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

Shaima Hejres, Amine Mahjoub and Nabil Hewahi

Abstract—— **The waste collection system (WCS) is part of the waste management system (WMS) and is a vital sector for a prosperous lifestyle and economy, especially in smart cities. This paper proposes hybrid integrative models of multi-objective Genetic Algorithms (GA) and Reinforcement Learning (RL) algorithms, aiming to optimize the waste collection process by finding the best solutions for routing problems with excellent tour plans in complex scenarios that minimize WCS cost while addressing its constraints in a multi-objective manner. The proposed system consists of several experiments utilizing hybrid algorithms that integrate GA and RL to generate synthetic data for experimental models, addressing specific predefined constraints such as drone capacity, energy consumption, flight time, flight distance, and service of all WCLs using a single Unmanned Aerial Vehicle (UAV). Each of the experiments is also divided into three parts, including 100, 500, and 1000 WCLs. The study first compares two hybrid algorithm models of GA and RL. The first proposed model starts its first phase by synthetic data generation with GA operations first 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 performances individually are compared with the results of the hybrid models' performances. The performance of hybrid models is better than that of individual models, while the GARL hybrid model is found to be the best approach among all models. It shows a notable improvement in fitness value and flight time compared to RLGA model, 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 only, 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 alone, 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 studies with similar data sets and constraints but different approaches, and have shown significant improvement. Hence, this integrative hybrid model synergizes the best features of both GA and RL algorithms and enhances applications in dynamic environments. The model achieves a reliable high-performance system under multiple constraints in complex environments.**

Index Terms—**Waste collection, waste management system, Genetic Algorithm (GA), Reinforcement algorithms (RL), Unmanned Aerial Vehicles (UAV), Path planning, multi-objective, hybrid model.**

Manuscript received February 21, 2025; revised August 27, 2025. This work was supported in part by the Department of Computer Science of Bahrain University, and special thanks to the University for providing the necessary resources and funding for this project.

S. Hejres is a Ph.D. candidate at the University of Bahrain, Zallaq, Bahrain (e-mail: 19990190@stu.uob.edu.bh).

A. Mahjoub is an Assistant Professor of Computer Science at the University of Bahrain, Zallaq, Bahrain (e-mail: amahjoub@uob.edu.bh).

N. Hewahi is a Professor of Computer Science, University of Bahrain, Zallaq, Bahrain (e-mail: nhewahi@uob.edu.bh).

I. Introduction

THE waste collection system (WCS) is one of the most promising sectors in the industry and one of the main systems that should be strengthened and improved. Many researchers have tried to enhance the WCS through tackling routing problems and providing solutions in this area. The WCS is part of the larger process of the waste management system (WMS). The WCS process comprises collecting, transporting, and controlling wastes in designated specific locations. Hence, WCS is suffering from high-cost operations that stimulate the researchers to investigate several methods of developing and enhancing the routing process to reduce the imperfections of traditional WCS. The traditional system is mainly based on inland waste collection vehicles and containers. Hence, researchers' investigations focus on methods that develop the routing process to manage and reduce high-cost transportation operations. However, the system may deal with many challenges and complex scenarios, including hazardous waste material that requires immediate attention and quick treatment. To address this issue, the integration of drone technology into the WCS becomes very helpful and necessary.

This method improves response time during emergencies and increases the efficiency of general waste management by ensuring public safety and environmental protection. The great demand to improve this sector comes from its contribution to a high quality of life in smart cities based on a healthy and clean environment. Hence, when a developed WCS is established, the effects reduce the rate of pollution and help decrease the negative impact of global warming by managing the huge amount of waste that needs to be controlled. This control process itself is comprised of several processes, such as recycling and safe waste incineration, especially biological or hazardous waste and non-decomposing materials. Industry revenue can be a challenge due to the high demands of the sector, which requires to improve immensely to increase profits.

This paper considers enhancing the WCS with an efficient model targeting multiple objectives through using a single drone visiting several waste locations to collect and dispose of waste at the main central landfill. The proposed model employs a hybrid approach that integrates two algorithms to provide a better performance than a single algorithm can achieve [1]. Optimization methods and machine learning (ML) techniques can make a very powerful combination that can solve complex problems such as routing problems [2].

The model's process has two phases, the first phase is employing a genetic algorithm (GA), which determines its best path and sets it as the initial state for operation by a machine learning algorithm, namely a Reinforcement learning algorithm (RL). Therefore, GA in the first phase can find the near-optimal path that minimizes the computational time and the cost, hence, handling complex and huge problems when the fitness function is well constructed.

RL, on the other hand, in phase two, can interact with the environment very appropriately through achieving an even better path based on managing a dynamic complex environment, and handling problems needing multi-objective solutions with more flexibility. Moreover, integrating GA with RL can leverage and synergize the supremacy of both methods [3].

The proposed model is designed to meet the multi-objective approach, addressing multiple constraints of WCS using a single drone. The drone starts its mission and ends it at the depot, hence collecting all wastes at all waste collection locations (WCLs) and then disposing of waste at the disposal location. The drone's energy consumption and waste container capacity constraints are first addressed as the UAV starts its tour with a full drone charge and an empty waste container to maximize its function in regard to its flying time and waste loading capacity. The drone returns to the depot when it is in a depleted condition, where it can recharge and continue the tour again. Hence, the drone needs to unload its waste when the waste capacity of the drone is almost full, or when the power is almost depleted and needs to be recharged. It should go to the disposal location to unload the waste and then continue its tour before going to the depot to recharge. Moreover, it must unload the waste at the disposal location before the tour ends. All WCLs must be visited to collect their waste, taking into account the shortest path distance, even if it passes through the same locations several times.

The rest of the framework of this paper is organized as follows. The second section analyzes GA and RL review papers in several studies that have applied these methods. The third section describes the research problem and explains the theory of the proposed solution model. The fourth section explains in detail the proposed model of the hybrid GA and RL algorithms as a solution to the problem. The fifth section shows the experimental results and analysis of the study. Finally, the last section presents the conclusion and future work directions.

II. Literature Review

Smart cities rely on intelligent systems, including AI that improves WCS, which are necessary to operate in any advanced automated civilization. Developing this aspect of the system influences various other aspects of this sector, such as the economy, health, and high quality of life. Therefore, the researchers investigate improving routing problems to increase the efficiency of WCS. The scope of this literature review is divided into two subsections, including several studies that examine individual and hybrid methods to address routing challenges.

A. Individual Methods in Routing Problem

Routing problems can be effectively solved using several methods. One such method is the reinforcement learning (RL) algorithm, which is well-compatible to dynamic environments, hence accommodating it to adjust and fine-tune its parameters during the training phase. This capability enhances the simple search tasks, particularly in the Deep Reinforcement Learning (DRL) algorithm, 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, which makes it adaptable for complex environments. In one study, DRL [4] is applied to address path planning problems for UAVs using a simulation model known as STAGE Scenario software. This method handles the situation as a survival mode, granting the avoidance of enemy attacks in both static and dynamic modes. The RL framework is employed exclusively with diverse algorithmic types, such as the Dueling Double Deep Q-Networks (D3QN) algorithm. In another study using RL-optimized quadcopter UAVs that consider specific types of UAVs through a supplementary controller [5]; they thus utilize thus utilizes the DRL algorithm to solve many problems, such as the Autonomous Motion Planning (AMP) problem. DRL is powerful in artificial fields for path planning through tackling static and dynamic situations, and overcoming complex environmental barriers by merging with the black hole potential field (BHPF) to enable agents' collaboration and obstacle avoidance [6]. Additionally, deep algorithms such as a double deep Q-network (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) [7] emerges with the Markov decision process and recurrent neural network, and Convolutional Neural Networks (CNNs) for spatial feature analysis. This method combines current rewards, states, and actions to decrease unnecessary exploration. To address the challenge of dynamic vehicle routing [8], a research study employs DRL, which includes neural network-based temporal difference learning along with the Markov decision process to solve customer routing issues. This method shows improvement in the value function and enabling better re-routing decisions involving the ability to switch and insert customer locations. The study also introduces DRLSA, based on simulated annealing, and compares its performance with 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. Moreover, a plethora of studies have indicated that the GA is one of the most effective algorithms for solving routing problems [1]; specifically in WMSs. In research conducted in the city of Mecca, Saudi Arabia, GA is applied in WMSs and shows its capability to improve dynamic routes. The results show that GA successfully reduces costs profoundly by addressing multiple objectives, such as consuming less 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 [9]. Similar research applying GA in solid waste management (SWM) systems [10] is conducted, taking into account factors such as vehicle capacity and time constraints to improve routing schedules. Furthermore, many WMSs leverage Internet of Things (IoT) tools within smart city systems to verify the best routes. A case study conducted in the Bakirkoy district of Istanbul examines GA's ability in path planning, revealing that it is superior to the traditional methods in optimizing routing solutions [11]. A novel GA [12] can solve the Inventory Routing Problem (IRP) to meet different customer requirements within a specified period based on incorporating a new chromosomal structure designed to represent solutions efficiently. Another case study conducted to solve the Asymmetric Capacity Vehicle Routing Problem (ACVRP) in Istanbul delivery operations that deliver bread daily using GA to improve delivery process efficiency shows positive outcomes [13].

B. Hybrid Methods in Routing Problem

Many studies highlight the benefits of combining different approaches to synergize the advantages of each approach. One such study combines machine learning and optimization algorithms to efficiently improve outcomes in a garbage management system [14]. This research applies a deep learning platform known as DenseNet121, operating it on 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 by 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 machine learning-based model via the optimization of hyperparameters, which are essential for model performance [15]. The study suggests a model that merges RL and GA algorithms. In this setup, the GA takes actions from RL to generate new generations; the 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 Capacitated Vehicle Routing Problem (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.[16] Compare two methods, the RL and the 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. [17] propose a novel vehicle routing problem for hazardous waste collection management. The addressed constraints 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 Minimum Time During Each Trip (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 between 41-48, and large group cases between 49-89. The performance is evaluated based on a narrow minimum threshold. The research results reveal 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 locations' trips. The study of Abdulsattar et al.[18] The 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 to 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 the 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 Ant Colony Optimization (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; and when it needs to be recharged, it goes to the waste disposal and then to the depot and thenceforth continues to unvisited WCLs. 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 is explored exhaustively, it could increase training time without efficient outcomes. On the other hand, if the exploitation focuses on current knowledge only, it could miss better solutions, since that of balancing exploration and exploitation could be a perfect solution. Furthermore, current methods often fail to adequately address the main challenges associated with WCSs. Considering these drawbacks, this proposed work aims to deliver a new valuable insight and proposes contributions by introducing a combination of optimization approaches and machine learning methodologies. This merging 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. This proposed study introduces a multi-objective model that addresses diverse constraints through the integrated application of GA and RL algorithms. The proposed integrated model aims to yield feasible solutions, which 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.

III. Problem Definition and Description

This paper investigates solving the problem of WCSs using a single UAV. The system is contingent on collecting waste from various WCLs, starting and ending at the same depot. The drone must visit all WCLs. When power runs low, it must go to the disposal location to unload its waste load and then return to the depot to recharge before continuing the tour. Likewise, if the UAV capacity load is nearly full, it should go to the disposal location to unload the waste load and then continue the tour. This approach aims to find the best route with the shortest distance via GA. The UAV capacity, energy endurance, and flight time are all essential data for planning effective mission strategies. The problem optimization graph is modeled based on the graph G(V, A), where V is the set of nodes, including the depot D, which is where the drone starts its tour with a full charge and empty space, and A is the set of edges connecting nodes. The drone returns when it needs to recharge, finishes the tour, and visits 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 W collected at a given location. When the drone's capacity load is full, it then unloads waste into the disposal location $V = \{N \cup D \cup S\}$, where V denotes a set of all nodes N, including the depot of the drone D and the waste disposal location S. Additionally, as A is the set of edges connecting nodes. Each edge is defined as the distance between nodes set as i to j with a nonnegative value d_{ij}. A drone traveling from one place to another consumes a certain amount of energy, which can be presented by $e_{i,j} = r \cdot d_{ij}$, where r denotes the rate of energy consumption. In this state, when the drone consumes most of the energy, it must return to depot D to recharge. UAV has a weight capacity of C. In case it reaches full capacity load, it travels to arrive at a disposal location S to unload and then finishes the unvisited waste nodes. The other case is when the energy of the UAV is almost depleted; then it goes to Depot D to charge. The drone has a maximum payload capacity that can be represented as MC and maximum power capacity ME, while the capacity constraint must not exceed the payload of the drone. The UAV requires multiple flights to cover all WCLs with the shortest distance for the sum of all flights within a certain time T. In general, this approach should reduce the total route distance and hence the cost. The problem is classified as an NP-hard problem 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 capacity [19]. The main objective of the model is to determine the most efficient waste collection route between designated locations that achieves the shortest path in the shortest time. Additionally, the model ensures that the volume of waste does not exceed the drone's cargo capacity. This means that the WCL cannot have a waste amount more than the capacity of the drone, so that the drone collects the waste of the location in one visit and does not have several visits to the same location to remove the remaining waste. If it does exceed the drone's cargo capacity, the drone unloads the waste at the disposal location and then continues servicing the remaining WCLs. Once all WCLs are serviced, the drone returns to the depot. Overall, this approach aims at minimizing the cost through minimizing total route fuel consumption, distance, and time. The most frequently used symbols in the mathematical model are listed in Table I.

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

1) Set the routing between different locations for a single drone to start its route from the depot and end at the same depot after it has fulfilled all waste collection operations.
2) Find the shortest path within the route mapping plan.
3) Determine the optimum predefined UAV capacity and flight time limits based on its battery charge limit.
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 operation process steps:

TABLE I: 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
dij	The distance between the nodes is set as i to j
r	Energy consumption rate
MC	Maximum load capacity
ME	Maximum energy capacity
C	Capacity of the drone

- UAV collects all the waste from all the WCLs.
- When the drone energy is almost depleted, it must go to the disposal and unload the waste load, then go to the depot to recharge, and then continue the tour until it finishes servicing all WCLs.
- When the drone reaches its maximum waste load capacity, it needs to go to the disposal location to unload the waste, then continue the tour until it finishes servicing all WCLs.

6) Calculate the efficiency value of the process when the UAV gives the fitness function of the shortest route with the least time and energy consumption to complete the routes.

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. Fig. 1, Fig. 2, Fig. 3, and Fig. 4 show examples of how the model works within only five waste locations, the drone operation process.

IV. Problem Modeling

The proposed model is an integration of GA and RL algorithms. All the details explaining the model process operations are in the next subsections.

A. Genetic Algorithms

The GA is operated in the first phase of the proposed model. GA is an efficient search technique that can optimize the solution based on natural sequenced processes. 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, as shown in Fig. 5. The subroutes consist of several routes; each route defines the distance between several WCLs that start from the depot and end 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, where the differences between each are shown in Fig. 6. Hence, Algorithm 1 illustrates the initial population algorithm that initiates routes from the depot. The drone finishes its route by going to the disposal location for unloading, returning to the depot for recharge, and then starting a new route to collect the waste from the remaining locations. The second operation process is the determination of the multi-objective fitness function that is required to satisfy the main constraints, which are to find an appropriate optimal short tour and to manage the waste load capacity with UAV energy. To do this, one way is to synchronize them into a single fitness value, where it is important to note that all constraints have the same properties. Hence, the multi-objective fitness function requires more algorithms to support the main requirements. Algorithm 2 shows the main component algorithms for the multi-objective fitness function. The fitness function is given in Equation 1.

$$\begin{aligned}\text{Fitness} = {} & \text{Cost paths penalty} \\ & + \text{capacity penalty} \\ & + \text{energy penalty} \\ & + \text{Penalty for Uncollected Locations.}\end{aligned} \tag{1}$$

1) The Cost paths penalty: Cost paths penalty represents the total travel distance between the route's edges of the WCLs, including the depot and disposal locations.

$$\begin{aligned}\text{Cost paths penalty} = {} & \text{1/Calculates the} \\ & \text{total distance for all routes} \\ & \text{(Euclidean distance).}\end{aligned} \tag{2}$$

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 shown in equation 3. It compares the distances to produce the best distance, that is, the shortest distance. Note that the best cost for each route is that with the minimum value produced from that route, hence it is the best cost path. Euclidean distance [20] is a popular method that is used to calculate the distance between two points. In our problem, it is used to calculate the distance between two waste locations (genes).

$$\text{Euclidean distance} = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{3}$$

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

2) Capacity penalty: Capacity penalty is used when the drone's capacity load exceeds the acceptable load (waste load limit). When the capacity limitation is

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 (DS) 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 (DS) 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 (DS) then return to the depot (D) on edge number 10.

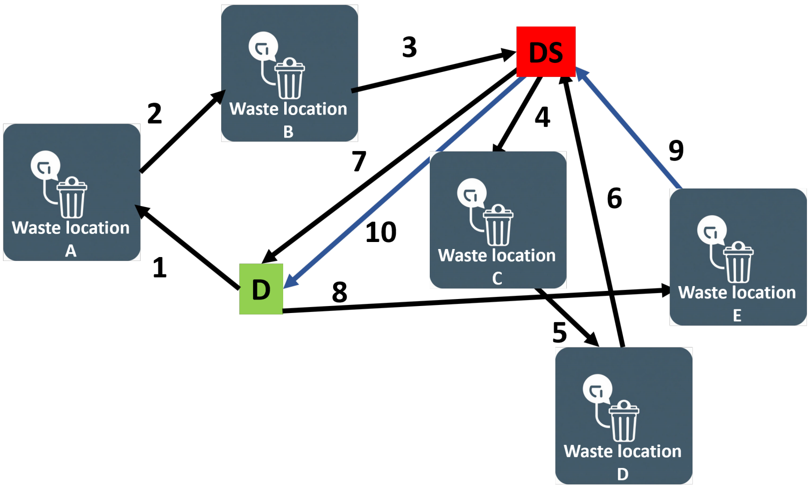

Fig. 1: Example of a Typical Waste Disposal and Recharging Case.

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 (DS) 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 (DS) then return to the depot (D) on edge number 8.

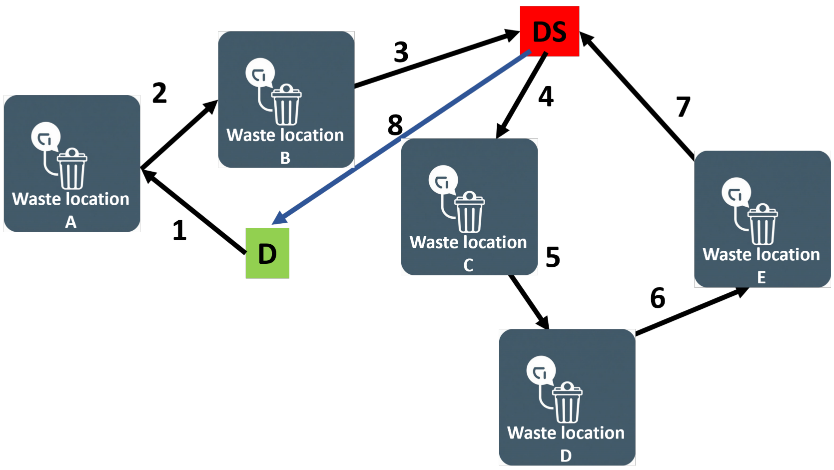

Fig. 2: Example of a Typical Waste Disposal Case.

violated, the penalty value increases and becomes a severely unacceptable condition. Otherwise, a zero value is added when there is no violation, and the drone's payload is within the drone's capacity.

$$\text{capacity\_penalty} = 1/\max(0, |\text{total\_load} - \text{drone\_capacity}|) \quad (4)$$

It should be noted that total_load represents the total weight of all the locations' waste load per drone carried, and drone_capacity is the maximum load that a drone can carry at a time. The capacity penalty ensures that the payload capacity of the drone is not exceeded. If the total weight of the drone's capacity load is less than the maximum waste weight capacity

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 (DS) 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 (DS) then return to the depot (D) on edge number 9.

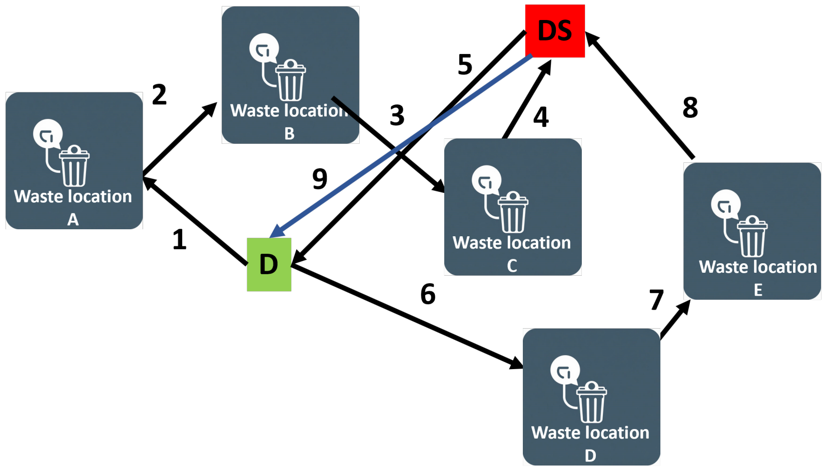

Fig. 3: Example of a Typical Recharging Case.

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 (DS) 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 (DS) 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 (DS) then return to the depot (D) on edge number 10.

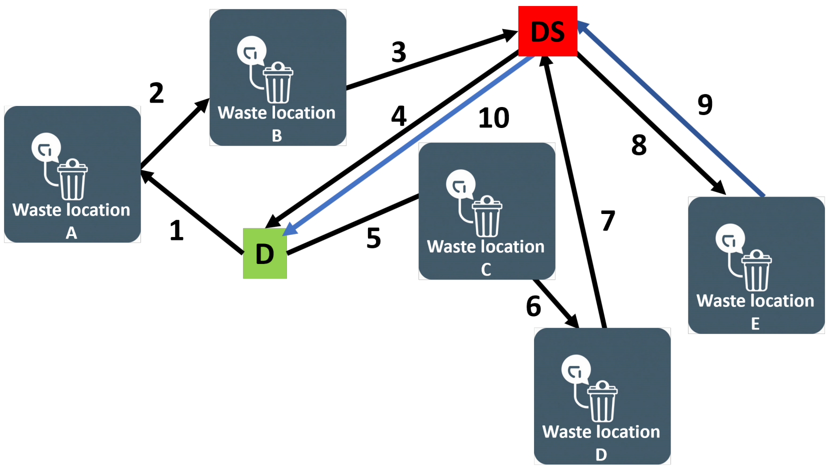

Fig. 4: Example of a Typical Recharging and Waste Disposal Case.

of the drone, the penalty will be zero. Otherwise, if it is equal to or more than the maximum weight, then it is calculated as the difference between the total_load and the drone_capacity. Moreover, it uses a Max(0) to ensure that the penalty value is not negative, which means that the penalty is only active when the capacity constraint is violated and the drone's capacity load is exceeded. In other words, if the capacity load is less than or equal to the UAV capacity, the UAV capacity will equal 0; thus, this equation will only be activated if the load is greater than the UAV capacity. The important part of the penalty is that the GA guarantees that solutions that do not violate its constraints are found.

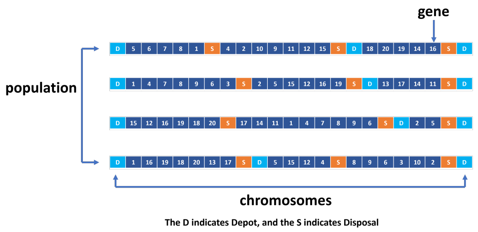

Fig. 5: Initial population

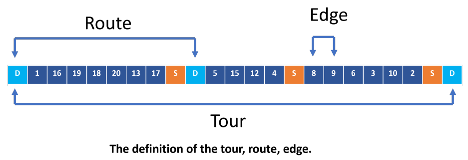

Fig. 6: The definition of the tour, route, edge

$$\text{flight\_time} = \frac{\text{route\_distance}}{\text{drone\_speed}}. \tag{5}$$

The Energy required, as well as the energy consumed for the disposal route, is indicated as the energy needed to go to the disposal site and dump off all waste loads. The amount of accommodated waste represents the waste load 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 [21] to reach the depot refers to the power needed to go to the depot to recharge. The drone's energy consumption is calculated similarly for all locations.

3) Energy penalty: The energy penalty represents the energy consumption during UAV operations for the entire tour, which can be done by summing the total calculation results of the 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 to follow up with servicing the remaining WCLs.
4) The penalty for Uncollected Locations represents the penalty for uncollected waste from locations that are necessary to ensure waste collection from the entire locations that have waste. It is calculated by the relation between the absolute count of locations respecting the unique locations that can be found by the difference between the total WCLs that the drone visited and the absolute count of locations to find the remaining locations that still need to be visited for waste collection. It is shown in the following equation:

$$\begin{aligned}\text{Penalty for Uncollected Locations} = \\ 1/(\text{total\_waste\_locations} \\ -\text{uncollected\_locations})\end{aligned} \tag{6}$$

5) Best Flight time is another efficiency factor that can be calculated within several functions to check

```
Start Initialize random population;
for _ in range(population_size) // based on data processing
do
    route =["depot"] // set depot as starting route
    while loop (unvisited_genes) do
        feasible_genes = [];
        for gene in unvisited_genes do
            distance = calculate_distance(current_location, gene);
            if distance ≤ remaining_capacity and distance ≤ remaining_energy=True then
                feasible_genes.append((gene,distance))
                Itr=all visited_genes
            end
            if not feasible_genes then
                if remaining_capacity = drone_capacity then
                    break;
                end
                else
                    UAV goes to disposal;
                end
            end
        end
        else
            sorted_genes=sorted(feasible_genes)
            selected_gene = sorted_genes[by less distance] ;
            remaining_capacity
            =int(selected_gene["WasteCapacity"]);
            remaining_energy = perform _trip
            (current _location , selected_gene);
        end
        else
            energy_available ≤ = energy_threshold;
            remaining_energy = drone_energy;
            UAV goes to disposal;
            UAV goes to the depot to recharge;
        end
        unvisited_genes = [gene for gene in waste_genes if gene not in route]
        for gene in unvisited_genes do
            route.insert(-1, gene) ;
        end
    end
    route.append("disposal")
    route.append("depot")
    print
end
```

Algorithm 1: Initialize Population Algorithm the system's performance. The first is the flight time, which calculates the edge time between two genes (two consecutive locations) as described before, and another function is the calculation of the route distance, which iterates over all genes in each edge and uses the total distance calculated for each edge to find the shortest time.

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. In this process, the tournament technique [22] is applied, which 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. Order crossover (OC), based on extensive research, is used to solve VRP problems [23]. The OC system selects a random subset of locations in the first parent (P1) and the second parent (P2) and then swaps the subset between P1 and P2. Moreover, in this research, the exchange mutation and the local search space are combined. Many works of literature regard the 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. The local search space used for routing problem solutions involves selecting and swapping two vertices, which can evaluate solutions and proceed to the next nearest neighbor. 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. Additionally, when it reaches the maximum iteration, the algorithm stops immediately. 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, it is considered that the algorithm has converged and stops the iterations. At each iteration of the loop, the current fitness is compared with the previous one and is evaluated and selected when it is better; hence, the best solution will be updated. The algorithm continues to iterate until it reaches the maximum iteration number or falls below the convergence threshold value, hence terminating at whichever condition is met first. This method is applied after each iteration to ensure the best system performance. 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 algorithm starts operating with a random selection of the population, which can help the algorithm in considering a way to avoid local optimization by exploring different regions of the solution space.
- The shifting of the population before the fitness evaluation provides population diversity, produces different combinations of locations (genes), and changes the order of the routes, 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 a literature review search and using trial-and-error techniques. The model can adapt to any type of parameter depending 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 of the GA operators are: tournament size, which is 5; the Mutation rate is 0.7; the crossover probability is 0.2; and generation number is 50.

B. The Reinforcement Algorithm

The second phase of the first hybrid approach to managing WCS is operated by the reinforcement algorithm (RL). In this part, we examine two different ways of integrating the GA with the RL algorithm. The next subsections describe the two ways of integrating GA with RL and how it is set up in the experiments to result in different outcomes for comparison in the analysis section.

1) The First method of incorporating the GA with the RL (GARL): The first scenario of the proposed model is shown in Fig. 8 for the hybrid integration of GA and RL (GARL). The RL is a type of [24] machine learning algorithm that works based on the main idea of trial and error. Agent actions depend on the current state, which is 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 [25] 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 gamma discount factor (γ). For example, if gamma 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 route to visit all the WCLs allocated. 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.

```
Start
Evaluate the population;
Set parameters;
def;
calculate_fitness,_components(solution,total_waste
_locations,drone_capacity,energy_available,energy
_disposal,energy_depot,drone_speed,nominal_power
_consumption,efficiency_factor,flight_duration,get
_site_capacity,calculate_route_distance,calculate
_energy_consumed for each route do
    if route_load = get_site_capacity(drone_route)
     then
        Calculate the total load for all distances
         Continue
        for each route do
            if route_load ≥ drone_capacity then
                num_of_disposal;
                and;
                depot_trips;
                is
                incremented
                total_load is decreased by route_load
                energy_available = energy_depot
                Apply energy_penalty
            end
            else if energy_available;
            ≤;
            energy_threshold
            =True then
                for each route do
                    Go to the disposal to empty the
                     load
                    Go to the depot to recharge
                    num_of_disposal;
                    and;
                    depot_trips;
                    is
                    incremented;
                end
            end
        end
        Energy;
        penalty#;
        Calculate;
        the;
        fitness;
        based;
        on
        constraints;
        cost_paths_penalty
        uncollected_locations_penalty
        capacity_penalty
    end
    Perform main fitness calculation
end
```

Algorithm 2: Algorithm of Fitness Functions

The choice of action state depends on the current state and the Q value, and 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 mostly employed to obtain the optimal path; the training phase is shown in Fig. 7 and Algorithm 4. The main formulation of RL is explained as follows:

$$Q(s,a) = Q(s,a) + \alpha \cdot (R + \gamma \cdot \max Q(s',a)) - Q(s,a) \tag{7}$$

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 the 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 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 WCLs allocated shown in Algorithm 2. In the training phase, the highest Q value is used primarily to obtain the optimal path; the training phase is shown in Algorithm 3. The main formulation of RL is explained as follows:

$$R(s,a) = F(s,a) \tag{8}$$

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 are chosen based on the values of the multi-objective fitness score.

$$F(s,a) = -10\left(FCost + FC + FE + FU\right) \tag{9}$$

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 regarding the UAV obligation to visit all WCLs during the entire tour. Fitness values give the agent feedback about the total values of the multi-fitness functions that will 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 on the basis of the highest Q-value in each state, considering constraints such as capacity and energy, the sequence of

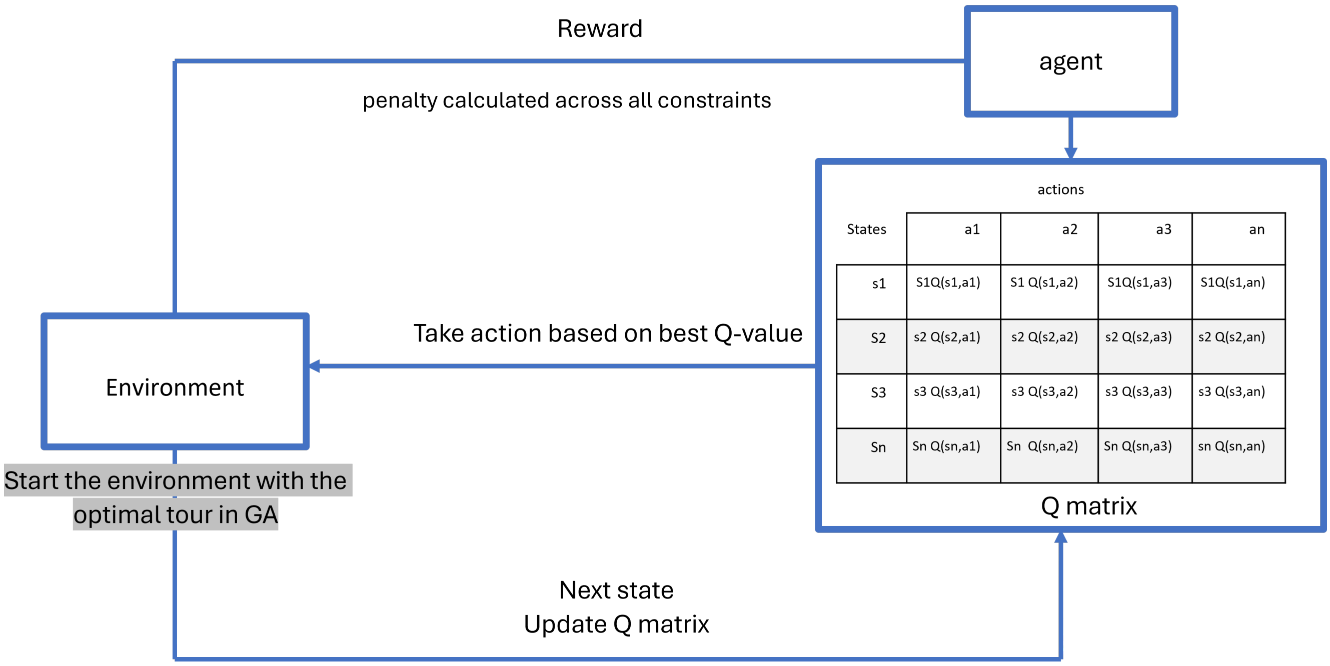

Fig. 7: The RL Process in the Model of Incorporating The GA with The RL (GARL)

these processes produces the best path for WCLs. A capacity penalty is calculated when the action taken for the next WCL violates the maximum capacity of the drone. Calculates the absolute difference between the current space capacity and the capacity state in the next location and sets -10 as a penalty value for an 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:

$$\text{capacity\_penalty}(s, a, s') = -10 \cdot (|c(s') - c(s)|) \tag{10}$$

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:

$$\text{energy\_penalty}(s, a, s') = -10 \cdot (|e(s') - e(s)|) \tag{11}$$

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. $(|e(s') - e(s)|)$ will calculate 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:

cost_paths_penalty(s, a, s') =-10 (1/total_distance(s, a, s')) (12)

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 as a penalty for a longer trip. 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 WCLs multiple times when needed based on the shortest route.

$$\begin{gathered}\text{penalty\_for\_uncollected\_locations}(s, a, s') = -10 \\ (1/(\text{total\_waste\_locations} - \text{uncollected\_locations}))\end{gathered} \tag{13}$$

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.

2) The second model that incorporates the RL first, then the GA (RLGA): The second model is to start with operations of the RL algorithm that can show better adaptation to dynamic environments, add known data, and exploit them to increase rewards, as shown in Fig. 10, while GA can essentially explore new potential solutions. Thus, this experiment starts with the steps of the RL algorithm based on the actions of the agent interacting with the environment. The RL interacts with the WCL data and hence 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 WCL to find the best

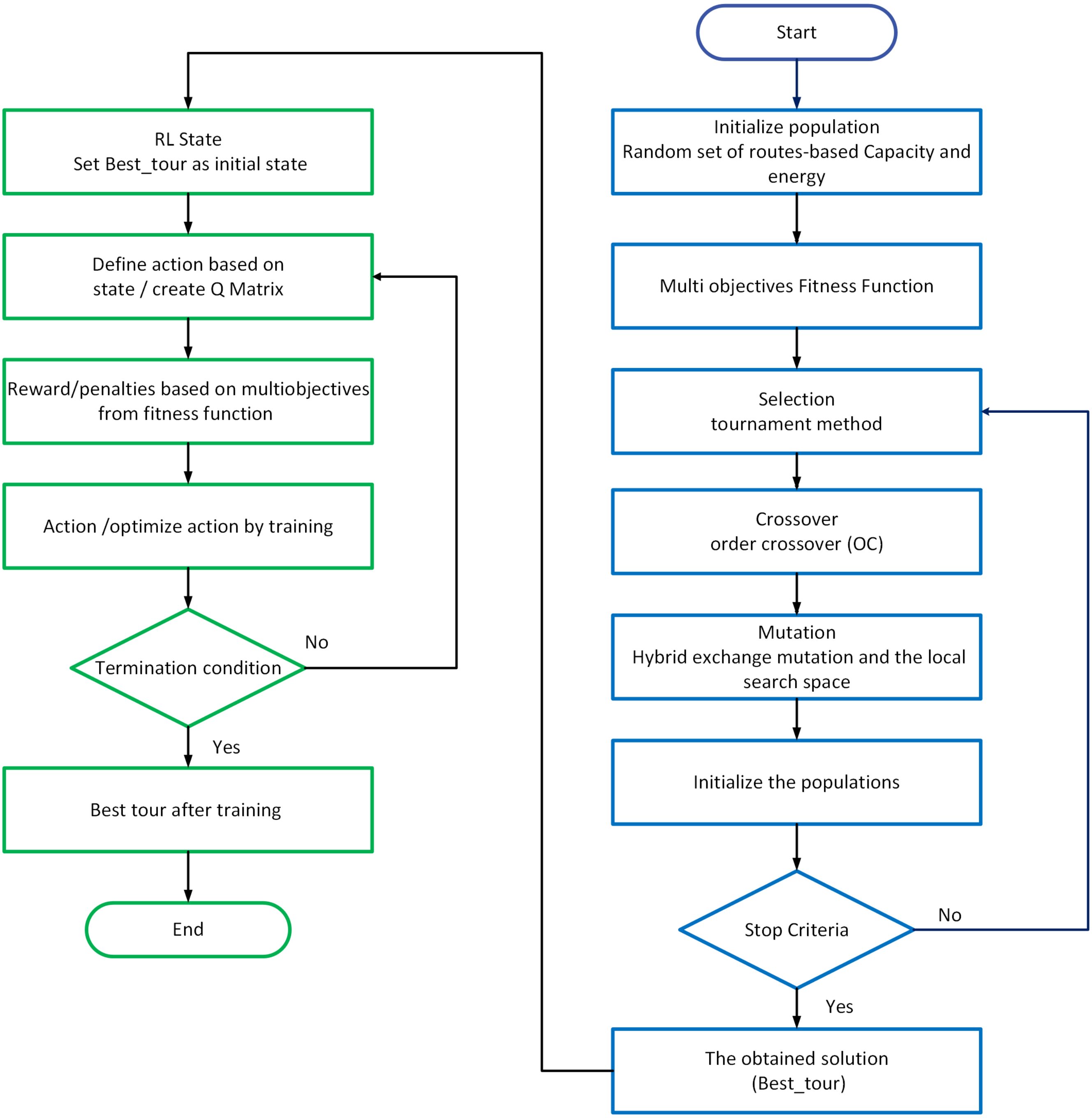

Fig. 8: The First Model that Incorporates the GA with the RL (GARL)

path based on the epsilon greedy policy within all constraints and requirements. It starts collecting the experiment values and updating the Q matrix based on the rewards received and the maximum expected future rewards. The process continues until an optimal policy is found that increases the reward values. The final tour output is then synchronized with GA and used as initial data to start within the GA operations instead of a random population, and it continues to generate more populations starting from this RL output. Then it goes through the same GA processes as the GA operations continue to refine the best RL tour output within GA process. Fig. 9 shows the hybrid integration of RL and GA in the second proposed hybrid model.

V. Experimentations and Results analysis

The proposed solution is implemented using Python 3, Google, and Jupyter; embedded with other libraries such as numpy, pandas, ga, random, math, copy, seaborn, pyplot, bese, deep, matplotlib, cluster, partial, plugins, folium, and ortools. All of these modules provide several functionalities for efficient optimization, data manipulation, plotting, and

```
Start
Initialize necessary variables and parameters
for episode in range(episodes) do
    state= env.reset(best_route) # set environment to
      get initial state as best route
    done = False
    episode_reward = 0
    for step in range(max_steps) do
        # iterates maximum number of steps
        action = agent.act(state);
        # The agent selects an action based on the
          current state
        next_state, reward, done = env.move(action);
        # environment updates with the chosen
          action, producing the next state and reward.
        for route in route_list: do
            #iterating through each element in the
              route_list
            if isinstance(route, int) then
                route_load = get_site_capacity(route)
                total_load += route_load
                total_distance + =
                  calculate_route_distance(route)
                num_locations + = 1
                continue
            end
        end
        for waypoint in route do

        end
        #Iterating through each waypoint in the route.
        Calculate the route_load
        Update total_load, total_distance, and
          num_locations
        if route_load ≥ drone_capacity: then
            Go to the disposal site to empty the load
            Update the energy_available and calculate
              the energy_penalty
            if energy_available ≤ energy_threshold
              then
                remaining_energy = drone_energy
                UAV goes to disposal
                UAV goes to the depot to recharge
            end
        end
        uncollected_locations=[total_waste_locations;
        -num_locations]
        capacity_penalty=total_load/(num_disposal_trips
          *drone_capacity)
        cost_paths_penalty = 1 /total_distance
        total_penalty = -10 * (energy_penalty +
          capacity_penalty + (1/(uncollected_locations
          + 1)) + cost_paths_penalty);
        Update the total reward
        total_reward += reward + total_penalty;
    end
    print
end
```

Algorithm 3: Training Algorithm

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, uses a compute engine backend (TPU) with RAM 6.12 GB/334.56 GB and disk 16.09 GB/225.33 GB.

Table II shows the experimental parameters for the GA part, and Table III shows the parameters for the RL part; the same parameters are used for the three scenario types used in this study. 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 use synthetic data, which is employed to apply a theoretical approach to a problem when there are insufficient real-world data [26]. 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 latitude and longitude, with the waste capacity of each WCL. The specified WCL data are based on the scale of specific WCLs, not on cities. The process of synthetic data generation is shown follow:

- 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 kilos and is randomly generated between 10 and 100 kilos.
- The type of waste, drone type, or time limit is not included in the synthetic data generation to allow moderation in the model based on real-world WCS specifications.
- The tour is generated based on several routes and paths that can be visited more than once if the distance is shorter.
- The waste capacity and the energy of the drone are based on the manufacturer's specifications. They are taken from the most popular type of drone used in WMSs and can be modified based on the drone used in the real world [27].

The accuracy and size of the data are crucial for producing reliable and predictable results. Since real WMS data may not be available, synthetic data becomes an ideal substitute. In addition, it supports data diversity, which is essential when real-world data cannot be captured across all scenarios. The effectiveness of using synthetic data over real-world data in terms of time and cost advantages is also notable. The data is utilized to evaluate the proposed model when conducting the experiments and analyzing the results.

The comparison is based on the specified population size and the WCL numbers as 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

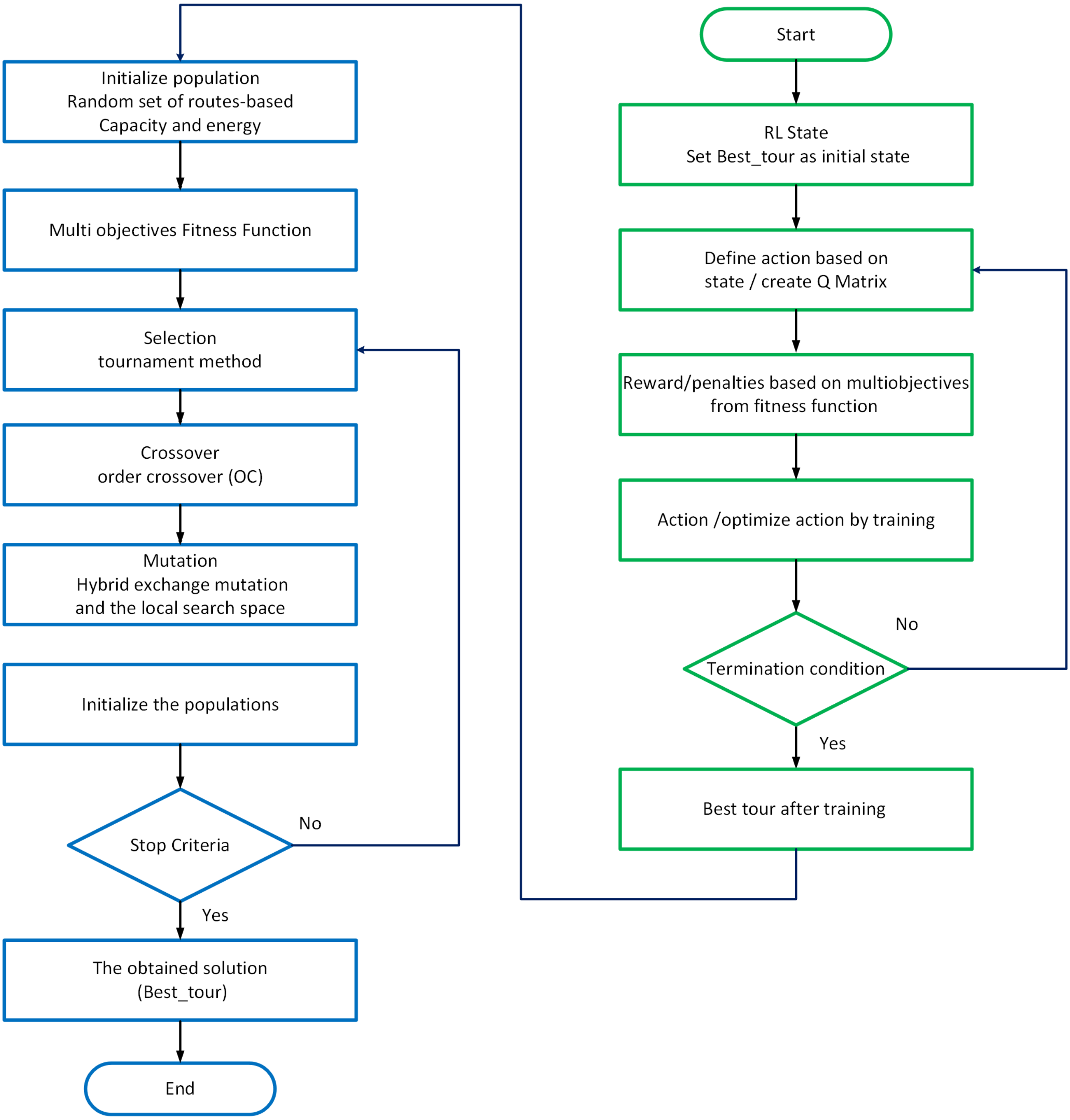

Fig. 9: The second model that incorporates the RL first, then the GA (RLGA)

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 [24]. In addition, the drone capacity is 200K and the drone power is 500W [25]. Table III shows the parameters of the experiments for RL, where the parameters change based on the number of data, such as the number of WCLs and the number of episodes; therefore, the maximum steps increase as the data increases. Whereas when the episodes are fixed to 1000 episodes, then the maximum number of steps is 100 for all the WCLs. Moreover, the other parameters are fixed, as epsilon is 0.1, which is the probability of a random action exploring better possibilities for improved actions by initial exploration with 100%, and this rate decreases with the progression of the training process 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

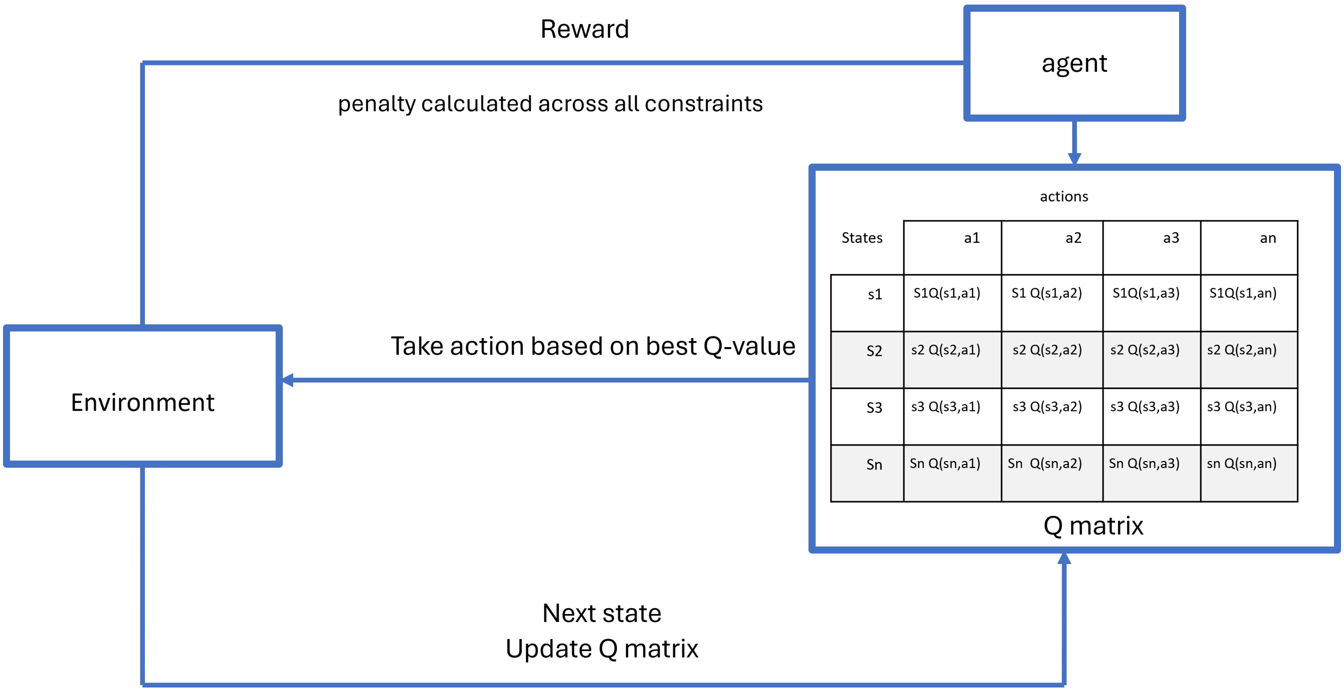

Fig. 10: The RL process in the second model that incorporates the RL first, then the GA (RLGA)

TABLE II: GA 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

TABLE III: RL Experiments Parameters

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

reducing the exploration over time. The epsilon minimum 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 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. 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 study includes four experiments that aim to compare the performance of algorithms individually or in combination to meet problem solving objectives. These experiments are used to find the best fitness values and flight time among these algorithms within certain constraints. Hence, these experiments employ algorithms either individually as individual algorithms GA and RL, or in combination as hybrid algorithms GARL and RLGA.

Table IV shows the best fitness values and the best flight time of the models in different sets of WCL numbers among all four experiments. From the experiments, the first prominent result is that the GARL approach records better results than

TABLE IV: 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

other approaches in fitness values, recording 4.75, 8.698, and 11.42 for 100, 500, and 1000 WCLs, respectively. Even in terms of flight time, its score is better than other approaches, scoring less time, which is recording 39.97 minutes for 100 WCLs, 199.18 minutes for 500 WCLs, and 374.2 minutes for 1000 WCLs. Secondly, the hybrid RLGA records better results than both single GA and RL approaches, but not the hybrid GARL. The fitness values are 5.28, 10.49, and 13.57, for 100, 500, and 1000 WCLs, respectively. The flight time is 41.02 minutes for 100 WCLs, 202.5 minutes for 500 WCLs, and 381.97 minutes for 1000 WCLs. The third experiment that examines GA as a single algorithm shows that it records better values than the values of the fourth experiment employing the single RL algorithms. In demonstration, the GA records a fitness value of 5.34, which is better than the RL value results, which record 5.44 for 100 WCLs. For 500 WCLs, GA records a fitness value of 11.16, which is better than the RL value result, which records 11.23. Furthermore, for 1000 WCLs, GA records a fitness value of 15.5, which is better than the RL value result, 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, GA records 203.5 minutes, which is better than the result of the RL value, which records 204.8 minutes. And finally, GA records 382.3 minutes, which is better than the RL value result, which is 384.9 minutes for 1000 WCLs. In conclusion, hybrid approaches generally perform 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 in the first phase, then GA. Additionally, all experiment values are calculated based on a running average, where for the 100 WCLs set, it takes an average of 10 runs, which comprises 10% of the WCLs number, then for the 500 WCLs set, it takes an average of 30 runs, which comprises 6% of the WCLs number, and last for the 1000 WCLs set, it takes an average of 50 runs, which comprises 5% of the WCLs number. 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 clearly in the flow charts in Fig.11, which demonstrates the best fitness values among all experiments, and in Fig. 12, which shows the flight time improvements among all four experiments.

Table V shows the percentage of improvement when comparing the proposed hybrid approaches and the single corresponding algorithms. Performance improvement is observed to increase successively among the three sets of WCL numbers using the hybrid approach that integrates both GA and RL. For the first and best hybrid approach, the performance of the GARL approach improves over that of the GA algorithm alone by 11.05%, 22.06%, and 26.32% for the three sets of 100, 500 and 1000 WCLs, respectively. Moreover, the performance improvement of the same hybrid algorithm over RL alone also shows successive improvements with an increasing number of WCLs, which record 12.68%, 22.55%, and 26.42% for the three 100, 500, and 1000 WCLs, respectively. The performance improvement of the proposed hybrid approach of RLGA also records improvements compared to GA alone and shows successive improvements with the increase in the number of WCLs of 1.12%, 6.00%, and 12.45% for the three 100, 500 and 1000 WCLs, respectively. And when compared with the RL approach for the three 100, 500, and 1000 WCLs, it records 2.94%, 6.59%, and 12.56%, respectively. Maximum improvements are observed with the best proposed approach, which is GARL. Consequently, the percentage of performance improvements is highest for the RLGA hybrid approach compared to the RL algorithm due to the difference between their highest and lowest performance efficiency, even compared to GA alone. In addition, successive improvements with an increase in the number of WCLs indicate that 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 Fig.13, which shows the performance improvement in percentages among all scenarios of the proposed hybrid algorithms compared to the performance of each other and individual algorithms, GA and RL algorithms alone. In Table VI, the timing improvements are shown along with the increase in the number of WCLs from 100 to 1000. When 100 WCLs sets are compared to the GA algorithm, the GARL is 2.27 minutes better, and when compared to the RL, it is 2.51 minutes better. For 500 WCLs sets, it improves by 4.32 minutes compared to GA and 5.62 minutes compared to RL. A similar but also successive improvement for 1000 WCLs sets is improved by 8.1 minutes compared to GA and by 10.7 minutes compared to RL. This demonstrates

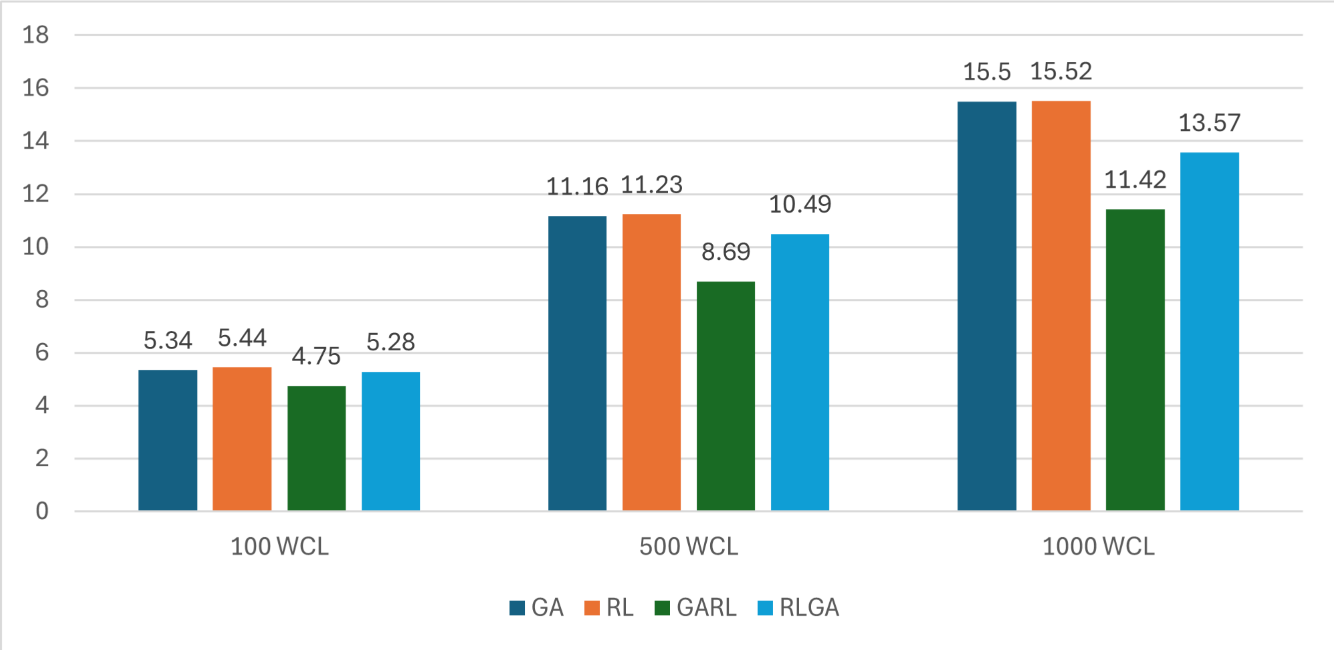

Fig. 11: Best fitness values.

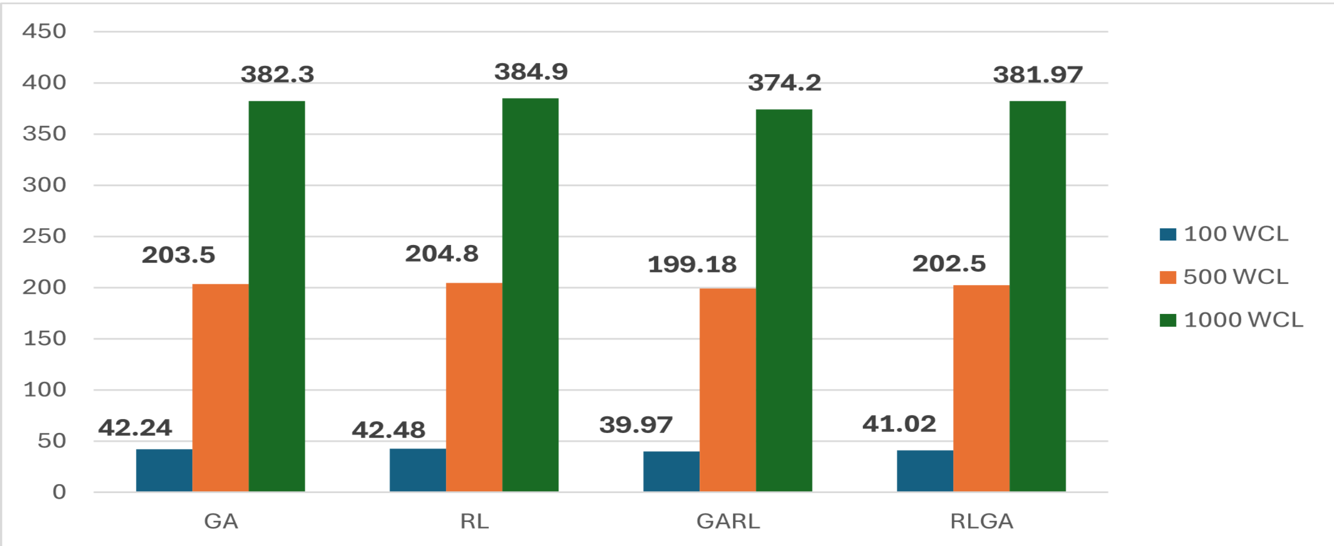

Fig. 12: Flight Time Values Among All Four Experiments.

that the improvement is greatest when compared to the performance of the RL algorithm alone, due to its lowest performance when compared to all other approaches. GA alone. When comparing GARL with the second proposed model, RLGA shows a better time improvement of 1.05 minutes for 100 WCLs. The improvement rises to 3.33 minutes for 500 WCLs and increases to 7.7 minutes for 1000 WCLs. 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. Moreover, the GARL model shows high performance based on improvements in the behavior of the model when it comes to increasing the number of WCLs. Hence, the higher the number of WCLs, the better the performance.

To analyze the results further, 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

TABLE V: The Experiment's Percentage of Improvement.

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

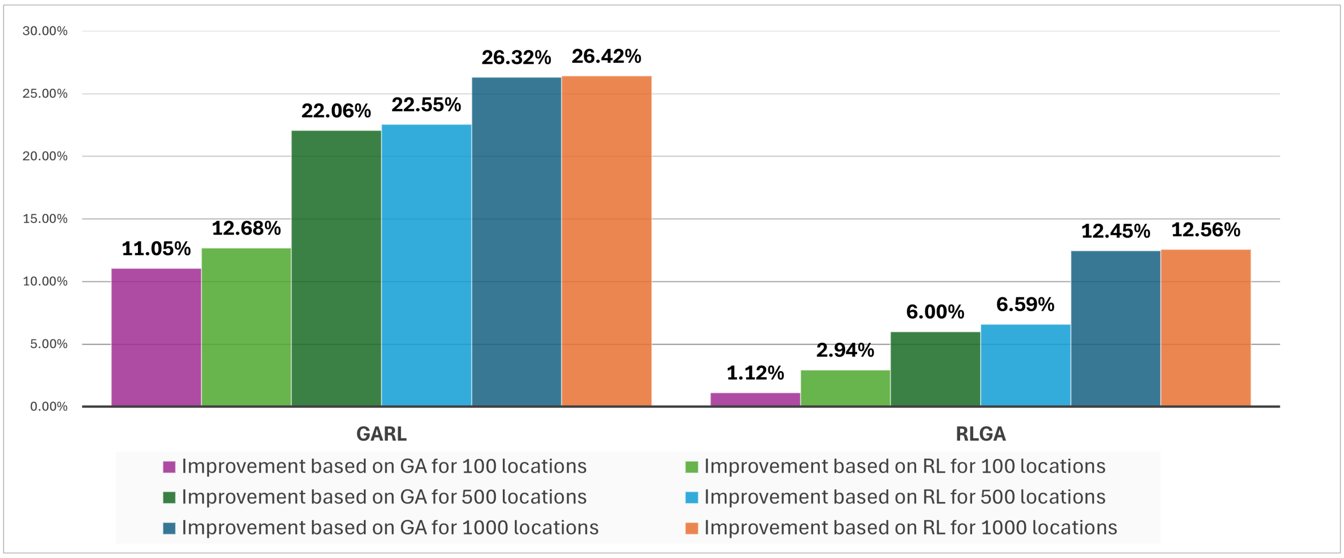

Fig. 13: Improvement Percentage Among All Algorithms Compared to the Individual Algorithm Performance (GA and RL).

TABLE VI: Experience Improvements Timing Result

Population waste locations	100 WCL		500 WCL		1000 WCL	
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	-

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 IV 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 as 0.548, 0.829, and 0.8 for 100,500, and 1000 WCLs sets, respectively. Thus, the results show more data consistency and less variability. 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. Hence, comparing GA and RL algorithms, RL gains better data consistency, although GA outperforms it in other evaluation variables. Furthermore, the experiment results of the hybrid proposed methods are compared with two studies presented in the literature review section that use similar data sets and address similar constraints and use other approaches. The first comparative research used a hybrid ACO [18], and the second research used a 2-phase approach, namely Maximum Waste in a MWMTT [17]. In this study, it is found that GARL scores better than the RLGA, which shows better results than both comparative studies that used similar three datasets. Each of the compared studies calculated 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, we take the best fitness of the comparative study and subtract it from the best fitness of the proposed model; then divide the value by the best fitness of the comparative study; and multiply the result by 100 to convert it to a percentage. This calculation reflects the percentage reduction in fitness error when comparing this study's results to the comparative study's results. 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, along with 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. Table VII and Table VIII show that the proposed hybrid methods prove to be better approaches to solving path planning problems than 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 in the third database are 58.09% for GARL and 50.11% for RLGA. The percentage of reduction in fitness error for the first database in the proposed hybrid approach of the comparative search hybrid (ACO) is 25.9%. The percentage of reduction in fitness error for the second database in the proposed hybrid approach of the comparative search hybrid (ACO) is 27.4%. Furthermore, the percentage of reduction in fitness error for the third database in the hybrid approach proposed by the hybrid comparative search (ACO) is 27.2%. Furthermore, Fig.14 demonstrates the percentage of reduction in fitness error in the proposed GARL and RLGA compared to the first, second, and third data sets used in the hybrid ACO.

Furthermore, Table VIII compares the results of the method (MWMTT) with the results of the proposed hybrid methods. At 100 WCLs, GARL improves by 86.8% and RLGA by 85.4%, better than the first database used in the compared MWMTT model. Furthermore, in the second database, GARL improves by 83.79% and RLGA by 81.86%. In 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. In the second database, GARL improves by 83.79% and RLGA by 81.86%, respectively. In the third database, GARL improves by 59.01% and RLGA by 50.57%. In addition, improvements were recorded at 1,000 WCL 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 compared to the research approach (MWMTT). Furthermore, the percentage of reduction in fitness error for the second database in the hybrid approaches is recorded as 29 compared to the research approach (MWMTT). In addition, the percentage reduction in fitness error for the third database in the hybrid approaches is recorded as 21.2 compared to the research approach (MWMTT). Moreover, Fig.15 presents a flowchart showing the percentage of reduction in fitness error in each of the different datasets of WCLs when comparing the proposed hybrid methods and each of the different datasets used by the MWMTT comparative research. 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 locations 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 comparable studies in the following respects:

- The proposed approach includes four models, and within each, three experiments covering the number of WCLs (100, 500, and 1000), one depot, and one waste disposal location. Comparative research with small cases between 10-40, medium cases 41-48, and large cases between 49-89.
- Regarding power consumption, the proposed approaches can lead to better results because the drone covers more locations in each trip, recharges when needed, and does not stop early before consuming all of the drone's power, as is the case in the compared studies.

TABLE VII: The comparison between the research methods (hybrid ACO) and the proposed approaches.

GA dataset (Number of locations)	Percentage reduction in fitness error of DS1 (Hybrid ACO)	Impro-vement via GARL	Impro-vement via RLGA.	Best fitness of DS2 (Hybrid ACO)	Impro-vement via GARL	Impro-vement based on RLGA	Best fitness of DS3 (Hybrid ACO).	Impro-vement via GARL	Impro-vement 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 VIII: The comparison between the research methods (MWMTT) and proposed approaches.

GA dataset (Number of locations)	Percentage reduction in fitness error of DS1 (MWMTT)	Impro-vement via GARL	Impro-vement via RLGA	Best fitness of DS2 (MWMTT)	Impro-vement via GARL	Impro-vement via RLGA	Best fitness of DS3 (MWMTT)	Impro-vement via GARL	Impro-vement 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%

- The linear program in comparative studies could not generate solutions that solve instances with more than 40 locations; hence, the lower bound is used to evaluate the performance of medium and large size experiments. Therefore, the results obtained were in line with those obtained for small instances, which is an obvious limitation that is overcome in our study approach.

To illustrate more the results for all experiments and algorithms were run multiple times to take an average based on WCLs, thus, for 100 WCLs, it was run over 10 times, Ratio and Proportion. Tables IX and X show the GARL and RLGA average results. For 500 WCLs, it runs more than 30 times, the ratio and proportion. Fig. 16 and Fig. 17 show the GIRL and RLGA average results. For 1000 WCLs, it runs over 50 times, Ratio and Proportion. Fig. 18 and Fig. 19 show the GARL and RLGA average results.

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 model over the results of the comparative studies and the achievement of the objectives of this study. Its external validity comes from its consistency 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, were not considered in this study. In conclusion, the best proposed model is the one that combines 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 simultaneously 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 route by GA, which is more reliable and near optimal. Its outcome population starts as the input

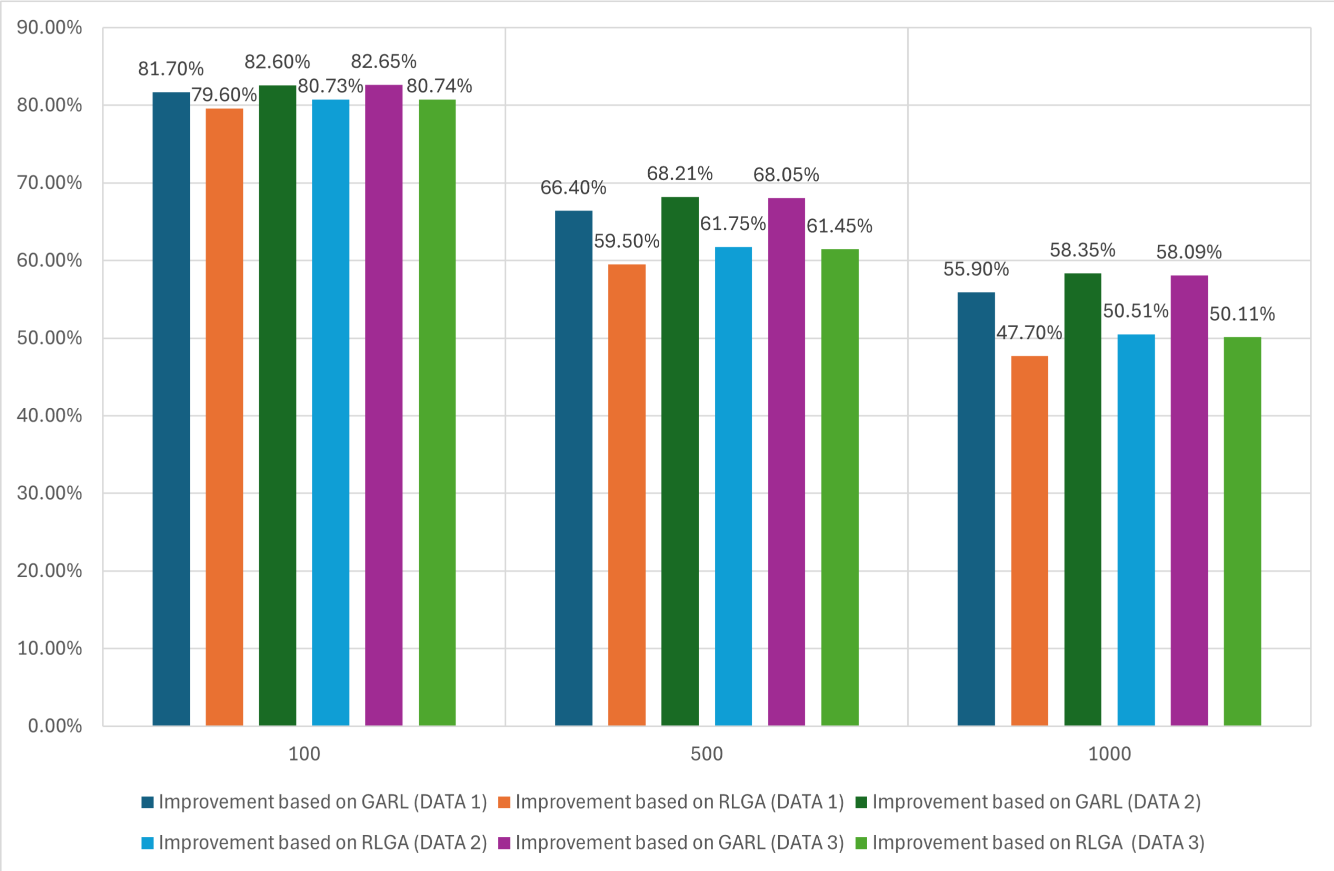

Fig. 14: The percentage reduction in fitness error that was compared between the proposed hybrid approaches with hybrid ACO.

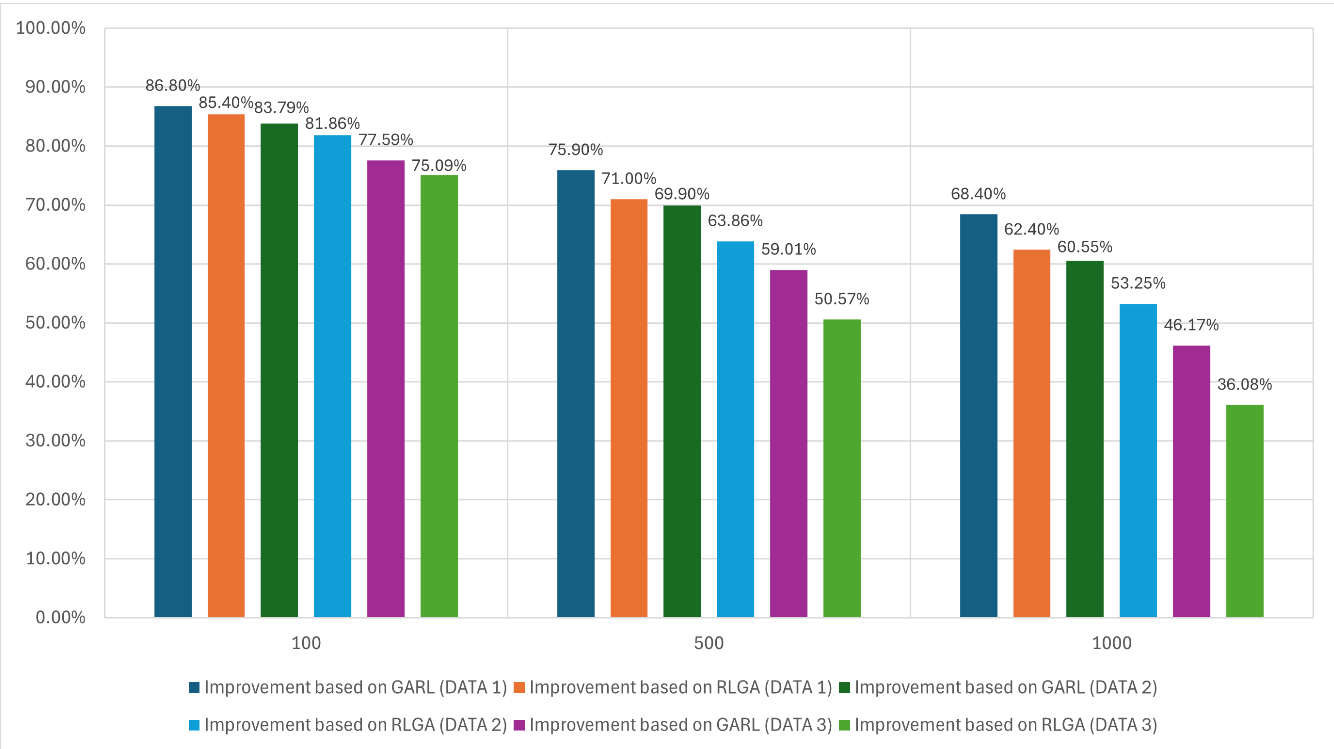

Fig. 15: The percentage reduction in fitness error that was compared between the proposed hybrid approaches with MWMTT.

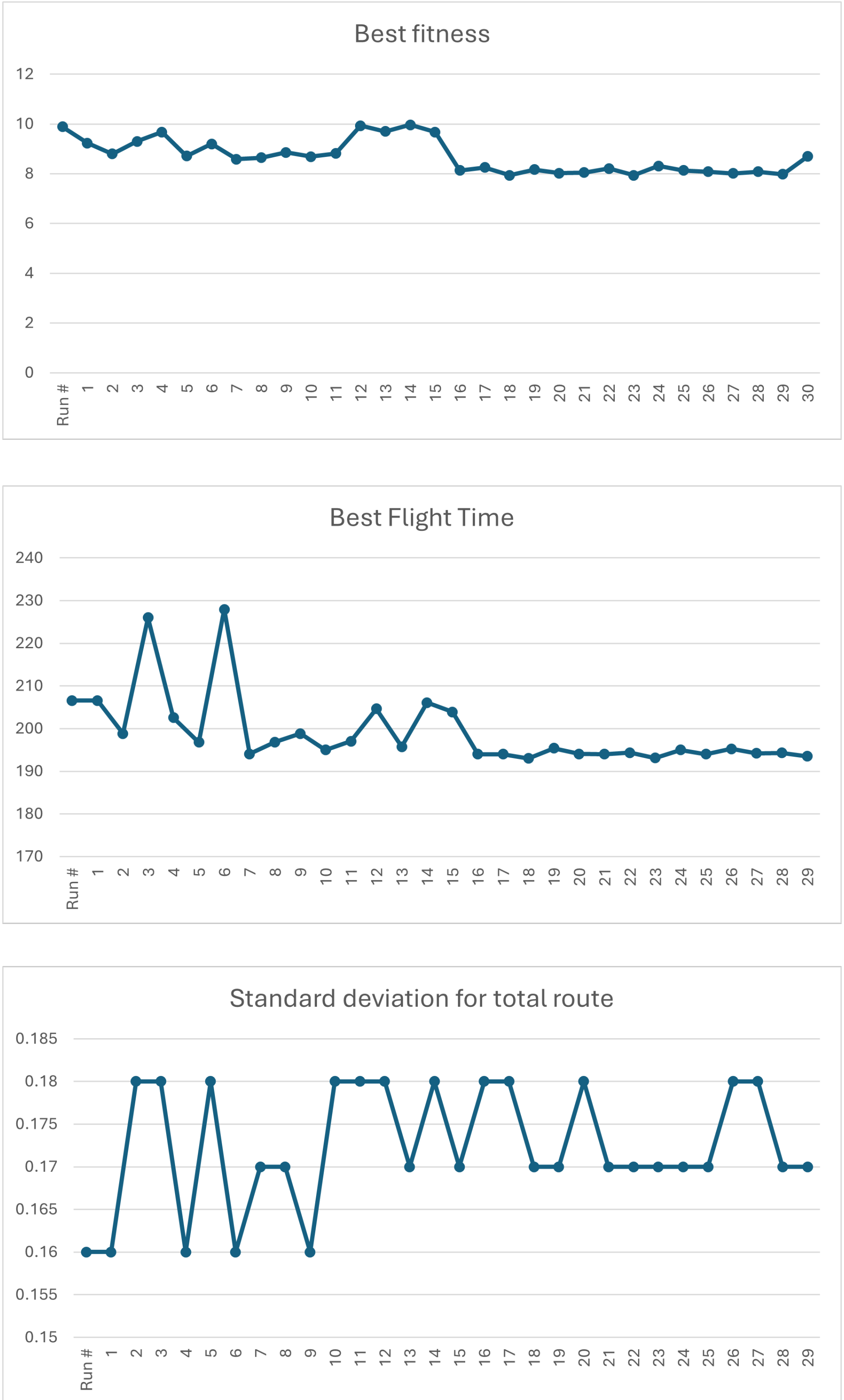

Fig. 16: The Results Obtained for GARL Performance Metrics for 500 WCLs

population in the second phase of the 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 performs well in complex problems with multiple variables and a high-dimensional space. Conversely, RL can lead to poor balance between exploration and exploitation, which can lead to missed opportunities at certain times or reduced

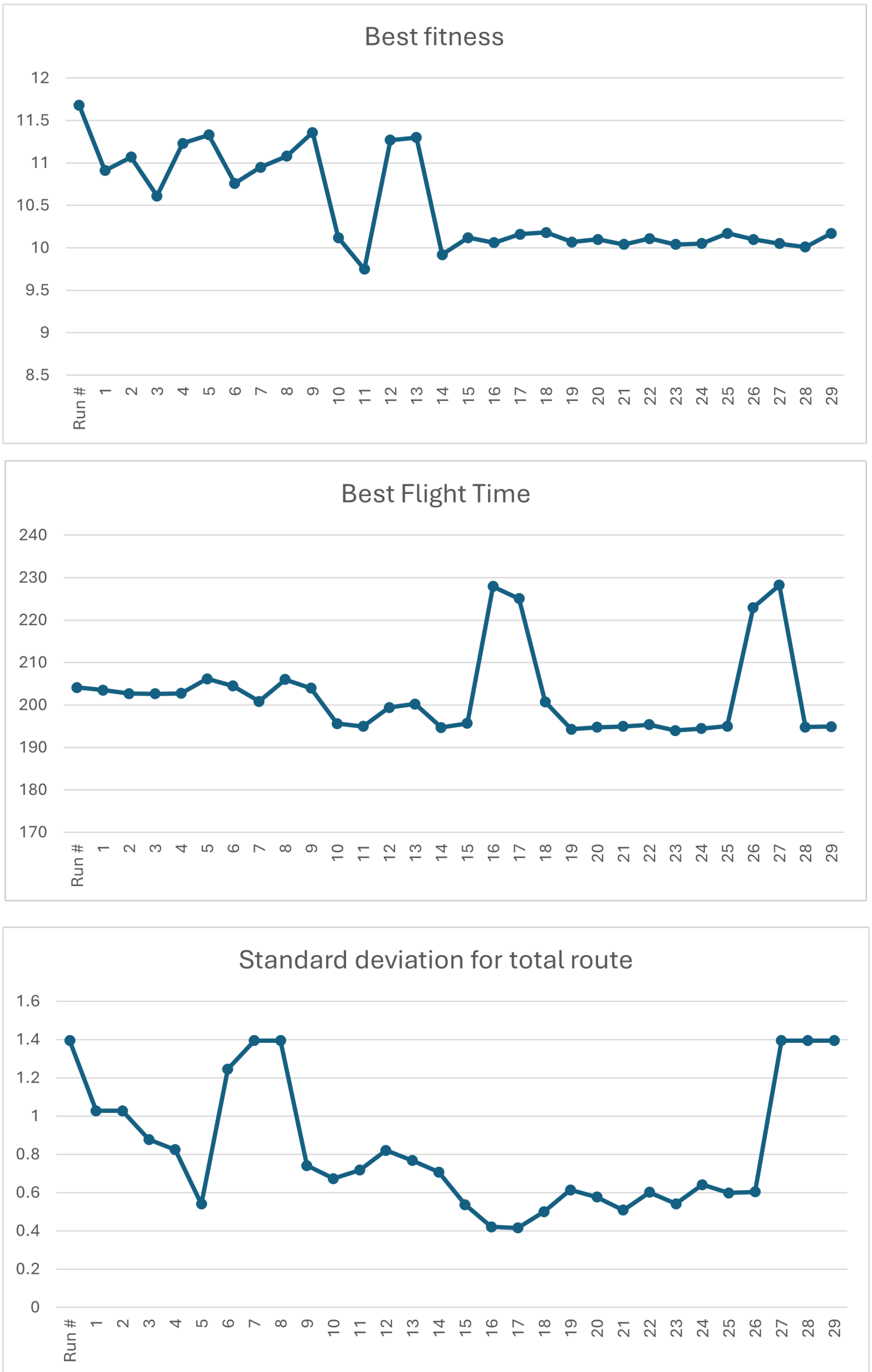

Fig. 17: The Results Obtained for RLGA Performance Metrics for 500 WCLs

performance. Furthermore, RL often requires more training time and computational resources to produce better results. Therefore, the results of the proposed model that combines GA and RL (GARL) are significantly better than those of GA and RL (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.

VI. Conclusions

This paper proposes multi-objective hybrid models of GA and RL to solve routing problems using a single UAV. It aims to enhance WCS by solving its routing problems

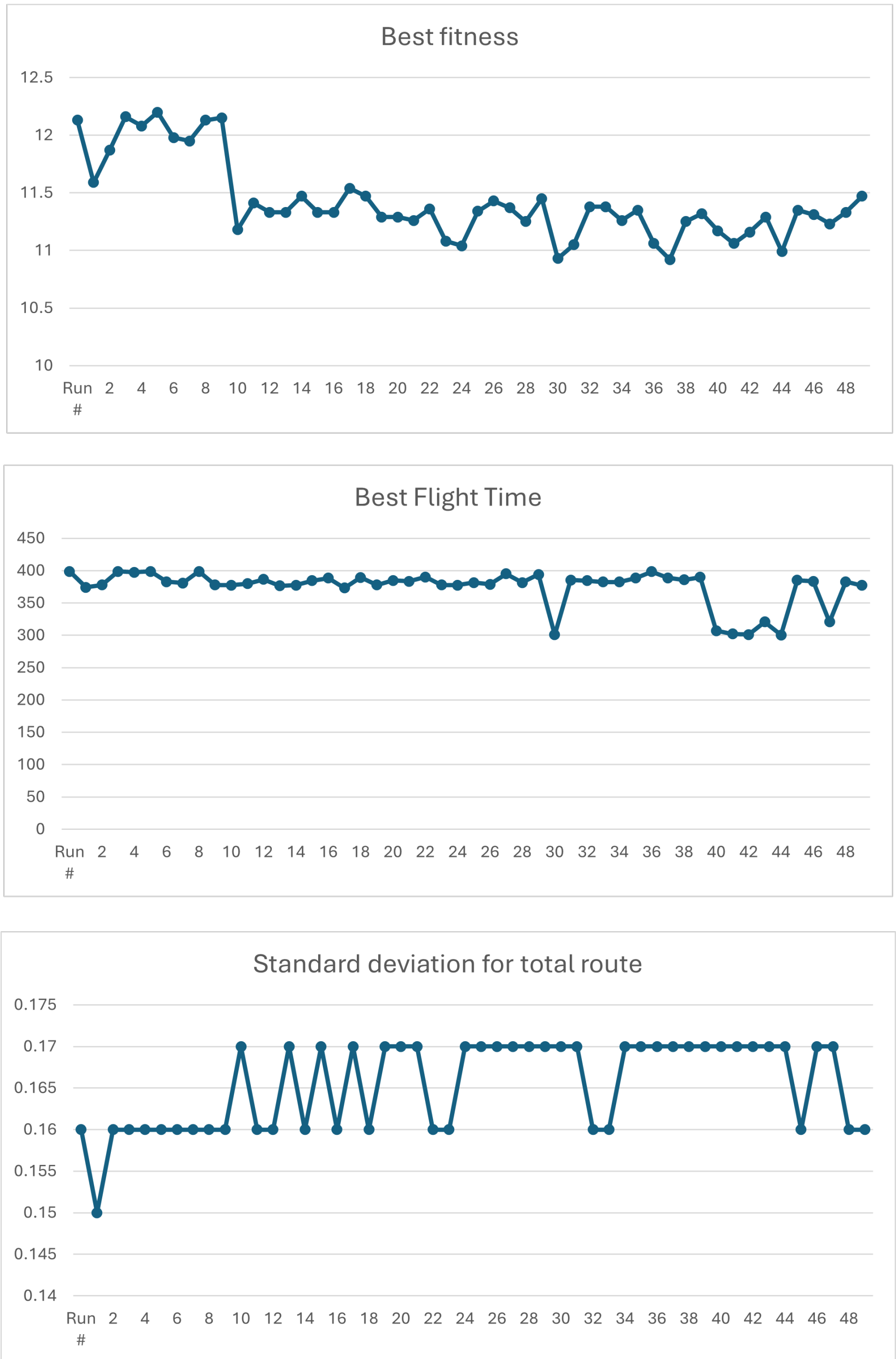

Fig. 18: The Results Obtained for GARL Performance Metrics for 1000 WCLs

in complex scenarios. The proposed models present hybrid and individual methods of the GA and RL algorithms. The models are applied using synthetic data, generating 1000 WCL data. Each of the experiments is divided into three sections, including 100, 500, and 1000 WCLs. The hypermeters of the experiments are stabilized for all experiments and are selected based on literature review and trial-and-error. The first model is GARL, which is divided into two phases. The first phase starts with GA operations that leverage initial random population selection based on a well-constructed fitness function that addresses various constraints such as UAV capacity and energy endurance while servicing all WCLs in the shortest tour distance and time. The second phase of the model operates the RL algorithm, which is synchronized with GA by taking its best population outcome as the initial state to generate the Q matrix in the learning phase. In the second proposed approach, RLGA, the algorithm's sequence is reversed,

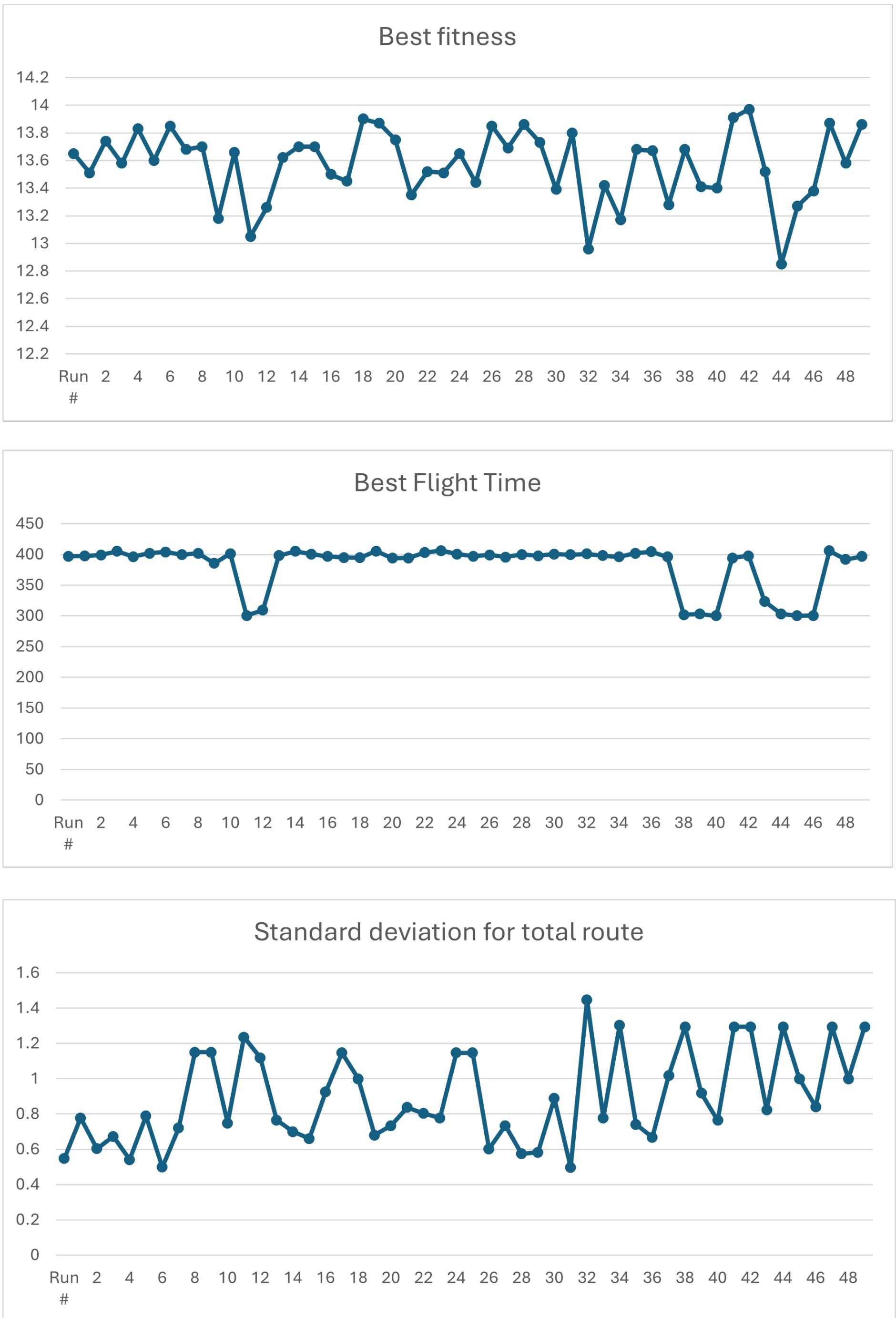

Fig. 19: The Results Obtained for RLGA Performance Metrics for 1000 WCLs

hence operating the RL algorithm first and importing its best tour output to be utilized as the initial population in GA instead of using a random one. Furthermore, the results of the two models are used to demonstrate the superiority of the hybrid approach over individual algorithms of GA and RL through comparison of the results of the four approaches. Hence, the results of individual GA and RL algorithms are used as benchmarks to rank the level of efficiency and superiority of the two hybrid approaches. Comparisons of the models' 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. The flight time is also calculated as another measure of the efficiency matrix of the proposed methods. The GARL gained the highest improvements over all other approaches in terms of fitness value and flight time. Compared to GA-only results, it recorded 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 the results of RL alone recorded 12.68% and 2.51

TABLE IX: The Results Obtained for GARL Performance Metrics for 100 WCLs

Metric	1	2	3	4	5	6	7	8	9	10	AVG
Best fitness	4.67	4.89	4.69	4.86	4.73	4.5	4.82	4.75	4.91	4.7	4.752
Best Flight Time	41.43	39.54	39.31	39.78	40.23	38.46	40.45	39.4	39.82	41.23	39.965
Standard deviation for total route	0.732	0.3887	0.6161	0.5359	0.7257	0.5009	0.5274	0.5744	0.6921	0.4713	0.57645

TABLE X: The Results Obtained for RLGA Performance Metrics for 100 WCLs

Metric	1	2	3	4	5	6	7	8	9	10	AVG
Best fitness	5.11	5.06	5.12	5.82	5.23	5.13	5.03	5.21	5.85	5.22	5.278
Best Flight Time	41	40.13	41.11	42.31	41.22	41	40.16	41.09	41.07	41.11	41.02
Standard deviation for total route	0.5802	0.407	0.5153	0.6698	0.6003	0.4343	0.4214	0.5013	0.4717	0.8822	0.54835

minutes in 100 WCL, 22.55% and 5.62 minutes in 500 WCL, and 26.42% and 10.7 minutes in 1000 WCL. Then, RLGA becomes the second-best performance approach. Hence, the proposed hybrid approaches show great potential in solving routing problems over individual algorithms by leveraging the control optimization power and synergy of both the GA and the RL together. In general, it provides an effective framework for identifying optimal or near-optimal solutions applicable to complex scenarios in the real world. Future studies should include studying the performance of multiple UAVs similar to the one in this study by distributing roles among them and using an efficient method similar to the approaches used in this study based on the required constraints.

REFERENCES

[1] S. Hejres, A. Mahjoub, and N. Hewahi, "Routing approaches used for electrical vehicles navigation: A survey," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 801–819, 2024.

[2] W. Hou, Q. Luo, X. Wu, Y. Zhou, and G. Si, "Multiobjective optimization of large-scale evs charging path planning and charging pricing strategy for charging station," *Complexity*, vol. 2021, no. 1, p. 8868617, 2021.

[3] S. Yoo, H. Kim, and J. Lee, "Combining reinforcement learning with genetic algorithm for many-to-many route optimization of autonomous vehicles," *IEEE Access*, vol. 12, pp. 26 931–26 942, 2024.

[4] C. Yan, X. Xiang, and C. Wang, "Towards real-time path planning through deep reinforcement learning for a uav in dynamic environments," *Journal of Intelligent & Robotic Systems*, vol. 98, pp. 297–309, 2020.

[5] X. Lin, Y. Yu, and C. Sun, "Supplementary reinforcement learning controller designed for quadrotor uavs," *IEEE Access*, vol. 7, pp. 26 422–26 431, 2019.

[6] Q. Yao, Z. Zheng, L. Qi, H. Yuan, X. Guo, M. Zhao, Z. Liu, and T. Yang, "Path planning method with improved artificial potential field—a reinforcement learning perspective," *IEEE Access*, vol. 8, pp. 135 513–135 523, 2020.

[7] R. Xie, Z. Meng, L. Wang, H. Li, K. Wang, and Z. Wu, "Unmanned aerial vehicle path planning algorithm based on deep reinforcement learning in large-scale and dynamic environments," *IEEE Access*, vol. 9, pp. 24 884–24 900, 2021.

[8] W. Joe and H. C. Lau, "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, 2020, pp. 394–402.

[9] A. Alwabli and I. Kostanic, "Dynamic route optimization for waste collection using genetic algorithm," in *2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA)*. IEEE, 2020, pp. 1–7.

[10] K. Bhargava, R. Gupta, A. Singhal, and A. Shrinivas, "Genetic algorithm to optimize solid waste collection," in *Proceedings of the 3rd International Conference of Recent Trends in Environmental Science and Engineering (RTESE'19)*, 2019, pp. 1–8.

[11] I. Aktemur, K. Erensoy, and E. Kocyigit, "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)*. IEEE, 2020, pp. 1–8.

[12] A. Farahbakhsh and A. S. Kheirkhah, "A new efficient genetic algorithm-taguchi-based approach for multi-period inventory routing problem," *International journal of research in industrial engineering*, vol. 12, no. 4, pp. 397–413, 2023.

[13] B. Meniz and F. Tiryaki, "Genetic algorithm approach to asymmetric capacitated vehicle routing: A case study on bread distribution in istanbul, türkiye," *Decision Science Letters*, vol. 13, no. 3, pp. 605–616, 2024.

[14] H. I. K. Fathurrahman, A. Ma'arif, and L.-Y. Chin, "The development of real-time mobile garbage detection using deep learning," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 7, no. 3, pp. 472–478, 2021.

[15] J. Quevedo, M. Abdelatti, F. Imani, and M. Sodhi, "Using reinforcement learning for tuning genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2021, pp. 1503–1507.

[16] W. Li, L. Wang, and A. Fei, "Minimizing packet expiration loss with path planning in uav-assisted data sensing," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1520–1523, 2019.

[17] J. Kaabi, Y. Harrath, A. Mahjoub, N. Hewahi, and K. Abdulsattar, "A 2-phase approach for planning of hazardous waste collection using an unmanned aerial vehicle," *4OR*, vol. 21, no. 4, pp. 585–608, 2023.

[18] K. Abdulsattar, Y. Harrath, and J. Kaabi, "Ant colony algorithm to solve a drone routing problem for hazardous waste collection," *Arab Journal of Basic and Applied Sciences*, vol. 30, no. 1, pp. 636–649, 2023.

[19] R. Baldacci, P. Toth, and D. Vigo, "Exact algorithms for routing problems under vehicle capacity constraints," *Annals of Operations Research*, vol. 175, pp. 213–245, 2010.

[20] L. Han, F. Gao, B. Zhou, and S. Shen, "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)*. IEEE, 2019, pp. 4423–4430.
[21] R. Shivgan and Z. Dong, "Energy-efficient drone coverage path planning using genetic algorithm," in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2020, pp. 1–6.
[22] P. A. Sarvari, I. A. Ikhelef, S. Faye, and D. Khadraoui, "A dynamic data-driven model for optimizing waste collection," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 1958–1967.
[23] E. Wibisono, I. Martin, and D. N. Prayogo, "Comparison of crossover operators in genetic algorithm for vehicle routing problems," 2021.
[24] L. Girlevicius, "Evaluation of solar powered systems in small scale uav designs," 2022.
[25] I. Seidu, B. Olowu, and S. Olowu, "Advancements in quadcopter development through additive manufacturing: A comprehensive review," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 11, no. 4, pp. 92–124, 2024.
[26] Y. Lu, M. Shen, H. Wang, X. Wang, C. van Rechem, T. Fu, and W. Wei, "Machine learning for synthetic data generation: a review," *arXiv preprint arXiv:2302.04062*, 2023.
[27] M. Ghosal, A. Bobade, and P. Verma, "A quadcopter based environment health monitoring system for smart cities," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2018, pp. 1423–1426.

Shaima Hejres is a Ph.D. candidate at the University of Bahrain. In 2016, she received a Master's degree in Information Technology in Computer Science from the University of Bahrain. She is a Senior Learning Technology Specialist in the Ministry of Education in Bahrain. In addition, she has been a certified trainer since 2006, and a coordinator with Microsoft and the Ministry of Education to provide the latest tools for government schools by preparing several training courses and content in technology in education. She is part of the team that established the distance learning project in public schools in the Kingdom of Bahrain. Moreover, develop various global events for educators and students.

Amine Mahjoub is an assistant professor of Computer Science at the University of Bahrain. He got his Ph.D. in Computer Science from the National Polytechnic Institute, Grenoble, France, in 2004. He worked as an attached teacher in the National Polytechnic Institute of Grenoble and in the Blaise Pascal University of Clermont-Ferrand in France, before being recruited as an assistant professor in 2006 at the University of Sousse, Tunisia. His teaching and research interests include combinatorial optimization problems, parallel computing scheduling problems, complexity theory, and advanced algorithms.

Nabil Hewahi is a professor of Computer Science at the University of Bahrain. He got his Ph.D. in Computer Science from Jawaharlal Nehru University, New Delhi, India, in 1994, M.Tech in Computer Science and Engineering from the Indian Institute of Technology, Bombay, India, in 1991, and a BSc in Computer Science from Al-Fateh University in 1986. Dr. Hewahi has published approximately 100 papers in well-known journals and conferences, and his main research interests include artificial intelligence, machine learning, and knowledge representation.