Applied Computer Science 2025, vol. 21, no. 2, pp. 1–17
https://doi.org/10.35784/acs_6872

Submitted: 2024-11-30 | Revised: 2025-02-01 | Accepted: 2025-04-07

Keywords: drone, UAV, path planning, task scheduling, complex networks

Ahmed KAMIL [1*], *Basim MAHMOOD* [1]
[1] University of Mosul, Iraq, ahmed.22csp21@student.uomosul.edu.iq, bmahmood@uomosul.edu.iq
[*] Corresponding author: ahmed.22csp21@student.uomosul.edu.iq

Integrating path planning and task scheduling in autonomous drone operations

Abstract

The efficiency and adaptability of drone operations depend heavily on two critical components: path planning and task scheduling. While the literature provides extensive research on these algorithms independently, there is a severe lack of studies addressing their combined impact on drone performance. Hence, this study aims to bridge this gap by developing a comprehensive framework that integrates three path planning algorithms (Spiral, Boustrophedon, and Hybrid) with four task scheduling algorithms (First-Come First-Served (FCFS), Shortest Processing First (SPF), Earliest Deadline First (EDF), and Priority). The hybrid path planning algorithm is proposed for this work. The framework evaluates each combination's performance based on key metrics, including elapsed time and energy consumption. A virtual simulation environment is designed and implemented for the sake of this study. The results show that combining the SPF scheduling algorithm with Hybrid path planning offers the best balance between time efficiency and energy consumption. The Boustrophedon path planning method shows the highest elapsed times and is generally less efficient than Hybrid and Spiral.

1. INTRODUCTION

1.1. Research overview

The use of drones has expanded significantly beyond recreational activities. These activities have permeated industries such as logistics, agriculture, surveillance, and disaster response (Hassanalian & Abdelkefi, 2017). As drones are integrated into these critical operations, the efficiency and sustainability of their performance become paramount. Path planning and task scheduling algorithms are key factors in drone performance. These factors play a significant role in the consumption of resources, which are limited to drones (Wang et al., 2023).

On the one hand, path planning is considered a critical aspect of drone operations. It ensures that drones fly efficiently and safely from source to target. One of the main goals of path planning algorithms is to optimize flight paths and ultimately minimize energy consumption, avoid obstacles, and meet mission-specific constraints (e.g., time or coverage requirements) (Aggarwal & Neeraj, 2020). Path planning algorithms, such as Boustrophedon and Spiral, attempt to guide drones to navigate and survey areas of interest with minimum cost (e.g., time and energy). On the other hand, task scheduling in drones involves assigning specific tasks within certain time constraints. The assignment process should take into account the limitations of the drones' resources, such as battery life, payload capacity, and computing power. The primary purpose of task scheduling is to improve the operational efficiency of drones, reduce mission completion times, and conserve energy. This is important in scenarios where drones operate autonomously in dynamic environments (Pasha et al., 2022). Scheduling algorithms have different strategies for prioritizing tasks. Variables such as task complexity, urgency, and drone capacity affect their performance.

The metrics used to evaluate the appropriateness of a path planning algorithm vary depending on the application. Metrics such as path length, time, energy consumption, memory consumption, coverage, and safety can be considered (Reda et al., 2024). In addition, the main challenges that drone path planning can struggle with (Chiang et al., 2020; Liu et al., 2024):

- Energy constraints: The limited resources of the drones make it necessary for the drones to navigate with efficient routes.

– Dynamic environment: Changes in the environment during operation cause the drone to continuously change its paths.
– Computational efficiency: During missions, drones must make decisions in real time, requiring efficient and accurate computation.

Furthermore, the role of task scheduling algorithms is to manage drone resources and efficiently adapt to dynamic environments. Thus, task scheduling algorithms face several challenges such as (Mao et al., 2024; Yang et al., 2025):

– Dynamic Task Arrival: This problem consumes the drone's energy because as tasks are added during missions, real-time rescheduling is required to reprioritize them.
– Dynamic environment: Any change in the environment should be taken into account during missions, with the goal of not missing any deadline for tasks, which is challenging.
– Resource constraints: Limited drone resources complicate the challenge, especially given the dynamic nature of drone missions during operations.

Both path planning and task scheduling in drones face several interrelated challenges that significantly impact their efficiency. These challenges require the development of robust, adaptive approaches and frameworks that can simultaneously address the requirements of efficient navigation and task execution.

1.2. Literature review

The literature on drone path planning and task scheduling highlights significant advances in improving the efficiency of drone operations. In path planning, early studies focused on deterministic methods such as Dijkstra's and A* (Mantoro et al., 2021). These two algorithms guarantee optimal paths in static environments. They were later extended by probabilistic methods such as Probabilistic RoadMap (PRM) and Random-exploring Random trees (RRT) to handle complex and high-dimensional spaces (Hüppi et al., 2022). In addition, other later work has developed hybrid approaches that combine different methods to ensure efficiency and real-time adaptability in dynamic environments. In addition, task scheduling has evolved from simple algorithms such as First-Come First-Served (FCFS) and Shortest Process First (SPF) to more advanced methods such as Earliest Deadline First (EDF) and Priority Scheduling. Other studies in the literature have explored the integration of path planning and task scheduling to address a variety of challenges (energy and real-time decision making). A study by Sung et al. (2019) presented a novel scheduling system for drone operations. They attempted to address the complexity and uncertainty inherent in drone operations and missions. The study introduced a flexible scheduling framework built with modular algorithms and used a simulation-based approach to evaluate scheduling solutions in dynamic environments. The key aspect of their research was the development of a prototype that demonstrated the feasibility and effectiveness of the proposed system. The study made several significant contributions to the field, such as the proposed conceptual design of the scheduling system and the identification of fundamental scheduling challenges in drone operations. Using a simulation-based method, their work provided a realistic assessment of scheduling performance under varying operational conditions (e.g., weather and obstacles). Duan et al. (2020) proposed a dynamic fault-tolerant task scheduling model (DSM-FNA) within a flexible network architecture (FNA) specifically designed for drone clusters. Their proposed architecture addresses the limitations of traditional drone networks by improving adaptability, flexibility, and resilience in dynamic environments. Their model used the Flexible Dynamic Scheduling Algorithm (FDSA) to optimize scheduling by dynamically adjusting drone capabilities and task requirements. The results showed that FDSA outperformed classical algorithms such as the Max-Min algorithm, especially during emergencies, by reducing task completion time and system communication load. The results also proved that DSM-FNA improved system flexibility and adaptability, while FDSA improved scheduling efficiency in real-time operation.

Other studies used different approaches to task scheduling, such as the study by Qin et al. (2021). The study addressed the challenge of task selection and scheduling in drone-enabled multiaccess edge computing (MEC). It attempted to solve this problem by formulating the ASSUMER problem, which is "a mixed-integer nonlinear programming (MINLP) problem that includes both integer and continuous variables and has been shown to be NP-hard", Qin et al. (2021). They proposed a Reconnaissance Task Scheduling Algorithm (RTSA) that transformed the problem into a binary integer programming problem. This allowed for more efficient and practical solutions. The RTSA algorithm was proved to be polynomial-time optimal and is supported by a bicriterion approximation guarantee of (1-e(-1))/2. Extensive simulations showed that their proposed algorithm significantly improves the overall reconnaissance utility and energy efficiency compared to other algorithms in the literature. The work of Khosiawan et al. (2019) developed a drone scheduling system for indoor environments. The system aimed to address a significant gap in current drone research. It categorized tasks into single and compound inspections and material handling tasks such as pickup, transport, and release. Their proposed method incorporated a heuristic-based approach using Particle Swarm Optimization (PSO) to efficiently generate high-quality schedules, which is critical for real-time drone operations. The results showed that the system was stable over different task datasets. Furthermore, You et al. (2022) explored the optimization of task scheduling and resource allocation in drone-enabled MEC systems. Their research introduced an iterative algorithm that minimized energy consumption and task completion time by decoupling and solving subproblems using block coordinate descent methods. The problem was mathematically formulated as a mixed-integer non-convex optimization focused on the joint management of drone trajectory, task scheduling, and resource allocation in Flying Ad-Hoc Networks (FANETs). The results showed a significant reduction in energy consumption, up to 71.9%, compared to other approaches in the literature. Another work proposed by Niu et al. (2022) focused on optimizing energy consumption and maintaining queue stability in drone networks. The study attempted to address the challenge of long-term task queue stability. It aimed to ensure that sudden task surges do not overwhelm drone operations. A two-step approach was proposed to minimize energy consumption by controlling the distribution of computational tasks between drones and mobile nodes. The researchers decoupled queue stability and energy consumption using Lyapunov optimization techniques. This technique allowed the researchers to find a tradeoff between system performance and energy efficiency. Simulations validated the effectiveness of these algorithms. It showed a significant level of energy savings of over 50% compared to traditional methods. As for the drone path planning literature, Du et al. (2022) developed an approach for drone coverage path planning. The approach was based on tight-envelope algorithms to obtain the optimal coverage area. The Jarvis method was used to keep the target, and then the invalid search area was removed to limit the total search area. A concave-convexity adaptive algorithm was also used to generate a drone search path. The results showed that their proposed approach efficiently reduced the invalid search area and minimized the turn frequency through boundary smoothing. Liu H. et al. (2024) proposed an approach to balance the social value of drones from one side and the power consumption and efficiency from the other side. The study included population distribution density and wind speed to create a well-established mathematical model for drones. The model reduced noise, threat to the public, and energy consumption in the trajectories of the drones. The study managed many issues simultaneously during drone operations. Therefore, it was considered multi-objective. Finally, Zhang et al. (2021) proposed a collision-free path planning approach for drones based on 3D voxel jump point search (JPS), which could work in complex urban areas. The approach also used Markov Decision Process (MDP) to dynamically avoid obstacles in real time. The results showed the efficiency of their approach compared to other approaches in the literature.

1.3. Problem statement and contribution

According to the literature, drones must navigate efficiently (path planning) while executing tasks in an effective sequence (task scheduling) to achieve optimal performance. As described in the previous section, the literature has studied these two aspects independently and has provided numerous algorithms for path planning and task scheduling. However, a critical gap remains in the study of their integration. This integration is crucial to answer the question: How do certain task scheduling algorithms perform when paired with certain path planning methods, and vice versa? Answering this question requires a comprehensive framework that provides developers with clear guidance on the optimal combinations of algorithms to maximize efficiency, minimize energy consumption, and meet operational constraints. Filling this gap is essential for improving the overall performance of drone systems. Therefore, the contribution and novelty of this work is to introduce a framework that integrates three drone path planning algorithms, Spiral, Boustrophedon, and Hybrid, with four task scheduling algorithms (FCFS, SPF, EDF, and Priority). Hybrid path planning is proposed in this thesis. It is a combination of Boustrophedon and Spiral algorithms. In addition, the combinations of path planning algorithms and task scheduling algorithms are efficiently evaluated using metrics such as energy and time consumption. This work provides practical recommendations for selecting algorithm pairs based on specific mission requirements, such as time-critical operations or energy conservation. It should be noted that this type of work (combining path planning and scheduling algorithms) is considered a severe lack in the literature. Also, the inclusion of concepts inspired by complex networks makes this work innovative compared to the literature.

The remaining sections are as follows: Section 2 presents the research methodology, including a description of the environment and simulation design, path planning algorithms, task scheduling algorithms, and evaluation metrics. Section 3 presents the experimental results along with the discussion. Finally, this work is concluded and the future directions are presented in Section 4.

2. RESEARCH METHOD

2.1. Simulation environment and representation

This section describes the design of the simulation environment and how it is transformed into a network model. The environment is designed to be a square of 1000000^{m2}, divided into four square sub-areas, each of 250000^{m2}. In addition, each of these subareas contains 25 cells of 10000^{m2}. The division of the environment into four sub-areas is based on the scenarios considered in this work. Each cell is a node, while the possible paths to the surrounding cells represent the edges. Collecting these nodes and edges results in a network model that contains 100 nodes (cells). Figure 1 shows the environment in the form of a (10x10) grid, and Figure 2 shows the network model of the simulation environment. It should be mentioned that the edges of the network model represent the trajectories that drones can navigate from one cell to another. Furthermore, network measures such as diameter, shortest path, and closeness centrality are computed. The path planning method used in this thesis is offline. Therefore, the computations aim to provide the drones with a complete view of the simulation environment during operation. The measurements used in the path planning algorithms of this work can be described as follows (Albert & Barabasi, 2002; Alfathe et al., 2025; Mahmood et al., 2020):

- Diameter: represents the distance between the two most distant nodes in the network model.
- Shortest Path: This is the shortest path between all network pairs.
- Closeness Centrality: It expresses how close a node is to other nodes in the network model. More specifically, the closeness of a node can be obtained from the average lengths of the shortest paths to other nodes within the network model. It can be calculated using the following equation:

$$Closeness\ (b) = \frac{N-1}{\sum_a d(a,b)} \tag{1}$$

where N represents the number of nodes in the network, $d(a,b)$ represents the distance between the pair (a, b).

In addition, all metric calculations are stored in the drone's memory for use in subsequent operations. The simulation environment is designed to mimic a real-world situation, representing an area that drones need to monitor. Unreal Engine 4.27 (Sanders, 2016) and Microsoft AirSim (Shah et al., 2018) are used to design and simulate the experiments; these tools have proven to be suitable for such work. The design can simulate the environment, the drones, and all the requirements of the experiments. These tools can simulate complex graphics, physical objects, and weather conditions. Each drone in the design is equipped with sensors and cameras. The Python 3.11.5 programming language and the Anaconda 2.5 platform are used. The hardware used to run the simulations represents a workstation computer with the following specifications CPU Intel(R) Core(TM) i9-8950HK 2.90GHz, 32GB RAM, and 4GB GPU.

1	2	3	4	5	6	7	8	9	10
36	37	38	39	40	41	42	43	44	11
35	64	65	66	67	68	69	70	45	12
34	63	84	85	86	87	88	71	46	13
33	62	83	96	97	98	89	72	47	14
32	61	82	95	100	99	90	73	48	15
31	60	81	94	93	92	91	74	49	16
30	59	80	79	78	77	76	75	50	17
29	58	57	56	55	54	53	52	51	18
28	27	26	25	24	23	22	21	20	19

Fig. 1. Environment grid and partitions.

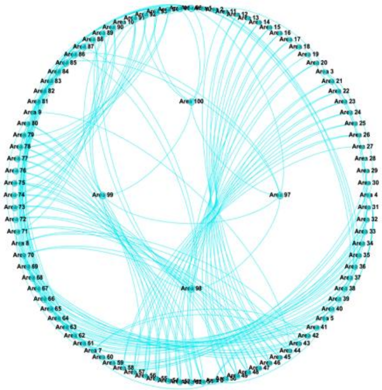

Fig. 2. Environmental network visualization, where each node represents an area in the environment and the edges represent the possible paths between them.

2.2. Path planning algorithms

2.2.1. Boustrophedon path planning algorithm

This path planning approach is mainly used by dynamic objects such as drones and robots to explore static and dynamic environments. Boustrophedon is inspired by how an ox follows a path when plowing the fields (agricultural areas). In addition, boustrophedon defines a particular sequence in an environment. It then follows this sequence to accomplish a mission of interest (Bähnemann et al., 2021). In the context of this work, the algorithm represents the simulation environment as a grid containing subareas (cells). The cells in the simulation environment have specific coordinates(x, y). A drone performs surveillance tasks and is responsible for a specific area within the simulation environment. The Start Point*(SP)* and End Point*(EP)* are represented by the following:

$$SP = (x_{Start}, y_{Start}), \tag{2}$$

$$EP = (x_{End}, y_{End}), \tag{3}$$

where x and y are the coordinates of the cells in the environment. The algorithm iterates through the cells for the drone to determine the following points during operation. The directional points can be defined as

$$PP = (x_i, y_i), \tag{4}$$

The generation of paths depends on the connection between the current point(x_i, y_i) and the neighboring point(x_{i+1}, y_{i+1}). The generated paths are considered as the flight sequence of the drone. The optimal path without obstacles is based on the "*back and forth*" of the boustrophedon. In case of obstacles, the optimal path is determined based on the $A*$ algorithm as described in Algorithm 1 (Zhang et al., 2022). The cost function $f(n)$ for a drone is identified by the cost of the drone to reach the endpoint and can be expressed as follows:

$$f(n) = g(n) + h(n) \tag{5}$$

where $g(n)$ represents the cost of arriving at cell n from SP, which is the sum of the costs of the paths to reach n. Meanwhile, $h(n)$ represents the heuristic estimate (e.g., Euclidean distance) of the cost from cell n to the final destination cell. Finally, the dynamics of the path (e.g., height or speed) are considered when defining the optimal path.

Algorithm 1: Steps of A* algorithm for selecting the optimal path.
Step 1: **Initialize** two sets of cells; Open Set (*OS*) and Closed Set (*CS*)
Step 2: **Add** *SP* to the *OS* with $f(SP) = g(Start) + H(Start)$
Step 3: **While** *OS* is Empty **Do**
    Select $n \in OS$ with minimum cost
    IF *n* is the destination cell, **Then**
        Path is found
    Else
        consider n's neighbors (n_b)
        IF ($n_b \notin OS$) **OR** (cost to reach n_b < previous cost) **Then**
            Update its cost
            Add it to *OS*
            Move the selected cell to the *CS*
        EndIF
    EndIF
EndWhile
Step 4: **IF** destination cell is reached **Then**
    Backtrack from destination cell and reconstruct the optimal path
EndIF
Step 5: **END**

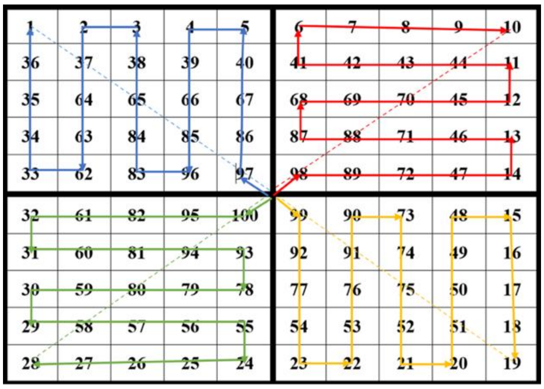

Fig. 3. Boustrophedon algorithm. The dotted lines are drone paths to return to SP, while the continuous lines are the regular paths

2.2.2. Spiral path planning algorithm

It is used to plan paths for autonomous systems such as drones systematically. Paths generation is based on spirals outward/inward from start to final destination. It aims to cover a whole environment in a smooth, continuous, and efficient pattern (Mourya et al., 2024). The spiral algorithm is based on polar coordinates with the drone's position. These coordinates are represented as a radius (χ) and an angle (θ) and then converted to Cartesian coordinates (x, y). It is formalized as follows:

$$x = r \times \cos(\theta) \tag{6}$$

$$y = r \times \sin(\theta) \tag{7}$$

During operation, the next position is determined by the increments of the radius (Δx) and angle ($\Delta\theta$). When the radius reaches its maximum, the path ends. The steps of the spiral algorithm are described in detail in Algorithm 2.

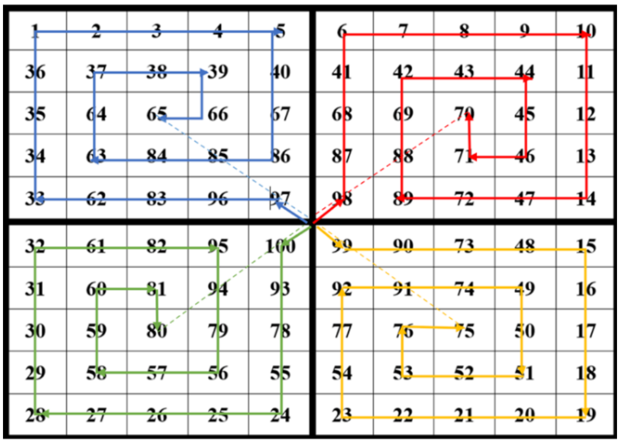

Fig. 4. Spiral path planning. The dotted lines are drone paths to return to SP, while the continuous lines are the regular paths

Algorithm 2: Steps of Spiral algorithm for path planning.

Step 1: **Define** the starting point (S), initial radius (χ), initial angle (θ), step size of $\Delta\chi$ and $\Delta\theta$, and end condition.
Step 2: **Increment** radius and angle

$$\chi = \chi + \Delta\chi$$
$$\theta = \theta + \Delta\theta$$

Step 3: **Convert** polar coordinates to Cartesian coordinates using Equation 6 and 7.
Step 4: **Store** x and y as a point in drone's path.
Step 5: **IF** end condition **THEN**
    Stop
  Else
    Go to Step 2
Step 6: **END**

2.2.3. Hybrid trajectory planning: Proposed algorithm

The proposed path planning algorithm starts with a spiral pattern until it reaches the maximum radius and follows a boustrophedon pattern. This strategy minimizes the energy consumption of the drone and the elapsed time to complete the tasks. The proposed hybrid strategy is described in Algorithm 3.

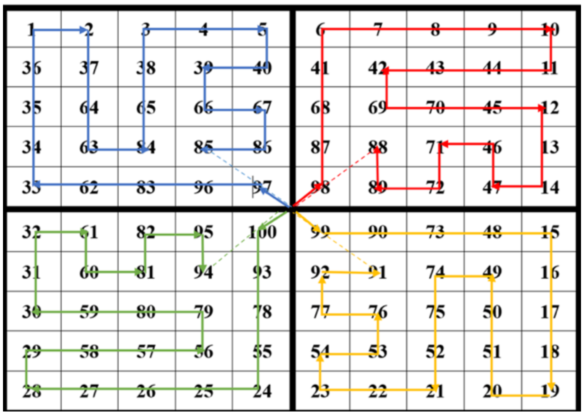

Fig. 5. Hybrid planning (proposed): The dotted lines are drone paths to return to SP, while the continuous lines are the regular paths

Algorithm 3: Steps of the proposed hybrid path planning algorithm.
Step 1: **Define** environment areas to be covered by drones
Define the *starting point*
Define the maximum *radius*
Define Boustrophedon strip width
Upload environment network model and its calculations (diameter, shortest paths, and closeness centrality) into the drone

Phase 1: Spiral Path Planning

Step 2: **Initialize** radius (χ), initial angle (θ) for Spiral path
Define the increments of radius ($\Delta\chi$) and angle ($\Delta\theta$)
Step 3: **IF** Obstacle **Then**
Move towards node C; C is one of the neighbor nodes that has highest closeness centrality
Else
Calculate next node on the Spiral path based on:
$\chi = \chi + \Delta\chi$; depends of the shortest path between the current node and the next
$\theta = \theta + \Delta\theta$
Convert polar coordinates to Cartesian coordinates based on Equations 6 and 7.
Store (x, y) coordinates as a node on the path.
Step 4: **IF** Radius = Diameter **THEN** ; reaching the maximum radius
Transition to Boustrophedon path planning (*Step 5*).
Else
Go to Step 3

Phase 2: Boustrophedon Path Planning

Step 5: **Identify** the remaining not covered nodes in the environment
SET SN = Diameter; where SN = (x_{start}, y_{start}) ; SN denotes coordinates of the start node.
Define the boundaries of the strip (w)
Step 6: **Repeat** back-and-forth trajectory planning for each strip in the environment
IF Obstacle **Then**
Move towards node C; C is one of the neighbors nodes that has highest closeness centrality
Else
Move in a straight line in one direction, and;
Let $y_{current} = y_{start} + (k \times w)$; where k is the strip index
Let $x_{start} = x_{start} + l$; where l is the length of strip
Turn around
Move back in the opposite direction
Continue Until the strip is covered
IF all strips not covered **Then**
Move to the next strip and repeat the processes in *Step 6*
Else
Combine the trajectories of phase 1 and phase 2.
Store the trajectory
Step 7: **END**

2.2.4. Scheduling algorithms

Task scheduling is critical in drone operations to optimize task completion while minimizing resource consumption. Scheduling also plays an important role in emergency drone operations. This thesis considers four known algorithms for task scheduling (Liu, 2006; Al-Kateeb & Abdullah, 2024). All scheduling algorithms in this work are preemptive because a task can be interrupted (Liu, 2006; Hasan & Al-Rizzo, 2020).

First Come First Serve (FCFS): Considered the simplest scheduling algorithm, it executes tasks based on their exact order of arrival. It does not consider the urgency of the tasks (e.g., priority or deadline). It is used in drone missions when the tasks are equally important and their arrival order represents the execution order. The completion time Ci for a task i in FCFS is formalized as follows:

$$C_i = \sum_{j=1}^{i} T_j \tag{8}$$

where T_j denotes the execution time of task j.

Earliest Deadline First (EDF): It schedules tasks based on their deadlines; that is, a task is performed with a priority that is the earliest deadline (Hadeed & Abdullah, 2021). Thisapproach is dynamic, as the priority changes as deadlines approach or new tasks arrive. This algorithm is used in drone missions when time is of the essence. For example, delivering medical supplies to specific areas in disaster situations is a high priority. The schedulability test of this algorithm can be formalized as follows:

$$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1 \tag{9}$$

where n represents the number of tasks.

Shortest Process First (SPF) prioritizes tasks based on the shortest processing time. The goal is to minimize the average completion time. This approach is also known as Shortest Job First (SJF). In drones, it is used to maximize throughput in missions. The average completion time can be expressed as:

$$Avg(C_i) = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{10}$$

Priority: This algorithm is useful for drone missions when there are tasks of mixed priority. A task i is necessarily scheduled before task j if:

$$P_i > P_j \tag{11}$$

2.2.5. Evaluation metrics

This paper will be judged on the following criteria:

- Energy: The energy consumed during the mission.
- Time: The time spent on missions.

The flight altitude that the drones follow in the simulations is 40 meters (see Figure 6). Our experiments have shown that this particular height is more efficient because it has fewer obstacles to overcome in the environment. For accuracy purposes, each experiment is run five times and then the average is considered.

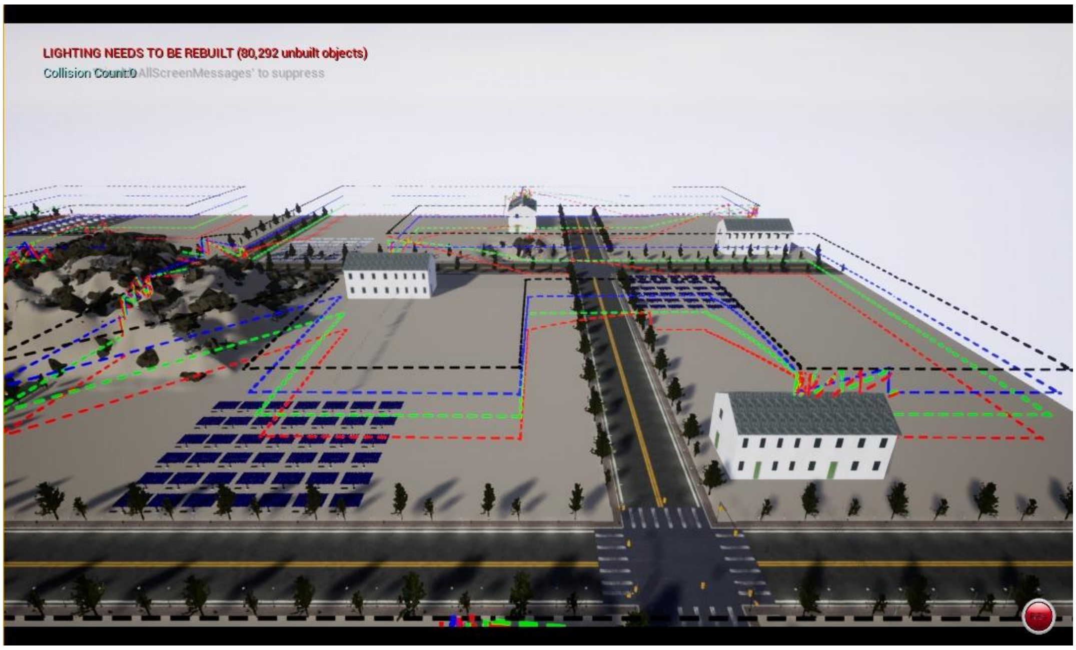

Fig. 6. Different heights are used by drones during operations. The red dotted lines represent a flying height of 10 meters, the green dotted lines represent a flying height of 20 meters, the blue dotted lines represent a flying height of 30 meters, and the black dotted lines represent a flying height of 40 meters. The 40 meters of height avoid more obstacles than the other heights. Therefore, this work considers it the optimal height for this environment

2.2.6. Research workflow

Figure 7 illustrates the workflow of this research and describes the detailed steps as follows:

- Establish a connection between the AirSim simulator and the Unreal Engine environment.
- Upload the positions of the subareas into the environment. The drones will visit these subareas.
- Determine the required flight altitude that the drones will use for all missions.
- Specify the path planning algorithm (Spiral, Boustrophedon, or Hybrid) that the drones will use during operations. Regardless of the selected algorithm, a drone will return to its home position when it completes its mission.
- The drone flies autonomously according to the coordinates (X, Y, Z) given by the path planning algorithm to survey, cover or monitor the area.
- The emergency task buffer is periodically checked. The buffer is designed to receive emergency tasks from the ground station and store them in real time.

- If the buffer has emergency tasks, the drone selects one based on the selected scheduling algorithm (FCFS, EDF, SPF, or Priority). The current task is interrupted and its information (e.g., position) is maintained with the goal of completing it later.
- Distance sensors and LiDAR are used to navigate around obstacles during the mission.
- The drones move using yaw and pitch as the drones in this work use a down camera.
- After completing all emergency tasks (e.g., the buffer is empty), the drone restores the maintained unfinished (non-emergency) tasks to complete them.

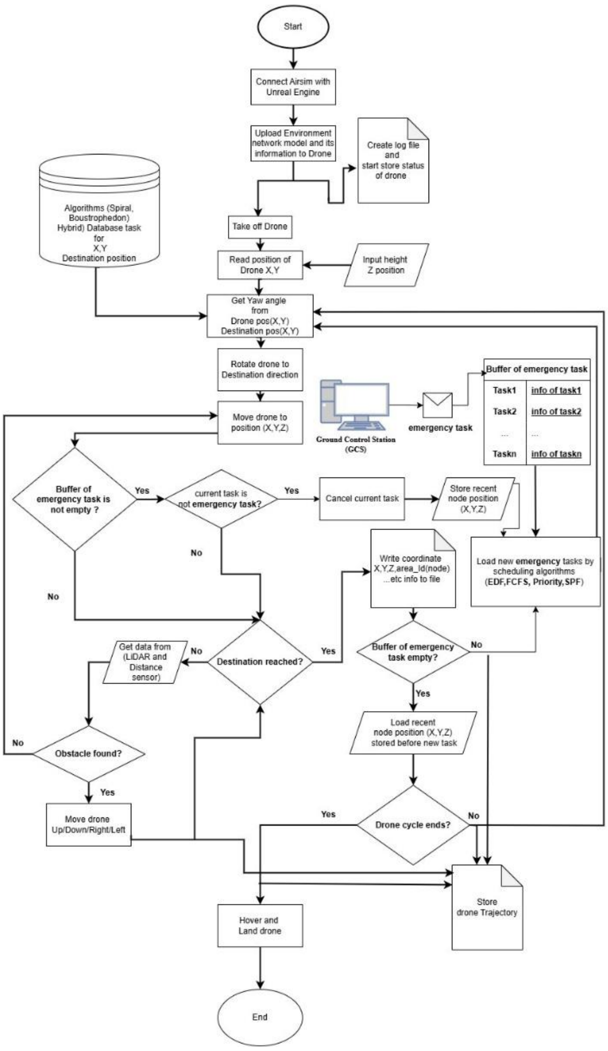

Fig. 7. The proposed workflow diagram

3. RESULTS AND DISCUSSIONS

3.1. Results

This section describes the simulation results. It is worth noting that the simulation environment is considered static and the path planning is offline. As mentioned above, the scheduling type is preemptive because a task can be interrupted when emergency tasks arrive. On the other hand, emergency tasks cannot be interrupted due to their criticality. The general settings of AirSim are described in Table 1.

Tab. 1. General settings of AirSim that are used in the simulations

Element	Parameter	Description	
SimMode	Multirotor	Use drone simulation.	
ViewMode	NoDisplay	This will freeze rendering and consume workstation power.	
ClockSpeed	1	This value means that the simulation clock has 1 second, which is the same as the real clock..	
Vehicles	Drone 1, Drone 2, Drone 3, Drone 4	The list of drones involved in the simulations.	
Vehicle_type	SimpleFlight	Drones (Quadcopter)	
Sensors	Camera	Camera: has five directions	front_center = True
			front_right = False
			front_left = false
			bottom_cente = True
			back_center = False
	Distance	Distance Sensor: it points to the front of the drone.	X Y Z = (0, 0, -1)
			Yaw Pitch Roll = (0, 0, 0)
			MinDistance = 0.2
			MaxDistance = 40
	LiDAR		NumberOfChannels = 64
			Range = 4
			PointsPerSecond = 100,000
			HorizontalFOVStart = -180
			HorizontalFOVEnd = 180
			VerticalFOVUpper = 90
			VerticalFOVLower = -90
			X Y Z = (0, 0, -0.1)
Environment Conditions	Wind	The wind speed values are based on the Remote Sensing Center at the authors' university. The unit used is meters per second.	Moderate Winds = 8 m/s Strong Winds = 13 m/s High Winds = 18 m/s

As mentioned above, the scheduling algorithms used in this work are SPF, FCFS, Priority, and EDF. Three path planning algorithms are used: Spiral, Boustrophedon, and Hybrid (proposed). The focus is on two key performance metrics: elapsed time and energy consumption. The results include 12 combinations of task scheduling and path planning. Each combination contains one path planning and one task scheduling approach.

As for the elapsed time evaluation, the combinations are analyzed and evaluated. This analysis aims to find the best integration between the task scheduling and path planning algorithms. Figure 8 shows the descriptive statistics of the elapsed time across scheduling and coverage path planning.

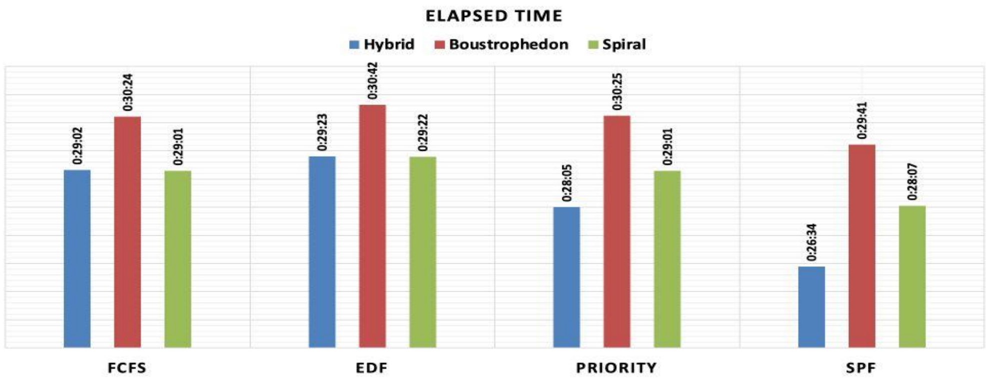

Fig. 8. Elapsed time obtained from the scheduling and coverage path planning algorithms

As observed, the hybrid coverage path planning achieves the lowest elapsed times. On the other hand, Boustrophedon shows the highest times, and Spiral is considered intermediate, but closer to the hybrid algorithm. Similar behavior is observed when evaluating battery consumption using the same combinations of algorithms (see Figure 9).

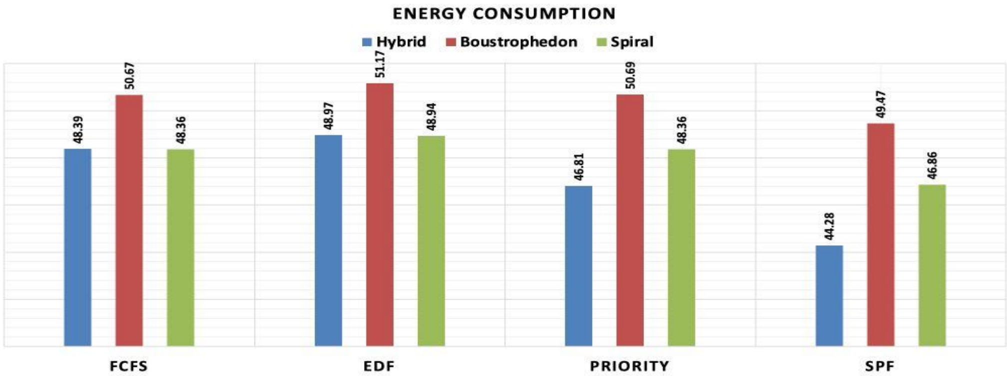

Fig. 9. Energy consumption obtained from the scheduling and coverage path planning algorithms

However, it is crucial to test the variations of the combination to evaluate its stability. For this purpose, box plots are generated considering all runs of the experiments. Each box plot shows five quartiles (min, lower, middle, upper, and max). In terms of elapsed time, Figure 10 shows that almost all combinations reflect stable behavior because the ranges of the quartiles are close to each other. However, the Hybrid-SPF combination shows more stable behavior than the other combinations. Also, the median (middle quartile) of the Spiral algorithm performs better than the other combinations in all combinations. Similar behavior is obtained when evaluating battery consumption, as shown in Figure 11.

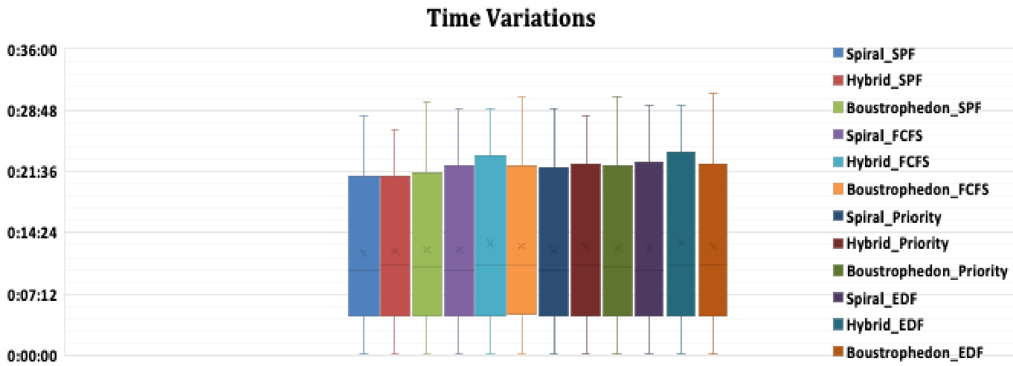

Fig. 10. Variations in the performance of the combinations between task scheduling and coverage path planning in terms of elapsed time

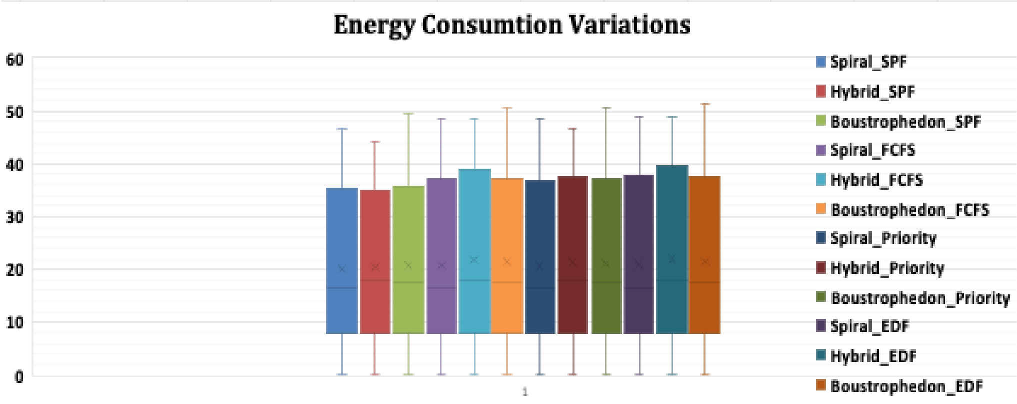

Fig. 11. Variations in the performance of the combinations between task scheduling and coverage path planning regarding energy consumption

In addition, based on the above results, further analysis and evaluation of the performance is required. Analysis of Variance (ANOVA) is performed to accurately evaluate the results and show whether the obtained results are statistically significant. The hypothesis test is performed on the mean values of the coverage path planning algorithms in terms of time consumption as follows:

- The null hypothesis (H0) states that the means of the coverage path planning algorithms are equal and can be formalized as follows:

$$\mu_{Hybrid} = \mu_{Boustrophedon} = \mu_{Spiral} \tag{12}$$

- The alternative hypothesis (H1) states that the means are not equal to each other:

$$\mu_{Boustrophedon} \neq \mu_{Spiral} \neq \mu_{Boustrophedon} \tag{13}$$

The confidence level used in this analysis is given by:

$$\text{Confedence Level} = \frac{95}{100} \rightarrow \alpha = 0.05 \tag{14}$$

The ANOVA results for Elapsed Time show that the *F-statistic* is 0.0089 and the p-value is 0.9911. These results reflect the fact that the difference between the three path planning methods (Hybrid, Spiral, and Boustrophedon) is not statistically significant. This is because the p-value is higher than 0.05 (significance level). This means that the three algorithms reflect similar variation behavior. However, the scheduling algorithms have a significant impact on the elapsed time.

Similarly, battery consumption is analyzed for each scheduling algorithm and coverage path planning method. The results show that hybrid coverage path planning consistently reflects lower energy consumption. On the other hand, Boustrophedon tended to consume more energy. Among the planning algorithms, SPF combined with Hybrid or Spiral reflected the most efficient battery usage.

As a final step in the analysis of this work, the proposed path planning algorithm is tested under different wind speeds to show its performance under different weather conditions. This test also adds a dynamic nature to the designed environment. Figure 12 shows the simulation environment from different angles, showing the performance of the drone using the proposed path planning algorithm as the wind speed varies. The red trajectories in the figure reflect the wind speed of 8 m/s, the green trajectories for 13 m/s, and the blue trajectories for 18 m/s. It is worth mentioning that these speeds are considered standard wind speeds during different seasons in the geographical areas of the researchers, based on the Remote Sensing Center in Mosul, Iraq. As can be seen, the trajectories are curved (not straight) due to the effect of the north wind on the drone. In addition, the slope of the trajectories is based on the wind speed used. This situation forces the drone to expend more effort (e.g., time and energy) to reach the target. Figure 13 shows the time consumed by a drone during operation. The time consumed by the drone increases as more areas are visited. Figure 14 shows the energy consumption of the drone. The battery consumption increases as more regions are visited within the simulation environment. Based on the two figures from, it can be observed that the drone performance decreased significantly when the wind speed exceeded 13 m/s. This means that there is a trade-off between wind speed and drone performance. Compared to the normal wind speed, the moderate (8 m/s) and strong (13 m/s) wind speeds have a slight difference. The high wind speed (18 m/s) has a significant impact on the performance of the drone and consumes a lot of flight time and energy.

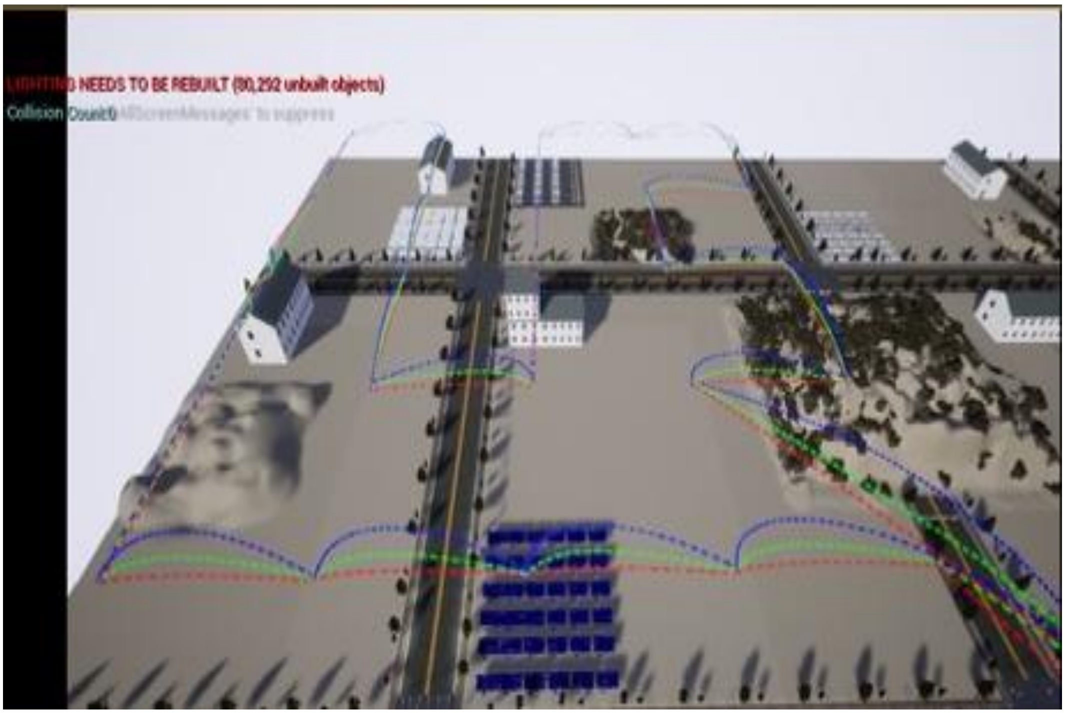

Fig. 12. Trajectories of the drone under different wind speeds using the proposed path planning algorithm

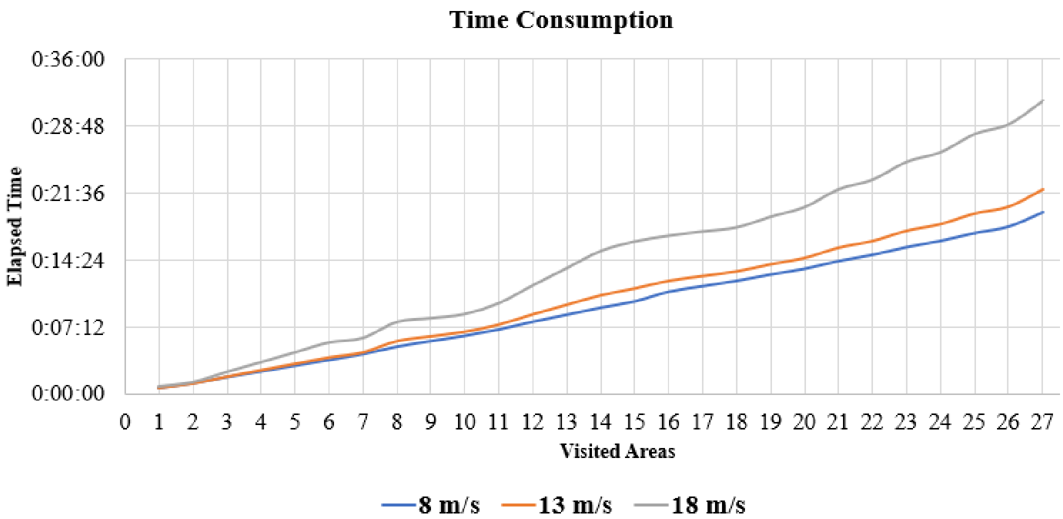

Fig. 13. Drone time consumption when facing wind speeds of 8, 13 and 18m/s using the proposed path planning algorithm. The x-axis represents the number of visited areas in the simulation environment, and the y-axis represents the time consumed in minutes

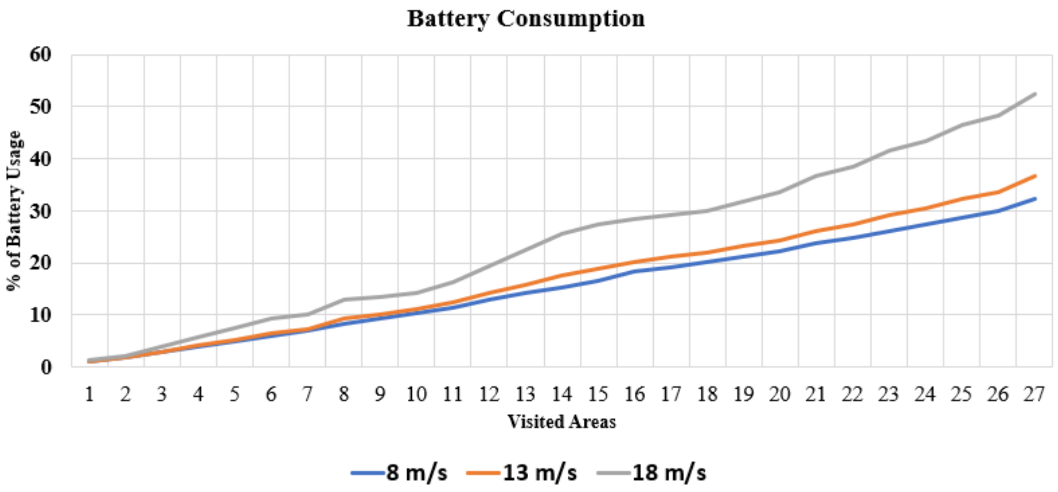

Fig. 14. Drone battery usage when facing wind speeds of 8, 13 and 18m/s using the proposed path planning algorithm. The x-axis represents the number of visited areas in the environment and the y-axis represents the battery usage percentage

4. DISCUSSION

According to the results, the innovative modeling of the environment as a network embedded in the drones reduced both the energy and time consumed by the drones. In other words, uploading the network model of the environment and its calculations into the drones plays an important role in extending the drones' vision to the environment. It enables the drones to make decisions more quickly. For example, when facing an obstacle, the closeness centrality is used to determine which direction is better in terms of distance. Since the closeness of a node is based on the shortest paths, the proposed algorithm successfully has a collision-free feature, which means that the number of collisions is zero in all simulations of the proposed approach. This makes the proposed path planning algorithm ideal for emergency and disaster applications or applications that require immediate response.

Furthermore, the combination of SPF scheduling with hybrid path planning consistently outperformed in terms of efficiency and energy consumption. However, when looking at the elapsed time for different combinations, there are discrepancies between the mean and median values. This is due to a small number of high latency cases where the algorithms encounter complex scenarios such as longer computation times for certain paths, the presence of obstacles that increase execution times, or the side effect of using certain task scheduling schemes such as EDF that can cause task bottlenecks that lead to longer execution times in certain runs. In addition, the results also show large standard deviations, which reflect high variability in elapsed time. In practice, this is expected due to the nature of drone path planning and task scheduling, such that EDF and priority scheduling sometimes create scheduling bottlenecks that extend execution time. In addition, the presence of obstacles can also lead to extreme values that contribute to high standard deviations. The locations of the obstacles, the target, and the direction of the drone can also be considered as causes. In addition, the results of testing the proposed path planning algorithm under different wind speeds showed that a high wind speed significantly affects the performance of the drones.

Recommendations for drone developers can be summarized as follows:

- Environment: The environment of interest should be accurately studied in terms of geography and other characteristics. Modeling the environment as a network model is efficient in supporting the decisions of the drones.
- Start and End Points: Determining the start and end points within an environment is a crucial aspect that should be studied in detail, as it plays a significant role in the consumption of drone resources.
- Objectives: Monitoring objectives should be well defined before an algorithm is adopted. This will help drone developers consider a wide range of potential events that may occur during drone operation.
- Mission Constraints: Special attention should be paid to constraints related to the drone's resources, the nature of the environment, and other mission requirements. Changes in the environment may occur during operations in dynamic environments such as weather conditions or when facing a flying object.

5. CONCLUSIONS

This work introduced a framework that integrates three drone path planning algorithms (Spiral, Boustrophedon, and Hybrid) with four task scheduling algorithms (FCFS, SPF, EDF, and Priority). Hybrid path planning, which is a combination of Boustrophedon and Spiral algorithms, was proposed in this work. The combinations of path planning and task scheduling algorithms were evaluated in terms of energy and time consumption. This work also highlights the critical role of combined scheduling and path planning algorithms in optimizing drone operations. The combination of SPF scheduling and hybrid path planning consistently outperformed in terms of efficiency and energy consumption. This makes it ideal for most applications, including emergency drone applications. Modeling the environment using network science approaches was highly efficient in minimizing drone resources. These findings are critical for drone developers seeking to optimize drone systems for time-sensitive missions or energy-constrained environments. The results highlight the need to adapt the algorithm based on operational priorities. One of the main limitations of the proposed algorithm is that it has not been tested in the presence of unexpected flying objects due to the lack of scenarios. For example, a bird attacks the drone during flight. Future work will explore the scalability of these results in more complex and dynamic operational scenarios. Also, more path planning and task scheduling algorithms can be involved and tested with different indicators and conditions.

Acknowledgments

The authors would like to thank the College of Computer Science and Mathematics and the ICT Research Unit at the University of Mosul, Iraq, for their support of this work.

Conflicts of Interest

This work does not have any conflict of interest to declare.

REFERENCES

Aggarwal, S., & Kumar, N. (2020). Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer communications*, *149*, 270-299. https://doi.org/10.1016/j.comcom.2019.10.014

Alfathe, M., Azhar, A., Ali Othman, A. J., & Kashmola, M. (2025). Adopting complex networks to detect cheat cases in electronic exams. *International Journal of Computing and Digital Systems*, *17*(1), 1-9. http://dx.doi.org/10.12785/ijcds/1571012424

Al-Kateeb, Z. N., & Abdullah, D. B. (2024). AdaBoost-powered cloud of things framework for low-latency, energy-efficient chronic kidney disease prediction. *Transactions on Emerging Telecommunications Technologies*, *35*(6), e5007. https://doi.org/10.1002/ett.5007

Bähnemann, R., Lawrance, N., Chung, J. J., Pantic, M., Siegwart, R. & Nieto, J., 2021. Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem. *Field and Service Robotics: Results of the 12th International Conference* (pp. 277-290). Springer Singapore. https://doi.org/10.1007/978-981-15-9460-1_20

Chiang, H.-T., HomChaudhuri, B., Smith, L., & Tapia, L. (2020). Safety, challenges, and performance of motion planners in dynamic environments. In N. M. Amato, G. Hager, S. Thomas, & M. Torres-Torriti (Eds.), *Robotics Research* (Vol. 10, pp. 793–808). Springer International Publishing. https://doi.org/10.1007/978-3-030-28619-4_55

Du, R., Zhao, T., Dong, X. & Zhu, R. (2022). September. An improved method for the UAV coverage path planning. *2022 4th International Conference on Robotics and Computer Vision (ICRCV)* (pp. 325-328). IEEE. https://doi.org/10.1109/ICRCV55858.2022.9953213

Duan, T., Wang, W., Wang, T., Chen, X., & Li, X. (2020). Dynamic tasks scheduling model of UAV cluster based on flexible network architecture. *IEEE Access*, *8*, 115448–115460. https://doi.org/10.1109/ACCESS.2020.3002594

Hadeed, W., & Abdullah, D. (2021). Real-time based big data and e-learning: A survey and open research issues. *AL-Rafidain Journal of Computer Sciences and Mathematics*, *15*(2), 225–243. https://doi.org/10.33899/csmj.2021.170044

Hasan, M. Z., & Al-Rizzo, H. (2019). Task scheduling in internet of things cloud environment using a robust particle swarm optimization. *Concurrency and Computation: Practice and Experience*, *32*(2), e5442. https://doi.org/10.1002/cpe.5442

Hassanalian, M., & Abdelkefi, A. (2017). Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, *91*, 99-131. https://doi.org/10.1016/j.paerosci.2017.04.003

Hüppi, M., Bartolomei, L., Mascaro, R., & Chli, M. (2022). T-prm: Temporal probabilistic roadmap for path planning in dynamic environments. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 10320-10327). IEEE. https://doi.org/10.1109/IROS47612.2022.9981739

Khosiawan, Y., Park, Y., Moon, I., Nilakantan, J. M., & Nielsen, I. (2019). Task scheduling system for UAV operations in indoor environment. *Neural Computing and Applications*, *31*(9), 5431–5459. https://doi.org/10.1007/s00521-018-3373-9

Liu, H., Tsang, Y. P. and Lee, C. K. M. (2024). A cyber-physical social system for autonomous drone trajectory planning in last-mile superchilling delivery. *Transportation Research Part C: Emerging Technologies*, *158*, 104448. https://doi.org/10.1016/j.trc.2023.104448

Liu, J. W. S. (2006). *Real-time systems*. Pearson Education.

Liu, S., Jin, Z., Lin, H., & Lu, H. (2024). An improve crested porcupine algorithm for UAV delivery path planning in challenging environments. *Scientific Reports*, *14*, 20445. https://doi.org/10.1038/s41598-024-71485-1

Mahmood, B., Sultan, N. A., Thanoon, K. H., & Khadhim, D. S. (2020). Collaboration networks: university of mosul case study. *AL-Rafidain Journal of Computer Sciences and Mathematics*, *14*(1), 117-133. https://doi.org/10.33899/CSMJ.2020.164679

Mantoro, T., Alamsyah, Z., & Ayu, M. A. (2021). Pathfinding for disaster emergency route using Sparse A* and Dijkstra Algorithm with augmented reality. *2021 IEEE 7th International Conference on Computing, Engineering and Design (ICCED)* (pp. 1-6). IEEE. https://doi.org/10.1109/icced53389.2021.9664869

Mao, X., Wu, G., Fan, M., Cao, Z., & Pedrycz, W. (2024). DL-DRL: A double-level deep reinforcement learning approach for large-scale task scheduling of multi-UAV. *IEEE Transactions on Automation Science and Engineering*, *22*, 1028-1044. https://doi.org/10.1109/TASE.2024.3358894

Mourya, G. S., Jami, A., Ajith, A., Seikh, A. H. & Kumar, J. S. (2024). Multi-Trajectory Drone surveillance: Amplifying monitoring capabilities in disaster management. *2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)* (pp. 1-6). IEEE. https://doi.org/10.1109/SCEECS61402.2024.10482271

Niu, Z., Liu, H., Lin, X., & Du, J. (2022). Task scheduling with UAV-assisted dispersed computing for disaster scenario. *IEEE Systems Journal*, *16*(4), 6429–6440. https://doi.org/10.1109/JSYST.2021.3139993

Pasha, J., Elmi, Z., Purkayastha, S., Fathollahi-Fard, A. M., Ge, Y.-E., Lau, Y.-Y., & Dulebenets, M. A. (2022). The drone scheduling problem: A systematic state-of-the-art Review. *IEEE Transactions on Intelligent Transportation Systems*, *23*(9), 14224–14247. https://doi.org/10.1109/TITS.2022.3155072

Qin, Z., Wang, H., Wei, Z., Qu, Y., Xiong, F., Dai, H., & Wu, T. (2021). Task selection and scheduling in UAV-Enabled MEC for reconnaissance with time-varying priorities. *IEEE Internet of Things Journal*, *8*(24), 17290–17307. https://doi.org/10.1109/JIOT.2021.3078746

Reda, M., Onsy, A., Haikal, A. Y., & Ghanbari, A. (2024). Path planning algorithms in the autonomous driving system: A comprehensive review. *Robotics and Autonomous Systems*, *174*, 104630. https://doi.org/10.1016/j.robot.2024.104630

Sanders, A. (2016). *An introduction to Unreal engine 4*. AK Peters/CRC Press.

Shah, S., Dey, D., Lovett, C. & Kapoor, A. (2018). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. *Field and Service Robotics: Results of the 11th International Conference* (pp. 621-635). Springer International Publishing. https://doi.org/10.1007/978-3-319-67361-5_40

Sung, I., Danancier, K., Delphine Ruvio, D., Guillemet, A., & Nielsen, P. (2019). A design of a scheduling system for an unmanned aerial vehicle (UAV) deployment. *IFAC-PapersOnLine*, *52*(13), 1854-1859. https://doi.org/10.1016/j.ifacol.2019.11.472

Wang, Z., Zhang, B., Xiang, Y., & Li, C. (2023). Joint task assignment and path planning for truck and drones in mobile crowdsensing. *Peer-to-Peer Networking and Applications*, *16*, 1668-1679. https://doi.org/10.1007/s12083-022-01389-2

Yang, W., Mao, Y., Chen, X., Chen, C., Lei, B., & He, Q. (2025). A method for simultaneously implementing trajectory planning and DAG task scheduling in multi-UAV assisted edge computing. *Ad Hoc Networks*, *166*, 103668. https://doi.org/10.1016/j.adhoc.2024.103668

You, W., Dong, C., Wu, Q., Qu, Y., Wu, Y., & He, R. (2022). Joint task scheduling, resource allocation, and UAV trajectory under clustering for FANETs. *China Communications*, *19*(1), 104–118. https://doi.org/10.23919/JCC.2022.01.009

Zhang, N., Zhang, M. and Low, K. H. (2021). 3D path planning and real-time collision resolution of multirotor drone operations in complex urban low-altitude airspace. *Transportation Research Part C: Emerging Technologies*, *129*, 103123. https://doi.org/10.1016/j.trc.2021.103123

Zhang, Z., Wu, J., Dai, J., & He, C., (2022). Optimal path planning with modified A-Star algorithm for stealth unmanned aerial vehicles in 3D network radar environment. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *236*(1), 72-81. https://doi.org/10.1177/09544100211007381