robotics

MDPI

Article

Three-Dimensional Path Planning Optimization for Length Reduction of Optimal Path Applied to Robotic Systems

Ilias Chouridis [1], Gabriel Mansour [2,*] and Apostolos Tsagaris [1]

[1] Department of Industrial Engineering and Management, International Hellenic University, 57001 Thessaloniki, Greece; iliachour@iem.ihu.gr (I.C.); tsagaris@ihu.gr (A.T.)
[2] Department of Mechanical Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
* Correspondence: mansour@auth.gr

Abstract: Path planning is an intertemporal problem in the robotics industry. Over the years, several algorithms have been proposed to solve it, but weaknesses are constantly identified by researchers, especially in creating an optimal path in a three-dimensional (3D) environment with obstacles. In this paper, a method to reduce the lengths of optimal 3D paths and correct errors in path planning algorithms is proposed. Optimization is achieved by combining the information of a generated two-dimensional (2D) path with the input 3D path. The 2D path is created by a proposed improved artificial fish swarm algorithm (AFSA) that contains several improvements, such as replacing the random behavior of the fish with a proposed one incorporating the model of the 24 possible movement points and utilizing an introduced model to assist the agent's navigation called obstacles heatmap. Moreover, a simplified ray casting algorithm is integrated with the improved AFSA to further reduce the length of the final path. The improved algorithm effectually managed to find the optimal path in complex environments and significantly reduce the length of the formed path compared with other state-of-the-art methods. The path was implemented in real-world scenarios of drone and industrial robotic arm applications.

Keywords: path planning; drone; robotic arm; navigation; artificial fish swarm algorithm; artificial intelligence

check for updates

Citation: Chouridis, I.; Mansour, G.; Tsagaris, A. Three-Dimensional Path Planning Optimization for Length Reduction of Optimal Path Applied to Robotic Systems. *Robotics* **2024**, *13*, 178. https://doi.org/10.3390/robotics13120178

Academic Editor: Charalampos P. Bechlioulis

Received: 6 November 2024
Revised: 11 December 2024
Accepted: 12 December 2024
Published: 14 December 2024

Copyright: © 2024 by the authors. 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

Path planning is considered a critical research area in robotics as it can expand the scope of mobile and industrial robots [1]. The development of industrial automation technologies [2] and flying robots [3], and the service [4] and medical industries [5] have significantly increased the employment of robots in production processes and increased their impacts on the daily lives of humans. The path planning problem determines a collision-free route that links a starting point to a target point. There are several solutions that can satisfy the described condition; however, the optimal solution for this problem can save time and energy worldwide, and increase productivity and the safety of robot use among others [6]. The path planning problem can be formulated as an optimization problem with several constraints and performance criteria, such as the shortest distance, the path's feasibility, and the compliance with the robot's kinematic confinements [7].

Numerous techniques have been developed over the past few years to deal with the path planning problem. Some of the available path planning algorithms are ant colony optimization [8], genetic algorithm [9], A* algorithm [10], particle swarm algorithm [11], artificial potential field algorithm [12], neural network algorithm [13], rapid exploration random tree (RRT) [14], memetic algorithm [15], and evolutionary algorithm [16]. Although the ant colony optimization algorithm is proven to work well for path planning tasks, it is relatively sensitive to the initial parameters, and the different initialization of the parameters can modify the algorithm's behavior and searchability [17]. The genetic algorithm depends

on experience for selecting relevant parameters and is susceptible to being "premature" [18]. The A* algorithm has limitations in locating the optimal solution for complex problems and the generated results are often monotonous and invalid [19]. The particle swarm algorithm requires high time complexity and it often falls into premature convergence. Moreover, identifying the optimal solution needs a large number of samples to estimate the posterior probability of the system, and the resampling stage often leads to a deficiency of the collected samples; as a result, the algorithm may not be able to find the optimal path solution in the desired time [19,20]. The artificial potential field method tends to fall into local optimal solutions and deadlocks [21]. The neural network requires a large amount of data for proper training and it is computationally expensive [22]. The memetic algorithms are exploring more possible routes in the environment but are prone to local optimal solutions. Evolution algorithms have slow evolution speed and premature convergence; as a result, they face difficulties with finding the global optimal solution [23].

Researchers have successfully applied the prementioned algorithms for solving the problem of finding the optimal path in both 2D [24–26] and 3D [27–29] environments with obstacles. All of the above methods have various combinations of advantages and disadvantages that make them more or less effective depending on the requirements and criteria of their application. A major open research field is to improve the behavior of the algorithms in order to mitigate the disadvantages and highlight the advantages. For example, Liu et al. proposed a 3D path planning algorithm for a UAV by combining particle swarm optimization with an adaptive sensitivity decision operator to solve the problems of local optima and slow convergence [30]. Fang et al. nominated an improved ant colony optimization algorithm integrated with an improved Q learning algorithm to overcome the low search efficiency and the fall of the local optimum in a 2D environment with obstacles [31]. Xu et al. suggested an improved A* algorithm that replaced the octet neighborhood with a rectangular boundary without obstacles in order to reduce the search time and the quantity of redundant nodes in the path planning process of the traditional A* algorithm [32].

The existing algorithms can successfully form paths in several 3D environments, but as the complexity and accuracy of representation of the physical world increases, due to their disadvantages, suboptimal paths are often encountered.

In the field of robotics, it is particularly important that algorithms are executed in an environment that simulates real-world conditions as closely as possible. Although the accurate representation of the real world in a digital model is an important open research challenge, several path planning algorithms have been applied to environments formed by different methodologies [33–35].

A promising environment modeling method in robotics is the grid method. In the grid method, the real world space is represented virtually by a set of discrete points. The grid method is widely used in path planning algorithms due to its ability to be easily updated with new environmental information and the similar operating principles of the sensors that are commonly used in robotics [36,37]. However, limited research has been conducted about the common adopted agent's eight possible movement points model and their influence on the length or the formation of the formed paths. Therefore, the researchers have not yet tried to apply the existing algorithms to models with more possible movement points.

Industrial and aerial robots are facing some common challenges. Navigation, security, and safety are some of them. The leading issue in flying and industrial robots is to find comparative and non-comparative positions. The challenge is the development of an automatic navigation system that will not depend on external human interventions. The robot's safety is a well analyzed topic, as the robots should be able to operate in adverse conditions while avoiding potential obstacles in the field [38,39].

In real-world applications, mobile or industrial robots are regularly needed to navigate in complex environments with dense obstacles. Assuming that a drone or an industrial robotic arm needs to move from point A to point B, it is necessary to design a 3D trajectory

that avoids stationary obstacles such as buildings or industrial equipment. In addition, due to the limited flight time of drones and the high value of production time of an industrial product, it is vital that the planned trajectory is as short as possible. Thus, the path implemented with these systems should definitely meet the abovementioned criteria to a great extent.

This paper proposes a method that can reduce the length of the optimal path obtained by a 3D path planning algorithm and correct path error cases. The 3D path information is combined with the data from solving the problem in a 2D environment. The 2D path is generated by a proposed improved modification of the artificial fish swarm algorithm (AFSA) that is suitable for solving the problem of finding the optimal path in a grid environment. The AFSA was improved by proposing the replacement of the random behavior of the fish with a new one. Instead of using the common eight possible movement points model, a model with twenty-four possible movement points is used to enhance the visual field of the fish and the ability to dynamically change the step parameter during the algorithm's execution. Moreover, the creation of a method called the obstacle heatmap that assists the agent to understand the effect of obstacles on its movement in a grid environment is proposed. The obstacle heatmap also captures the density of obstacles in the map so that the environment and the possible optimal routes can be explored more widely. The integration of the improved AFSA with a ray casting algorithm is additionally introduced. Furthermore, an analysis was performed on the effect that the number of possible movement points an agent can navigate to from a given location can have on the length and shape of the final path in the grid environment. Another thing that is included in this work is the introduction of the shrink coefficient μ for a more precise definition of obstacles in the grid environment. The effects of the orientation of the grid environment on the course and the length of the formed path are also examined and tested in two different formed 2D environments. Finally, the proposed algorithms and methods were examined through multiple experiments in different environments. They were used successfully in accomplishing real-world tasks such as pick and place applications with a six-degree-of-freedom industrial robotic arm and quadcopter drone navigation in an urban environment.

2. Materials and Methods

Industrial robots and drones have some commonalities in terms of path planning. Initially they can move with flexibility and speed in all 3 dimensions of a predetermined space. In their predetermined space there is a reference point O (0, 0, 0) from which the coordinates of their trajectory points are measured. The predetermined space limits their navigation capabilities to areas located within it, and the robotic arm is restricted by the length of its arms and the drone by its battery power. To find a path that represents the real-world conditions, a 3D grid approach was adopted.

In order to solve the path planning problem, the ant colony optimization algorithm described in [40] was used. The ant colony optimization algorithm was highly compatible and efficient for finding a large number of optimal paths. Nevertheless, it was observed that in the case of creating a dense grid with a large number of points, the ant method has the problems of a slow convergence speed, increased computational time, low search efficiency, and sometimes the optimal path could not be found but a large part of the output path converged to the optimum.

In the 3D grid approach, the robot can move over some discrete points located in the center of the rectangular parallelepipeds and representing a set of spatial coordinates. The denser the discrete points are, the smaller the set of spatial coordinates that each discrete point represents and the more precision can be achieved in the robot's movements between obstacles. However, the high density of discrete points increases their number and hence the complexity of the environment, making it difficult to find the optimal path.

The proposed method combines the 3D path from a thin or a dense grid with the 2D path of a denser grid. This approach allows us to overcome the above challenges and

to find a high-precision optimal path in a computationally efficient way. The 2D path is calculated based on a starting and an ending point on a plane of the 3D environment, as shown in Figure 1.

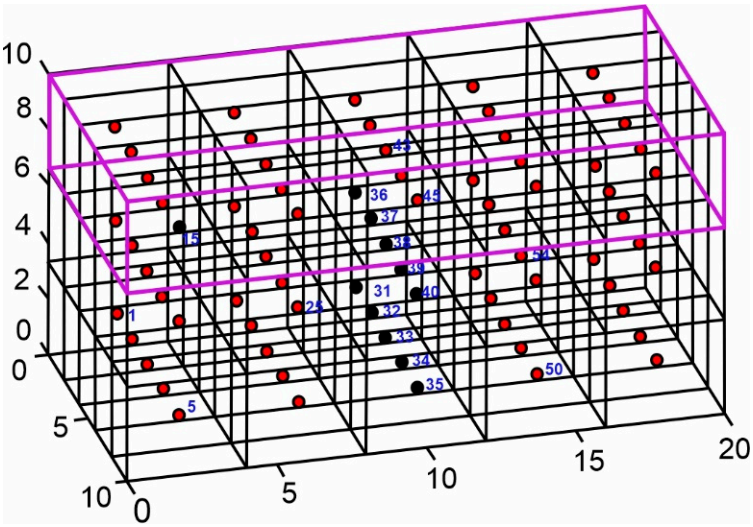
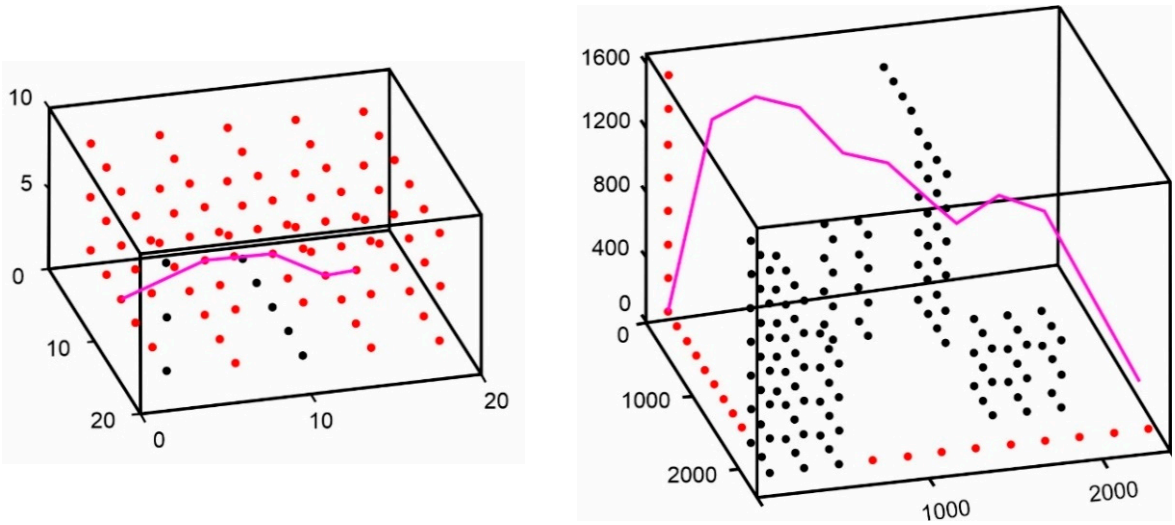


Figure 1. Section of the reference plane from the 3D environment.

Figure 2a shows the 3D path resulting from the application of the method [40] to a thin grid and Figure 2b shows the results in a dense grid environment. The discrete points where the robot can move are represented in red, while the obstacles are represented in black. The formed path is shown in magenta.



(**a**) (**b**)

Figure 2. Three-dimensional path planning gid environments: (**a**) 3D path calculation in a thin environment and (**b**) 3D path calculation in a dense environment.

2.1. Two-Dimensional Path Calculation

A disadvantage of the ant colony optimization method is the sensitivity of the algorithm's converge to the initialization of the parameters. For the case where a single path needs to be formed in a 3D or 2D environment, multiple tests are conducted with different initial parameters in order to find the optimal path. This process is time consuming and computationally ineffective when multiple paths need to be formed in a dense grid. Some of the criteria taken into consideration in order to ensure the efficiency of the 2D path

formation method are the speed coverage of the algorithm, the identification of the optimal solution, and the ability of the method to be used in a dense environment. Moreover, it is essential that the 2D path planning algorithm is not highly dependent on the tuning of the initial parameters. In addition to ant colony optimization, the algorithms described in the Introduction were considered to be used, but due to their disadvantages, an improved AFSA was chosen for this kind of optimization. The AFSA is a parallel stochastic algorithm characterized by good robustness, strong global searchability, large tolerance of parameter settings, and an insensitivity to the initial values, which are ideal for the studied case [41].

2.2. Grid Environment

The proper environment model is essential both for the valid algorithm's operation and for efficient robot movement. The precise execution of the robot's tasks depends significantly on the accuracy of the environment's information. The process of finding the optimal path can benefit the robot's navigation only when it is executed in proper environment model, depending on the type of robot and the task being performed. In addition, all the information about the existence of obstacles, the obstacle-free areas, and the directions that the robot can move are displayed in the environment. In this paper, a 2D grid environment was used in order to be consistent with the results of the 3D path formed by [40] and to be able to compare the effectiveness, reliability, and contribution of the proposed method. The 2D grid method divides a spatial set of coordinates into square sub-regions, which are represented by discrete points. The discrete points are located in the center of every sub-region. The only coordinates of the environment where the robot can move are those of discrete points. To determine the set of coordinates and the size of the grid environment, 2 methods are used: the simple and the advanced grid environments.

The simple grid environment method creates the environment by projecting the coordinates of the start and end points of the 3D path onto the reference plane. Namely, it creates a rectangle that encloses the generated path. The rectangle's boundaries are defined by the squared subregion's coordinates representing the discrete points of the 3D grid. In order to ensure consistency between the 3D and 2D grids, each subregion of the 3D grid is divided into n_g X n_g 2D subregions. The parameter n_g is an odd number so that the coordinates of the discrete point from the 3D grid are transferred without change to some discrete point of the 2D grid. Figure 3a shows an example of the division of a 3D grid sub regions to an odd number of n_g 2D subregions. If n_g is an even number, as shown in Figure 3b, the coordinates of the 3D discrete points do not correspond to any 2D discrete point.

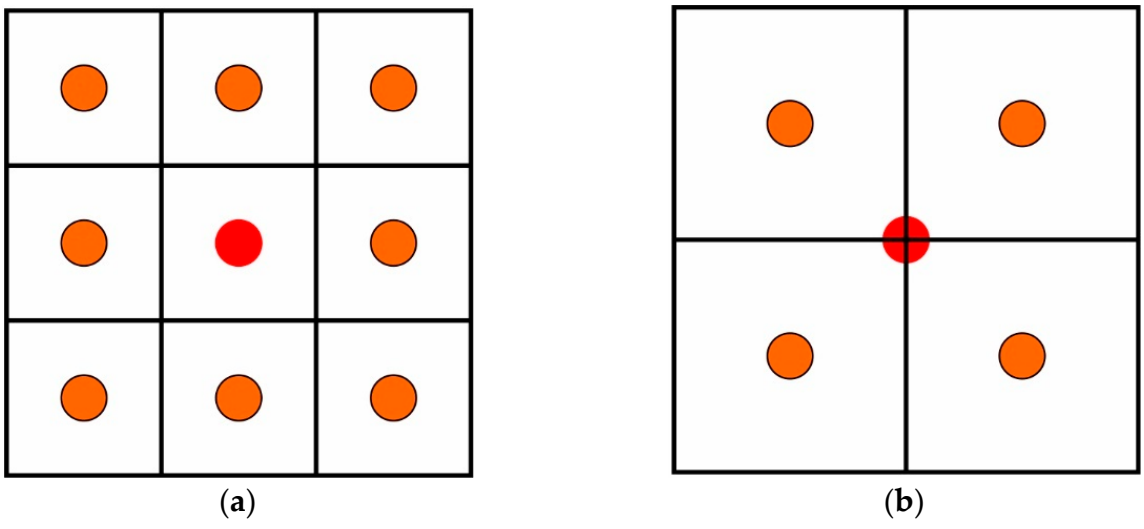


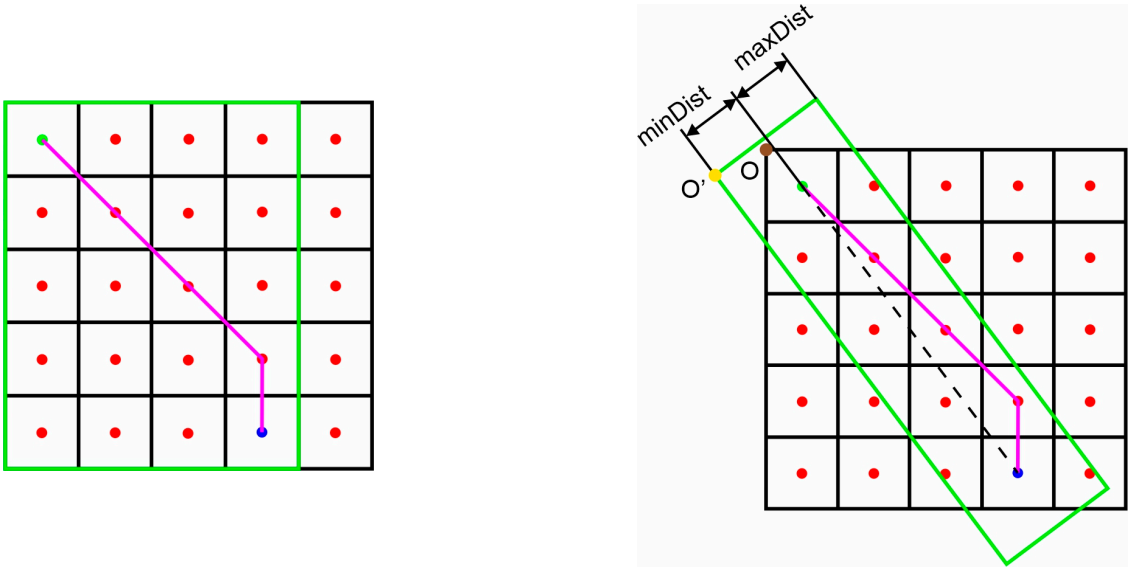
Figure 3. Divided grid environment: (**a**) grid divided into an odd number of subregions and (**b**) grid divided into an even number of subregions.

The simple grid environment method is effective but eventuates in a large number of points and possible path options that are not used from the algorithm or do not lead to the optimal solution. Therefore, the advanced grid environment method is proposed in order to minimize the number of discrete points that do not contribute to finding the optimal path. The advanced grid environment method creates an environment that includes only the necessary points around the optimal path, eliminating all other points that are far away from it. Firstly, a straight reference line is created that connects the projection coordinates of the start and end points in the reference plane. The green and blue points in Figure 4b represent these points, respectively. Then, above and below the reference line, two parallel lines are created with minDist and maxDist, respectively. Initially, as shown in Equation 1, these distances are equal:

$$\text{minDist} = \text{maxDist} = \frac{\sqrt{2}}{2} \cdot \text{ad}, \tag{1}$$

where ad is the length of a subregion of the 3D grid. Then, for each point on the path, it is initially checked whether it is above or below the reference line and then its vertical distance from the reference line is calculated. Depending on the area where it is located, it is examined whether the calculated distance is greater than the value of minDist or maxDist. If the distance is greater, the minDist or maxis' value is updated to the point's distance. Equation (2) is used to calculate the perpendicular distance from a point to the reference line.

$$\text{Perp\_dist} = \frac{\left|(x_e - x_s)\cdot(y_t - y_s) - (x_t - x_s)\cdot(y_e - y_s)\right|}{\sqrt{(x_e - x_s)^2 + (y_e - y_s)^2}}, \tag{2}$$



(**a**) (**b**)

Figure 4. New environment modeling methods: (**a**) demarcation of the simple grid environment and (**b**) demarcation of the advanced grid environment.

In Equation (2), the indexes e and s represent two points on the line and the index t represents the examined target point. Therefore, similarly to the simple grid environment, the number n_g is defined. It is also important that the coordinates of the 3D starting and end discrete point match the coordinates of the 2D grid points. For this reason, the size of the subregions is recalculated based on the distance between the starting and end points in a way that satisfies the above constraint and not strictly the n_g number. This has the result that there is not always a complete correspondence between the subareas of the 3D grid

and the 2D grid, but discrete points in the 2D grid have the coordinates of those of the 3D grid. This is achieved by calculating the divisors of the distance and finding the distance that is closest to the one resulting from the inputted n_g. With the new distance and number n_g, the distance of the boundaries of the 2D grid from the start and end point is defined. At this distance, two vertical lines are created on the reference line. The n_g number also affects the values of minDist and maxDist. The points of intersection of the vertical lines with the parallel lines are the boundaries of the 2D grid. The upper left point is considered as the new zero point O′(0, 0). The new zero point is shown in yellow in Figure 4b.

Figure 4a presents the simple grid environment and the Figure 4b presents the advanced grid environment.

Due to the fact that the AFSA will be executed on the new grid, the coordinates of the start and end points need to be transferred to point O′(0, 0) of the new grid. Equations (3) and (4) show the formulas used for calculating the coordinates of the points based on point O′(0, 0) of the new grid, where the index new is the result of the new point, p refers to the coordinates of the point based on the point O(0, 0, 0) of the 3D grid and ref indicates the coordinates of the point O′(0, 0) from the point O(0, 0, 0) of the 3D grid.

$$x_{new} = x_p - x_{ref}, \tag{3}$$

$$y_{new} = y_p - y_{ref}, \tag{4}$$

For the advanced grid environment method, it is necessary to additionally calculate the rotation angle of the new coordinate system. A line parallel to the Y-axis is created starting from the reference point O′(0, 0). The angle θ between the new line and the axis y′ of the new system is calculated. Equations (5) and (6) are used for rotating the system clockwise and counterclockwise, respectively, where the index final refers to the coordinates of the rotated system and new to the coordinates of the transferred system.

$$\begin{bmatrix} x_{final} \\ y_{final} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix}, \tag{5}$$

$$\begin{bmatrix} x_{final} \\ y_{final} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix}, \tag{6}$$

2.3. Identification of the Path's Critical Points

The start and end points of the optimized 2D paths are defined by the critical points of the 3D path. The critical points are identified by taking into account the starting and end points of a 3D path. Then, the 3D path is projected on the 2D reference plane. For the advanced grid environment, an additional procedure is followed in which a search for the monotonicity changes and the inflection points of the generated function is conducted. These points are further investigated as to their proximity to an obstacle that justifies a change in the path's monotonicity or determines if it is a potential error in the final path. The nearby obstacle search is conducted in the 3D environment. If there is an obstacle near the investigated point, it is considered a critical point. The 2D optimization is conducted between critical points. Due to the fact that the final path must be continuous and gaps or random jumps in the robot's motion should not appear, if there are more than 2 critical points cp = (1, 2, . . . n_c), the end point of a path is considered as the starting point of the next one until $n_c - 1$ paths are formed, where n_c is the number of the critical points.

Figure 5 shows an example of critical point identification for simple grid cp = (1, 53) and for an advanced grid cp = (1, 30, 53).

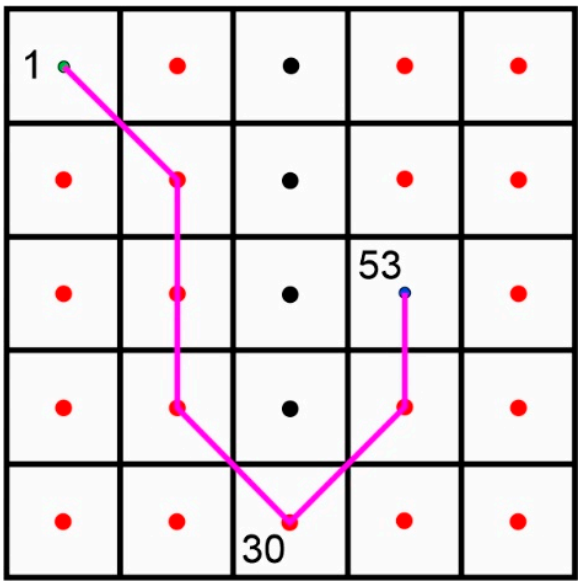


Figure 5. Critical path identification example.

2.4. Obstacles

The existence of the obstacles in the grid is expressed by prohibiting the movement of the robot in certain discrete points. The free discrete point is assigned as 1 and the obstacle as 0. The obstacles registered on the intersection plane of the 3D grid are automatically recorded in the 2D grid environment as well. The recording is performed by taking into account the area that represents the discrete point in the 3D grid. In the 2D grid, the same area is transferred, but because it is denser, the multitude of discrete points is larger. For the case of the advanced grid, it performs an additional process. Often, some points of the 2D grid are located outside of the area defined by the 3D grid. Because this area is unknown, to ensure the integrity of the robot, these points are registered as obstacles. This fact favors the use of the advanced grid even more because the points where the optimal path can be found and unproductive path searches are limited.

The main disadvantages of transporting real obstacles to a grid environment is that even if an obstacle does not fill the whole area represented by a discrete point, the expansion method is used to extend obstacle's dimensions to fill the entire area of the discrete point. In a thin 3D environment, this phenomenon can lead to finding an optimal path for the given 3D environment but, in the real world, the robot is located far away from the obstacles and travels extra distances or chooses longer routes to its destination, especially in an environment with dense small obstacles. It is important to mention that the denser 2D grid allows a more detailed representation of real-world obstacles and reshaping of 3D obstacles through the addition or removal of additional obstacle discrete points.

Figure 6a shows an example of the registration of obstacles on the 3D grid, Figure 6b shows the obstacle registration of the advanced grid environment, and Figure 6c shows the obstacle registration of the simple grid environment.

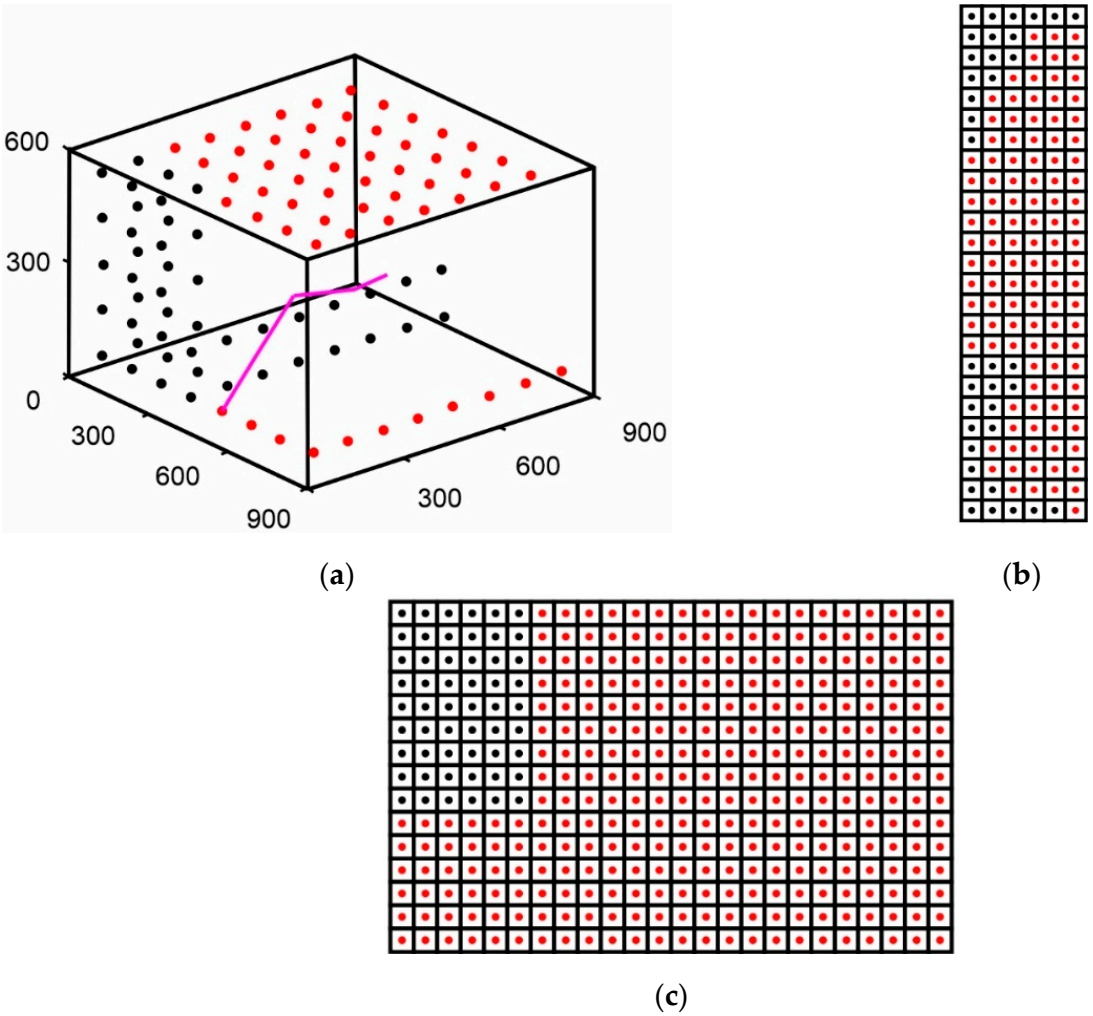


Figure 6. (**a**) Three-dimensional environment with obstacles; (**b**) advanced grid obstacle registration; and (**c**) simple grid obstacle registration.

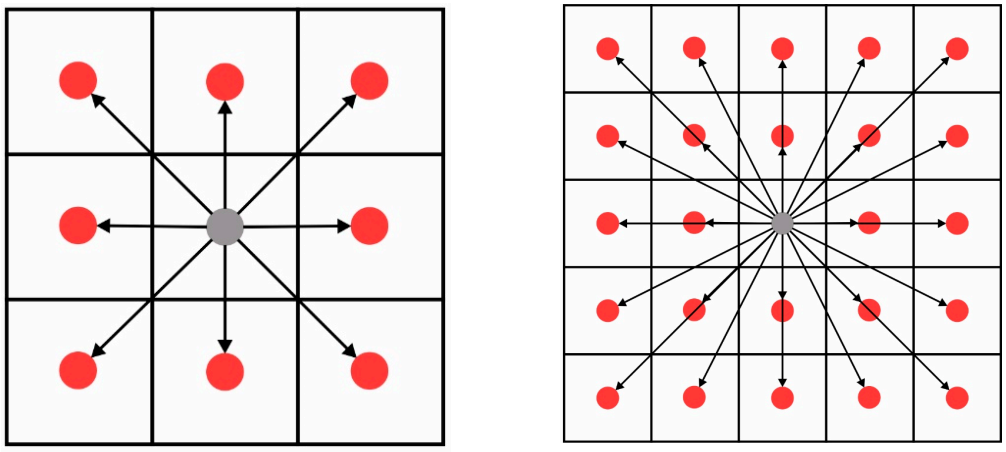
2.5. The Artificial Fish Swarm Algorithm

The AFSA simulates the natural behavior of a fish swarm for solving optimization problems and exploits the fish swarm information and their formation in the environment to solve the path planning problem.

2.5.1. Robot Movement

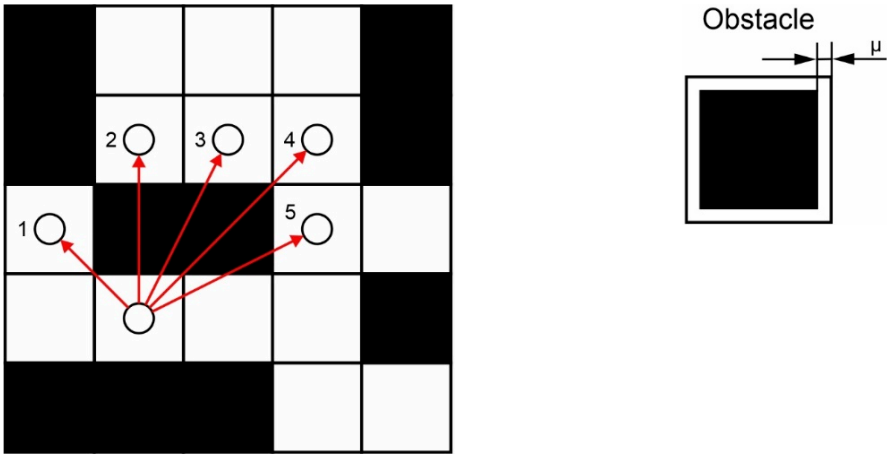
In the grid environment, it is very common among researchers to use a model with 8 possible points for robot navigation from a definite position to the next. In this paper, a model with 24 possible points was used, which greatly increased the possible directions and range of movement of the artificial fishes. Figure 7a shows the model with 8 possible points and Figure 7b shows the model with 24 possible points.

The use of the 24 point model instead of the 8 point model creates a phenomenon where some of the possible points are not obstacles but if the robot travels to them, it will collide with an interim obstacle. Figure 8a shows the 5 possible scenarios of the described phenomenon. In the first scenario, the path connecting the two points passes exactly through the coordinates of the obstacle. In order to eliminate the phenomenon of obstacle over-expansion and not reject the connection between the 2 points, the shrinking coefficient μ was introduced, which uniformly reduces the dimensions of the obstacle by a value μ. Figure 8b shows the results for the shrinking coefficient μ.



(**a**) (**b**)

Figure 7. Models with different possible navigation points: (**a**) 8 possible navigation points and (**b**) 24 possible navigation points.



(**a**) (**b**)

Figure 8. Obstacle representation: (**a**) obstacle intersection scenarios for 24 possible navigation points and (**b**) shrinking coefficient μ.

In scenarios 2–5, the formed path connecting the definite position at which robot is located with the possible navigation points of interest intersects the sides of the obstacle grid square in at least 1 point. Thus, for any given obstacle point located within the robot's range of motion, the 4 straight line segments that define the obstacle's grid square are identified. A simplified ray casting algorithm was used to detect the potential obstacle collision. A ray is created from the robot's definite position to the next possible point and it is examined if there is any points of intersection with any of the identified obstacles' line segments. The ray is considered as a straight line segment. Its bounded coordinates are defined by the coordinates of the robot's definite position and the next possible point. Equations (7) and (8) are used to identify if there is an intersection point between two straight line segments.

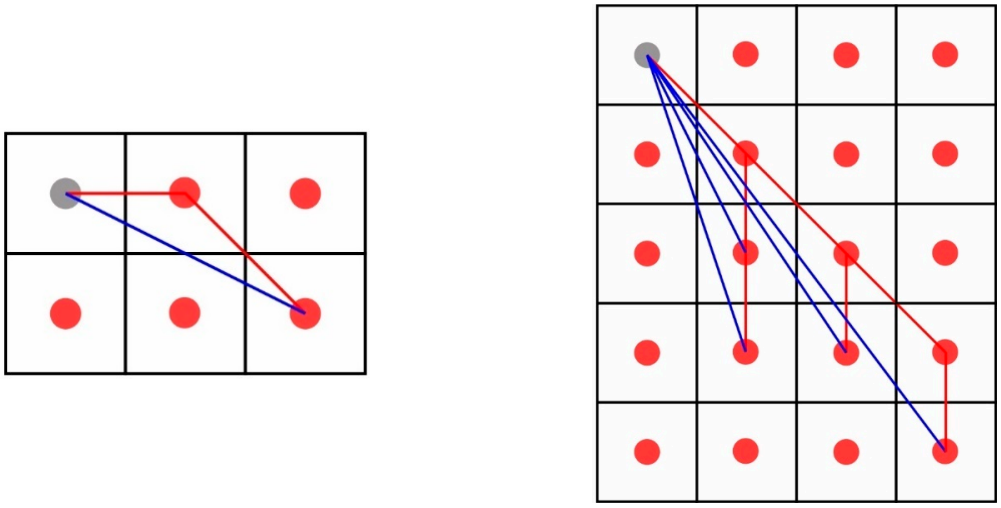
$$t = \frac{(x_1 - x_3)\cdot(y_3 - y_4) - (y_1 - y_3)\cdot(x_3 - x_4)}{(x_1 - x_2)\cdot(y_3 - y_4) - (y_1 - y_2)\cdot(x_3 - x_4)}, \tag{7}$$

$$u = \frac{(x_1 - x_2)\cdot(y_1 - y_3) - (y_1 - y_2)\cdot(x_1 - x_3)}{(x_1 - x_2)\cdot(y_3 - y_4) - (y_1 - y_2)\cdot(x_3 - x_4)}, \tag{8}$$

An intersection point will exist if $0 \leq t \leq 1$ and $0 \leq u \leq 1$. The index 1 represent the robot's definite position, 2 the next possible point, 3 the starting point of the straight line segment of the obstacle, and 4 the end point of the straight line segment of the obstacle.

2.5.2. Grid Movement Analysis

In a grid environment, in order to quickly and efficiently find an optimal path, due to the finite computational capabilities and the architecture of existing path planning algorithms, the robot's movement is limited to a finite range of points such as the 8 or 24 points mentioned above. So, for the robot to navigate from a starting to an end point while in the physical world, it could move linearly without being constrained from its kinematic model, or when moving from an obstacle, it is forced to follow the optimal path resulting from the limitation of the finite range of motion. Figure 9a shows with a red line an example of the finite range of points phenomenon and with a blue line the real-world optimal path. An analysis was conducted to study how the above constraints affect the length of the optimal path.



(**a**) (**b**)

Figure 9. Optimal path length analysis in the real world and the grid environment: (**a**) comparison between the finite range of points phenomenon and real-world point movement; and (**b**) expansion of the example in Figure 9a to more points.

The analysis compares the length of the optimal path obtained by applying the method of 8 possible points with the length of the path that would result if this restriction did not exist and the two points could be joined by a straight line. The analysis is carried out in an environment without obstacles so that its result is as unfavorable as possible for replacing the real-world environment by the grid and highlight as much as possible the disadvantages of the constraint. Figure 9b shows the optimal path resulting from the constraint for 4 points in red and the real-world optimal path in blue.

The real-world optimal path is always shorter. Figure 10 shows the percentage of Equation (9) for 9 points. The Y-axis contains the value of the percentage reduction in the path length for the case where the robot moves from a starting to an ending point without being constrained by the method of 8 possible movement points. The starting point is considered to be located on the first row of the first column of the 2D grid environment. The X-axis contains the row number of the grid where the ending point is located. Meanwhile, the formed curve refers to the number of the grid column where the end point is located. For the first row and column, no analysis was needed because the routes were identical. Percentages on the Y-axis are mapped to integer values on the X-axis.

$$P\% = \frac{d_c - d_r}{d_c} \times 100\%, \quad (9)$$

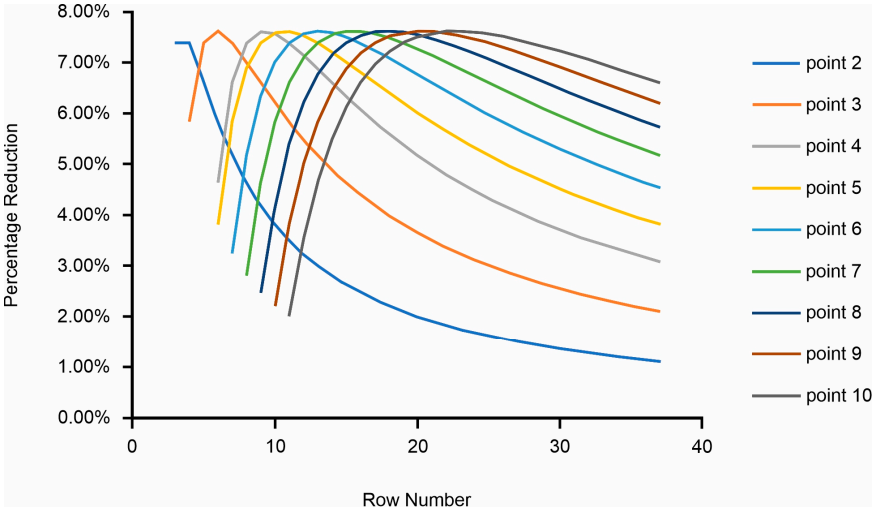


Figure 10. Percentage decrease in the grid movement analysis.

The value d_c represents the length of the constraint path and d_r the length of the real-world optimal path. It is observed that even for the points that are far away from the starting point, such as the 7th point, the maximum path reduction percentage remains the same as the previous points at rate of 7.61%. The difference is observed in the distance of the point where the maximum reduction is underlined from the reference point on the optimal path. Table 1 summarizes the maximum percentage of length reduction for each curve.

Table 1. Maximum percentage reduction for each curve.

	2	3	4	5	6	7	8	9	10
MAX	**7.38%**	**7.61%**	**7.61%**	**7.61%**	**7.61%**	**7.61%**	**7.61%**	**7.61%**	**7.61%**

2.5.3. Traditional Artificial Fish Swarm Algorithm

The AFSA finds the optimal path by reproducing the preying behavior, swarming behavior, following behavior, and random behavior of fish swarms in nature. The following parameters need to be set for the algorithm's operation:

- The number of maximum iterations, MaxIt;
- The maximum number of steps until a complete path formed, MaxStepsNum;
- The population's size of the fish swarm, numFish;
- The fish swarm's number of attempts to find the optimal location, fishTryNum;
- The crowding factor, δ;
- The range of the visual field;
- The step size of the fish.

Figure 11 depicts the model of AFSA. X_i (i = 1,2, . . . N) is the position of the swarm in the search area, X_j is the next possible position of the swarm, and $Y_s = f(X_i)$ is the value of the objective function calculated using Equation (10).

$$f(x, y) = \sqrt{\left(x_i - x_{ending}\right)^2 + \left(y_i - y_{ending}\right)^2}, \quad (10)$$

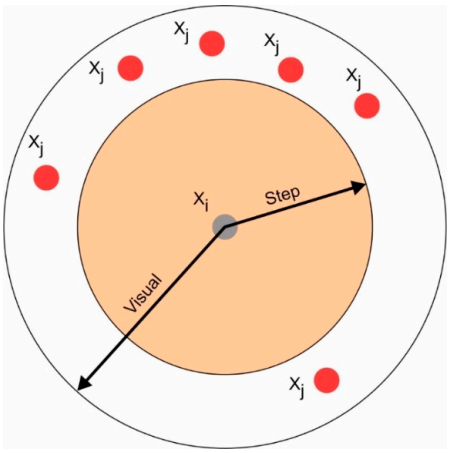


Figure 11. Artificial fish swarm algorithm environment model.

Prey behavior is the most important behavior in the AFSA. In this behavior, the fish randomly selects another position in the visual field, as shown in Equation (11). Then, it compares the value of the objective function Y_j with the value of the current location Y_i. If the value of Y_j is higher than Y_i, the artificial fish will move one step closer to the new position, as shown in Equation (12). If the value is lower, the artificial fish will repeat the process until a higher concentration position is found or the reselection number is higher than the attempt number, fishTryNum. If the fish cannot find a position with higher value, random behavior is activated. This process allows the fish to search for positions using the objective function. In Equation (12), $||X_j - X_i|| = d_{ij}$ is the distance between the current location i, the next possible location j, and the time t.

$$X_j(t+1) = X_i(t) + \text{Visual} \cdot \text{Rand}(), \tag{11}$$

$$X_i(t+1) = X_i(t) + \frac{X_j(t) - X_i(t)}{||X_j(t) - X_i(t)||} \cdot \text{Step} \cdot \text{Rand}(), \tag{12}$$

Swarming behavior is a behavior often found in natural fish when searching for food or avoiding danger. The artificial fish uses it to increase the efficiency of exploration by evaluating the number of fish in the current search area. The fish calculates the center position based on the distribution of its partner X_j. Two conditions need to be satisfied for choosing the center position. Firstly, the value of objective function needs to be higher than the current one, and, secondly, the center position needs to not be densely crowded. The center position is calculated using Equation (13), where n_f is the number of fish in the examined position. Equation (14) is used to calculate the crowding degree of the current fish population, $\delta \in (0,1)$. If the above conditions are not satisfied, then prey behavior is performed. Equation (15) moves the fish a step toward the center position.

$$X_c = \sum_{j=1}^{n_f} \frac{X_j}{n_f}, \tag{13}$$

$$\frac{n_f}{N} \leq \delta, \tag{14}$$

$$X_i(t+1) = X_i(t) + \frac{X_C(t) - X_i(t)}{||X_C(t) - X_i(t)||} \cdot \text{Step} \cdot \text{Rand}(), \tag{15}$$

In the following behavior, the artificial fish observes the position of the other fish in its visual field. If the value of the objective function in the optimal position is higher than the current position and the position is not crowded, the artificial fish moves one step toward

the optimal position. If the above conditions are not satisfied, the artificial fish performs prey behavior. Equation (16) is used to perform the step to the optimal position.

$$X_i(t+1) = X_i(t) + \frac{X_j(t) - X_i(t)}{|| X_j(t) - X_i(t) ||} \cdot \text{Step} \cdot \text{Rand}(), \quad (16)$$

The random behavior is a default behavior for prey behavior. The fish randomly moves within its visual field. This type of movement allows fish to unguidedly explore possible paths and avoid a reselection of the local optimum. Equation (17) describes the random movement process.

$$X_i(t+1) = X_i(t) + \text{Step} \cdot \text{Rand}(), \quad (17)$$

2.5.4. Improved Artificial Fish Swarm Algorithm

The traditional AFSA with some improvements was used to find the optimal path in a grid environment with obstacles.

In prey behavior, when the number of tries fishTryNum is exceeded, instead of activating the random behavior from the examined points, the one with the highest value of the objective function is chosen.

The traditional AFSA has no other means of approaching the target point besides the objective function. The fish often deviate from the optimal path and create long paths that consume computational time and do not contribute to finding the optimal path. A maximum number of steps "MaxStepsNum" until a complete path formed is defined. When the points of the formed path exceed it and the fish has not reached its final destination, the formed path is discarded. Thus, the formation of a new path starts from the beginning until the maximum number of iterations "MaxIt" is reached. A new intermediate maximum number of steps "intMaxStepsNum" is introduced where if the fish has not reached or exceeded some coordinates for a given number of steps, the formed path is discarded. This number is applicable in environments where the optimal path must necessarily pass through some given point, for example, between some obstacles, as shown in Figure 12. The yellow point is a point that the optimal path must definitely pass. These given points, as well as the necessary number of steps until the fish reaches them, are found by the 3D path.

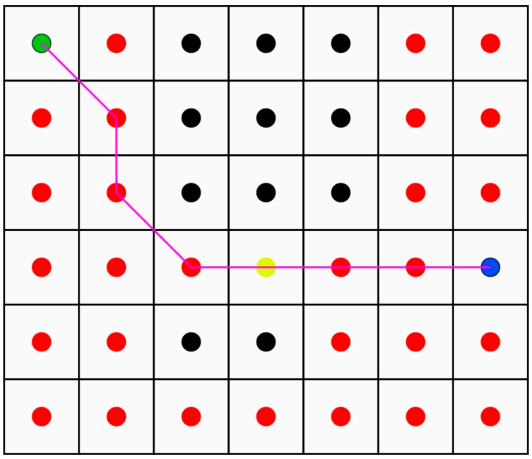
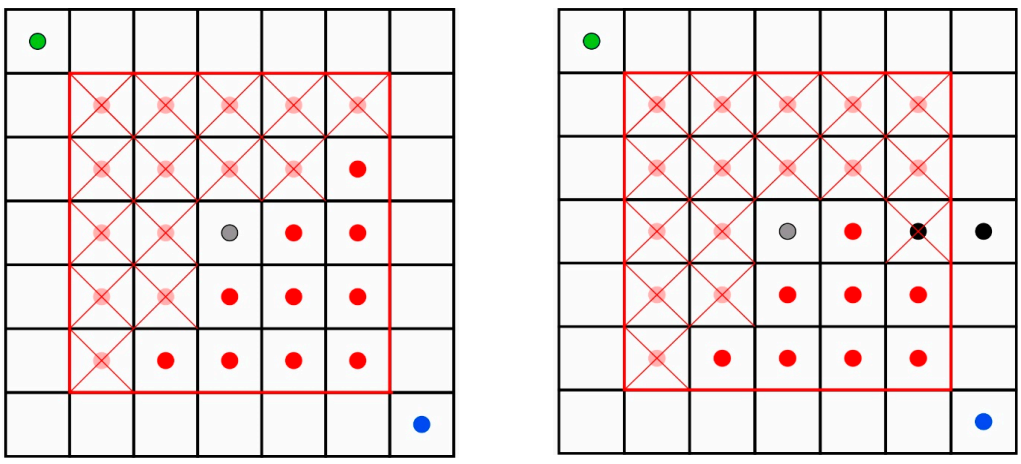


Figure 12. An example part of an environment where the optimal path must pass through a specific point.

The view range and the step have important roles in the traditional AFSA. A large step size can lead to unnecessary oscillations and make it difficult to find the optimal path. On the other hand, if the step size is too small, it significantly affects the speed of the fish and the convergence of the algorithm. An extended view range allows the fish to concentrate

more easily on the search for global solutions and speeding up the algorithm's convergence from the early stages. However, the convergence speed gradually slows down in later stages. However, a shorter view range transfuses the fish with the ability to search for local solutions but the computational time is increased, slowing down the algorithm's converge speed and increasing the probability of falling into local extrema.

The model with 24 possible navigation points used in this paper allows a range of steps $\{1,\ \sqrt{2},\ 2,\ \sqrt{5}, 2\cdot\sqrt{2}\ \}$ and the view range remains constant. The different step sizes and the multiple direction choices assist the fish in exploring the environment with fewer iterations without significantly affecting the algorithm's convergence speed. However, the number of possible next movement points is quite large and, due to the random movement of the fish, it is quite difficult to form an optimal path in a dense environment without postprocessing the path using optimization algorithms. Also, the AFSA does not interact with the environment and the previous solutions; as a result, the wrong selected points cannot be corrected by the next swarm of fish. This problem was solved by a priori eliminating points whose selection distances the fish from the optimal path. In the first stage of elimination, the distance d_{pe} of all possible points separate from the ending point is calculated. Then, for each point, the distance d_{pe} is compared with the distance d_{ce} from the point where the fish is located to the ending point, and if $d_{pe} > d_{ce}$, the point is eliminated. Figure 13a shows an example of the first elimination stage. The position of the artificial fish is represented in gray, the starting point in green, the final point in blue, and the X symbol in the grid cells indicates the elimination of these points. It is also proposed a second stage of elimination, where the effectiveness of the movement of the fish to each of the remaining points is examined. Assuming that the fish has moved to the new examined point, its possible movement points are detected and the first stage of the elimination algorithm is performed for the new examined point. Then, the new points are compared with the point where the fish is actually located, and if its position is more favorable, the examined point is deleted. The existence of common points between the examined point and the point where the fish is located is also examined. If, after eliminating the possible points of the examined point, only the points where the fish can move from its current position remain, then the examined point is eliminated again. Figure 13b shows an example of points that were eliminated in the second stage of elimination.



(**a**) (**b**)

Figure 13. Elimination stages: (**a**) first stage of the elimination and (**b**) second stage of the elimination.

In order to increase the efficiency of the fish's navigation and to facilitate the finding of the optimal path, the obstacle heatmap is suggested. The map contributes to the exchange of information between the algorithm and the environment. The map reflects the location and the effect that obstacles may have on finding the optimal route. The following parameters are used to generate the map:

- The heat value of the obstacle, obsHeatVal;
- The coefficient of increasing the heat across the map, heatIncrCoef;
- The updated heat value of the points in the first heatmap, obsNearHeatVal;
- The updated heat value of the points in the second heatmap, obsNearHeatVal2.

Each district point of the 2D environment corresponds to a point on the map. Firstly, for every obstacle point in the grid environment, an initial heat value of the obstacle, obsHeatVal, is defined on the map. In the second step, each of the obstacles is considered separately as a point where the fish can move freely on it, and for the points of its visual field, their values are defined on the obstacle heatmap according to Equation (18). If there are any obstacles within their visual field, their value on the map remains as it was defined.

$$\mathrm{hv} = \mathrm{hv} + \frac{\mathrm{d_p}}{2 \cdot \sqrt{2}} \cdot \mathrm{obsNearHeatVal} \cdot (1 + \mathrm{heatIncrCoef}), \tag{18}$$

In Equation (18), hv is the heat value of the point of interest on the obstacle heatmap and $\mathrm{d_p}$ is the distance of the point from the obstacle. After completing the process for all the obstacle points, the same process is repeated for all the points that the robot can move to freely and their values on the map have been changed. The equation used to update their values on the map is Equation (19). In this part of the map update, in addition to the obstacles, the points whose values have changed on the map since the first part of the process are excluded and their values are not updated again. In this way, the values of the heatmap's points are smoothed and the proximity and the obstacle density of an area are transformed to arithmetic values.

$$\mathrm{hv} = \mathrm{hv} + \mathrm{obsNearHeatVal2}, \tag{19}$$

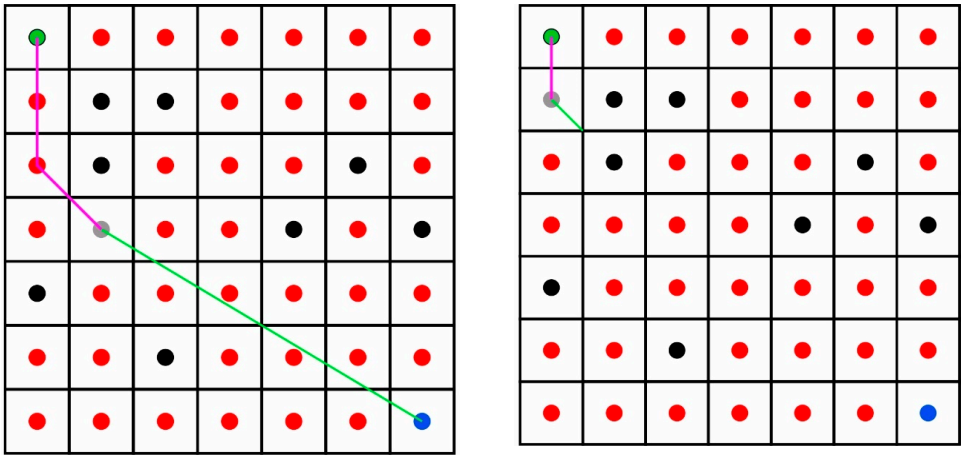
In the final step of creating the map for each point, the heat value is updated according to Equation (20), taking into account the number of the next possible points. This separates areas with a high density of obstacles, as well as areas of the map that have reduced movement possibilities, such as edges and corners.

$$\mathrm{hv} = \mathrm{hv} + \frac{\mathrm{obsNearHeatVal} - \mathrm{obsNearHeatVal2}}{2} \cdot \frac{\mathrm{np}}{24}, \tag{20}$$

In Equation (20), np is the number of the next possible points when the fish is located at the point of interest. The map is used to navigate the fish by comparing the obstacle's heatmap value of the possible points for the fish's position with the average value of all the heatmap's points. Five different comparison areas are created, and for each area, a bonus is used as the objective function of the points that are located at a given distance from the point where the fish is placed. The bonus increases or decreases the value of the objective function by a given percentage, making the points less or more attractive to fish. This encourages fish to move in specific directions and step lengths for the different areas of the map. Also, the bonus for the different comparison areas may vary.

In this paper, the continuous use of the map throughout the navigation process, as well as its random activation or deactivation, are introduced. In the random activation or deactivation of the map during the navigation process, a random number is generated that determines the activation or inactivation of the map.

Another important contribution of this paper is the combination of a simplified ray casting algorithm with the AFSA. In this process, the artificial fish is equipped with an artificial laser, which helps it to navigate in complex and simple environments. The laser casts a ray from the point where the fish is located to the end point. If the ray is not obstructed by an obstacle, the fish reaches the final destination. If the ray detects an obstacle, the AFSA is performed as described above to find a new point and repeat the process. Figure 14a shows an example of casting a laser beam to the end point without an intervening obstacle. Figure 14b shows an example of a laser beam blocked by an obstacle. The laser method is accomplished using Equations (7) and (8).



(**a**) (**b**)

Figure 14. Laser beam casting examples: (**a**) successful laser beam casting and (**b**) a laser beam obstructed by an obstacle.

The distance between two discrete points is calculated using Equation (21).

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad (21)$$

The calculation of the path's total length is expressed as the sum of the distance of the visited points using Equation (22).

$$L = \sum_{a=1}^{q-1} \sqrt{(x_{a+1} - x_a)^2 + (y_{a+1} - y_a)^2}, \quad (22)$$

To convert the resulting 2D path into 3D, the 3D path found on the 3D grid is used. The first step of the process is to plot the resulting 2D path and the 3D path on a common grid, with common start and end points. Then, the 3D path is projected onto the reference plane to be converted to a new 2D path. Then, for each discrete point of the projected path, vertical lines are drawn parallel to the axis where the greatest differences between the paths are found and the points of intersection of the vertical lines with the resulting optimized 2D path are found. Figure 15 shows an example of the process. For each intersection point found, it is assigned the coordinate of the 3D path that is not captured on the reference plane. In the example of Figure 15, it is the coordinate of the Z-axis. In the case where some discrete points have a common intersection point on the optimized path, as the points noted in Figure 15, then the coordinate of the last discrete point is assigned. The linear interpolation method is used for the rest of the points of the 2D path that are not assigned with the third coordinate.

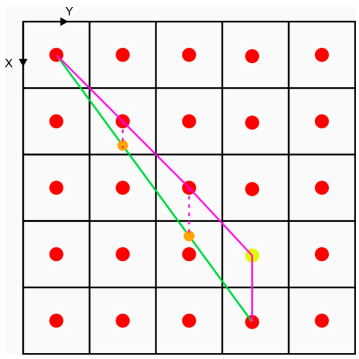


Figure 15. The process of converting the optimal 2D path to a 3D path.

2.6. Flow Charts

Figure 16 shows the flow chart of AFSA behaviors.

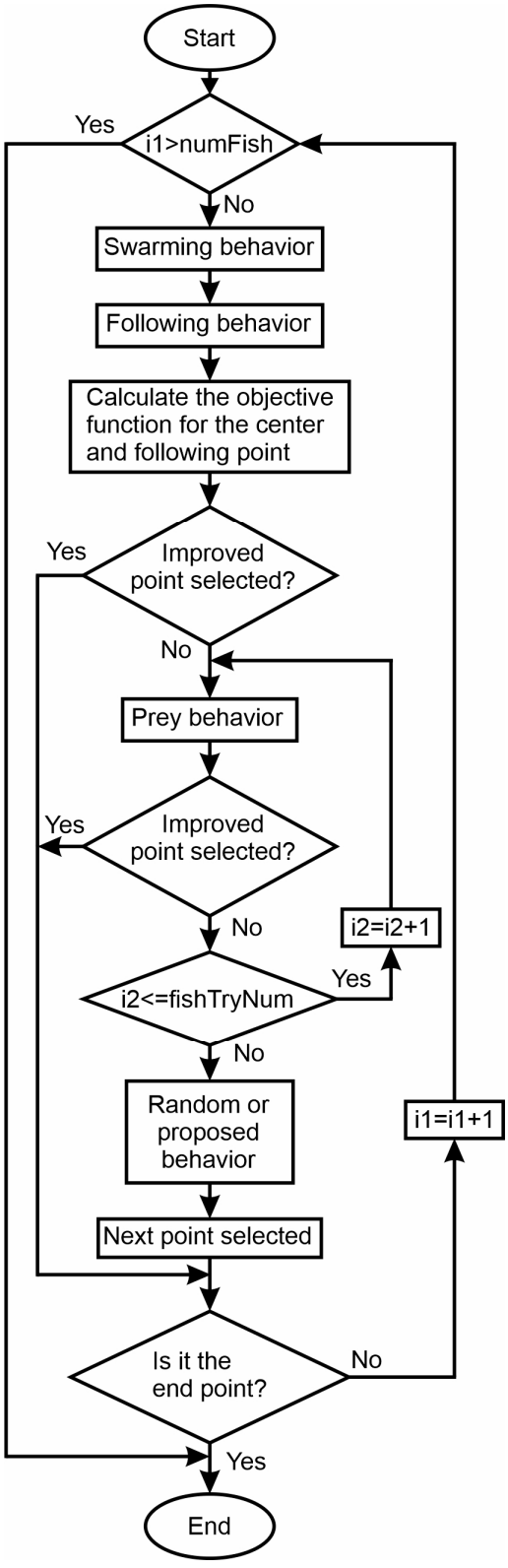


Figure 16. Artificial fish swarm algorithm behavior flow chart.

Figure 17 shows the flow chart of the proposed method.

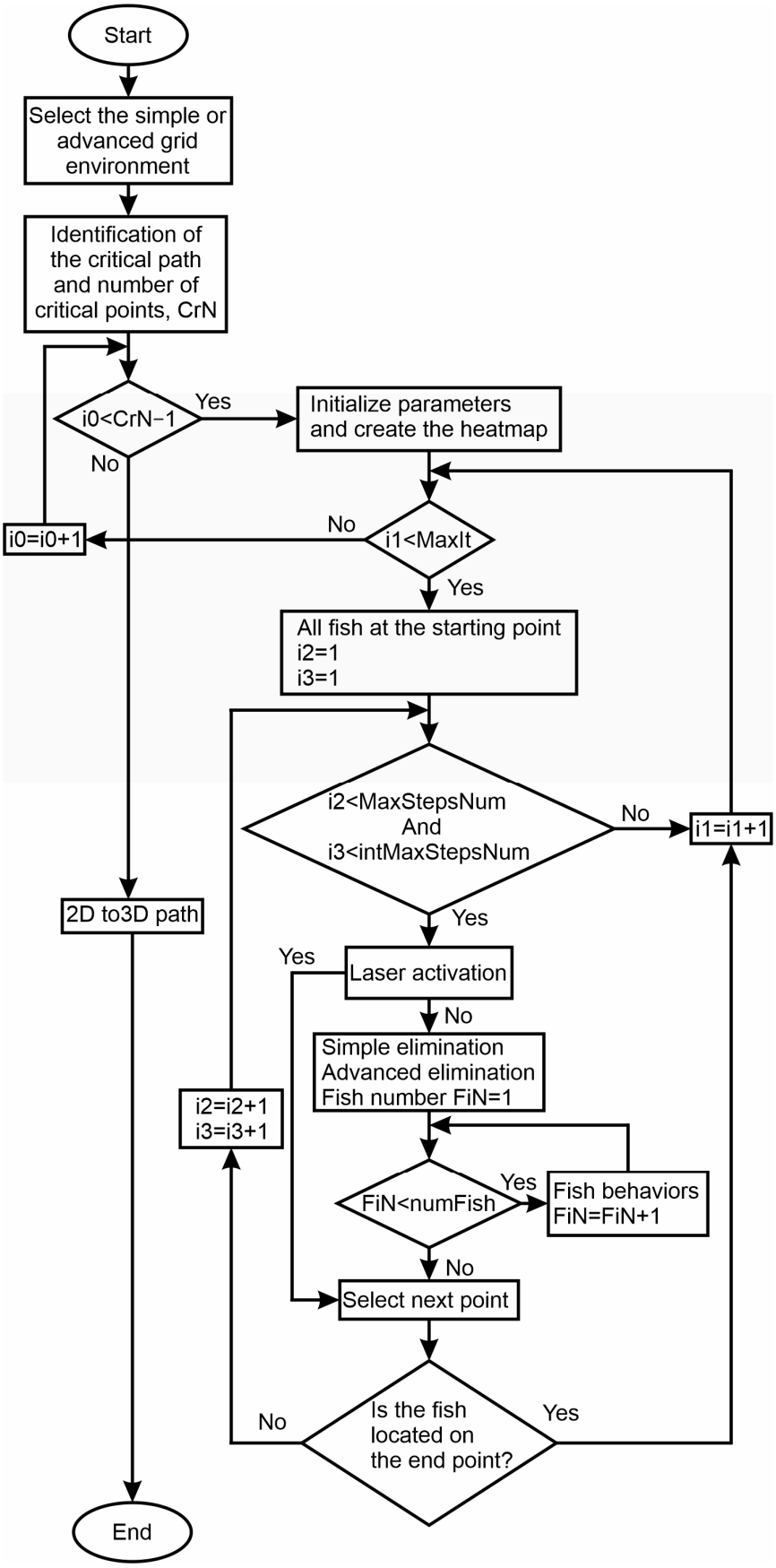


Figure 17. Flow chart of the proposed algorithm.

3. Results

Experiments were conducted to verify the effectiveness and the applicability of the proposed methods. Moreover, the results and the contributions were validated through different scenarios. The comparison between the traditional AFSA and the proposed methods is presented.

Comparing the simple and the advanced grid methods for optimizing the same 3D path depicted in Figure 2a, the same length optimized path is obtained, as shown in Figure 18a,b. In Figure 19a, the route obtained using the method with 24 possible points is plotted in blue, and the route obtained by applying the laser is plotted in magenta. In Figure 18b, the path resulting from the laser is identical to that of the 24 possible movement points. The difference between the two methods is more noticeable when the laser is not activated and the path is formed only with the 24-point method. Additionally, when the laser is not activated, there may be a difference in the length and shape of the formed path, which is due to the finite range of motion of points phenomenon and is verified by Figure 10. For the simple grid method, the length of the path obtained using the laser is 200 while with the 24 possible point method, it is 202.5798. For the advanced grid method, the path length is 200 and remains the same for both practices. The main difference is in the course of the designed path, where in simple grid method, a straight line may turn into a polynomial function.

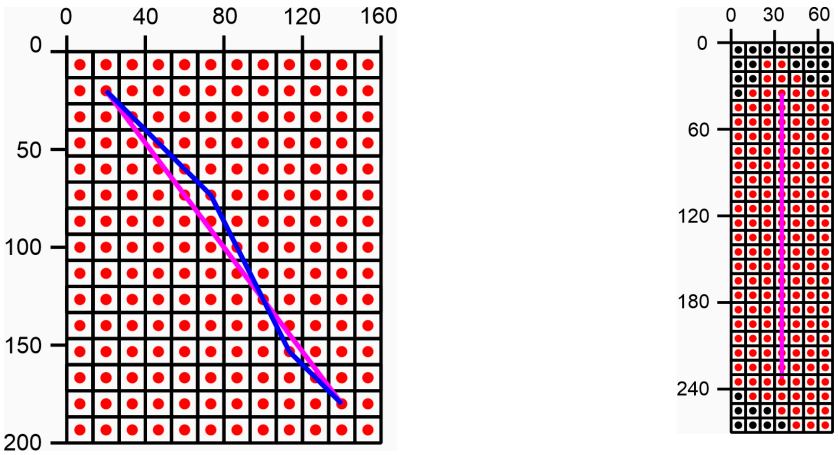


Figure 18. Simple and advanced grid environment creation: (**a**) path planning using simple environment modeling and (**b**) path planning using advanced environment modeling.

The advanced grid method creates discrete movement points for an area of 18,900 mm^2, while the simple grid method creates discrete movement points for an area of 32,000 mm^2. This implies that for the same number of discrete points, the advanced grid method creates a denser grid with more obstacles, thus increasing the accuracy of the planned route, while the presence of obstacles eliminates some choices of the next possible point selection and expedites the fish to find the optimal path without waste time exploring paths that are far from the optimal. On the other hand, the simple grid method allows the exploration of paths in an environment with the same dimensions as those used for the 3D path search. In this way, routes that are significantly different from the optimal path can be examined more extensively.

The proposed AFSA was tested in both environments with and without obstacles. The obstacle-free environment has more possible navigation points and the general behavior of the algorithm can be examined explicitly, while the environment with the obstacle requires advanced simultaneous movement to avoid local optima and obstacles.

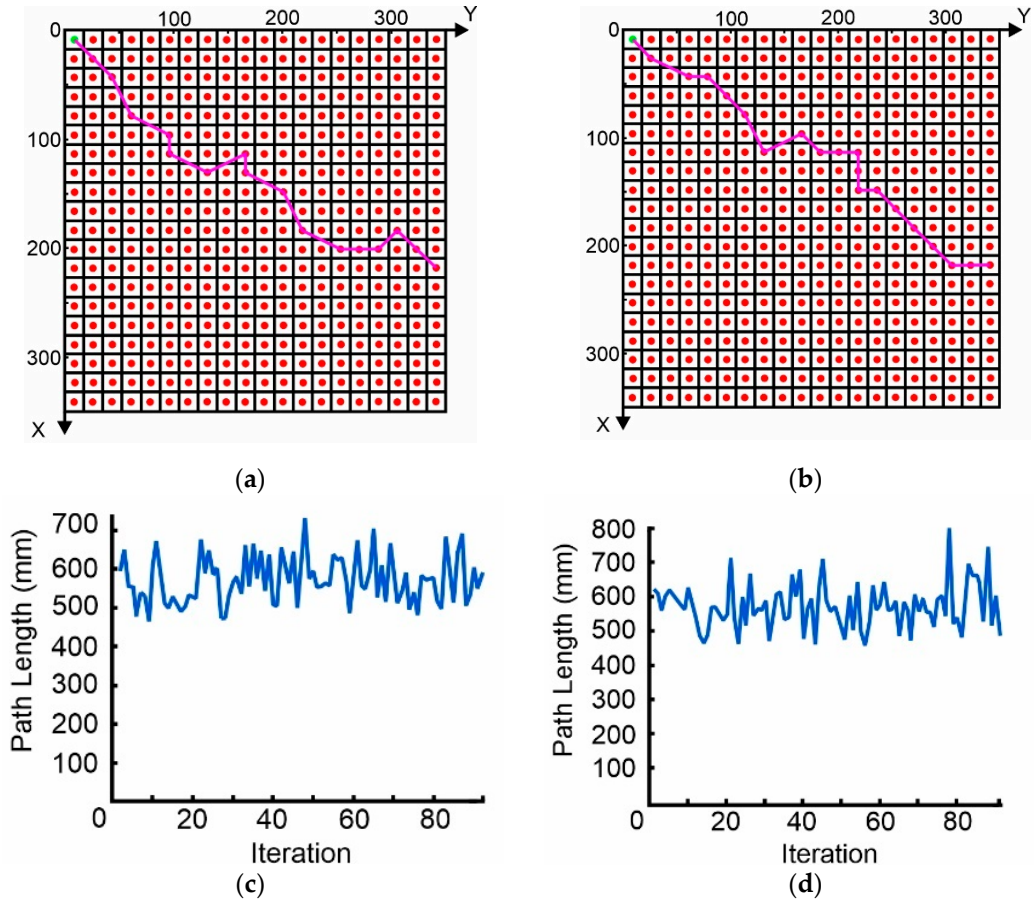


Figure 19. Experimental results based traditional and proposed behaviors: (**a**) path planning using prey and random behaviors; (**b**) path planning using prey and proposed behaviors; (**c**) diagram of the paths formed using prey and random behaviors; and (**d**) diagram of paths formed using prey and proposed behaviors.

The experiments presented for the comparison between the methods were performed in the same environments with the same parameters, namely, MaxIt = 90, MaxStepsNum = 40, numFish = 150, fishTryNum = 15, $\delta = 0.2$, $\mu = 0$, and the 24 possible movement point method. With respect to the obstacle's heatmap creation, obsHeatVal = 150, heatIncrCoef = 0.25, obsNearHeatVal = 90, and obsNearHeatVal2 = 40 were defined. The bonuses were divided in the following way. If the heatmap value, hv, for the point where the fish is located is less than 0.92·mh, then the objective function of the possible points that are distanced da or 2·da is calculated by the coefficient bh = 0.85, where mh is the mean of heatmap values and da is the length of the side from an elementary square of the grid. If hv $\leq$ 1.08·mh and hv $\geq$ 0.92·mh for the points distanced $\sqrt{2}$·da or da, then bh = 0.90. If hv > 1.08·mh and hv $\leq$ 1.28·mh for the points distanced $\sqrt{5}$·da, the bh = 0.82. If hv > 1.28·mh for the points distanced $2\cdot\sqrt{2}$·da or 2·da, the bh = 0.82.

The effects of each behavior on the algorithm's response and the finding of the optimal path are presented elaborately. Figure 19a shows the path formed by the fish algorithm having only the random and prey behaviors. Figure 19b shows the path formed by replacing the random behavior with the proposed behavior. The length of the path with the proposed behavior is 2.63% shorter. It is also noticed that this change has led to a reduction in the optimal path's oscillation, as well as a reduction in pointless wandering. The proposed methodology seems to explore more of the map due to the stronger variations observed in the path length diagram, but the reduction of unpredictable random behavior ultimately

leads to a shorter final path. Figure 19c,d show the diagrams with the lengths of the paths formed during the algorithm execution for each case.

Figure 20a shows the optimal path formed by replacing random behavior with the proposed behavior, prey behavior, and following behavior. A 1.36% reduction in the length of the optimal path is observed with respect to the path's length from the behaviors in Figure 19b. Figure 20b shows the optimal path obtained by replacing the following behavior with the swarming behavior. A 0.25% reduction in path length is observed compared to the results obtained from behaviors in Figure 19b. The path length in Figure 20a is 0.11% less than the path length in Figure 20b. Table 2 summarizes the resulting path lengths for the different behaviors. As demonstrated in Table 2, the length of the path obtained by replacing the random behavior with the proposed methodology is very close to the length of the paths obtained using the traditional swarming and following behaviors with the prey behavior individually.

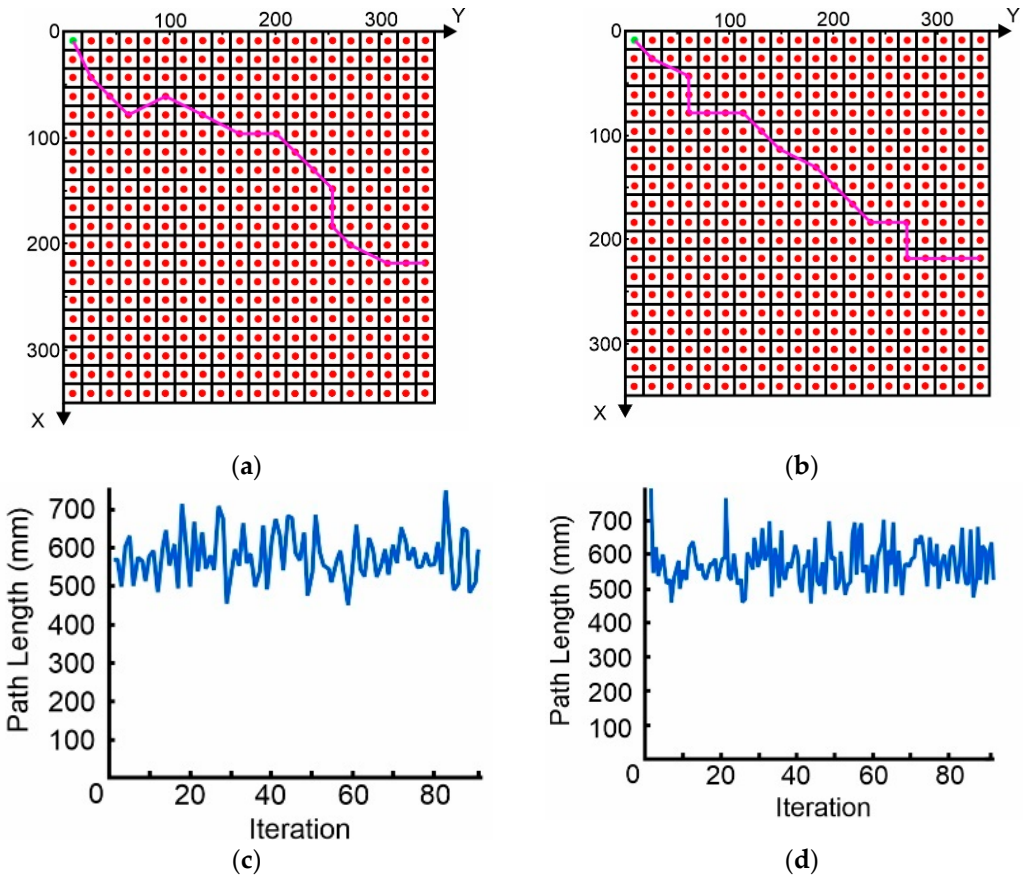


Figure 20. Influence of the AFSA's behaviors on path formation: (**a**) path planning using prey, proposed, and following behaviors; (**b**) path planning using prey, proposed, and swarming behaviors; (**c**) diagram of the paths formed using prey, proposed, and following behaviors; and (**d**) diagram of the paths formed using prey, proposed and swarming behaviors.

Table 2. Lengths of the formed paths for the different tested behaviors.

Behavior	Path Length
Prey + Random	467.6620
Prey + Proposed	455.3835
Prey + Proposed + Following	449.1484
Prey + Proposed + Swarming	454.2548

Figure 21a,b shows the path formed by the AFSA using all the described behaviors in an environment without and with obstacles, respectively. Figure 21c,d shows the diagrams with the lengths of the paths formed during the algorithm execution for each environment. The greater variation observed in the graphs of path lengths formed in the environment with obstacles compared to the obstacle-free environment is owed to the fish's decision to head to areas around the obstacles in order to avoid them, which nevertheless exclude them from the optimal path. The fish succeed in reaching the end point, but the length of their final path is longer than the optimal route. Table 3 contains the path lengths obtained using the proposed AFSA and all four behaviors in environments with and without obstacles.

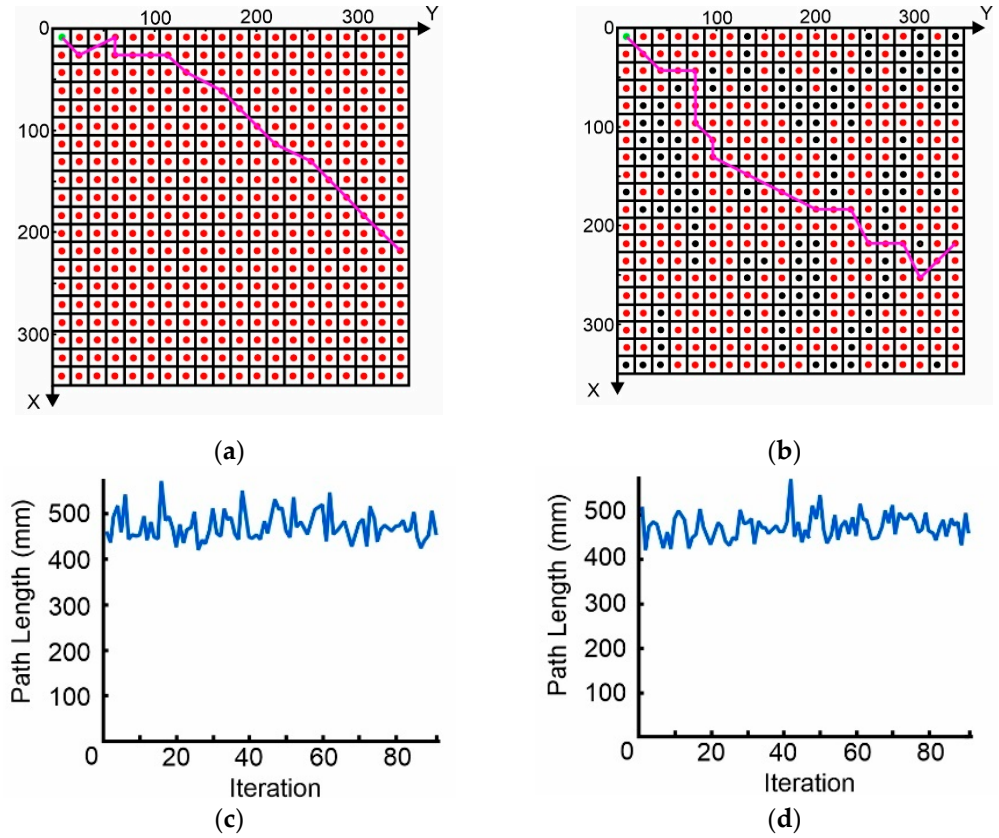


Figure 21. Results of the AFSA with the proposed behaviors in environments with and without obstacles: (**a**) path planning using the proposed AFSA in an obstacle-free environment; (**b**) path planning using the proposed AFSA in an environment with obstacles; (**c**) diagram of the paths formed using the proposed AFSA in an obstacle-free environment; and (**d**) diagram of the paths formed using the proposed AFSA in an environment with obstacles.

Table 3. Lengths of the paths formed in environments with and without obstacles.

Behavior	Path Length	
	Obstacle Free	**Obstacle**
Proposed AFSA	434.8809	494.3996

Figure 22a shows the path formed by applying the initial elimination of the possible movement points and Figure 22b shows the results obtained by additionally applying the advanced elimination. The elimination process seems to favor the selection of points closer to the end point and significantly reduces the unnecessary oscillations. The advanced elimination method appears to encourage the selection of points closer to the target compared to simple elimination. This happens because advanced elimination further reduces the next possible points and the fish is more likely to choose the closest point to the end point

among the available choices. However, it seems that the points closest to the end point are not always chosen, which favors space exploration and a search for alternative routes in an environment with obstacles rather than just the prearranged approach to the final destination and the repeated formation of only one path. Figure 22c,d shows the resulting path length iteration diagrams for each case.

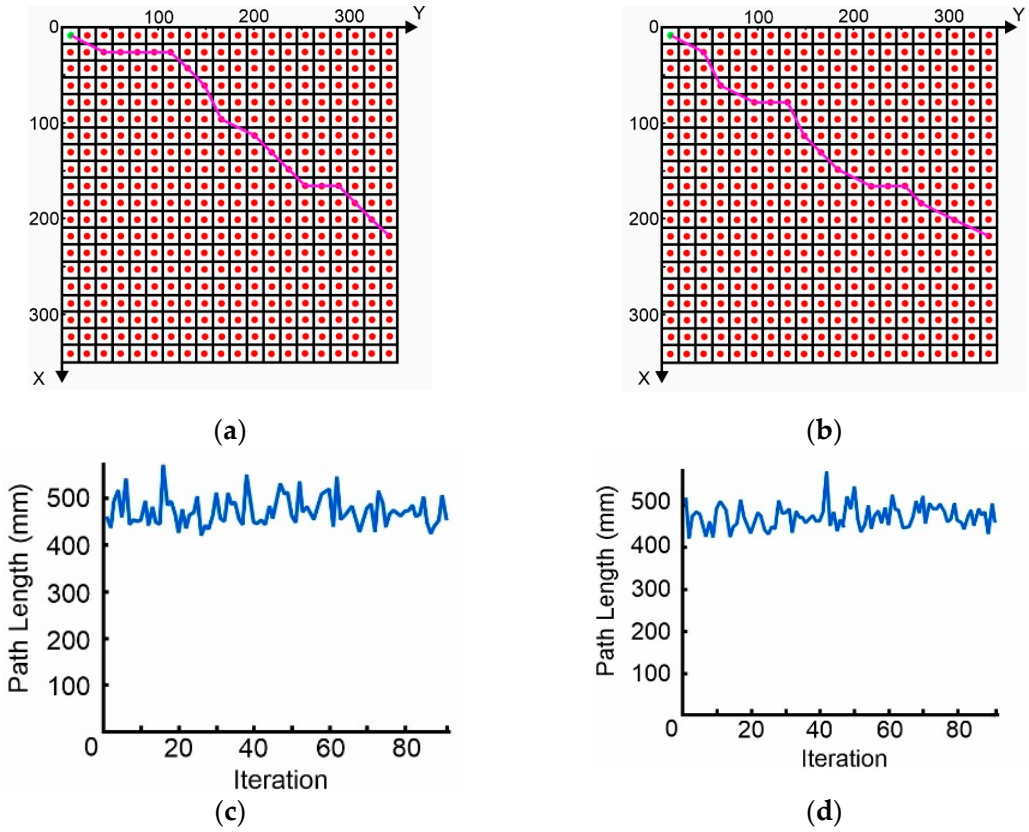


Figure 22. Influences of initial and advanced elimination on the formatted path: (**a**) path planning using initial elimination; (**b**) path planning using initial and advanced eliminations; (**c**) diagram of paths formed using initial elimination; and (**d**) diagram of paths formed using initial and advanced eliminations.

Figure 23a shows the results of a path formed by activating the heatmap and Figure 23b shows the results of randomly activating the heatmap. Figure 23c,d shows the diagrams for each process. The heatmap use seems to stabilize the formed path and eliminate all unnecessary oscillations that lead to an increase in the length of the formed path. It can also be seen from the path length diagrams that the formed routes are shorter compared to the previous methodologies and the solutions with reduced lengths are systematically approached. Table 4 summarizes the results.

Table 4. Lengths of the paths formed using the elimination and obstacle heatmap methods.

Behavior	Path Length
Proposed AFSA + Simple Elimination	420.3835
Proposed AFSA + Simple + Advanced Elimination	418.1645
Proposed AFSA + Simple Elimination + Heatmap	417.2659
Proposed AFSA + Simple Elimination + Random Heatmap Activation	413.2498

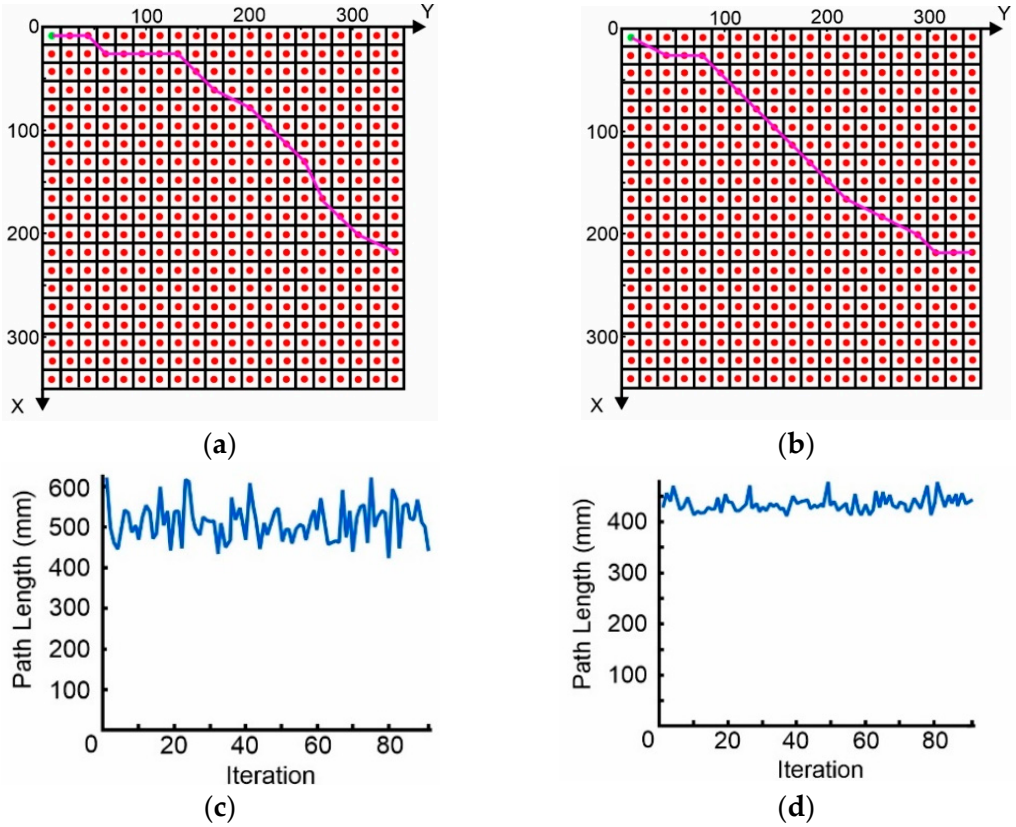


Figure 23. Results of the obstacle's heatmap contribution: (**a**) path planning using the obstacle's heatmap; (**b**) path planning using the randomly activated obstacle's heatmap; (**c**) diagram of the paths formed using the obstacle's heatmap; and (**d**) diagram of the paths formed by randomly activating the obstacle's heatmap.

Figure 24a and Figure 24b shows the results eventuated after the heatmap calculation process for the environments in Figure 21a and Figure 21b, respectively. The numerical values of the resulting heatmaps between the two environments differ significantly.

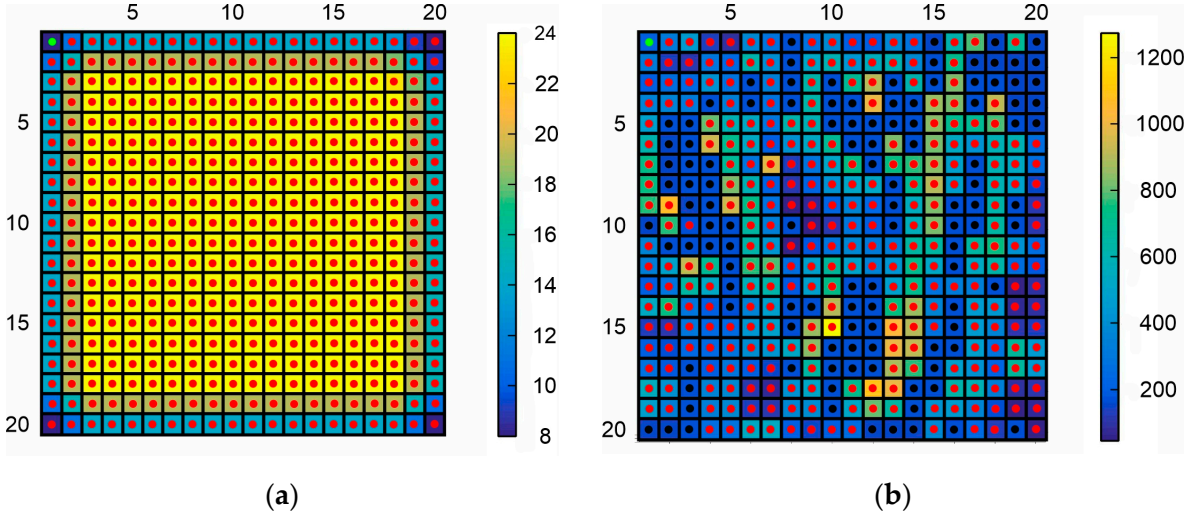


Figure 24. Representation of the obstacle heatmap values: (**a**) resulting obstacle heatmap for an obstacle-free environment and (**b**) resulting obstacle heatmap for an environment with obstacles.

Figure 25a–d shows the results using all the suggestions at once in different environments and Figure 26a–d shows the subsequent diagrams for each case. An environment

without obstacles and three with obstacles are presented. It is worth noting that in the environment of Figure 25b, there are very narrow passages that lead to the end point, while in the environment of Figure 25c, there is one optimal solution and lots of possible solutions lead to dead ends.

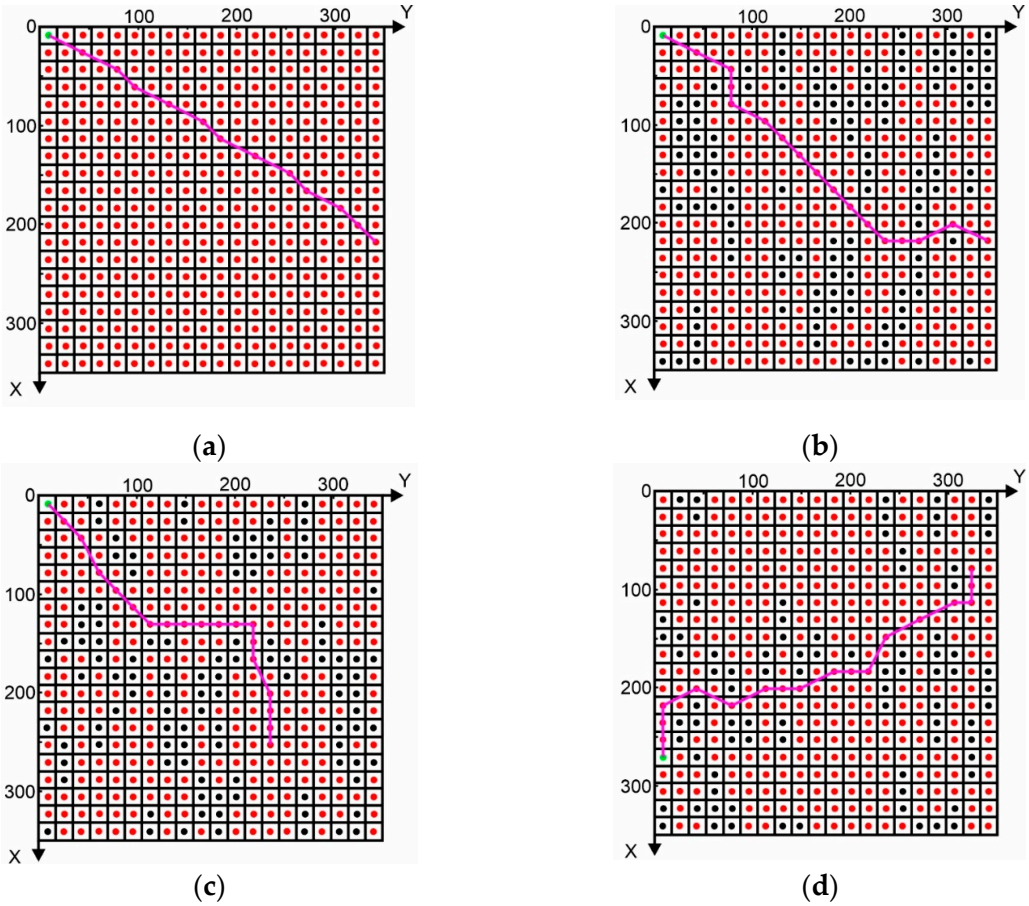


Figure 25. Results of the improved AFSA: (**a**) path planning using the proposed algorithm in an obstacle-free environment; (**b**) path planning using the proposed algorithm in an environment with obstacles; (**c**) path planning using the proposed algorithm in an environment with obstacles; and (**d**) path planning using the proposed algorithm in an environment with obstacles.

The algorithm's behavior is examined in terms of the capabilities of finding the optimal solution among them. In the environment of Figure 25d, the algorithm is tested from a starting point lower than the end point. The optimal path was found successfully for all cases, but especially for the case in Figure 25c, more iterations were needed, while it was the only successful path formed. Table 5 summarizes the results of the path lengths in Figure 25.

Table 5. The lengths of the paths in Figure 25.

Environment	Path Length
a	397.6620
b	442.0147
c	394.5061
d	448.9183

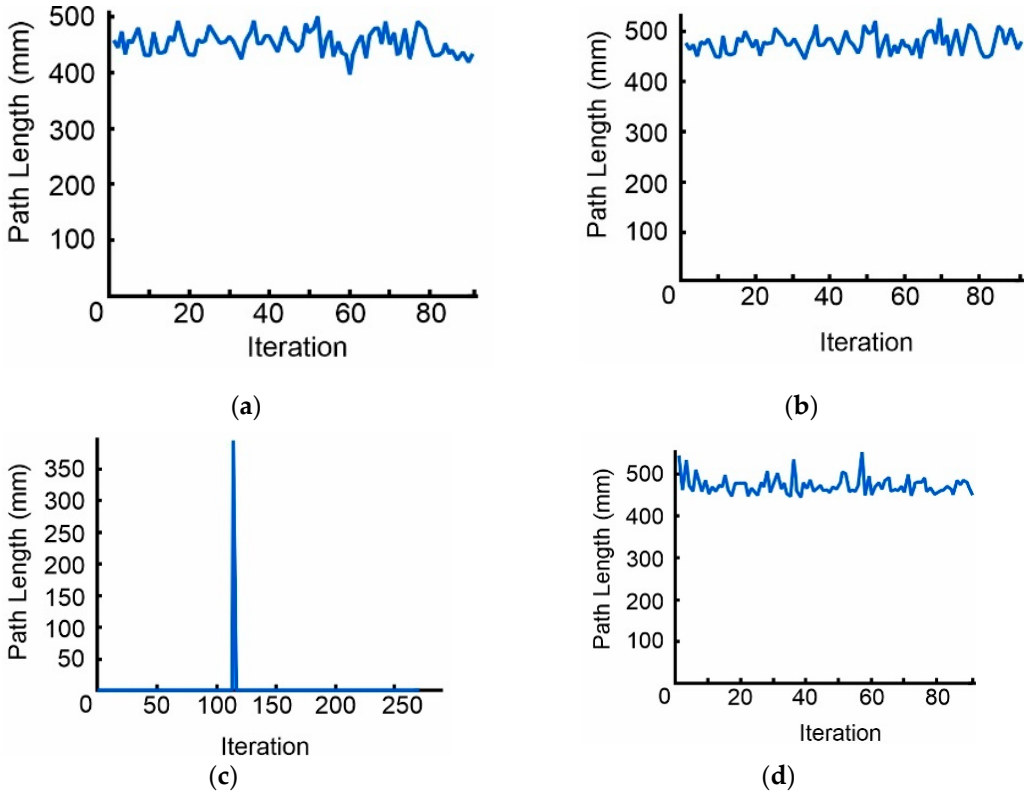


Figure 26. Diagrams of the formed paths for the environments in Figure 25a–d: (**a**) diagram of the formed path for the environment in Figure 25a; (**b**) diagram of the formed path for the environment in Figure 25b; (**c**) diagram of the formed path for the environment in Figure 25c; and (**d**) diagram of the formed path for the environment in Figure 25d.

Figure 27a–d shows the results of combining the AFSA enhanced with the laser beam in different environments with and without obstacles and Figure 28a–d shows the diagram with the lengths of the formed paths, respectively. This combination causes a reduction in the length of the optimal path.

Table 6 summarizes the result of all the above experiments. The integration of the laser for the environments a and c reduces the length of the optimal path by 4.3984 and 2.8875 length units or 1.1% and 0.73%, respectively. In environments b and d, it is not noticed any difference in the length of the formed path between the two methods. This similarity occurs because the end point is located behind an obstacle, and as a result, the laser is activated from a very close distance and the formed path is identical to that obtained by the method of 24 possible movement points.

Table 6. The lengths of the paths in Figure 27.

Environment	Path Length
a	393.2636
b	442.0147
c	391.6186
d	448.9183

The laser is notably useful in cases where the environment has a large area without obstacles or the end point is far from obstacles and can be approached linearly from another point in the environment. The laser would not be able to operate effectively without the AFSA because it would encounter an inability to overcome the obstacles and select a new possible movement point optimally.

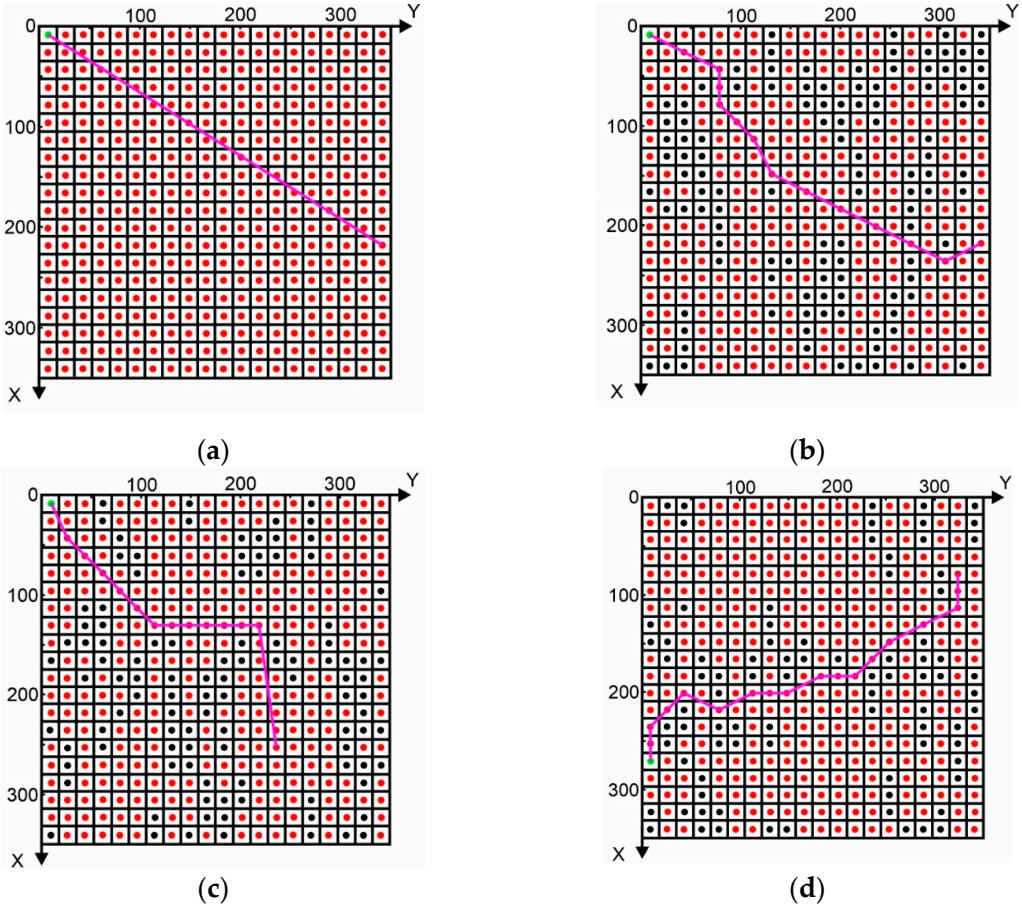


Figure 27. Path planning results using the proposed algorithm integrated with the ray casting algorithm: (**a**) path planning using the proposed algorithm integrated with the ray casting algorithm in an obstacle-free environment; (**b**) path planning using the proposed algorithm integrated with the ray casting algorithm in an environment with obstacles; (**c**) path planning using the proposed algorithm integrated with the ray casting algorithm in an environment with obstacles; and (**d**) path planning using the proposed algorithm integrated with the ray casting algorithm in an environment with obstacles.

Figure 29a shows in blue the result of an optimal 3D path formed by the ant colony optimization method and Figure 29b shows the result after the proposed 2D optimization.

Figure 30a shows in blue the result of a suboptimal path obtained by the ant method and Figure 31b shows the result obtained after the proposed 2D optimization. Table 7 summarizes the results for the lengths of the 3D paths. In the case where the 3D path is optimal, the application of the proposed methodology is reducing the final path by 2.28%. While for the non-optimal 3D path, the final route decreases by 10.01%. It is noteworthy that the non-optimal path does not deviate much from the optimal path, only two district points were selected as suboptimal during its formation process.

Table 7. The lengths of the paths in Figures 29 and 30.

Environment	Path Length
Optimal 3D path	249.0455
Improved optimal 3D path	243.3525
Non-Optimal 3D path	2542.4
Improved Non Optimal 3D path	2287.9

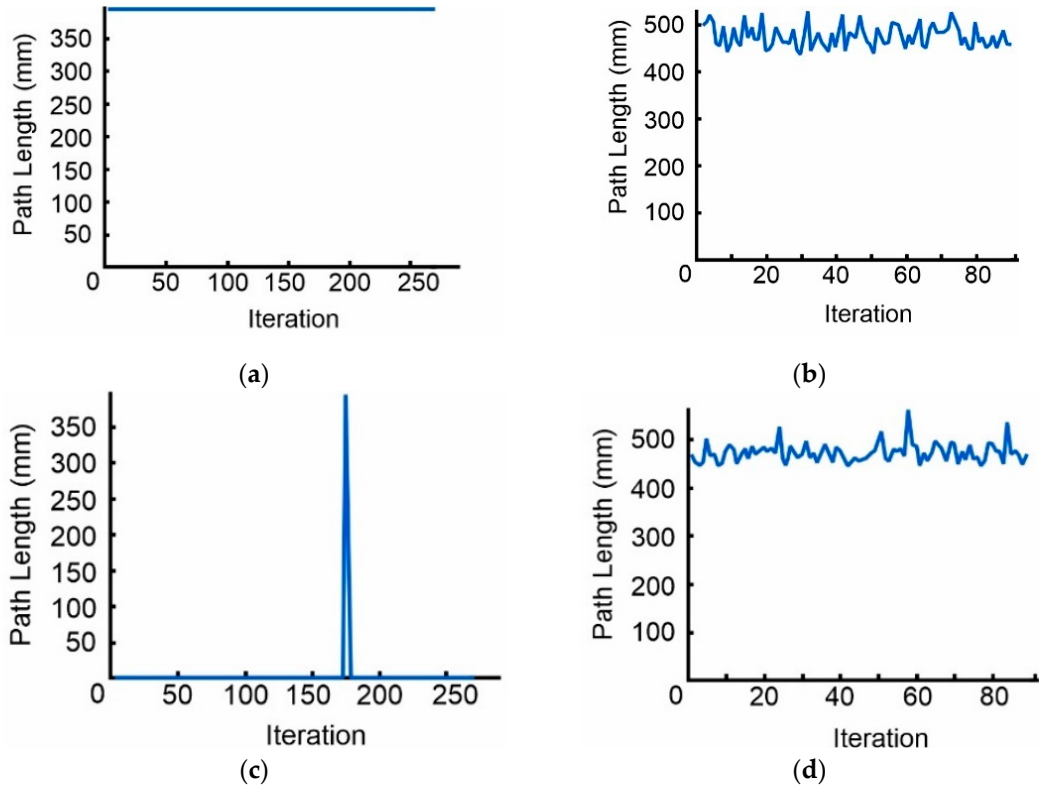


Figure 28. Diagram of the formed paths for the environments in Figure 27a–d: (**a**) diagram of the formed path for the environment in Figure 27a; (**b**) diagram of the formed path for the environment in Figure 27b; (**c**) diagram of the formed path for the environment in Figure 27c; and (**d**) diagram of the formed path for the environment in Figure 27d.

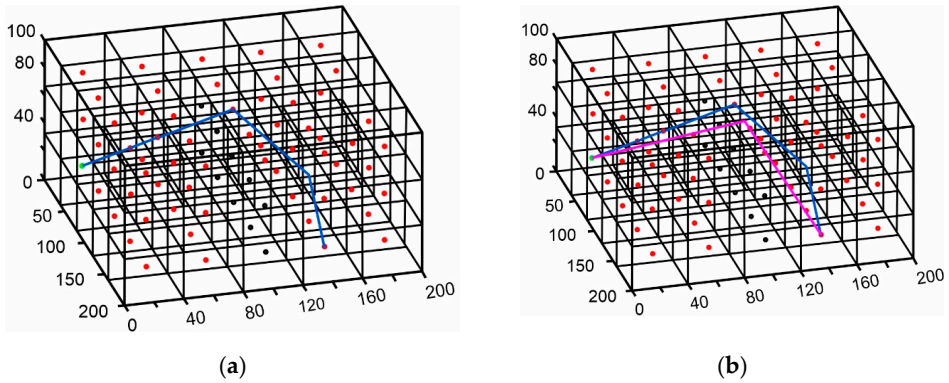


Figure 29. Results for the optimal path length reduction: (**a**) optimal 3D path using the ant colony optimization algorithm and (**b**) result for the optimized path using the proposed methodology.

Figure 31a shows the results for the proposed AFSA for the eight possible movement points and Figure 31b for the twenty-four possible movement points. Figure 31c and Figure 31d shows the results for the proposed AFSA integrated with the ray casting algorithm for the eight and twenty-four possible movement points, respectively. Table 8 summarizes the results of the path lengths in Figure 31. The length of the path formed with the 24 possible movement points is shorter by 4.72%. Also, the suggested intMaxStepsNum parameter was implemented. It is noticed that the intMaxStepsNum parameter led to a higher number of rejected paths without affecting the optimal solution. The application of the laser appeared to lead to a greater reduction in the path formed with the eight possible points of movement. This is due to the fact that the part of the path from the point where

the laser was activated to the end point formed by the method with eight possible points is the best possible path but the limitation of the finite range of motion is applied. In this case, the limitation was not so discernible in the method with 24 possible points due to the short path length, more possible directions of movement, and the larger view range. The integration of the ray casting algorithm reduces the total path length by 2.53% and 0.53% for the eight and twenty-four possible movements points, respectively. Figure 32a–d shows the diagrams with the lengths of the formed paths for the cases in Figure 31a–d, respectively.

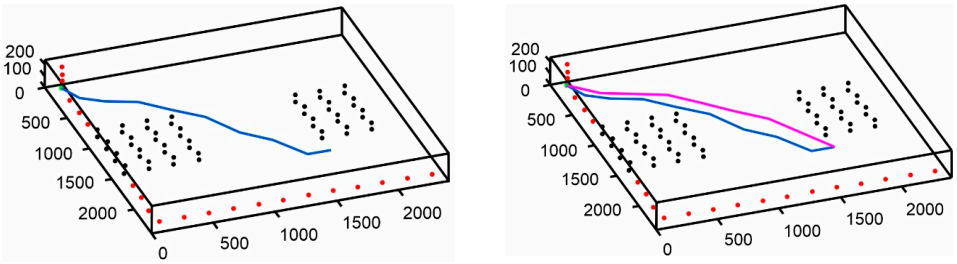


Figure 30. Results of path error corrections: (**a**) suboptimal 3D path using the ant colony optimization algorithm and (**b**) result for the optimized path using the proposed methodology.

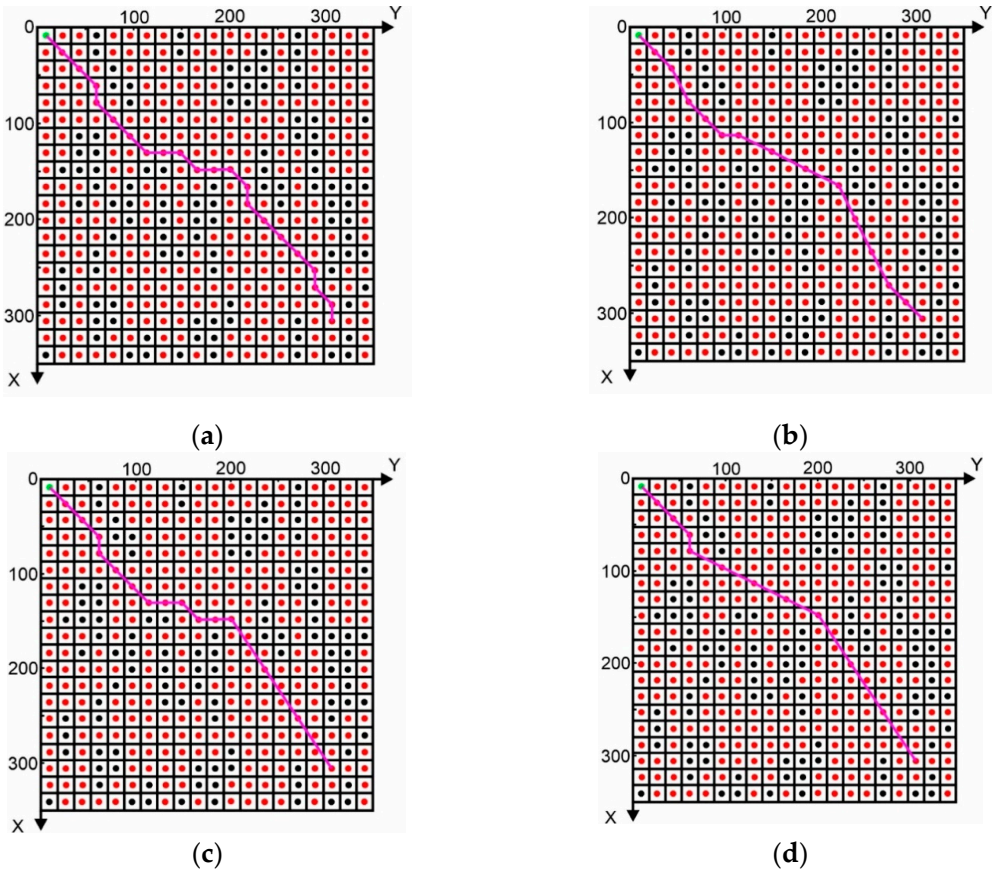


Figure 31. Path planning results: (**a**) path planning using the proposed algorithm for 8 possible movement points; (**b**) path planning using the proposed algorithm for 24 possible movement points; (**c**) path planning using the proposed algorithm integrated with the ray casting algorithm for 8 possible movement points; and (**d**) path planning using the proposed algorithm integrated with the ray casting algorithm for 24 possible movement points.

Table 8. The lengths of the paths in Figure 31.

Possible Movement Points	Path Length
8	461.7336
24	439.9108
8 + Laser	450.0326
24 + Laser	437.5624

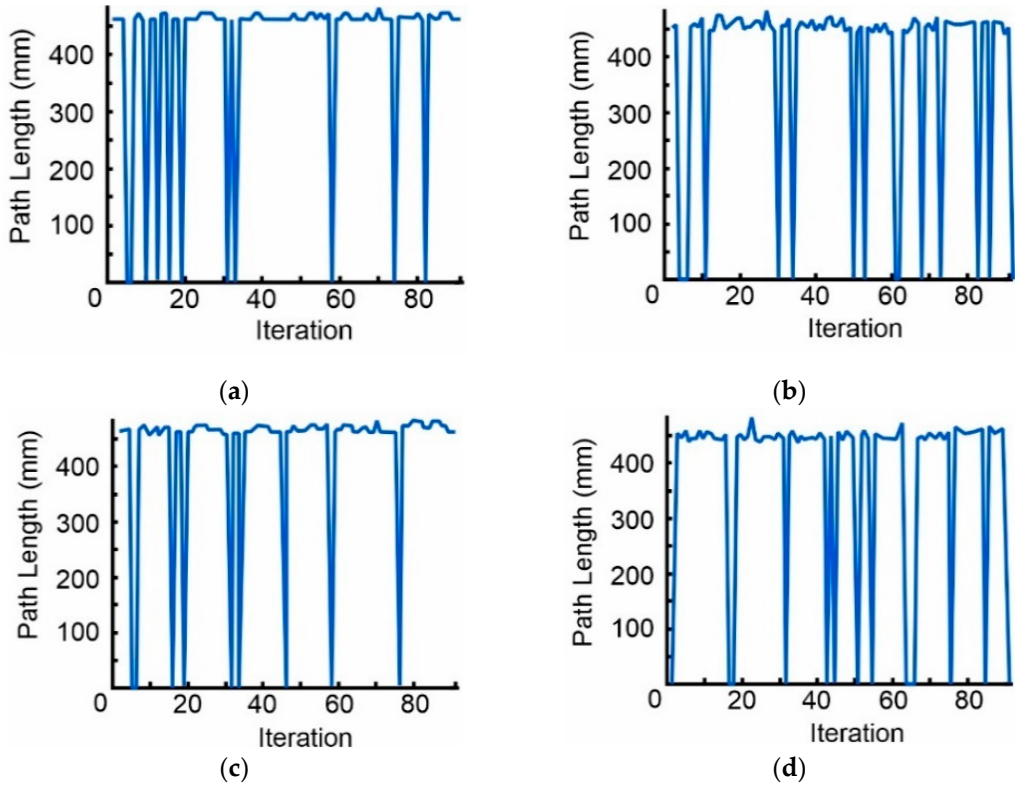


Figure 32. Diagrams of formed paths for the environments in Figure 31a–d: (**a**) diagram of the formed path for the environment in Figure 31a; (**b**) diagram of the formed path for the environment in Figure 31b; (**c**) diagram of the formed path for the environment in Figure 31c; and (**d**) diagram of the formed path for the environment in Figure 31d.

The results of path formation using the eight possible movement points in Figure 31a can be considered equivalent to the results of state-of-the-art (SOTA) methods in grid environments, such as [8,31,42]. The resulting path is the shortest possible path that can be obtained using the eight possible movement point models. However, as mentioned above, the integration of the ray casting algorithm and the use of the 24 possible movement points in the proposed algorithm can reduce further the length of the formed path obtained from the SOTA methods. A more accurate comparison between the proposed method and SOTA methods could only be made if researchers propose improvements to SOTA methods that adopt the 24 possible movement point model and the ray casting algorithm proposed in this paper.

Several comparative experiments conducted in a 20 × 20 grid environment with obstacles. The experiments were carried out to compare four algorithms: the traditional AFSA, named tr-AFSA; the traditional AFSA using the proposed behavior, pr-AFSA; the pr-AFSA combined with the proposed methodologies of simple–advanced elimination, the heatmap, and the random activation of the heatmap, impr-AFSA; and the impr-AFSA with the laser activation, laser-AFSA. In all algorithms, the model of 24 possible motion points was used with common parameter values identical to those presented at the beginning of

the section. To make the comparison easier and more accurate, the presented time measures only the execution of the algorithms and excludes common procedures, such the creation of the environment. In Figure 33a–d, the results of the formed paths for tr-AFSA, pr-AFSA, impr-AFSA, and laser-AFSA are presented, respectively. The starting point is shown in green and the end point in blue. Table 9 summarizes the results for the maximum and minimum acceptable and average lengths of the formed paths for each algorithm separately. The lengths of the paths that were not successful and were rejected are not taken into account in the minimum acceptable length. The percentage column shows the reduction in the length of the path in relation to the longest path among the algorithms, namely, from tr-AFSA. Table 10 summarizes the results for the maximum, minimum, and average number of nodes of the formed paths for each algorithm separately.

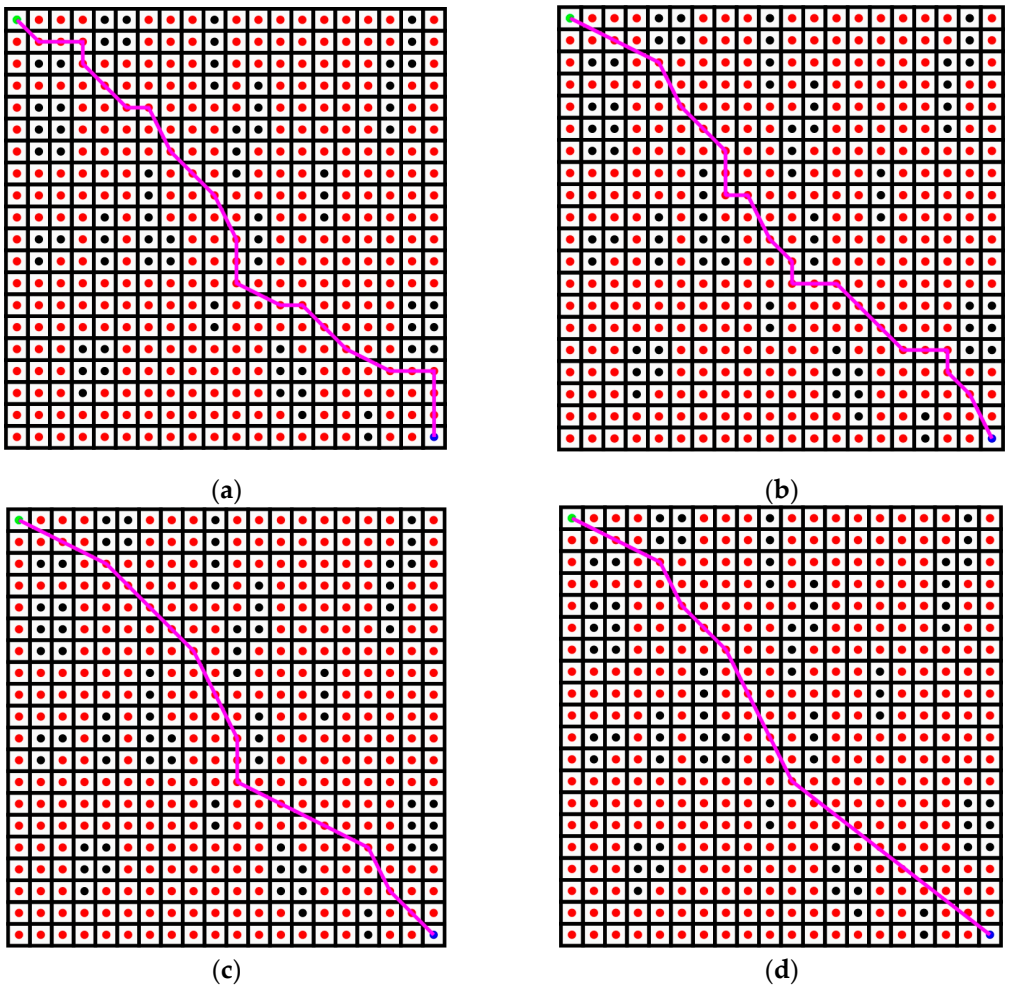


Figure 33. Path planning results: (**a**) path planning using tr-AFSA; (**b**) path planning using pr-AFSA; (**c**) path planning using impr-AFSA; and (**d**) path planning using laser-AFSA.

Table 9. The lengths of the paths in Figure 33.

Algorithm	Max. Length	Min. Length	Average	Perc. Red.
tr-AFSA	17,997	3238.6	5620.8	0%
pr-AFSA	16,657	3158.4	5526.9	2.47%
impr-AFSA	3664	2979.3	2224.4	8.00%
laser-AFSA	3559.7	2902.9	3021.8	10.36%

Table 10. The number of nodes of the paths in Figure 33.

Algorithm	Max. Node Num.	Min. Node Num.	Aver. Node Num.
tr-AFSA	128	17	33.05
pr-AFSA	112	17	33.84
impr-AFSA	16	12	14.3
laser-AFSA	15	9	10.41

Table 9 shows that that tr-AFSA and pr-AFSA form path with higher maximum and average lengths than impr-AFSA and laser-AFSA. Also, the lengths of their shortest paths are larger than the other two. This implies that due to the high number of possible movement points, the fish wander within the map, forming paths that are not optimal. This is verified by Figure 34, where it is seen that the dropout numbers of tr-AFSA and pr-AFSA are low. Comparing Figures 33a and 33b, the change in the behavior of pr-AFSA encourages the selection of the additional points that are in the model of 24 possible movement points, while tr-AFSA selects more points from the 8 possible movement point model. This fact also explains the small percentage reduction in the length of the shortest path of pr-AFSA compared to that of tr-AFSA. Both algorithms successfully form paths, but these paths are not the most optimal.

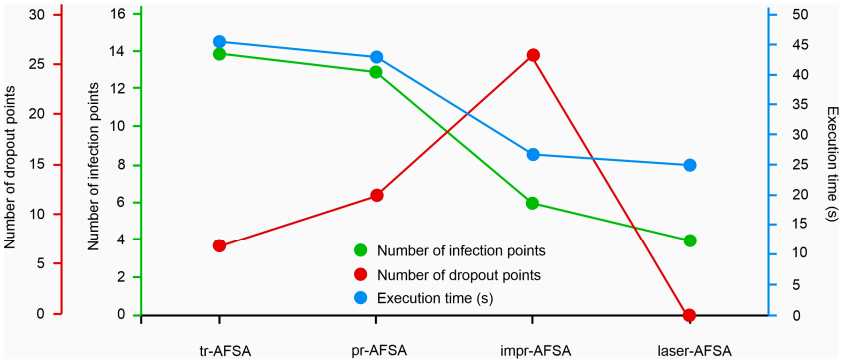


Figure 34. Diagram of the number of infection points, number of drop out points, and execution time for tr-AFSA, pr-AFSA, impr-AFSA, and laser-AFSA.

The impr-AFSA and laser-AFSA algorithms form paths with shorter lengths, while the smaller difference between the maximum and minimum values of the formed paths indicates that the strategies adopted favor the selection of the next movement point of the artificial fish. Both algorithms can find the optimal path; however, the difference in the length of the final path is due to the activation of the laser. This is also confirmed by Table 10, where a large reduction in the maximum and minimum numbers of nodes of the formed paths is observed. Figure 34 shows a significant reduction in the number of infection points. The integration of the laser seems to further reduce the infection points. An important observation from Figure 34 is the significant reduction in the execution time of the algorithm for the cases of tr-AFSA and pr-AFSA. While these algorithms are more computationally demanding, they need less time to locate the end point. This implies that the additional calculations resulting from the proposed methodologies are particularly effective for the navigation of the artificial fish. Especially for the integration of the laser, although there may be several failed attempts to activate it, it seems that they are not as time-consuming and the successful attempts can successfully cover the additional computational costs. The higher time required by tr-AFSA and pr-AFSA is attributed to the large number of possible movement points and the fish wandering through the environment until they reach the final point.

From Figure 34, it can be seen that impr-AFSA had the highest number of rejected paths, whereas laser-AFSA had no rejected routes. This is attributed to the structure of the

environment, when the fish is located at the point where the laser is activated, it can be directed in more directions that can lead to the rejection of the path, and activating the laser at that point ensures that the fish approaches the final target.

The proposed methods have been tested and applied successfully to solve real-world challenges. A Dji mavic air 2 drone and a Kawasaki RS010N industrial robot were used for the experiments. Figure 35a–d presents the process and results of applying the proposed methods for solving the 3D path planning problem for the drone's case. Figure 35a shows the outdoor complicated environment for the drone path planning process. In this scenario, the drone takes off from the green spot and has to fly over the building and land on the back side. On top of the building, there are some smaller obstacles. First, the environment that simulates the flight area is created. The buildings are considered as obstacles. In Figure 35a, the obstacles near the initial 3D path are marked with different colors. Figure 35b shows the 2D environment, and the obstacles in Figure 35a are matched with the appropriate color. To find the 2D path, the simple grid method and the proposed improved AFSA were used. Figure 35c shows the environment of 3D path planning, and the initial 3D path is plotted in red color and the final path in magenta. The orange obstacle was not defined in the 3D environment and the initial formed path passed through it. The obstacle was not defined because its dimensions were different from the space represented by each discrete point and the phenomenon of obstacle over-expansion would occur. Figure 35d shows the drone used for this case.

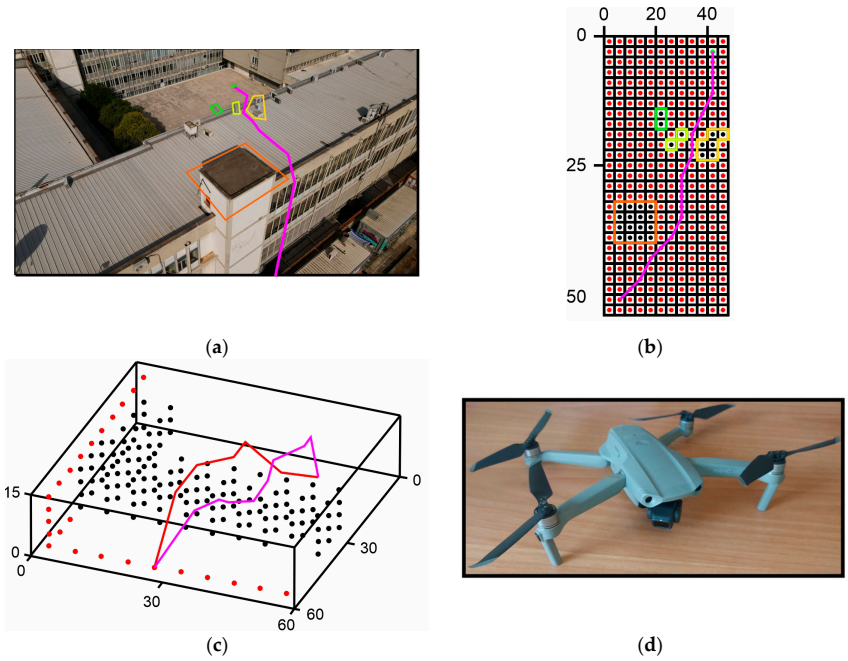


Figure 35. Application of the proposed method to drone path planning: (**a**) real-world environment of the drone case study; (**b**) 2D path planning using the proposed algorithm for 24 possible movement points in simple grid environment; (**c**) 3D environment of the formed paths; and (**d**) Dji mavic air 2 drone.

A common application of industrial robotic arms is their use to perform pick-and-place operations. Path planning to perform these tasks can be performed online, offline, or a combination of the two. The accelerations and velocities of the robot joints in addition to other factors, such as the preparation time of each execution cycle, the presence of static or dynamic obstacles, the components' rearrangement. and the path planning generation time, play important roles in the choice between online and offline path planning [43]. In this study, the proposed methodology was used for offline collision-free trajectory planning in an environment with artificial static obstacles. The simulation was performed at low and medium speeds of the robotic arm, and a special emphasis was placed on the accuracy of the

path execution. It is worth noting that during the execution of the formed path, regardless of the speed, there was no collision of the robotic arm with the obstacles, even though the formed path passed very close to the obstacles. The path was chosen to pass close to the obstacles, because in the manufacturing industry, the cost of the products depends on production time [44]. Thus, the contribution capabilities of the proposed algorithm in real conditions and whether it can be applied to solve real-world problems were examined. For the experiments, a Kawasaki RS010N industrial robotic arm was used.

Figure 36a–h highlights the results for implementing the suggested methodologies for finding the optimal 3D paths and navigating safely the end effector of the industrial robotic arm case. Figure 36a shows the experimental 3D environment for the robotic arm path planning challenge. The 3D environment includes two obstacles and a gripper is attached to robot's end effector. The purpose of this experiment is to transport a small object from a starting location to an end point. Figure 36b–f shows the different stages of path execution. Figure 36h shows the results of using the advanced grid environment and the proposed improved AFSA integrated with the ray casting algorithm. The 3D environment is not very dense and the over-expansion phenomenon of these thin obstacles is appointed. Figure 36g shows the modeled 3D environment, where the initial 3D path is plotted in red and the final path in magenta. The length of the initial 3D path is 289.7470 mm, while the length of the final path is 286.3273 mm.

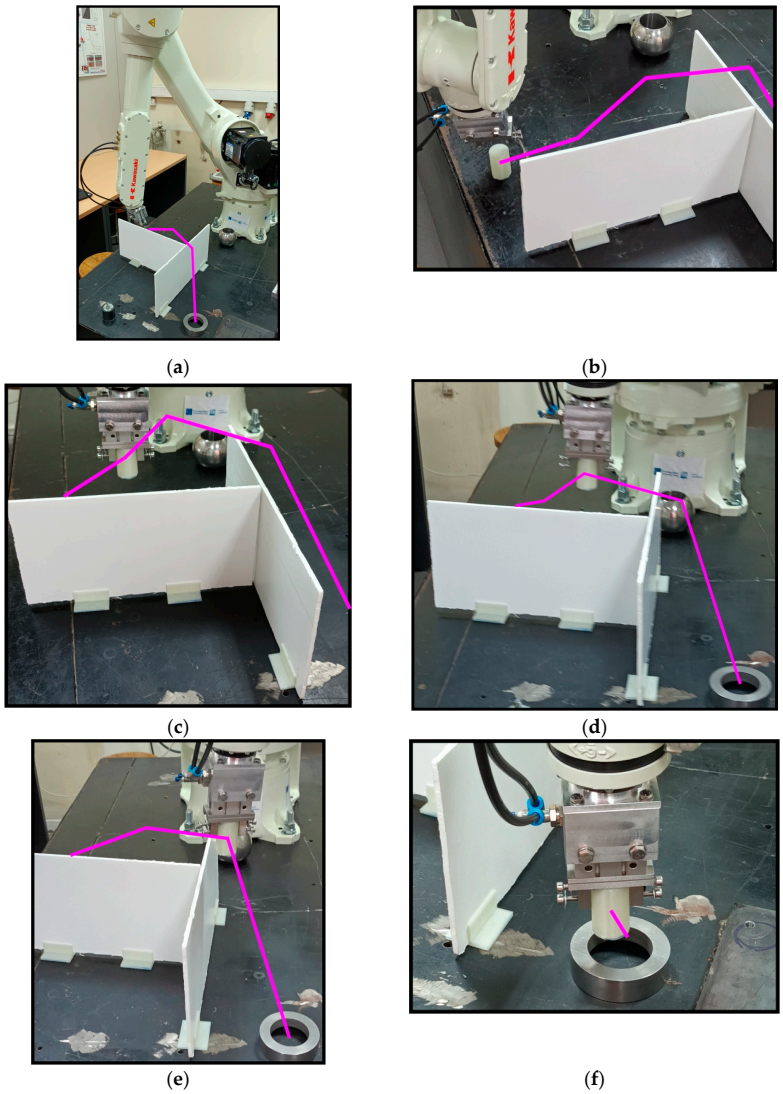
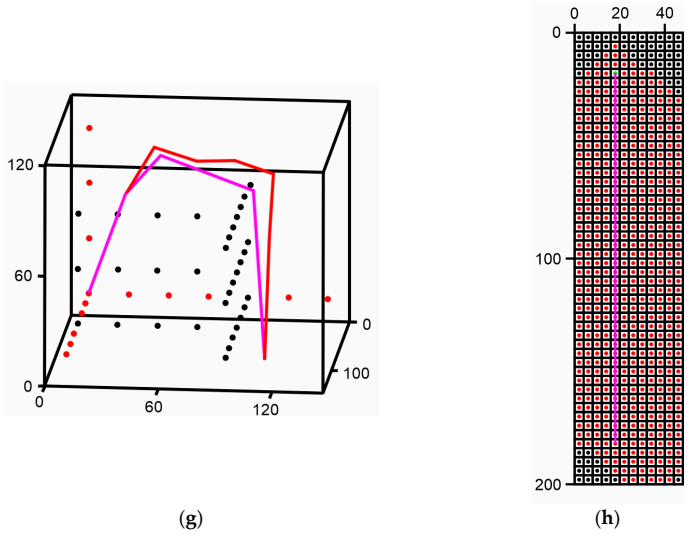


Figure 36. *Cont.*



(**g**) (**h**)

Figure 36. Application of the proposed method to an industrial robotic arm: (**a**) real-world environment with obstacles for the industrial robot case study; (**b**–**f**) intermediate points of execution of the resulting path by the industrial robotic arm; (**g**) 3D environment of the formed paths; and (**h**) 2D path planning using the improved AFSA integrated with the ray casting algorithm in the advanced grid environment.

4. Discussion

The simple and the advanced grid methods are effectively used to create a grid around the starting and end points of the path to be optimized. The simple grid method is particularly effective for exploring alternative paths in the environment around the 3D path. However, it has more discrete points than the advanced grid method and the formed path is examined in a grid like the one where the 3D path was formed. On the other hand, the advanced grid method creates a more strictly defined grid exclusively around the 3D path. This results in a denser grid environment and the collinear placement of the starting and end points within the grid. The paths explored are exclusively within the field of the optimal path and the creation of the advanced grid environment is more complex than the simple grid environment. The length of the resulting path for the two environments are identical, and if there is any deviation, is it is negligible; however, its course can differ substantially.

Replacing the random behavior in the AFSA by the proposed methodology improved the accuracy of path planning, reduced the length of the found path, while at the same time explored extensively more potential optimal paths in the map. Simple and advanced eliminations reduce the number of possible movement points so that the next discrete point is chosen among the points that contribute to finding the shortest path to the end point. The obstacles heatmap proposed in this paper facilitates the exchange of information between the environment's features and the algorithm. The obstacles heatmap reflects the navigation possibilities of each discrete point and the impacts of obstacles on the environment and finding the optimal path. Its use eradicates unnecessary oscillations, stabilizes the trajectory of the fish, and reduces the total length of the final path found. The improved algorithm using all the proposed techniques effectively finds the best possible path, avoiding collisions and unnecessary wandering in areas of the map that are excluded from the optimal path. The proposed integration of the laser beam with the AFSA reduces the length of the optimal path in cases where the end point is not located behind obstacles or its approach is not strictly defined by only a narrow path. It also empowers the fish to form paths in more directions of movement compared to the methods with twenty-four or eight possible points and helps them quickly navigate through areas of the environment where there are no intervening obstacles between the fish's position and the ending target point.

The proposed method for 3D path optimization can reduce the lengths of optimal 3D paths and correct cases of path errors. The percentage reduction in the length of the optimal routes cannot exceed the maximum possible 7.61%, as indicated in Table 1. However, apart from reducing the length, it effectively solves the problem of over-expanding the obstacles. The simulation of the real-world environment in a dense grid environment allows the accurate dimensioning of obstacles and the generation of paths with higher accuracy and matching real conditions. Regarding the suboptimal path case, the proposed method achieves a significant reduction in the final path length. The percentage of the final reduction depends on the magnitude of the deviation of the 3D path from the optimal path. The comparison between the methods with eight and twenty-four possible movement points showed that the 24 point method forms shorter paths and the fish are moving more flexibly in the environment. The laser's integration for the 8 point method underlines a greater percentage of path reduction. However, the final path's length is higher than the path length of the 24 point method. The shortest path is formed when the laser integration and the 24 point model are combined.

The research successfully decreased the lengths of optimal paths and corrected path errors on a priori known 3D paths. However, the proposed solution still has some limitations, which are described as follows:

- The results of the final 3D path depend partially on the quality of the input 3D path. In cases where very poor quality 3D paths are imported and the definition of the critical points contradicts the direction that the actual optimal path should follow, the algorithm will only improve the input path but the final results will not be the global optimal path.
- The algorithm can be applied only to a static environment with predefined obstacles. The proposed solution cannot be applied in a dynamic environment.

The ideas and proposition presented in this paper can be extended further and effectively contribute to the improvement of the existing literature. Future research suggestions include the following:

- The extension of the proposed improved artificial fish swarm algorithm to 3D environments with obstacles;
- The adaptation and study of the effectiveness of the 24 possible movement point model in other path planning algorithms like the particle swarm algorithm, ant colony optimization, genetic algorithms etc.;
- The integration of the presented ray casting algorithm to other path planning algorithms and a comparison of resulting paths;
- The extension of the heatmap method and its applications to other agent-based algorithms;
- The replacement of random heatmap activation by a different artificial intelligence approach that recognizes the cases where the agent needs the help of the map to navigate.

Author Contributions: Conceptualization, I.C., G.M. and A.T.; methodology, I.C., G.M. and A.T.; software, I.C. and G.M.; validation, G.M. and A.T.; formal analysis, G.M. and A.T.; investigation, I.C., G.M. and A.T.; data curation, G.M. and A.T.; writing—original draft preparation, I.C., G.M. and A.T.; writing—review and editing, G.M. and A.T.; visualization, I.C.; supervision, G.M. and A.T.; project administration, G.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data will be available upon request from the corresponding authors.

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

References

1. Sánchez-Ibáñez, J.R.; Pérez-del-Pulgar, C.J.; García-Cerezo, A. Path Planning for Autonomous Mobile Robots: A Review. *Sensors* **2021**, *21*, 7898. [CrossRef]

2. Matisková, D.; Čakurda, T.; Marasová, D.; Balara, A. Determination of the Function of the Course of the Static Property of PAMs as Actuators in Industrial Robotics. *Appl. Sci.* **2021**, *11*, 7288. [CrossRef]
3. Khadidos, A.O.; Al-Darraji, I.; Khadidos, A.O.; Tsaramirsis, G. A New Tilted Aerial Robotic Platform: Modeling and Control. *Alex. Eng. J.* **2023**, *84*, 126–137. [CrossRef]
4. Wang, D.; Ma, E.; Leung, X.Y. Working with Robots: A Job Design Perspective of Hospitality Employees' Collaboration Intentions with Service Robots. *J. Hosp. Tour. Manag.* **2024**, *61*, 66–77. [CrossRef]
5. Mahmood, T.; Jaleel, A.; ur Rehman, U. Determination of the Most Influential Robot in the Medical Field by Utilizing the Bipolar Complex Fuzzy Soft Aggregation Operators. *Expert Syst. Appl.* **2024**, *251*, 123878. [CrossRef]
6. Aggarwal, S.; Kumar, N. Path Planning Techniques for Unmanned Aerial Vehicles: A Review, Solutions, and Challenges. *Comput. Commun.* **2020**, *149*, 270–299. [CrossRef]
7. Atyabi, A.; Ward Powers, D.M. Review of Classical and Heuristic-Based Navigation and Path Planning Approaches. *Int. J. Adv. Comput. Technol.* **2013**, *16*, 1–15.
8. Huo, F.; Zhu, S.; Dong, H.; Ren, W. A New Approach to Smooth Path Planning of Ackerman Mobile Robot Based on Improved ACO Algorithm and B-Spline Curve. *Rob. Auton. Syst.* **2024**, *175*, 104655. [CrossRef]
9. Bakdi, A.; Hentout, A.; Boutami, H.; Maoudj, A.; Hachour, O.; Bouzouia, B. Optimal Path Planning and Execution for Mobile Robots Using Genetic Algorithm and Adaptive Fuzzy-Logic Control. *Rob. Auton. Syst.* **2017**, *89*, 95–109. [CrossRef]
10. Xu, D.; Yang, J.; Zhou, X.; Xu, H. Hybrid Path Planning Method for USV Using Bidirectional A* and Improved DWA Considering the Manoeuvrability and COLREGs. *Ocean Eng.* **2024**, *298*, 117210. [CrossRef]
11. Huang, C.; Zhou, X.; Ran, X.; Wang, J.; Chen, H.; Deng, W. Adaptive Cylinder Vector Particle Swarm Optimization with Differential Evolution for UAV Path Planning. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105942. [CrossRef]
12. Das, M.S.; Sanyal, S.; Mandal, S. Navigation of Multiple Robots in Formative Manner in an Unknown Environment Using Artificial Potential Field Based Path Planning Algorithm. *Ain Shams Eng. J.* **2022**, *13*, 101675. [CrossRef]
13. Hao, L.; Liu, D.; Du, S.; Wang, Y.; Wu, B.; Wang, Q.; Zhang, N. An Improved Path Planning Algorithm Based on Artificial Potential Field and Primal-Dual Neural Network for Surgical Robot. *Comput. Methods Programs Biomed.* **2022**, *227*, 107202. [CrossRef] [PubMed]
14. Tu, H.; Deng, Y.; Li, Q.; Song, M.; Zheng, X. Improved RRT Global Path Planning Algorithm Based on Bridge Test. *Rob. Auton. Syst.* **2024**, *171*, 104570. [CrossRef]
15. Zhang, X.; Chen, J.; Xin, B.; Peng, Z. A Memetic Algorithm for Path Planning of Curvature-Constrained UAVs Performing Surveillance of Multiple Ground Targets. *Chin. J. Aeronaut.* **2014**, *27*, 622–633. [CrossRef]
16. Hasircioglu, I.; Topcuoglu, H.R.; Ermis, M. 3-D Path Planning for the Navigation of Unmanned Aerial Vehicles by Using Evolutionary Algorithms. In Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, Atlanta, GA, USA, 12–16 July 2008; pp. 1499–1506. [CrossRef]
17. Shan, D.; Zhang, S.; Wang, X.; Zhang, P. Path-Planning Strategy: Adaptive Ant Colony Optimization Combined with an Enhanced Dynamic Window Approach. *Electronics* **2024**, *13*, 825. [CrossRef]
18. Sarkar, R.; Barman, D.; Chowdhury, N. Domain Knowledge Based Genetic Algorithms for Mobile Robot Path Planning Having Single and Multiple Targets. *J. King Saud. Univ. Comput. Inf. Sci.* **2022**, *34*, 4269–4283. [CrossRef]
19. Qin, H.; Shao, S.; Wang, T.; Yu, X.; Jiang, Y.; Cao, Z. Review of Autonomous Path Planning Algorithms for Mobile Robots. *Drones* **2023**, *7*, 211. [CrossRef]
20. Zheng, L.; Yu, W.; Li, G.; Qin, G.; Luo, Y. Particle Swarm Algorithm Path-Planning Method for Mobile Robots Based on Artificial Potential Fields. *Sensors* **2023**, *23*, 6082. [CrossRef]
21. Wu, Z.; Dai, J.; Jiang, B.; Karimi, H.R. Robot Path Planning Based on Artificial Potential Field with Deterministic Annealing. *ISA Trans.* **2023**, *138*, 74–87. [CrossRef] [PubMed]
22. Bian, T.; Xing, Y.; Zolotas, A. End-to-End One-Shot Path-Planning Algorithm for an Autonomous Vehicle Based on a Convolutional Neural Network Considering Traversability Cost. *Sensors* **2022**, *22*, 9682. [CrossRef]
23. Gul, F.; Rahiman, W.; Nazli Alhady, S.S. A Comprehensive Study for Robot Navigation Techniques. *Cogent Eng.* **2019**, *6*, 1632046. [CrossRef]
24. Jaiswal, S.; Soumya, S. Low-Cost Path Planning in 2D Environment Using A∗ Algorithm by Considering Slope of the Obstacle. *IFAC-Pap.* **2022**, *55*, 783–788. [CrossRef]
25. Lamini, C.; Benhlima, S.; Elbekri, A. Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning. *Procedia Comput. Sci.* **2018**, *127*, 180–189. [CrossRef]
26. Li, Y.; Wei, W.; Gao, Y.; Wang, D.; Fan, Z. PQ-RRT*: An Improved Path Planning Algorithm for Mobile Robots. *Expert Syst. Appl.* **2020**, *152*, 113425. [CrossRef]
27. Gul, O.M. Energy-Aware 3D Path Planning by Autonomous Ground Vehicle in Wireless Sensor Networks. *World Electr. Veh. J.* **2024**, *15*, 383. [CrossRef]
28. Sun, B.; Niu, N. Multi-AUVs Cooperative Path Planning in 3D Underwater Terrain and Vortex Environments Based on Improved Multi-Objective Particle Swarm Optimization Algorithm. *Ocean Eng.* **2024**, *311*, 118944. [CrossRef]
29. Bolourian, N.; Hammad, A. LiDAR-Equipped UAV Path Planning Considering Potential Locations of Defects for Bridge Inspection. *Autom. Constr.* **2020**, *117*, 103250. [CrossRef]

30. Liu, Y.; Zhang, X.; Guan, X.; Delahaye, D. Adaptive Sensitivity Decision Based Path Planning Algorithm for Unmanned Aerial Vehicle with Improved Particle Swarm Optimization. *Aerosp. Sci. Technol.* **2016**, *58*, 92–102. [CrossRef]
31. Fang, W.; Liao, Z.; Bai, Y. Improved ACO Algorithm Fused with Improved Q-Learning Algorithm for Bessel Curve Global Path Planning of Search and Rescue Robots. *Rob. Auton. Syst.* **2024**, *182*, 104822. [CrossRef]
32. Xu, X.; Zeng, J.; Zhao, Y.; Lü, X. Research on Global Path Planning Algorithm for Mobile Robots Based on Improved A*. *Expert Syst. Appl.* **2024**, *243*, 122922. [CrossRef]
33. Ayawli, B.B.K.; Appiah, A.Y.; Nti, I.K.; Kyeremeh, F.; Ayawli, E.I. Path Planning for Mobile Robots Using Morphological Dilation Voronoi Diagram Roadmap Algorithm. *Sci. Afr.* **2021**, *12*, e00745. [CrossRef]
34. Wu, G.; Atilla, I.; Tahsin, T.; Terziev, M.; Wang, L.C. Long-Voyage Route Planning Method Based on Multi-Scale Visibility Graph for Autonomous Ships. *Ocean Eng.* **2021**, *219*, 108242. [CrossRef]
35. Promkaew, N.; Thammawiset, S.; Srisan, P.; Sanitchon, P.; Tummawai, T.; Sukpancharoen, S. Development of Metaheuristic Algorithms for Efficient Path Planning of Autonomous Mobile Robots in Indoor Environments. *Results Eng.* **2024**, *22*, 102280. [CrossRef]
36. Duchon, F.; Babinec, A.; Kajan, M.; Beno, P.; Florek, M.; Fico, T.; Jurišica, L. Path Planning with Modified a Star Algorithm for a Mobile Robot. *Procedia Eng.* **2014**, *96*, 59–69. [CrossRef]
37. Sun, C.; van der Tol, R.; Melenhorst, R.; Pacheco, L.A.P.; Koerkamp, P.G. Path Planning of Manure-Robot Cleaners Using Grid-Based Reinforcement Learning. *Comput. Electron. Agric.* **2024**, *226*, 109456. [CrossRef]
38. Belmonte, L.M.; Morales, R.; Fernández-Caballero, A. Computer Vision in Autonomous Unmanned Aerial Vehicles—A Systematic Mapping Study. *Appl. Sci.* **2019**, *9*, 3196. [CrossRef]
39. Mannar, S.; Thummalapeta, M.; Saksena, S.K.; Omkar, S.N. Vision-Based Control for Aerial Obstacle Avoidance in Forest Environments. *IFAC-PapersOnLine* **2018**, *51*, 480–485. [CrossRef]
40. Mansour, G.; Chouridis, I.; Tsagaris, A. Finding the Optimal Path in a 3D Environment with Predefined Obstacles. *Int. J. Adv. Mechatron. Syst.* **2024**, *11*, 50–62. [CrossRef]
41. Neshat, M.; Sepidnam, G.; Sargolzaei, M.; Toosi, A.N. Artificial Fish Swarm Algorithm: A Survey of the State-of-the-Art, Hybridization, Combinatorial and Indicative Applications. *Artif. Intell. Rev.* **2014**, *42*, 965–997. [CrossRef]
42. Zhao, J.; Deng, C.; Yu, H.; Fei, H.; Li, D. Path Planning of Unmanned Vehicles Based on Adaptive Particle Swarm Optimization Algorithm. *Comput. Commun.* **2024**, *216*, 112–129. [CrossRef]
43. Tipary, B.; Kovács, A.; Erdős, F.G. Planning and Optimization of Robotic Pick-and-Place Operations in Highly Constrained Industrial Environments. *Assem. Autom.* **2021**, *41*, 626–639. [CrossRef]
44. Elassal, A.; Abdelaal, M.; Osama, M.; Elhnidy, H. Low-Cost Parallel Delta Robot for a Pick-and-Place Application with the Support of the Vision System. e-Prime—Advances in Electrical Engineering. *Electron. Energy* **2024**, *8*, 100518. [CrossRef]

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.