Network (NN) and GA. Hence, the predefined clusters model PtrA* is designed to solve the decision-making order of visit locations because it is used to well determine the order of visit locations which builds the graph. The A* search algorithm reduces training time because faster Cluster Heads (CHs) need less searching as it reduces energy consumption. Another study uses RL in their system for multi-driver vehicles (Holler et al., 2019), and RL would be effective in the part of learning to dispatch and reposition orders. The system achieves great results by using a driver-centric system where each driver maximizes their rewards and then cumulate all the rewards for all drivers together. Furthermore, the driver-centric Deep Q-Network (DQN) software used in training improves the stability of the model. Zhang et al. (W. Zhang et al., 2020) proposed a QRewriter DDQN decision dispatching algorithm that makes real-time decisions to allocate resources such as drones to handle specific tasks based on high level requirements while coordinating with other drones. This proposed algorithm con-

tains two techniques; DDQN and QRewriter module that adjusts parameters of neural network and Q table to maximize rewards and can rewrite inputs to optimize DDQN dispatching policy that ensures the entire performance of the algorithm is tested using a designed simulator. A multi-layer (Cui & Wang, 2021) based on RL path planning algorithm for multiple UAVs is investigated. The proposed algorithm consists of a higher layer that handles local data and a lower layer that considers global information. To illustrate; local path planning deals with unknown information using online policies such as PSO and fuzzy logic. In contrast, global path planning implies that the information is known, which typically uses offline policies such as the Voronoi diagram and potential field method. Both layers are connected to plan a collision-free path. Furthermore, a Bspline curve approach is used to add smoothing maneuvers to the UAVs. Additional research addresses the large-scale fleet management problem (K. Lin et al., 2018) used by DRL based on the proposed contextual multiagent RL framework divided into two contextual Deep QNetwork (cDQN) and contextual Advantage Actor-Critic (cA2C). The cA2C designs a centralized value network to estimate long-term rewards and a decentralized policy to adapt to different actions without modifying the overall framework, where each agent has its execution policy. This approach increases flexibility and scalability to deal with different environmental situation. Multi-UAV autonomous path planning algorithm with a RL-free model (Y. Chen et al., 2022) relies on a centralized training process for all UAVs instead of training each UAV separately and then coordinating each action together. Centralized training improves the decision-making process between all UAVs to achieve better cooperation. In addition, a RNN is added to obtain historical information and prevent the model from getting stuck in the local optima condition. The reward function is designed to handle multiple objectives; hence the model can adapt to dynamic environments. The GA is investigated for high per-

formance in solving the routing problem in multiple UAVs. Using a novel algorithm called Genetic Trajectory Planning Algorithm with Variable Population Size (GTPAVP) (Asim et al., 2021) which solves the problem of Internet of Things Devices (IoTDs) deployment using UAV-assisted Mobile Edge Computing (MEC) system. It aims to reduce energy consumption by reducing UAV flight distance and hence time. The proposed model consists of two stages; the first is to obtain variable population size to update the deployment of Stopping Points (SPs), and the second stage is to find the number and order associated with SPs of UAVs to optimize the best path. An Adaptive Genetic Algorithm (AGA) (Xiong et al., 2023) is developed by adding the Sine Cosine Particle Swarm Optimization (SCPSO) method in disaster rescue environments using multiple UAVs. AGA preprocesses the tasks of UAVs with an adaptive crossover and mutation rate based on fitness change. In addition, roulette and elite retention are used to improve the efficiency of AGA. On the other hand, SCPSO plans an optimal path through the search space to explore global optimization better. To create a powerful method, different algorithms are associated together, hence getting the advantage of compounding powerful methods. Several researchers have explored the combination of RL and GA algorithms. Song et al. (Song et al., 2022) proposed a multi-vehicle-assisted multi-UAV Path planning algorithm (MVP) with Mobile Crowd Sensing (MCS) technology which is important for navigation control. It includes an Efficient Heuristic Allocation algorithm (EHA) which considers the route design of ground vehicles. On the other hand, GA and RL is designing the route for multiple UAVs. This model reduces the number of vehicles and time needed and increases profits. To solve dynamic environmental problems, a proposed Neural Genetic Algorithm (NGA) (Hamid et al., 2023) model is used by combining GA and neural network. The exponential based fitness function and the inverse time variable are modified, which achieves faster convergence and

performs better in dynamic environments. To illustrate exponential-based fitness, it combines linear and exponential components that can increase visibility, hence any small change in the input can effectively change the fitness function output. In addition, the inverse time average takes 1 divided by the time value in the fitness function, which allows the Unmanned surface vehicle (USV) to move in a shorter time, perform better for the fitness output, and reduce the time for the entire tour. Additionally, the deep learning algorithm combined with the GA achieves high results in solving multiple drones routing problems. Deep Learning trained by GA (DLGA) (Pan et al., 2021) is proposed, which combines GA and deep learning to solve the path planning problem of multiple drones to collect data scenarios. In this model, the GA is generating the path planning of multiple drones, stores it in the experiment replay and then uses the sample to train it in CNN. Since there is no convergence process during CNN optimization, the computation time is reduced. A newly proposed method for solving multiagent planning problems for autonomous package delivery investigates DQN (Tarhan & Ure, 2024) and includes several methods. One method is decomposition, which is a process that breaks down tasks into subtasks and solves each separately and then combines them through an approximate function and deep correction, which leads to decent learning stability. Another method is Curriculum Learning, which is a technique that gradually improves learning by starting with learning simpler tasks and then increasing the complexity gradually to more complex tasks for effective learning. It also prioritizes experience replay to increase convergence by selecting the high-priority experiments, hence GA is evaluating and optimizing each policy based on the rewards received through the fitness function. Li et al. (Y. Li et al., 2019) proposes a two-level framework to solve the problem of autonomous trajectory generation for obstacle avoidance and complex environment handling using multirotor UAVs. This framework consists of a higher level of locations

from the current position to the destination and a lower level between two adjacent locations. DRL is used to learn the interactions between these two levels, and GA tuning process can find the optimal path and minimize the control energy. To solve the task compilation problem using a multi-UAV system (Tang et al., 2023), a new Digital-Twin (DT)-assisted task assignment and DRL algorithm is proposed. The approach consists of initial task assignment and task reassignment. The task assignment is divided into sub-tasks based on the area distance which is processed by GA. While the task reassignment is based on learning the task features using DT, and then this process generates a Q value which is applied to DQN to reduce the training time and improve the completion ratio under the time constraint. Despite the promising results of previous studies, there are notable drawbacks associated with the existing approaches, particularly concerning reducing energy consumption and supporting dynamic and complex environments. Also, in most of the recent works related to WCS, less constraints are considered than those addressed in this thesis model. Moreover, there are very few works related to WCS using multiple UAVs and most of the existing work related to WCS is focused on single UAV. Furthermore, operating multiple UAVs routes requires more sophisticated techniques than a single UAV needs to achieve balance and stability between UAVs, which is ensured by this thesis proposed model.

2.3.4 Summary

This literature review has presented several studies that focus on finding best practices for optimal or near-optimal solutions utilizing different algorithms aiming at improving the waste collection process through identifying the best appropriate tours that can minimize the cost. Most of the available research examines those algorithms in different ways, which shows several drawbacks such as exhaustive computation time and some of the research

neglects to support dynamic methods in real life, which limits their practical application. Furthermore, over reliance on the experimental simulation implementation using static sampling dataset may hamper the method's performance when applied in real-world. In addition, current approaches often overlook or fail to adequately address the main challenges associated with WCSs. Moreover, in even more complex problems, the efficiency of these approaches in managing WCS can be reduced. Considering these drawbacks and challenges in other research and signifying the demonstrated insufficiency of study in this field, this motivated the proposed work which aims to deliver a new valuable insight and proposes contributions by introducing new valuable and applicable approaches with optimal or near-optimal results and added future contributions by introducing new efficient approaches into the WCS. The experimenting process in this thesis depends on gradually evolving in complexity to gradually substantiate the application efficiency of the algorithms while tested in different measures. The experimentation gradual increasing complexity employs performance efficiency comparisons in different measures; i.e. from single algorithm to hybrid integrative algorithms; from lower to higher number of WCLs; comparing different sequence of algorithms employment in the hybrid models in their respective phases, eg. GARL or RLGA; and then exploring the UAVs performance efficiency in regard to the increase in number from single to multiple. Hence, this thesis first presents an approach that uses a multi-objective GA method which addresses constraints in the initial population, uses multi-objective fitness functions, and employs different search methods; especially incorporating two types into the mutation process. Overall, the proposed GA aims to achieve decent solutions to vital complex problems. Then presents a proposed study that introduces a multi-objective model which addresses diverse constraints through the combined application of GA and RL algorithms. The proposed integrated model aims to

yield feasible solutions, where it utilizes the optimal path identified by GA as an initial state for ML algorithms. Hence the algorithms evaluate their rewards under the same fitness function constraints. This approach not only contributes to a deeper understanding of algorithmic incorporation in WCS, but also aims to improve operational efficiency in the field. This merging of optimization approach and ML methodologies intends to improve the model's effectiveness by synergizing the strengths of both approaches. Thus, the relative lack of studies utilizing this approach influences our proposed study to employ GA and RL algorithms in WCSs, hence proving the importance of the proposed study. Also, in all the previous studies, none has combined the approaches adopted in this work to utilize each approach's power synergistically in WCSs. Moreover, this work advances further by introducing a third study employing a combination of optimization and ML approaches with clustering algorithms. This combination aims to improve the effectiveness of the model by synergizing the strengths of all of these approaches. The advantage of using different algorithms and the use of each type enables them to support practical applications and improve the performance of WCS models. Thus, again the relative lack of studies using this approach influences our proposed study aiming to reduce cost while minimizing energy consumption, flight time, and total route distance. Hence, this proposed study presents a multi-objective approach that addresses the above-mentioned various constraints through the integrative application of GA, RL, K-means and A* search algorithms. This model approach utilizes GA in the first stage to find feasible path solutions and then uses the optimal or near-optimal path determined by the GA as the initial state for machine RL algorithms in the second stage. In addition, K-means and A* search algorithms are used in both stages to distribute tasks among multiple drones, ensuring the best efficiency and minimal cost of the WCS model.

Chapter 3

Unmanned Aerial Vehicle for Waste Management System Using Genetic Algorithms

3.1 Introduction

UAV is used as a waste collection vehicle (WCV) ensuring that it effectively collects waste, services all WCLs and disposes of it in a central landfill or a treatment and/or recycling station. The importance of qualifying the WCS (Kaabi et al., 2023) in assisting, protecting and preserving the environment from pollution threats and mitigating the effects of climate change is evident. Hence, improved WCS provides an improved quality of life through improving overall health by preventing the spread of rodents and diseases. Moreover, the need for reducing transportation costs, which account for almost 70–80% of the total WCS costs (Fang et al., 2023). This chapter il-

illustrates a proposed model of multi-objective GA, which is one of the most powerful optimization methods that can effectively deal with complex problems, especially to solve routing problems and provide robust solutions that can work simultaneously with multiple constraints to find optimal or near-optimal solutions (Pajares et al., 2018). It focuses on the WCS requirements to apply multiple objectives methods (Hou et al., 2021) based on implementing a powerful multi-fitness function addressing various constraints, such as minimizing cost, trip time and path distance, while conserving energy (Beigi et al., 2022). Hence, this proposed model aims to address challenges through exploring and analyzing the potential solutions to increase the efficiency of WCVs in collecting and delivering all waste from all WCLs.

3.2 Problem Definition and Description

This study investigates the problem of WCSs and proposes solving it with GA using a single UAV. Hence, this approach aims to find the best UAV route with the shortest distance via GA. The system entails collecting waste from various WCLs starting and ending at the same depot. UAV's capacity, energy, and trip time limits based on the drone's charging endurance, are considered. The drone must visit all WCLs. When power runs low, it must go to a disposal location to empty its waste load and then return to the depot to recharge before continuing the tour. Likewise, if the UAV's capacity is nearly full, it should go to the disposal location to empty the waste load and then continue the tour. The optimization problem graph is modeled based on the graph $G(V, A)$, where V is the set of nodes $V = \{v_0, v_1, v_2, \dots, v_n, v_{n+1}\}$ where $v_0 = D$ and $v_{n+1} = S$, D the depot where the drone starts their tour with a full charge and empty space. It returns when the drone needs to recharge, finishes the tour, or meets all N locations of the waste nodes. Each node will be visited only once and has a positive weight which is the amount

of waste collected at a given location W . When the drone is full, it will release waste into the disposal location S . Additionally, $A = \{(i, j), d_{ij} \mid i, j \in \mathbb{N}, i \neq j\}$, while A is the set of edges connecting nodes. Each edge is defined as the distance between nodes set as i to j with nonnegative value d_{ij} . A drone traveling from one place i to another j consumes a certain amount of energy which can be represented by $e_{ij} = r \cdot d_{ij}$, where r denotes the rate of energy consumption. When the energy is almost exhausted, the UAV must go back to the depot to recharge, and before recharging, it has to dispose of its waste. Each UAV has a weight capacity of C . If it reaches full capacity, it has to unload at a disposal location S and then finishes its remaining tours. The other case is when the UAV's energy is almost depleted, then it goes to depot to charge. The drone has a maximum payload capacity that can be represented as MC . Moreover, maximum power capacity ME ; hence, these limits of the drone must not be exceeded. The UAV requires multiple flights to satisfy all WCLs with the shortest distance for all flights within a certain time T , which accounts for all routes, charging time, and unloading waste. Each waste location has a specific capacity indicated as WC . In general, this should reduce the cost and the total route distance. The problem is classified as an NP-hard problem (Baldacci et al., 2010) that seeks the optimal or near-optimal route for visiting specific locations, considering the capacity constraints related to the amount of waste to be collected and the drone's load and energy capacity. The main objective of the model is to determine the most efficient waste collection route between designated locations, achieving the shortest path in the shortest time. Additionally, the model ensures that the volume of collected waste does not exceed the drone's cargo capacity. If it does, the drone will offload the waste at the disposal location and then continues servicing the remaining locations. Once all locations are serviced, the drone returns to the depot. Overall, this approach aims at minimizing both the cost and total route distance. The most

Table 3.1: The Mathematical Model Notation

Symbol	Notation
G	Graph
V	A set of nodes
D	A Depot
N	Location of the waste node
S	Disposal locations
A	A set of edges connecting the nodes
d_{ij}	The distance between node i to node j
r	Energy consumption rate
MC	Maximum payload capacity of the drone
ME	Maximum energy
WC	Waste location in a certain node
C	Capacity of the drone
e_{ij}	The energy consumption between node i to node j
F_{ij}	The flight time between node i to node j
ds	Drone speed
fd	Flight duration
UN	Unallocated location

frequently used symbols in the mathematical model are listed in Table 3.1.

The optimization method includes the main proposed objectives, which address several constraints as follows:

1. Set the routing path map between different locations for a single drone to start its route from the depot and end at the same depot after it has fulfills all waste collection operations.
2. Find the shortest path within the route plan.
3. Determine the optimum predefined UAV capacity and flight time limits based on its charging capacity.
4. Determine the number of tours of the UAV needed to collect all the

waste from all the WCLs.

5. Identify the WCS via the drone's operations process steps:
 - UAV collects all the waste from all the WCLs.
 - When the drone energy is almost depleted, it must go to disposal and unload waste, then go to the depot to recharge, then continue the tour until it serves all the WCL.
 - When the drone reaches its maximum load capacity, the drone needs to go to the disposal location to unload the waste, then continues the tour until it finishes servicing all WCLs.
6. Calculate the efficiency when the UAV gives the shortest path of fitness with the least time to finish its routes.

3.3 The proposed Model

This section demonstrates the main GA work process (Han et al., 2019). To illustrate the proposed model of the multi-objective GA, which explains all the details of the process model operations and modifications from the traditional GA, in the following subsection.

3.3.1 Genetic Algorithms

The GA is the main operator of this proposed model. GA is applied in many applications as an efficient search technique that can improve the solution based on natural processes. The process begins with generating a random set of solutions called chromosomes, and these solutions are labeled as individuals. It includes all the information about the solution on the chromosome, which evolves from a series of iterations called generations. In each generation, the quality of the initial solutions must be evaluated by the

fitness function. The fitness function procedures involve different methods of selecting a chromosome from the parent population, and can crossover by exchanging pieces and executing a mutation process to create the next generation, hence repeating this process until the best solution converges.

3.3.1.1 Initial Population

The population in this problem is randomly selected to represent the entire tour, comprising several sub-routes representing some chromosomes and each WCL representing the gene shown in Figure 3.1. The sub-routes consist of several routes; each route defines the distance between several WCLs that starts from the depot and ends at the disposal; or it can start from disposal and end at the depot interchangeably. These paths or routes determine the entire tour. Each path or tour includes edges representing the distance between two WCLs; the differences between each two are shown in Figure 3.2. In this part, constraints can be added to the program from the beginning to ensure that the best path is selected. In other words, the routes begin at the depot with consideration for the constraints; i.e. when the drone is load-free with its maximum capacity, and when its battery is fully charged. The drone must end its tour by going to the disposal location, returning to the depot for charging and starting a new route to collect waste from the remaining locations. Hence, Algorithm 1 shows the initialization population algorithm that starts routes with the depot in light of these constraints. The drone must end its route by going to the disposal location, returning to the depot for charging and starting a new route to collect waste from the remaining locations. All these routes are referred to as sub-routes that combine to form the entire tour of the drone.

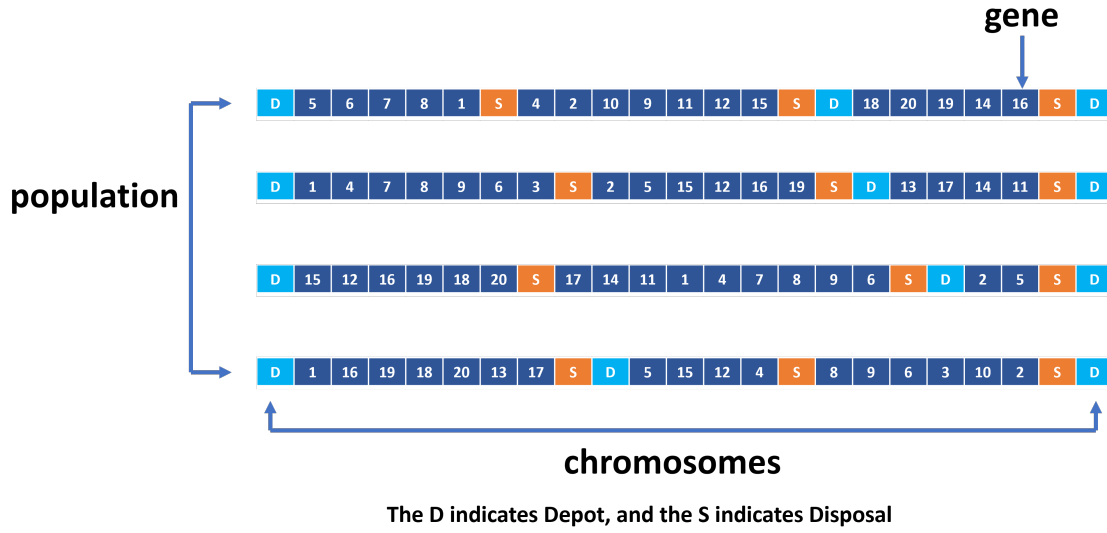


Figure 3.1: Initial Population.

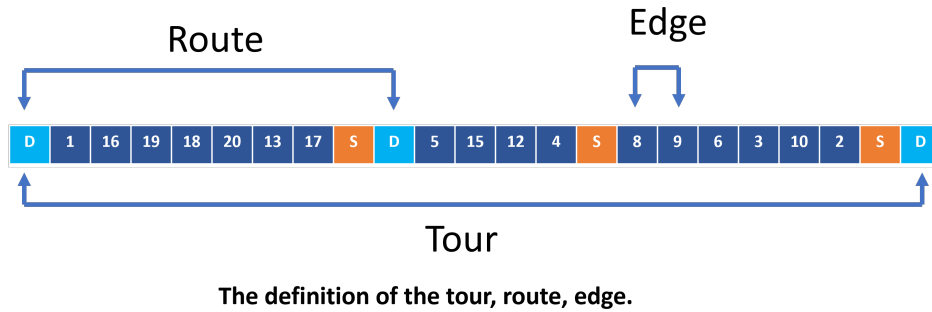


Figure 3.2: The Definitions of the Tour, Route, and Edge.

3.3.1.2 Fitness Function

In this problem, the multi-objective fitness function required to satisfy the main constraints is to find a short tour and manage the waste load capacity with UAV energy, as shown in Equation 3.1. To do this, one way is to combine them into a single fitness value; and it is important to note that all constraints have the same properties. The fitness function includes various constraints as a trade-off for the efficiency metric. Although we assign equal weights to all constraints in the cost function, which reflects a balance

among all constraints and factors, a single metric may lead to a degradation in the overall effectiveness of the fitness function. This approach is regarded as the best route despite its complexity in the calculations. In simpler terms, minimizing the total distance also minimizes the total flight time and reduces energy consumption. Maximum capacity is another factor that may not consistently impact the efficiency of the fitness function; it will only influence the fitness function when there is a capacity violation, necessitating a change in the route to the next node or a diversion to a disposal site to unload waste. Therefore, determining the best route requires a balance between minimizing the travel distance, reducing the flight time, and reducing the energy consumption while maximizing the collected capacity. This approach is shown in the sum approach equation, and each part of the equation is discussed in separate subparagraphs. The multi-objective fitness function requires more algorithms to support the main requirements, and Algorithm 2 shows these functions; such as waste container capacity calculation, energy required to travel to the disposal and depot locations, total route distance calculation, flight time calculation, distance between each pair of locations, total route loads, energy consumption calculation, energy availability calculation, total number of depot location visits, and calculation of fitness components, among others. Algorithm 3 shows the main component algorithms for the multi-objective fitness function algorithm which is detailed in the next sub-paragraph.

$$\begin{aligned}
 \text{Fitness} = & \text{Cost paths penalty} \\
 & + \text{capacity penalty} \\
 & + \text{energy penalty} \\
 & + \text{Penalty for Uncollected Locations.}
 \end{aligned}
 \tag{3.1}$$

A. Cost Paths Penalty

The cost paths penalty represents the total travel distance with the edges of the routes between the WCLs, including the depot and the disposal location, shown in Equation 3.2. The cost paths penalty represents the total travel distance with the edges of the routes between the waste locations, including the depot and the disposal location.

$$\text{Cost paths penalty} = 1/\sum d_{ij}(\text{Euclidean distance}) \quad (3.2)$$

The above equation indicates that the penalty for cost paths is calculated by the penalty for the total distance of the tour, which is calculated using the Euclidean distance formula that represents the distance between two edges d_{ij} . Equation 3.3 calculates the best distance, which is the shortest distance.

Note that the best cost with the minimum value produced from each route is the best cost path. Euclidean distance (Han et al., 2019) is a popular method that is used to calculate the distance between two points. In our problem, it calculates the distance between two WCLs (genes).

$$\text{Euclidean distance} = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \quad (3.3)$$

The previous equation indicates the distance between two genes by finding the difference in X-coordinates and Y-coordinates, then to find the Euclidean distance results it takes the square root of the sum of both coordinates of the two genes (points).

B. Capacity Penalty

The capacity penalty is used when the drone's capacity exceeds the acceptable load. When capacity limitation is violated, the penalty value increases and becomes a severely unacceptable condition. Otherwise, a zero value is added when there is no violation, and therefore the drone payload is within

the drone's capacity; shown in Equation 3.4.

$$\text{capacity-penalty} = 1 / \sum_i (WC_i - C) \quad \text{where} \quad (WC_i - C) > 0 \quad (3.4)$$

$WC_i > WC_{i+1}$ The next waste node has more waste and is better than the nearby node, it will be selected.

$WC_i < WC_{i+1}$ The next waste node has less waste and is worse than the nearby node, it should be ignored and the next node selected.

It should be noted that the total load represents the total waste of the WC loads of waste collection from all locations carried by the drone, and the C capacity of the drone is the load it can carry. The capacity penalty contains three conditions. First, the condition that usage $(WC - C) > 0$ ensures that the denominator is positive and nonzero, which prevents undefined division. The second condition, $WC_i > WC_{i+1}$, indicates that if the waste at node i is greater than at node $i + 1$, the next waste node contains more waste, it is considered a more favorable candidate than a nearby node and is selected. The third condition states that if $WC_i < WC_{i+1}$, meaning the waste at node i is less than at node $i + 1$, the node with more waste nearby will be chosen and the node with fewer wastes will be ignored. The capacity penalty is designed to ensure that the drone does not exceed its payload capacity. When the total collected waste is below the maximum capacity, no constraint is violated. However, if the total waste is equal to or exceeds the capacity, the penalty is calculated as the excess amount above the capacity of the drone. This mechanism ensures that the capacity constraint is respected while prioritizing nodes with higher waste. Furthermore, if the total waste WC is equal to the capacity C , then the difference between them is zero $(WC - C) = 0$, occurs only when the tour starts and the waste has not yet been collected, or when the drone empties its waste at the disposal location. Therefore, the capacity penalty will not be activated in these cases.

Algorithm 1 Initialize Population Algorithm

```
1: Start
2: Initialize random population
3: for _ in range(population_size) #based on data processing
4: route = ["depot"] #set depot as starting route with indentation
5: while loop (unvisited_genes)
6:     feasible_genes = []
7:     for gene in unvisited_genes:
8:         distance = calculate_distance(current_location, gene)
9:         if distance  $\leq$  remaining_capacity and distance  $\leq$  remaining_energy = True
10:            feasible_genes.append((gene, distance))
11:            Itr = all_visited_genes
12:        if not feasible_genes:
13:            if remaining_capacity = drone_capacity:
14:                break
15:            else
16:                UAV go to disposal
17:        else
18:            sorted_genes = sorted(feasible_genes)
19:            selected_gene = sorted_genes[by less distance]
20:            remaining_capacity = int(selected_gene["WasteCapacity"])
21:            remaining_energy = perform_trip(current_location, selected_gene)
22:        else
23:            energy_available  $\leq$  energy_threshold
24:            remaining_energy = drone_energy
25:            UAV go to disposal
26:            UAV go to depot to recharge
27:            unvisited_genes = [gene for gene in waste_genes if gene not in route]
28:            for gene in unvisited_genes:
29:                route.insert(-1, gene)
30: route.append("disposal")
31: route.append("depot")
32: Print population
```

C. Energy Penalty

The energy penalty represents the energy consumption during UAV operations throughout the tour, which can be calculated by summing the total energy consumed between each location. Therefore, the drone needs to unload its cargo for disposal, then go to the depot to recharge, and then continue following Equation 3.5, which can be illustrated as follows:

$$\text{Energy_penalty} = 1/\sum e_{ij} \quad (3.5)$$

$$e_{ij} = r \times F_{ij} \quad (3.6)$$

To calculate the energy consumed between two locations e_{ij} , the energy consumed per unit of time r , which calculates the amount of energy consumed r by the UAV during a specific time, indicates the efficiency or productivity of the endurance energy for units of time that can be calculated in minutes or hours. The energy consumed per unit of time multiplied by the flight time required to cross any distance between two positions F_{ij} is shown in Equation 3.6. However, the energy consumption per unit time must be calculated by multiplying the nominal power consumption npc , which is the average power consumed by the drone during its flight, measured in watts (W). The efficiency factor ef is known as the efficiency system to convert energy into useful energy without waste, and is then divided by the duration of flight time fd , which is the flight ability of the drone; with the time taken during the flight given in seconds. All parameters in Equation 3.7 are defaulted and taken from drone and power specifications (manufacturer data). This function takes into account the requirements of the energy system along with the rate at which the energy system is converted to useful use.

$$r = (npc \times ef)/fd \quad (3.7)$$

Furthermore, the required flight time fd between two points must be calculated taking the distance between two genes (points) d_{ij} divided by the speed of the drone ds in terms of the ability to travel a distance of kilometers per hour (km/h) as shown in Equation 3.8:

$$fd = d_{ij}/ds \quad (3.8)$$

The energy required, as well as the energy consumed for the disposal route, is indicated as the energy needed to go to the disposal location and dump all the waste loads. The amount of waste accommodated represents the waste capacity to be dumped at the disposal site. The available energy is the remaining power in the drone in a certain location. Additionally, the energy required (Shivgan & Dong, 2020) for the depot refers to the power needed to go to the depot to charge. The energy consumption of the drone is calculated similarly for all locations.

D. Penalty for Uncollected Locations

The penalty for Uncollected Locations UN represents the Uncollected WCLs penalty that is necessary to collect from all locations that still have waste. It is calculated by the relation between the absolute count of locations with respect to the unique locations that can be found by the difference between the total WCLs that the drone visited and the remaining locations that need to be visited for waste collection. It is shown in the following Equation 3.9:

$$\text{UN Penalty} = 1/\sum(WCL - UN) \quad (3.9)$$

E. Calculate the Best Time

The best time can be calculated within several functions. The first is the flight time function, which calculates the edge time between two genes (two consecutive locations/sites) as described before; and another function is the

calculation of the route distance shown in Equation 3.10, which iterates over all genes on each edge and uses the total distance calculated for each edge to find the shortest time. The functions used to find the best route time can be listed as follows:

- Calculate distance (Euclidean distance).
- Flight time function.
- Calculate route distance.

$$\text{calculate route distance} = \sum \text{calculate_distance}(\text{gene1}, \text{gene2}) \quad (3.10)$$

Algorithm 2 Algorithm of Assisting Functions of Fitness Functions.

```

1: def get_energy_consumed_per_unit_time
2: def get_site_capacity(site)
3: def energy_disposal(solution, disposal_site,
4: drone_speed, calculate_route_distance)
5: def energy_depot(solution, disposal_site,
6: depot_site, drone_speed, calculate_route_distance)
7: def calculate_route_distance(route)
8: def calculate_flight_time(route, drone_speed)
9: def calculate_distance(gene1, gene2)
10: def calculate_route_load(route, get_site_capacity)
11: def calculate_energy_consumed(location1, location2,
12: drone_speed, get_energy_consumed_per_unit_time,
13: nominal_power_consumption, efficiency_factor, flight_duration)
14: def is_almost_empty_energy(energy_available, energy_threshold)
15: def calculate_num_depots_trips(drone_routes,
16: calculate_route_load, get_site_capacity, drone_capacity)
17: def calculate_fitness_components(solution, total_waste_locations,
18: drone_capacity, energy_available, energy_disposal, energy_depot,
19: drone_speed, nominal_power_consumption,
20: efficiency_factor, flight_duration, get_site_capacity,
21: calculate_route_distance, calculate_energy_consumed)

```

3.3.1.3 Genetic Operators

Genetic operators are essential components of the GA process, which aims to enhance the population by exploring a space of candidate solutions to find optimal or near-optimal solutions. GA operators work iteratively until better solutions are found by optimizing complex constraints solutions. These factors can be explained in the forthcoming subparagraph.

A. Selection

The selection process comprises the selection of the best intermediate population for each generation; henceforth, followed by the crossover and mutation processes to produce the next generation. There are several selection techniques that select the best chromosomes. In this proposed model, the tournament technique (Sarvari et al., 2020) is applied. The tournament method selects random populations in several iterations and the selected groups are the most suitable based on the constraints of the problem. The tournament approach is a very effective method. It is not complicated and does not require calculating properties or preservation of variety, which means that even lower fitness values can be selected, provided that the performance is well within the tournament. Furthermore, the tournament parameters can be easily modified as the tournament size. Figure 3.3 illustrates an example of a tournament in a WMS: First, a tournament of size 5 is chosen; which, in this example, contains 20 waste locations in one round that compete with each other. Second, the top five individuals of the tournament are ranked based on their performance in the fitness function. Third, the fitness function of each individual is evaluated so that the best fitness value is selected, which is passed on to the next generation. This process is repeated until the required number of individuals is selected for the next generation.

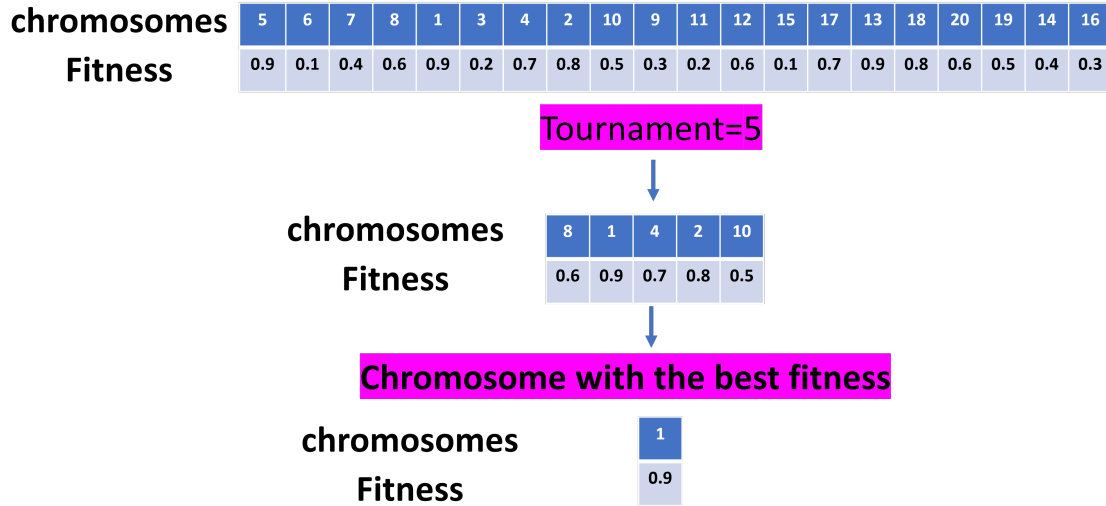


Figure 3.3: The Example of a Tournament in WMS.

B. Crossover

This process produces two offspring from two parents. The aim of this method is to produce the best offspring by mixing the properties of the two different parents. In this model, the order crossover (OC) is used based on extensive research to solve VRP problems (Wibisono et al., 2021). The OC system selects a random subset of locations in the first parent (P1) and the second parent (P2), and then swaps between P1 and P2 in the next subset. The unidentified subgroup locations in both P1 and P2 must then be re-added in the first offspring and the second offspring. WCLs to be filled with offspring should not be previously determined. If they are previously specified, the WCL will be replaced with the corresponding WCL that has not been selected or visited before. Additionally, each corresponding WCL is moved in an orderly manner, and if selected in the offspring, that location must be skipped and moved to the next until all WCLs have been visited and populated by the new population. To explain OC in an example, the figures below include 20 WCLs nodes; $(N) = 20$ and is rendered by a single drone. Figure 3.4 shows an example of a P1 including the depot (D) and disposal (DS) locations. In addition, Figure 3.5 shows a representation of

the P2 example.

Algorithm 3 Algorithm of Fitness Functions.

```
1: Start
2: Evaluate the population
3: Set parameters
4: def calculate_fitness_components(solution, total_waste_locations,
5:   drone_capacity, energy_available, energy_disposal,
6:   energy_depot, drone_speed,
7:   nominal_power_consumption, efficiency_factor,
8:   flight_duration, get_site_capacity, calculate_route_distance,
9:   calculate_energy_consumed)
10: For
11:   route_load = [get_site_capacity(drone_route)]
12:   Calculate total load for all distances
13:   Continue
14:   For
15:     if route_load  $\geq$  drone_capacity
16:       num_disposal_trips is incremented
17:       total load is decreased by route_load
18:       energy_available = energy_depot
19:       energy_penalty
20:     Elif
21:       energy_available  $\leq$  energy_threshold = true
22:       num_disposal_trips is incremented
23:       Go to disposal to empty
24:       Go to depot to recharge
25:   Energy penalty # Calculate the fitness based on constraints
26:   cost_paths_penalty
27:   penalty_uncollected_locations
28:   capacity_penalty
29:   Main fitness calculation
```



Figure 3.4: Example of P1.



Figure 3.5: The Representation of P2.

Crossover order: in the first step, a random location is selected for the subset of both P1 and P2 shown in Figure 3.6; and then the remaining WCLs are filled in as the rule for the crossover order. Figure 3.7 shows that the offspring reproduce E1 and E2 according to the crossover order.

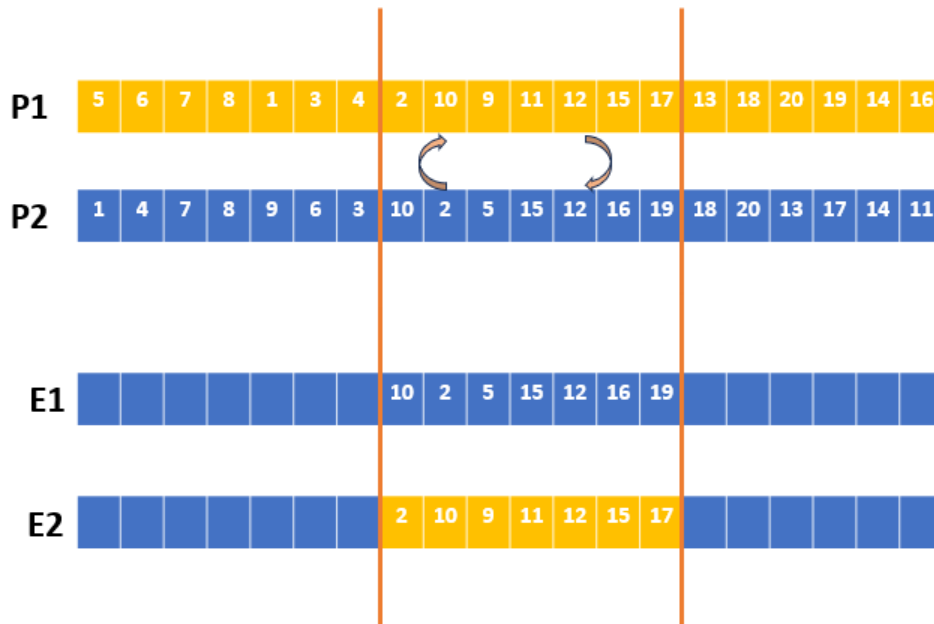


Figure 3.6: The Subset Randomly Located from both P1 and P2 to Reproduce Offspring.

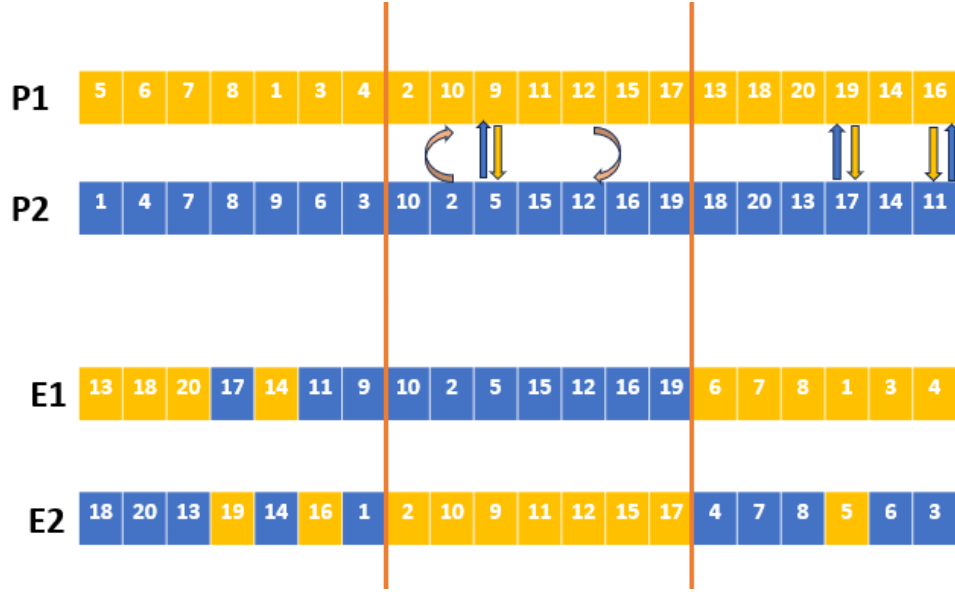


Figure 3.7: The Full Remaining WCLs as the Rule of the Crossover Order.

C. Mutation

Mutation is a process that comes up with the idea of a GA. That mutation rarely occurs in a new population. A mutation changes one or more genes to produce a new generation of new solutions, such as changing the order of WCLs. The literature has recommended that the mutation rate, which is the probability that the gene will change, preferably start as low as 1%, with the mutation range often averaging between 1 and 5%. This research combines the exchange mutation (Mohammed et al., 2012) and the local search space (Yuan et al., 2022). Many works in the literature regard exchange as a well-known approach to solving many routing problems, because it can provide a diversity of populations and avoid stagnation in the local optima (Sajid et al., 2021).

Moreover, first, the exchange mutation process involves selecting a WCL and swapping between the locations as shown in Figure 3.8. Then, after the population process finishes in exchange mutation, it incorporates local search space mutation, which is a heuristic optimization technique that can improve the quality of solutions. Thus, the local search space explores the

nearby node for a slight motion change and calculates incremental steps to find the best solution. The local search space (Erdelić et al., 2019) used for routing problem solutions employed the Swap-based Hill Climbing Local Search, which is a type of local search space that involves selecting and swapping two vertices. This allows for the evaluation of solutions and movement towards the best nearest neighbor. The hybrid mutation approach enhances the mutation process by transferring the resulting set after the swap mutation process to the Swap-based hill climbing local search, ensuring that both local and global solutions are explored and adding robustness. Figure 3.9 shows an example of the local search space mutation on a given path. In the following example, the path starts and ends at the depot (D) and visits the disposal before the end of the path. The path has 20 WCLs when the random value is less than the mutation rate. It triggers a local search space mutation where it picks any two random locations and swap the order between them. This example defines locations 6 and 8, so the edge changes from 5,6,7,8 to 5,8,7,6; thereby updating the population accordingly based on the best solutions provided. If a better neighbor location is available, it will be updated until no better solution is possible. For instance, if location number 7 is considered the best solution, it will change to 5, 7, 8, 6.

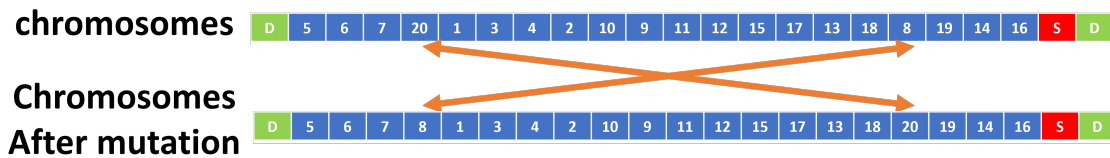


Figure 3.8: Example of Exchange Mutation.

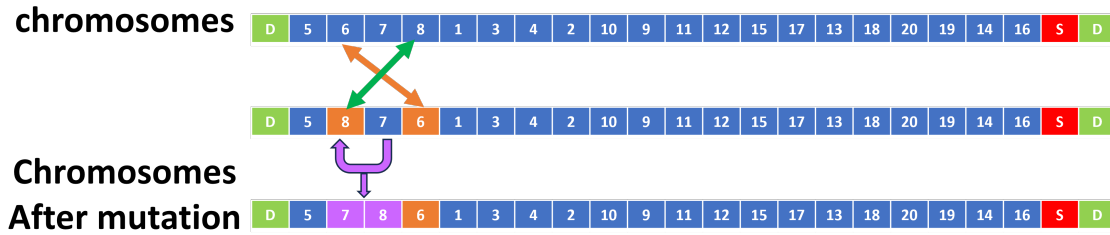


Figure 3.9: Example of Local Search Space (Swap-based Hill Climbing Local Search).

3.3.1.4 Termination of GA

The GA process continues until it obtains solutions that achieve the main objectives of finding the optimal path based on the shortest distance with respect to capacity and energy constraints. Furthermore, when it reaches the maximum iteration, the algorithm stops immediately. In addition, the convergence threshold value is used to improve fitness, which compares the best fitness of the current tour with the fitness of the previous tour. The algorithm then checks the fitness improvement in subsequent iterations; hence, when it is below the convergence value and considering that the algorithm has converged, it stops the iterations. At each iteration of the loop, the current fitness is compared to the previous one and is evaluated when it is better; hence, the best solution is updated. The algorithm continues to iterate until it reaches the maximum iteration number or falls below the convergence threshold value, hence terminating whichever conditions are met first. This method is applied after each iteration to ensure the best system performance. Figure 3.10 shows the complete system model. To illustrate the model, several techniques are used to overcome the local optimum, which are discussed in the following points:

- In the initialization phase, the population uses a random selection for the start of the algorithm, which can help the algorithm explore different regions of the solution space, considering one way to avoid local

optimization.

- The shifting of the population before the fitness evaluation provides population diversity, produces different combinations of locations (genes), and changes the route order, which reduces the local optimum.
- The use of a convergence threshold allows the space to be explored until certain parameters stop the convergence to a local optimum.
- The GA parameters are carefully selected based on literature review research and using trial-and-error techniques. A series of experiments are conducted to evaluate different combinations of parameters. Among these, the results reveal a set of parameters that consistently produced the best performance. It includes an efficiency factor between 0.1 and 1.0, a tournament size ranging from 2 to 7, a mutation rate ranging from 0.01 to 0.7, and a crossover probability ranging from 0.2 to 1.0. These best parameters are identified through comparative analysis and are utilized in the final experiments, as demonstrated in Table 3.2. Additionally, the model can be adjusted to any type of parameter based on the type of drone. The parameters selected in our approach are: the drone power, which is set to 200KW; the power consumption is 30W/s; the nominal power consumption is 100W, which specifies the amount of energy used during any given operation; the efficiency factor is 0.9, which defines the efficiency of converting energy into useful energy; the maximum flight time is 2.5 hours; and the drone speed is 60 kilometers per hour. Moreover, the parameters for the GA operators are: tournament size, which is 5; the mutation rate is 0.7; the crossover probability is 0.2; and the generation number is 50.

Table 3.2: The Mathematical Model Notation

Exp. No.	Efficiency Factor (0.1 – 1.0)	Tournament Size (2 – 7)	Mutation Rate (0.01 – 0.7)	Crossover Probability (0.2 – 1.0)	Best Fitness Avg. for 100 WCL	Best Fitness Avg. for 500 WCL	Best Fitness Avg. for 1000 WCL
1	0.1	2	0.05	0.2	5.723	11.352	15.321
2	0.2	3	0.01	0.3	6.432	11.432	15.467
3	0.3	4	0.10	0.4	6.321	12.754	15.874
4	0.4	5	0.20	0.5	6.645	12.543	15.765
5	0.5	6	0.30	0.6	6.134	12.467	15.543
6	0.6	7	0.40	0.7	6.211	12.123	15.223
7	0.7	6	0.50	0.8	7.432	12.234	15.643
8	0.8	6	0.60	0.9	8.213	13.765	16.432
9	0.9	5	0.70	0.2	5.154	10.967	14.934
10	1.0	5	0.70	1.0	6.110	11.123	15.765

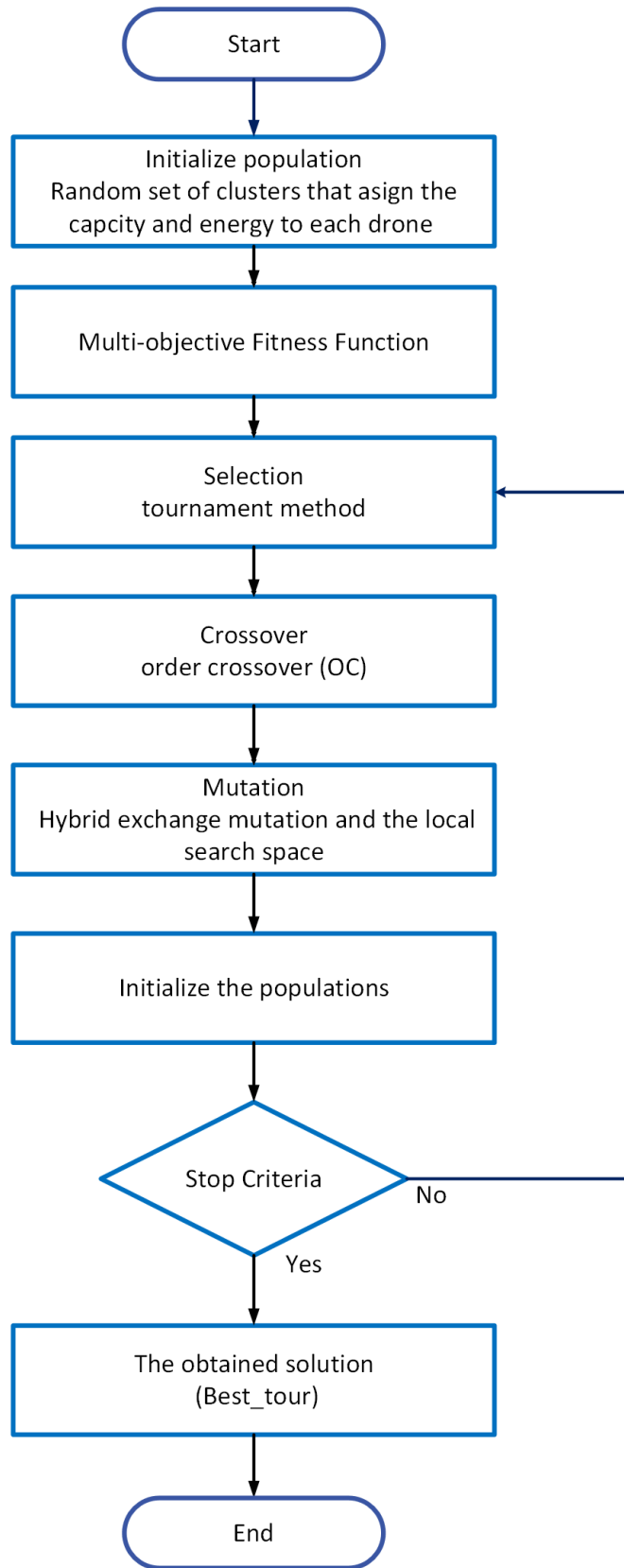


Figure 3.10: GA System Model.

3.4 Experimental Results and Analysis

The proposed solution is implemented using Python 3 and Jupyter; embedded with other libraries such as numpy, pandas, ga, random, math, copy, seaborn, pyplot, base, deep, matplotlib, cluster, partial, plugins, folium, and ortools. All of these modules provide several functionalities for efficient optimization, data manipulation, plotting, and support of advanced scientific computing. This system is generated on an Intel(R) Core (TM) i7-2670QM CPU laptop with 2.20 GHz RAM under Microsoft Windows 10 Pro. Table 3.3 shows the parameters of the experiments. The values of these parameters can be adapted based on the types of drones that can be utilized in the actual system that is eligible for this research. The experiments utilize synthetic data, which is employed to apply a theoretical approach to a problem when there are insufficient real-world data. This type of data is designed to mimic actual data in the real world, which consists of real WCLs in the Kingdom of Bahrain, comprising the latitude and longitude of each WCL along with the corresponding capacity of the waste container. Figure 3.11 shows a partial example of the generated data. The specified locations' data are not based on cities, but on the scale of specific WCLs. The process of synthetic data generation is shown as follows:

- The graph nodes are randomly selected within the borders of the Kingdom of Bahrain in land areas, hence excluding marine areas.
- The maximum waste capacity of each WCL is set at no more than 100 kg and is randomly generated between 10 and 100 kg.
- The type of waste, the type of drone or the time limit is not included in the synthetic data generation for flexibility to allow moderation in the model based on real-world WCS.
- The tour is generated based on several routes, and any path can be visited more than once if it can reduce the distance.

- The waste capacity and the energy of the drone are based on the manufacturer's specifications. In this experiment, they are taken from the most popular type of drone used in WCSs and can be modified based on the drone used in the real world.

The accuracy and size of the data are crucial for producing reliable and predictable results. Since real management data systems may not exist, synthetic data becomes an ideal substitute. Additionally, using such data supports data diversity, which is essential when real-world data cannot be apprehended across all scenarios. The effectiveness of using synthetic data over real-world data in terms of time and cost is also notable. The data is utilized to evaluate the proposed GA, control the experiments, and analyze the results. The synthetic data set will not include weather conditions or terrain differences, and the waste capacity tends to increase with higher population densities. Although a model developed using a specific dataset from a particular country can be generalized to other urban areas, key parameters such as geographic distributions and the model itself can be adapted to different urban environments. The number of datasets used should closely reflect real-world scenarios. Future work may include data from multiple cities.

	A	B	C
1	Latitude	Longitude	WasteCapacity
2	25.9599	50.69633	22
3	26.11658	50.81261	79
4	26.07737	50.5184	92
5	26.35069	50.61944	71
6	25.96938	50.79728	84
7	26.18332	50.56312	20

Figure 3.11: A Partial Example of Generated Data.

The comparison is based on the specified data of population size and

Table 3.3: Experiments Parameters.

Parameter	Value
population size	100,500,1000
drone speed	60 KM/h
energy consumed per second	30 W/s
the nominal power consumption	100 W
efficiency factor	0.9
drone capacity	200 k
drone energy	500 W
tournament	size=5
converge rate	0.01
Number of iterations	1000
Mutation rate	0.7
crossover probability	0.2
Generation numbers	50

WCL numbers as 100, 500, and 1000 WCLs respectively. The parameters are fixed; as the maximum capacity is 500 for each WCL weighted in kilograms, the speed of the drone is set to 60 km/h, the flight duration during the flight which is measured in minutes, the power consumed per second is 30 W/s, the nominal power consumption is set to 100 W, then the efficiency factor is set to 0.9 indicating the efficiency of the used system. All these parameters are taken from the manufacturing data or the factory specifications. In addition, the capacity of the drone is 200K, and the power of the drone is 500W. The proposed model does not select the type of drones to generalize this model to real-world systems based on the type used by the system and the purpose of future specifications of UAV models. The features were selected based on a range of the most commonly used types of drones for waste collection systems or based on well-known kinds of drones, such as heavy hybrid VTOL drones and heavy multi-rotor drones. Table 3.4 displays the best fitness values and the best execution time

Table 3.4: Experimental Results.

population waste locations	100	500	1000
Best fitness	5.154	10.967	14.934
Best time	1 hours	2 hours	3 hours
Execution time	4.382 seconds	24.913 seconds	86.123 seconds
Standard deviation	0.752	1.026	1.385

of the model obtained in different numbers of WCL. The model examines the multi-objective improvement in fitness functions by finding the shortest distance and collecting the maximum waste amount while consuming less energy for the entire tour. Figure. 3.12, Figure. 3.13, Figure. 3.13, and Figure. 3.15 show examples of how the model works within just five waste locations, which can include both conditions or one of which can occur in these two cases:

- When power is almost depleted, the drone should empty the waste load at the disposal location, then go to the depot and recharge.
- When the capacity load of the drone is full, it needs to unload its waste at the disposal location.

In the first case, when the numbers of WCLs are only 100 locations assigned to a single drone, the best fitness recorded in the route number of 50 routes is 5.154 with an execution time of 4.382 seconds. In the second case, when the number reached 500 WCLs; it recorded a fitness of 10.967 with an execution time of 24.913 seconds. Finally, in the last case with 1000 WCLs, it recorded a fitness of 14.934 with an execution time of 86.123 seconds. It is also observed that there is a positive outcome displayed in an inverse relationship between the number of WCLs and the best fitness and the execution time; as the number of WCLs increases, the best fitness recorded in the route number and the execution time decrease relatively.

The tour includes both waste unloading and recharging the UAV. First, the UAV waste load capacity is almost full and needs to be unloaded. It goes from WCL B to the disposal location (S) to unload the waste cargo and then it continues its tour to collect the waste from the remaining WCLs. When UAV charge is almost depleted when it reaches WCL D during the tour; then recharging is needed. From WCL D it goes to the disposal location (S) first to unload the waste cargo and then goes to the Depot (D) for recharging. Then it continues its tour to collect the waste from the remaining WCLs and after completing serving all the locations, it needs to dump the waste again in the disposal location (S) then return to the depot (D) on edge number 10.

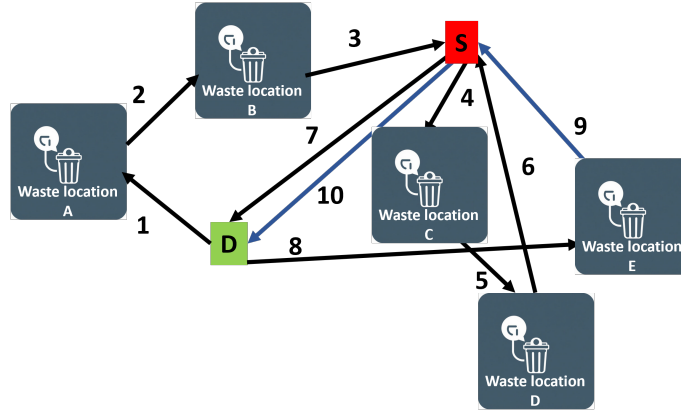


Figure 3.12: Example of a Typical Waste Disposal and Recharging Case.

The UAV waste load capacity is full when it reaches WCL B during the tour; then waste unloading is needed. From WCL B it goes to the disposal location (S) to unload the waste cargo. Then it continues its tour to collect the waste from the remaining WCLs and after completing serving all the locations, it needs to dump the waste again in the disposal location (S) then return to the depot (D) on edge number 8.

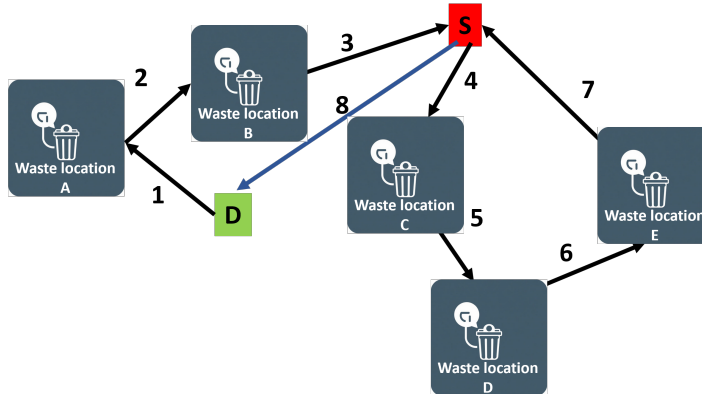


Figure 3.13: Example of a Typical Waste Disposal Case.

When UAV charge is almost depleted as it reaches WCL C during the tour; then recharging is needed. From WCL C it goes to the disposal location (S) first to unload the waste cargo and then goes to the Depot (D) for recharging. Then it continues its tour to collect the waste from the remaining WCLs and after completing serving all the locations, it needs to dump the waste again in the disposal location (S) then return to the depot (D) on edge number 9.

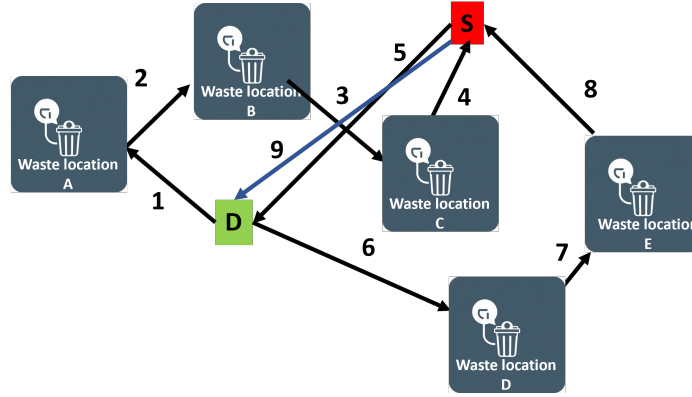


Figure 3.14: Example of a Typical Recharging Case.

The tour includes both recharging and emptying the waste. First, When UAV charge is almost depleted when it reaches WCL B during the tour; then recharging is needed. From WCL B it goes to the disposal location (S) first to unload the waste cargo and then goes to the Depot (D) for recharging. Then it continues its tour to collect the waste from the remaining WCLs. Secondly, the UAV waste load capacity is almost full when it reaches WCL D and needs to be unloaded. It goes from WCL D to the disposal location (S) to unload the waste cargo and then it continues its tour to collect the waste from the remaining WCLs. After completing serving all the locations, it needs to dump the waste again in the disposal location (S) then return to the depot (D) on edge number 10.

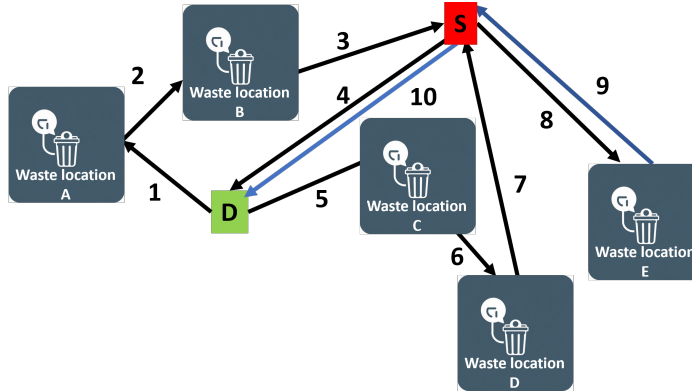


Figure 3.15: Example of a Typical Recharging and Waste Disposal Case.

This indicates that as the amount of data expands, the problem becomes more complex and the GA performs even better as the complexity increases. Therefore, the best fitness is registered when the number of WCLs is 1000, taking into account all constraints. Figure 3.16, Figure 3.17, and Figure 3.18

show the comparison between the number of WCL versus the execution time with the satisfaction of constraints. In other words, the experiment includes multiple dump locations and demonstrates the relationship between the number of routes and multi-fitness constraints. The histogram demonstrates the positive visual impact of the number of routes and the overall fitness of the proposed model. When the number of WCLs and routes increases, the results show further improvement. Figure 3.19 shows a comparison of the best fitness for each different number of WCLs to evaluate the performance of the model. In general, it is observed that there is a positive outcome indicated by the relative inverse relationship between the number of routes and the fitness function; hence, when the number of routes increases, the value of the fitness function decreases relatively, which means that when the problem becomes more complex, the quality of the fitness increases. Henceforth, Figure 3.20 shows that the relative execution time decreases significantly as the number of WCLs served increases. Thus, complexity and computational time increase as the number of routes increases, while improving the system's performance with a larger search space to obtain the best solutions.

To analyze the results, standard deviation techniques are used to determine the manner of variation of the data and its outline from the average dataset. When the standard deviation decreases, it means that it is close to the mean of the dataset and indicates the consistency of the data. On the other hand, when the standard deviation is larger, it means more variance, and the data have a greater mean spread than a central tendency. The amount of variance affects the prediction and accuracy of the results. The standard deviation calculation process in the model takes place in steps as follows:

- Calculate average fitness values for each population.
- Calculate the standard deviation of the distance for each gene by subtracting the average fitness value.

- Sum the squared deviation for each distance for the entire population.
- Divide the total sum squared deviation by the total genes minus one.
- This process will be repeated for each population group.

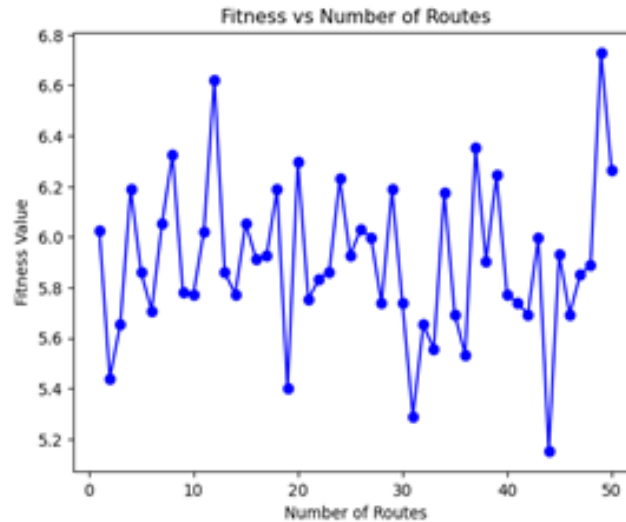


Figure 3.16: WCS Diagrams for 100 WCLs Show the Relationships Between Fitness Constraints and the Number of Routes.

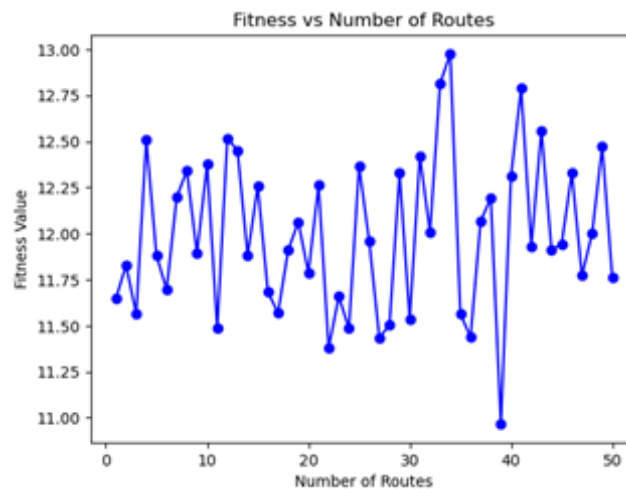


Figure 3.17: WCS Diagrams for 500 WCLs Show the Relationships Between Fitness Constraints and the Number of Routes.

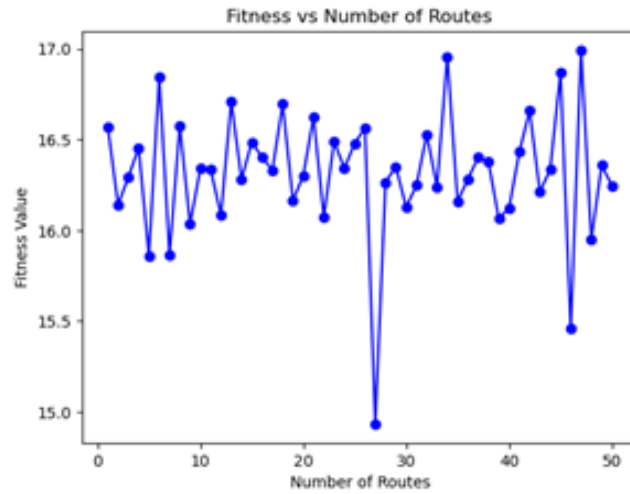


Figure 3.18: WCS Diagrams for 1000 WCLs Show the Relationships Between Fitness Constraints and the Number of Routes.

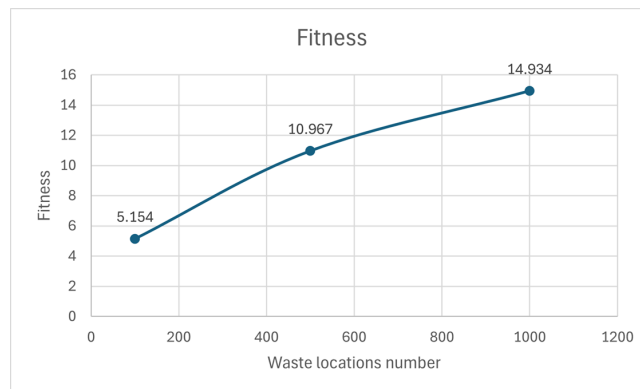


Figure 3.19: Comparison of the Best Fitness Values Between Different WCLs.



Figure 3.20: Comparison of the Execution Time Between Different WCLs.

The formula for calculating the standard deviation is as follows:

$$\sigma = \sqrt{\frac{\sum(n - \mu)^2}{N}} \quad (3.11)$$

Where σ represents the standard deviation population, n is the data value, μ is the average fitness value, and N is the total number of all values. Figure 3.21, Figure 3.22 and Figure 3.23 shows the standard divisions for each of the three sets of WCLs. The comparison of the three sets shows that the higher the WCL, the higher the standard deviations, which shows more variability in the data based on the high complexity, which is recorded in 100 locations as 0.752, in 500 locations as 1.026, and in 1000 locations as 1.385, which is shown in Table 3.3.

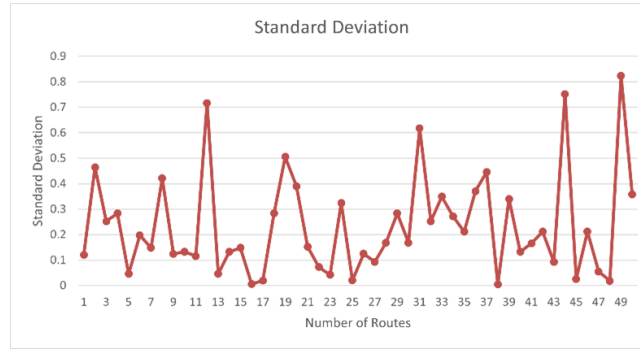


Figure 3.21: The Standard Deviations Diagram for a WMS with 100 WCLs.

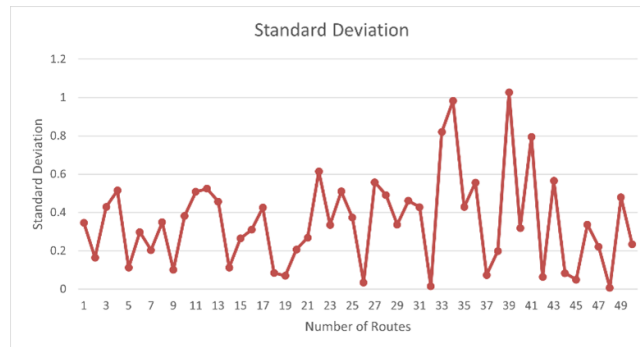


Figure 3.22: The Standard Deviations Diagram for a WMS with 500 WCLs.

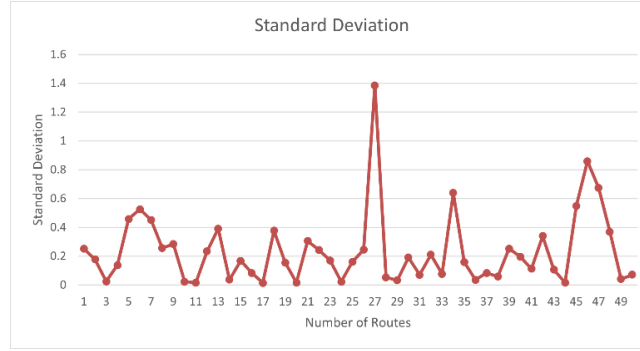


Figure 3.23: The Standard Deviations Diagram for a WMS with 1000 WCLs.

The results of the experiment are compared with two other studies that address similar constraints and use other approaches. The first comparative research used a hybrid ACO (Abdulsattar et al., 2023), and the second research used a two-phase approach, namely Maximum Waste in a MWMTT (Kaabi et al., 2023). In this comparison, GA scores are found to be better than both comparative studies that used three similar datasets. Each of the studies compared and calculated the percentage of reduction in fitness error as an indicator of the equations for the three datasets with the datasets of the experiments based on the number of WCL. However, this proposed model does not use the same dataset that was used in both comparative research; it only has the same problem to solve and addresses similar constraints. To compare the two studies with our study, the best fitness of each comparative study must first be obtained. The following formula is used to calculate the percentage of reduction in fitness error and the best fitness in the comparative studies, which are then compared with the best fitness from the experiment of the proposed model. Then the best fitness of the comparative study is subtracted from the best fitness of the proposed model; then the value is divided by the comparative study's best fitness; and the result is multiplied by 100 to convert it to a percentage. This calculation reflects the percentage of reduction in fitness error when comparing this study with each of the comparative studies. Following this calculation, a quantification

of the improvement of different algorithms that address similar problems is presented. The formulas that were adapted for the three datasets used in the comparative research differ from those used in this model. The following formulas correspond to each dataset number used in the proposed model:

1. The first Dataset:

- First Dataset representing the compared experiments datasets with 100 WCLs. The best fitness in the First comparative research =25.9

The best fitness in the second comparative research =36.1

The best fitness in this model experiment =5.154

Percentage of Reduction in fitness error = (best fitness in Percentage of Reduction in fitness error = (best fitness in comparative research -best fitness in this experiment)/ best fitness in comparative research $\times 100$

Percentage Reduction in fitness error from the First comparative research = $(25.9-5.154)/ 25.9 \times 100=80.8\%$

Percentage Reduction in fitness error from the second comparative research = $(36.1-5.154)/ 36.1 \times 100=85.72\%$

- First Dataset representing the compared experiments dataset with 500 WCLs The best fitness in the First comparative research =25.9

The best fitness in the second comparative research =36.1

The best fitness in this model experiment = 10.967

Percentage of Reduction in fitness error = (best fitness in comparative research -best fitness of this experiment)/ best fitness in comparative research *times*100

Percentage of Reduction in fitness error from the First compara-

tive research = $(25.9-10.967)/ 25.9 \times 100=57.61\%$

Percentage of Reduction in fitness error from the second com-

parative research = $(36.1-10.967)/ 36.1 \times 100=69.62\%$

- First Dataset representing the compared experiments datasets with 1000 WCLs. The best fitness in the First comparative research =25.9

The best fitness in the second comparative research =36.1

The best fitness in this model experiment =14.934

Percentage of Reduction in fitness error = (best fitness in comparative research -best fitness experiment)/ best fitness in comparative research $\times 100$

Percentage of Reduction in fitness error = (best fitness in comparative research -best fitness experiment)/ best fitness in comparative research $\times 100$

Percentage of Reduction in fitness error from the First comparative research = $(25.9-14.934)/ 25.9 \times 100=42.8\%$

Percentage Reduction in fitness error from the second comparative research = $(36.1-14.934)/ 36.1 \times 100=58.63\%$

2. The Second Dataset:

- Second Dataset representing the compared experiments datasets with 100 WCLs. The best fitness in the First comparative research =27.4

The best fitness in the second comparative research =29

The best fitness in this model experiment =5.154

Percentage of Reduction in fitness error = (best fitness in comparative research -best fitness experiment)/ best fitness in comparative research $\times 100$

Percentage of Reduction in fitness error from the First comparative research = $(27.4-5.154)/ 27.4 \times 100=81.2\%$

Percentage of Reduction in fitness error from the second comparative research = $(29-5.154)/ 29 \times 100=82.23\%$

- Second Dataset representing the compared experiments datasets with 500 WCLs. The best fitness in the First comparative research =27.4

The best fitness in the second comparative research =29

The best fitness in this model experiment =10.967

Percentage of Reduction in fitness error = (best fitness in comparative research -best fitness experiment)/ best fitness in comparative research $\times 100$

Percentage of Reduction in fitness error from the First comparative research = $(27.4-10.967)/ 27.4 \times 100=59.9\%$

Percentage of Reduction in fitness error from the second comparative research = $(29-10.967)/ 29 \times 100=62.18\%$

- Second Dataset representing the compared experiments datasets with 1000 WCLs. The best fitness in the First comparative research =27.4

The best fitness in the second comparative research =29

The best fitness in this model experiment =14.934

Percentage of Reduction in fitness error = (best fitness in comparative research -best fitness experiment)/ best fitness in comparative research $\times 100$

Percentage of Reduction in fitness error from the First comparative research = $(27.4-14.934)/ 27.4 \times 100=45.7\%$

Percentage of Reduction in fitness error from the second com-

$$\text{parative research} = (29-14.934)/29 \times 100 = 48.5\%$$

3. The third Dataset:

- Third Dataset representing the compared experiments datasets with 100 WCLs. The best fitness in the First comparative research = 27.2

The best fitness in the second comparative research = 21.2

The best fitness in this model experiment = 5.154

Percentage of Reduction in fitness error = (best fitness in comparative research - best fitness experiment) / best fitness in comparative research $\times 100$

Percentage of Reduction in fitness error from the First comparative research = $(27.2-5.154)/27.2 \times 100 = 81.1\%$

Percentage of Reduction in fitness error from the second comparative research = $(21.2-5.154)/21.2 \times 100 = 75.69\%$

- Third Dataset representing the compared experiments datasets with 500 WCLs. The best fitness in the First comparative research = 27.2

The best fitness in the second comparative research = 21.2

The best fitness in this model experiment = 10.967

Percentage of Reduction in fitness error = (best fitness in comparative research - best fitness experiment) / best fitness in comparative research $\times 100$

Percentage of Reduction in fitness error from the First comparative research = $(27.2-10.967)/27.2 \times 100 = 59.6\%$

Percentage of Reduction in fitness error from the second comparative research = $(21.2-10.967)/21.2 \times 100 = 48.27\%$

- Third Dataset representing the compared experiments datasets with 1000 WCLs. The best fitness in the First comparative research =27.2

The best fitness in the second comparative research =21.2

The best fitness in this model experiment =14.934

Percentage of Reduction in fitness error = (best fitness in comparative research -best fitness experiment)/ best fitness in comparative research $\times 100$

Percentage of Reduction in fitness error from the First comparative research = $(27.2-14.934)/ 27.2 \times 100=45.2\%$

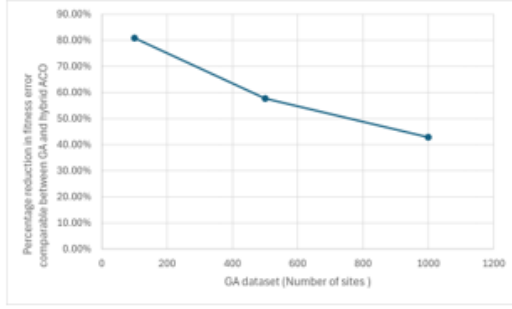
Percentage of Reduction in fitness error from the second comparative research = $(21.2-14.934)/ 21.2 \times 100=29.56\%$

As demonstrated in previous results, GA proves to be a better approach to solving path-planning problems than the hybrid ACO and MWMTT method. Moreover, when the number of WCLs that need to be served increases, the percentage of reduction in fitness error decreases, and vice versa. Overall, GA has much higher performance compared to approaches in both studies which is shown in Table 3.5 and Table 3.6; a comparison between hybrid ACO and MWMTT against the model proposed in this study. However, these results could also have been influenced by the nature of the dataset and the number of instances used in the comparative research datasets. For example, these datasets contain several locations that range from 21 to 981, which is less than 1000 locations in the proposed model, which can affect the reliability of the comparison of the results.

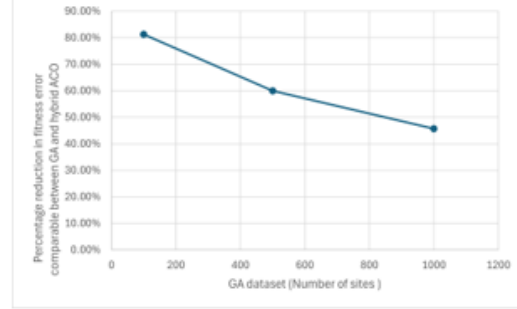
Figure 3.24 shows the percentages of reduction in fitness error that are compared between GA and hybrid ACO for each different dataset used by comparative research. Shows the decrease in the percentage of error reduction in several WCLs compared to the first, second, and third sets of the

dataset used in the hybrid ACO. Figure 3.25 shows the percentage of reduction in fitness error that is compared between GA and MWMTT for each different dataset used by the comparative research. Shows the decrease in the percentage of error reduction in several WCLs compared to the first, second, and third sets of the dataset used in the MWMTT. Figure 3.26 shows the percentage of reduction in fitness error for GA compared to comparable research in the three different datasets used in hybrid ACO and MWMTT, which demonstrates a negative relationship that when the WCL increases, the percentage of fitness error decreases due to the increase in model complexity and consequently enhances the performance of GA.

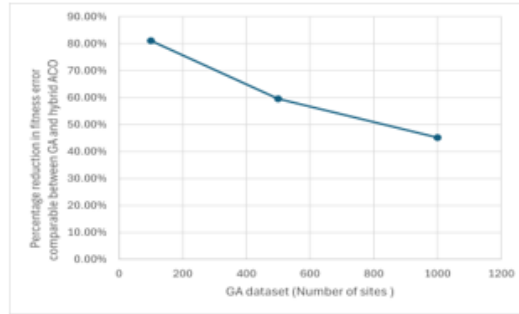
The comparative studies are presented above in the literature review section. Both studies are selected for comparison with the method proposed by this study because they all address similar constraints, such as finding the shortest distance and the best tours, the waste loading capacity of the drone and the constraints of energy consumption. However, the datasets used in the compared studies are slightly different from this proposed method; for example, the maximum number of WCLs is slightly less than those in our approach, hence, giving this proposed GA method a better ratio and increased reliability of results, even if it is more complex due to the increased number of locations. Hence, although the difference in WCL numbers in the three different studies can be seen from one angle as a comparison limitation, it can be seen from another angle as an increased reliability of the results and a greater strength of our study model due to the increased complexity. Furthermore, all three experiments have similar logistics, where the drones start and end with one depot; UAVs go to the depot when they need to recharge or to go for load disposal when waste capacity is almost full and then finish any unvisited locations. However, the proposed GA method works slightly differently in case the power is almost depleted. It has to go to the disposal point to unload and then to the depot to recharge.



Percentage reduction in fitness error of GA compare with the First comparative research for the first dataset.



Percentage reduction in fitness error of GA compare with the First comparative research for the second dataset.



Percentage reduction in fitness error of GA compare with the First comparative research for the third dataset.

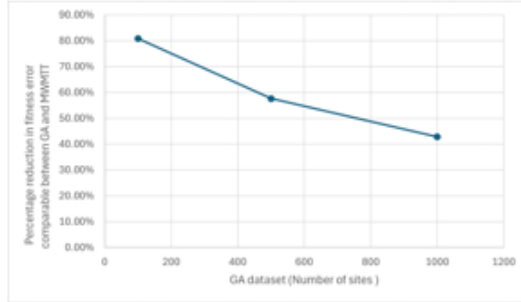
Figure 3.24: Percentage Reduction in Fitness Error of GA Compared with the First Comparative Research of Hybrid ACO.

The comparative search of GA creates a cluster path plan for the drone that does not allow it to recharge until it finishes its cluster trip. Therefore, our proposed method can lead to better results because it covers more WCLs in each trip and does not stop early until all the power of the drone is consumed, unlike the case in the compared studies. Also, when the proposed GA method is compared to the tight lower bound method used in previous comparative studies, the discrepancy between the percentages of reduction in fitness error of those studies compared to the results of the proposed method is very high consistently in all three datasets, with the GA results having very low fitness error and high percentages of reduction in fitness error. In addition, our approach includes three experiments that cover the number of locations (100, 500, and 1000), one depot, and one waste

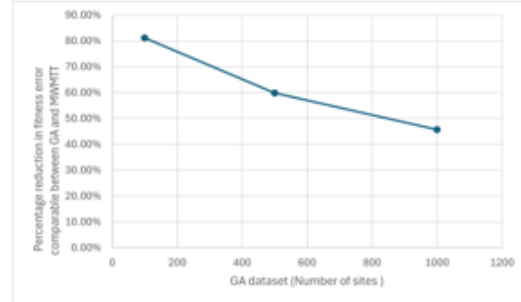
disposal location. whereas comparative research is conducted with small cases between 10-40, medium cases 41-48, and large cases between 49-89. Therefore, our model is superior in terms of complexity when compared to the number of cases in other researches. Moreover, our model represents the solution paths in which each chromosome is considered as a complete solution tour consisting of several routes to complete this tour. Whereas, in the other studies compared, the difference in the solution path lies in phase 2, where a linear program is used to group trips into clusters, allowing the use of an optimal drone flight capacity. Furthermore, a linear program is developed that complements a tight lower bound to validate their approach. The need for the two programs was because the linear program could not generate solutions that solve instances with more than 40 locations; hence, the lower bound is used to evaluate the performance of the approach for medium and large-sized instances. Therefore, the results obtained were in line with those obtained for small-sized instances, which is an obvious limitation that was overcome in our study approach.

Moreover, based on the literature review, GA has been shown in various studies to be better than many methods when comparing algorithms to find the most suitable or best algorithms to solve route problems. It has been found that GA is better in several aspects than ACO and PSO in solving complex and multi-objective problems. GA has been used by most optimization algorithms efficiently in routing problems (Juneja & Nagar, 2016) (Sonmez et al., 2015) (Zhao et al., 2018). In addition, many studies have used GA as the backbone of their model based on its high performance and superiority over ACO, which may cause a large computational time in the real world (Sarvari et al., 2020). Furthermore, GA has been shown to better handle many constraints, such as finding the shortest distance, efficient computational time for real-world systems, reducing energy consumption, and capacity constraints (Alwabli & Kostanic, 2020) (Bhargava et al., 2019)

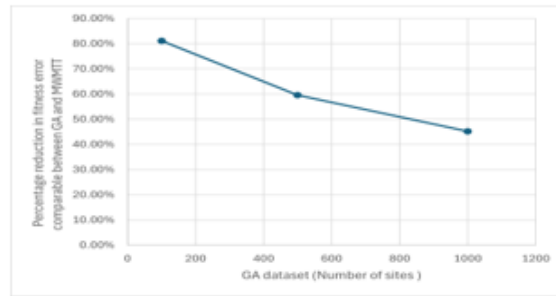
(Dolinina et al., 2019). Hence, the results obtained by using our approach are in line with many of the previous studies that used GA as a solution approach in their methodologies.



Percentage reduction in fitness error of GA compare with the second comparative research for the first dataset.



Percentage reduction in fitness error of GA compare with the second comparative research for the second dataset.



Percentage reduction in fitness error of GA compare with the second comparative research for the third dataset.

Figure 3.25: Percentage Reduction in Fitness Error of GA Compared with the Second Comparative Research of MWMTT.

In general, for all the results discussed above, the proposed method was found to have high reliability, internal validity, and external validity. The high reliability comes as mentioned previously from the higher number of subjects tested; the internal validity comes from the positive results and the outperformance of the model in the compared studies and fulfilling the objectives of this study; and the external validity comes from its consistency with previous research results with the possibility of generalization and applicability of the results in the real world. It can be applied to any type of in-

frastructure and type of UAV with any number of locations as needed. This approach is based on theoretical and experimental computational methods to examine the efficiency of an algorithmic modification that is capable of using real-world datasets and integrating them with the system. The adopted approach has been tested in various locations, 100, 500, and 1000, and in all the cases the results were encouraging, which gives a good indication that this approach works well in a large number of locations. The interesting point is that, whenever the number of locations increases, the number of tours is reduced in ratio. The significant role of the electric vehicle in industry and environmental protection can be confirmed by research studies and from our observations, it can be concluded that the main challenges and constraints that need to be solved and the different approaches used to improve its system can be identified. Most of the challenges that an electric vehicle can face are constraints that need to be considered, such as minimizing the cost and trade-offs between all constraints, such as energy, capacity, shortest path, and travel time. However, addressing these constraints should minimize computation time. The logistics of the model can be influenced to find the best solutions. The mathematical solution navigation through GA can have an optimal solution for a problem that is defined as NP-hard. Variation in results and even the computation time is expected in the real world and can be acceptable and supported in a dynamic environment. Our approach can work very well in a normal environmental climate, where there is no wind and with steady communication. However, if such perturbations exist, our approach might not work well as we do not consider such kind of disruption in our simulated model. Such disruptions may hinder the speed, takeoff, and landing of the drones. In addition, drone sensors might not work properly, in addition to problems that might be caused by communication loss difficulty.



Figure 3.26: Percentage Reduction in Fitness Error of GA Compared with Both Comparative Researches in the Three Different Datasets.

3.5 Summary

WMS is an important economic sector that helps to support health and improve the quality of life in smart cities. This study proposes a multi-objective GA aiming at improving the WCS process through identifying the best appropriate tours that can minimize the cost. The proposed model aims to satisfy a multi-objective approach that addresses constraints with a single UAV. It aims at improving performance by reducing flight distance, time, and power consumption while maximizing delivered waste load capacity. incorporating several modifications to the GA, such as adding distance minimization, energy conservation, and capacity maximization mechanisms in the random initial generation. In the process of customizing the fitness function and applying a hybrid space search during mutation, two hybrid types will be utilized to improve mutation by exploring a broader solution space more efficiently. The GA is well designed to find feasible solutions while considering three quantities of waste in WCLs produced from synthetic data. The experimental results show that the proposed approach outperforms other previous research approaches addressing the same constraints and with similar datasets. It should be noted that the proposed ap-

Table 3.5: The Comparison Between the First Research (Hybrid ACO) and Thesis Approach Within GA.

GA dataset (Number of locations)	Percentage reduction in fitness of DS1 in (hybrid ACO)	Best ness GA	fit- of reduction in error com-parable between GA and hybrid ACO	Percentage Indication of fitness prove-ment	Best DS2 (hybrid ACO)	fit- of ness in GA	fit- of reduction in error com-parable between GA and hybrid ACO	Percentage Indication of fitness prove-ment	Best DS3 (hybrid ACO)	fit- of ness in GA	fit- of reduction in error com-parable between GA and hybrid ACO	Percentage Indication of fitness prove-ment
100	25.9	5.154	80.8%	↑	27.4	5.154	81.2%	↑	27.2	5.154	81.1%	↑
500	25.9	10.967	57.61%	↑	27.4	10.967	59.9%	↑	27.2	10.967	59.6%	↑
1000	25.9	14.934	42.8%	↑	27.4	14.934	45.7%	↑	27.2	14.934	45.2%	↑

Table 3.6: The Comparison Between the Second Research MWMTT and Thesis Approach Within GA.

GA dataset (Number of locations)	Percentage reduction in fitness of error of DS1 in MWMTT	Best ness GA	fit- of reduction in error com-parable between GA and MWMTT	Percentage Indication of improvement	im- prove-ment	Best ness DS2 MWMTT	fit- of ness in GA	fit- of reduction in error com-parable between GA and MWMTT	Percentage Indication of improvement	im- prove-ment	Best ness DS3 MWMTT	fit- of ness in GA	fit- of reduction in error com-parable between GA and MWMTT	Percentage Indication of improvement
100	36.1	5.154	85.72%	↑	29	5.154	82.23%	↑	21.2	5.154	75.69%	↑		
500	36.1	10.967	69.62%	↑	29	10.967	62.18%	↑	21.2	10.967	48.27%	↑		
1000	36.1	14.934	58.63%	↑	29	14.934	48.5%	↑	21.2	14.934	29.56%	↑		

proach achieves an average reduction in errors in 100 instances compared to previous research by more than 81%. For 500 instances, the reduction in error reached more than 59%, while in 1000 instances, the error was reduced by more than 44%. Those promising results proved that GA utilization can improve the WCS process by identifying the best appropriate tours that can minimize the cost through distance minimization, energy conservation, and capacity tolerance maximization mechanisms.

Chapter 4

A Hybrid Approach of Genetic Algorithms and Reinforcement Learning for Waste Collection Using Unmanned Aerial Vehicle

4.1 Introduction

This chapter discusses the second proposed model that employs a hybrid approach which integrates two algorithm methods to provide a further enhanced performance than a single algorithm can achieve (Hejres et al., 2024). Consider a WCS with an efficient approach targeting multiple objectives using a single drone visiting several WCLs to collect and dispose of waste at the main central landfill, which is the same problem as the previous model in Chapter 3. While the difference here is the use of a very powerful combination that can solve complex problems, namely the optimization methods and ML techniques used to solve routing problems (Hou et al., 2021). Moreover, integrating GA with RL can leverage and synergize the supremacy of both methods to produce even better results (Yoo et al., 2024), which will be demonstrated in the following sections.

4.2 Proposed Solution

Since this study focuses on the same problem as the previous study, the definition and description of the problem are as explained in Section 3.2 above. The proposed model is an integration of GA and RL algorithms. This model examines several experiments, including the individual approaches of (GA, RL) and hybrid approaches, each consisting of two phases. The first hybrid approach starts phase one with GA which determines the best path and sets it as the initial state for RL algorithm operations in phase two; while in the second hybrid approach a reversed sequence of algorithms is applied, hence starting with RL operations in the first phase and then GA in the second phase, as shown in Figure 4.1.

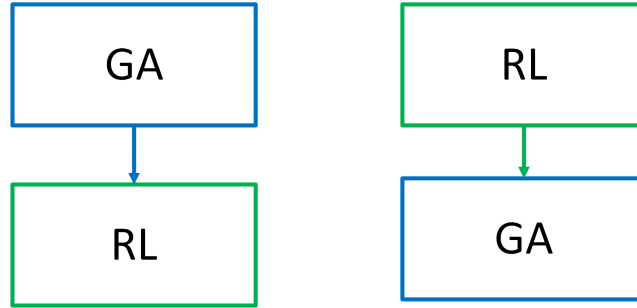


Figure 4.1: Proposed Solutions with Two Hybrid Approaches.

All the details explaining the models processes operations are in the next subsection.

4.2.1 Genetic Algorithms

The GA is the first part of the first proposed model to be utilized, which is the same as discussed in Section 3.3 above. In the first operation of the GA, the initial population is randomly selected considering the energy and capacity constraints. It comprises several subroutes represented by chromosomes, and hence each WCL is represented by a gene. The subroutes consist

of several routes; each route defines the distance between several WCLs that starts from the depot and ends at the disposal point; or it can start from the disposal point and end at the depot interchangeably. These paths or routes determine the entire tour. Each path or tour includes edges representing the distance between two WCLs. Genetic operators are essential components of the GA process, which aims to enhance the population by exploring a spectrum of candidate solutions to find optimal or near-optimal solutions. The GA process operations continue until solutions are obtained that achieve the main objectives of finding the optimal path based on the shortest distance with respect to capacity and energy constraints. Furthermore, the convergence threshold value is used to improve fitness, comparing the best fitness of the current tour and the fitness of the previous tour. The algorithm checks the fitness improvement in subsequent iterations, and when it is below the convergence value, iterations are stopped. Hence, the GA continues iterating until it reaches the maximum iteration number or falls below the convergence threshold value, henceforth terminating at whichever conditions are met first.

4.2.2 The Reinforcement Algorithm

In this study, we examine two different ways of integrating GA with the RL algorithm. The RL is the second part of the first proposed hybrid approach to manage WCS, while it is utilized in the first part of the second proposed hybrid approach. The next subsections describe the two ways of merging GA with RL and how they are set up in the experiments to produce different results for comparison purposes in the analysis section.

4.2.2.1 Genetic Algorithm followed by Reinforcement Learning

GA followed by RL (GARL) is the first proposed model shown in Figure 4.3 for the hybrid integration of GA and RL. RL is a type of ML algorithm that works based on the main idea of trial and error (Girlevicius, 2022). It operates through agent actions depending on the current state successively used in the dynamic environment. Hence, RL learns the optimal policy to solve the problem based on incremental rewards. Here, a policy is a mapping plan with an expected approximate action that takes into account the current state of the environment, where the appropriate action taken is the outcome. RL uses an epsilon-greedy policy (Seidu et al., 2024) in which the action is taken corresponding to the highest Q value and some random actions exploring other locations. The rewards are calculated as the accumulation of the total rewards received from each step of the system obtained by the optimal RL policy over time. The cumulative reward is obtained by a (γ) discount factor. For example, if (γ) is less than 1, it means that the agent prefers the current state over future states. The reward in the model takes into account all the constraints of the fitness function of the GA in terms of capacity, energy, and the shortest path to visit all the allocated locations. The Q value is calculated based on the action and state pairs at each step, and this value determines the best action/optimal action using Q-Learning. The choice of action state depends on the current state and the Q value, which starts from the initial state, from the best path output taken from the GA results. Hence, the RL agent navigates the routing problem by exploring the reward in a Q table and exploring the expected future rewards for each step in the loops. Gradually, in the training phase, the highest Q value is used mostly to obtain the optimal path; the training phase is shown in Figure 4.2 and Algorithm 4. The main Equation 4.1 of RL is explained as follows:

$$Q(s, a) = Q(s, a) + \alpha \cdot (R + \gamma \cdot \max Q(s', a)) - Q(s, a) \quad (4.1)$$

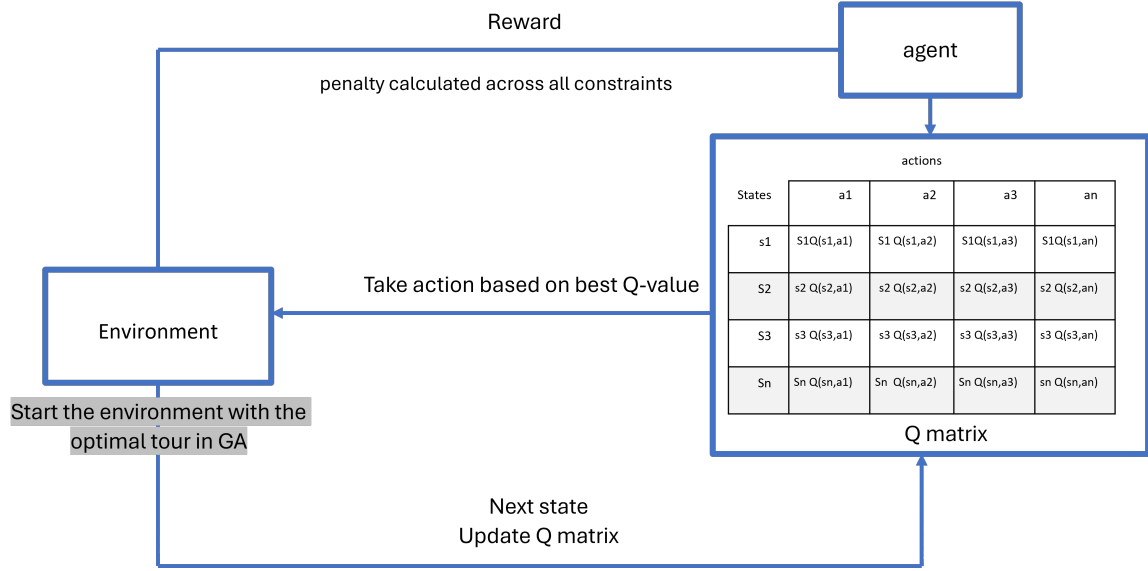


Figure 4.2: The RL Process in the Model of Incorporating the GA with the RL (GARL)

$Q(s, a)$ represents the Q-value of a given state and action, indicating the expected future reward based on both s and α , which is the learning rate alpha at which the information determines that the new information overrides the old information in updates of the Q value. R is the immediate reward gained from the action in a certain state s . γ is a discount factor that represents a constant value of future rewards over time. The $\max Q(s', a)$ represents the highest Q value for the next state s and all corresponding actions a . This equation describes the convergence that continuously interacts with the environment by updating the Q values to the optimal Q values. This approach shows simplicity and is based on a model-free environment. The main formulations of RL are explained in Equation 4.2 and Equation 4.3 as follows:

$$R(s, a) = F(s, a) \quad (4.2)$$

$R(s, a)$ represents the reward of action a in a certain state s , and $F(s, a)$ represents the constraints taken into account in the fitness function that the state and action take based on the values of the multi-objective fitness score.

$$F(s, a) = -10(FCost + FC + FE + FU) \quad (4.3)$$

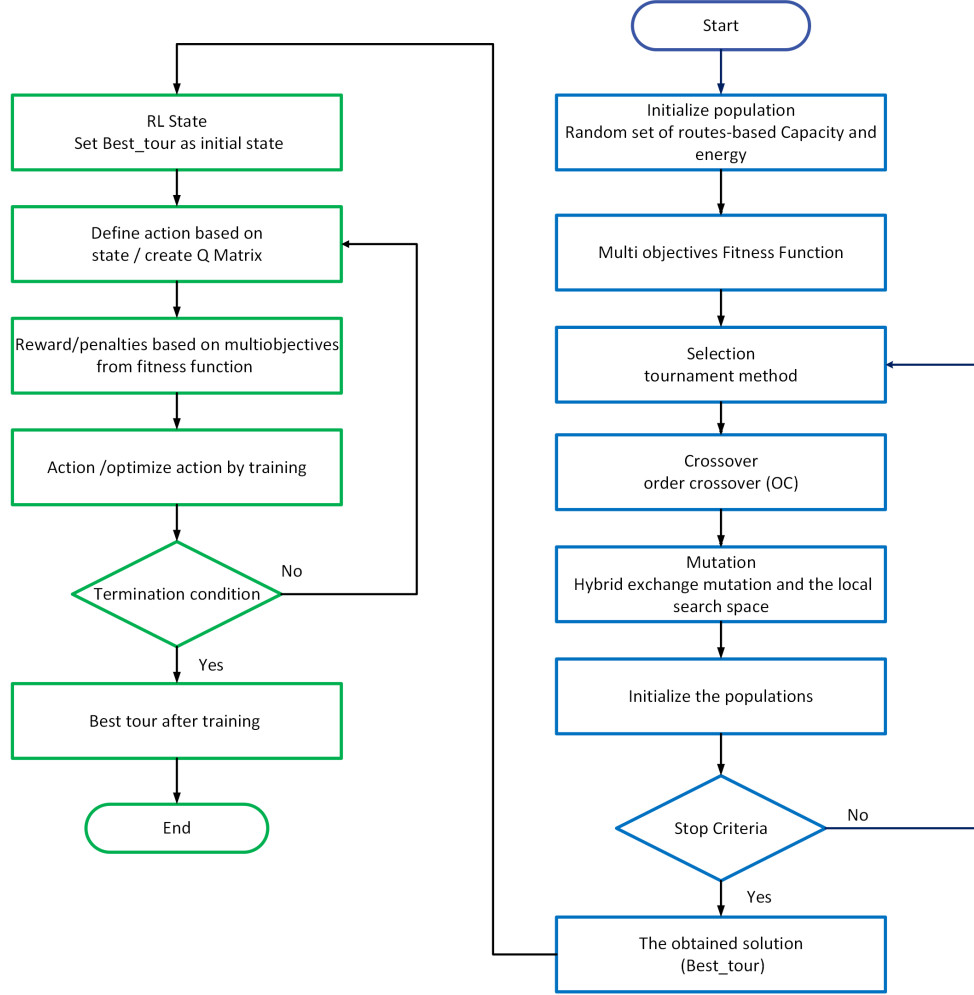


Figure 4.3: The First Method of Incorporating the GA with the RL (GARL).

The multi-fitness function can be calculated for all constraints. FCost represents the total distance of the tour, which calculates the edge distance between each WCL, including the depot and disposal sites. FC represents the capacity constraint, FE is the energy constraint, and FU is the constraint that the UAV must visit all locations in the entire tour. Fitness values give the agent feedback for the total values of the multi-fitness functions that maximize the total reward scores. In other words, these fitness values correspond to the state s and the appropriate optimal actions taken. When actions are determined based on the highest Q value in each state, considering constraints such as capacity and energy, the sequence of these processes pro-

Algorithm 4 Training Algorithm

```
1: Start
2: Initialize necessary variables and parameters
3: for episode in range(epochs) do
4:   state = env.reset(best_route) #Set environment to get initial state as the best route
5:   done = False
6:   episode_reward = 0
7:   for step in range(max_steps) do #Iterates for a maximum number of steps
8:     action = agent.act(state) #The agent selects an action based on the current state
9:     next_state, reward, done = env.move(action) #Environment updates based on the selected action
10:    for route in route_list do #Iterating through each element in the route_list
11:      if isinstance(route, int) then
12:        route_load = get_site_capacity(route)
13:        total_load = total_load + route_load
14:        total_distance = total_distance + calculate_route_distance(route)
15:        num_locations = num_locations + 1
16:        continue
17:      end if
18:      for waypoint in route do #Iterating through each waypoint within the route
19:        Calculate the route_load
20:        Update total_load, total_distance, and num_locations
21:        if route_load  $\geq$  drone_capacity then
22:          Go to the disposal site to empty the load
23:          Update energy_available and calculate the energy_penalty
24:          if energy_available  $\leq$  energy_threshold then
25:            remaining_energy = drone_energy
26:            UAV goes to disposal
27:            UAV goes to depot to recharge
28:          end if
29:        end if
30:      end for
31:    end for
32:    uncollected_locations = total_waste_locations - num_locations
33:    capacity_penalty = total_load / (num_disposal_trips  $\times$  drone_capacity)
34:    cost_paths_penalty = 1 / total_distance
35:    total_penalty = -10  $\times$  (energy_penalty + capacity_penalty + (1 / (uncollected_locations + 1)) +
    cost_paths_penalty)
36:    total_reward = total_reward + reward + total_penalty
37:  end for
38: end for
39: Print Results
```

duces the best path map for WCLs. A capacity penalty is calculated when the action taken for the next location violates the maximum capacity of the drone. Calculate the absolute difference between the current space capacity and the capacity state in the next location and set -10 as a penalty value for the overrun capacity change. It is used to help the RL agent resist the change in maximum UAV capacity. It is displayed in the following Equation 4.4:

$$\text{capacity_penalty}(s, a, s') = -10 \cdot (|c(s') - c(s)|) \quad (4.4)$$

s represents the current state; a is an action for a given state s ; s' is the next state after taking the action; and -10 is a negative value assigned to the penalty ability. $(|c(s') - c(s)|)$ is the absolute difference between the maximum capacity of the next state s' and the maximum capacity of the current state s . An energy penalty is calculated when the action taken for the next location violates the maximum energy of the drone. Calculate the absolute difference between the energy of the UAV in the current location and the energy of the UAV in the next location, and set -10 as the override penalty value in the energy change to minimize energy consumption. It is presented in the following Equation 4.5:

$$\text{energy_penalty}(s, a, s') = -10 \cdot (|e(s') - e(s)|) \quad (4.5)$$

s represents the current state; a is the action for a specific state; and s' is the next state after taking the action; and -10 is a negative value assigned for an energy penalty ability. $(|e(s') - e(s)|)$ calculates the absolute difference between the maximum energy of the next state s' and the maximum energy of the current state s . For the cost penalty, it is important to find the shortest path that is added to obtain the overall rewards. In other words, a cumulative inverse of the total distance is calculated for all routes, with a lower penalty assigned for the shorter path and a higher penalty for the longer path. This is expressed in the following Equation 4.6:

$$\text{cost_paths_penalty}(s, a, s') = -10 \cdot 1 / \text{total_distance}(s, a, s') \quad (4.6)$$

s represents the current state s ; a is the action for a given state; s' is the next state; and -10 is a negative value assigned to a longer trip penalty. The agent must visit all the WCLs in the map plan, gets a penalty if it misses any of the locations, and can visit the locations multiple times when needed based on the shortest route. This is expressed in the following Equation 4.7:

$$\text{UN Penalty}(s, a, s') = -10(1 / \sum(WCL - UN)) \quad (4.7)$$

s represents the current state s ; a is the action for a given state; s' is the next state; and -10 is a negative value assigned to the penalty for the remaining waste locations.

The state indicates the current location of the UAV and determines the next destination based on penalty calculations, with a state space size ranging from 100 to 1000, depending on the waste locations used in the research. The choice of the next state depends on environmental parameters such as remaining energy, shortest distance, reduction in flight time, and maximization of waste load. To clarify, maximization of waste load involves decision-making based on waste load at the next location; if processing that location would exceed the drone's waste load limit and collecting that load would exceed the drone's capacity, it will disregard that location and consider another option with a lower load that does not exceed the drone's capacity. If no suitable candidate is found that meets the capacity limit, the drone will head to a disposal location to empty its load and continue to collect uncollected waste from the remaining locations. The action corresponding to the next valid location, which is of a discrete type, is determined by the method of visiting locations one at a time with a drone until all have been visited. This decision is based on selecting the action that results in better reward

value calculations, considering penalties from various constraints. The reward function is designed to prioritize locations based on their suitability for a serve, influenced by a penalty function. This function includes evaluations for multiple constraints, such as energy penalties, capacity penalties, penalties for uncollected waste locations (if some are left unvisited), and cost penalties for increased distance to the location. A significant penalty value of -10 is applied to strongly discourage violations of constraints, as detailed in the penalty rationale, which highlights the cost of neglecting to avoid invalid actions.

The convergence of the RL algorithm can be effectively monitored and terminated by tracking total rewards and average rewards. Criteria such as maximum episodes, minimal reward change over consecutive episodes, policy evaluation results, and fitness evaluation are used to determine when the algorithm has achieved a satisfactory solution. The convergence criteria effectively ensure that the agent learns a strong policy while also avoiding unnecessary training once satisfactory performance is achieved. This includes certain convergence techniques as outlined follows:

- **Maximum Episode:** The learning process will stop early and terminate when the maximum number of episodes is reached, which is set to 1000 episodes. Early stopping prevents overtraining and saves computational resources.
- **Minimal changes in reward over consecutive episodes:** Convergence will occur when the Q values are updated according to the actions taken, the rewards received and the estimated future rewards. Furthermore, by monitoring the average change in total rewards over the latest episodes until it becomes stable, we can assess convergence. In the other world, if the average reward falls below a threshold of 0.01 over the last 10 episodes, then the algorithm is considered to have converged.

- **Policy Evaluation:** The convergence technique integrates the epsilon-greedy policy with a strategy for monitoring convergence, which is based on the observation of minimal changes in reward over consecutive episodes. This method tracks current and cumulative rewards. After every 10 episodes, the average reward is calculated and it is assessed whether the average reward is fluctuating; if it drops below the threshold of 0.01, this indicates that the learning process has converged. This signifies that the agent is no longer making improvements and has stabilized its policy. Along with the epsilon decay mechanism achieves a balance between exploration and exploitation. This strategy guarantees that the agent converges effectively, demonstrating stable performance and behavior in its policy.
- **Fitness Evaluation:** The fitness function used leads to a lower fitness value with less penalty, indicating that the solution is more effective and therefore acceptable.

4.2.2.2 Reinforcement Learning followed by Genetic Algorithm

RL followed by the GA (RLGA) model is to start with RL algorithm operations that can show better adaptation to dynamic environments, add known data and exploit it to increase rewards, as shown in Figure 4.4, while GA can essentially further explore new potential solutions. Thus, this experiment starts with the operations of the RL algorithm based on the actions of the agent interacting with the environment. The RL interacts with the location data and starts a Q table of zeros, since it has no knowledge of what the values of these actions are in any state. The agent starts exploring the environment and calculating the distances between each location to find the best path based on the epsilon greedy policy within all constraints and requirements. It starts collecting the experience values and updating the Q matrix based on the rewards received and the maximum expected future rewards.

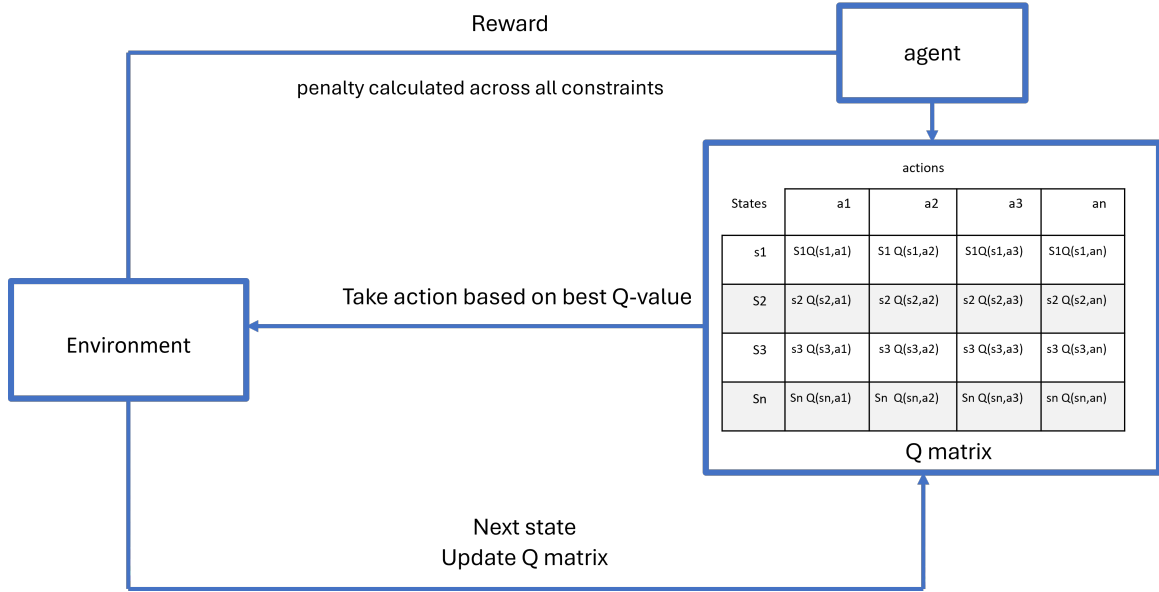


Figure 4.4: The RL Process in the Second Model That Incorporates the RL First then the GA (RLGA)

The process continues until an optimal policy is found that increases the reward values. The final tour output is used as initial data to start with in the GA operations instead of a random population, and it continues to generate more populations from this start. Then it goes through the same GA process as the GA operations refine the best RL tour within the GA process. Figure 4.5 shows the hybrid integration of RL and GA in the proposed hybrid RLGA model.

4.3 Experimentations Results and Results Analysis

The proposed solution modules provide several functionalities for efficient optimization, data manipulation, plotting, and support of advanced scientific computing. This system is generated on an Intel(R) Core (TM) i7-2670QM CPU laptop with 2.20 GHz RAM under Microsoft Windows 10 Pro, using compute engine backend (TPU) with RAM 6.12 GB/334.56 GB and

disk 16.09 GB/225.33 GB. The experimental parameters of GA that were mentioned previously in Chapter 3, Section 3.4, in Table 3.2 and the RL parameters shown in Table 4.1 are used for the three scenarios used in this experiment. These parameter values can be adapted on the basis of the type of drones that can be utilized in the actual system that is eligible for this research. The experiments use synthetic data. The comparison is based on the specified population size and the WCL numbers of 100, 500, and 1000 WCLs, respectively. The GA and RL parameters are fixed and carefully selected on the basis of trial and error and other previous studies. The maximum capacity is 500 for each WCL weighed in kilograms. The speed of the drone is set to 60 km/h; the flight duration is the time consumed during the flight, which is measured in minutes; the power consumed per second is 30 W/s; the nominal power consumption is set to 100 W; then the efficiency factor is set to 0.9 indicating the efficiency of the system being used. All these parameters are taken from the manufacturing data or the factory specifications (Girlevicius, 2022). In addition, the drone capacity is 200K and the drone power is 500W (Seidu et al., 2024). Table 4.1 shows the parameters of the experiments for RL, where the parameters change depending on the number of data, such as the number of WCLs and the number of episodes; therefore, the maximum steps increase as the data increase. Whereas when it fixes the episodes to 1000 episodes, then the maximum number of steps is 100 for all the WCLs. Moreover, the other parameters are fixed as epsilon 0.1, which is the probability that a random action explores better possibilities for improved actions by initial exploration with 100%, and this rate decreases with training until the agent learns its optimal policy. Epsilon decay is set to 0.995, which reduces by 0.5%, which is multiplied by this value after each iteration, effectively reducing the exploration over time. The epsilon minimal parameter is set to 0.01, which means that the exploration rate will not drop below 1%.

The learning rate is set as 0.1, and its control weight is based on updating new information and comparing it with existing data. The higher the value, the faster the exploration will be, while the lowering of the Q values leads to gradual convergence. The discount factor or gamma is set at 0.5%, considering the important influence of future rewards in decision making, where a value of 0 takes only the current reward, while a value of 1 takes into account the acquisition of future rewards. The maximum step is the maximum number of actions each episode can take, hence preventing the agent from falling into an infinite loop. Episodes are the vital parameter that allows an agent to interact with its environment to improve its policy through learning. The size of the state is the number of all possible states of the environment. The action size is the number of possible actions the agent can take in the environment. The model examines the multi-objective improvement in fitness functions by finding the shortest distance while collecting the maximum waste amount and consuming less energy for the entire tour.

The study includes four experiments that aim to compare the performance of algorithms in meeting the objectives of solving the problem. The goal of the experiments is to find the best fitness values and flight time among these algorithms within certain constraints. These algorithms are divided into individual algorithms GA and RL, and hybrid algorithms GARL and RLGA. Table 4.2 shows the best fitness values and the best flight time of the models in different numbers of WCLs among the four experiments. From the experiments, the GARL recorded better results than other approaches in fitness values, recording 4.75, 8.698, and 11.42 for 100, 500, and 1000 WCLs, respectively. Even the flight time is less than other approaches: 39.97 minutes for 100 WCLs, 199.18 minutes for 500 WCLs, and 374.2 minutes for 1000 WCLs. Secondly, the hybrid RLGA recorded better results than both single GA and RL approaches, but not the hybrid

GARL. The recorded fitness values are 5.28, 10.49, and 13.57, for 100, 500, and 1000 WCLs, respectively.

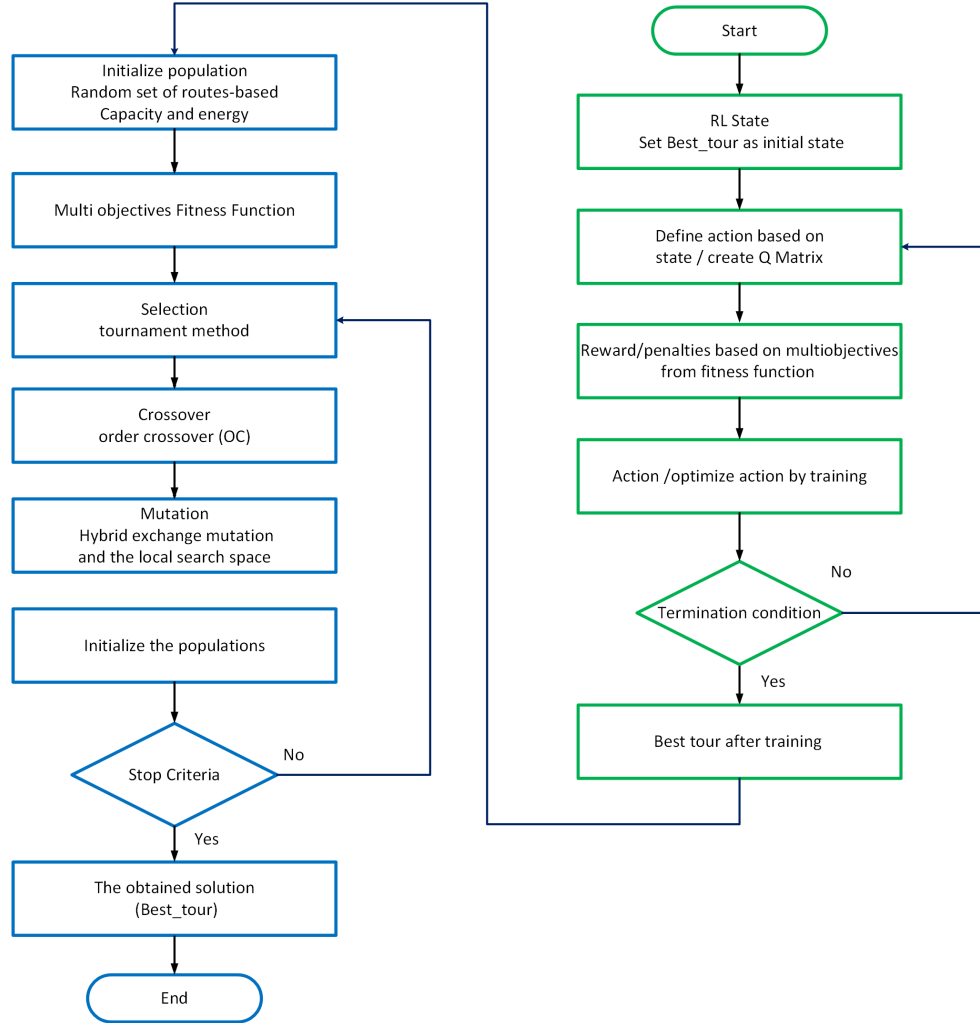


Figure 4.5: The Second Model That Incorporates the RL First, then the GA (RLGA).

Whereas the flight time is 41.02 minutes for 100 locations, 202.5 minutes for 500 WCLs, and 381.97 minutes for 1000 WCLs. The experiment that examines GA as a single algorithm shows that it records better values than those of single RL algorithms. GA records a fitness value of 5.34 that is better than the RL value, which records 5.44 for 100 WCLs. For 500 WCLs, GA records a fitness value of 11.16, which is better than RL, which records 11.23. Furthermore, for 1000 WCLs, GA recorded a fitness value of

15.5, which is better than RL, which records 15.52. As for flight time, GA records slightly better than RL, with 42.24 and 42.48 minutes, respectively, for 100 WCLs. For 500 WCLs,

Table 4.1: RL Parameters in Experiments

Parameter	Value
epsilon	1.0
learning_rate	0.1
state_size	100, 500, 1000
action_size	100, 500, 1000
Exploration Rate	0.5%
Gamma discount factor	0.9
epsilon_decay	0.995
epsilon_min	0.01
max_steps	100,500,1000 locations=100
episodes	100,500,1000 locations=1000

GA records 203.5 minutes, which is better than RL, that records 204.8 minutes. GA records 382.3 minutes, which is better than RL, which is 384.9 minutes for 1000 WCLs. In conclusion, the hybrid approach generally performs better than the algorithms that perform separately. Moreover, the hybrid approach starting with GA then RL performs better than the hybrid approach starting with RL then GA. Additionally, all experiment values are calculated based on a running average for 100 WCLs, which takes an average of 10 runs, then for 500 WCLs, that takes an average of 30 runs, and last for 1000 WCLs, that takes an average of 50 runs. Hence, this shows an inverse relationship between the number of WCLs and the average number of runs, where the more the number of WCLs, the less the average number

of runs is needed. Moreover, those results are presented in the flow charts in Figure 4.6, which demonstrates the best fitness values among all experiments, and in Figure 4.7, which shows the improvements in flight time among the four experiments.

Table 4.2: Experiments Results

Population waste locations	100				500				1000			
Number of Run	10 times				30 times				50 times			
Type of model	GA	RL	GARL	RLGA	GA	RL	GARL	RLGA	GA	RL	GARL	RLGA
Best Fitness	5.34	5.44	4.75	5.28	11.16	11.23	8.69	10.49	15.5	15.52	11.42	13.57
Best Flight Time	42.24	42.48	39.97	41.02	203.5	204.8	199.18	202.5	382.3	384.9	374.2	381.97
Standard deviation for total route	0.705	0.169	0.173	0.548	0.799	0.176	0.172	0.829	0.787	0.167	0.165	0.8

Table 4.3 shows the percentage of improvement when comparing the proposed hybrid approach with the corresponding single algorithms. Performance improvement is observed to increase between the three sets of WCL numbers using the hybrid approach that integrates both GA and RL. For the first and best hybrid approach, the GARL algorithm performance improves over the GA algorithm alone by 11.05%, 22.06%, and 26.32% for the three 100,500 and 1000 WCL, respectively.

Moreover, the performance improvement of the same hybrid algorithm over RL alone records 12.68%, 22.55%, and 26.42% for the three 100,500 and 1000 WCLs. The performance improvement of the second proposed hybrid approach of RLGA also records improvements compared with GA alone of 1.12%, 6.00%, and 12.45% for the three 100, 500, and 1000 WCLs, respectively. Compared with RL alone for the three 100, 500, and 1000 WCLs, it records 2.94%, 6.59%, and 12.56%, respectively. The maximum improvement is observed with the best proposed approach, which is GARL.

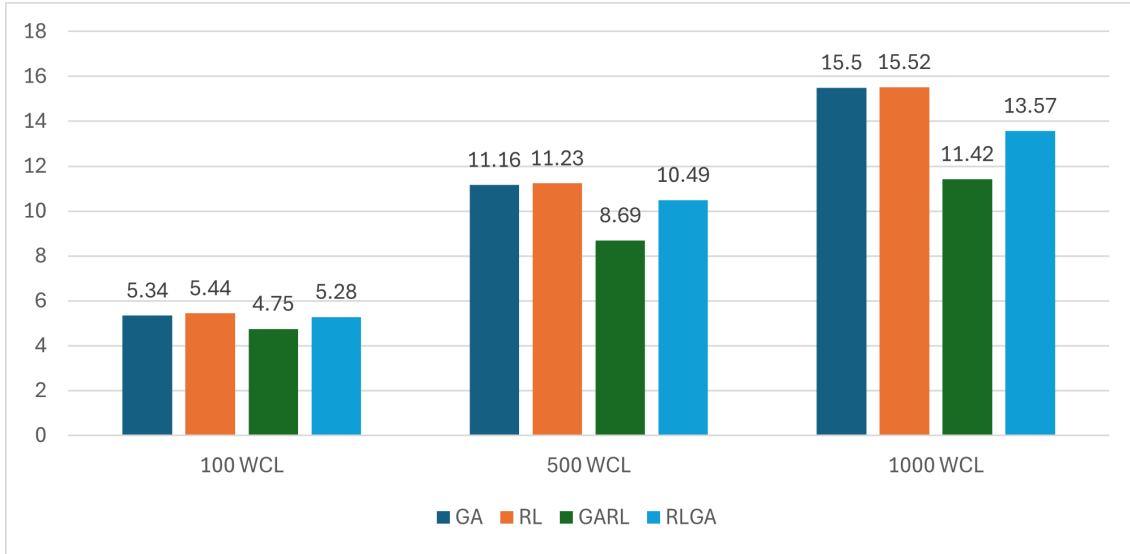


Figure 4.6: Best Fitness Values.

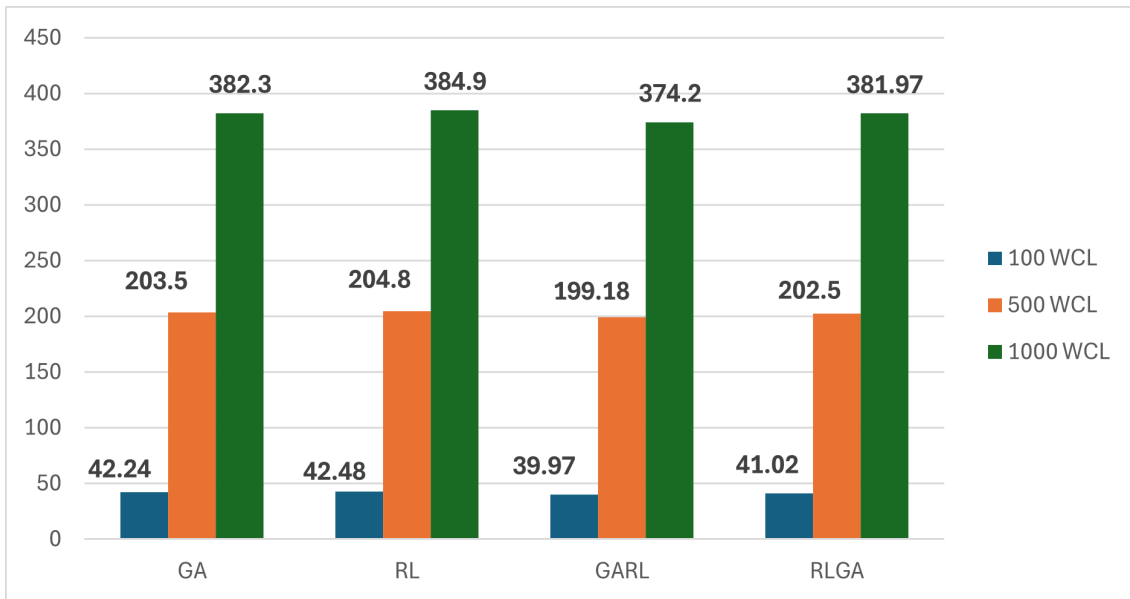


Figure 4.7: Flight Time Values Among All Four Experiments.

Consequently, the percentage of performance improvements is higher for the RLGA hybrid approach compared to the RL algorithm due to its lower performance efficiency, even compared to GA alone. In addition, when the problem is more complex, the performance efficiency improves as shown in the model, where the performance efficiency increases as the

number of WCLs increases. This is demonstrated in Figure 4.8, which shows the performance improvement in percentages among all scenarios of the proposed hybrid algorithms compared to the performance of the other algorithms and to individual algorithms, GA and RL algorithms alone.

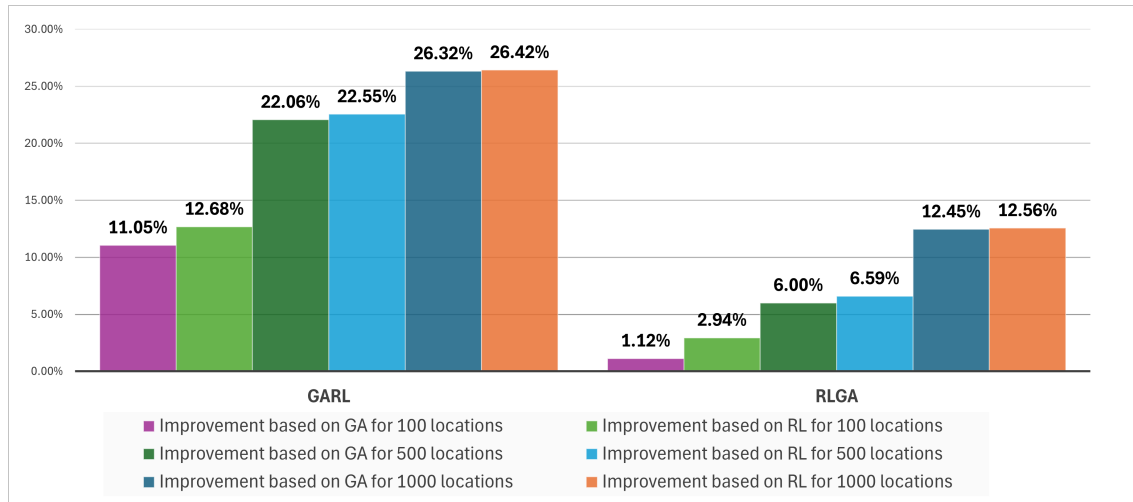


Figure 4.8: Improvement Percentage Among All Algorithms Compared to the Individual Algorithm Performance (GA and RL).

In Table 4.4, the timing improvements are shown along with the increasing number of WCLs from 100 to 1000. When 100 WCLs are compared to the GA algorithm, the GARG is 2.27 minutes better, and when compared to the RL, it is 2.51 minutes better. For 500 WCLs, it improves by 4.32 minutes compared to GA and 5.62 minutes compared to RL. A similar improvement for 1000 WCLs is enhanced by 8.1 minutes compared to GA and by 10.7 minutes compared to RL. Furthermore, the second proposed model, RLGA, performs better than individual algorithms in terms of timing at 100 waste locations, with an improvement of 1.22 minutes compared to GA and an advantage of 1.46 minutes over RL. For 500 waste locations, RLGA improves the timing by 1 minute compared to GA and shows a 2.3 minutes improvement over RL. In the case of 1000 waste locations, RLGA gains 0.33 minutes over GA and achieves a 2.93 minutes improvement over RL.

Table 4.3: The Experiment's Percentage of Improvement

Population waste locations	100		500		1000	
Type of model	GARL	RLGA	GARL	RLGA	GARL	RLGA
The improvement percentage compared with GA	11.05%	1.12%	22.06%	6.00%	26.32%	12.45%
The improvement percentage compared with RL	12.68%	2.94%	22.55%	6.59%	26.42%	12.56%

Table 4.4: Experience Improvements Timing Result

Population waste locations	100		500		1000	
Type of model	GARL	RLGA	GARL	RLGA	GARL	RLGA
GA Time Comparison (Min)	2.27	1.22	4.32	1	8.1	0.33
RL Time Comparison (Min)	2.51	1.46	5.62	2.3	10.7	2.93
RLGA Time Comparison (Min)	1.05	-	3.33	-	7.7	-

This demonstrates that the improvement is greater when compared to the performance of the RL algorithm alone due to its lower performance when compared to GA alone. Moreover, the GARL model shows high performance based on the improvements in the model behavior when it comes to increasing the number of WCLs. Hence, the higher the number of WCLs, the better the performance.

To further analyze the results, standard deviation techniques are used to determine the variation of the data and how the data is spread out from the average dataset. When the standard deviation decreases, it means it is close to the mean of the dataset and indicates the consistency of the data. On the other hand, when the standard deviation is larger, it means more variance and the data has more mean spread than central tendency. The amount of variance affects the results prediction and accuracy. Table 4.2 illustrates that

the standard deviation shows that the hybrid approach with GARL scores better than the other hybrid approach, RLGA, as GARL shows less variance of 0.173, 0.172, and 0.165 compared to RLGA variances of 0.548, 0.829, and 0.8 for 100, 500, and 1000 WCLs, respectively. Thus, the results show more data consistency and less variability. Whereas the RL algorithm, however, has a lower fitness value, which is recorded as 0.169, 0.176, and 0.167 when compared to GA, which is recorded as 0.705, 0.799, and 0.787 for 100, 500, and 1000 WCLs, respectively. Compared to GA, the RL gains better data consistency, although GA outperforms it in other evaluation variables. Furthermore, the experiment results of the proposed hybrid methods are compared with two studies presented in the literature review section that address similar constraints and use other approaches. The first comparative research used a hybrid ACO (Abdulsattar et al., 2023), and the second research used a two-phase approach, namely Maximum Waste in a MWMTT (Kaabi et al., 2023). In this study, GARL is found to score better than the RLGA, and both proposed hybrid models show better results than the comparative studies that used similar three datasets. Each of the compared studies calculates the percentage of reduction in fitness error as an indicator of the equations for the three datasets, with the experiments' datasets based on the number of WCLs. To illustrate this, the best fitness of the comparative study is taken and subtracted from the best fitness of the proposed model; then the value is divided by the best fitness of the comparative study; and the result is multiplied by 100 to convert it to a percentage. This calculation reflects the percentage reduction in fitness error when comparing this study to the comparative study. However, these proposed experimental models do not use the same database used in both comparative research; they just have the same problem to solve and address similar constraints. To compare the two studies with our proposed methods, the best fitness from each comparative study must first be obtained. The percentage of reduction in fitness error

and the best fitness in the comparative studies with the best fitness from the hybrid experiments of the proposed model are also obtained. Table 4.5 and Table 4.6 show that the proposed hybrid methods prove to be a better approach to solving path planning problems than both the hybrid ACO method and the MWMTT method. Moreover, when the number of WCLs that need to be served increases, the percentage of reduction in fitness error decreases and vice versa. In general, GARL has significantly higher performance than other proposed methods when compared to comparative studies. To illustrate this, Table VII compares the hybrid ACO system with the proposed hybrid methods. At 100 WCLs, GARL improves by 81.7%, and RLGA improves by 79.6% better than the first database used in the compared hybrid ACO model. Additionally, at the second database, GARL improves by 82.6%, and RLGA improves by 80.73%. At the third database, GARL improves by 82.65%, and RLGA improves by 80.74%. At 500 WCLs, GARL improves by 66.4%, and RLGA improves by 59.5% better than the first database used in the compared hybrid ACO model. At the second database, GARL improves by 68.21% and RLGA by 61.75%. At the third database, GARL improves by 68.05% and RLGA by 61.45%. Additionally, at 1,000 WCLs, improvements are 55.9% for GARL and 47.7% for RLGA, better than the first database used in the compared hybrid ACO model. Improvements with the second database are 58.35% for GARL and 50.51% for RLGA. Improvements with the third database are 58.09% for GARL and 50.11% for RLGA. The percentage of reduction in the fitness error for the first database in the proposed hybrid approach from the comparative search hybrid ACO is 25.9%. The percentage of reduction in the fitness error for the second database in the proposed hybrid approach from the comparative search hybrid ACO is 27.4%. Additionally, the percentage of reduction in the fitness error for the third database in the proposed hybrid approach from the comparative search hybrid (ACO) is 27.2%. Additionally, Figure 4.9

demonstrates the percentage of reduction in fitness error in proposed GARL and RLGA when compared to the first, second, and third datasets used in the hybrid ACO. Furthermore, Table 4.6 compares the MWMTT method results with the proposed hybrid methods' results. At 100 WCLs, GARL improves by 86.8% and RLGA by 85.4%, better than the first database used in the compared MWMTT model. Additionally, in the second database, GARL improves by 83.79% and RLGA by 81.86%. At the third database, GARL improves by 77.59% and RLGA by 75.09%. At 500 WCLs, GARL improves by 75.9% and RLGA by 71.0%, better than the first database used in the compared MWMTT model. At the second database, GARL improves by 83.79% and RLGA by 81.86%, respectively. At the third database, GARL improves by 59.01% and RLGA by 50.57%. In addition, improvements are recorded at 1,000 WCLs better than the first database used in the compared MWMTT model, with improvements of 68.4% for the GARL database and 62.4% for the RLGA database. Improvements in the second database are 60.55% for the GARL database and 53.25% for the RLGA database. Improvements in the third database are 46.17% for the GARL database and 36.08% for the RLGA database. The percentage of reduction in fitness error for the first database in the proposed hybrid approaches is recorded as 36.1 in comparison to the MWMTT research approach. Furthermore, the percentage of reduction in fitness error for the second database in the hybrid approaches is recorded as 29 in comparison to the MWMTT research approach. In addition, the percentage reduction in fitness error for the third database in the hybrid approaches is recorded as 21.2 in comparison to the MWMTT research approach. Moreover, Figure 4.10 presents a flowchart showing the percentage of reduction in fitness error in each of the different WCLs datasets when comparing the proposed hybrid methods and each of the different datasets used by the comparative research MWMTT. However, these results may also be affected by the nature of the dataset and the number

of cases used in the datasets of the comparative researchers. For example, these datasets contain several WCLs ranging from 21 to 981, which is less than 1000 WCLs in the proposed model, which may affect the reliability of the comparison of the results. The proposed methods are superior to the comparable studies in the following regards:

- The proposed approach comprises four models, each with three experiments that cover the number of WCLs (100, 500, and 1000), a depot, and a waste disposal location. Comparative research with small cases between 10-40, medium cases 41-48, and large cases between 49-89. Hence, the higher figures of our proposed models give them higher reliability and strength.
- Regarding power consumption, the proposed approaches can yield better results because they cover more locations on each trip, recharge when needed, and do not stop early before consuming all of the drone's power, as was the case in the compared studies.
- The linear program could not generate solutions that solve instances with more than 40 sites, hence the lower bound is used to evaluate the performance of the medium- and large-sized experiments. Therefore, the results obtained were in line with those obtained for small-sized instances, which is an obvious limitation that is overcome in our study approach.

Overall, for all the above results, the proposed methods have been proven to have high reliability with internal and external validity. These methods demonstrate the efficiency of algorithmic modification that can use real-world datasets and integrate them into the system. Its internal validity comes from the positive results and the superiority of the outcomes of the proposed models over the results of the comparative studies and the achievement of the objectives of this study. Its external validity comes from its consistency

Table 4.5: The Comparison Between the Methods Research (hybrid ACO) and the Proposed Approaches Within GA.

GA dataset (Num. locations)	Percentage reduction in fitness error of DS1 (hybrid ACO)	Improvement via GARL	Improvement via RLGA	Best fit of DS2 (hybrid ACO)	Improvement via GARL	Improvement via RLGA	Best fit of DS3 (hybrid ACO)	Improvement via GARL	Improvement via RLGA
100 WCL	25.9	81.7%	79.6%	27.4	82.6%	80.73%	27.2	82.65%	80.74%
500 WCL	25.9	66.4%	59.5%	27.4	68.21%	61.75%	27.2	68.05%	61.45%
1000 WCL	25.9	55.9%	47.7%	27.4	58.35%	50.51%	27.2	58.09%	50.11%

Table 4.6: The Comparison Between the Methods Research MWMTT and Proposed Approaches Within GA.

GA dataset (Num. locations)	Percentage reduction in fitness error of DS1 (MWMTT)	Improvement via GARL	Improvement via RLGA	Best fit of DS2 (MWMTT)	Improvement via GARL	Improvement via RLGA	Best fit of DS3 (MWMTT)	Improvement via GARL	Improvement via RLGA
100 WCL	36.1	86.8%	85.4%	29	83.79%	81.86%	21.2	77.59%	75.09%
500 WCL	36.1	75.9%	71.0%	29	83.79%	81.86%	21.2	59.01%	50.57%
1000 WCL	36.1	68.4%	62.4%	29	60.55%	53.25%	21.2	46.17%	36.08%

with previous research, with the possibility of generalizing the result and applying it to real-world situations. Several physical factors, such as climate change, wind, take-off, and landing time, are not considered in this study. In conclusion, the best proposed model is the one that integrates GA and RL (GARL). When analyzing the algorithm process, it is noticed that the routes are optimized across a large number of solution spaces to reach a near-optimal solution. It can accommodate various constraints to minimize costs based on minimizing energy consumption, flight time, and total distance through robust fitness functions. The GA is highly flexible in adapting to multiple dynamic scenarios and complex environments, as it can handle large amounts of data and find logical solutions through parallel processing. Parallel processing is highly efficient, as it can explore more solutions si-

multaneously while accelerating computation and optimization using large amounts of data. One of the most notable features of the GA is its ability to avoid local optimal outcomes and find global optimal outcomes, enabling the search for better paths in complex environments due to its reliance on derivatives. The GA consists of more robust techniques, based on its ability to handle data noise and its flexibility to change, making it more suitable for real-world applications. In addition to its ability to handle constraints multi-objectively, the generalization of the solution is based on the finding of the best tour by GA, which is more reliable and near optimal. Its outcome population starts as the input population in the second phase of the first model, producing better results than the proposed second model, which uses RL in the first stage and then GA. RL, on the other hand, can ensure efficiency improvements over time and more navigation, hence dynamic optimization and adapting to changes to improve real-time decision making. RL also performs well in complex problems with multiple variables and high-dimensional space. Conversely, RL can lead to poor balance in exploration and exploitation, which can lead to missed opportunities at certain times or reduced performance. Furthermore, RL often requires more training time and computational resources to produce better results. Therefore, the results of the proposed model GARL are significantly better than those of RLGA. The figures show the fitness values, which reflect the balance between all the calculated constraints, including energy consumption, total distance, capacity, and servicing all WCLs.

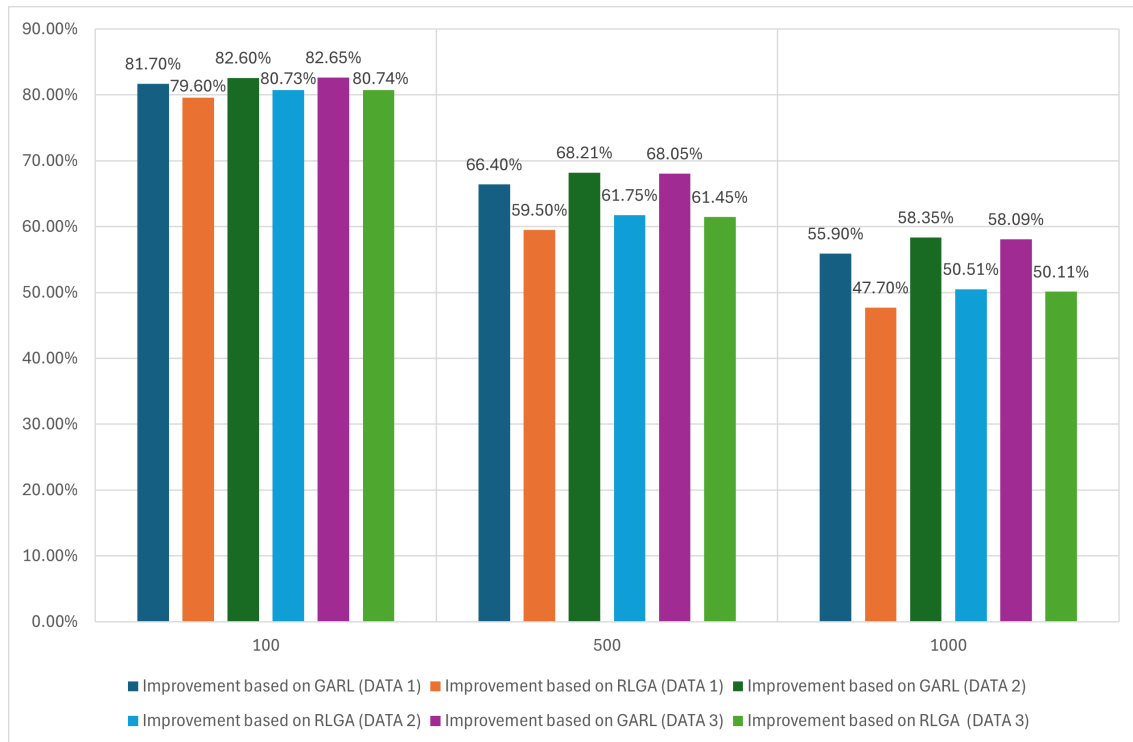


Figure 4.9: The Percentage Reduction in Fitness Error That Was Compared Between the Proposed Hybrid Approaches with Hybrid ACO.

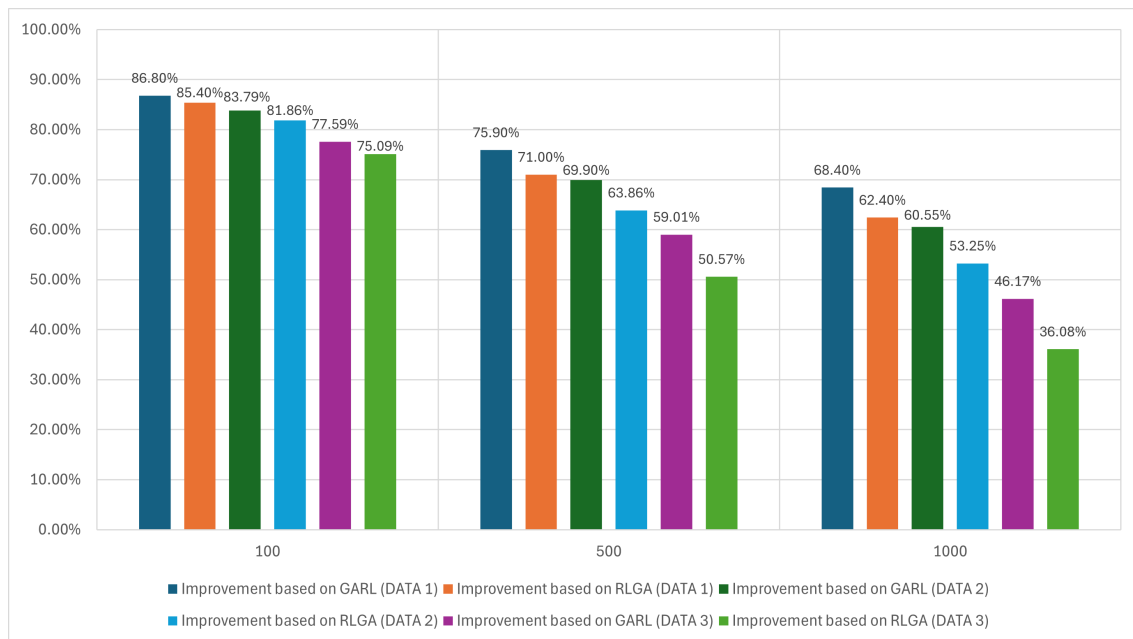


Figure 4.10: The Percentage Reduction in Fitness Error That Was Compared Between the Proposed Hybrid Approaches with MWMTT.

4.4 Summary

The WCS is part of the WMS and is a vital sector for a prosperous lifestyle and economy, especially in smart cities. This thesis proposes another study aiming at finding the best solution for routing problems with suitable tours in complex scenarios that minimizes WCS cost while addressing its constraints multi-objectively by a hybrid integrative model of multi-objective GA and RL to optimize the WCS process. The proposed system consists of several experiments utilizing hybrid algorithms with two phases integrating GA and RL to generate synthetic data for the experimental models within specific predefined constraints, such as drone capacity, energy consumption, flight time, flight distance and servicing all WCLs using a single UAV. Each of the four experiments is also divided into three stages, including 100, 500, and 1000 WCLs. The system first compares two hybrid algorithms of the GA and RL. The first proposed model starts the first phase with GA operations first by synthetic data generation and then proceeds with RL operations in the second phase (GARL), and the second model starts the first phase with RL and then GA in the second phase (RLGA). Then, two more proposed models that test the GA and RL performance individually are compared with the hybrid models. The performance of hybrid models is better than that of individual models, while the GARL hybrid model is found to be the best approach amongst all. It records a notable improvement in fitness value and flight time compared to RLGA, recording higher performance percentages of 10.04%, and 2.56% minutes at 100 WCLs; 17.16% and 1.64% minutes at 500 WCLs; 15.84% and 2.03% minutes at 1000 WCLs. Moreover, compared to the GA algorithm, it shows higher performance percentages of 11.05% and 2.27 minutes at 100 WCLs; 22.06% and 4.32 minutes at 500 WCLs; and 26% and 8.1 minutes at 1000 WCLs. In addition, compared to the RL algorithm, it records higher performance percentages of 12.68% and 2.51 minutes at 100 WCLs; 22.55% and 5.62 minutes at 500 WCLs;

and 26.42% and 10.7 minutes at 1000 WCLs. The results of the proposed hybrid models have also been compared with the results of other previous works and have shown significant improvement. Hence, this integrative hybrid model synergizes the best features of both GA and RL algorithms and enhances application in dynamic environments. The model achieves a reliable high-performance system under multiple constraints in complex environments.

Chapter 5

Multiple UAVs for Waste Management Using a Hybrid Optimization and Machine Learning Approach

5.1 Introduction

This chapter discusses a third study that aims to improve WCS by solving the same problem targeted in the previously proposed models by finding the best solution to trajectory problems through appropriate tours that minimize costs while addressing constraints, however, using multiple drones. Multiple UAVs are used as waste collection vehicles, ensuring that WCL assignments are efficiently distributed among those multiple UAVs based on the nearest WCLs and finally disposed of in a central landfill. The proposed models utilize a hybrid approach that integrates four methods: optimization methods/GA, machine learning techniques/RL, K-means algo-

rithm, and A* algorithms, which can be a very powerful combination to solve complex problems such as routing problems. Hence, the model consists of two stages. The first stage is operating with GA, which takes its best outcome path and sets it as an initial state used by a machine learning algorithm, namely RL. Therefore, in the first stage, the GA can find a near-optimal path that minimizes computation time and reduces cost, thus handling complex and large problems when building efficient, well-defined fitness functions. On the other hand, in the second stage, RL can interact with the environment more appropriately by achieving an optimal path based on a complex dynamic environment, and handle problems that require multi-objective solutions with greater flexibility. This model also uses a K-means algorithm in both stages to distribute WCL assignments among all drones based on the locations nearest to them, which are defined as clusters. The A * algorithms are then used internally for each drone path to find the shortest path. The reassignment technique is an added technique that is used to ensure that a drone completes its tasks first before allowing it to receive a new task from another drone, assigning it more tasks. This technique achieves balance and reduces model operations' costs. This combination of algorithms leverages and synergizes the strengths of each algorithm to produce a powerful model.

5.2 Problem Definition and Description

As mentioned above, this third study of our thesis aims to improve WCS by solving the same problem targeted in the previously proposed models by finding the best solution to trajectory problems through appropriate tours that minimize costs while addressing constraints, however, using multiple UAVs. The model includes multiple drones that use the dynamic task reassignment technique to adjust the operations of the collection process and

balance drone task performance. Each drone is assigned tasks to collect waste from a certain number of specified locations. In addition, the model ensures that the waste load volume does not exceed the drone payload capacity. Hence, before this occurs, the drone unloads the waste at the disposal location and then continues servicing the remaining locations. Furthermore, when the drone's power almost runs out and it cannot continue the entire route, it returns to the depot, recharges, and then continues to collect waste from the remaining WCLs that need to be visited. Once all assigned locations have been served, the drone returns to the depot. Thence, to further improve the model and reduce cost, another proposed approach, namely the reassignment technique, is to reassign tasks among drones to add more balance to the entire process. In general, this approach aims to minimize the cost by reducing the overall distance along the route.

The optimization problem graph is modeled on the graph $G(V, A)$, where V is the set of nodes $V = \{v_0, v_1, v_2, \dots, v_n, v_{n+1}\}$, there is a set number of drones represented as $K = \{K_1, K_2, \dots, K_m\}$, where d represents a unique drone and m is the number of drones. The routes of each drone can be defined as $K_i = \{D, K_1^i v_1, K_2^i v_2, \dots, K_{n_i}^i v_{n_i}, S, D\}$, D is the depot from which the drone starts their tour from with full charge and empty tank space. In other words, each drone is assigned a certain route that includes several WCLs. The model mechanism to fulfill constraints and problem notations is used in the same way as previously modeled and is illustrated in Section 3.2.

5.3 Proposed Solution

The proposed solution further improves the hybrid approach from the previous study that uses the best proposed model GARL that integrates the GA and RL algorithms in a two-stage process, using the K-means algorithm to

cluster multiple tasks for each drone based on the nearest locations, address capacity and energy constraints, and then use the A* clustering algorithm to adjust each route based on the shortest distance. The operations of the two added algorithms are run through the two stages of the GARL model. The proposed hybrid model is shown in Figure 5.1. To illustrate more, previous studies on UAV-based WCS have limited research on multiple UAVs, particularly on minimizing energy consumption and managing dynamic environments. Many of these studies have not adequately addressed the various constraints and instead prioritized handling them; when one constraint is preferred over the other, it will decrease the effectiveness of specific situations in the model. Several proposed models, which do not adequately consider real-world constraints, are not applicable in practical situations and experience significant computational overhead due to inefficient coverage techniques. In contrast, the proposed approach addresses these issues by combining the GARL model, which is recognized as a highly effective model for solving WCS for a single UAV. GA and RL can optimize task assignment and the learning process, thus improving the decision-making model. The model addresses many constraints to support dynamic and complex environments. To adapt this model for use with multiple UAVs, the advanced GARL model is utilized and modified to manage multiple UAVs. This model integrates the strengths of both GA and RL. Additionally, K-means clustering is incorporated to categorize areas based on the number of drones utilized in the model and can be adjusted according to the available UAVs based on the shortest distance between the WCLs, allowing parallelism among waste locations. Once the routes for a specific cluster are established, it also considers capacity and energy constraints. The method was chosen for its simplicity and scalability, particularly important for large and dynamic urban areas, and it reduces computational overhead as it has been utilized in numerous research studies. Additionally, k-means can be

enhanced with other algorithms to complement their functionality, as it is believed that each algorithm can introduce various handling constraints and together can create an effective model. Based on this concept, the A* algorithm is implemented within each cluster to ensure that the shortest distance is found for each cluster, promoting energy efficiency and lowering costs. The A* algorithm was selected because it has proven effective in finding the shortest distance in various studies. These hybrid methods enable UAV models to improve and adapt their task allocation over time within complex and dynamic environments, thus addressing the limitations found in other studies by offering a scalable, efficient and adaptable solution for real-world WCS using UAVs.

5.3.1 Genetic Algorithms

The GA is the first part of the proposed model to be implemented. The initial population is generated randomly based on capacity and energy constraints. The population represents the entire tour and consists of several subroutes represented by certain chromosomes; each of which represents different solutions. Each of these solutions contains information such as the WCL, which is represented by a gene that is classified as a unique WCL. The subprograms consist of several routes; each route determines the distance between several WCLs, starting from the depot and ending at the disposal, or can start from the disposal and end at the depot interchangeably. Each route, or tour, includes edges that represent the distance between two WCLs. The model uses a multi-fitness function that trades off all constraints and system objectives. The quality of the fitness function must be evaluated in each generation. This depends on addressing the main constraint of finding a short tour, managing waste capacity, using drone energy, and visiting all WCLs as a multi-fitness function.

The selection of a chromosome from the parent population is based on

GA operators. The WCLs to be populated by offspring must not be pre-selected. If pre-selected, the WCL is replaced by a corresponding WCL that has not been previously visited. Additionally, each corresponding WCL is moved in an orderly manner; hence, if it is identified in the offspring, that location must be skipped and continue moving to the next location until all locations have been visited and populated. The mutation process combines exchange mutation and local search. The exchange mutation involves selecting a WCL and swapping between location selections. The local search space used for routing problem solutions involves selecting and swapping two vertices that can evaluate solutions and move to the next nearest neighbor. The key point is that tasks distributed among multiple drones are based on the K-means algorithm.

5.3.2 K-means Algorithm

The clustering of each route is built on the k-means algorithm within the WCL mapping system, which considers K distinct clusters with similar features and shared conditions which are shown in Figure 5.2. Each cluster of drones has a center called the center of gravity. The K-means aim to minimize the total squared distance between the WCL and the centers of gravity of the clusters. The algorithm randomly selects the center of gravity. Each waste dataset is clustered based on the closest center of gravity, creating K clusters. In each run, the algorithm reorganizes the clusters taking the average of all data locations. As the center of gravity continually changes, the locations are reassigned to the closest center of gravity. This process continues until the sum of distances between WCLs is minimized or the maximum number of iterations is reached. As a result, a cluster K is created for all drones based on the shortest distance and given the energy and capacity constraints that must be addressed for the locations of all drones. The distance between each node is calculated and the energy and capacity

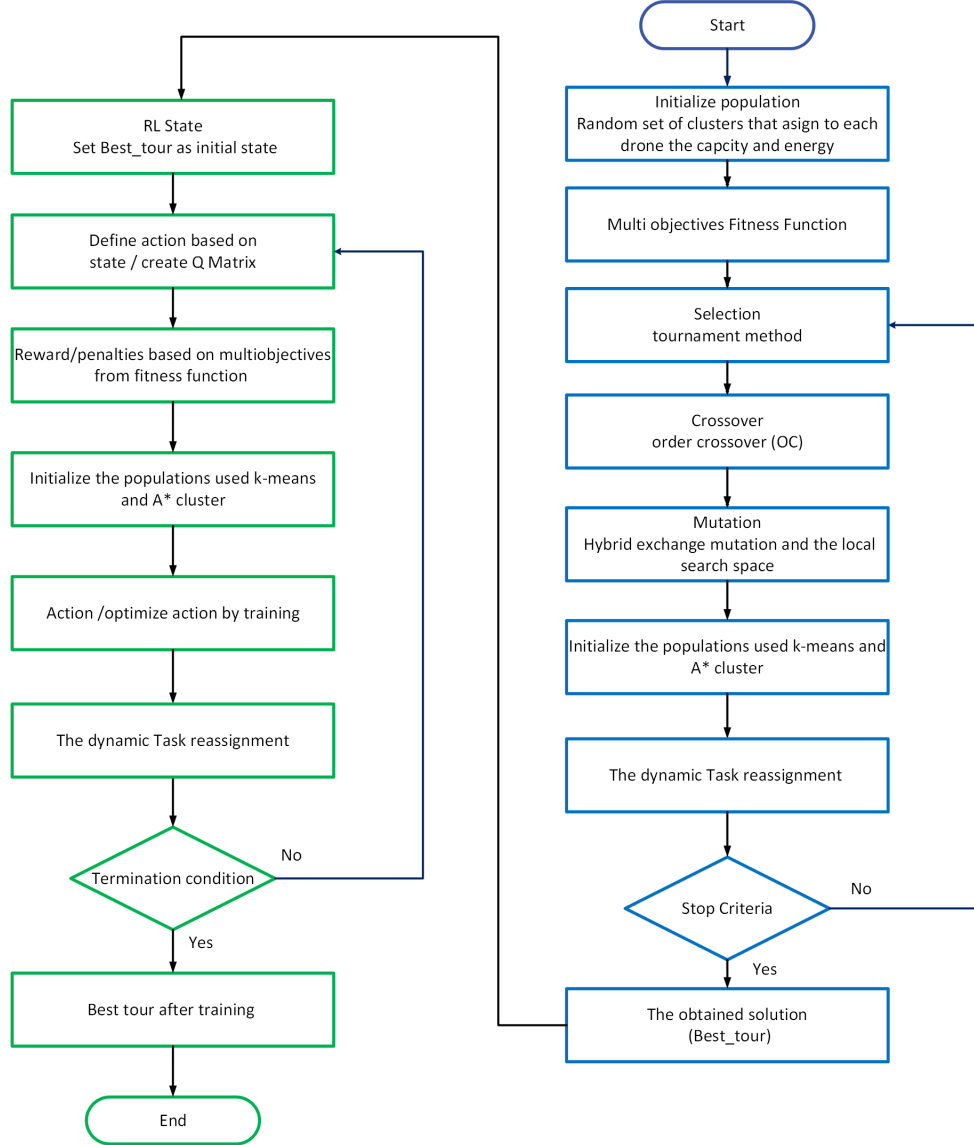


Figure 5.1: The Proposed Hybrid Model for Multiple UAVs.

measures to the next node are calculated. Hence, if the UAV exhausts its capacity load, it goes to the disposal to be unloaded; and also, when it runs out of energy, it goes to the disposal to be unloaded, then goes to the depot to recharge and continue serving all nodes. It can optimize its performance and cost based on reassignment techniques. Additionally, when assigning the clusters to all drones with different numbers of WCLs to be serviced, another approach is to reassign tasks to the drones with fewer WCLs as-

Algorithm 5 Cluster Method for Multiple Drones K-Means Cluster Algorithm

```
1: Start
2: Initialize nearest_neighbor function
3: Initialize the routes for the drones
4: sorted_genes = []
5: Set current_location = start_location (remaining_capacity, remaining_energy)
6: while loop is not complete do
7:   next_gene = min(genes, key=lambda x: calculate_distance(current_location, x))
   Based on nearest locations
8:   distance = calculate_distance(current_location, next_gene) Find the nearest unvis-
   ited location
9:   if remaining_capacity  $\leq 0$  then
10:     route.append(disposal) Go to disposal if capacity is full
11:   end if
12:   if remaining_energy  $\leq$  threshold_energy then
13:     route.append(disposal)
14:     route.append(depot) Go to disposal to empty waste, then recharge at depot
15:   end if
16:   if nearest_location is valid then
17:     (loc_idx, loc_data) = nearest_location
18:     route.append(loc_idx) Add waste location to the route
19:     visited_locations.add(loc_idx) Mark this location as visited
20:     Update drone capacity and energy based on collected waste and distance
21:     Optimized route for each drone
22:   end if
   If no more locations are accessible, return to disposal and then depot
23:   route.append(disposal)
24:   route.append(depot)
25: end while
```

signments from the drones with more WCL assignments to maximize the utilization of available resources. The entire process is repeated until all WCLs are serviced in a specified time. Thence, after K-means clusters the routes for each drone, the A* algorithm is used on each route for each drone to find the shortest route from the initial depot to the end of each route to further minimize the cost of each route. This process is shown in Algorithm 5 and Algorithm 6.

5.3.3 A-star (A*) Algorithm

After the K-means clustering process has compiled the routes for each drone, the A* algorithm is used on each drone's route to find the shortest route from the initial depot to the end of each route using a weighted graph to minimize the cost of each route. Sets a node based on the output of the K-means clustering process and determines the final node within that cluster. The cost for each node is weighted based on the shortest distance, while considering the energy and capacity costs. If it becomes exhausted due to the waste load, it is disposed of for unloading. However, if it runs out of energy by the next node, it goes to the disposal to unload the waste and then returns to the depot to recharge and continue serving all nodes. To find the best node, iterates by finding the lowest value fit node, which is considered to be the best cost. These nodes are updated if they find a better fit than the current best node, and hence the least-fit node is removed from the open set and added to the closed set. The current node's neighbor is evaluated, and if it is in the close set, it is ignored. Otherwise, it is added to the open set and evaluated based on fitness calculations based on the shortest distance, the lowest energy, and the largest capacity costs. If it is already in the open group, it is compared with the new value and updated based on the best fit. This process is continuously updated until the open group is empty and reaches the final node, which is the goal node shown in Figure 5.2. It can optimize its performance

and cost based on reassignment techniques. Hence, when the clusters are assigned to all drones, it uses another approach where it reassigns some of the tasks to the drones that have fewer WCL assignments than the drones with more WCL assignments to better utilize the available resources as much as possible. The entire process is continuously repeated until all WCLs are served in a given time. This process is shown in Algorithm 6. Moreover, the A* algorithm process is activated after the cluster is created from the K-means algorithm. It will start and end at the depot. The process will be illustrated as follows:

- Initialize the Open set, which stores nodes for waste locations that need exploration. Each entry contains the location of the waste, the total remaining cost, the energy, and the capacity.
- Initialize the Closed set to track visited waste nodes.
- Calculate the cost between each pair of nodes and from the start node to the final goal node.
- The cost is initialized with a heuristic, the Euclidean distance from start to destination.
- The nodes with the lowest cost are considered the best nodes, and if these nodes are already in the Open set, they will be ignored in favor of visiting another node. When the current node is the target node, the path will be constructed.
- To explore neighbors, add the current node to the Close set, then retrieve all neighbors that are not the current node.
- The neighbors will evaluate each node by calculating the energy costs and usage of the waste capacity from the current node to the neighboring node. If a node is already in the closed set or if the energy

or capacity is violated (due to low energy or full capacity), the node will skip that neighbor and select another one. To illustrate the energy cost, if the energy consumed is greater than what is available or if the next node cannot be reached with sufficient power, it will skip this node. Regarding capacity usage, if the capacity for the next node would surpass the drone's capacity, that node will be skipped. If the new node has a better score, the route will be updated and considered the best route. In addition, if the new neighbor is not already in the queue, it will be added to the open set with updated scores and tentative energy and capacity values until the optimal route within resource limits is found.

The travel cost function for each cluster is described in Equation 5.1: as follows:

$$f(n) = g(n) + h(n) \quad (5.1)$$

$f(n)$ represents the estimated total cost, $g(n)$ is the accumulated travel cost function from the start to the current node, measuring the cumulative energy used over a specific distance from the depot node to the current node. The capacity usage is considered only as a feasibility filter and does not directly contribute to $g(n)$. $h(n)$ represents the heuristic cost estimation from the current node to the goal, calculated using the Euclidean distance, which estimates the energy required to travel from the current node to the goal.

5.3.4 Reinforcement Learning Algorithm

The second phase of the hybrid approach to optimize WCS is RL (Andrew, 1998). The best tour of the GA output represents the initial state that starts with the state space of the environment. The K-Means algorithm is used to cluster waste points around the centroid, which is based on the shortest distance created from geographic coordinates. After the cluster is initialized,

the A* algorithm is used to find the shortest waste collection route distance from each cluster on each route to achieve the best fitness value. As a result, each drone operates within its cluster, and the RL algorithm agent navigates the routing problem by storing the reward in the Q table and storing the expected future rewards for each step in the loops. The Epsilon greedy policy (R. Liu et al., 2022) is used to find the optimal policy based on incremental rewards. Figure 5.2 shows the adopted procedure for clustering using multiple drones. RL learns the best policy based on these cumulative rewards. The rewards are calculated as the accumulation of the total rewards received from each route based on the number of drones that are considered to have the best rewards from the entire tour. Each drone that successfully collects the assigned amount of waste within its mission at the given geographic coordinates and with the least amount of energy consumed receives positive rewards. On the other hand, drones that include inefficient routes with locations out of range or exceeding the energy and capacity defined limits receive negative rewards. The algorithm performs continuous improvement based on the iterative experience of the RL agent. In addition, the k-means algorithm allows the RL agent to learn from the structure of the environment. The goal of the proposed method is to reduce the complexity and allow the RL agent to be more tolerant, and hence optimize the routes better. Training ends when it finds the best fitness solution or when the iteration reaches the maximum number of iterations.

5.3.5 The Dynamic Task Reassignment Approach

The dynamic task reassignment approach is a procedure used to adjust and improve the efficiency of the clustering process, and hence further reduce costs. This approach works with specific technologies that will track all current statuses of drones, including their energy and capacity statuses, their locations, the number of WCLs required to be serviced, and the time they

take to complete their tour. If any drone terminates its tour early, it is identified and designated as a mission-free drone. Then tasks are reassigned, taken from most of the drones that still have WCLs to be visited by taking a calculated number (for example, half) of the remaining locations and re-assigning them to the mission-free drone. This process continues until all tasks have been performed or all WCLs have been serviced. This method ensures balanced mission assignment and greater efficiency while reducing the time, energy, and route distance of the entire tour; hence, reducing the total cost. In simpler terms, the moderately sensitive approach to task re-assignment based on load balancing among drones allows up to 50% of the tasks to be reassigned during the optimization phase. The selection of tasks for reassignment is determined by the nearest neighboring waste locations among the drones. This approach will lower the computational cost, which tends to be higher, making it more efficient due to improved energy consumption and reduced distance. In summary, this method features extensive exploration with intelligent constraints, resulting in better quality outcomes, faster convergence, and enhanced model efficiency. The algorithm 7 shows the reassignment technique.

Figure 5.3 shows an example of the reassignment method where Drone 2 has fewer tasks, including a 9 WCL assignment, and Drone 1 has more tasks, such as a 17 WCL assignment. Therefore, Drone 2 terminates its task early and is free to take on more tasks from the active drone. To do this, the reassignment program checks the drone's current location and calculates all the required data (distance, capacity, and power). Then, the WCLs are divided between the free drone and the active drone to achieve balance. In the example, the remaining WCLs are not distributed evenly, but rather are divided on the basis of clusters of the nearest locations. Therefore, Drone 2 reassigned with Drone 1. The route is updated by reassigning tasks so that Drone 1 reduces its WCL assignment from 17 to 14, while Drone 2

increases its number of WCL assignments from 9 to 12. Hence, a relative balance between the drone tasks' allocation is ensured.

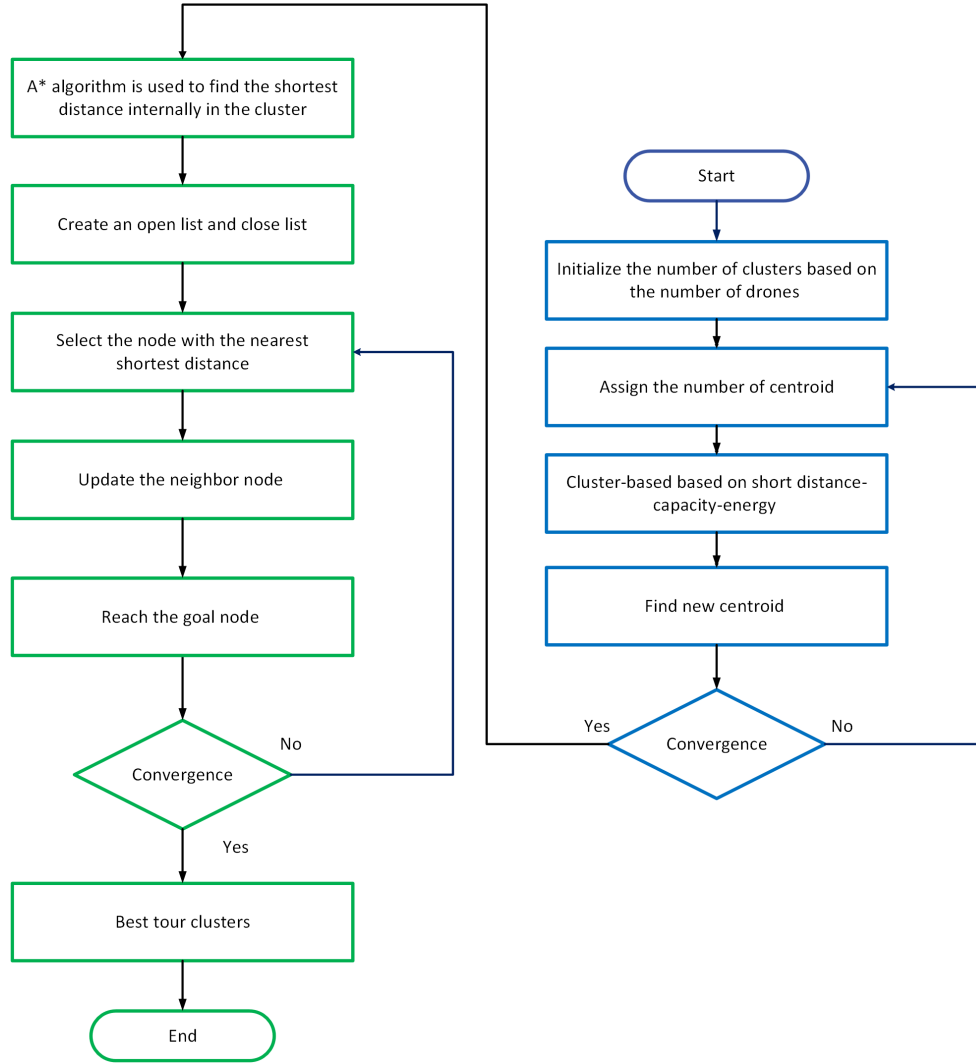


Figure 5.2: The Clusters of the Multiple Drones.

5.4 Experimental Results and Results Analysis

The proposed solution environments are the same as those described in Chapter 4, Section 4.3. The same applies to the GA operation parameters that were previously described in Chapter 3, Section 3.4, in Table 3.2. Furthermore, the parameters in the RL operations are as mentioned in the

Algorithm 6 Cluster Method for Multiple Drones using A* Algorithm

```
1: Start
2: Initialize the open set as a priority queue with the start node
3: open_set = []
4: heapq.heappush(open_set, (0, start))
5: came_from = {} # To reconstruct the path
6: for gene ∈ waste_genes do # Initialize the g_score and f_score dictionaries for all nodes
7:     g_score[gene["Index"]] = float('inf')
8:     f_score[gene["Index"]] = float('inf')
9: end for
10: g_score[start["Index"]] = 0
11: f_score[start["Index"]] = calculate_distance(start, goal)
12: closed_set = set() # Track visited nodes
13: while open_set is not empty do
14:     current_priority, current, energy = heapq.heappop(open_set)
15:     if current["Index"] ∈ closed_set then
16:         continue
17:     end if
18:     if current == goal then
19:         continue reconstruct_path(came_from, current)
20:     end if
21:     closed_set.add(current["Index"])
22:     neighbors = [gene for gene in waste_genes if gene != current]
23:     for neighbor ∈ neighbors do
24:         Compute Energy And Capacity(current, neighbor)
25:         if neighbor["Index"] ∈ closed_set then
26:             continue
27:         end if
28:         if tentative_energy > energy_limit or tentative_capacity > capacity_limit then
29:             continue # Skip this neighbor
30:         end if
31:         if tentative_g_score > g_score[neighbor["Index"]] then
32:             came_from[neighbor["Index"]] = current
33:             g_score[neighbor["Index"]] = tentative_g_score
34:             f_score[neighbor["Index"]] = tentative_g_score + calculate_distance(neighbor, goal)
35:             # Add neighbor to open set if it's not already there
36:             if neighbor ∉ [i[1] for i in open_set] then
37:                 heapq.heappush(open_set, (f_score[neighbor["Index"]], neighbor, tentative_energy, tentative_capacity))
38:             end if
39:         end if
40:     end for
41: end while
42: Return the Optimal path from start to goal
```

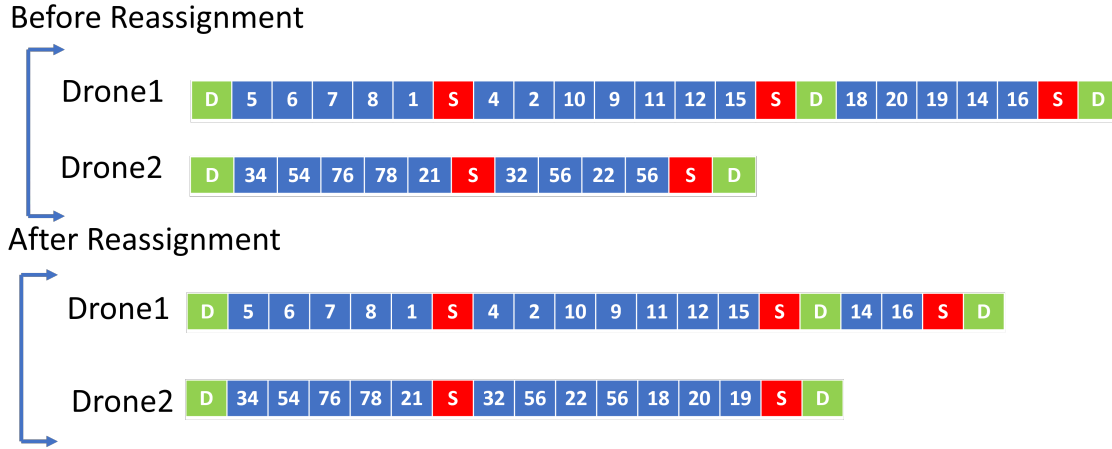


Figure 5.3: The Dynamic Task Reassignment Approach.

original RL parameters in Chapter 4, Section 4.4, in Table 4.1 and Table 5.1. Henceforth, the same parameters are used for all three models used in this experiment; i.e., the before and after reassignment of multiple drones and the single drone model. The values of these parameters can be adapted based on the types of drones that can be utilized in the actual system that is eligible for this research. The comparison is based on the population size specified datasets, which are WCL numbers as 100, 500, and 1000 WCLs, respectively. The capacity tolerance parameter for each WCL is set as a maximum capacity of 500, weighted in kilograms. The speed of the drone is set to 60 km / h, the duration of flight is the time consumed during the flight which is measured in minutes, the power consumed per second is 30 W/s, the nominal power consumption is set to 100 W, then the efficiency factor is set to 0.9 indicating the efficiency of the system being used, and all these parameters are taken from the manufacturer's data or factory specifications (Girlevicius, 2022). In addition, the drone capacity is 200K and the drone power is 500W (Seidu et al., 2024).

The main model consists of integrations of GA and RL, with fixed parameters carefully selected based on trial and error and other previous studies. The K-means and A* clustering algorithms are further used to effi-

Algorithm 7 Reassignment Technique

```
1: Start
2: Initialize metrics and working copies for each drone
3: working_routes[d] = {route: copy, remaining_capacity: full,
   energy_used: 0, distance: 0} for all drones
4: total_distance, total_waste, total_energy, total_flight_time = 0
5: route_count = len(drones)
6: # Compute initial load and update remaining capacity drone  $\in$  working_routes
7: total_waste = sum(waste for waste locations)
8: drone['remaining_capacity'] = drone_capacity - total_waste
9: # Reassignment loop
10: changed = True
11: while changed do
12:   changed = False
13:   # Identify completed and active drones
14:   completed = drones with no route or full load
15:   active = drones with route and free capacity
16:   if active not empty then
17:     busiest = active drone with most waste points
18:     # Reassign tasks to completed drones
19:     for loc  $\in$  busiest.route do
20:       if loc is waste location then
21:         waste = get waste at loc
22:         for receiver  $\in$  completed do
23:           if receiver has capacity  $\geq$  waste then
24:             transfer loc, update capacities, set changed = True, break
25:           # Update drones with optimized routes and compute metrics
26:           for did  $\in$  drones do
27:             drones[did]['route'] = working_routes[did]['route']
28:             # Calculate distance to each waste location and back to depot
29:             # Calculate accumulated_waste  $\geq$  drone_capacity, reset or reroute
30:             # Compute energy = distance  $\times$  energy_consumption_per_km
31:             # Estimate flight time, update total metrics accordingly
32: Return Optimal path from start to goal
```

Table 5.1: Experiments Parameters for RL Phase.

Parameter	Value
Drones Number	5
epsilon	1.0
learning_rate	0.001
state_size	100, 500, 1000
action_size	100, 500, 1000
Exploration Rate	0.5%
Gamma_discount_factor	0.95
epsilon_decay	0.995
epsilon_min	0.01
max_steps	100,500,1000 locations=100
episodes	100,500,1000 locations=1000
num_runs	20
n_init	10
max_iter	300

ciently cluster the WCLs assigned to drones. Multiple drone missions are assigned and then reassigned. These assignment and reassignment processes, based on dynamic techniques, are used to compare the results before and after reassignment, and hence the distribution of WCLs based on the proposed methods. In addition, a third experiment has been conducted with the same model, which uses GA and RL algorithms for a single drone. This third experiment is used as a benchmark reference to be compared with the other experiments. Hence, the proposed system consists of several experiments using hybrid algorithms that integrate GA and RL within multiple UAVs to generate synthetic data that creates the best path plans for the UAVs in the experimental models while addressing predefined constraints. This work also proposes adding two more methods, K-means and A* search algorithms, for dynamic assignment and reassignment of tasks among multiple drones during the two-phase process of the hybrid GARL model. Fur-

thermore, the study compares the methods' application results before and after task reassignment among multiple drones and, thence, also compares both models with a benchmark model using a single drone to compare the efficiency of the techniques between all models. This benchmarking helps in detecting incremental efficiency in regard to two measures: the number of drones (from single to multiple) and the addition or absence of dynamic tasks assignment and reassignment techniques. Three criteria are used to evaluate the performance of the models by recording and comparing significant improvements in flight time, energy consumption, and total distance. To conclude, the proposed model for multiple drones with the reassignment dynamic techniques recorded better results compared to the proposed model with multiple drones without using the reassignment dynamic techniques; and both methods outperformed the benchmark model using a single drone with regard to the flight time, energy consumed, and total route distance.

Table 5.1 shows the operational parameters for RL. The parameters change based on the number of data, such as the number of WCLs and the number of episodes; therefore, the maximum number of steps increases as the data increases. Whereas when the episodes are fixed at 1000 episodes, the maximum number of steps is 100 for all WCLs. Furthermore, the other parameters are fixed, with the epsilon set to 0.1, the probability of a random action exploring better possibilities for improved actions through initial exploration is 100%. This rate decreases with training until the agent learns its optimal policy. The epsilon decay is set to 0.995, which reduces by 0.5%, which is multiplied by this value after each iteration, thus effectively reducing the exploration over time. The epsilon minimum parameter is set to 0.01, which means that the exploration rate will not be less than 1%. The k-means parameters are: num runs = 20, the number of runs of the k-means algorithm with different centroids; n init = 10, which indicates the number of runs of the k-means algorithm with different initial centroids; max iter

= 300, the maximum number of iterations of the k-means algorithm. The learning rate is set as 0.1, and its control weight is based on updating new information and comparing it with existing data. The higher the value, the faster the exploration will be; while the lowering of the Q values will lead to gradual convergence. The discount factor or gamma is set at 0.5%, considering the important influence of future rewards in decision making, where a value of 0 takes only the current reward, while a value of 1 takes into account the acquisition of future rewards. The maximum step is the maximum number of actions each episode can take, preventing the agent from falling into an infinite loop. Episodes are the vital parameter that allows an agent to interact with its environment to improve its policy through learning. State size is the number of all possible states of the environment. The action size is the number of possible actions the agent can take in the environment. The experiments include three comparisons of two models with task cluster algorithms between multiple drones before and after reassignment to address dynamic task distribution and achieve problem objectives, and with a single drone model. These experiments seek to find optimal cost values based on the shortest distance and flight time, as well as the least energy consumption.

Table 5.2: The Performance Metrics.

Population waste locations	100			500			1000		
Number of Run	10 times			30 times			50 times		
Type of model	Before Reassigning	After Reassigning	GARL	Before Reassigning	After Reassigning	GARL	Before Reassigning	After Reassigning	GARL
Best Flight Time (in Min)	22.2	15.24	39.97	84.13	59.58	199.18	167.1	110.36	374.2
Energy Consumed (in Wh)	3798.82	1591.5	71946	8051.02	3133.09	158488	24969.9	9036.87	673560
Total Distance (in Km)	22.16	14.44	40	84.12	59.60	199.1	167.01	110.28	374.2

Table 5.2 shows the performance metrics, including the best flight time,

energy consumption, and total distance of the model at different numbers of WCL in the three experiments. The experiments show that the dynamic task performance is better after reassignment than before reassignment in terms of flight time, both of which in turn outperform the single-drone approach. The flight time for the 100 WCLs before reassignment is 22.2 minutes; after reassignment, it is 15.24 minutes, and for the single-drone approach it is 39.97 minutes. The flight time for 500 WCLs is 84.13 minutes before reassignment, 59.58 minutes after reassignment, and 199.18 minutes for the single-drone approach. The flight time for the 1000 WCLs before reassignment is 167.1 minutes; after reassignment is 110.36 minutes; and it is 374.2 minutes for the single-drone approach. Secondly, the energy consumption for the 100 WCLs is recorded better after the reassignment than before the reassignment; as it is 3798.82 Wh before the reassignment, it becomes 1591.5 Wh after the reassignment; and for the single-drone approach, it is 71946 Wh. At 500 WCLs, the energy consumption before reassignment is 8051.02 Wh; after reassignment it is 3133.09 Wh; and for the single drone, it is 158488 Wh. At 1000 WCLs, it is 24552.94 Wh before the reassignment, 9036.87 Wh after the reassignment, and for the single drone, it is 673560 Wh. As for the third criteria, which is the total distance of each tour, it is recorded better after the reassignment than before the reassignment. For the 100 WCLs, it is 22.16 km before the reassignment and reduces to 14.44 km after the reassignment and is the longest for the single-drone approach, which is 40 km. Similarly, for the 500 WCLs, it records 84.12 km before the reassignment and reduces to 59.60 km after the reassignment and is the longest for the single-drone approach, which is 199.1 km. And for the 1000 WCLs, it records 167.01 km before the reassignment, 110.28 km after the reassignment, and 374.2 km for the single drone. Additionally, all experiment calculations are based on the average runs, which for 100 WCLs takes an average of 10 runs; then for 500 WCLs, which takes an average of 30

runs; and finally for 1,000 WCLs, which takes an average of 50 runs.

Table 5.3 shows the percentage improvement when comparing the proposed approach before and after dynamic reassignment techniques based on the average of different runs for various population sizes, i.e., 100, 500, and 1000 WCLs. It is observed that the improvement increases across the three stages of WCLs numbers with the increment of their numbers using both the proposed approach and the reassignment techniques, and even achieves better performance when using K-means and A* algorithms to cluster WCLs across all five UAVs. The improvements shown at 100 WCL before and after reassignment are 30.77% in flight time, 58.1% in energy consumption, and 30.89% in total distance. Furthermore, at 500 WCLs, the improvement in flight time is 31.61%, in energy consumption is 61.08%, and in total distance is 31.64%. Finally, at 1000 WCLs, the improvements gained are 33.96% in flight time, 70.18% in energy consumption, and 30.89% in total distance. Above and beyond, the more complex the problem, the better the performance efficiency is demonstrated. As shown in the model, a positive relationship is observed where the performance efficiency increases as the number of WCLs increases. Figure 5.4, Figure 5.5, and Figure 5.6 show improvements in the three main parameters: flight distance, energy consumed, and total distance, illustrating improvements in the efficiency of the proposed algorithms when the WCL increased. Figure 5.4 shows the improvements from before to after dynamic reassignment techniques. Figure 5.5 shows the improvements in the results before reassignment compared to the single-drone benchmark model. Finally, Figure 5.6 shows the improvements in the results after dynamic reassignment techniques compared to the benchmark single drone model.

Table 5.3: The Experimental Results Showing Percentage Improvements Between Before and After the Dynamic Reassignment Techniques.

Population Waste Locations	100	500	1000
Number of Runs	10 times	30 times	50 times
Flight Time	30.77%	31.61%	33.96%
Energy Consumed	58.1%	61.08%	70.18%
Total Distance	30.89%	31.64%	30.89%

Table 5.4 shows the percentage of improvement when comparing the proposed approaches before and after dynamic reassignment techniques in multiple UAVs versus a single-drone model. The improvements are observed across the three WCL number sets using both the proposed approach and reassignment techniques, and even better performance is achieved when using the K-means and A* algorithms to cluster WCLs amongst all drones. The improvements in 100 WCLs before reassignment are 44.46%, 94.72%, and 44.6%; and after reassignment are 61.87%, 97.79%, and 63.9%, compared to the single drone in flight time, energy consumption, and total distance, respectively.

Furthermore, at 500 WCLs, the improvements are 57.76%, 94.92%, and 57.75% before reassignment; and 70.09%, 98.02%, and 70.07% compared to the single drone in flight time, energy consumption, and total distance, respectively. Finally, in 1,000 WCLs, the improvement in flight time is 55.34% before reassignment and 70.51% after reassignment; and in energy consumption, it is 96.29% before reassignment and 98.66% after reassignment; and last, for total distance, the improvement is 55.37% before reassignment and 70.53% after reassignment.

This demonstrates that the proposed approach can be used effectively utilizing multiple UAVs for WCS by integrating GA with RL along with K-means and A* algorithms, which have been proven to efficiently optimizing individually in previous researches; however, the novelty of our approach

comes from the integration of all of these algorithms in it and proving its highest performance achievement.

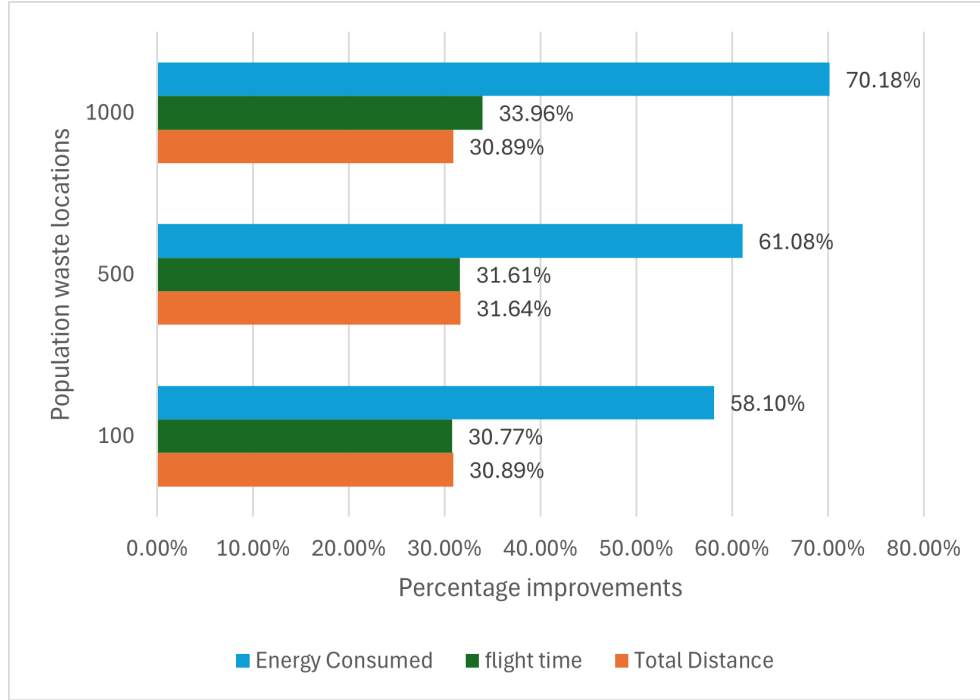


Figure 5.4: The Improvements from Before to After the Dynamic Reassignment Techniques.

GA can improve the solution of complex problems with multiple constraints, and the proposed model uses a highly efficient multi-objective fitness function that supports dynamic environments with various scenarios. Furthermore, enabling its integration with RL algorithms can expand real-time decision-making and better solve complex problems based on the learning ability of the algorithms to find optimal policies and optimize autonomous and dynamic environments. Additionally, the K-means algorithm can be used to obtain better clustering among multiple UAVs based on nearest-location partitioning, which can reduce the route distances and hence the cost of the process. Moreover, it can be useful in real-time applications based on iterative clustering and enforcing balanced functioning among all UAVs.

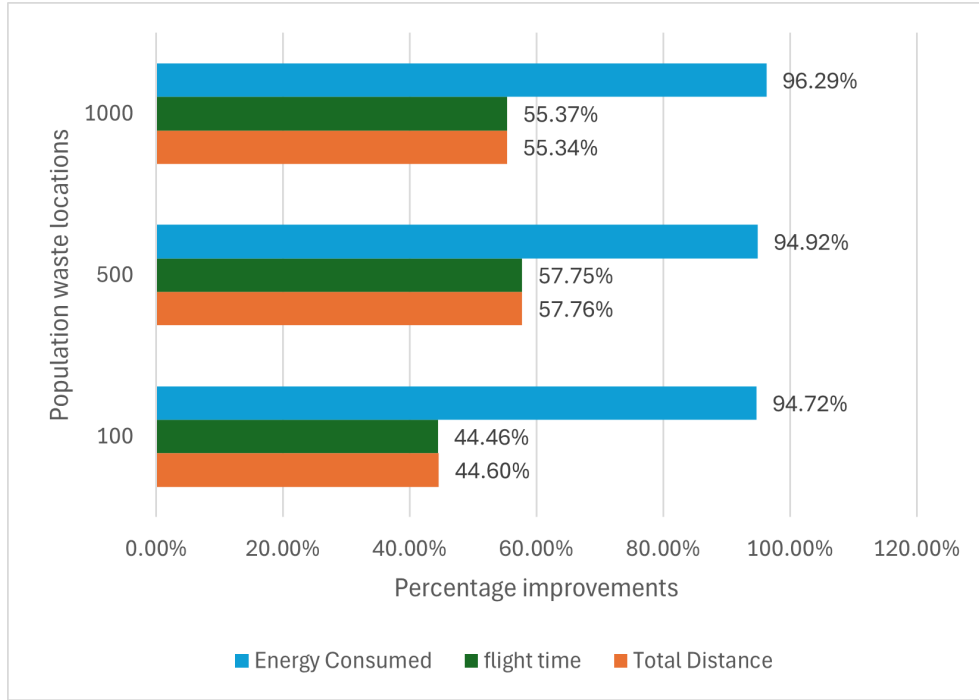


Figure 5.5: The Improvements of Results Before Reassignment Compared to the Benchmark Single-Drone Model.

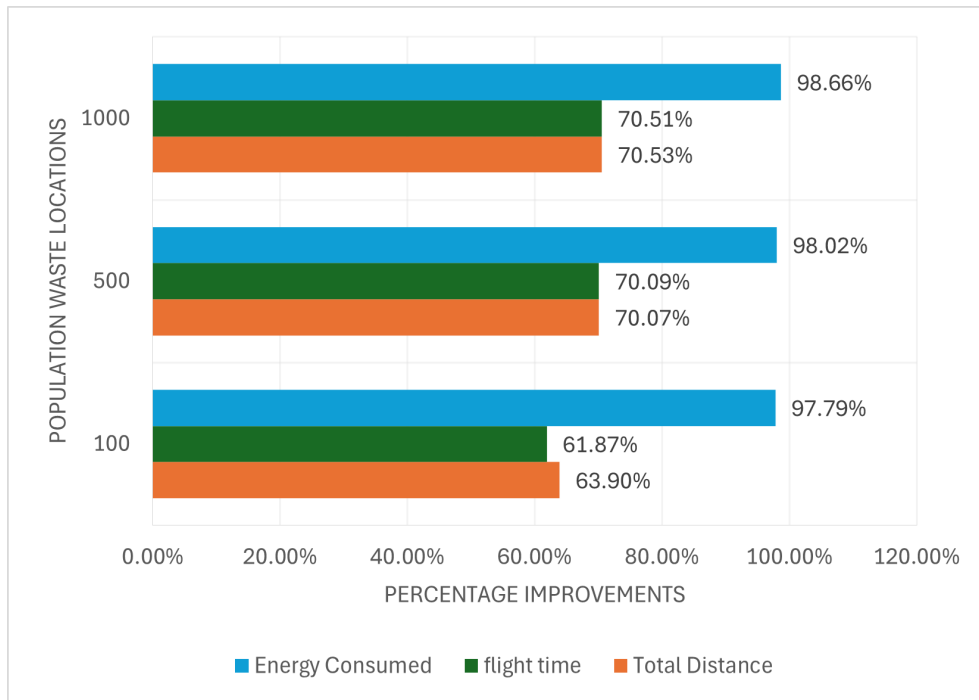


Figure 5.6: The Improvements of Results After Dynamic Reassignment Techniques Compared to the Benchmark Single Drone Model.

Thereafter, to enhance finding the best route after clustering, A* is used internally on each route to find the shortest path and further enhance the search process to find the best path, which can be used in dynamic environments. To assess how well the proposed method using K-Means clustering and A* search can scale, both theoretical and empirical analyses are performed. The K-Means clustering algorithm has a computational complexity of $O(Nku)$, where N represents the number of waste locations, k denotes the number of clusters (drones), which is set at 5 based on the number of drones utilized in the model, and u indicates the number of iterations required to reach convergence. The number of drones, denoted by k , and the number of iterations u , are relatively small and kept constant, which simplifies the scaling of the algorithm; therefore, the growth is based on the number of waste locations. The A* search algorithm, while recognized for its exponential worst-case complexity, is advantageous due to prior clustering. By minimizing the search space for each drone and utilizing an informed heuristic function that incorporates energy costs and capacity constraints, A* maintains efficient practical performance even with a growing number of locations. Empirical results demonstrate that the system effectively scales with larger problem sizes when compared to alternative models. The Table 5.3 and Table 5.4 outline the computation time, energy consumption, and efficiency improvements as the quantity of waste locations rises. Also, the proposed concept of reassignment relies on balancing missions amongst all UAVs, where when any UAV completes its missions, it will be reassigned new missions from other UAVs that did not complete their missions. Hence, new missions can be reassigned from other UAVs with more missions and more WCLs to be serviced, potentially giving the mission-free UAVs half of their missions. If the WCLs are equally divided, the other UAVs are given their missions equally, or if asymmetrical, it can be divided again based on the nearest locations of their clusters. These proposed techniques can en-

able WCS development and demonstrate high reliability and efficiency with validity and consistency in achieving better optimizations. It reduces costs related to energy consumption and decreases the flight time and total distance required, hence achieving the objectives of the proposed approach in real-time and dynamic environments. Many physical factors, such as climate change, wind, and takeoff and landing times, are not considered in this study.

Table 5.4: Comparing the Three Experiment Results.

Population waste locations	100		500		1000	
Number of Run	10 times		30 times		50 times	
The comparison methods	Before Reassigning	After Reassigning	Before Reassigning	After Reassigning	Before Reassigning	After Reassigning
Best Flight Time (in Min)	44.46%	61.87%	57.76%	70.09%	55.34%	70.51%
Energy Consumed (in Wh)	94.72%	97.79%	94.92%	98.02%	96.29%	98.66%
Total Distance (in Km)	44.6%	63.9%	57.75%	70.07%	55.37%	70.53%

5.5 Summary

The WCS process is a vital part of the WMS, which is a vigorous sector for a thriving lifestyle and a prosperous economy, especially in smart cities. This thesis proposes a hybrid integrative approach that integrates different algorithms multi-objectively, namely GAs, RLs with K-means, and A* search algorithms to cluster and ensure an efficiently balanced distribution of WCLs assignments among multiple drones. This approach aims at optimizing the WCS process by finding the best solution for path problems

by suitable tours that minimize costs while addressing constraints multi-objectively. The proposed system consists of several experiments using hybrid algorithms that integrate GA and RL within multiple UAVs to generate synthetic data that creates best path plans for the UAVs in the experimental models while addressing pre-defined constraints. This work also proposes adding two more methods, K-means and A* search algorithms, for dynamic assignment and reassignment of tasks among multiple drones during the two-phase process of the hybrid GARL model. Furthermore, the study compares the methods' application results before and after task reassignment among multiple drones and, thence, also compares both models with a benchmark model using a single drone to compare the efficiency of the techniques across all models. This benchmarking helps detect incremental efficiency regarding two measures: the number of drones (from single to multiple) and the addition or absence of dynamic task assignment and reassignment techniques. Three criteria are used to evaluate the performance of the models by recording and comparing significant improvements in flight time, energy consumption, and total distance. The results show that the dynamic tasks reassignment technique model demonstrates better performance than the tasks pre-reassignment model. However, both models achieve superior performance to the single-drone model, i.e. the dynamic reassignment technique model outperforms the other two models, followed by the pre-reassignment mission model, and the single-drone model comes last. The results of the first comparison between before and after the reassignment models show that the dynamic reassignment technique model is superior in terms of improving flight time by 32.11%, reducing energy consumption by 63.12%, and reducing total distance by 31.14%. Further experiments are comparing the two proposed models before and after the reassignment mission in multiple UAVs with the third model, which is the single-drone model. The improvement results recorded of the dynamic mis-

sion reassignment model from the single-drone model are demonstrated by: reducing flight time by 67.49%, reducing energy consumption by 98.16%, and reducing total distance by 68.17%. Improvements recorded of the pre-reassignment model from the single-drone model are demonstrated by: reducing flight time by 52.52%, energy consumption by 95.31%, and total distance by 52.57%. The results demonstrate the superiority of the dynamic task reset, showing significant improvements in all criteria. Thus, this hybrid integrative model utilizes the best features of both the GA and RL algorithms, along with K-means and A* search algorithms, to synergically optimize UAV performance through their applications in dynamic environments. Hence, the proposed model approach achieves a reliable and high-performance system for multiple drones under multiple constraints and complex environments.

Chapter 6

Conclusion and Recommendations

The tremendous demand and prospects for further growth in drone-based WCS are attracting increasing interest, hence necessitating expanded research and development to implement robust models applicable to real-world systems. The use of ML and optimization approaches contributes to the production of effective problem solutions for WCS, which is a vital part of WMS. Therefore, in this thesis, we present an effort consisting of multiple studies and experiments to contribute to the field of research in this rapidly growing area. The experimentation process in these studies depends on the gradual evolution in complexity to gradually substantiate the application efficiency of the algorithms used and prove their high performance while being tested on different measures. The experimentation gradual increasing complexity employs performance efficiency comparisons in different measures; i.e. from single algorithm to hybrid integrative algorithms; from lower to higher number of WCLs; comparing different sequence of algorithms employment in the hybrid models in their respective phases, e.g. GARL or RLGA; and then exploring the UAVs performance efficiency regarding the increase in number from single to multiple. Hence, this thesis first presents an approach that uses a single algorithm in a single UAV, then proceeds with multiple studies and experiments in a logical sequence to experiment with more and more complex approaches with complex data.

In Chapter 3, we conduct a study that proposes a multi-objective GA model to solve the routing problem in the WMS with a single UAV. It aims to address the challenges and constraints inherent in WCS with complex scenarios through satisfying multiple objectives with highly efficient solutions. The proposed GA model enhances certain operations, starting from the initial random population by incorporating constraints derived from the drone's capacity and energy limits. Additionally, the model includes a multiobjective fitness function to consider various constraints such as finding the optimal route based on the shortest distance, the maximum waste load capacity, the least time, and the least energy consumption. The multiobjective fitness function, which is the most complex part of the model, integrates several equations simultaneously to determine the best fitness for these different constraints. The genetic operations employed in the model include selection tournament, order crossover, and combined exchange mutation, along with a local search space approach in the mutation process. Incorporating multiple operations in the mutation process enhances exploration, preventing convergence to local minima, and facilitating the discovery of superior solutions. The model is applied to a single drone utilizing synthetic data that generates data of 1000 WCLs. The experiment is divided into stages, including 100, 500, and 1000 WCLs. Furthermore, execution time increases with the escalation in the number of data points. The results of the model are compared with the results of two comparative studies incorporating a similar dataset and constraints with different approaches. The first study employed a hybrid ACO, while the second utilized a novel two-phase approach, MWMTT. Each comparative study employed similar datasets, consisting of three sets that are compared across all three experiments in our study. The results demonstrate that the proposed GA model outperforms both comparative study models, reducing the average error by 61.52% and 62.27% than those in the models of the first and second stud-

ies, respectively. The model employs a minimum optimization strategy by halting optimization after a certain number of iterations and setting a convergence threshold value. The GA exhibits significant capability in finding the best solution while satisfying multiple constraints and efficiently addressing routing problems. Overall, it provides a robust framework for identifying optimal or near-optimal solutions applicable in real-world scenarios. Hence, this study answers research question 1 (RQ1).

In chapter 4, we conduct another study that focuses on a multi-objective hybrid approach. The approach presents four models utilizing GA and RL to solve the routing problem using a single UAV. It aims to enhance WCS by solving the routing problems in complex scenarios. The model proposes several hybrid and individual methods of GA and RL algorithms. The models are applied using synthetic data, generating 1000 datasets of WCLs. The experiments are divided into stages, including 100, 500, and 1000 waste locations. The hyperparameters of the experiments are stabilized for all experiments and are selected based on a literature review and trial and error process. The first model approach is GARL, which is constructed as a two-phase model; the first phase starts with the GA, which leverages an initial random population based on a well-constructed fitness function addressing various constraints such as the UAV's capacity and energy by servicing all the WCLs in the shortest tour distance and time. The second phase of the model employs an RL algorithm that takes the best GA population outcome as the initial state to generate the Q matrix in the learning phase. The second proposed model approach is RLGA, which reverses the sequence of algorithms through employing the RL algorithm in the first phase and importing the best tour outputs from RL as the initial population in GA in the second phase instead of using a random one. Furthermore, the results of the two models are used to demonstrate the superiority of the hybrid approaches over the remaining two model approaches, which are the individual algo-

rithms of GA and RL, through comparison of the four approaches results. Hence, the results of individual algorithms of GA and RL are used as benchmarks to rank the superiority of the two hybrid approaches. The results are based on fitness values that take into account the trade-off between several constraints, such as capacity, energy, flight time, distance, and servicing all WCLs. Flight time is calculated as another efficiency matrix measure of the proposed methods. The GARL model gained the highest improvements over all other approaches in regard to fitness value and flight time. When compared to GA alone, its results record higher improvements by 11.05% and 2.27 minutes in 100 WCLs, 22.06% and 4.32 minutes in 500 WCLs, and 26.32% and 8.1 minutes in 1000 WCLs. In addition, its comparison with RL alone results in records higher improvements by 12.68% and 2.51 minutes in 100 WCLs, 22.55% and 5.62 minutes in 500 WCLs, and 26.42% and 10.7 minutes in 1000 WCLs. Then the RLGA comes to be the second-best performance approach after GARL. The proposed hybrid approaches show greater potential to solve routing problems over individual algorithms by leveraging the control optimization power and synergy of both the GA and RL together with added higher performance weight to the sequence of algorithms employment in the hybrid approach; i.e., employing GA in the first phase, then RL in the second phase. Overall, this hybrid approach provides an effective framework for identifying optimal or near-optimal solutions applicable to real-world scenarios. Therefore, this study answers research question 2 (RQ2). Hence, the logical sequence of the next studies includes studying applying an efficient method similar to the best approach conducted in this study based on the required constraints in multiple UAVs similar to the one used in this model by distributing roles among them to measure the efficiency of a higher number of drones using the same approach or even a more complex one.

Therefore, in chapter 5, we conduct another study that focuses on an ap-

proach that constructs multi-objective hybrid approach that integrates different algorithms as GA with RL algorithm, and further utilizes K-means algorithm with A* search algorithm to cluster tasks among multiple drones and hence efficiently solve the routing problem in WCS. The dynamic reassignment method is used to further enhance the efficiency of the WCS model and further reduce costs, aiming to develop WCS through solving NP-problem sort of routing problems in complex scenarios. The models' approach is composed of a two-phase process. The first phase is the GA process phase, which utilizes an initial random population and a well-constructed fitness function based on various constraints such as energy consumption, flight time, and visiting all WCLs in the shortest tour distance. The second phase of the model relies on an RL algorithm process, which can expand real-time decision-making and better solve complex problems based on the learning ability of the algorithm to find optimal policies and optimize autonomous and dynamic environments. K-means with A* algorithms are additionally used in both phases of the model for dynamic assignment and reassignment of tasks to distribute tasks among a given number of drones set to five, during the two-phases process of the hybrid GARL approach. The study compares the two models before and after reassignment techniques in multiple UAVs and thence compares each of the two with another benchmark model of a single drone regarding three criteria, which are flight time, energy consumption, and total tour distance. The single drone benchmarking is used to compare the techniques' efficiency across all models and helps detect incremental efficiency regarding two measures: the number of drones (from single to multiple) and the addition or absence of dynamic tasks assignment and reassignment techniques. This approach is applied to multiple drones using synthetic data, generating 1,000 data for WCLs. Each of the three experiments are divided into three stages, including 100, 500, and 1,000 WCLs. The results of the first comparison between before and after the re-

assignment models show that the dynamic reassignment technique model is superior in terms of improving flight time by 32.11%, reducing energy consumption by 63.12%, and reducing total distance by 31.14%. Further experiments are being conducted to compare the two proposed models before and after the reassignment mission in multiple UAVs with the third model, which is the single-drone model. The recorded improvement results of the dynamic mission reassignment model from the single-drone model are demonstrated by: reducing flight time by 67.49%, reducing energy consumption by 98.16%, and reducing total distance by 68.17%. Recorded Improvements of the pre-reassignment model from the single-drone model are demonstrated by: reducing flight time by 52.52%, energy consumption by 95.31%, and total distance by 52.57%. Therefore, this study answers research question 3 (RQ3) by showing the effect of the proposed hybrid approaches using four algorithms: GA, RL, K-means, and A* to solve WCS cost problems using multiple drones. It also answers research question 4 (RQ4) since the experimental results show that the model using dynamic reassignment techniques outperforms the other two models. Additionally, the multiple drones' two models, based on balancing tasks across multiple drones to efficiently accomplish objective, are achieving significantly better results than the single drone model. The model approach also demonstrates a positive relationship where improvements in performance efficiency in regard to all criteria are observed as the data increases, specifically an increased number of WCLs, which indicates increased data complexity. Hence, the highly significant improvements demonstrated in the study results signify the optimal or near-optimal solutions that can be identified for WCS problems, especially in complex scenarios. Overall, the proposed approach provides a good framework for identifying optimal or near-optimal solutions applicable to real-world and dynamic complex environmental scenarios. Future work will include studying the UAV routing problem while taking into ac-

count other constraints such as weather effects, UAV landing and takeoff time, and obstacle avoidance using an efficient approach based on the existing constraints.

References

- Abcouwer, N., Daftry, S., del Sesto, T., Toupet, O., Ono, M., Venkatraman, S., ... Yue, Y. (2021). Machine learning based path planning for improved rover navigation. In *2021 ieee aerospace conference (50100)* (pp. 1–9).
- Abdallah, M., Arab, M., Shabib, A., El-Sherbiny, R., & El-Sheltawy, S. (2020). Characterization and sustainable management strategies of municipal solid waste in egypt. *Clean Technologies and Environmental Policy*, 22, 1371–1383.
- Abdulsattar, K., Harrath, Y., & Kaabi, J. (2023). Ant colony algorithm to solve a drone routing problem for hazardous waste collection. *Arab Journal of Basic and Applied Sciences*, 30(1), 636–649.
- Aktemur, I., Erensoy, K., & Kocyigit, E. (2020). Optimization of waste collection in smart cities with the use of evolutionary algorithms. In *2020 international congress on human-computer interaction, optimization and robotic applications (hora)* (pp. 1–8).
- Alcántara Marín, A., Capitán Fernández, J., Cunha, R., & Ollero Baturone, A. (2021). Optimal trajectory planning for cinematography with multiple unmanned aerial vehicles. *Robotics and Autonomous Systems*, 140, Article number 103778..

- Alizadeh, M., Wai, H.-T., Scaglione, A., Goldsmith, A., Fan, Y. Y., & Javidi, T. (2014). Optimized path planning for electric vehicle routing and charging. In *2014 52nd annual allerton conference on communication, control, and computing (allerton)* (pp. 25–32).
- Alwabli, A., & Kostanic, I. (2020). Dynamic route optimization for waste collection using genetic algorithm. In *2020 international conference on computer science and its application in agriculture (icosica)* (pp. 1–7).
- Andrew, A. M. (1998). Reinforcement learning:: An introduction. *Kybernetes*, 27(9), 1093–1096.
- Anghinolfi, D., Paolucci, M., & Tonelli, F. (2016). A vehicle routing problem with time windows approach for planning service operations in gas distribution network of a metropolitan area. *IFAC-PapersOnLine*, 49(12), 1365–1370.
- Asim, M., Mashwani, W. K., Belhaouari, S. B., & Hassan, S. (2021). A novel genetic trajectory planning algorithm with variable population size for multi-uav-assisted mobile edge computing system. *IEEE Access*, 9, 125569–125579.
- Assaf, R., & Saleh, Y. (2017). Vehicle-routing optimization for municipal solid waste collection using genetic algorithm: The case of southern nablus city. *Civil and environmental engineering reports*, 26(3), 43–57.
- Baldacci, R., Toth, P., & Vigo, D. (2010). Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research*, 175, 213–245.
- Beigi, P., Rajabi, M. S., & Aghakhani, S. (2022). An overview of drone energy consumption factors and models. *Handbook of Smart Energy Systems*, 1–20.

- Bhargava, K., Gupta, R., Singhal, A., & Shrinivas, A. (2019). Genetic algorithm to optimize solid waste collection. In *Proceedings of the 3rd international conference of recent trends in environmental science and engineering (rtese'19)*.
- Bihun, R., & Lytvyn, V. (2022). Optimization of garbage removal within a territorial community. *Eastern-European Journal of Enterprise Technologies*, 1(3), 115.
- Chao, Y., Xiaojia, X., & Wang, C. (2020). Towards real-time path planning through deep reinforcement learning for a uav in dynamic environments. *Journal of Intelligent & Robotic Systems*, 98(2), 297–309.
- Chen, C., & Zhou, Y. (2021). Application of deep reinforcement learning algorithm in smart finance. In *Modern management based on big data ii and machine learning and intelligent systems iii* (pp. 40–48). IOS Press.
- Chen, Y., Dong, Q., Shang, X., Wu, Z., & Wang, J. (2022). Multi-uav autonomous path planning in reconnaissance missions considering incomplete information: A reinforcement learning method. *Drones*, 7(1), 10.
- Chen, Z., Lu, M., Zhou, Y., & Chen, C. (2021). Information synergy entropy based multi-feature information fusion for the operating condition identification in aluminium electrolysis. *Information Sciences*, 548, 275–294.
- Cui, Z., & Wang, Y. (2021). Uav path planning based on multi-layer reinforcement learning technique. *Ieee Access*, 9, 59486–59497.
- Demeyer, S., Goedgebeur, J., Audenaert, P., Pickavet, M., & Demeester, P. (2013). Speeding up martins' algorithm for multiple objective shortest path problems. *4or*, 11(4), 323–348.

- Dolinina, O., Pechenkin, V., Gubin, N., Aizups, J., & Kuzmin, A. (2019). Development of semi-adaptive waste collection vehicle routing algorithm for agglomeration and urban settlements. In *2019 IEEE 7th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)* (pp. 1–6).
- Erdelić, T., Carić, T., et al. (2019). A survey on the electric vehicle routing problem: variants and solution approaches. *Journal of Advanced Transportation*, 2019.
- Fang, B., Yu, J., Chen, Z., Osman, A. I., Farghali, M., Ihara, I., . . . Yap, P.-S. (2023). Artificial intelligence for waste management in smart cities: a review. *Environmental Chemistry Letters*, 21(4), 1959–1989.
- Farahbakhsh, A., & Kheirikhah, A. S. (2023). A new efficient genetic algorithm-taguchi-based approach for multi-period inventory routing problem. *International journal of research in industrial engineering*, 12(4), 397–413.
- Fathurrahman, H. I. K., Ma'arif, A., & Chin, L.-Y. (2021). The development of real-time mobile garbage detection using deep learning. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, 7(3), 472–478.
- Girlevicius, L. (2022). *Evaluation of solar powered systems in small scale uav designs*.
- Hamid, N., Dharmawan, W., & Nambo, H. (2023). Dynamic path planning for unmanned surface vehicles with a modified neuronal genetic algorithm. *Applied System Innovation*, 6(6), 109.
- Han, L., Gao, F., Zhou, B., & Shen, S. (2019). Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots. In

2019 ieee/rsj international conference on intelligent robots and systems (iros) (pp. 4423–4430).

Hejres, S., Mahjoub, A., & Hewahi, N. (2024). Routing approaches used for electrical vehicles navigation: A survey. *International Journal of Computing and Digital Systems*, 15(1), 801–819.

Holler, J., Vuorio, R., Qin, Z., Tang, X., Jiao, Y., Jin, T., ... Ye, J. (2019). Deep reinforcement learning for multi-driver vehicle dispatching and repositioning problem. In *2019 ieee international conference on data mining (icdm)* (pp. 1090–1095).

Hou, W., Luo, Q., Wu, X., Zhou, Y., & Si, G. (2021). Multiobjective optimization of large-scale evs charging path planning and charging pricing strategy for charging station. *Complexity*, 2021.

Hu, C., Meng, Z., Qu, G., Shin, H.-S., & Tsourdos, A. (2021). Distributed cooperative path planning for tracking ground moving target by multiple fixed-wing uavs via dmppc-gvd in urban environment. *International Journal of Control, Automation and Systems*, 19(2), 823–836.

Joe, W., & Lau, H. C. (2020). Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers. In *Proceedings of the international conference on automated planning and scheduling* (Vol. 30, pp. 394–402).

Juneja, M., & Nagar, S. (2016). Particle swarm optimization algorithm and its parameters: A review. In *2016 international conference on control, computing, communication and materials (iccccm)* (pp. 1–5).

Kaabi, J., Harrath, Y., Mahjoub, A., Hewahi, N., & Abdulsattar, K. (2022). A 2-phase approach for planning of hazardous waste collection using an unmanned aerial vehicle. *4OR*, 1–24.

- Kaabi, J., Harrath, Y., Mahjoub, A., Hewahi, N., & Abdulsattar, K. (2023). A 2-phase approach for planning of hazardous waste collection using an unmanned aerial vehicle. *4OR*, 21(4), 585–608.
- Khan, S. I., Qadir, Z., Munawar, H. S., Nayak, S. R., Budati, A. K., Verma, K. D., & Prakash, D. (2021). Uavs path planning architecture for effective medical emergency response in future networks. *Physical Communication*, 47, 101337.
- Li, W., Wang, L., & Fei, A. (2019). Minimizing packet expiration loss with path planning in uav-assisted data sensing. *IEEE Wireless Communications Letters*, 8(6), 1520–1523.
- Li, Y., Eslamiat, H., Wang, N., Zhao, Z., Sanyal, A. K., & Qiu, Q. (2019). Autonomous waypoints planning and trajectory generation for multi-rotor uavs. In *Proceedings of the workshop on design automation for cps and iot* (pp. 31–40).
- Lin, K., Zhao, R., Xu, Z., & Zhou, J. (2018). Efficient collaborative multi-agent deep reinforcement learning for large-scale fleet management. *arXiv preprint arXiv:1802.06444*.
- Lin, X., Yu, Y., & Sun, C. (2019). Supplementary reinforcement learning controller designed for quadrotor uavs. *IEEE Access*, 7, 26422–26431.
- Liu, Q., Wang, W., & Hua, X. (2021). Path planning method for electric vehicles based on freeway network. *Journal of Advanced Transportation*, 2021.
- Liu, R., Piplani, R., & Toro, C. (2022). Deep reinforcement learning for dynamic scheduling of a flexible job shop. *International Journal of Production Research*, 60(13), 4049–4069.

- Lu, Y., Gu, J., Xie, D., & Li, Y. (2019). Integrated route planning algorithm based on spot price and classified travel objectives for ev users. *IEEE Access*, 7, 122238–122250.
- Mazzia, V., Salvetti, F., Aghi, D., & Chiaberge, M. (2020). Deepway: a deep learning estimator for unmanned ground vehicle global path planning. *arXiv preprint*.
- Meniz, B., & Tiryaki, F. (2024). Genetic algorithm approach to asymmetric capacitated vehicle routing: A case study on bread distribution in istanbul, türkiye. *Decision Science Letters*, 13(3), 605–616.
- Mesquita, R., & Gaspar, P. D. (2021). A novel path planning optimization algorithm based on particle swarm optimization for uavs for bird monitoring and repelling. *Processes*, 10(1), 62.
- Mohammed, M. A., Ahmad, M. S., & Mostafa, S. A. (2012). Using genetic algorithm in implementing capacitated vehicle routing problem. In *2012 international conference on computer & information science (iccis)* (Vol. 1, pp. 257–262).
- Nguyen, D. D., Rohacs, J., & Rohacs, D. (2021). Autonomous flight trajectory control system for drones in smart city traffic management. *ISPRS International Journal of Geo-Information*, 10(5), 338.
- Özoğlu, B., Çakmak, E., & Tuğçe, K. (2019). Clarke & wright's savings algorithm and genetic algorithms based hybrid approach for flying sidekick traveling salesman problem. *Avrupa Bilim ve Teknoloji Dergisi*, 185–192.
- Pajares, A., Blasco, X., Herrero, J. M., & Reynoso-Meza, G. (2018). A multiobjective genetic algorithm for the localization of optimal and nearly

- optimal solutions which are potentially useful: nevmoga. *Complexity*, 2018(1), 1792420.
- Pan, Y., Yang, Y., & Li, W. (2021). A deep learning trained by genetic algorithm to improve the efficiency of path planning for data collection with multi-uav. *Ieee Access*, 9, 7994–8005.
- Platanitis, K. S., Kladis, G. P., & Tsourveloudis, N. C. (2020). Safe flyable and energy efficient uav missions via biologically inspired methods. In *2020 international conference on unmanned aircraft systems (icuas)* (pp. 1868–1877).
- Qie, H., Shi, D., Shen, T., Xu, X., Li, Y., & Wang, L. (2019). Joint optimization of multi-uav target assignment and path planning based on multi-agent reinforcement learning. *IEEE Access*, 7, 146264–146272.
- Quevedo, J., Abdelatti, M., Imani, F., & Sodhi, M. (2021). Using reinforcement learning for tuning genetic algorithms. In *Proceedings of the genetic and evolutionary computation conference companion* (pp. 1503–1507).
- Ramesh, G., & Praveen, J. (2021). Artificial intelligence (ai) framework for multi-modal learning and decision making towards autonomous and electric vehicles. In *E3s web of conferences* (Vol. 309).
- Reddy, A. A., Gangadhar, B., Muthukumar, B., & Mayan, J. A. (2019). Advanced garbage collection in smart cities using iot. In *Iop conference series: materials science and engineering* (Vol. 590, p. 012020).
- Ruan, Q., Zhang, Z., Miao, L., & Shen, H. (2013). A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, 40(6), 1579–1589.
- Sabet, S., & Farooq, B. (2022). Green vehicle routing problem: State of the art and future directions. *arXiv preprint arXiv:2202.01695*.

- Sajid, M., Singh, J., Haidri, R. A., Prasad, M., Varadarajan, V., Kotecha, K., & Garg, D. (2021). A novel algorithm for capacitated vehicle routing problem for smart cities. *Symmetry*, *13*(10), 1923.
- Sangeetha, V., Ravichandran, K., Shekhar, S., & Tapas, A. M. (2019). An intelligent gain-based ant colony optimisation method for path planning of unmanned ground vehicles. *Defence Science Journal*, *69*(2), 167–172.
- Sarvari, P. A., Ikhelef, I. A., Faye, S., & Khadraoui, D. (2020). A dynamic data-driven model for optimizing waste collection. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1958–1967).
- Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation science*, *48*(4), 500–520.
- Seidu, I., Olowu, B., & Olowu, S. (2024). Advancements in quadcopter development through additive manufacturing: A comprehensive review. *International Journal of Scientific Research in Science, Engineering and Technology*, *11*(4), 92–124.
- Shibl, M., Ismail, L., & Massoud, A. (2021). Electric vehicles charging management using machine learning considering fast charging and vehicle-to-grid operation. *Energies*, *14*(19), 6199.
- Shivgan, R., & Dong, Z. (2020). Energy-efficient drone coverage path planning using genetic algorithm. In *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)* (pp. 1–6).
- Sitek, P., & Wikarek, J. (2019). Capacitated vehicle routing problem with pick-up and alternative delivery (cvrppad): model and implementation using hybrid approach. *Annals of Operations Research*, *273*(1), 257–277.

- Song, J., Liu, L., Liu, Y., Xi, J., & Zhai, W. (2022). Path planning for multi-vehicle-assisted multi-uavs in mobile crowdsensing. *Wireless Communications and Mobile Computing*, 2022(1), 9778188.
- Sonmez, A., Kocyigit, E., & Kugu, E. (2015). Optimal path planning for uavs using genetic algorithm. In *2015 international conference on unmanned aircraft systems (icuas)* (pp. 50–55).
- Tang, X., Li, X., Yu, R., Wu, Y., Ye, J., Tang, F., & Chen, Q. (2023). Digital-twin-assisted task assignment in multi-uav systems: A deep reinforcement learning approach. *IEEE Internet of Things Journal*, 10(17), 15362–15375.
- Tarhan, F. A., & Ure, N. K. (2024). Genetic-algorithm-aided deep reinforcement learning for multi-agent drone delivery. *Drones*, 8(3), 71.
- Torabbeigi, M., Lim, G. J., Ahmadian, N., & Kim, S. J. (2021). An optimization approach to minimize the expected loss of demand considering drone failures in drone delivery scheduling. *Journal of Intelligent & Robotic Systems*, 102(1), 1–15.
- Wang, D., Ding, A., Chen, G., & Zhang, L. (2023). A combined genetic algorithm and a* search algorithm for the electric vehicle routing problem with time windows. *Advances in Production Engineering & Management*, 18(4).
- Wang, T., Zhang, B., Zhang, M., & Zhang, S. (2021). Multi-uav collaborative path planning method based on attention mechanism. *Mathematical Problems in Engineering*, 2021(1), 6964875.
- Wibisono, E., Martin, I., & Prayogo, D. N. (2021). Comparison of crossover operators in genetic algorithm for vehicle routing problems.

- Wu, C., Zhou, S., & Xiao, L. (2020). Dynamic path planning based on improved ant colony algorithm in traffic congestion. *IEEE Access*, 8, 180773–180783.
- Wu, Y., Wu, S., & Hu, X. (2021). Multi-constrained cooperative path planning of multiple drones for persistent surveillance in urban environments. *Complex & Intelligent Systems*, 7(3), 1633–1647.
- Xie, R., Meng, Z., Wang, L., Li, H., Wang, K., & Wu, Z. (2021). Unmanned aerial vehicle path planning algorithm based on deep reinforcement learning in large-scale and dynamic environments. *IEEE Access*, 9, 24884–24900.
- Xiong, T., Liu, F., Liu, H., Ge, J., Li, H., Ding, K., & Li, Q. (2023). Multi-drone optimal mission assignment and 3d path planning for disaster rescue. *Drones*, 7(6), 394.
- Yan, C., Xiang, X., & Wang, C. (2020). Towards real-time path planning through deep reinforcement learning for a uav in dynamic environments. *Journal of Intelligent & Robotic Systems*, 98, 297–309.
- Yao, Q., Zheng, Z., Qi, L., Yuan, H., Guo, X., Zhao, M., ... Yang, T. (2020). Path planning method with improved artificial potential field—a reinforcement learning perspective. *IEEE Access*, 8, 135513–135523.
- Yoo, S., Kim, H., & Lee, J. (2024). Combining reinforcement learning with genetic algorithm for many-to-many route optimization of autonomous vehicles. *IEEE Access*, 12, 26931–26942.
- You, C., Lu, J., Filev, D., & Tsotras, P. (2019). Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. *Robotics and Autonomous Systems*, 114, 1–18.

- Yuan, J., Liu, Z., Lian, Y., Chen, L., An, Q., Wang, L., & Ma, B. (2022). Global optimization of uav area coverage path planning based on good point set and genetic algorithm. *Aerospace*, 9(2), 86.
- Yue, L., Yang, R., Zhang, Y., Yu, L., & Wang, Z. (2022). Deep reinforcement learning for uav intelligent mission planning. *Complexity*, 2022.
- Yusuf, I., Baba, M. S., & Iksan, N. (2014). Applied genetic algorithm for solving rich vrp. *Applied Artificial Intelligence*, 28(10), 957–991.
- Zeng, W., Miwa, T., & Morikawa, T. (2017). Application of the support vector machine and heuristic k-shortest path algorithm to determine the most eco-friendly path with a travel time constraint. *Transportation Research Part D: Transport and Environment*, 57, 458–473.
- Zhang, W., Wang, Q., Li, J., & Xu, C. (2020). Dynamic fleet management with rewriting deep reinforcement learning. *IEEE Access*, 8, 143333–143341.
- Zhang, Z., Chen, Z., Zhang, J., Liu, Y., Chen, L., Yang, M., ... others (2024). Municipal solid waste management challenges in developing regions: A comprehensive review and future perspectives for asia and africa. *Science of the Total Environment*, 172794.
- Zhang, Z., Wu, L., Zhang, W., Peng, T., & Zheng, J. (2021). Energy-efficient path planning for a single-load automated guided vehicle in a manufacturing workshop. *Computers & Industrial Engineering*, 158, 107397.
- Zhao, Y., Zheng, Z., & Liu, Y. (2018). Survey on computational-intelligence-based uav path planning. *Knowledge-Based Systems*, 158, 54–64.

- Zhou, Z., Liu, Z., Su, H., & Zhang, L. (2021). Intelligent path planning strategy for electric vehicles combined with urban electrified transportation network and power grid. *IEEE Systems Journal*.
- Zhu, B., Bedeer, E., Nguyen, H. H., Barton, R., & Henry, J. (2021). Uav trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 70(9), 9540–9554.

ملخص الدراسة

في هذا العصر، يعتمد الطلب المتزايد على المركبات الكهربائية (EVs) على التكامل العالي لهذه التقنية في العديد من المجالات بالتالي، يثير مجال بحثي مهم الاهتمام بإيجاد الحل الأمثل لتخطيط مسار المركبات الكهربائية. من ناحية أخرى، يُعد نظام إدارة النفايات (WMS) قطاعًا اقتصاديًا مهمًا يساعد في دعم الصحة وتحسين جودة الحياة في المدن الذكية. ومن ثم، تتناقص هذه الأطروحة البحث في المركبات الكهربائية؛ وتحديدًا المركبات الجوية غير المأهولة (UAV)، وتقنيات التوجيه لمساعدة الباحثين على اختيار النهج المناسب لتطوير نظام قائم على تقنيات التحسين، والتعلم الآلي، أو تقنيات النهج الهجين، وخاصة نظام جمع النفايات (WCS). يمكن لتقنيات النهج الهجين التي تجمع بين التحسين/الخوارزمية الجينية (GA) وتقنيات التعلم الآلي/التعلم المعزز (RL) أن تساهم في إنتاج حلول فعالة للمشكلات لنظام جمع النفايات (WCS)، وهو جزء حيوي من نظام إدارة النفايات (WMS).

في هذه الدراسة، يتناول البحث مشكلة جمع النفايات باستخدام الطائرات بدون طيار (UAVs)، بهدف تحسين الأداء وخفض التكلفة واستهلاك الطاقة. حيث تم اقتراح ثلاثة مناهج مختلفة، ويركز المنهج الأوليان على طائرة بدون طيار واحدة، بينما يتضمن المنهج الثالث تنسيق طائرات بدون طيار متعددة. يستخدم المنهج الأول خوارزمية GA متعددة الأهداف. أما المنهج الثاني فيقترح دمج الخوارزميات GA مع RL في نموذج هجين. يجمع المنهج الثالث بين GA مع RL وخوارزمية k-means و A* لتنسيق طائرات بدون طيار متعددة. في جميع هذه المناهج المقترحة، تم أخذ العديد من القيود التشغيلية في الاعتبار، بما في ذلك سعة الطائرات بدون طيار ووقت الطيران وغيرها. حيث حقق نهج الخوارزمية GA متعددة الأهداف انخفاضًا كبيرًا في الخطأ مقارنةً بالبحوث السابقة: أكثر من 81% لـ 100 موقع للنفايات، وأكثر من 59% لـ 500 موقع للنفايات، وأكثر من 44% لـ 1000 موقع للنفايات. أما النهج الثاني، الذي يطبق GA مع RL، حقق أداء أعلى بنسبة 14.35% مقارنةً بالنهج المعتمد على GA. أما استراتيجية الطائرات بدون طيار المتعددة (النهج الثالث)، تم تطوّر نهجين: أحدهما مع إعادة توزيع مهام الطائرات بدون طيار والآخر بدونها. أظهرت النتائج أن إعادة توزيع مهام الطائرات بدون طيار حسن الأداء بشكل ملحوظ، حيث قلل زمن الرحلة بنسبة 32.113%، واستهلاك الطاقة بنسبة 63.120%، والمسافة المقطوعة الإجمالية بنسبة 31.140% مقارنةً بالسيناريو بدون إعادة توزيع المهام.

جامعة البحرين

كلية تقنية المعلومات



النهج الهجين القائم على تقنيات التعلم الآلي وطرق التحسين لإدارة جمع النفايات باستخدام المركبات الكهربائية

أطروحة مقدمة كجزء من متطلبات الحصول على
إعداد درجة الدكتوراة في علوم الحوسبة والمعلومات

شيماء إبراهيم هجرس

١٩٩٩٠١٩٠

إشراف

ا.د. نبيل الحويحي

(أستاذ)

دأمين محبوب

(أستاذ مشارك)

مملكة البحرين جامعة البحرين

يونيو ٢٠٢٥