Path Planning for Unmanned Aerial Vehicles in Dynamic Environments: A Novel Approach Using Improved A * and Grey Wolf Optimizer

Ali Haidar Ahmad, Oussama Zahwe, Abbass Nasser, Benoit Clement

▶ **To cite this version:**

Ali Haidar Ahmad, Oussama Zahwe, Abbass Nasser, Benoit Clement. Path Planning for Unmanned Aerial Vehicles in Dynamic Environments: A Novel Approach Using Improved A * and Grey Wolf Optimizer. World Electric Vehicle Journal, 2024, 15 (11), 10.3390/wevj15110531 . hal-04790333

HAL Id: hal-04790333
https://hal.science/hal-04790333v1

Submitted on 19 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

World Electric Vehicle Journal

MDPI

Article

Path Planning for Unmanned Aerial Vehicles in Dynamic Environments: A Novel Approach Using Improved A* and Grey Wolf Optimizer

Ali Haidar Ahmad [1,*], Oussama Zahwe [2], Abbass Nasser [3] and Benoit Clement [1,*]

[1] Lab-STICC UMR CNRS 6285, ENSTA Bretagne, 29200 Brest, France
[2] Vectrawave Device, 221001 Lannion, France; oussama.zahwe@vectrawave.com
[3] Holy-Spirit University of Kaslik (USEK), Jounieh P.O. Box 446, Lebanon; abbass.nasser@ieee.org
* Correspondence: ali.haidar@ensta-bretagne.org (A.H.A.); benoit.clement@ensta-bretagne.fr (B.C.)

Abstract: Unmanned aerial vehicles (UAVs) play pivotal roles in various applications, from surveillance to delivery services. Efficient path planning for UAVs in dynamic environments with obstacles and moving landing stations is essential to ensure safe and reliable operations. In this study, we propose a novel approach that combines the A* algorithm with the grey wolf optimizer (GWO) for path planning, referred to as GW-A*. Our approach enhances the traditional A algorithm by incorporating weighted nodes, where the weights are determined based on the distance from obstacles and further optimized using GWO. A simulation using dynamic factors such as wind direction and wind speed, which affect the quadrotor UAV in the presence of obstacles, was used to test the new approach, and we compared it with the A* algorithm using various heuristics. The results showed that GW-A* outperformed A* in most scenarios with high and low wind speeds, offering more efficient paths and greater adaptability.

Keywords: unmanned aerial vehicle; A* algorithm; grey wolf optimizer; path planning; weighted graph

check for updates

Citation: Haidar Ahmad, A.; Zahwe, O.; Nasser, A.; Clement, B. Path Planning for Unmanned Aerial Vehicles in Dynamic Environments: A Novel Approach Using Improved A* and Grey Wolf Optimizer. *World Electr. Veh. J.* **2024**, *15*, 531. https://doi.org/10.3390/wevj15110531

Academic Editor: Leilei Zhao and Liguo Zang

Received: 9 September 2024
Revised: 11 November 2024
Accepted: 12 November 2024
Published: 18 November 2024

Copyright: © 2024 by the authors. Published by MDPI on behalf of the World Electric Vehicle Association. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The field of drone technology, also known as unmanned aerial vehicles (UAVs), has become highly attractive due to its relevance to a wide range of industries. Drones are now being used in different sectors such as delivery, which has gained a lot of attention and has become an effective marketing tool. For example, major companies like Amazon have launched projects such as Amazon Prime Air, which uses drones to deliver products [1]. The agriculture and conservation sectors also benefit from drones, as they are used for crop monitoring, pesticide spraying [2], wildlife monitoring, and anti-poaching efforts [3]. Today, emergency response teams use drones for real-time disaster assessment and management [4], and a proposed study explores the use of drones for search and rescue activities in underground mines [5].The construction industry leverages drones for site surveying and progress tracking [6].

With this expanded utility comes the critical necessity to devise effective path planning methodologies to ensure safe and fast navigation of drones across dynamic landscapes [7]. UAV path planning benefits from diverse methodologies aimed at improving navigation efficiency in real-world scenarios. Among these methodologies, some path planning methods are based on traditional optimization techniques, such as nature-inspired methods that use mathematical models to simulate the collective behavior of natural systems. Examples of these nature-inspired methods include particle swarm optimization (PSO) and ant colony optimization (ACO) [8], which draw inspiration from the behaviors of birds and ants, respectively.

Additionally, grid-based methods like the A* algorithm and Dijkstra's algorithm provide reliable paths for robots, including UAVs. The A* (Algorithm 1) [9], a widely used

pathfinding method, employs a heuristic search to efficiently determine the shortest path between two points on a graph.

By intelligently balancing heuristic estimation with actual cost, A* navigates complex environments while optimizing time and resource usage [10]. Similarly, Dijkstra's algorithm, a classic graph search technique, finds the shortest path from one source vertex to all other vertices in a weighted graph. It achieves this by iteratively exploring neighboring vertices, selecting the vertex with the smallest known distance, and updating the distances of neighboring vertices accordingly [10].

Furthermore, drones use optimization methods such as the genetic algorithm (GA), inspired by natural selection processes, to iteratively optimize trajectories through selection, crossover, and mutation [11].

According to [12], environments can be categorized into two main types: static environments and dynamic environments. Static environments maintain constant conditions over time, including fixed obstacles, destinations, and weather patterns. On the other hand, dynamic environments are characterized by changing conditions, such as moving obstacles and changing weather conditions.

In this paper, we explore a complex 2D flight environment where a quadrotor drone seeks to intercept a moving landing station amid fixed obstacles while facing the effect of wind. Both the UAV and the landing station are exposed to the presence of fixed obstacles, and the UAV must navigate around them to reach its target. In addition, the presence and strength of wind further complicates the path planning process. To adapt to these dynamic conditions, the drone adjusts the weight assigned to the obstacles using the GWO algorithm.

2. Related Work

A lot of researchers have been actively exploring the impact of wind on how drones move. They have been carrying this out by suggesting novel approaches. In [13], the author presents a cost-effective solution for drones to follow predefined routes while maintaining stable camera focus on targets, eliminating the need for expensive wind sensors. This innovative approach not only enhances UAV navigation but also addresses wind effects more efficiently and affordably. In a recent study [14], the author proposes an efficient particle swarm optimization (PSO) metaheuristic algorithm to reduce energy consumption, which is known as a not computationally intensive algorithm[15], This approach recognizes that although the drone's flight path may span three dimensions, its constant altitude allows it to be simplified and treated as a two-dimensional problem. Additionally, researchers are continually enhancing existing, well-known methods like A* (Algorithm 1), reflecting the continuous quest for more reliable flight strategies.

The authors in [16] present an enhanced version of the A* algorithm, which incorporates both distance and wind data to optimize navigation. The authors employ a collision avoidance technique to avoid obstacles while their intelligent graph creation method considers the complexities of wind patterns. A comparative analysis favors probabilistic route map (PRM) graphs, highlighting their effectiveness in navigation. PRM is a robotics algorithm utilized to refine routes from the robot's starting point to its intended destination [17]. The study recommends the hybrid A* angle for conflicting wind scenarios and emphasizes the advantages of on-board processing in UAVs, specifically probabilistic road mapping. This research establishes foundations for wind-optimized global planning, enhancing quadcopter navigation amid real-time wind changes and limited communication resources. Additionally, the study proposes a collision verification method applied to outdoor terrain scenarios with different wind types. Furthermore, A* has a major limitation in generating paths in dynamic environments, such as those containing wind [18]. The wind effect was calculated [19] assuming that the aircraft flies at a constant flight level and at a constant true airspeed, and under this assumption, they use the following equation of motion (1):

$$\dot{x}(t) = V_a \cos(\theta(t)) + W_x(x, y)$$
$$\dot{y}(t) = V_a \sin(\theta(t)) + W_y(x, y) \quad (1)$$

where (x, y) is the aircraft position, θ is the heading angle, Va is the true airspeed, Wx(x, y) is the east component of the wind, and Wy(x, y) is the north component of the wind.

Algorithm 1: A* Algorithm

Procedure A^*(start, map, goal)
begin
close ← NULL — *Closed list, stores visited nodes*
open ← start — *Open list, stores unvisited nodes*
G[start] ← g[start] = 0, H[start] ← h[start] — *Cost and heuristic*
F[start] ← H[start] — *Evaluation function $F = G + H$*
while open ≠ ∅ **do**
current_Node ← Node with the minimum F value in open — *Select best node*
open ← open \ {current_Node} — *Remove from open list*
close ← close ∪ {current_Node} — *Add to closed list*
if current_Node = goal **then**
return best_path — *Path found!*
end if
neighbor_Nodes ← All neighbor Nodes of current_Node — *Explore neighbors*
for N ∈ neighbor_Nodes **do**
if N ∈ close **then** — *Skip closed nodes*
do nothing
else if N ∈ open **then**
$G[N_{\text{new\_calculated}}], H, F$ ← calculated N's G, H, F — *Update values*
if $G[N \in \text{open}] > G[N_{\text{new\_calculated}}]$ **then**
N's parent ← current_Node — *Update parent*
end if
else
N's parent ← current_Node — *Set new parent*
end if
end for
end while
end

- **open**: The open list, containing nodes that are being considered for exploration. Initially, this list holds only the starting node.
- **close**: The closed list, which stores nodes that have already been fully explored, meaning their neighbors have been evaluated.
- **G**: The cost function, representing the actual cost of moving from the start node to the current node.
- **H**: The heuristic function that estimates the cost from the current node to the goal, typically based on a chosen heuristic like Euclidean or Manhattan distance.
- **F**: The evaluation function, calculated as $F = G + H$, which is used to determine the node with the lowest estimated total cost to the goal.
- **parent**: For each node, the parent is the node from which the current node was reached. This allows the path to be reconstructed once the goal is reached.

The paper [20] presents a method integrating YOLO v4 and DeepSORT for automatic vehicle detection and tracking in urban environments. The proposed algorithm enhances UAV capabilities in traffic monitoring by combining deep learning techniques with an interactive multimodel particle filter for high-precision positioning of maneuvering targets, where You Only Look Once version 4 (YOLOv4) is an advanced object detection algorithm designed to provide high accuracy and speed in detecting objects within images [21].

The paper by [22] presents an enhanced A* algorithm tailored for vessel path planning, addressing limitations of traditional methods. By incorporating risk models for obstacles, including factors like currents, traffic separation, and berthing constraints, the algorithm achieves a balance between path length and navigation safety. Simulation and real-world experiments confirm its effectiveness in mitigating collision risks and adhering to navigation rules, offering a practical solution for optimizing vessel routes. In addition, the concept of incorporating weighted obstacles into the A* algorithm presented in this article offers valuable insights for UAV trajectory planning. By adapting a similar approach to assign weights to obstacles based on factors such as wind speed, terrain elevation or dynamic obstacles, the improved A* algorithm could be effectively applied to drone navigation in dynamic environments. This adaptation could improve the efficiency and safety of drone trajectory planning, making it a promising avenue for future research and development in this field.

In [9], the authors propose an enhanced A* algorithm specifically designed for path planning in medical testing laboratories. This improved version incorporates a bidirectional search strategy and an advanced heuristic that considers node angles. Additionally, the algorithm features path optimization techniques such as the removal of redundant nodes and path smoothing using cubic B-spline curves.

Furthermore, in [23], the authors introduce an improved A* algorithm tailored for probabilistic air pollution detection using UAVs. This variant of the A* algorithm leverages a probabilistic exploration and target search strategy. It enhances the heuristic function by incorporating pollution concentration data, allowing for more accurate and efficient navigation in environments with dynamic obstacles such as pollution sources.

A recent addition to this arsenal is GWO, introduced by S. Mergalili et al. in 2014 [24]. Inspired by the social hierarchy and hunting behavior of grey wolves, GWO provides an efficient way to search for optimal solutions by mimicking the cooperative strategies observed in wolf packs, Figure 1. By leveraging the principles of social behavior, GWO enhances UAV path planning capability and efficiency in dynamic environments.

In [25], Li et al. proposed an innovative approach to UAV path planning based on an improved GWO, specifically designed to address the challenges of dynamic and complex environments. The study enhances traditional GWO by introducing adaptive convergence and weighting factors, which dynamically adjust the search process to improve global exploration and local refinement. Unlike traditional methods such as A*, which are briefly mentioned for comparison, the improved GWO algorithm focuses on optimizing 3D paths while accounting for factors like fuel consumption and terrain threats. The simulation results demonstrate that the adaptive GWO offers superior path quality, stability, and convergence speed compared to conventional GWO and other algorithms. This research showcases the potential of AGWO for real-world applications in areas such as disaster relief and complex terrain navigation.

In [26], Zhang et al. propose an improved A* algorithm for global path planning for unmanned surface vehicles (USVs). This improved algorithm introduces a bidirectional search strategy and an advanced heuristic function that incorporates a vertical distance node deviation factor. Additionally, the algorithm employs an 8-neighborhood search method, improving the efficiency and accuracy of path planning.

In our previous work [27], we presented two methods based on differential game theory to address dynamic tasks, specifically the process of docking a UAV to a moving landing station amid wind effects in a cooperative scenario between the UAV and the landing station. Based on this foundation in differential game theory, Isaacs [18] illustrates

the complexities of chase games involving two players, each with different speeds. Isaacs' analysis highlights the challenge of predicting the future positions of the two players when sudden changes in direction are possible. Similarly, in our scenario, where the landing station and the UAV interact dynamically, we use a simplified approach known as target interception (TI). This method involves comparing the actual position of the drone with its assumed position based on a constant speed and direction, allowing us to detect and compensate for any external influences on the drone's trajectory.

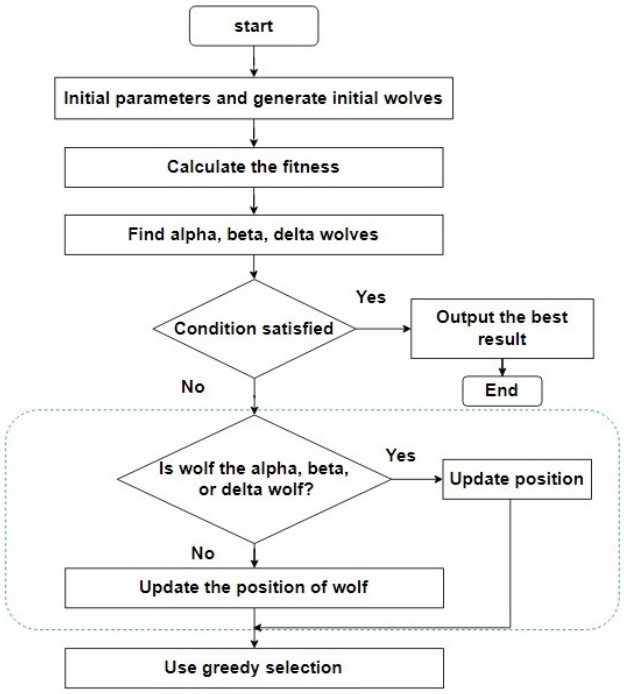

Figure 1. GWO calculation algorithm.

3. Dynamic UAV Trajectory Planning with A* and GWO (GW-A*)

In our proposed solution, grey wolf A* (GW-A*), we address the challenge of navigating UAVs in dynamic environments, where they must intercept a moving landing station while navigating through obstacles. To simplify the calculations, we model the environment as a 2D grid, assuming that the UAV flies at a constant altitude, similar to the approach used in [28], where the landing station initially plans its path using regular A* with a Euclidean distance heuristic to circumvent obstacles, Equations (2) and (3).

$$f(n) = g(n) + h(n) \tag{2}$$

$$h(n) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3}$$

where f(n) is the estimated path to reach the goal node through node n, g(n) is the cost to reach node n from the starting point, and h(n) is the heuristic cost to reach the goal from node n.

Subsequently, the UAV calculates its trajectory based on the path of the landing station, using an A* algorithm with weighted nodes and a Euclidean distance heuristic function to efficiently navigate around obstacles. This approach assigns higher weights to the nodes closest to the obstacles, Equation (4), ensuring safe maneuvering around potential obstacles. Each node's weight is determined by calculating the Manhattan distance between the node

and nearby obstacles, Equation (5), providing a measure of proximity to potential obstacles along the path.

$$\text{NodeWeight}(x, y) = \text{OW} \times \min_{\text{obs} \in \text{obstacles}} (|x - \text{obs.x}| + |y - \text{obs.y}|) \tag{4}$$

$$Manhattandistance = |x1 - x2| + |y1 - y2| \tag{5}$$

- NodeWeight(x, y): The weight assigned to the node at position (x, y).
- OW: ObstacleWeight: A constant representing the weight factor for obstacles equal to 1.
- obs.x and obs.y: The coordinates of each obstacle in the environment.
- $|x - \text{obs.x}|$ and $|y - \text{obs.y}|$: The Manhattan distance between the node and each obstacle.
- min function: Selects the minimum Manhattan distance among all obstacles.

Furthermore, the A* algorithm incorporates a heuristic Euclidean distance function to optimize the UAV's trajectory towards the landing station intercept. Considering the speed difference between the UAV (15 m/s) and the landing station (10 m/s), the UAV strategically selects the optimal interception point along the generated trajectory, increasing the efficiency and accuracy of the interception maneuver. Utilizing the GWO, the UAV fine-tunes the node weights to minimize path length, enhancing efficiency in dynamic environments. The GWO algorithm iteratively optimizes the weights to ensure the shortest possible path while adhering to safety constraints. GW-A* illustrates the iterative process of the GWO algorithm, and Figure 2 [29] illustrates the iterative process of the GWO algorithm, demonstrating its mechanism for exploring the search space, updating the positions of wolves, and converging towards optimal solutions.

Additionally, during flight, the UAV dynamically adjusts its trajectory based on wind conditions, detecting wind velocity based on deviations in its expected coordinates (x, y). When the UAV's actual position differs from its planned position, it calculates the discrepancy and uses this deviation to estimate wind speed and direction. Wind detection prompts the UAV to modify its node weights dynamically, prioritizing safer paths in the presence of adverse wind conditions. By adjusting node weights in real-time, the UAV can adapt its trajectory to mitigate the impact of wind and ensure safe navigation throughout its mission. The displacement caused by wind is calculated using method (1), which takes into account both wind speed and direction, adjusting the UAV's position accordingly. To evaluate performance, we compare this approach with traditional A* using four different heuristic functions in identical scenarios. Through this simulation, we assess the efficacy of our proposed approach in dynamic environments with varying wind conditions.

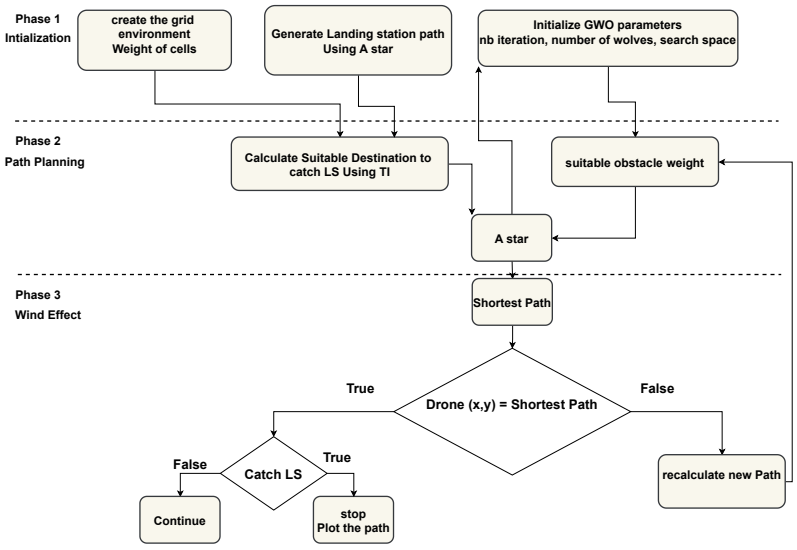

Figure 2. The flowchart of the proposed approach (GW-A*).

4. Implementation and Results

In our study, we systematically evaluated the performance of our novel approach within a complex 2D flight environment where a quadrotor UAV seeks to intercept a moving landing station while navigating through fixed obstacles and facing the effects of wind. The quadrotor UAV, known for its high maneuverability and ability to turn without a limiting radius, is well-suited for this environment. Our approach seamlessly integrates A* and GWO methodologies to effectively address scenarios with weighted obstacles. The UAV must strategically maneuver around these obstacles to reach its target, which adds an additional layer of complexity to the path planning process. This evaluation involved a comparative analysis against the traditional A* algorithm, employing three distinct heuristic functions: Euclidean distance (Equation (3)), Manhattan distance (Equation (5)), and diagonal distance (Equation (6)).

$$D_{\text{diagonal distance}}(p_1, p_2) = \max(|x_2 - x_1|, |y_2 - y_1|) \tag{6}$$

where D is the diagonal distance between two points p_1 and p_2 on a grid, (x_1, y_1) are the coordinates of point p_1, while (x_2, y_2) are the coordinates of point p_2. A total of 250 tests were conducted, divided into four types of implementations. The first implementation comprised 130 simulations with random wind speeds and wind impact angles. The second implementation involved thirty tests with random wind impact angles and wind speeds within a specific range, while the third implementation also included 30 tests but with a different wind speed range. Finally, the fourth implementation consisted of 60 tests with a fixed wind impact angle, where wind speeds incrementally increased over a series of tests. All random numbers for wind speed and angle were generated using Python's random.randint(a, b) and random.uniform(a, b) methods, both based on a pseudo-random number generator (PRNG). The random.randint() method generates random integers to simulate discrete wind speeds, while random.uniform() generates floating-point numbers for a more realistic representation of wind angle variations.

During all the implementations, we use a 2D grid of 8000 square meters, divided into eighty cells, each with a size of 10 × 10 m. The grid configuration, illustrated in Figure 3, shows that the value '0' represents an empty cell, while the value '1' denotes an obstacle. To visualize the grid, we employed a Python script utilizing the matplotlib library. The script iterates over each cell in the grid, plotting a black square marker for each cell with a value of '1'. The generated map is shown in Figure 4. The obstacles are strategically placed to reflect various real-world scenarios, such as narrow corridors, open spaces with isolated structures, and dense groups of buildings. This diverse layout ensures a comprehensive analysis of different navigation challenges and spatial dynamics.

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 1, 1, 1, 1, 1, 1, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
[0, 1, 1, 0, 1, 1, 1, 0, 1, 0],
[0, 1, 1, 0, 1, 1, 1, 0, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
[0, 1, 0, 0, 0, 0, 1, 1, 1, 0],
[1, 0, 0, 1, 1, 0, 0, 0, 0, 1]
```

Figure 3. Grid layout code snippet.

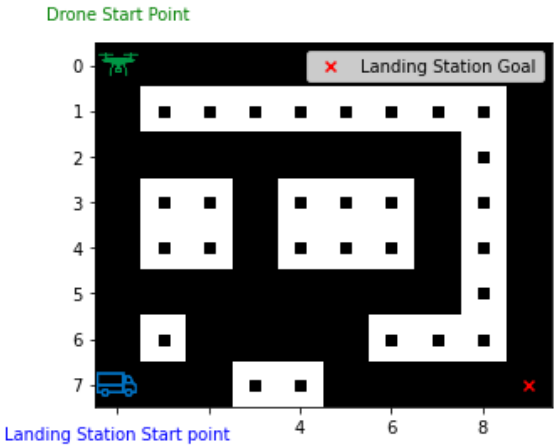

Figure 4. Visual representation of the grid layout.

4.1. Implementation with Random Wind Conditions (speed and impact angle)

In the first phase of our experimentation, we conducted 130 runs to evaluate the performance of our novel approach against the traditional A* algorithm with Euclidean distance heuristic. The random variables, including wind direction ranging from -180° to 180° and wind speed varying between 0 and 6 m/s.

The implementation results presented in Table 1, indicate that during the 130 runs, our proposed approach GW-A* outperformed the traditional A* algorithm 105 times, which is approximately 80% of the time. In contrast, the A* algorithm performed better in twenty-five instances, which accounts for about 20% of the time. GW-A* had a lower median, indicating better performance metrics compared to A*. However, the higher standard deviation suggests less predictability than A*, which has a lower standard deviation. This may be a disadvantage in situations where consistent performance is crucial. The average difference in performance between the two approaches was approximately 1.8 m, with a total difference of 224.7 m, indicating the efficiency of the proposed GW-A* approach.

Table 1. Results of implementing random wind conditions: variations in speed and impact angle.

Performance Indicators	Approach	
	A*	**GW-A***
Superiority Frequency	25	105
MIN (m)	112.6	108.4
MAX (m)	121.8	123.8
Average (m)	112.60	110.87
Median (m)	112.01	109.05
Standard Deviation (m)	2.80	3.87
Sum of Traveled Distance (m)	14,638.5	14,413.7
Net Differences (m)	224.7	

4.2. Implementation with Random Wind Conditions (Impact Angle, Speed Less then 3 m/s)

In the second phase of our experiment, we carried out 30 trials to further evaluate the performance of our approach under different wind conditions. For this phase, we generated random wind speeds between 0 and 3 m/s and random wind impact angles between -180° and 180°.

The results of this phase, as shown in Table 2, in the row labeled 'Random (wind direction) & Random (wind speed) <3 m/s', indicate that the proposed approach outperforms the traditional A* algorithm in 24 out of 30 trials, which is 80% of the time. In contrast, the A* algorithm performs better in six trials, accounting for 20% of the time. These results

are approximately the same as those mentioned in Section 4.2. Similar to previous findings, GW-A* consistently shows a smaller median but a higher standard deviation. With a median of 109.0 m compared to A*'s median of 110.64 m, GW-A* generally performs better under these conditions. However, the higher standard deviation for GW-A* (4.22) compared to A* (1.79) indicates that while GW-A* often outperforms A*, its results are more variable and less predictable, especially in scenarios with low wind speeds. When we analyzed the mean and net differences between the two methods, we found that the results were fairly close, with a total difference of just 1.5 m over the 30 trials and a mean difference of 0.6 m. This small difference suggests that while GW-A* shows overall better performance, both approaches yield similar results in terms of total traveled distance in less severe wind conditions.

4.3. Implementation with Random Wind Conditions (Impact Angle, Speed Greater than 3 m/s)

To further test our approach, the third phase of our experiment involved 30 trials under more intense wind conditions. We generated random wind speeds between 3 and 6 m/s and random wind impact angles between -180° and 180°.

The results, as shown in Table 2, indicate that as the wind strength increases, the performance of the regular GW-A* algorithm remains similar to the previous stage in terms of the percentage of winning, the median, and the standard deviation. However, we can see that the standard deviation of A* is now higher than in the previous stage and is closer to that of GW-A*. However, the overall difference between our proposed approach and the regular A* algorithm increased to 69.74 m, reflecting the greater impact of intense wind conditions. When divided by the number of runs, this represents a higher average difference compared to the previous two implementations. In addition, the average difference between the two methods increased to 2.32 m, indicating a greater disparity in performance under more challenging wind conditions.

Table 2. Results comparing random wind conditions for speeds above and below 3 m/s.

Wind Speed	Approach	Superiority Frequency	Median (m)	Standard Deviation (m)	Sum of Traveled Distance (m)	Net Differences (m)
less than 3 m/s	GW-A*	24	109.0	4.22	3331.8	1.53
	A*	6	110.64	1.79	3330.3	
greater than 3 m/s	GW-A*	25	109.9	4.35	3352.6	69.7
	A*	5	114.08	3.93	3422.43	

4.4. Implementation with Incremental Wind Speed from 0 to 6 m/s

During the last implementation, we conducted sixty tests with a fixed wind impact angle of 45 °, incrementally increasing the wind speed from 0.1 m/s to 6 m/s in steps of 0.1 m/s for each test. This allowed us to assess how changes in speed affect the results. The aim was to provide a comprehensive analysis of the performance of each approach under different environmental conditions, characterized by different wind speeds. The simulation results, presented in Figure 5, show the number of wins for each approach within three different subsections of wind speeds: low speeds from 0.1 to 1.8 m/s, medium speeds from 1.9 to 3.4 m/s, and high speeds from 3.5 to 6 m/s. By dividing the wind speeds into these subsections, we can better analyze and compare the robustness and effectiveness of each approach in different wind conditions. This subdivision allows us to understand which approaches are more adaptable and perform better under certain environmental dynamics. The chart in Figure 5 indicates that GW-A* consistently outperforms the A* method across all wind speed ranges tested. The simulation results show that GW-A* consistently outperformed A* at both high and low wind speeds. However, A* demonstrated relatively high performance at moderate wind speeds, specifically between 1.9 and

3.4 m/s. This suggests that while GW-A* is generally more robust in a wider range of wind conditions, A* may be competitive in some moderate wind scenarios.

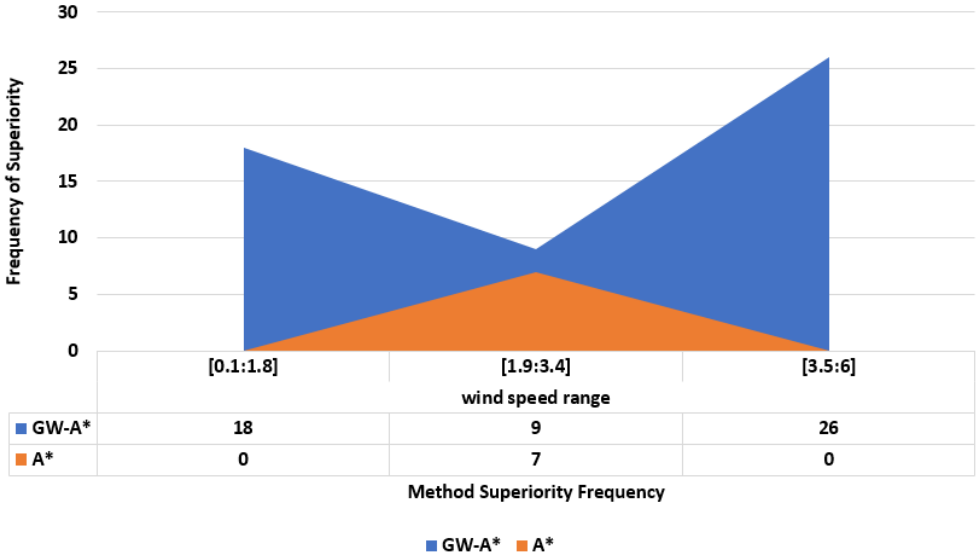

Figure 5. Performance comparison between GW-A* and A* across incremental wind speed ranges from 0 to 6 m/s, with an impact angle of 45 °.

5. Conclusions

In this paper, we introduced a novel approach for UAV path planning in dynamic environments influenced by wind. Our method enhances the traditional A* algorithm by incorporating weighted nodes, where the weights are determined based on the distance from obstacles and further optimized using GWO. This combination allows for the generation of paths that are not only the shortest but also the safest, considering both the dynamic nature of the environment and the proximity to obstacles.

The experimental results demonstrated notable improvements in path safety and efficiency in both low and high wind scenarios, where the A*-GWO outperformed traditional A* by generating safer, more adaptive paths. A-GWO consistently produced more robust paths in complex environments with dynamic factors like wind, even so, it did not universally outperform A* in all scenarios. This suggests that A*-GWO offers significant advantages in specific conditions but may not always be the best choice for environments with moderate dynamics. However, it is important to acknowledge that in moderate wind conditions, the performance of the traditional A algorithm remained competitive, occasionally matching or even slightly exceeding the A-GWO in terms of simplicity and computational speed.

6. Future Work

Based on our simulation results of previous achievements, our next objectives are to apply a machine learning model in order to select the most suitable path planning method based on specific environmental features. We aim to develop a robust machine learning model capable of dynamically adapting to changing conditions by analyzing and predicting the optimal path planning strategy. This approach will involve creating a comprehensive dataset from our current approaches and training the model to recognize patterns and make informed decisions. By incorporating machine learning, we anticipate further improvements in UAV trajectory planning, enhancing both efficiency and safety in complex environments. Additionally, we plan to conduct real-world experiments to validate and improve our simulation results using UAVs with key features specifically designed for effective testing and monitoring. We will use programmable drones with open-source flight controllers (e.g., Pixhawk) for customization and path-planning integration. Equipped with high-precision GPS, IMU sensors, and telemetry systems, these UAVs

will allow real-time monitoring of position, speed, and external influences. To ensure the reliability of the data, we will add sensors to measure wind and environmental conditions and use ground control software to make real-time mission adjustments and record detailed data. This setup will create a robust test environment, allowing us to accurately assess the drones' performance in real-world conditions, as shown in Table 3.

Table 3. Comparison of A* Algorithm Variants.

Method	Objective	Application	Heuristic Function	Path Optimization	Handling Obstacles	Algorithm Improvements	Limitations
Basic A* Algorithm	Find shortest path	General robotics	Euclidean distance	None	Static obstacles	Basic	Limited to static environments
[23]	Probabilistic air pollution detection	UAVs	Enhanced heuristic considering pollution concentration	Optimization based on pollutant data	Dynamic obstacles (pollution sources)	Enhanced with environmental data processing	Dependent on accurate environmental data
[9]	Path planning in medical testing laboratories	Mobile robots in medical labs	Improved heuristic considering node angles and bidirectional search	Removal of redundant nodes and path smoothing using cubic B-spline curves	Static environments; does not consider dynamic obstacles	Enhanced with bidirectional search, node removal, and path smoothing	Primarily designed for static environments; needs real-world validation
[26]	Efficient path planning for surface vehicles	Surface vehicles in dynamic environments	Manhattan, Euclidean, and Chebyshev distances; dynamic window	Path smoothing using spline curves	Smoothing to match actual navigation trajectories	Reduces computational time and node traversing	Bidirectional search may fail to meet under certain conditions
[25]	Optimal path planning considering threat and fuel costs	UAVs in complex and dynamic 3D environments	Adaptive mechanism with three weighting factors	Dynamic adjustment based on alpha, beta, and delta wolf positions	Effective handling of various path planning scenarios	Improved convergence and avoidance of local optima	Less effective on unimodal benchmark functions due to adaptive weighting
Proposed Approach GW-A*	Dynamic path planning considering wind and varying speeds	UAVs in dynamic environments with moving landing stations	Euclidean distance with weighted nodes for obstacles	Minimizes path length and adjusts trajectory dynamically	Navigates fixed obstacles; dynamically adjusts for wind	Integrates GWO for optimizing node weights and paths	Requires real-time wind data and dynamic adjustments

Author Contributions: Conceptualization, A.H.A; Methodology, O.Z. and B.C.; Writing— review and editing, A.H.A., O.Z., A.N. and B.C.; Supervision, O.Z., A.B. and B.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shavarani, S.M.; Nejad, M.G.; Rismanchian, F.; Izbirak, G. Application of hierarchical facility location problem for optimization of a drone delivery system: A case study of Amazon prime air in the city of San Francisco. *Int. J. Adv. Manuf. Technol.* **2018**, *95*, 3141–3153. https://doi.org/10.1007/s00170-017-1363-1.
2. Green, R.; Johnson, L. Applications of Drones in Precision Agriculture. *J. Agric. Sci.* **2023**, *28*, 234–245. https://doi.org/10.4567/jas.2023.234567.

3. Williams, D. Using UAVs for Wildlife Monitoring and Anti-Poaching. *Conserv. Sci. Technol.* **2023**, *22*, 89–102. https://doi.org/10.6543/cst.2023.789012.
4. Lee, K.; Kim, M.; Park, H. Drone Technology in Disaster Response: Real-Time Assessment and Management. *J. Emerg. Manag.* **2023**, *17*, 147–159. https://doi.org/10.3210/jem.2023.654321.
5. Zimroz, P.; Trybała, P.; Wróblewski, A.; Góralczyk, M.; Szrek, J.; Wójcik, A.; Zimroz, R. Application of UAV in Search and Rescue Actions in Underground Mine—A Specific Sound Detection in Noisy Acoustic Signal. *Energies* **2021**, *14*, 3725. https://doi.org/10.3390/en14133725.
6. Chen, P. Drone Applications in Construction: Site Surveying and Progress Tracking. *J. Constr. Eng.* **2023**, *31*, 305–318. https://doi.org/10.7890/jce.2023.345678.
7. Zhang, F.; Liu, H. Path Planning Methodologies for UAV Navigation. *J. Robot. Autom.* **2023**, *29*, 12–25. https://doi.org/10.5432/jra.2023.123456.
8. Foo, J.L.; Knutzon, J.; Oliver, J.; Winer, E. Three-Dimensional Path Planning of Unmanned Aerial Vehicles Using Particle Swarm Optimization. In Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA, USA, 6–8 September 2006; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2006. https://doi.org/10.2514/6.2006-6995.
9. Ju, C.; Luo, Q.; Yan, X. Path Planning Using an Improved A-star Algorithm. In Proceedings of the 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan), Jinan, China, 23–25 October 2020; pp. 23–26. https://doi.org/10.1109/PHM-Jinan48558.2020.00012.
10. Bolourian, N.; Hammad, A. LiDAR-equipped UAV path planning considering potential loca- tions of defects for bridge inspection. *Autom. Constr.* **2020**, *117*, 103250. https://doi.org/10.1016/j.autcon.2020.103250.
11. Shivgan, R.; Dong, Z. Energy-Efficient Drone Coverage Path Planning using Genetic Algorithm. In Proceedings of the 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR), Newark, NJ, USA, 11–14 May 2020; pp. 1–6. https://doi.org/10.1109/HPSR48589.2020.9098989.
12. Yang, Y.; Xiong, X.; Yan, Y. UAV Formation Trajectory Planning Algorithms: A Review 2023. *Drones* **2023**, *7*, 62. https://doi.org/10.3390/drones7010062.
13. Jayaweera, H.M.P.C.; Hanoun, S. Path Planning of Unmanned Aerial Vehicles (UAVs) in Windy Environments. *Drones* **2022**, *6*, 101. https://doi.org/10.3390/drones6050101.
14. Chan, Y.; Ng, K.K.; Lee, C.; Hsu, L.T.; Keung, K. Wind dynamic and energy-efficiency path planning for unmanned aerial vehicles in the lower-level airspace and urban air mobility context. *Sustain. Energy Technol. Assess.* **2023**, *57*, 103202. https://doi.org/https://doi.org/10.1016/j.seta.2023.103202.
15. Ercan, M.; Li, X. Particle Swarm Optimization and Its Hybrids. *Int. J. Comput. Commun. Eng.* **2013**, *2*, 52–55. https://doi.org/10.7763/IJCCE.2013.V2.135.
16. Thanellas, G.A.; Moulianitis, V.C.; Aspragathos, N.A. A spatially wind aware quadcopter (UAV) path planning approach. *IFAC-PapersOnLine* **2019**, *52*, 283–288. https://doi.org/10.1016/j.ifacol.2019.08.084.
17. Alarabi, S.; Luo, C.; Santora, M. A PRM Approach to Path Planning with Obstacle Avoidance of an Autonomous Robot. In Proceedings of the 2022 8th International Conference on Automation, Robotics and Applications (ICARA), Prague, Czech Republic, 18–20 February 2022; pp. 76–80. https://doi.org/10.1109/ICARA55094.2022.9738559.
18. Choset, H.; Lynch, K.; Hutchinson, S.; Kantor, G.; Burgard, W. *Principles of Robot Motion: Theory, Algorithms, and Implementations*; Intelligent Robotics and Autonomous Agents Series; MIT Press: Cambridge, MA, USA, 2005.
19. Girardet, B.; Lapasset, L.; Delahaye, D.; Rabut, C.; Brenier, Y. Generating optimal aircraft trajectories with respect to weather conditions. In Proceedings of the ISIATM 2013, 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management, Toulouse, France, 8–10 July 2013.
20. Liu, X.; Zhang, Z. A Vision-Based Target Detection, Tracking, and Positioning Algorithm for Unmanned Aerial Vehicle. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1–12. https://doi.org/10.1155/2021/5565589.
21. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934.
22. Liu, C.; Mao, Q.; Chu, X.; Xie, S. An Improved A-Star Algorithm ConsideringWater Current, Traffic Separation and Berthing for Vessel Path Planning. *Appl. Sci.* **2019**, *9*, 1057. https://doi.org/10.3390/app9061057.
23. Ha, I.K. Improved A-Star Search Algorithm for Probabilistic Air Pollution Detection Using UAVs. *Sensors* **2024**, *24*, 1141. https://doi.org/10.3390/s24041141.
24. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.
25. Zhang, W.; Zhang, S.; Wu, F.; Wang, Y. Path Planning of UAV Based on Improved Adaptive Grey Wolf Optimization Algorithm. *IEEE Access* **2021**, *9*, 89400–89411. https://doi.org/10.1109/ACCESS.2021.3090776.
26. Zhang, H.; Tao, Y.; Zhu, W. Global Path Planning of Unmanned Surface Vehicle Based on Improved A-Star Algorithm. *Sensors* **2023**, *23*, 6647. https://doi.org/10.3390/s23146647.
27. Ahmad, A.H.; Zahwe, O.; Nasser, A.; Clement, B. Path Planning Algorithms For Unmanned Aerial Vehicle: Classification, Performance, and Implementation. In Proceedings of the 2023 3rd International Conference on Electrical, Computer, Communications

and Mechatronics Engineering (ICECCME), Tenerife, Canary, 19–21 July 2023; pp. 1–6. https://doi.org/10.1109/ICECCME57830.2023.10252168.
28. Song, X.; Hu, S. 2D Path Planning with Dubins-Path-Based A* Algorithm for a Fixed-Wing UAV. In Proceedings of the 2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE), Beijing, China, 17–19 August 2017; pp. 69–73. https://doi.org/10.1109/CCSSE.2017.8087897.
29. Yu, X.; Jiang, N.; Wang, X.; Li, M. A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning. *Expert Syst. Appl.* **2023**, *215*, 119327. https://doi.org/10.1016/j.eswa.2022.119327.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.