IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 July 2024, accepted 30 July 2024, date of publication 2 August 2024, date of current version 12 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3438079

RESEARCH ARTICLE

Guided Genetic Algorithm for Solving Capacitated Vehicle Routing Problem With Unmanned-Aerial-Vehicles

ALI NAJM JASIM AND LAMIA CHAARI FOURATI
Digital Research Center of Sfax (CRNS), Laboratory of Signals Systems Artificial Intelligence Networks (SM@RTS), University of Sfax, Sfax 3029, Tunisia
National School of Electronics and Telecommunications of Sfax, Sfax 3029, Tunisia
Corresponding author: Ali Najm Jasim (alinajmm385@gmail.com)

ABSTRACT This study proposes a capacitated vehicle routing problem (CVRP) approach to optimise Vehicle Routing Problem (VRP) and pesticides spraying. The VRP consists of finding the route which covers every point of a certain area of interest. This paper considers a search and pesticides spraying mission, using group of Unmanned Aerial Vehicles (UAVs). In this scenario, the objective is to minimise the total battery consumption level and tank consumption level to not exceed their maximum battery and tank capacities. A hybrid metaheuristic optimisation algorithm is formulated by integrating Genetic Algorithm (GA) with a guided local search algorithm called guided genetic algorithm (GGA). The performance of the proposed GGA algorithm is compared to four single-solution based metaheuristic algorithms (Guided Local Search [GLS], Tabu Search [TS], Simulated Annealing [SA], and Iterated Local Search [ILS]) and two population-based metaheuristics algorithms (GA and Particle Swarm Optimisation [PSO] algorithm). The results revealed that the proposed GGA outperformed the other algorithms in most instances. GGA showed competitive results, closely following the TS's performance across different scenarios. The evaluation of GGA is conducted by analysing its mean, standard deviation, best solution, and worst solution of ten iterations. In addition, Wilcoxon signed-rank test is conducted across a total of 36 instances. The optimisation results and discussion provide confirmation that the proposed GGA method beat the compared algorithms.

INDEX TERMS Capacitated vehicle routing problem, genetic algorithm, guided local search, path planning, UAV.

I. INTRODUCTION

Agriculture is the main source of food globally and is encountering serious difficulties such as rising food demand, food safety, security issues, and the need for environmental protection, preservation of water, and sustainability. The current practices are expected to persist due to the expected worldwide population reaching 9.7 billion by 2050 [1]. Satellites, drones, and human-crewed-aircraft can improve the farming methods and be adependable solution in agriculture sector [2]. Drones, often referred to as Unmanned-Aerial-Vehicles (UAVs), Unmanned-Aircraft-Systems (UAS), and remotely-piloted-aircraft, provide numerous advantages over other remote-sensing technologies. On cloudy days, for instance, drones are capable of transmitting high-resolution and high-quality images [3]. Other advantages are their transfer speed and availability [4]. Drones are simpler to setup and maintain and more cost-effective than aircraft [2]. Therefore, using drones in agriculture consider as an effective solution to address the mentioned challenges [4].

UAVs have revolutionised multiple sectors and tasks by utilising their capability to fly autonomously without a human operator on board. Originally created for military use, UAVs have smoothly moved into civilian sectors [5], [6]. In addition to combat operations, reconnaissance, and surveillance [7], they are employed in agricultural practices [8], drone delivery [9], traffic management [10], medical emergencies [11], tracking operations [12], and smart cities' tasks [13]. UAVs

The associate editor coordinating the review of this manuscript and approving it for publication was Geng-Ming Jiang.

are essential for reaching inaccessible or remote locations and are vital tools in emergency situations like disaster rescue missions [14], [15].

Flight planning and UAV management tasks are closely related and encompass area coverage, recharging and data collection, search operations, routing for multiple locations, route and mission planning, and operational aspects of a drone network [16].

A. COVERAGE PATH PLANNING PROBLEM FOR PRECISION AGRICULTURE

The Coverage Path Planning (CPP) is categorised as a subtopic of motion planning in robotics. It involves the creation of a path for a robot to systematically examine all locations within a specified scenario [17]. The use of UAVs for CPP has been extensively studied in the literature. Researchers have explored numerous techniques for dividing the area of interest and evaluating performance using metrics such as path length, energy consumption, mission completion time, and area overlapping [18]. To address the precision agriculture issues, the task of CPP can be defined in two categories [19]:

(i) CPP_1 involves monitoring field and crop conditions when the influence of the analysed surface is not anticipated.

(ii) CPP_2 pertains to the application of pesticides, phytocides, fertilizers, etc., while considering potential effects on the cultivated fields.

The standard solution for CPP is to use the back-and-forth algorithm [20]. Cabreira et al. [18] found that spiral and waves are ineffective for the CPP_2 task. Since many turns consume battery power faster and negatively affect the quality of spraying.

There is frequently a requirement for algorithms that optimise the route of one or more UAVs for specific applications. Linear or nonlinear programming methods are inadequate for handling large-scale objects and complex objective functions. Artificial intelligence technologies such as machine learning, fuzzy logic, swarm optimisation, hybrid methodologies, and evolutionary programming are employed in such situations [21]. The literature mentions several optimisation techniques such as genetic algorithm (GA), particle swarm optimisation (PSO), ant colony optimisation (ACO), Hybrid Spider Monkey Optimization (HSMO) algorithm [22], Raccoon Family Optimization (RFO) algorithm [23], artificial neural network (ANN), and learning-based methods. Among the options mentioned, GA represents 21% of publications and is considered one of the most widely used flight planning algorithms [24].

The GA simulates the Darwinian concept of natural selection. GA involves the existence of certain potential solutions, commonly referred to as organisms, individuals, or creatures. During the execution of the GA, the population size remains constant, and poorer solutions (individuals) are substituted with better ones by stochastic selection based on the fitness function [25], [26].

Throughout the initial stages of the research, GA was evaluated alongside other algorithms as a potential method for addressing the challenge of UAV path planning. For instance, [27] investigates the disparity in performance between PSO and GA. According to the authors, PSO was able to obtain high-quality solutions at a faster rate. Simultaneously, [28] illustrated the outcomes of the comparison tests conducted in a three-dimensional setting. Parallel GA outperformed PSO in selecting superior trajectories. The cost function included various factors like the length of the route, altitude of the flight, presence of danger zones, power of the UAV, potential collisions, fuel consumption, and trajectory smoothing for UAVs equipped with rigid wings.

Later on, GA were applied to tackle increasingly intricate problems including real-time flight planning [29]. The cost function in such scenario included several factors: the path length, the average path height, the intersection of radar zones, the path exceeding the power and range of the UAV, and paths intersecting with the earth's surface. In order to enhance the initial population of the GA, the authors of [30] utilised ACO. This approach facilitated the acceleration of the solution process in a simulated environment that had several barriers.

In [21], the authors claimed that the GA, along with other methods, is employed to optimise the coverage of a specific area, while considering constraints such as time limitations and path feasibility. Whereas [31] demonstrated the application of GA with the ''Travelling Salesman Problem'' to solve the problem of coverage with avoiding obstacles. The field is partitioned into obstacle-free cells, and back-and-forth motion is executed for covering the whole area within each of these cells. in addition, [32] considered GA as a strategy to reduce the energy consumption of the UAVs and optimise the number of tours performed by them. However, it is important to take into account that this optimisation objective has certain constraints as it does not consider the expenses associated with carrying out CPP.

B. VEHICLE ROUTING PROBLEM

Based on the review conducted by [33] and the analysis of the previous section, it has been found that all the existing UAV's CPP solutions released in recent years are essentially modified versions of either the Travelling Salesman Problem (TSP) or the Vehicle Routing Problem (VRP).

The TSP is a problem that involves finding the optimal solution among a set of possible combinations. The algorithm calculates the most efficient path, which includes all cities in a given list, visiting each city only once and returning to the starting city. The problem's challenge lies in the objective of the travelling salesman to minimise the overall distance covered. In their study, the authors in [34] introduced a novel routing model named the Multiple Travelling Salesman Problem (mTSP) with UAVs. This model incorporates both trucks and UAVs for efficient last-mile delivery. The CPP-TSP problem is addressed in [35] by employing a Fast NN-2Opt based GA to efficiently determine the path for the UAV in order

to rapidly cover all regions. The study [36] introduced a piecewise linear inter-region path method to address the CPP-TSP problem. The technique aims to minimise travel distance while assuring comprehensive coverage of the entire area. It is important to note that the primary focus of this study is on the VRP as it can be formulated as CVRP which consider the capacity of UAVs' tank and battery.

The VRP is a problem that involves optimising combinations and making decisions using integer programming. It provides a general solution for the travelling salesman problem. The VRP aims to optimise the routes for a fleet of vehicles to efficiently deliver goods to a specific group of clients. The goal of the VRP is to minimise the overall cost of the routes. VRP is commonly employed in the scenario where items stored at a central depot are transported to clients who have made orders for these items. For solving the CPP-VRP, four algorithms of the ACO framework are implemented in [37], two versions of the ACO and the Min–Max Ant. The purpose of the algorithms was to minimise the overall energy consumption by focusing on the main non-constant and controllable factor that affects energy consumption in a delivery vehicle (Electric ground vehicles [EGVs] with UAVs), which is the weight of the payload. In their study, [38] examined the CPP-VRP by incorporating a Time Windows variant. In this variant, the agents' capacity is considered as one, and the goal is to supply each client (target region) with many vehicles simultaneously while adhering to the specified time windows. Their objective was to create a strategy to ensure that all locations were adequately covered within specific time frames, while minimising the total distance travelled. Additionally, they aimed to provide a prompt solution by taking into account the constraint that each agent has limited fuel. In order to address the problem at hand, [38] introduced the Simplex VRP Algorithm (SVA), which utilises the technique of clustering the target locations based on their time windows. Subsequently, transportation problems are generated gradually by considering each cluster and the available UAVs. Xie et al. [35] studied the CPP-VRP for rechargeable UAVs with a constraint on mission time. The study focused on scenarios where many trips per round are necessary due to restricted battery capacities. The objective was to determine the optimal quantity of UAVs to be deployed and the flight trajectories that minimise the overall mission cost. Their problem was expressed as a mixed-integer programming (MIP) model with the objective of minimising the total cost of electric charging, UAV usage, and penalties incurred due to mission time limit violations. This optimization model was solved by using the branch-and-bound algorithm.

A cumulative VRP approach was proposed by [39] to optimise humanitarian CPP. They considered conducting a search and rescue operation, employing a fleet of UAVs. The goal was to minimise the total sum of arrival times at all places inside the designated area, hence achieving the search with the least amount of delay. Three iterations of a Parallel Weighted Greedy Randomised Adaptive Search Procedure-Variable Neighbourhood Descent (GRASP-VND) algorithm have been implemented to solve the cumulative VRP for humanitarian CPP. In their research, [40] examined the cumulative CPP-VRP, which was utilised for the distribution of humanitarian assistance following a natural disaster. UAVs were employed to aid trucks in the transportation of packages to clients. A mathematical model was developed to account for battery usage and solved using a heuristic algorithm based on a widely recognised cluster-first, route-second methodology. In the study conducted by [41], a Parallel GA with Variable Neighbourhood Search (PGA-VNS) was designed to address a mixed integer linear programming model aimed at minimising the total losses caused by fire spots. This model considered the constrained flying range and load impact limitations of UAVs. The findings could aid in optimising the timing and routing of fire engines or UAVs to minimise the damages caused by forest fires. In a study conducted by [42], the researchers examined the CPP-VRP for numerous rechargeable heterogeneous UAVs with multiple journeys, taking into account mission length and payload carrying limits. The objective was to ascertain the specific categories and number of UAVs to be deployed, as well as their flight trajectories, in order to minimise monetary cost. This expenditure encompasses the combined costs of recharging energy for each UAV, renting the UAVs, and the penalties incurred for failing to meet the mission time deadline. GA was implemented to resolve the mathematical model in this study.

Recently, [43] solved capacitated VRP with objective to optimize the cost in terms of distance. Hybrid Genetic and Simulated Annealing (HGSA) Algorithm was proposed to optimize the total travelled distance. Vinh et al. [44] created an efficient computational framework for assigning tasks and planning for a diverse group of UAVs participating in a search mission. The UAVs were autonomously assigned to survey the designated region of interest using the Mixed Integer Linear Programming (MILP) technique. The algorithm was used to assign tasks and coordinate the movements of many UAVs as they searched a certain area on a farm. The researchers in [19] examined the CPP-VRP involving a multiple UAVs used to address coverage problems that may arise during monitoring and the execution of agrotechnical treatments. The proposed approach involves utilising GA to strategically plan the flight of a group of UAVs. This planning was done in conjunction with a mobile ground platform, which serves the purpose of recharging and refuelling the UAVs. This approach enables the computation of a flyby to address the problem of covering fields with various shapes and allows for the selection of the most efficient subset of UAVs from a given set of devices.

Therefore, a Guided Genetic Algorithm (GGA) algorithm is proposed to solve the current problem. Following [45], the brief literature gap for current CPP-VRP, and their solution approaches is reported in Table 1.

TABLE 1. Literature of CPP-VRP with solution approaches.

Reference	Facility	CPP-VRP	Model objectives	Solution approach	Application
[37]	EGVs with UAVs	*	Minimising energy consumption	ACO Min–Max Ant	Transportation
[38]	UAVs	*	Time windows, minimising distance, and limited fuel.	SVA	Transportation
[35]	UAVs	*	Minimizing mission cost , battery capacities	Exact method (Branch-and-Bound)	Agriculture
[39]	UAVs	*	Minimising arrival times and minimum latency.	GRASP-VND	Search and Rescue mission in agriculture
[40]	UAVs with trucks	*	Battery consumption	Heuristic algorithm (cluster-first, route-second)	Transportation
[41]	UAVs	*	Minimising losses with limited flight range and load impact constraints.	PGA-VNS	Forest Fire-Fighting
[42]	UAVs	*	Minimising the monetary cost with mission time and payload carrying constraints	GA	Agriculture
[43]	UAVs	*	Distance cost	HGSA	Agriculture in mart cities
[44]	UAVs	*	Minimising UAVs' turns number and coverage rows number	Rotating Calipers Algorithm	Search and coverage Mission in agriculture
[19]	UAVs	*	Recharging cost	GA	Coverage mission in agriculture
This paper	UAVs	*	Minimising the total battery consumption level and tank consumption level to not exceed their maximum battery and tank capacities	GGA	Coverage, path planning and pesticides spraying in agriculture

In addition, many studies have shown that the GGA that can be constructed by incorporating the Guided Local Search (GLS) on top of a specialised GA [16]. The GGA algorithm has been utilised to address many problems, such as permutation flowshop scheduling problems [17], automated crash reproduction [18], multidimensional Knapsack problem [19], analogue circuit optimisation problem [20], multi-objective power distribution system reconfiguration problem [21], larger constraint assembly problem [22], optimising performance of convolutional neural networks, the processor configuration problem [23], the generalised assignment problem [24], and the radio link frequency assignment problem [25], [26].

Despite these incredible efforts, no study has yet proposed the use of GGA to address the capacitated VRP (CVRP), which is considered a theory gap in the existing literature.

It is worth mentioning that the CPP necessitates the use of a discrete algorithm because of the significant variability in the problem space. Combining GLS with GA is strongly suggested for solving these types of issues, since it provides notable benefits. GA has specialised versions that are specifically tailored to efficiently solve discrete issues.

Employing continuous algorithms, such as the hybrid spider monkey optimisation technique or the raccoon family optimisation algorithm, alongside GLS would augment the intricacy of the challenge owing to the fundamentally disparate characteristics of these algorithms. Implementing these continuous methods would necessitate a two-stage methodology, which is not straightforward and may introduce additional computational complexity.

Considering these factors, we chose GA for this study because of its established effectiveness in addressing discrete optimisation problems and its compatibility with GLS.

C. RESEARCH CONTRIBUTIONS

The primary contributions of the research can be briefly summarised as follows:

- The implementation of a CVRP aims to minimise the total battery consumption level and tank consumption level in determining the nodes required pesticide application.
- The CVRP presentation emphasises the reduction of overall battery consumption level and the optimisation of pesticide application capacity by considering specific nodes.
- The CVRP mathematical formulation makes a theoretical contribution to the VRP literature.
- The implementation of a meta-heuristic algorithm, specifically hybrid GA and GLS, to tackle the complexity of this problem.
- This paper reformulates GGA to solve CVRP with UAVs.
- The proposal of an efficient methodology for addressing the CVRP, derived from the findings of the computational tests.

The remaining of the article is structured as follows. In section II, CPP problem is explained. The mathematical representation of the UAVs routing problem is presented in section III. Section IV includes the formulation of the proposed GGA. The experimental results and discussion are explained in section V followed by the conclusion and future work in section VI.

II. CPP PROBLEM

This study focuses on a path planning and search scenario using a group of UAVs that can take off and land vertically. The objective is to have the UAVs cover all points in a specific

region of interest while minimising the overall battery and tank consumption. The following assumptions are made:

- All UAVs are taken off and land from a predetermined location, known as the control centre.
- Each UAV is equipped with a camera or sensor that is positioned to face downwards and has a square field of vision.
- The speed of the UAVs during flight remains constant to enable the camera or sensor to effectively observe the desired features on the ground.
- The maximum battery capacity and tank capacity of each UAV are finite and known in advance.
- The UAVs are not affected by any external forces, such as weather conditions.

Recently, [46] evaluated UAVs to identify the best one based on their endurance, payload, and dimensions. The authors found that DJI Agras MG-1 was the best among 6 UAVs under the 8 routers category. It is specifically designed for agricultural applications, including pesticide spraying and aerial monitoring. The battery of the DJI Agras MG-1 is sufficient to spray one tank of liquid. This battery has run out within a flying time of 8-12 minutes. The DJI Agras MG-1 is designed for crop management and has a liquid loading capacity of up to 10 kilogrammes. It can reach a maximum speed of 8 metres per second. The experiments focus on the features of the DJI Agras MG-1. The idea is that UAVs are used to collect data and determine the specific areas that need to be sprayed with pesticide.

Task assignment is integral components of CPP, involving the exploration or coverage of specific areas of interest. The CPP with multiple UAVs, can be transformed into a VRP, by construction a graph that coverage is reached when a set of nodes is visited at least once by a UAV. The decisions to be made when formulating the CPP to a VRP pertain to the process of constructing the graph, the way coverage is obtained, the constraints imposed by the vehicles and the objective function to be minimized [47]. The capacitated VRP (CVRP) [48] is NP-hard as it is an extension of the classical VRP. Its objective function is formulated in a way which minimises the sum of arrival times at the nodes instead of the total routing cost in the classical VRP.

The aim of this study is to minimise the total battery consumption level and tank consumption level of UAVs by determining the optimal routes to apply pesticides. The process involves the UAV departing from control centre, collecting data within the target area, identifying specific locations requiring pesticide application, and then returning to the control centre for recharging. Then, one or more UAVs start a route based on the collected data for spraying pesticides. The challenge of the first route is the minimisation of the total battery consumption level, whereas the challenges of the second route are the minimisation of the total battery consumption level and tank consumption level of pesticides. The motivation of this research is to provide a valuable contribution to solve VRP based UAVs.

Therefore, a metaheuristic algorithm is proposed in this study to determine the best rout of each UAV during each trip. The battery and tank consumption levels of UAV are minimized using a meta-heuristic based hybrid GA with a GLS. The proposed solution manages better the usage of the UAV's charging and tang of pesticides by eliminating many unneeded returns to the control centre and therefore minimizes the total battery consumption level and tank consumption level of pesticides.

As the UAVs dealing with capacity of tang and the capacity of battery, the topic being addressed is strongly connected to the CVRP. Nevertheless, the capacity of the UAVs to carry pesticides and the requirement for frequent recharging add complexity to the problem being studied, beyond that of the CVRP. As far as we know, no previous research has investigated our problem within the specified constraints, which considered a research gap.

III. MATHEMATICAL REPRESENTATION OF THE UAVS ROUTING PROBLEM

Let the node set N, which consists of the nodes 0, 1, …, n, and n+1. Node 0 represents the initial node, while node n+1 represents the last node. The set of edges E connects every pair of nodes. The problem is defined on an undirected graph $G = (N, E)$, where each edge (i, j) is associated with a battery consumption u_{ij} and tank consumption z_{ij}. The set of nodes to be visited, denoted as N', is defined as $N' = N \setminus \{0\}$. Each node $i \in N'$ must be visited exactly once by one of the UAVs. The number of UAVs is set to $R \geq 1$, while the maximum battery and tank capacities of each UAV are C_b, and C_t, respectively. Variables b_i^k and t_i^k denote the battery and tank consumption levels of UAV k at node i. The x_{ij}^k is a binary variable, where $x_{ij}^k = 1$ if UAV k traverses edge (i, j) from node i to node j, and $x_{ij}^k = 0$ otherwise. The objective of the CVRP, given in Eq. (1), minimizes the total battery consumption level and tank consumption level while ensuring that each route starts and terminates at node 0, and the total battery and tank consumption do not exceed C_b, and C_t, respectively. The mathematical formulation of the CVRP model can be expressed as follows:

$$\text{minimize} \sum_{k=1}^{R} \sum_{i \in N'} \left(b_i^k + t_i^k \right) \tag{1}$$

$$\text{s.t.} \sum_{j \in N} x_{ji}^k = \sum_{j \in N} x_{ij}^k, \quad \forall i \in N', \tag{2}$$

$$\sum_{k=1}^{R} \sum_{j \in N} x_{ij}^k = 1, \quad \forall i \in N' \tag{3}$$

$$\sum_{j \in N} x_{0j}^k = 1, \tag{4}$$

$$\sum_{j \in N} x_{0j}^k = 1, \tag{5}$$

$$\sum_{i \in N} \sum_{j \in N} x_{ij}^k u_{ij} \leq C_b, \tag{6}$$

$$\sum_{i \in N} \sum_{j \in N} x_{ij}^k z_{ij} \leq C_t, \tag{7}$$

$$b_i^k + u_{ij} - \left(1 - x_{ij}^k\right) G \leq b_j^k, \tag{8}$$

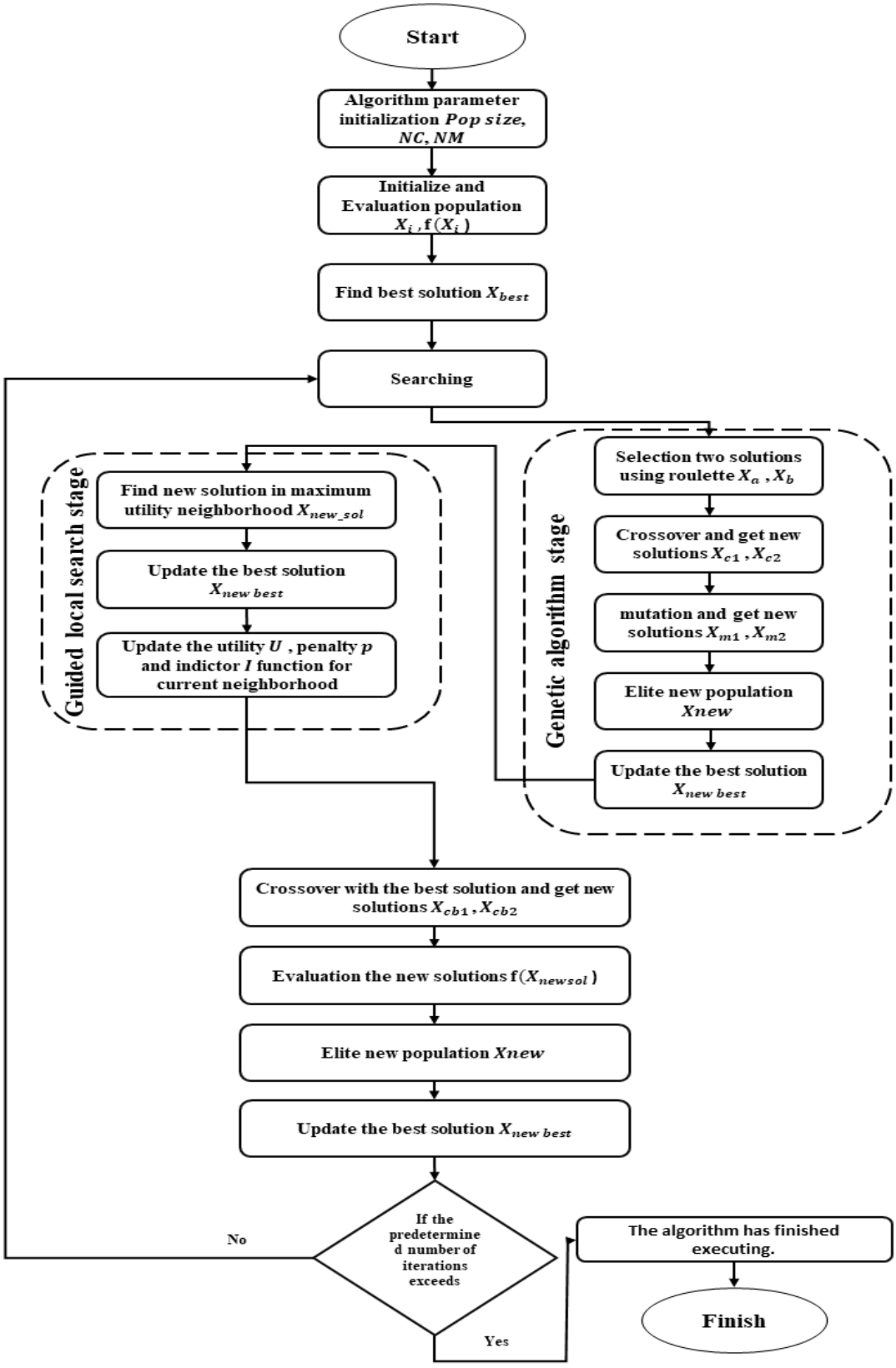

FIGURE 1. Flowchart of the proposed GGA.

$$t_i^k + z_{ij} - \left(1 - x_{ij}^k\right) G \leq t_j^k, \tag{9}$$

$$b_i^k, t_i^k \geq 0, \quad \forall i \in V, \tag{10}$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i \in N, \ \forall j \in N, i \neq j, \tag{11}$$

where

$$\forall i \in N \setminus \{n+1\}, \quad \forall j \in N, \ \forall k \in \{1, \ldots, R\}$$

In the above constraint (2), a UAV arriving at node i must depart from it. Constraint (3) ensures that the UAV visits each node only once. Constraints (4)-(5) ensure that a route must start and end at node 0, which represents the UAV control centre. Constraints (6)-(7) ensure that the total battery and tank consumption of each UAV must not exceed their maximum battery capacity C_b and tank capacity C_t, respectively. Constraints (8)-(9) calculate the battery and tank consumption at the nodes. By utilising a large positive constant G, the formation of sub-tours is prevented. The formulation of the constraints is derived from the formulas commonly used for VRPs with time windows. Constraints (10) ensures the values battery and tank consumption are non-negative. Constraint (11) restrict the values of x_{ij}^k to binary.

The set of N' vertices in the CVRP application on the CPP is obtained by the cellular approximation method. Each UAV route starts and terminates at node 0, and this node can be determined apart from the grid formation. The battery consumption level b_i^k and tank consumption level t_i^k between nodes i and j are determined as the Euclidean distance between them.

IV. THE PROPOSED GGA

This section focuses on the solution techniques for the model specified in equations (1) to (11). The proposed GGA is given in Figure 1. This figure shows the sequential stages of the GA algorithm. The initial step involves determining the algorithm's parameters, such as the probability of crossover and mutation, the population size, and generating the neighbourhood structure for the problem. Then the subsequent step involves creating the initial population and evaluating the objective function for each solution within the population. Afterword executing each of the GA's steps to generate a new population. This involves the selection of new solutions, followed by the implementation of the crossover operation, and finally, performing of the mutation process. The next step involves selecting the new population by an elite process, wherein the new solutions are merged with the existing population. The selection of the new population is based on the value of the objective function.

The second section outlines the procedures of the GLS algorithm. This algorithm aims to improve the best solution identified within the GA population. In the first stage, the solution is improved by incorporating the best neighbour discovered in the GLS algorithm. Subsequently, the utility and penalty function are updated based on the solution's quality. The quality of the solutions for each neighbourhood. In the second solution, the GA's population is combined with the new solution generated by the GLS algorithm. This combined population is then used to select a new population for subsequent iterations. The aforementioned steps are repeated till the end of the predetermined number of iterations.

Initial experimental findings indicate that the mixed integer programming (MIP) model poses significant challenges in solving, even for very minor occurrences. As an example, the process of calculating an instance with 20 nodes and 5 UAV exceeds 3600 seconds. Therefore, we use meta-heuristic methods to acquire answers that are close to optimum within an acceptable timeframe. We present hybrid GA with a guided local search algorithm called guided genetic algorithm (GGA) to address the issue. The GGA utilises the first-in-first-out (FIFO) node sequence as a baseline to construct UAV's routes.

The FIFO based GGA operates in a manner similar to a generic GA. Let T represent the population size and M represents the maximum number of multiplication iterations. In the context of the UAV delivery challenge, we examine a scenario with N nodes and O UAVs. The chromosomes undergo random initialization using a permutation process, starting from $1 to N$ and continuing until the population size T is attained. Here, we analyse each individual drone individually, rather than focusing on various categories of drones. For every individual UAV a, $a = 1, 2, \ldots, O$, we use $time_a$ to denote the overall duration required for UAV a to complete its delivery mission and return to the control center.

To convert chromosomes into UAV routes, we use the First-In, First-Out (FIFO) principle to allocate nodes. on this context, we consider the chromosome as a queue of nodes $(O_1, O_2, \ldots, O_N)$ and a queue of drones $(d_1, d_2, \ldots, d_O)$, where each node or UAV is now represented by a gene on the chromosome.

Subsequently, the nodes comprising the node queue are allocated to the first UAV inside the UAV queue. If the weight of the node surpasses the residual load capacity of the first available UAV, or if the battery's flight range is insufficient to allow the UAV's completion of this tour, we allocate the package to the subsequent UAV in the queue. The preceding UAV will proceed to visit its nodes in the designated sequence and thereafter return to the control center. In the event that the last UAV in the queue is unable to successfully load the remaining packages, the queue is reset to its initial state, and the remaining items are allocated to the first UAV as its subsequent delivery assignment after its first trip. In this approach, the order in which UAVs return to the control center is disregarded, and the node assignment is solely determined based on the sequence of the UAV queue.

Figure 2 depicts a demonstrative instance of the procedure. Before the delivery, as seen in Figure 2, the nodes are allocated to the first UAV in the order of the nodes list. Regarding node 3, the charge of its cargo above the remaining charge level of $\boldsymbol{UAV_1}$, or the distance of the journey exceeds the flying range of $\boldsymbol{UAV_1}$. Consequently, the first journey of $\boldsymbol{UAV_1}$ will begin at coordinates control center $\rightarrow 4 \rightarrow 12 \rightarrow 6 \rightarrow$ control center. Subsequently, the subsequent packages

will be disseminated, starting with $\boldsymbol{UAV_2}$ and the subsequent UAVs. Nevertheless, the weight of node 3's cargo surpasses the flying capability of $\boldsymbol{UAV_2}$, preventing it from travelling to node 3 and returning to the hub due to its limited flight range. Consequently, the $\boldsymbol{UAV_2}$ has the responsibility of delivering this cargo to other nodes in a sequential manner until the remaining capacity becomes insufficient or reaches its maximum flying range.

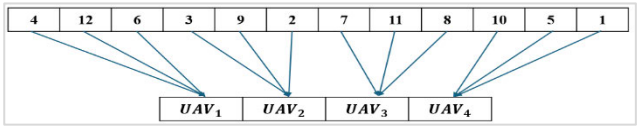

FIGURE 2. UAV allocation based on FIFO.

This study used a hybrid method that combines the genetic algorithm with the guided local search algorithm to address the issue at hand. The concept of the memetic algorithm was embraced as a means of integrating two algorithms, enabling the enhancement of the optimal solution within the population via the utilisation of the guided local search algorithm. In this approach, the genetic algorithm's operations are executed in the first stage to address the issue at hand. Subsequently, the guided local search is used to enhance the optimal solution acquired inside the neighborhood throughout each iteration. Presented below are the sequential instructions for executing the proposed algorithm in order to solve the problem:

1) ENCODING AND DECODING

In the present work, the chromosomal coding methodology used involves the partitioning of genes into distinct sections, denoted as $Chrom = \{\text{Chr}_1, \ldots, \text{Chr}_n\}$ where n represents the total number of regions separated. The composition of each gene consists of all customers inside a certain subregion. Each gene, denoted as $\text{Chr}_i\{i = 1, \ldots, n\}$, reflects the outcome of path-planning for each respective subregion. The genetic sequence Chr_i is represented as a collection of numbers, including the distribution sequence comprising the nodes of the given area.

Figure 3 illustrates a chromosome with 12 nodes and a total of four tours. This chromosome may be further split into four sub-regions. Chr_1 comprises three distinct customer sites, namely 4, 12, and 6. Within this chromosome, the sequence of customer access follows a sequential pattern, commencing with the control center and proceeding to visit node 4, followed by node 12, and concluding with a visit to node 6, before returning to the control center.

2) GENETIC ALGORITHM STAGE

The execution of the genetic algorithm is contingent upon three fundamental steps, namely selection, crossover, and mutation. The solutions included in the crossover stage were chosen using a roulette wheel selection method during the selection step. This approach computes the likelihood of each solution's contribution to the population, and subsequently,

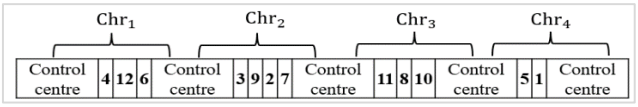

FIGURE 3. Genetic algorithm chromosome consists of four routes and twelve nodes.

two solutions are randomly chosen from the population. Solutions with a high potential for contribution are more likely to be chosen.

During the crossover step, the two selected solutions from the selection stage are randomly divided into two halves. Subsequently, two new solutions are generated by replacing each piece of each solution with its corresponding section, In Figure 4, the crossover phase of two solutions is shown, with each solution consisting of 10 nodes, this study will use a two-stage crossover methodology. The first stage involves random crossing, as previously elucidated. The second stage is performing the crossover procedure using the best solution identified within the population.

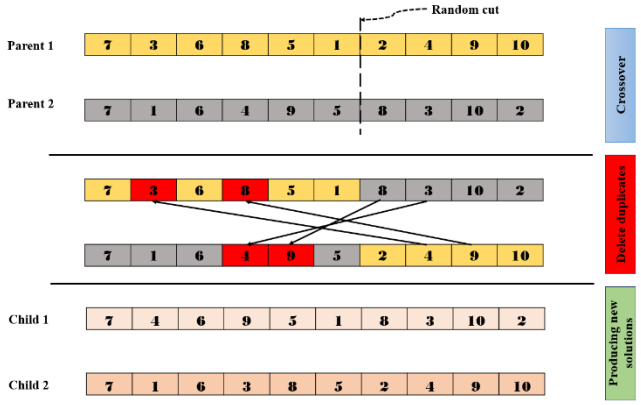

FIGURE 4. Genetic crossover of GA (ten nodes).

During the mutation phase, the process will be executed according to three distinct procedures. In each cycle, a random selection is made for each process. This research will use the approach of swop, Reversion, and Insertion. The swop technique involves the random selection of two points from a given solution, followed by the execution of a swap operation between them. The Reversion procedure involves selecting a random part inside the solution and then inverting it within the solution. In the process of insertion, a randomly selected piece is picked and then put into a different location inside the solution. The mutation processes using the three approaches are shown in Figure 5.

3) GUIDED LOCAL SEARCH STAGE

In the directed local search algorithm stage, the algorithm will be relied upon to improve the best solution found in the population of the genetic algorithm. This stage helps in improving the performance of the genetic algorithm by integrating the solution obtained in this algorithm with the solutions of the

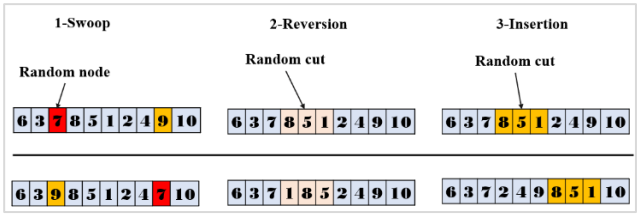

FIGURE 5. Genetic mutation of GA in three strategies.

genetic algorithm in the crossing stage with the best solution. The operation of this algorithm occurs through the following stages:

a: NEIGHBORHOOD STRUCTURE

Based on one of the local search approaches, all potential neighborhoods are delineated inside each solution at this juncture. The local search techniques in this research will use the swop approach. Consequently, the local search will include all conceivable points. The size of the solution is assumed to be equal to n, which represents the total number of feasible neighborhoods. The value is equivalent to $(n(n-1)/2)$. The technique of generating the neighborhood matrix is shown in Figure 6.

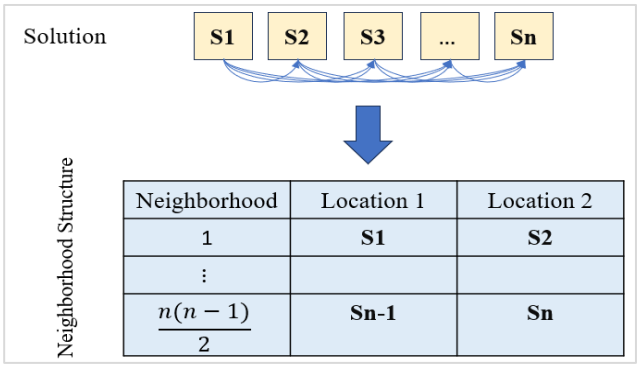

FIGURE 6. Neighborhood structure for the swop method.

b: PROCEDURES

The application of this technique is based on the maximise of utility derived from each search operation conducted inside the neighbourhood structure. Put simply, the neighbourhoods that exert a more significant influence on the solution quality are granted a higher advantage in order to enhance their utilisation in the process of improvement. This advantage is determined by the following equation:

$$\text{util}\,(\text{s}, f_i) = I_i\,(s) \cdot \frac{c_i}{1+p_i} \tag{12}$$

where

c_i: the objective function of solution s after use feature f_i

pi: penalty function

$$I_i\,(s) = \begin{cases} 1 & \text{solution s has property i;} \\ 0 & otherwise \end{cases}$$

The detailed GGA is shown in Algorithm 1.

Algorithm 1 Guided Genetic Algorithm GGA

Input:
Pop size : population size *NC*:number of Crossover *NM*: number of mutations
max *iter*: maximum number of iterations
$N_k = \{N_1, N_2 \cdots, N_m\}$: neighborhood structure
N_m is the mth neighborhood structure
$pen_k = \{pen_1, pen_2 \cdots, pen_m\}$ // penalty function for each neighborhood
$I_k = \{I_1, I_2 \cdots, I_m\}$ // Indicator function for each neighborhood
$uit_k = \{uit_1, uit_2 \cdots, uit_m\}$ // Uitility function for each neighborhood

//Genetic algorithm stage

1: Initialize pop; Generate initialized populations
2: **for** *gen* = 1 to *max iter* **do**
3: **for** *i* = 1 to *pop size* **do**
4: calculate the fitness value for each solution
5: **end for**

found the best solution in the population *best_sol*
// Crossover
Pop_C = [];

6: **for** *i* = 1 to *NC*/2 **do**
7: select two solutions s1 & s2 using roulette strategies
8: crossover and get two new solutions ns1 & ns2;
9: Pop_C = [Pop_C+ns1 & ns2]
10: **end for**

//Mutation
Pop_M = [];

11: **for** *i* = 1 to *NM* **do**
12: select a random solution from the population
13: mutation using swop and get a new solutions ns1;
14: Pop_M = [Pop_M+ns1]
15: **end for**
16: select new population from [pop+ Pop_C+ Pop_M]
17: find the best solution in population *best_pop*

//Guided local search algorithm stage

18: **for** k = 1 to *m* **do**
19: **if** $uit_k = \max(uit)$ **then**
20: $new\_best\_solution = local\_search(best\_sol, N_k)$
21: calculate the fitness value for the new solution
22: **if** the fitnessofnew solution best of fitness *best_pop*
23: $best_{pop}$ = new solution
24: $pen_k = pen_k + 1$ //update penalty function
25: **Else**
26: $I_k = 1$
27: **end if**
28: **end if**
29: $uit_k = I_k \cdot \frac{c_k}{1+p_k}$ //update utility function
30: **end for**

// Crossover with the best solution
Pop_C_best = [];

31: **for** *i* = 1 to *NC*/2 **do**
32: select random solutions s1 using roulette strategies
33: crossover with the best solution and get new solutions ns1;
34: Pop_C_ best = [Pop_C_ best +ns1]
35: **end for**
36: select new population from [pop + Pop_C_ best]
37: **repeat**
38: **end for**
39: Best solution

V. COMPUTATIONAL EXPERIMENT

In this section, simulation studies are conducted to evaluate the performance of proposed GGA algorithm. Firstly, the evaluation methodology is discussed and the results of the experiments are presented. Secondly, the performance of the GGA algorithm is compared with six benchmark techniques. Thirdly, the findings are discussed in detail.

A. EXPERIMENTAL SETTINGS AND COMPARATIVE METHODS

All methods are implemented using MATLAB R2023a, and the experiments are conducted on a Windows 10 Pro (64bit) operating system with an Intel Core i7-11375H, CPU running at 3.30 GHz, 32GB RAM and 500GB storage. The number of nodes is set to 25, 36, 49, 64, 81, 100, 121, 144, 169, 225, 256, with the number of UAVs varying between 1 UAV to 6 UAVs depending on the number of nodes. The distance between nodes is assumed to be uniform and the path of UAVs are expected to be square, except when the UAVs need to return to the control centre. The UAVs return to the control centre in three situations: when the battery needs to be recharged, when the tank needs to be refilled with pesticides, and when the tour is completed.

The parameter settings (parameter tuning) for the experiment are given in Table 2. The performance of GGA is substantially influenced by the crossover rate, mutation rate, number of chromosomes in the population, and maximum number of population iterations. We systematically vary the crossover and mutation rates at values of 0.2, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9. We then execute the GGA ten times and measure the average objective value for each combination of crossover and mutation rates. The optimal outcome is obtained when the crossover rate is set to 0.4 and the mutation rate is set to 0.8. Thus, we set the crossover rate to 0.4 and the mutation rate to 0.8 in the subsequent experiments. In the experiments, the performance of proposed GGA algorithm is compared to four single-solution based metaheuristic algorithms and two population-based metaheuristics algorithms:

Guided Local Search (GLS) [49] is a metaheuristic approach that was developed to address combinatorial optimisation problems. GLS utilises a penalty-based method to effectively engage with the local improvement procedure. such a mechanism can escape local optima, hence enhancing the efficiency and robustness of the classical local search algorithms.

Tabu Search (TS) [50] is an algorithm that iteratively searches across different neighbourhoods, with the neighbourhoods changing dynamically. TS improves the effectiveness of local search by excluding nodes that have already been visited. To do so, loops in the search space can be prevented and it becomes possible to avoid being stuck in local optima.

Simulated Annealing (SA) [51] is a common metaheuristic approach for optimising problems characterised by large search spaces. The search process in SA begins with an initial solution and iteratively decreases a control parameter until it converges to the best solution. The main benefit of SA is its ability to avoid being stuck in local minima and instead converge towards a global minimum.

Iterated Local Search (ILS) [52] is a popular metaheuristic approach that is based on a single solution. ILS utilised by many academic due to its ability to effectively handle complex optimisation problems. The success of ILS can be attributed to the possession of several desired attributes commonly seen in metaheuristics, including simplicity, accuracy, flexibility and speed.

Genetic Algorithm (GA) [53] is a population-based technique. It is a metaheuristic approach that utilises population characteristics to direct the search. GA manages and enhances many solutions that can yield a high-quality solution to the optimisation problems.

Particle Swarm Optimisation (PSO) algorithm [54] is inspired by the collective behaviour observed in swarms of particles (animals and insects), such as birds and fish. In the PSO, those particles are used to guide the search. To identify the food regions, particles rely on their personal memories of the event, as well as the information shared by their group.

TABLE 2. Parameter settings for the experiment.

Parameters	Value
Limitation of x axis and y axis (square dimension)	0-100
Node reequipment for charge or fertilization	10-25
Number of nodes	25, 36, 49, 64, 81, 100, 121, 144, 169, 225, 256
Number of UAVs	1, 2, 3, 4, 5, 6
UAVs' type	1
Each battery covers	6,000-10,000 m^2
Maximum tank capacity for each UAV	10 L
Binary variable indicating UAV k traverses edge (i, j)	$x_{ij}^k = 1$ or $x_{ij}^k = 0$
Maximum number of population iterations	500
Numbers of chromosomes in population	60
The crossover rate	0.4
The mutation rate	0.8

B. EVALUATION PROCEDURE

The experimental findings are shown using the mean, standard deviation, minimum value (best solution), maximum value (worst solution), and the Wilcoxon signed-rank test.

- **Mean** ($\bar{x}$) represents the average value of a number of runs. It is calculated by summing up all the ran values and then dividing by the total number of runs.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{13}$$

where

x_i represents each run of the algorithm.
n represents the total number of runs.

- **Standard *Deviation (STD)*** measures the dispersion or spread of a number of runs from the mean. It indicates

how much individual run deviate from the average.

$$STD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{14}$$

where

x_i represents each run of the algorithm.
n represents the total number of runs.
$\bar{x}$ represents the mean value.

- **Best Solution (Min)**: the best solution is the minimum value achieved by the optimisation algorithm, such as the lowest objective function value attained by the compared algorithms.
- **Worst Solution (Max)**: the worst solution is the maximum value among the solutions obtained by the optimisation algorithm, representing the least desirable outcome.
- **Wilcoxon signed-rank test**: it is a statistical method that measures the disparity between two sets of data [55]. It offers an alternative way to assess the position of the data, taking into account both the size and direction of the disparities. The test examines the following hypotheses:

$$H_0 : mean\,(Algorithm_1) = mean\,(Algorithm_2),$$
$$H_1 : mean\,(Algorithm_1) \neq mean\,(Algorithm_2),$$

where

$Algorithm_1$ and $Algorithm_2$ represent the outcomes of two algorithms.

The statistical test also serves to assess the relative performance between two algorithms. In this context, let d_i represent the disparity in performance scores between two algorithms in solving the i^{th} out of n problems. The R^+ is defined to represent the total ranks for the problems in which $Algorithm_1$ performs better than $Algorithm_2$ and R^- to represent the total ranks for the problems in which $Algorithm_2$ performs better than $Algorithm_1$. It is worth noting that the ranks of $d_i = 0$ are evenly divided among these sums. In cases where these sums contain an odd number of elements, one of them is excluded from consideration.

$$R^+ = \sum_{d_i>0} \text{rank}\,(d_i) + \frac{1}{2}\sum_{d_i=0} \text{rank}\,(d_i) \tag{15}$$

$$R^- = \sum_{d_i<0} \text{rank}\,(d_i) + \frac{1}{2}\sum_{d_i=0} \text{rank}\,(d_i) \tag{16}$$

In the experiment, MATLAB serves as the tool for computing the $p - value$ to compare the algorithms at a predetermined significance level of $\alpha = 0.05$. The null hypothesis is invalidated when the calculated $p - value$ falls below the specified significance level. Notably, R^+ denotes a high-performing algorithm with a superior mean value compared to other algorithms across various experimental scenarios. Specifically, when $\left(R^+ = \frac{n\times(n+1)}{2}\right)$, it indicates that this algorithm beats all others algorithms across all experimental settings.

C. EXPERIMENTAL RESULTS AND DISCUSSION

The experimental results are presented based on the number of nodes and UAVs in Tables 2, 3, 4, and 5. The Mean, Min, and Max represent the average, minimum, and maximum values of the 10 runs of each algorithm. STD represents the results of the standard deviation. In Table 2, the number of nodes is 25, 36, and 49. The number of UAVs with these nodes were one, two, and three.

In the experimental scenario where 25 nodes and 1 UAV were utilised, the GA yielded the most optimal solution, achieving a minimum value of 508.60, as illustrated in Table 3. Subsequently, the proposed GGA closely followed with a minimum value of 513.93. Over 10 iterations, the GA demonstrated superior performance compared to all other algorithms, achieving a mean value of 511.26. Following the GA in terms of performance were the GGA, SA, PSO, TS, GLS, and ILS. When 25 nodes and 2 UAVs were used, GLS emerged as the top performer, achieving the most optimal solution with a minimum value of 298.26. Followed by the proposed GGA and SA, both of which resulted in a minimum value of 312.79. In addition, GLS consistently outperformed the other algorithms across 10 iterations, demonstrating a mean value of 316.05. The subsequent algorithms that were evaluated in terms of performance were ILS, SA, TS, GGA, GA, and PSO. By utilising 3 UAVs with 25 nodes, the GGA algorithm obtained the most optimal solution, achieving a minimum value of 236.17. The GGA consistently surpassed all other algorithms, with 10 iterations mean value of 240.84. The GA, SA, GLS, TS, PSO, and ILS algorithms performed well, following the GGA in terms of performance.

When the number of nodes is increased to 36, the GA was the best algorithm, followed by the proposed GGA, achieving superior performance across all experiments. For 10 iterations, GA algorithm demonstrated mean values of 610.13 with single UAV and 360.1581 with two UAVs. Followed by GGA, GLS, TS, SA, ILS, and PSO algorithms with single UAV, and the GGA, SA, TS, GLS, ILS, and PSO algorithms with two UAVs. By utilising 3 UAVs, the GGA maintained its superiority, with a minimum value of 276.75, while the GA had a mean value of 284.74 across 10 iterations, outperforming all other algorithms. Following the GA in terms of performance were the GGA, GLS, SA, TS, PSO, and ILS algorithms. The graphical representation in Figure 20 in Appendix illustrates the performance of the algorithms using 1, 2, and 3 UAVs with 36 nodes.

Upon further increase to 49 nodes, the proposed GGA emerged as the top performer, surpassing all algorithms in terms of both the most optimal solution and mean values across 10 iterations, regardless of the number of UAVs employed. The performance of the algorithms using 3 UAVs with 25, 36 and 49 nodes is visually depicted in Figures 7, 20 and 21 in Appendix.

In Table 4, the number of nodes is 64, 81, 100, and 121. The number of UAVs with these nodes were, two, three, and four. The analysis of the best solution (min values) revealed that GGA consistently demonstrated competitive performance,

TABLE 3. Experimental results with 25, 36, and 49 nodes.

Number of Nodes	Number of UAVs		GGA	GLS	TS	SA	ILS	GA	PSO
25	1	Mean	513.93	544.98	540.22	524.61	552.51	**511.26**	539.48
		STD	0.00	20.48	21.28	15.11	3.70	3.77	6.72
		Min	513.93	530.50	525.17	513.93	549.89	**508.60**	534.73
		Max	513.93	559.45	555.26	535.30	555.13	513.93	544.23
	2	Mean	324.70	**316.05**	321.93	316.25	308.24	325.29	378.09
		STD	16.84	25.15	1.41	5.49	11.60	4.91	23.62
		Min	312.79	**298.26**	320.94	312.36	300.04	321.81	361.39
		Max	336.61	333.83	322.93	320.13	316.44	328.76	394.79
	3	Mean	**240.84**	259.60	259.87	248.58	270.73	244.25	265.07
		STD	6.60	28.43	6.71	7.61	14.46	2.08	15.10
		Min	**236.17**	239.50	255.13	243.20	260.50	242.78	254.40
		Max	245.50	279.71	264.62	253.96	280.95	245.73	275.75
36	1	Mean	610.61	640.31	661.47	670.27	678.88	**610.13**	801.12
		STD	4.55	14.57	3.98	26.50	39.87	5.22	48.16
		Min	607.38	630.01	658.65	651.53	650.69	**606.44**	767.07
		Max	613.83	650.61	664.28	689.01	707.07	613.83	835.18
	2	Mean	363.90	402.06	388.30	381.63	415.69	**360.16**	468.50
		STD	5.21	12.57	16.31	17.34	13.65	10.33	45.63
		Min	360.21	393.17	376.77	369.36	406.03	**352.85**	436.23
		Max	367.58	410.95	399.83	393.89	425.34	367.46	500.76
	3	Mean	289.96	300.29	308.57	301.20	340.81	**284.74**	338.23
		STD	18.68	8.03	12.61	21.58	3.13	1.10	8.04
		Min	**276.75**	294.61	299.65	285.94	338.60	283.96	332.55
		Max	303.16	305.97	317.49	316.46	343.02	285.52	343.91
49	1	Mean	**711.84**	759.02	763.51	763.60	898.63	718.34	1105.15
		STD	8.37	20.16	11.34	34.92	103.17	0.83	2.09
		Min	**705.92**	744.77	755.49	738.90	825.68	717.76	1103.68
		Max	717.76	773.28	771.53	788.29	971.59	718.93	1106.63
	2	Mean	**414.35**	503.91	472.83	456.94	475.07	452.14	648.96
		STD	6.17	43.57	3.63	14.72	31.04	47.27	6.88
		Min	**409.99**	473.10	470.26	446.54	453.12	418.71	644.09
		Max	418.71	534.71	475.39	467.35	497.02	485.56	653.82
	3	Mean	**316.50**	353.24	347.56	351.81	384.20	333.67	446.29
		STD	2.38	1.60	11.39	4.87	43.34	3.80	42.52
		Min	**314.82**	352.11	339.51	348.37	353.55	330.99	416.23
		Max	318.18	354.37	355.61	355.26	414.85	336.36	476.36

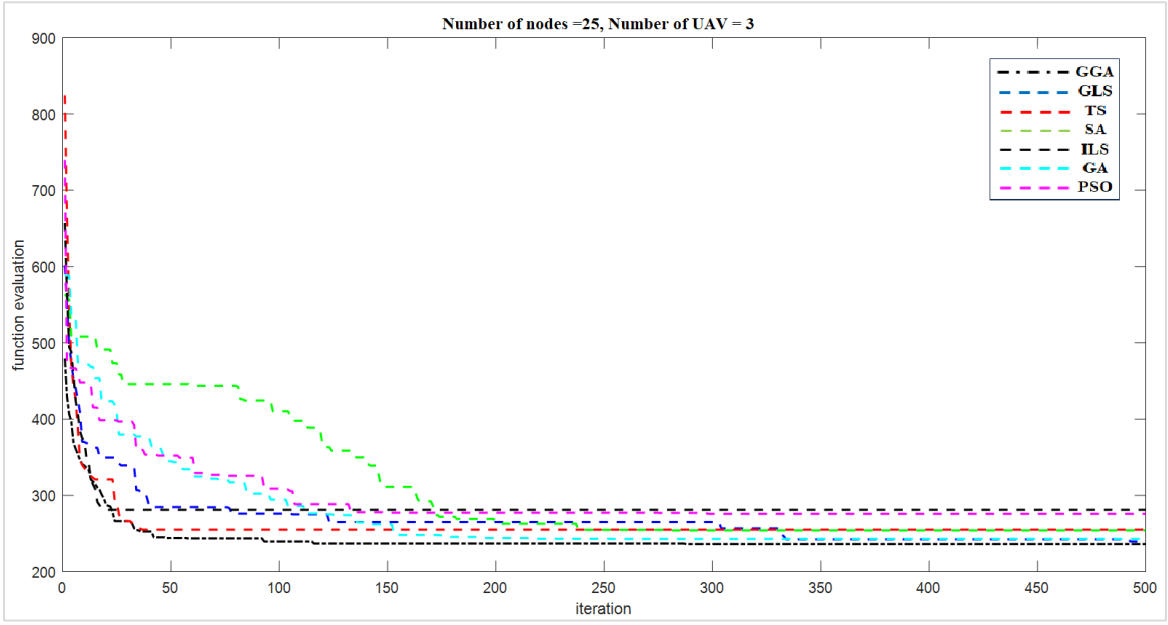

FIGURE 7. Algorithms' performance with 25 nodes and 3 UAVs.

with minimum values ranging from 509 to 703 across different settings of nodes and UAVs. Furthermore, the mean value of GGA across different scenarios provided valuable insights into its overall effectiveness. The analysis of mean

TABLE 4. Experimental results with 64, 81, 100, and 121 nodes.

Number of Nodes	Number of UAVs		GGA	GLS	TS	SA	ILS	GA	PSO
64	2	Mean	**514.58**	566.77	530.49	537.31	567.01	534.69	805.60
		STD	6.92	24.40	26.48	29.57	0.38	7.07	40.33
		Min	**509.68**	549.51	511.76	516.40	566.74	529.70	777.08
		Max	519.47	584.02	549.22	558.23	567.28	539.69	834.11
	3	Mean	**368.81**	415.90	426.62	417.05	454.23	422.32	624.83
		STD	1.17	11.37	2.83	0.84	3.47	0.50	38.69
		Min	**367.99**	407.86	424.63	416.45	451.78	421.97	597.47
		Max	369.64	423.95	428.62	417.64	456.69	422.68	652.19
	4	Mean	**327.84**	349.38	358.92	381.06	406.38	388.28	553.85
		STD	0.77	5.11	13.97	5.06	43.30	19.26	46.82
		Min	**327.29**	345.77	349.04	377.48	375.77	374.66	520.74
		Max	328.39	352.99	368.80	384.64	437.00	401.90	586.96
81	2	Mean	**551.45**	621.21	598.49	635.97	757.72	653.46	1017.44
		STD	8.24	9.95	8.06	2.32	105.90	63.83	55.55
		Min	**545.62**	614.17	592.79	634.33	682.84	608.33	978.17
		Max	557.27	628.25	604.18	637.61	832.61	698.59	1056.72
	3	Mean	**426.73**	516.77	470.04	503.53	484.68	510.27	814.86
		STD	9.82	29.50	5.09	27.87	25.58	56.44	38.72
		Min	**419.79**	495.91	466.44	483.82	466.60	470.36	787.49
		Max	433.67	537.63	473.63	523.24	502.77	550.18	842.24
	4	Mean	**362.53**	424.83	375.12	421.75	463.31	463.87	694.31
		STD	2.11	3.10	12.30	0.81	4.23	31.77	1.17
		Min	**361.03**	422.64	366.43	421.18	460.32	441.41	693.49
		Max	364.02	427.02	383.82	422.32	466.30	486.34	695.14
100	2	Mean	**636.61**	730.53	698.95	805.59	836.75	788.29	1373.48
		STD	27.05	15.44	29.83	10.49	19.59	4.93	16.00
		Min	**617.48**	719.62	677.86	798.17	822.90	784.81	1362.17
		Max	655.73	741.45	720.05	813.01	850.61	791.78	1384.79
	3	Mean	487.35	565.52	535.63	637.54	607.99	683.26	1092.87
		STD	5.21	26.51	57.70	62.22	18.70	47.87	18.66
		Min	**483.66**	546.78	494.82	593.55	594.76	649.41	1079.68
		Max	491.03	584.27	576.43	681.54	621.21	717.11	1106.06
	4	Mean	**404.22**	492.11	459.46	523.18	470.59	564.99	875.72
		STD	2.00	15.24	7.33	57.70	24.38	40.96	17.71
		Min	**402.81**	481.33	454.28	482.38	453.35	536.03	863.19
		Max	405.63	502.89	464.64	563.98	487.83	593.96	888.24
121	2	Mean	**725.42**	908.22	774.51	864.31	1010.18	1092.08	1782.79
		STD	30.51	79.87	0.12	37.49	27.89	1.03	7.57
		Min	**703.85**	851.74	774.42	837.80	990.46	1091.34	1777.44
		Max	746.99	964.70	774.59	890.82	1029.90	1092.81	1788.14
	3	Mean	**525.97**	611.01	605.29	733.06	749.01	792.39	1292.90
		STD	4.53	25.69	12.05	9.42	21.20	23.45	85.05
		Min	**522.77**	592.85	596.77	726.40	734.02	775.80	1232.76
		Max	529.17	629.18	613.81	739.72	764.01	808.97	1353.04
	4	Mean	**454.44**	515.78	488.50	597.19	598.12	735.13	1083.04
		STD	16.37	0.37	34.37	23.95	9.40	53.73	17.28
		Min	**442.87**	515.52	464.20	580.25	591.47	697.14	1070.82
		Max	466.02	516.04	512.80	614.12	604.77	773.12	1095.26

values indicated that GGA generally achieved mean values ranging from 514 to 725, showing GGA ability to consistently deliver satisfactory performance across diverse settings of nodes and UAVs. GGA was capable of providing reliable optimization solutions for CVRP contributing to improved efficiency and performance. In comparison to the other six algorithms, GGA exhibits competitive performance in terms of both minimum and mean values. While specific performance may vary depending on the settings of nodes and UAVs. GGA consistently demonstrates its efficacy in optimizing UAV networks, often closely followed by TS, GLS, and TS, in terms of minimum and mean values. The performance of the algorithms using 4 UAVs with 64, 81, 100, and 121 nodes is illustrated in the graphic representation shown in Figures 8 and 22, 23, and 24 in Appendix, respectively.

In Table 5, the number of nodes is 144, 169, 196, and 225. The number of UAVs with these nodes were, three, four, and five. Similar to the results given in Table 4, the GGA consistently emerged as a strong performer, exhibiting competitive mean values across all nodes and UAVs scenario. For instance, in scenarios with 144 nodes and 3 UAVs, GGA demonstrated a mean value of 613.15, with a minimum of around 601.76, showing its ability to consistently deliver efficient solutions. Moreover, TS,GLS, and ILS also got robust mean values, though slightly lower than GGA in most cases. These algorithms, while not surpassing GGA, still demonstrated strong optimization capabilities. In contrast,

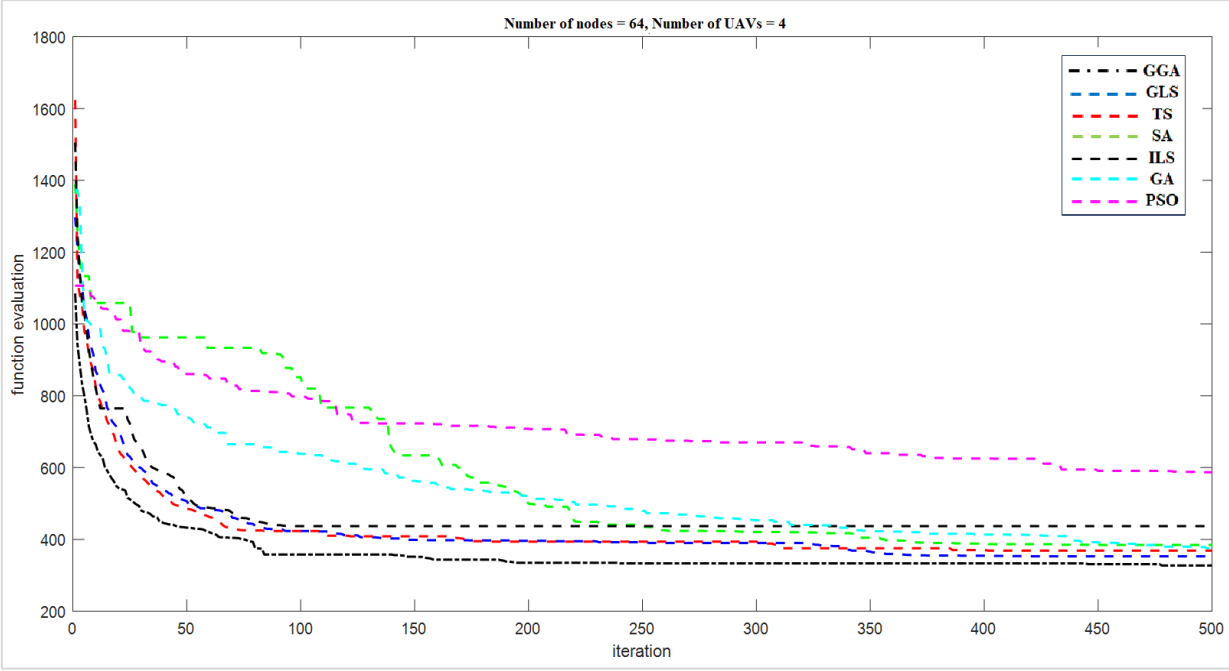

FIGURE 8. Algorithms' performance with 64 nodes and 4 UAVs.

algorithms such as SA, GA, and PSO exhibited relatively lower mean values across different scenarios. The performance of the algorithms using 5 UAVs with 144, 169, 196, and 255 nodes is illustrated in the graphic representation shown in Figures 9 and 24, 25, and 26 in Appendix, respectively.

In Table 6, the number of nodes is 256 and the number of UAVs utilised were, four, five, and six. The proposed GGA consistently emerged as a strong performer. For instance, in scenarios with 256 nodes and 4 UAVs, GGA demonstrated a mean value of 859.70, with a minimum of 852.03, indicating its ability to consistently deliver efficient solutions. Additionally, TS, GLS, and ILS algorithms also demonstrated robust mean values, although slightly lower than GGA in most cases. These algorithms gave competitive mean values, ranging from 1156.22 to 1229.37, highlighting their effectiveness in solving CVRP. Conversely, SA, GA, and PSO algorithms exhibited relatively lower mean values across different configurations. Despite showing some level of optimization effectiveness, their mean values suggested potential variability in performance. For example, in scenarios with 6 UAVs, GA exhibited a mean value of 1564.99, indicating its effectiveness but also highlighting variability in performance compared to GGA and other algorithms. Figure 10 shows the results of the algorithms' performance with 256 nodes and 6 UAVs.

Overall, the proposed GGA outperformed the other algorithms in most instances. GGA showed competitive results, closely following the TS's performance across different scenarios. In addition, GA shew good performance particularly when the number of nodes and UAVs was limited. Additionally, the remaining algorithms (i.e. GLS, ILS, SA, GA, and PSO) demonstrated effectiveness, although generally trailing behind the GGA and TS. However, as the number of nodes increased, the superiority of the GGA became more noticeable, surpassing all other algorithms in terms of both optimal solutions and mean values across iterations. This scalability of the GGA underscores its potential as a robust optimization technique for CVRP across various problem sizes and configurations.

D. COMPUTATIONAL TIME AND EFFICIENCY

The main objective of our proposed model is to minimise the overall consumption of batteries and tanks, while ensuring that they do not exceed their maximum capacities. This strategy places a higher importance on conserving energy (batteries and tanks) rather than minimising distance.

This section reports and analyses the computation time of the proposed GGA and the benchmark algorithms, including GLS, TS, SA, ILS, GA, and PSO. More precisely, the computation time of the models being compared is determined in second. Although the proposed method is a hybrid of GA and GLS, it functions efficiently within an acceptable duration, as given in Table 7.

The table presents a comparison of the computational durations for several metaheuristic algorithms in different scenarios, which are determined by the number of nodes and UAVs. With an increase in the number of nodes and UAVs, there is a consistent pattern of increased processing time observed across all methods. This pattern demonstrates the increasing complexity of problem. For instance, the computational time for GGA grows from 3.48 seconds when using 25 nodes and 1 UAV to 18.01 seconds when using 256 nodes and 4 UAVs. Other methods exhibit

TABLE 5. Experimental results with 64, 81, 100, and 121 nodes.

Number of Nodes	Number of UAVs		GGA	GLS	TS	SA	ILS	GA	PSO
144	3	Mean	613.15	706.12	673.62	835.73	794.35	1013.63	1642.62
		STD	16.12	19.15	3.63	43.79	56.65	49.49	46.00
		Min	**601.76**	692.58	671.05	804.76	754.30	978.63	1610.09
		Max	624.55	719.66	676.19	866.69	834.41	1048.62	1675.15
	4	Mean	502.62	641.26	583.12	704.60	737.13	895.92	1233.27
		STD	1.59	23.19	16.79	19.58	64.20	28.76	69.61
		Min	**501.50**	624.86	571.25	690.76	691.74	875.58	1184.05
		Max	503.74	657.66	594.98	718.45	782.53	916.26	1282.50
	5	Mean	453.99	509.61	624.35	605.32	638.58	819.81	1109.99
		STD	2.68	1.73	128.13	8.57	48.39	87.60	40.54
		Min	**452.10**	508.39	533.75	599.26	604.36	757.87	1081.32
		Max	455.88	510.83	714.96	611.38	672.79	881.75	1138.66
169	3	Mean	688.58	821.74	751.35	968.04	940.04	1240.93	1867.22
		STD	13.96	32.29	24.17	120.11	50.22	14.10	37.80
		Min	**678.71**	798.91	734.26	883.11	904.53	1230.96	1840.49
		Max	698.46	844.57	768.45	1052.97	975.55	1250.90	1893.94
	4	Mean	581.75	694.12	635.20	874.23	772.37	1156.12	1549.95
		STD	24.75	68.10	33.62	31.18	70.63	9.41	100.33
		Min	**564.24**	645.96	611.43	852.18	722.42	1149.47	1479.00
		Max	599.25	742.27	658.97	896.28	822.31	1162.78	1620.89
	5	Mean	524.79	638.02	638.82	728.22	630.00	1009.25	1372.20
		STD	22.58	44.23	76.90	24.60	20.27	6.89	13.34
		Min	**508.82**	606.74	584.44	710.82	615.66	1004.38	1362.77
		Max	540.75	669.29	693.19	745.62	644.33	1014.12	1381.63
196	3	Mean	762.89	960.96	900.18	1141.00	1053.24	1411.12	2229.62
		STD	7.58	91.04	45.90	17.10	59.11	87.26	110.43
		Min	**757.53**	896.58	867.73	1128.91	1011.45	1349.42	2151.54
		Max	768.25	1025.34	932.64	1153.09	1095.04	1472.82	2307.71
	4	Mean	618.39	918.46	755.72	920.31	890.65	1343.07	1830.16
		STD	13.09	143.07	45.36	13.89	62.82	56.17	90.31
		Min	**609.13**	817.30	723.65	910.49	846.24	1303.35	1766.30
		Max	627.64	1019.63	787.79	930.13	935.07	1382.79	1894.02
	5	Mean	575.36	676.31	679.10	823.26	840.81	1242.66	1642.26
		STD	49.19	60.65	5.23	40.32	90.19	13.77	84.18
		Min	**540.58**	633.42	675.40	794.75	777.04	1232.93	1582.74
		Max	610.14	719.19	682.80	851.77	904.59	1252.40	1701.78
225	3	Mean	876.28	1121.45	1026.75	1357.94	1178.43	1717.02	2709.10
		STD	4.60	25.39	24.56	5.34	39.43	46.86	18.68
		Min	**873.03**	1103.49	1009.38	1354.16	1150.55	1683.88	2695.89
		Max	879.53	1139.40	1044.12	1361.71	1206.31	1750.16	2722.30
	4	Mean	740.56	878.36	878.48	1171.78	970.05	1542.17	2183.00
		STD	45.85	48.49	2.47	63.05	46.14	37.33	61.20
		Min	**708.14**	844.08	876.74	1127.19	937.42	1515.77	2139.72
		Max	772.98	912.65	880.23	1216.36	1002.67	1568.56	2226.27
	5	Mean	647.67	749.75	750.77	941.44	868.11	1412.37	1860.70
		STD	24.44	40.68	7.49	17.18	62.22	148.85	31.77
		Min	**630.39**	720.98	745.47	929.29	824.12	1307.12	1838.24
		Max	664.95	778.51	756.07	953.59	912.11	1517.63	1883.17

TABLE 6. Experimental results with 256 nodes.

Number of Nodes	Number of UAVs		GGA	GLS	TS	SA	ILS	GA	PSO
256	4	Mean	**859.70**	1156.22	968.87	1229.37	1064.07	1690.98	2608.94
		STD	10.84	26.52	10.19	30.11	52.70	103.81	28.84
		Min	**852.03**	1137.46	961.66	1208.08	1026.81	1617.57	2588.55
		Max	867.37	1174.97	976.07	1250.67	1101.34	1764.38	2629.33
	5	Mean	**723.07**	871.32	871.96	1128.74	1024.59	1641.05	2154.74
		STD	19.27	2.52	28.03	69.97	83.02	16.11	61.36
		Min	**709.45**	869.54	852.14	1079.27	965.89	1629.65	2111.35
		Max	736.69	873.11	891.78	1178.22	1083.29	1652.44	2198.13
	6	Mean	**687.79**	893.16	783.86	988.98	914.99	1565.00	1985.85
		STD	17.99	143.74	7.82	6.48	87.63	24.17	73.30
		Min	**675.07**	791.52	778.33	984.39	853.03	1547.91	1934.02
		Max	700.52	994.80	789.39	993.56	976.95	1582.09	2037.68

similar patterns, suggesting that greater problem sizes require increased processing resources.

The PSO consistently demonstrates the shortest processing times among the algorithms in all cases, suggesting its

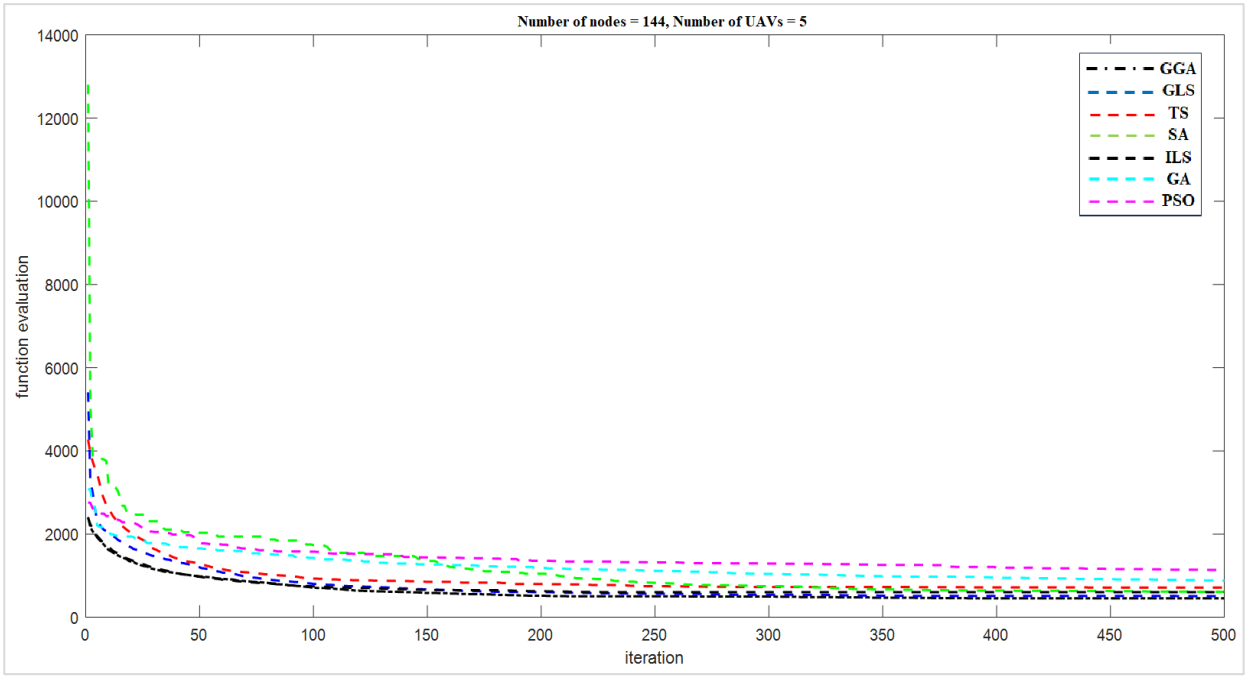

FIGURE 9. Algorithms' performance with 144 nodes and 5 UAVs.

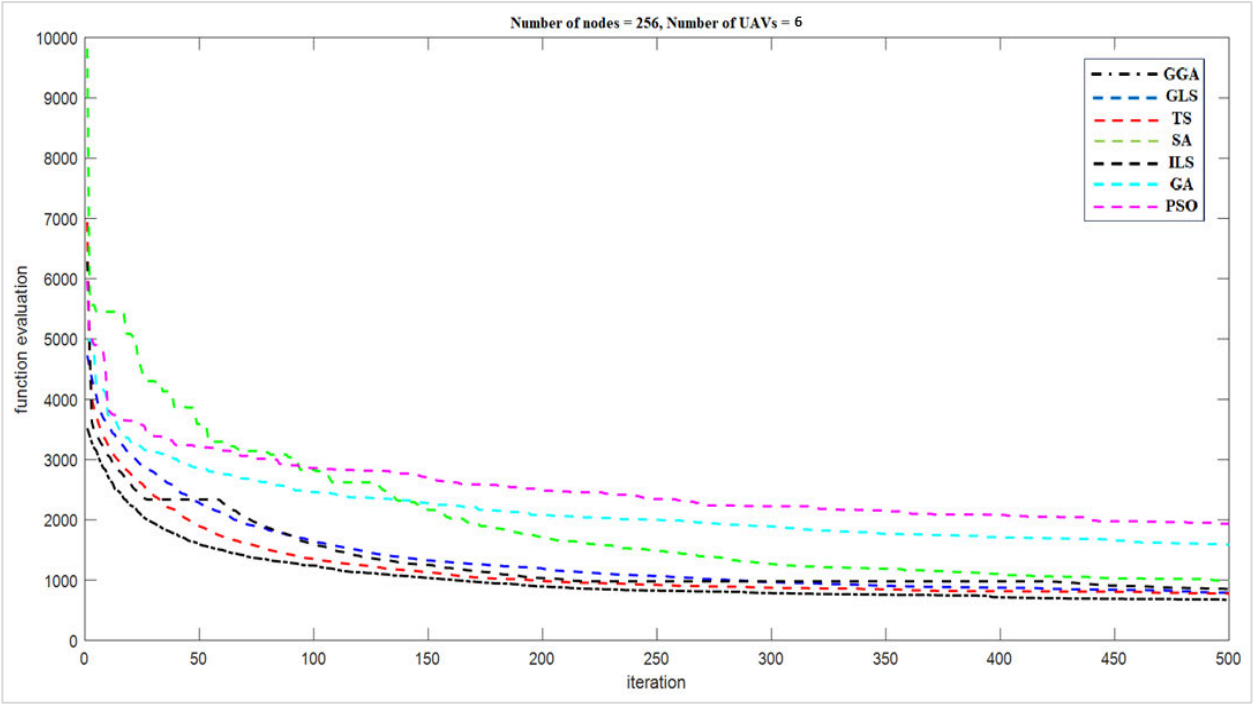

FIGURE 10. Algorithms' performance with 256 nodes and 6 UAVs.

computational efficiency and applicability for situations that demand prompt responses. GLS exhibits excellent computational efficiency, frequently ranking as the second fastest. As an example, when there are 25 nodes and 1 UAV, the GLS algorithm requires 1.25 seconds, whereas the PSO algorithm takes 2.19 seconds. On the other hand, TS and ILS typically exhibit the longest computational durations. As an instance, the TS algorithm requires 2,619.14 seconds to complete while using 256 nodes and 5 UAVs. In contrast, the ILS algorithm takes 2,734.84 seconds to complete under

TABLE 7. Computational time.

Number of Nodes	Number of UAVs	GGA	GLS	TS	SA	ILS	GA	PSO
	1	3.48	**1.25**	7.58	3.20	9.06	4.36	2.19
25	2	3.91	**1.56**	8.91	3.05	11.17	7.50	2.42
	3	2.50	**1.95**	9.92	4.14	12.03	5.41	2.58
	1	2.89	**2.42**	18.13	3.36	20.94	5.58	2.50
36	2	2.70	2.66	21.09	3.44	24.77	6.56	**2.58**
	3	3.40	3.20	23.91	3.75	26.56	6.75	**2.73**
	1	**2.73**	5.39	38.52	3.67	53.52	6.36	**2.73**
49	2	2.70	6.09	42.73	3.83	46.72	6.23	**2.66**
	3	**3.09**	5.70	44.14	3.59	52.50	7.62	3.13
	2	**2.85**	13.83	90.31	6.88	146.33	9.06	2.97
64	3	3.13	11.80	81.41	3.75	86.88	7.69	**2.66**
	4	**2.66**	12.03	85.63	3.83	92.73	7.55	3.05
	2	3.01	19.61	127.50	4.30	135.31	9.41	**2.97**
81	3	**3.40**	22.89	164.77	4.69	165.39	8.94	3.98
	4	**3.20**	25.31	182.42	5.16	175.39	8.44	**3.20**
	2	4.02	44.77	231.48	4.53	236.33	9.70	**3.36**
100	3	4.14	41.95	247.81	4.77	258.91	9.91	**3.36**
	4	4.18	42.11	265.23	5.08	276.56	10.88	**3.36**
	2	4.53	67.66	384.38	4.45	381.33	9.57	**3.59**
121	3	4.26	75.16	395.94	5.08	416.41	11.72	**3.98**
	4	6.48	77.97	418.20	5.16	439.22	12.35	**3.59**
	3	5.94	131.88	600.39	5.70	654.22	13.41	**4.69**
144	4	7.58	140.39	715.47	6.56	728.91	14.08	**4.61**
	5	7.54	155.16	741.02	6.72	754.69	16.03	**4.61**
	3	8.95	233.67	1033.05	6.80	1057.81	15.90	**5.08**
169	4	8.55	233.67	1053.05	6.80	1085.00	17.50	**4.77**
	5	6.68	243.83	1108.20	6.88	1081.48	13.80	**4.61**
	3	7.19	337.81	1266.17	5.94	1327.81	14.95	**4.38**
196	4	9.92	319.22	1310.31	6.41	1353.91	16.13	**4.53**
	5	7.66	378.28	1431.25	7.19	1442.58	17.81	**4.53**
	3	12.38	528.44	1821.95	6.80	1863.44	16.20	**4.69**
225	4	12.62	509.84	1864.22	6.64	1943.59	16.47	**4.53**
	5	14.49	556.64	2165.70	7.97	2287.81	23.13	**5.55**
	4	18.01	902.58	2647.97	7.27	2702.89	18.00	**4.69**
256	5	16.33	796.56	2619.14	7.11	2734.84	19.30	**5.00**
	6	16.52	695.31	2719.14	7.11	3064.84	22.69	**6.09**

the identical conditions. These findings indicate that TS and ILS may not be the most suitable options for applications requiring fast computing.

Increasing the number of UAVs has an ability to increase the computational duration for all algorithms. This is clearly demonstrated by the increase from 1 to 3 UAVs in all node configurations. For instance, when there are 25 nodes, the computing time for GGA grows from 3.48 seconds with 1 UAV to 2.50 seconds with 3 UAVs, despite a temporary drop at 3 UAVs. This highlights the occurrence of non-linear behaviour in certain instances. The GLS algorithm demonstrates a more consistent and predictable increase in processing time as more UAVs are added, suggesting a more reliable scaling pattern.

Each algorithm can be analysed using certain observations. The GGA algorithm, which is a combination of GA and GLS, has a complex structure that results in considerable calculation times. However, it is highly efficient in terms of tank and battery use. GGA and GA demonstrate moderate computing speeds, striking a balance between the rapid PSO and GLS, and the slower TS and ILS. Their performance exhibits a proportional rise with the addition of nodes and UAVs. The SA has shorter computational durations in comparison to TS and ILS, however it is often slower than PSO and GLS. The data illustrates an obvious correlation between the number of nodes and UAVs and the corresponding rise in computational time. Both TS and ILS exhibit considerable processing times, rendering them less favourable for large-scale applications where rapid computation is crucial.

The PSO has superior performance compared to other algorithms like GGA and TS in highly complex scenarios, such as those involving 256 nodes and 6 UAVs. PSO achieves a processing time of 6.09 seconds, while GGA takes 16.52 seconds and TS takes 2,719.14 seconds.

E. STATISTICAL ANALYSIS

The Non-Parametric Wilcoxon signed-rank test was utilised to assess if there is a statistically significant difference between the proposed GGA and six other algorithms (i.e. GLS, TS, SA, ILS, GA, and PSO). Table 8 presents the statistical results of this test. The $p - value$ higher than 0.05 indicate statistically significant differences among the algorithms being compared. If $R^+ > R^-$, then the proposed GGA significantly outperforms the compared method; otherwise, the compared algorithm outperforms GGA significantly. Bold denote the optimal outcomes of the comparative algorithms. The indicator (ind) represents the disparity in performance scores between the two algorithms. If $ind = +1$,

then the proposed GGA outperforms the other algorithm. If $ind = -1$, then the other algorithm outperforms the proposed GGA. If $ind = 0$, then the performance of the two algorithms is equal.

Based on the test results given in Table 8, the null hypothesis is rejected if the computed $p-value$ is less than 0.05. This indicates that there is a substantial difference between the suggested GGA and the other algorithms. When comparing GLS and ILS to GGA, the value of R^+ was consistently higher than R^- in all instances, except for the second instance where $ind = -1$. When comparing TS, SA, and TSO to GGA, the value of R^+ was consistently higher than R^- in all instances, and the value of ind was equal to 1. When comparing GA to GGA, the value of R^+ was consistently higher than R^- in all instances, except for the first and fourth instances where the value of ind was -1.

F. THEORETICAL AND PRACTICAL IMPLICATIONS

The study has important theoretical implications for the ongoing development in CVRP field and optimisation techniques. This research expands the current body of knowledge on metaheuristic methods for tackling intricate routing problems by reformulating the GGA specificaly for CVRP. The combination of GA and GLS enhances the set of methods used to address VRP, particularly in the context of UAVs-based pesticide spraying missions. The study shows that GGA is not only a feasible alternative, but also a superior approach in many instances when compared to other well-known metaheuristic algorithms. This study enhances the theoretical knowledge of how hybrid algorithms might utilise the strengths of their own approaches to obtain improved performance in optimisation tasks. In addition, future researchers and developers can benefit from the proposed method to solve their own problems.

Practically speaking, the results of this study have significant implications for the practical use of UAVs in agricultural environments. Efficiently managing the consumption of batteries and tanks is essential for maximising the effectiveness and efficiency of UAV operations, especially in precision agriculture where limited resources are a major concern. The suggested GGA algorithm can result in cost savings and improved operational efficiency by reducing battery consumption and optimising pesticide usage. The real-world benefits are apparent in the enhanced performance measures, such as decreased average and variability in solution quality, resulting in more dependable and foreseeable UAV operations. Moreover, the comprehensive assessment, which incorporates the Wilcoxon signed-rank test across several scenarios, offers strong proof of the GGA's efficacy, establishing it as a realistic tool for UAV routing and spraying assignments. This can result in a wider acceptance of UAVs for pesticide application, ultimately promoting more sustainable farming practices.

VI. CASE STUDY AND MANAGERIAL INSIGHTS

The case study utilises the proposed GGA in conjunction with six benchmark algorithms to address the CVRP in which three UAVs are assigned to spray pesticides over the Baqubah farm in Iraq. This comparative analysis evaluates the effectiveness of GGA in optimising UAV routes, specifically designed for agricultural pesticide application. The geographical map is uplouded into a specialised software toolkit, as depicted in Figure 11. Through the graphical user interface, users establish the Region of Interest (ROI), outlining 61 nodes that represent distinct pesticide application zones, as shown in Figure 12.

FIGURE 11. The uploaded map of Baqubah farm to be surveyed by three UAVs.

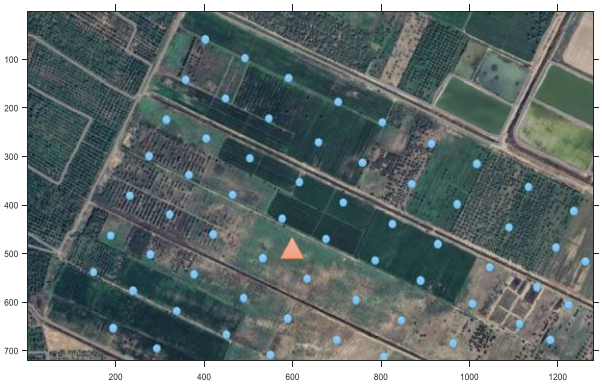

FIGURE 12. ROI with 61 nodes represent distinct pesticide application zones.

Following the parameters configuration based on Table 1 for the algorithms, the case study problem undergoes a defined capacity. The GGA and the 6 benchmark algorithms' simulations yield results are shown in Figures 13-19. It can be observed that GGA took the shortest path between nods. However, PSO shows the worst scenario in terms of coverage path by the three UAVs.

The computational costs and time of executing the algorithms in the case study are given in Table 9. Out all the algorithms examined, the GGA exhibits the highest level of computing efficiency, with a time requirement of 181.92 second. This metric measures the algorithm's capacity to minimise the computational resources required while efficiently optimising routes for pesticide application. On the

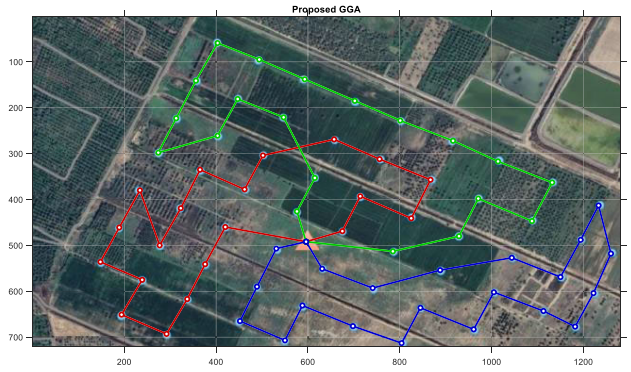

FIGURE 13. The path based on GGA.

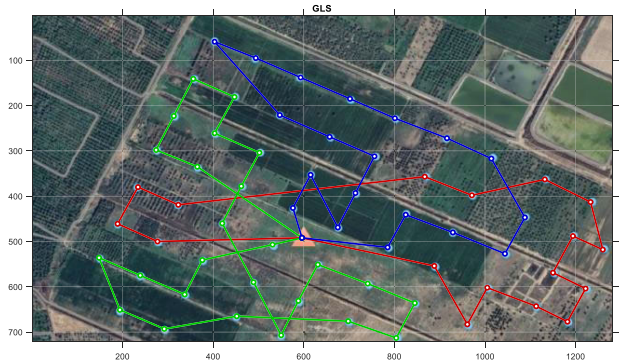

FIGURE 14. The path based on GLS.

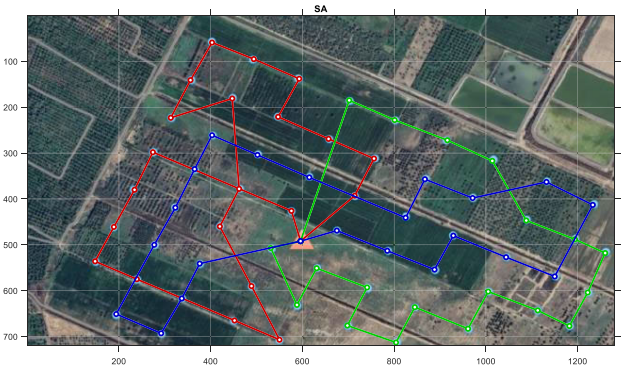

FIGURE 15. The path based on SA.

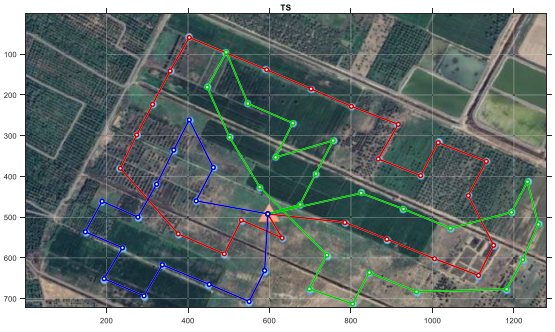

FIGURE 16. The path based on TS.

other hand, ILS has the greatest computational time of 287.45 second, which suggests that it requires more time and processing resources compared to other methods.

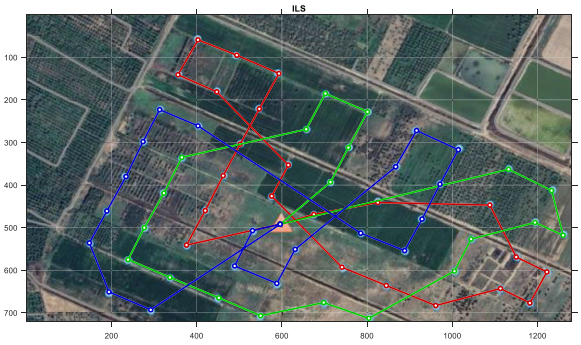

FIGURE 17. The path based on ILS.

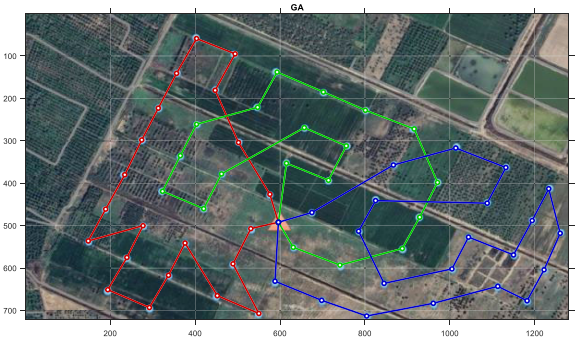

FIGURE 18. The path based on GA.

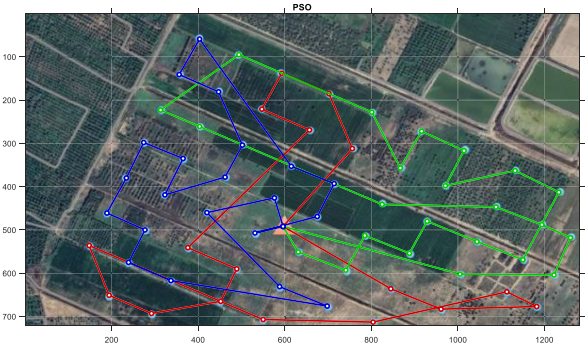

FIGURE 19. The path based on PSO.

The computational costs for the remaining methods are as follows: GLS is 158.02, TS is 283.20, SA is 128.50, GA is 121.39, and PSO is 127.47. The results demonstrate differing levels of efficiency and resource utilisation across several metaheuristic approaches while solving the UAV routing optimisation problem.

The high efficiency demonstrated by GGA indicates that it has the potential to be successfully used in real-world applications, as it is able to effectively optimise routes while still being computationally practical.

This research focuses on the CVRP for UAV-based pesticide spraying missions. However, the principles of the proposed GGA can be applied to a wider range of optimisation problems. The GA can be customised to enhance the efficiency of routing and resource allocation in several scenarios, including: Municipalities can utilise GGA to strategically plan optimal routes for waste collection vehicles,

TABLE 8. Wilcoxon signed-rank test results.

Instances	Nodes No.	UAVs No.	GLS VS GGA				TS VS GGA				SA VS GGA				ILS VS GGA				GA VS GGA				PSO VS GGA			
			P-value	R+	R-	*ind*	P-value	R+	R-	*ind*	P-value	R+	R-	*ind*	P-value	R+	R-	*ind*	P-value	R+	R-	*ind*	P-value	R+	R-	*ind*
1		1	0.5	3	0	1	0.5	3	0	1	1	3	0	1	0.5	3	0	1	1	1	2	-1	0.5	3	0	1
2	25	2	0.5	0	3	-1	1	2	1	1	1	2	1	1	0.5	0	3	-1	1	2	1	1	0.5	3	0	1
3		3	0.5	3	0	1	0.5	3	0	1	1	2	1	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
4		1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	1	0	3	-1	0.5	3	0	1
5	36	2	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	1	2	1	1	0.5	3	0	1
6		3	1	2	1	1	0.5	3	0	1	1	2	1	1	0.5	3	0	1	1	2	1	1	0.5	3	0	1
7		1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
8	49	2	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	1	3	0	1	0.5	3	0	1
9		3	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
10		2	0.5	3	0	1	1	2	1	1	1	2	1	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
11	64	3	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
12		4	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
13		2	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
14	81	3	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
15		4	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
16		2	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
17	100	3	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
18		4	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
19		2	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
20	121	3	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
21		4	0.5	3	0	1	1	2	1	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
22		3	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
23	144	4	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
24		5	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
25		3	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
26	169	4	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
27		5	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
28		3	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
29	196	4	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
30		5	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
31		3	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
32	225	4	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
33		5	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
34		4	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
35	256	5	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1
36		6	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1	0.5	3	0	1

TABLE 9. Algorithms' computational costs and time in the case study.

Algorithms	Cost	Time
The proposed GGA	4560.34	181.92
GLS	5582.62	158.02
TS	5253.21	283.20
SA	5269.80	128.50
ILS	5861.34	287.45
GA	5001.79	121.39
PSO	7087.49	127.47

ensuring an optimal distribution of load capacities and thus minimising operational expenses. Emergency-services, including ambulances and fire engines, can gain advantages from optimised routing to decrease response times and enhance service coverage. GGA is capable of optimising the allocation of goods from warehouses to sellers, guaranteeing the effective utilisation of vehicle capacity and minimising transportation expenses. Transportation authorities can utilise GGA to strategically plan bus routes that optimise geographical coverage, taking into account factors such as vehicle capacity limitations.

The GGA is a versatile and robust algorithm because it combines the explorative abilities of GAs with the GLS. The ability of this model to maintain a balance between exploration and exploitation is extremely beneficial for a range of optimisation problems where the nearly optimum solutions is the highest priority. Notable characteristics that improve its usefulness include:

- The GGA's ability to effectively handle larger problem sizes, as shown in the CVRP scenarios with different numbers of nodes and UAVs, suggests its potential for scaling in other optimisation problems of large scale.
- The GGA can be customised by adjusting the GA operators and GLS to match the specific constraints and objectives of various optimisation tasks.
- The statistical analysis and comparison of GGA with other metaheuristic algorithms demonstrate its resilience, confirming it as a dependable option for various applications.

This research highlights the adaptability and potential for broad application of the GGA, making it a valuable optimisation tool that may be utilised in many domains beyond just UAV-based pesticide spraying.

VII. CONCLUSION AND FUTURE WORKS

The successful application of precision agriculture relies on the efficient optimisation of agrotechnical procedures in order to greatly enhance crop yields. UAVs serve as invaluable tools for data collection and carrying out specific agricultural operations. Nevertheless, despite their potential, numerous unsolved obstacles hinder the broad implementation of UAV technology for precision agriculture. One of these tasks is VRP to solve CVRP. These tasks might be more challenging when taking into account the limitations of battery and tank capacities in UAV operations. In this paper, the

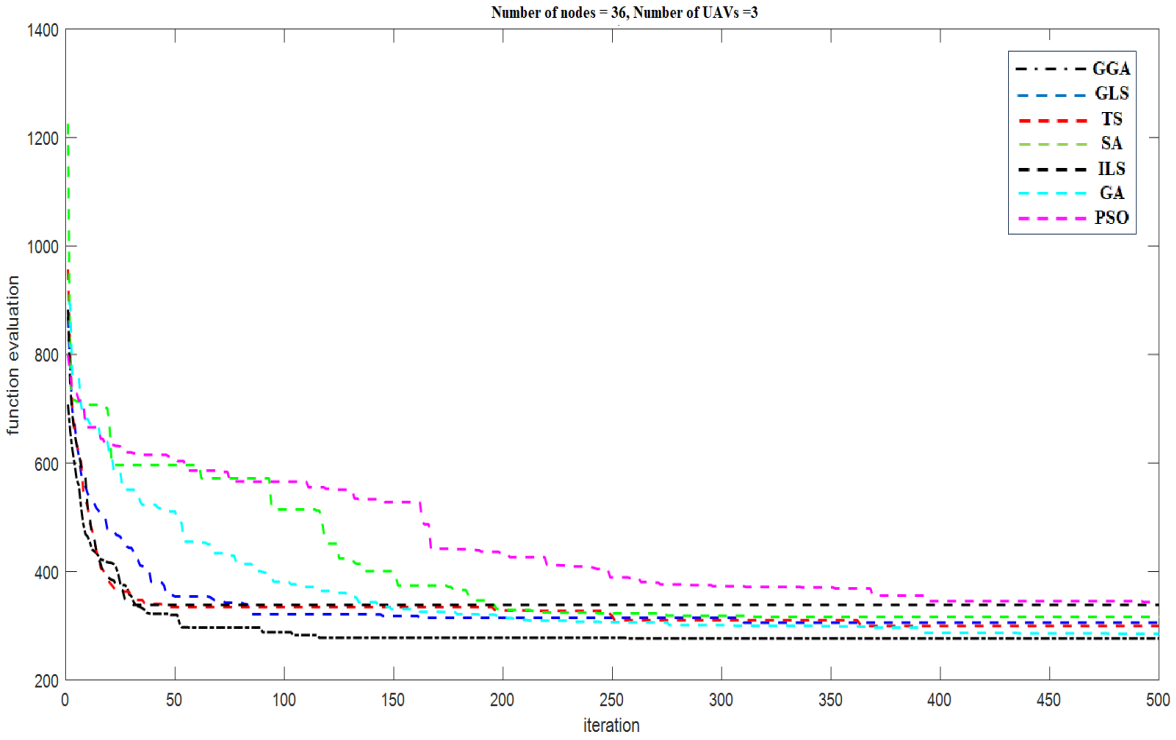

FIGURE 20. Algorithms' performance with 36 nodes.

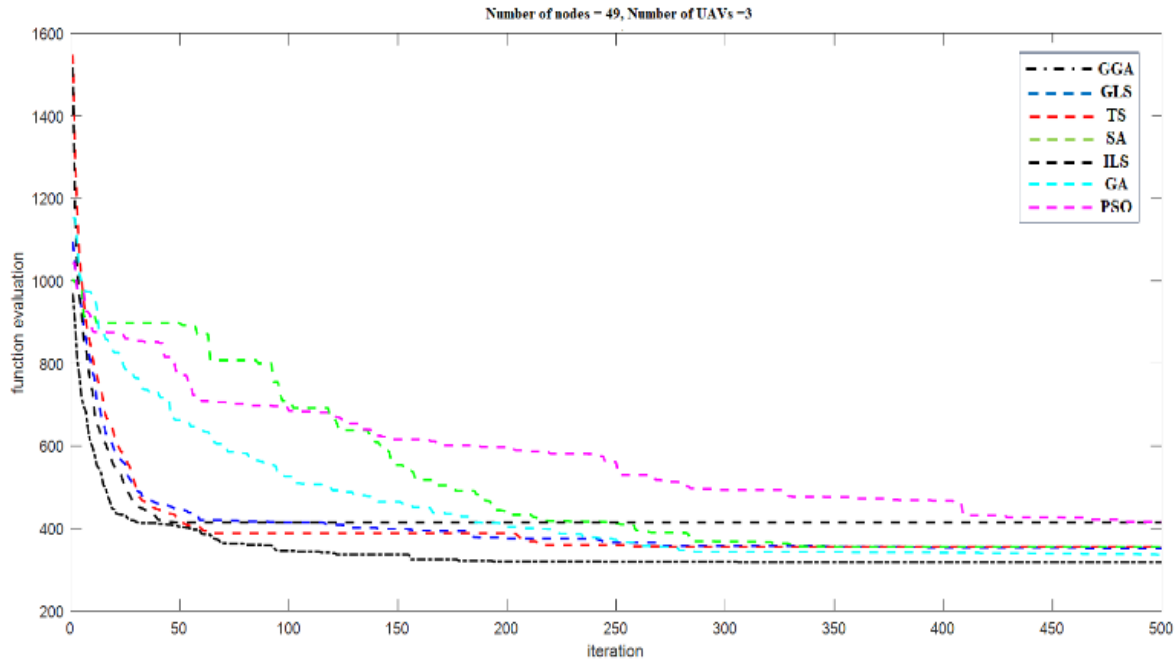

FIGURE 21. Algorithms' performance with 49 nodes.

authors have proposed a hybrid metaheuristic optimisation algorithm known as the GGA, which integrates GA with a GLS algorithm.

For the experiment, the number of nodes was set to 25, 36, 49, 64, 81, 100, 121, 144, 169, 225, 256, with the number of UAVs varying between 1 UAV to 6 UAVs depending on the number of nodes. The results confirm that the proposed GGA successfully resolves the CVRP for UAV-based pesticide spraying missions. Regardless of the number of nodes and UAVs involved, GGA continually demonstrated competitive performance by consistently reaching lower mean and minimum values and maintaining low standard deviations across

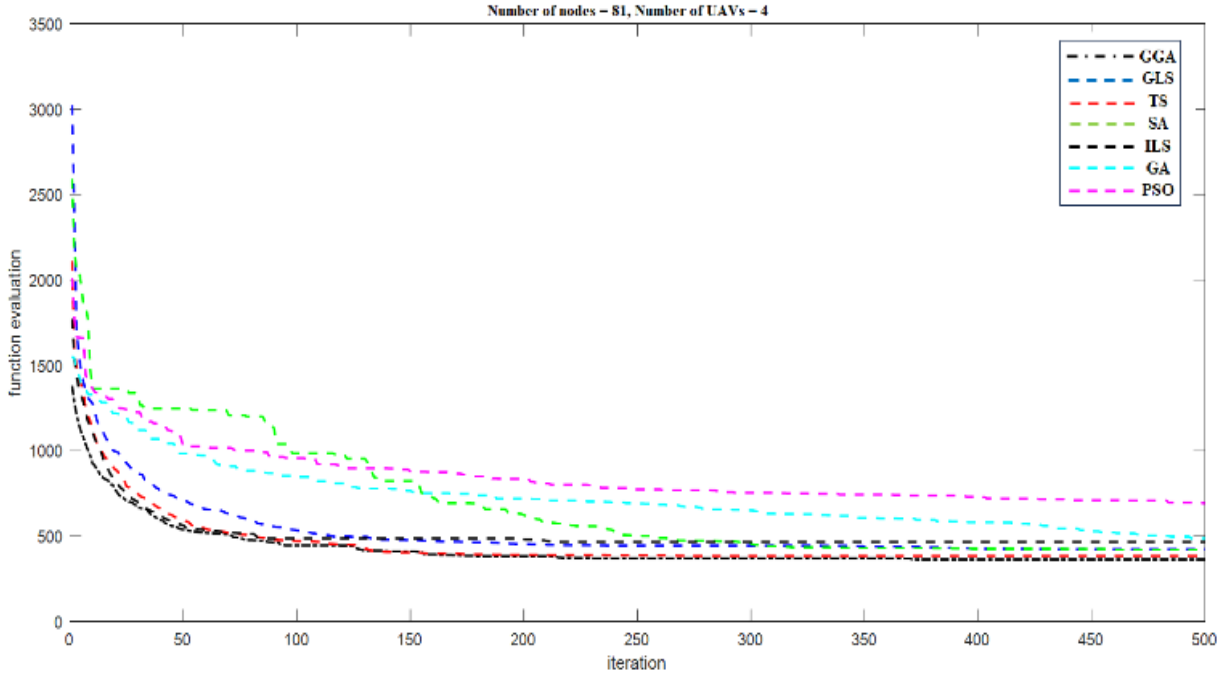

FIGURE 22. Algorithms' performance with 81 nodes.

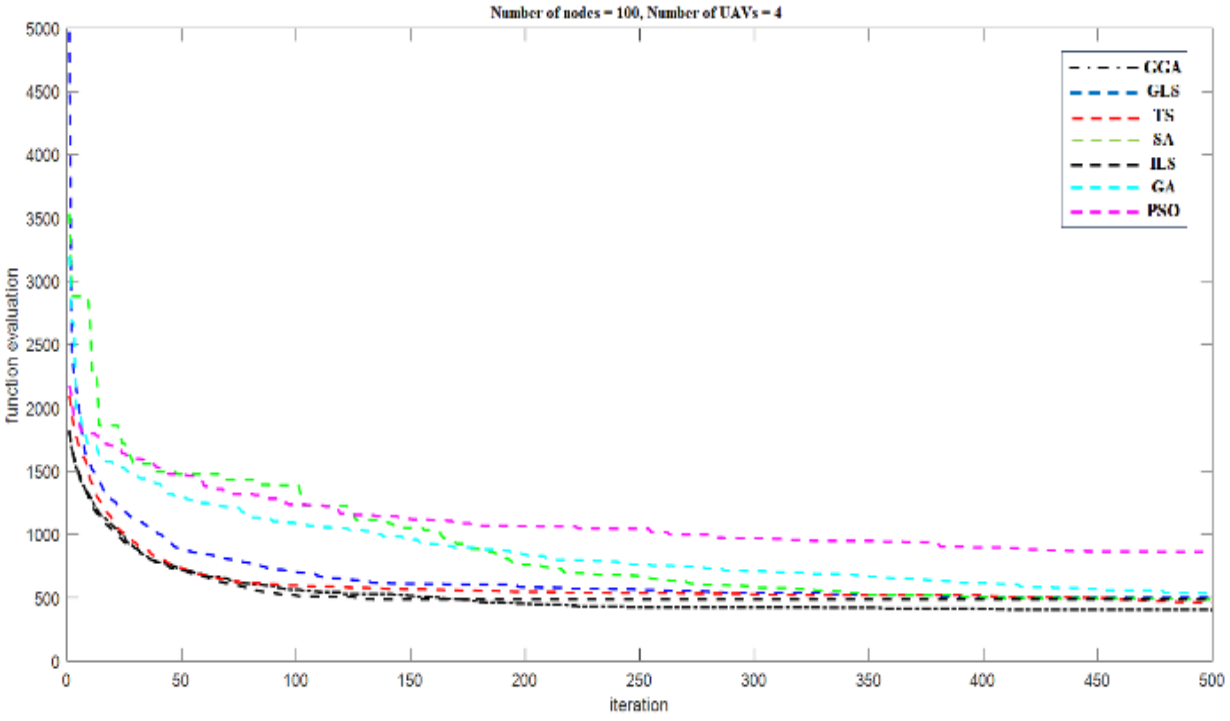

FIGURE 23. Algorithms' performance with 100 nodes.

most iterations. At first, the GA provided better performance when there were fewer nodes and UAVs. However, as the complexity increased, it became clear that the GGA was superior. With an increasing number of nodes, GGA consistently outperformed competing algorithms, demonstrating its robust performance even in the largest cases with 256 nodes. When compared to other metaheuristic algorithms such as TS, GLS, ILS, SA, and PSO, GGA consistently achieved superior results, especially as the size of the problem increased.

According to the Wilcoxon signed-rank test results, GGA consistently outperformed GLS and ILS in most instances, except for one where the performance was slightly lower.

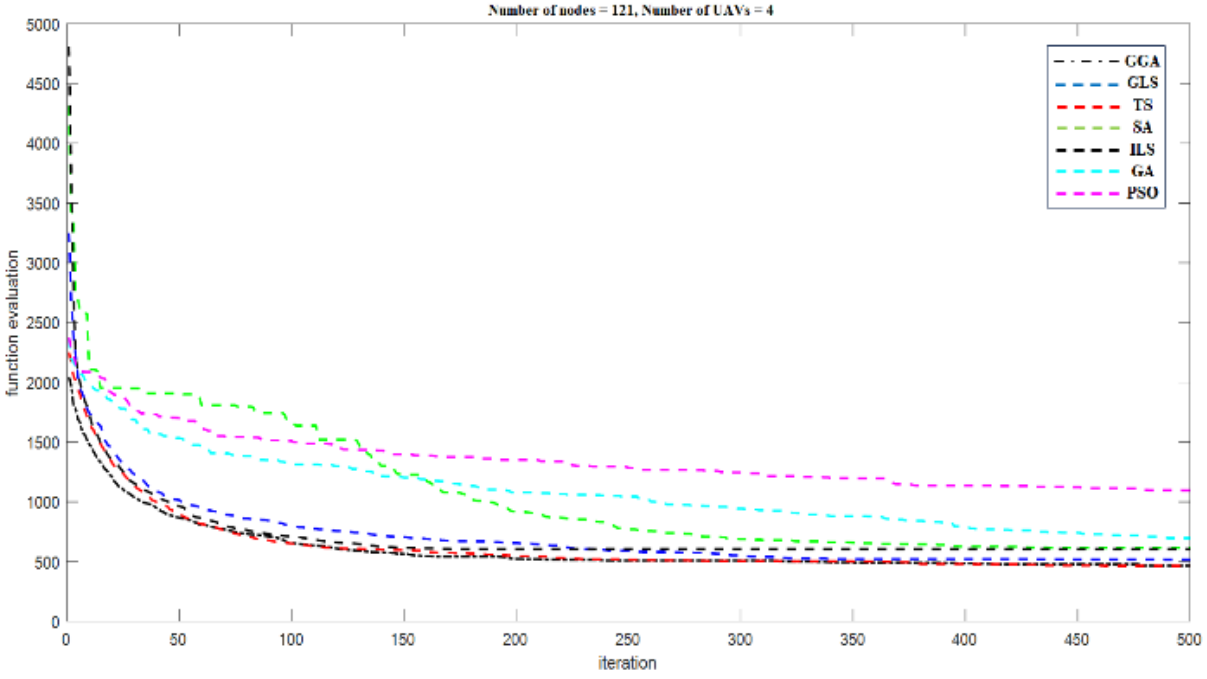

FIGURE 24. Algorithms' performance with 121 nodes.

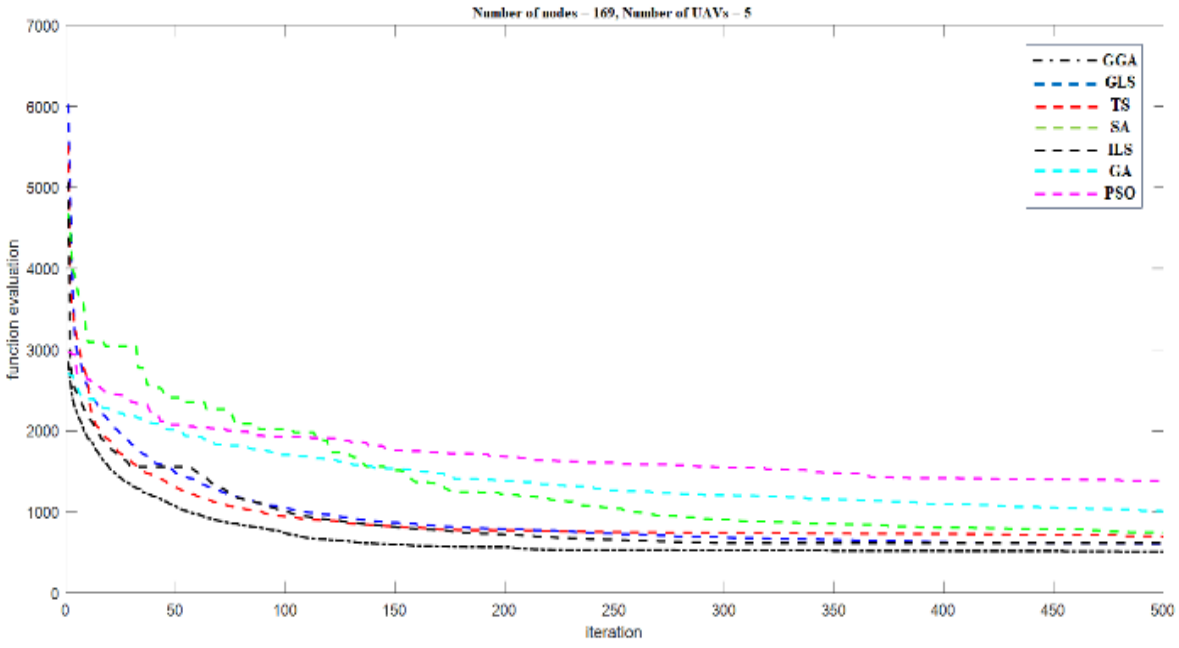

FIGURE 25. Algorithms' performance with 169 nodes.

When compared to TS, SA, and PSO, GGA showed superior performance in all instances. Similarly, GGA outperformed GA in most cases, with only two exceptions where GA performed better. Overall, the statistical analysis confirms that GGA generally outperforms the other algorithms, demonstrating its effectiveness and robustness in solving CVRP.

The limitations of the proposed method: (i) the proposed method may rely on certain assumptions or simplifications of the real-world agricultural environment, which could limit its applicability in complex and dynamic agricultural settings. (ii) Deploying UAVs for precision agriculture involves practical challenges such as flight regulations, weather conditions, equipment reliability, and operational costs. (iii) While the proposed method has shown promising results, its scalability to large-scale agricultural operations with numerous UAVs and complex constraints remains a challenge. (iv) Many

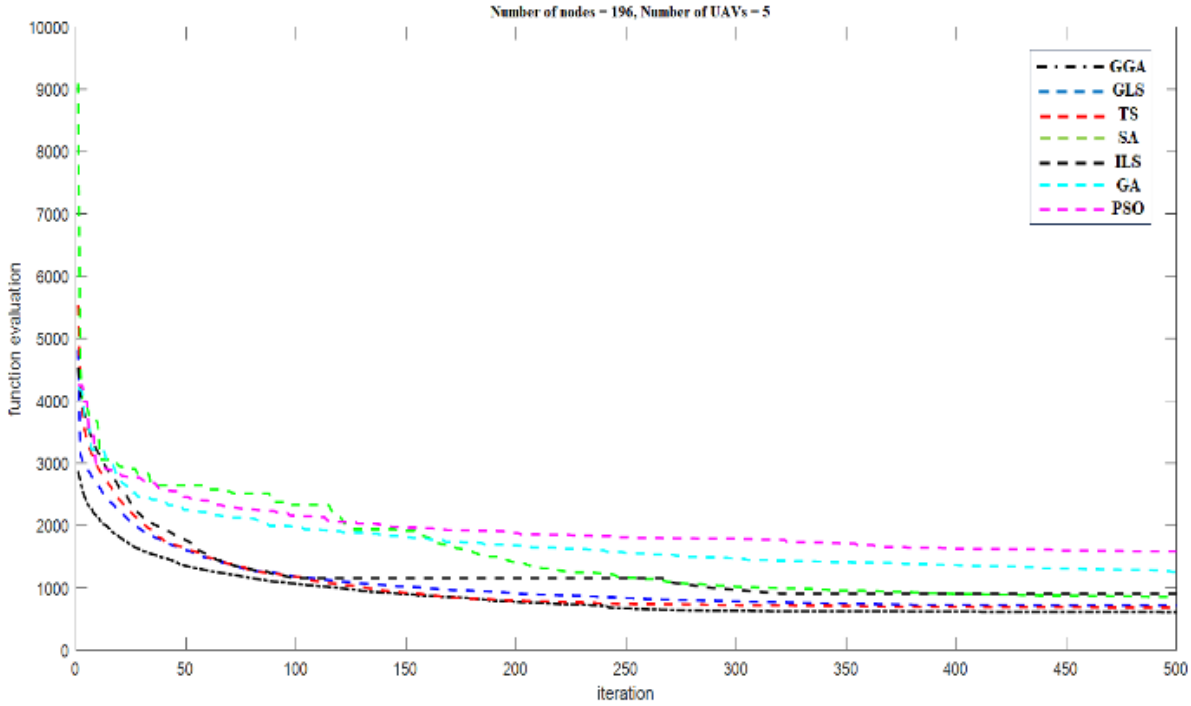

FIGURE 26. Algorithms' performance with 196 nodes.

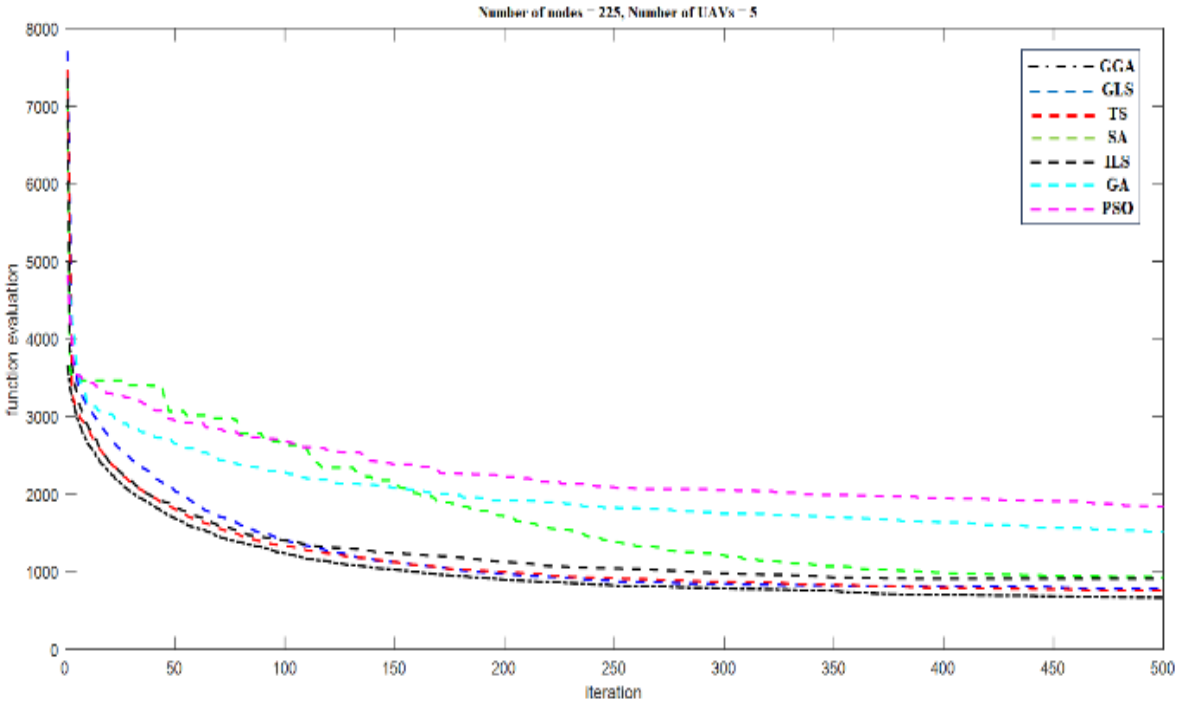

FIGURE 27. Algorithms' performance with 225 nodes.

agricultural tasks involve optimizing multiple objectives simultaneously, such as minimizing cost, maximizing coverage, and reducing environmental impact, which is not covered in this paper.

In the future research these limitations can be addressed by: (i) exploring more realistic models and incorporating additional constraints and factors into the optimization model. (ii) the proposed method should be evaluated and validated

under real-world conditions to assess its feasibility and effectiveness in practical agricultural settings. (iii) developing scalable algorithms and techniques that can handle larger problem instances efficiently. (iv) exploring multi-objective optimization techniques to find trade-offs between conflicting objectives. (v) Future studies can investigate the utilisation of other meta heuristic algorithms such ashybrid spider monkey optimisation approach and the raccoon family optimisation algorithm with GLS in a two-stage procedure to tackle the CVRP. (vi) Future research can employ mTSP to solve CVRP or VRP. (vii) Incorporating distance metrics to the proposed model to further enhance the model's applicability and efficiency.

APPENDIX

See Figures 20–27.

REFERENCES

[1] D. Dorling, "World population prospects at the UN: Our numbers are not our problem?" in *The Struggle for Social Sustainability*. Bristol, England: Policy Press, 2021, pp. 129–154.

[2] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, "A review on UAV-based applications for precision agriculture," *Information*, vol. 10, no. 11, p. 349, Nov. 2019, doi: 10.3390/info10110349.

[3] S. Manfreda et al., "On the use of unmanned aerial systems for environmental monitoring," *Remote Sens.*, vol. 10, no. 4, p. 641, Apr. 2018, doi: 10.3390/rs10040641.

[4] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios, "A compilation of UAV applications for precision agriculture," *Comput. Netw.*, vol. 172, May 2020, Art. no. 107148, doi: 10.1016/j.comnet.2020.107148.

[5] L. Campioni, F. Poltronieri, C. Stefanelli, N. Suri, M. Tortonesi, and K. Wrona, "Enabling civil–military collaboration for disaster relief operations in smart city environments," *Future Gener. Comput. Syst.*, vol. 139, pp. 181–195, Feb. 2023, doi: 10.1016/j.future.2022.09.020.

[6] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Comput. Commun.*, vol. 149, pp. 270–299, Jan. 2020, doi: 10.1016/j.comcom.2019.10.014.

[7] V. Sehrawat, "Legal status of drones under LOAC and international law," *Penn State J. Law Int. Affairs*, vol. 5, p. 164, Jul. 2017.

[8] R. Mesquita and P. D. Gaspar, "A path planning optimization algorithm based on particle swarm optimization for UAVs for bird monitoring and repelling–simulation results," in *Proc. Int. Conf. Decis. Aid Sci. Appl. (DASA)*, Nov. 2020, pp. 1144–1148, doi: 10.1109/DASA51403.2020.9317271.

[9] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017, doi: 10.1109/TSMC.2016.2582745.

[10] R. Rumba and A. Nikitenko, "The wild west of drones: A review on autonomous- UAV traffic-management," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 1317–1322, doi: 10.1109/ICUAS48674.2020.9214031.

[11] S. I. Khan, Z. Qadir, H. S. Munawar, S. R. Nayak, A. K. Budati, K. D. Verma, and D. Prakash, "UAVs path planning architecture for effective medical emergency response in future networks," *Phys. Commun.*, vol. 47, Aug. 2021, Art. no. 101337, doi: 10.1016/j.phycom.2021.101337.

[12] X. Wang, H. Zhao, T. Han, H. Zhou, and C. Li, "A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem," *Appl. Soft Comput.*, vol. 78, pp. 240–260, May 2019, doi: 10.1016/j.asoc.2019.02.037.

[13] N. Mohamed, J. Al-Jaroodi, I. Jawhar, A. Idries, and F. Mohammed, "Unmanned aerial vehicles applications in future smart cities," *Technological Forecasting Social Change*, vol. 153, Apr. 2020, Art. no. 119293, doi: 10.1016/j.techfore.2018.05.004.

[14] P. Doherty and P. Rudol, "A UAV search and rescue scenario with human body detection and geolocalization," in *AI: Advances in Artificial Intelligence*. Cham, Switzerland: Springer, 2007, pp. 1–13.

[15] M. Erdelj and E. Natalizio, "UAV-assisted disaster management: Applications and open issues," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2016, pp. 1–5, doi: 10.1109/ICCNC.2016.7440563.

[16] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458, Dec. 2018, doi: 10.1002/net.21818.

[17] H. Choset, "Coverage for robotics—A survey of recent results," *Ann. Math. Artif. Intell.*, vol. 31, no. 1, pp. 113–126, 2001.

[18] T. Cabreira, L. Brisolara, and P. R. Ferreira, "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, p. 4, Jan. 2019, doi: 10.3390/drones3010004.

[19] R. I. Mukhamediev, K. Yakunin, M. Aubakirov, I. Assanov, Y. Kuchin, A. Symagulov, V. Levashenko, E. Zaitseva, D. Sokolov, and Y. Amirgaliyev, "Coverage path planning optimization of heterogeneous UAVs group for precision agriculture," *IEEE Access*, vol. 11, pp. 5789–5803, 2023, doi: 10.1109/ACCESS.2023.3235207.

[20] J. F. Araujo, P. B. Sujit, and J. B. Sousa, "Multiple UAV area decomposition and coverage," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl. (CISDA)*, Apr. 2013, pp. 30–37.

[21] S. M. Ahmadi, H. Kebriaei, and H. Moradi, "Constrained coverage path planning: Evolutionary and classical approaches," *Robotica*, vol. 36, no. 6, pp. 904–924, Jun. 2018, doi: 10.1017/s0263574718000139.

[22] J. Mumtaz, Z. Guan, L. Yue, Z. Wang, S. Ullah, and M. Rauf, "Multi-level planning and scheduling for parallel PCB assembly lines using hybrid spider monkey optimization approach," *IEEE Access*, vol. 7, pp. 18685–18700, 2019, doi: 10.1109/ACCESS.2019.2895954.

[23] M. Rauf, Z. Guan, L. Yue, Z. Guo, J. Mumtaz, and S. Ullah, "Integrated planning and scheduling of multiple manufacturing projects under resource constraints using raccoon family optimization algorithm," *IEEE Access*, vol. 8, pp. 151279–151295, 2020, doi: 10.1109/ACCESS.2020.2971650.

[24] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," *Knowl.-Based Syst.*, vol. 158, pp. 54–64, Oct. 2018, doi: 10.1016/j.knosys.2018.05.033.

[25] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 1–20, Jun. 1994, doi: 10.1007/bf00175354.

[26] M. Kantardzic, "Genetic algorithms," in *Data Mining: Concepts, Models, Methods, and Algorithms*, 2011, pp. 385–413.

[27] S. Panda and N. P. Padhy, "Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design," *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1418–1427, Sep. 2008, doi: 10.1016/j.asoc.2007.10.009.

[28] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013, doi: 10.1109/TII.2012.2198665.

[29] V. Jamshidi, V. Nekoukar, and M. H. Refan, "Analysis of parallel genetic algorithm and parallel particle swarm optimization algorithm UAV path planning on controller area network," *J. Control, Autom. Electr. Syst.*, vol. 31, no. 1, pp. 129–140, Feb. 2020, doi: 10.1007/s40313-019-00549-9.

[30] Y. V. Pehlivanoglu and P. Pehlivanoglu, "An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems," *Appl. Soft Comput.*, vol. 112, Nov. 2021, Art. no. 107796, doi: 10.1016/j.asoc.2021.107796.

[31] T. H. Pham, Y. Bestaoui, and S. Mammar, "Aerial robot coverage path planning approach with concave obstacles in precision agriculture," in *Proc. Workshop Res., Educ. Develop. Unmanned Aerial Syst.*, Oct. 2017, pp. 43–48, doi: 10.1109/RED-UAS.2017.8101641.

[32] R. Shivgan and Z. Dong, "Energy-efficient drone coverage path planning using genetic algorithm," in *Proc. IEEE 21st Int. Conf. High Perform. Switching Routing (HPSR)*, May 2020, pp. 1–6, doi: 10.1109/HPSR48589.2020.9098989.

[33] I. Khoufi, A. Laouiti, and C. Adjih, "A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles," *Drones*, vol. 3, no. 3, p. 66, Aug. 2019, doi: 10.3390/drones3030066.

[34] P. Kitjacharoenchai, M. Ventresca, M. Moshref-Javadi, S. Lee, J. M. A. Tanchoco, and P. A. Brunese, "Multiple traveling salesman problem with drones: Mathematical model and heuristic approach," *Comput. Ind. Eng.*, vol. 129, pp. 14–30, Mar. 2019, doi: 10.1016/j.cie.2019.01.020.

[35] J. Xie, L. R. Garcia Carrillo, and L. Jin, "Path planning for UAV to cover multiple separated convex polygonal regions," *IEEE Access*, vol. 8, pp. 51770–51785, 2020, doi: 10.1109/ACCESS.2020.2980203.
[36] M. Höffmann, S. Patel, and C. Buskens, "Optimal coverage path planning for agricultural vehicles with curvature constraints," *Agriculture*, vol. 13, no. 11, p. 2112, Nov. 2023, doi: 10.3390/agriculture13112112.
[37] N. A. Kyriakakis, T. Stamadianos, M. Marinaki, and Y. Marinakis, "The electric vehicle routing problem with drones: An energy minimization approach for aerial deliveries," *Cleaner Logistics Supply Chain*, vol. 4, Jul. 2022, Art. no. 100041, doi: 10.1016/j.clscn.2022.100041.
[38] F. Semiz and F. Polat, "Solving the area coverage problem with UAVs: A vehicle routing with time windows variation," *Robot. Auto. Syst.*, vol. 126, Apr. 2020, Art. no. 103435, doi: 10.1016/j.robot.2020.103435.
[39] N. A. Kyriakakis, M. Marinaki, N. Matsatsinis, and Y. Marinakis, "A cumulative unmanned aerial vehicle routing problem approach for humanitarian coverage path planning," *Eur. J. Oper. Res.*, vol. 300, no. 3, pp. 992–1004, Aug. 2022, doi: 10.1016/j.ejor.2021.09.008.
[40] I. Hamdi, "Solving the cumulative capacitated vehicle routing problem with drones," *J. Ind. Prod. Eng.*, vol. 41, no. 4, pp. 344–361, May 2024, doi: 10.1080/21681015.2024.2304570.
[41] X. Li, N. Chen, H. Ma, F. Nie, and X. Wang, "A parallel genetic algorithm with variable neighborhood search for the vehicle routing problem in forest fire-fighting," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 1, pp. 1–17, Sep. 2024, doi: 10.1109/TITS.2024.3395930.
[42] K. Phalapanyakoon and P. Siripongwutikorn, "Route planning of heterogeneous unmanned aerial vehicles under recharging and mission time with carrying payload constraints," *J. Ind. Eng. Manage.*, vol. 16, no. 2, p. 215, May 2023.
[43] M. Sajid, A. Jafar, and S. Sharma, "Hybrid genetic and simulated annealing algorithm for capacitated vehicle routing problem," in *Proc. 6th Int. Conf. Parallel, Distrib. Grid Comput. (PDGC)*, Nov. 2020, pp. 131–136, doi: 10.1109/PDGC50313.2020.9315798.
[44] K. Vinh, S. Gebreyohannes, and A. Karimoddini, "An area-decomposition based approach for cooperative tasking and coordination of UAVs in a search and coverage mission," in *Proc. IEEE Aerosp. Conf.*, Mar. 2019, pp. 1–8, doi: 10.1109/AERO.2019.8741565.
[45] J. Mumtaz, K. A. Minhas, M. Rauf, L. Yue, and Y. Chen, "Solving line balancing and AGV scheduling problems for intelligent decisions using a genetic-artificial bee colony algorithm," *Comput. Ind. Eng.*, vol. 189, Mar. 2024, Art. no. 109976, doi: 10.1016/j.cie.2024.109976.
[46] A. N. Jasim, L. C. Fourati, and O. S. Albahri, "Evaluation of unmanned aerial vehicles for precision agriculture based on integrated fuzzy decision-making approach," *IEEE Access*, vol. 11, pp. 75037–75062, 2023, doi: 10.1109/ACCESS.2023.3294094.
[47] G. Avellar, G. Pereira, L. Pimenta, and P. Iscold, "Multi-UAV routing for area coverage and remote sensing with minimum time," *Sensors*, vol. 15, no. 11, pp. 27783–27803, Nov. 2015, doi: 10.3390/s151127783.
[48] S. U. Ngueveu, C. Prins, and R. Wolfler Calvo, "An effective memetic algorithm for the cumulative capacitated vehicle routing problem," *Comput. Operations Res.*, vol. 37, no. 11, pp. 1877–1885, Nov. 2010, doi: 10.1016/j.cor.2009.06.014.
[49] A. S. Silva, J. Lima, A. M. T. Silva, H. T. Gomes, and A. I. Pereira, "Time-dependency of guided local search to solve the capacitated vehicle routing problem with time windows," in *Proc. Int. Conf. Optim., Learn. Algorithms Appl.*, 2024, pp. 93–108.
[50] W. Sun, J.-K. Hao, X. Lai, and Q. Wu, "Adaptive feasible and infeasible Tabu search for weighted vertex coloring," *Inf. Sci.*, vol. 466, pp. 203–219, Oct. 2018, doi: 10.1016/j.ins.2018.07.037.
[51] T. Guilmeau, E. Chouzenoux, and V. Elvira, "Simulated annealing: A review and a new scheme," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Jul. 2021, pp. 101–105, doi: 10.1109/SSP49050.2021.9513782.
[52] M. Avci, "An effective iterated local search algorithm for the distributed no-wait flowshop scheduling problem," *Eng. Appl. Artif. Intell.*, vol. 120, Apr. 2023, Art. no. 105921, doi: 10.1016/j.engappai.2023.105921.
[53] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.
[54] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022, doi: 10.1109/ACCESS.2022.3142859.
[55] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011, doi: 10.1016/j.swevo.2011.02.002.

ALI NAJM JASIM received the B.Sc. degree in software engineering from the University of Imam Ja'afar Al-Sadiq, Baghdad, Iraq, in 2012, and the M.Sc. degree in computer science/information technology from Upsi University, Malaysia, in 2018. He is currently an Engineer with the Foundation of Alshuhda, Iraq. His research interests include education application, decision-making, artificial intelligence, and medical informatics.

LAMIA CHAARI FOURATI is currently an Associate Professor with the Computer Science and Multimedia Higher Institute. Her scientific publications have met the interest of the scientific community and her work has been published in very good journals and conferences. Her research interests include the conception and validation of new protocols and mechanisms for emerging network technologies, digital telecommunication networks, and the applications that impact positively their society through the conception of innovative protocols and mechanisms for health monitoring, remote systems monitoring, environmental control, and energy-saving approaches.