agriculture

Article

Development of an Improved GWO Algorithm for Solving Optimal Paths in Complex Vertical Farms with Multi-Robot Multi-Tasking

Jiazheng Shen, Tang Sai Hong *, Luxin Fan, Ruixin Zhao, Mohd Khairol Anuar b. Mohd Ariffin and Azizan bin As'arry

Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang 43400, Selangor, Malaysia; gs63073@student.upm.edu.my (J.S.); gs59924@student.upm.edu.my (L.F.); gs63714@student.upm.edu.my (R.Z.); khairol@upm.edu.my (M.K.A.b.M.A.); zizan@upm.edu.my (A.b.A.)
* Correspondence: saihong@upm.edu.my

Abstract: As the global population grows, achieving Zero Hunger by 2030 presents a significant challenge. Vertical farming technology offers a potential solution, making the path planning of agricultural robots in vertical farms a research priority. This study introduces the Vertical Farming System Multi-Robot Trajectory Planning (VFSMRTP) model. To optimize this model, we propose the Elitist Preservation Differential Evolution Grey Wolf Optimizer (EPDE-GWO), an enhanced version of the Grey Wolf Optimizer (GWO) incorporating elite preservation and differential evolution. The EPDE-GWO algorithm is compared with Genetic Algorithm (GA), Simulated Annealing (SA), Dung Beetle Optimizer (DBO), and Particle Swarm Optimization (PSO). The experimental results demonstrate that EPDE-GWO reduces path length by 24.6%, prevents premature convergence, and exhibits strong global search capabilities. Thanks to the DE and EP strategies, the EPDE-GWO requires fewer iterations to reach the optimal solution, offers strong stability and robustness, and consistently finds the optimal solution at a high frequency. These attributes are particularly significant in the context of vertical farming, where optimizing robotic path planning is essential for maximizing operational efficiency, reducing energy consumption, and improving the scalability of farming operations.

Keywords: vertical farming system; grid map; agriculture robots; grey wolf optimizer

Citation: Shen, J.; Hong, T.S.; Fan, L.; Zhao, R.; Mohd Ariffin, M.K.A.b.; As'arry, A.b. Development of an Improved GWO Algorithm for Solving Optimal Paths in Complex Vertical Farms with Multi-Robot Multi-Tasking. *Agriculture* **2024**, *14*, 1372. https://doi.org/10.3390/agriculture14081372

Academic Editor: Jiehao Li

Received: 14 July 2024
Revised: 13 August 2024
Accepted: 14 August 2024
Published: 15 August 2024

1. Introduction

At the United Nations General Assembly in 2015, 17 Sustainable Development Goals (SDGs) (Figure 1) were proposed and adopted by all member states. SDG 2, entitled "Zero Hunger", was thus established [1]. However, reports and results to date indicate that achieving SDG 2 by 2030 will remain a challenge. This is due to the fact that the world's population is growing, while the amount of available arable land is simultaneously decreasing [2,3]. In order to solve the food problem, vertical farming has been proposed as a potential solution.

Vertical farming (Figure 2) has been proposed as one of the solutions to the problem of farming. Vertical farms improve land utilisation and space utilisation by growing crops in multi-storey buildings or vertical structures. This method reduces transport costs and carbon emissions while providing produce to city dwellers. Vertical farms are very space-efficient, featuring multiple levels of cultivation racks, each used to grow specific crops. Vertical farms are a form of precision agriculture where all factors affecting crop growth can be precisely controlled. Real-time monitoring of crop growth and conditions can feed into a crop growth model, which predicts yield, time to maturity, and other crop attributes. A large number of agriculture robots (agri-robots) are used in this process, some of which are fully automated [4,5]. The extensive automation and deployment of numerous agri-robots have led to a notable increase in the energy consumption of the entire vertical farming system. Therefore, rational path planning for a large number of agri-robots in the complex environment of vertical farms is one of the most important ways to reduce energy consumption [6,7].

Figure 1. Sustainable Development Goals [1].

Figure 2. VFS cultivation rack [4].

Path planning is one of the most important techniques for multiple agri-robots to work autonomously in vertical farms and an important challenge in achieving unmanned vertical farms [8]. Vertical farms are characterised by vertical operations, limited space, and dense crops, and thus require high path planning capabilities from agri-robots [9]. The path planning problem for agri-robots can be formulated as an optimisation problem, which aims to find the shortest feasible path from the starting point to the task point for an agri-robot under different optimisation requirements and different environmental and task constraints [10,11]. These requirements may be shortest travelling distance, shortest task time, etc. [12]. These constraints may include agri-robot state constraints, task constraints, and vertical farm environment constraints, etc. [13]. Considering the key role of path optimisation in improving the autonomy and economy of robotic systems, its research has received increasing attention.

Currently, researchers around the world have conducted extensive studies on path optimization for mobile robots, utilizing various algorithms and strategies to enhance performance. For example, Ouyang et al. [14] proposed adding an adaptive weighting strategy and a Lévy flight strategy into the sparrow algorithm to solve the path optimisation problem for mobile robots, which improved the search accuracy of the algorithm and reduced the drawbacks of falling into local optimums and greater randomness. Pehlivanoglu [15] proposed a new initial population enhancement method that accelerated the convergence speed of the algorithm and reduced the number of required objective function evaluations by 70%, improving the autonomous path planning capability of UAVs. Li et al. [16], in the navigation problem of mobile robots, proposed the Multi-Strategy and Improved DBO (MSIDBO) algorithm, which effectively avoids falling into a local optimum due to premature convergence. Yuan et al. [17] proposed an improved PSO algorithm based on differential evolution in the mobile robot path planning problem, in which "high-intensity training" for the global optimum position improves the accuracy of the arithmetic search method and obtains a better mobile path.

The main contributions of this study are as follows: first, a multi-tasking path planning model for multi-agri-robots is proposed based on the environmental constraints of vertical farms. Second, an improved GWO algorithm based on Elite Preservation (EP) and Differential Evolution (DE) is proposed. Third, the proposed EPDE-GWO algorithm is successfully applied to the Vertical Farming System Multi-Robot Trajectory Planning (VFSMRTP) model, which is improved in terms of global optimality-seeking ability, average path length, robustness and stability compared to other comparable algorithms.

2. Materials and Methods

The overall workflow diagram of this study is shown in Figure 2. Firstly, the large-scale vertical cultivation racks of a vertical farm were preprocessed, the farm environment was rasterised based on the camera's monitoring data, and the VFSMRTP model was assembled using the objective function and the constraint function (Section 2.1). Next, the basic GWO algorithm was improved by introducing a population diversification strategy based on the DE idea and a population renewal strategy based on the elite retention strategy (Section 2.2). Finally, the proposed EPDE-GWO algorithm was used to solve the VFSMRTP model and compared with other comparable algorithms for performance evaluation (Section 2.3).

2.1. Building the VFSMRTP Mode

2.1.1. Vertical Farms Environmental Gridding

The key to performing agri-robot scheduling, and resolving trajectory conflicts between multiple agricultural robots, is to accurately model the Vertical Farming System (VFS) environment [18]. The real scene is accurately represented in the model by an abstract digital map. Obstacles, paths, cultivation racks, loading piles, start points, task points, and so on are included in the model. The environment model must be highly accurate and represent the environment information in detail. The environmental model must also have good extensibility and transferability to adapt to different research and application needs, and must be able to support the execution of the trajectory planning algorithm. Therefore, we gridded the data from the camera (Figure 2) and used different colours to indicate different meanings in the visualised raster map (Figure 3). White represents passable nodes, red represents the start position, black indicates obstacles, green signifies agricultural task points, yellow represents the moving agricultural task storage area, and purple denotes the charging position.

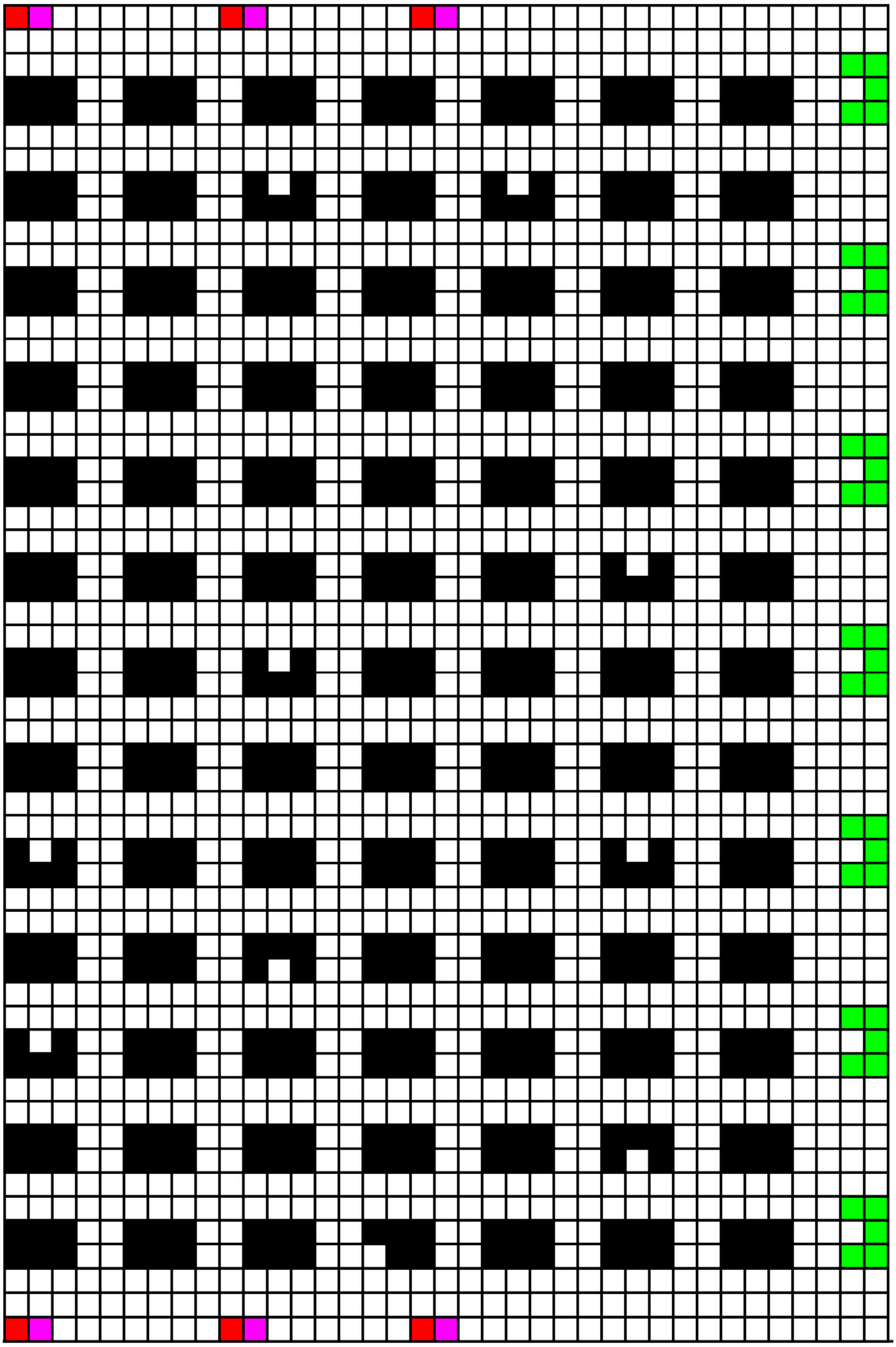

Figure 3. VFS grid map.

2.1.2. Problem Description and Analysis

The optimisation problem of agri-robot trajectory planning in vertical farms can be described as follows: The VFS consists of agri-robot start points, charging points, task points and obstacle zones. Pending tasks are generated in the VFS at each location. Tasks are executed by agri-robots and then disappear, transforming into obstacle zones. The system can only perform task assignment and optimal trajectory planning for tasks that have already appeared. Tasks have different levels of urgency, which is reflected by different priorities (weights). A quantity of agri-robots and a + n(t) quantity of tasks exists in the VFS. Each agri-robot can only perform one task at a time, and each task can only be performed by one agri-robot. Agri-robots consume electrical energy at a fixed rate. The sum of the electrical energy required for the agri-robot's task path and the electrical energy required for the agri-robot to move to the nearest charging point should be greater than the remaining electrical energy of the agri-robot. The grid g where any task $i \in I$ is located is known in advance. Agri-robot trajectory planning requires determining the travel path for each agri-robot to perform each task, including the sequence of grids travelled and the dwell time at each grid. The decision objectives are: Complete all tasks with the lowest total energy consumption of agri-robots.

The following reasonable assumptions are made:

1. All agri-robots are available at time zero.
2. Agri-robots travel at a constant speed.
3. Agri-robots can travel horizontally or vertically and are not allowed to travel in diagonal directions.
4. The time taken for an idle agri-robot to reach a constant speed, and for the moving agri-robot to reach idle state, are extremely short, and can be considered negligible.
5. The case of equipment failure is not taken into account.
6. The time required for the execution of all tasks is uniform.

2.1.3. Building the VFSMRTP Model

The objective function (1) is the shortest path, and the objective function in this paper is similar to Candra et al. [19]. Optimizing for the shortest (min) path helps in reducing the distance travelled (D) by the robots, which in turn reduces energy consumption and operational costs. Additionally, shorter paths lead to more predictable and coordinated movements of the robots, reducing the risk of collisions and traffic congestion. This objective also contributes to reducing the wear and tear on the robots, enhancing their longevity and reliability.

$$F = min\ D \tag{1}$$

The constraint function in this paper is similar to Tian, Varga and Lee [20–22]. Equations (2)–(5) represent the relationship between each agri-robot's task sequence, travel path, travel time window, loading task and its subset, respectively. Equations (6)–(8) represent the equations for the total travel time of the agri-robot and the total time to complete the task. Equation (9) then represents the fact that each agri-robot can only pass through one node at a time. Equations (10) and (11) represent the process of path planning to ensure that agri-robots do not appear at the same node at the same time, thus reducing the generation of conflicts at the start of path planning. Equation (12) shows that only one agri-robot is allowed to recharge at each charging station at any given time. Equation (13) represents the decision process of the agri-robot charging according to the distance to select the corresponding charging station, and calculates the arrival time after determining the charging station. Equation (14) represents the time sequence constraint when two agri-robots travel to the same charging station. Equations (15) and (16) state that if the agri-robot has a lower priority, the time used by the agri-robot to pass the path section k is the sum of the time used by the agri-robot travelling at normal speed and the waiting time required by the agri-robot to avoid conflicts. Equation (17) represents the capacity constraint in each path segment.

$$U_a = \sum_{a \in A} u_{am}, m \in m_a \tag{2}$$

$$V_a = \sum_{a \in A} v_{am}, m \in m_a \tag{3}$$

$$W_a = \sum_{a \in A} w_{am}, m \in m_a \tag{4}$$

$$R_a = \sum_{a \in A} r_{am}, m \in m_a \tag{5}$$

$$T_a = \sum_{j \in V_a} \sum_{k \in V_a} \sum_{a \in A} t^a_{jk} x_{ajkd} + \sum_{m \in R} \sum_{a \in A} \sum_{n \in N} \left(T^n_{am} + T^n_{am'}\right) \tag{6}$$

$$TE_a = T_a + \sum_{m \in U_a} \sum_{a \in A}^{s} w^a_m \left(T^a_{e,m} - T^a_{s,m}\right), a \in A \tag{7}$$

$$T = maxTE_a, a \in A \tag{8}$$

$$\sum_{j \in V} \sum_{k \in V} x_{ajjk} = 1, \forall a \in A \tag{9}$$

$$\sum_{a \in A} \sum_{k \in V} x_{ajkt} = 1, \forall j \in V \tag{10}$$

$$\sum_{a \in A} \sum_{j \in V} x_{ajkt} = 1, \forall k \in V \tag{11}$$

$$\sum_{a \in A} y^n_{at} = 1, \forall n \in N \tag{12}$$

$$T^n_{am} = max(w_{am}) + \frac{min(d_{e_m n})}{v}, \forall a \in A, \forall m \in R, n \in N \tag{13}$$

$$T^a_{e,m} \leq T^{a'}_{s,m'}, T^n_{am} \leq T^n_{a'm'}, \forall a, a' \in A, \forall m, m' \in R, n \in N \tag{14}$$

$$P_a \leq P_b, \forall a, b \in A \tag{15}$$

$$t^a_{jk} = t^{wa}_{jk} + t^{na}_{jk}, \forall a \in A, \forall j, k \in V \tag{16}$$

$$C_{jkt} * x_{ajkt} \leq C_{jk} * x_{ajkt}, \forall j, k \in V, a \in A \tag{17}$$

2.2. Proposed EPDE-GWO Algorithm

2.2.1. Basic GWO Algorithm

The GWO algorithm is a group intelligence optimisation algorithm proposed by Mirjalili et al. [23] based on the observed behaviour of grey wolves engaged in hunting and the formation of social hierarchies. The grey wolf pack exhibits a strict pyramidal hierarchy. The α-wolves in Tier 1 are the alpha wolves of the grey wolf pack. As such, they bear responsibility for leading the pack and making decisions on its behalf. The β wolves in tier 2 are subordinate to the alpha wolf and are tasked with assisting the alpha wolves in decision-making and pack management. The δ Tier 3 wolves are the executors and are responsible for carrying out the decisions made by the α and β wolves. The ω pack in level 4 is responsible for the monitoring and maintenance of the relationships within the pack.

In the fundamental GWO algorithm, the first-, second- and third-best solutions, designated as α-, β- and δ-wolf, respectively, represent the optimal, suboptimal and least optimal outcomes for the problem at hand. Furthermore, the ω-wolf guides the search for the optimal solution, after which the status of the wolf pack is updated [24,25]. The fundamental principle of the GWO algorithm is to emulate the behaviour of a grey wolf pack hunting its prey. This is achieved by directing the entire pack towards an optimal solution through the actions of the α-, β-, and δ-wolves. Additionally, the algorithm employs a shrinking enclosure and random search strategy to facilitate global optimisation. The algorithm employs the social hierarchical structure and collaborative mechanism of the wolf pack to effectively search for the global optimal solution within the solution space [23,26]. The evolutionary equations for the fundamental GWO algorithm were initially proposed by Mirjalili et al. [23] and shown from (18) to (22):

$$D = \left| C \cdot X_p(t) - X(t) \right| \tag{18}$$

$$X(t+1) = X_p(t) - A \cdot D \tag{19}$$

$$C = 2 \cdot r_1 \tag{20}$$

$$A = 2a \cdot r_2 - a \tag{21}$$

$$a = 2 \cdot (1 - \frac{t}{T}) \tag{22}$$

In these equations, D represents the distance between an individual grey wolf and its target, while C serves as the coefficient for adjusting this distance. $X_p(t)$ with $p = 1, 2, 3$, denotes the position of the target at iteration t. The position of a specific grey wolf is given by $X(t)$, and $X(t+1)$ indicates the updated position for the subsequent iteration, based on the positions from the previous generation. The coefficient A is used for adjusting the position, and r_1 and r_1 are uniformly distributed random variables within the interval $[0, 1]$. The parameter a functions as the convergence factor, where t indicates the current iteration number and T signifies the maximum number of iterations.

The position update equation has been proposed by Mirjalili et al. [23]. Equations (18)–(22) show the target position of the hunting target relative to α wolf, β wolf, and δ wolf. Equations (23)–(25) show $X_1(t)$, $X_2(t)$, and $X_3(t)$, respectively. Equation (26) shows the process of guiding the wolves to approach the hunting target.

$$X_1(t) = X_\alpha(t) - A \cdot D_\alpha \tag{23}$$

$$X_2(t) = X_\beta(t) - A \cdot D_\beta \tag{24}$$

$$X_3(t) = X_\delta(t) - A \cdot D_\delta \tag{25}$$

$$X(t+1) = \frac{X_1(t) + X_2(t) + X_3(t)}{3} \tag{26}$$

where $X_\alpha(t)$, $X_\beta(t)$, and $X_\delta(t)$ represent the positions of the α wolf, β wolf, and δ wolf, respectively, at time t. D_α, D_β, and D_δ are the distances of the α wolf, β wolf, and δ wolf from the target location, respectively.

The position of an individual grey wolf corresponds to the value of the fitness function for this solution. Therefore, the fitness function L for trajectory planning is defined as follows:

$$L = \sum_{k=1}^{dim} \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \tag{27}$$

In Equation (27), dim represents the problem dimension, corresponding to the total number of nodes along the path. The variable k indicates the number of nodes that the path traverses. x_k and y_k denote the coordinates of the k-th path node in the X and Y directions, respectively. The fitness function L calculates the path length by summing the distances between consecutive nodes along the path [27,28].

2.2.2. Population Diversification Based on DE Strategies

The environment of vertical farms is characterized by complexity and diversity. To prevent the GWO algorithm from falling into premature convergence and obtaining locally optimal solutions, it is essential to maintain population diversity during the re-optimization process. The path optimization problem also requires the algorithm to have adaptability to satisfy multiple constraints. Therefore, introducing the population diversity strategy of DE is appropriate for improving the drawbacks of GWO. This approach allows the GWO algorithm to maintain population diversity in complex environments, enhances its global search capability, prevents premature convergence, and provides flexibility and adaptability to different optimization scenarios [29].

Additionally, the DE strategy has several advantages: it is simple to implement, easy to operate, and can provide high-quality solutions in a short time [30]. In the field of trajectory planning, many researchers have used DE to improve various algorithms. For example, Lim et al. [31] proposed selective DE-hybridized Particle Swarm Optimization with adaptive factor (DEAPSO) and selective DE-hybridized Quantum-behaved PSO (SDEQPSO) for efficient trajectory planning of AUVs. Mousa & Hussein [32] proposed a fusion algorithm combining the Ant Colony Optimization (ACO) and DE to optimize Unmanned Aerial Vehicle (UAV) paths.

Rainer & Kenneth [33] first proposed DE. DE is a typical multi-objective optimization algorithm, well-suited for finding the overall optimum in a multi-dimensional space. The idea of differential evolution is largely inspired by Genetic Algorithm (GA). Both methods start with an initial population obtained randomly. Researchers use the fitness values of all individuals as selection criteria, and several operations are performed, including crossover, mutation, and selection [34]. The DE process can be summarized in the following four points.

1. Population initialisation

The initial population is obtained through the mechanism of uniform distribution by referring to the following equation.

$$X_i(0) = \{x_{i,1}(0), x_{i,2}(0), x_{i,3}(0)\ldots, x_{i,n}(0)\} \tag{28}$$

$$\begin{array}{l} x_{i,j}(0) = x_{j\min} + \mathrm{rand}(0,1) \cdot (x_{j\max} - x_{j\min}) \\ i = 1,2,3,\cdots, NP; j = 1,2,3,\cdots, n \end{array} \tag{29}$$

where $X_i(0)$ refers to the initial population. $x_{i,j}(0)$ specifically refers to the value taken in the $j-th$ dimension. rand (0,1) represents a value in the range of [0,1], which is highly random. $x_{jmax}\ and\ x_{jmin}$ denote the upper and lower bounds of the j-th dimension. NP typically falls within a certain range of $5 \times n$ and $10 \times n$, but generally, it is greater than or equal to $4 \times n$. Here, n refers to the number of dimensions.

2. Variant operations

The difference vector can be defined as follows:

$$D(k) = X_{p1}(k) - X_{P2}(k) \tag{30}$$

where $D(k)$ refers to the difference vector. $p1, p2$ are numbers in the range of [1, 2, . . . , NP], which have randomness. For superimposing the target vector to other individuals, refer to Equation (31):

$$V_i(k+1) = X_{p1}(k) + F \cdot [X_{p2}(k) - X_{p3}(k)] \tag{31}$$

where F refers to the scaling factor, which is usually taken as 0.5.

3. Cross operations

The target vector individual $\mathrm{x}_i(k)$ is cross-processed together with the variation vector $v_i(k+1)$ to obtain $u_i(k+1)$, to ensure that $\mathrm{x}_i(k)$ can evolve, and at least one of $u_i(k+1)$ is contributing to $\nu_i(k+1)$, as shown in Equation (32):

$$u_i(k+1) = \begin{cases} v_{i,j}(k+1), if\ rand(0,1) \le CR \\ x_{i,j}(k), \text{else} \end{cases} \tag{32}$$

where rand (0,1) is the value in the range of [0,1], which has randomness. *CR* refers to the cross-probability factor, which is in the range of [0,1].

4. Selection operations

The selection of the test vector $u_i(k+1)$ and the target vector $x_i(k)$ is carried out through the objective function, and iterated until the composite stopping condition according to the following equation, as shown in Equation (33):

$$x_i(k+1) = \begin{cases} u_i(k+1), f(u_i(k+1)) < f(x_i(k)) \\ x_i(k), f(u_i(k+1)) \geq f(x_i(k)) \end{cases} \tag{33}$$

where f is the fitness function [35,36].

2.2.3. Population Renewal Based on an EP Strategy

Using GWO for multi-robot multi-tasking path optimization in vertical farms faces challenges such as inefficiency and poor stability. To address these issues, the use of population updating based on an EP strategy is appropriate. Without an elitist preservation strategy, there is a risk that outstanding individuals may be lost during the evolutionary process, especially in the early stages of mutation or crossover operations. The elitist preservation strategy ensures the survival of these excellent individuals and accelerates the convergence of the GWO algorithm. By directly retaining the optimal individuals, the algorithm can approach the optimal solution region faster and reduce unnecessary exploration, thus improving overall optimization efficiency and algorithm stability. Moreover, a reasonable elitist preservation strategy can maintain population diversity while preserving excellent individuals [37].

In the field of path optimization, many researchers have used EP strategy to improve the performance of various algorithms. For example, Huo et al. [38] proposed a Swap-and-Judge Simulated Annealing (SJSA) algorithm to solve the path planning problem of unmanned aircraft in disaster relief. This algorithm improves the efficiency of generating feasible neighbouring solutions. In SJSA, an operational reservation strategy is used to enhance global convergence. Wang et al. [39] proposed an Elite-duplication GA (EGA) strategy to solve the path planning problem of unmanned surface ships by incorporating elite and diversity operations into the GA.

2.2.4. Comparative Analysis and Innovations of EPDE-GWO

By incorporating both Differential Evolution (DE) and Elitist Preservation (EP) strategies into the traditional GWO framework, the EPDE-GWO algorithm is formulated to enhance the overall optimisation process. Thus, as shown in Algorithm 1 the pseudocode for the EPDE-GWO algorithm outlines the key steps involved in this enhanced optimisation process. Each step of the algorithm is designed to exploit the strengths of both DE and EP to ensure that the search process is both comprehensive and efficient. The pseudocode provides a clear and structured representation of how these strategies are integrated into the GWO framework, demonstrating the algorithm's ability to handle complex optimisation tasks with improved accuracy and reliability.

In comparison to the traditional GWO algorithm, the EPDE-GWO algorithm introduces significant improvements in key areas. The integration of Differential Evolution (DE) strategies enhances the global search capability, allowing the algorithm to more effectively navigate complex, multimodal search spaces. Traditional GWO may sometimes struggle with premature convergence, particularly in environments with multiple local optima. By incorporating DE, the EPDE-GWO algorithm ensures a broader exploration of the search space, reducing the likelihood of getting trapped in suboptimal solutions and improving the overall quality of the optimization results. Additionally, the Elitist Preservation (EP) strategy in the EPDE-GWO algorithm ensures that the best solutions are retained throughout the optimization process. This addresses a limitation in the traditional GWO, where high-quality solutions might be lost due to the stochastic nature of the algorithm.

By preserving elite solutions across generations, the EPDE-GWO algorithm accelerates convergence and increases the stability of the results.

Algorithm 1. Elitist Preservation Differential Evolution Grey Wolf Optimizer (EPDE-GWO)

Input: SearchAgents_no, Max_iter, lb, ub, dim, fobj
Output: Alpha_pos, Convergence_curve

Initialize the population of wolves (Positions), and set parameters aaa, AAA, CCC.
Initialize α\alphaα, β\betaβ, δ\deltaδ positions and their fitness scores.

Evaluate the fitness of each wolf.
Update α, β, δ positions based on fitness scores.

for t = 1 to Max_iter do
    Update wolf positions using the differential evolution strategy.
    Evaluate the new fitness of each wolf.
    Update α, β, δ based on new fitness scores.
    Apply elitist selection to preserve the best wolves (α, β, δ).
    Update wolf positions using α, β, δ information.
    Record the best fitness value (α score) in the Convergence_curve.
end for

Return the best solution found (Alpha_pos) and the Convergence_curve

These enhancements make the EPDE-GWO algorithm particularly well-suited for challenging optimization tasks that demand both precision and robustness. In the context of vertical farm path planning, for example, the complexity of the environment with its multiple obstacles, constraints, and the need for efficient space utilization requires an algorithm that can effectively balance exploration and exploitation. The improved global search capability ensures that the algorithm can thoroughly explore potential paths, while the elitist preservation guarantees that the best-found solutions are not discarded. As a result, the EPDE-GWO algorithm not only finds shorter and more efficient paths but also does so with greater consistency and reliability, making it a powerful tool for optimizing operations in vertical farming and similar complex domains.

2.3. Solving the VFSMRTP Model Using the EPDE-GWO Algorithm

In order to validate the performance benefits of EPDE-GWO algorithm for path optimisation in vertical farms, we compared the performance of EPDE-GWO with other classical optimisation algorithms including PSO, GA, SA, DBO and GWO in solving VFSMRTP models. The baseline algorithms selected for comparison, GA, SA, DBO and PSO, were chosen based on their established effectiveness in solving optimisation problems, particularly in contexts similar to multi-robot path planning. GA and PSO are well-established optimisation algorithms that are widely used due to their versatility and effectiveness in global search tasks. SA was selected for its robustness in escaping local optima through its probabilistic technique, which allows it to explore a wide range of potential solutions and avoid premature convergence. DBO was selected due to its distinctive approach, which draws inspiration from the navigational strategies of dung beetles. This approach offers robust capabilities in maintaining a balance between exploration and exploitation. The selected algorithms represent a diverse set of approaches, thereby facilitating a comprehensive evaluation of the proposed EPDE-GWO algorithm's performance.

The hardware environment utilized for the experiments included an Intel(R) Core Trade Mark (TM) i7-11800H processor as the experiment platform (Intel, Santa Clara, CA, USA), with a clock speed of 2.30 gigahertz (GHz), 32 gigabyte (GB) of random-access memory (RAM), and a GeForce Ray Tracing eXtreme (RTX) 3060 graphics processor with 6GB of video memory (NVIDIA, Santa Clara, CA, USA). The software environment uses Matrix Laboratory (MATLAB) R2023b and Microsoft Excel (Microsoft 365) for model construction and algorithm simulation. The experiments were repeated 50 times in the same experimental environment with the algorithm parameters shown in Table 1. The path length of each iteration of each algorithm was recorded for each experiment. The optimal path length, the worst path length, the average path length and the standard deviation of each algorithm were summarised and calculated and tabulated for comparison. In addition, the average convergence curves for each algorithm were plotted to illustrate the convergence performance of the different algorithms.

Table 1. Algorithm parameters.

Algorithm	Parameters	
GA	$n = 2$ $p1 = 0.8$ $lb = 0$	$Max\_iter = 100$ $p2 = 0.3$ $ub = 1$
PSO	$n = 20$ $w = 0.4$ $c2 = 1.2$ $lb = 0$	$Max\_iter = 100$ $c1 = 1.2$ $v = randn(size(x))$ $ub = 1$
SA	$T0 = 1000$ $Tr = 0.99$ $lb = 0$	$Max\_iter = 100$ $ub = 1$
DBO	$n = 20$ $p = 0.2$ $lb = 0$	$Max\_iter = 100$ R = Linearly decreases from 1 to 0 $ub = 1$
GWO, EPDE-GWO	$n = 20$ $p1 = 0.8$ $p2 = 0.3$ $lb = 0$	$Max\_iter = 100$ a = Linearly decreases from 2 to 0 $c = 2r$ $ub = 1$

3. Results

3.1. Results of Ablation Study

3.1.1. Effectiveness of Population Diversification Based on DE Strategies

To facilitate a clear understanding of the experimental setup, Table 2 lists the key symbols, terms, and a brief description. This table provides a concise reference for the various metrics used in the analysis. This table helps to contextualise the results and highlights the importance of the comparative performance metrics discussed in this section.

Table 3 describes that the minimum number of convergences for both GWO and DE-GWO is one iteration, which implies that in some experiments both algorithms find the optimal solution quickly. The maximum number of convergences for GWO is 36 iterations, while that for DE-GWO is 65 iterations. The maximum number of convergences for DE-GWO is higher than that of GWO, which indicates that in some experiments DE-GWO requires more iterations to find the optimal solution. The average convergence number of GWO is 12.4 iterations, while that of DE-GWO is 22.8 iterations. The average convergence number of DE-GWO is higher than that of GWO, which indicates that DE-GWO needs more iterations than GWO in order to reach the optimal solution in some experiments. The standard deviation of GWO is 9.7 while that of DE-GWO is 18.3. The standard deviation of DE-GWO is higher than that of GWO, which indicates that the convergence number of DE-GWO fluctuates more in different experiments, meaning the performance of DE-GWO is not as stable as that of GWO.

Table 2. Glossary of terms for comparative experiments.

Term/Symbol	Description
Min convergence iteration	The minimum number of iterations required for the algorithm to converge to a solution
Max convergence iteration	The maximum number of iterations required for the algorithm to converge to a solution
Mean convergence iteration	The average number of iterations required for the algorithm to converge across all trials
Standard deviation	A measure of the variation in convergence iterations across different runs
Min path length	The shortest path length found by the algorithm in a particular trial
Max path length	The longest path length found by the algorithm in a particular trial
Mean path length	The average path length found by the algorithm across all trials
Top 3 best and worst solutions	The three best and three worst path lengths found across all trials
IQR (Interquartile Range)	The range between the first quartile (Q1) and the third quartile (Q3) of the path lengths
Optimal path length	The shortest possible path length achievable by the algorithm in the given environment
Mean number of iterations	The average number of iterations taken across all trials
Average iteration time	The average time taken per iteration during the algorithm's execution

Table 3. Iteration counts of GWO and DE-GWO comparison.

Algorithm	Min Convergence Iteration	Max Convergence Iteration	Average Convergence Iteration	Standard Deviation
GWO	1	36	12.4	9.7
DE-GWO	1	65	22.8	18.3

Table 4 shows that the minimum path length for GWO is 824 m, while for DE-GWO it is 810 m. The smaller minimum path length of DE-GWO suggests that it can find shorter paths than GWO in some experiments, indicating better performance in some cases. The maximum path lengths for both GWO and DE-GWO are 948 m, suggesting that the worst-case scenario is the same for both algorithms. The average path length for GWO is 865.3 m, while for DE-GWO it is 850.4 m. This indicates that, on average, DE-GWO finds shorter paths, meaning its overall performance is better than that of GWO. The standard deviation of GWO is 16.5, while that of DE-GWO is 23.9. The larger standard deviation of DE-GWO indicates that the path lengths vary more in different experiments, showing less stability compared to GWO.

Table 4. Path lengths of GWO and DE-GWO comparison.

Algorithm	Min Path Length (m)	Max Path Length (m)	Mean Path Length (m)	Standard Deviation
GWO	824	948	865.3	16.5
DE-GWO	810	948	850.4	23.9

Figure 4 depicts the distribution of the top three best and worst solutions. The horizontal axis represents the different algorithms, while the vertical axis indicates the number of occurrences. The blue colour represents the top three best solutions, and the red colour represents the top three worst solutions. The figure shows that DE-GWO achieves nine counts of the top three best solutions, whereas GWO only achieves three counts. This indicates that DE-GWO has a better ability to find more global optimal solutions, and the quality of these global optimal solutions is higher than those found by GWO.

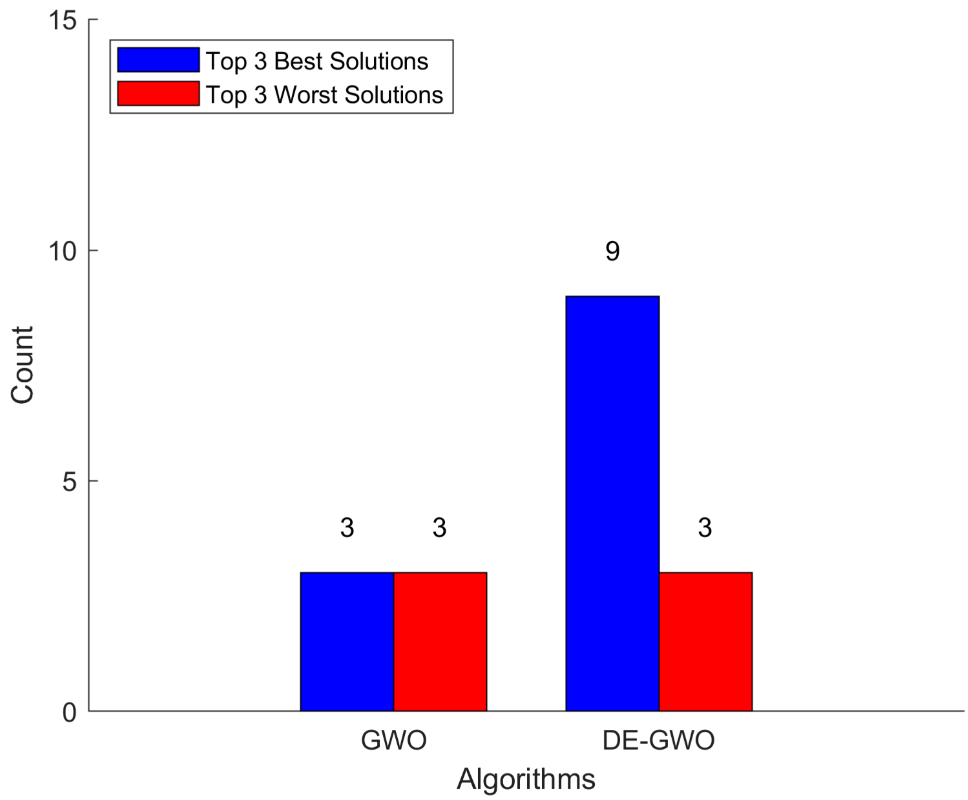

Figure 4. Distribution of top three best and worst solutions.

The average path lengths of the GWO and DE-GWO algorithms in each iteration are calculated, and convergence curves are plotted to show the algorithms' convergence during the iterations. As shown in Figure 5, it can be seen that the GWO algorithm reaches the convergence state faster than the DE-GWO algorithm during the iteration process, i.e., the optimal path is found within a smaller number of iterations. However, the convergence value of the GWO algorithm is larger than that of DE-GWO, indicating that GWO falls into a local optimum. The slope of the DE-GWO convergence curve is significantly steeper than that of GWO, indicating that DE-GWO approaches the global optimum solution faster than GWO during the search process.

The discrepancies in convergence behaviour between GWO and DE-GWO can be directly attributed to the algorithmic enhancements introduced in DE-GWO. The DE strategies introduced enhancements to the exploration and refinement mechanisms. These strategies permit DE-GWO to sustain diversity within the population of solutions, which is vital in preventing the algorithm from becoming trapped in local optima. By enhancing the global search capabilities, DE-GWO is able to explore a more extensive range of potential paths, thereby increasing the probability of identifying more optimal solutions. Consequently, DE-GWO is not only capable of identifying more global optimal solutions, but also of producing solutions of a superior quality. Nevertheless, this also gives rise to a greater degree of variability in the results, as evidenced by the elevated standard deviation.

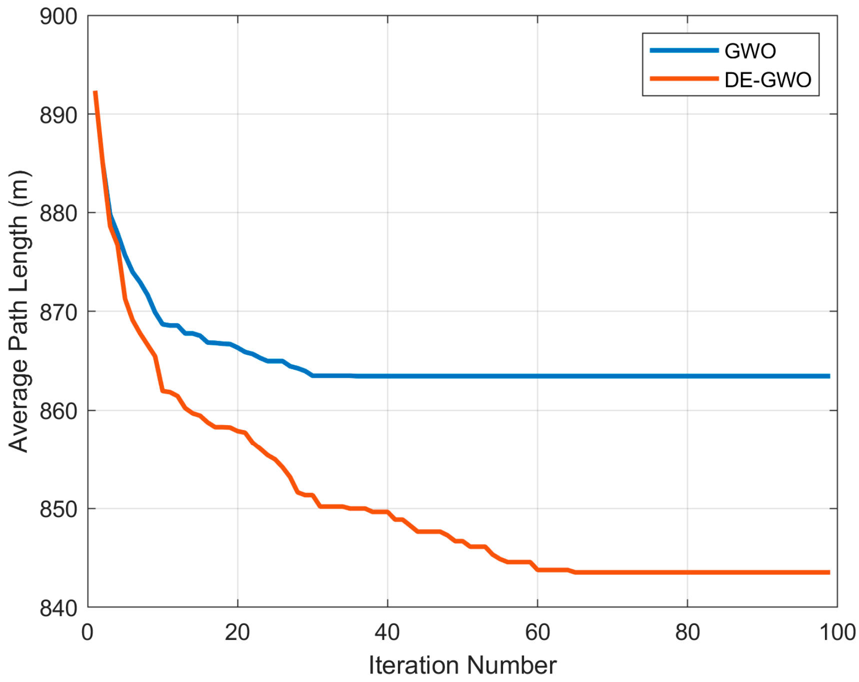

Figure 5. Convergence curves of GWO and DE-GWO.

3.1.2. Effectiveness of EP Strategy

Figure 6 shows a box plot of converged path lengths. The box plots show the distribution of converged path lengths for each algorithm in the experiments. EP-GWO's median converged path length is lower. The box has smaller Interquartile Range (IQR), indicating a more concentrated data distribution and higher stability. There are a small number of outliers, but the overall fluctuation is small. GWO's median of the converged path length is higher. The box has larger IQR, indicating a more dispersed data distribution and lower stability. A larger number of outliers exist, indicating greater fluctuations in the results.

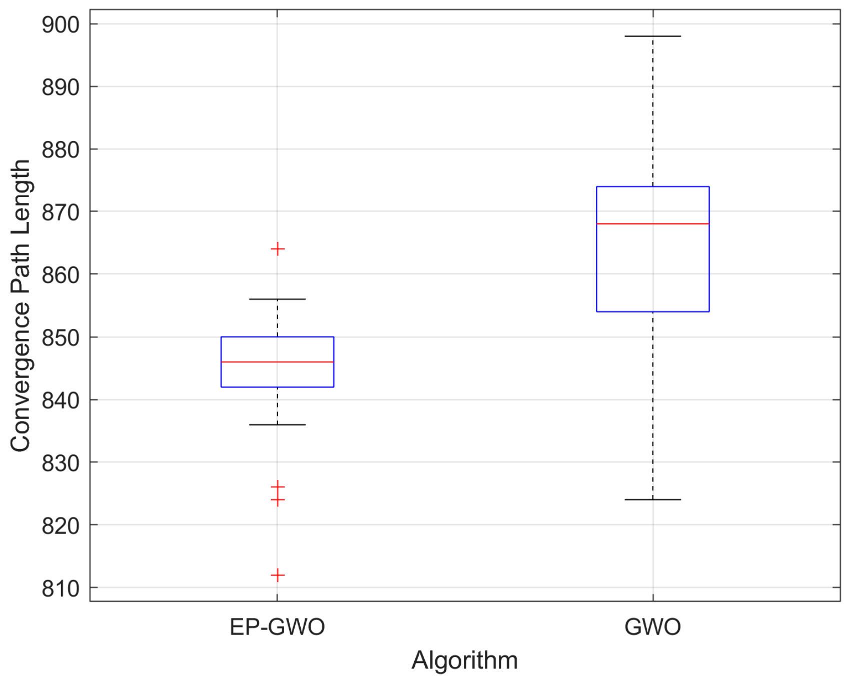

Figure 6. Box plots of convergence path lengths for EP-GWO and GWO.

Figure 7 is a box plot of the number of converged iterations. EP-GWO's median number of converged iterations is lower. The box has smaller IQR, indicating a more concentrated data distribution and higher stability. A small number of outliers exists, indicating slower convergence in individual cases. The number of iterations ranges from 1 to 26, and most of the data are clustered around the median 7 iterations. GWO's median (9.5) number of converged iterations is higher. The box has larger IQR, indicating a more dispersed data distribution and lower stability. There are some outliers, indicating that the results are more volatile.

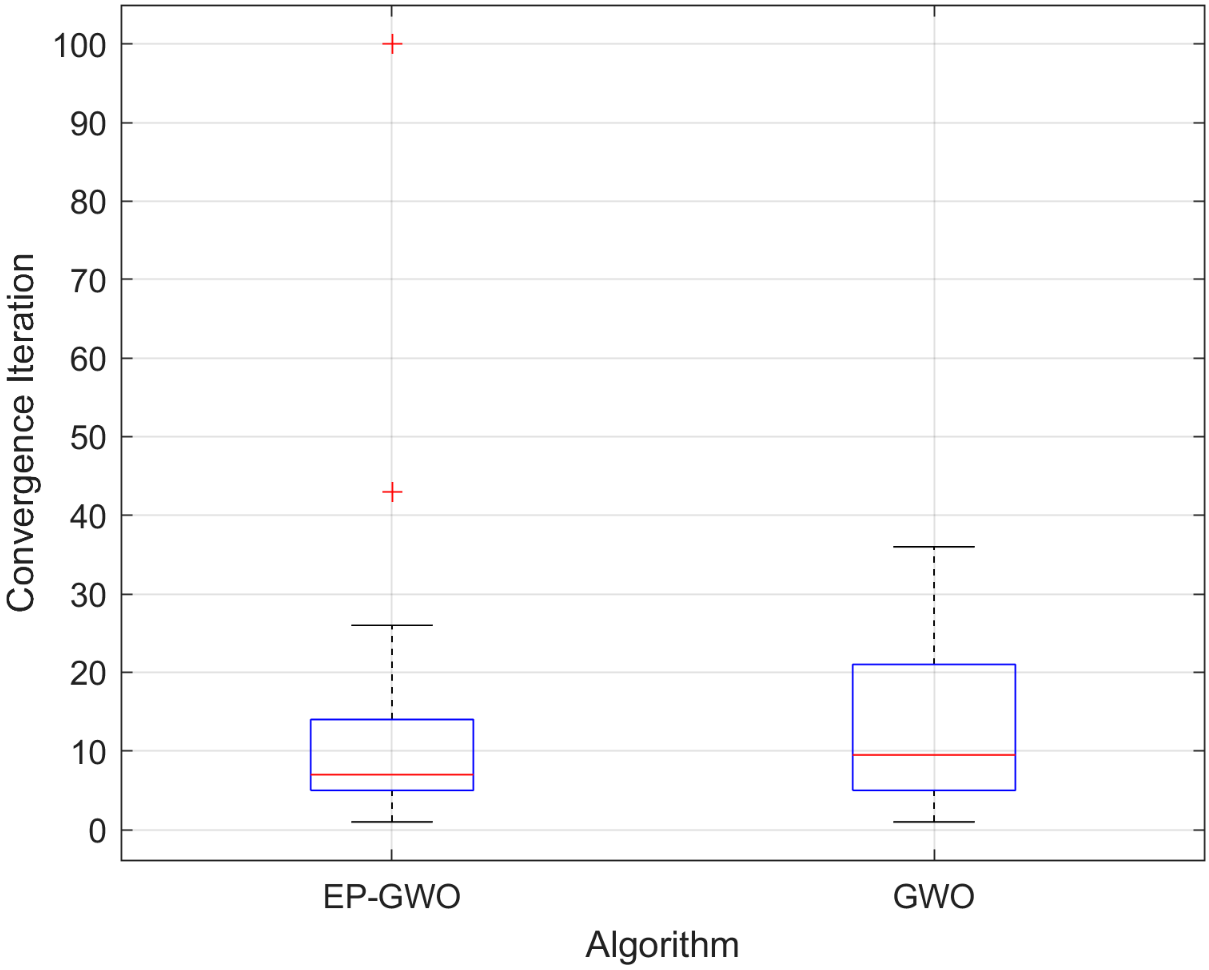

Figure 7. Box plots of convergence iterations for EP-GWO and GWO.

The results presented in Figures 8 and 9 demonstrate the substantial benefits of the Elitist Preservation strategy in the EP-GWO algorithm. The lower median values and smaller IQR in both converged path lengths and iteration counts indicate that EP-GWO consistently achieves more stable and reliable optimisation outcomes in comparison to GWO. By retaining and refining the optimal solutions throughout the iterative process, EP-GWO reduces variability and enhances convergence efficiency, resulting in a reduction in fluctuations and a tighter distribution of results. In contrast, the higher median values, larger IQR, and greater number of outliers observed in GWO reflect its susceptibility to variability and instability. The results demonstrate the significance of the EP strategy, as outlined in EP-GWO, for attaining consistent and efficient performance in intricate optimisation tasks.

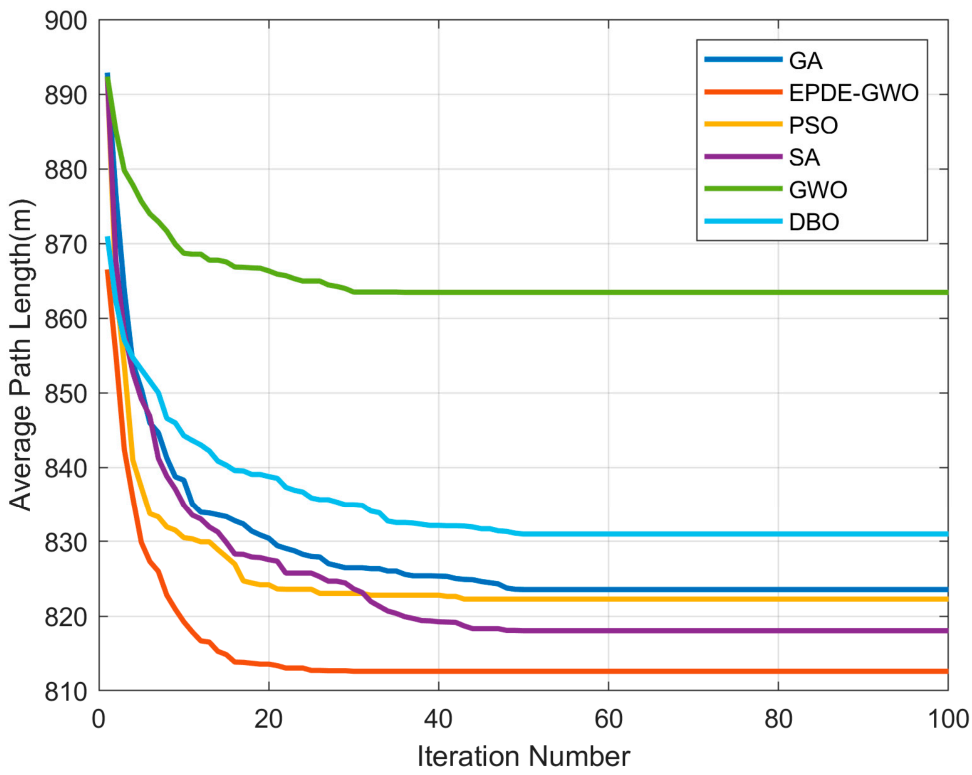

Figure 8. Average convergence curves of different algorithms.

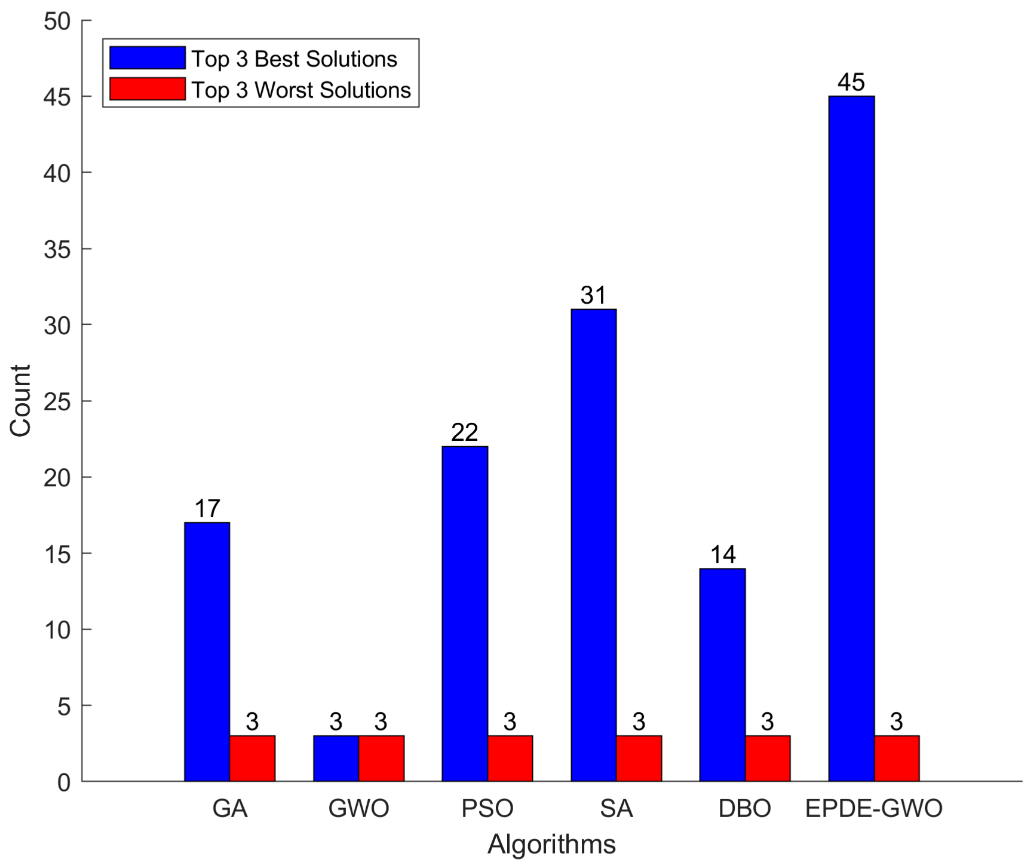

Figure 9. Distribution of top three best and worst solutions.

3.2. EPDE-GWO Algorithm Performance Analysis

The primary objective of this section is to undertake a comparative analysis of EPDE-GWO with other established classical optimisation algorithms, including PSO, GA, DBO, SA and GWO. The performance of each of these algorithms will be evaluated when solving the VFSMRTP model in a complex vertical farm environment, with a view to validating the performance advantages of EPDE-GWO for path optimisation.

3.2.1. Comparison of Energy Consumption

From Table 5, it can be seen that the optimal path lengths of all algorithms are 810 m, except for GWO, which has an optimal path length of 824 m. This indicates that under optimal conditions, all algorithms except GWO find the same shortest paths. The worst path length of EPDE-GWO is 912 m, which is lower than that of the other algorithms (948 m). This indicates that the EPDE-GWO algorithm still finds better paths in the worst-case scenario, while the other algorithms have relatively longer path lengths in the worst-case scenario. The mean path length of EPDE-GWO is 815.1 m, which is lower than that of the other algorithms, indicating that EPDE-GWO outperforms the others in terms of overall performance. The mean path lengths of GA, PSO, DBO, and SA are relatively close to one another, with respective values of 828.3 m, 825.2 m, 835.0 m, and 823.6 m. GWO has the highest path mean value of 865.3 m, indicating the worst performance among the compared algorithms.

Table 5. Experimental path length results comparison.

Algorithm	Min Path Length (m)	Max Path Length (m)	Mean Path Length (m)	Standard Deviation
GA	810	948	828.3	17.2
EPDE-GWO	810 *	912 *	815.1 *	10.4 *
PSO	810	948	825.2	16.2
SA	810	948	823.6	18.2
GWO	824	948	865.3	16.5
DBO	810	926	835.0	19.6

Note: * Best in class.

Figure 8 shows the average convergence curves of different algorithms. The horizontal axis represents the iteration number, the vertical axis represents the average path length, and different colours indicate different algorithms. It can be seen that the EPDE-GWO algorithm converges to shorter path lengths faster than the other algorithms during the iteration process. Additionally, as the number of iterations increases, the average path length for EPDE-GWO remains the lowest, indicating that EPDE-GWO is superior to the other algorithms in terms of global optimality-finding ability.

From the statistics of the experimental results (Table 4) and the average convergence curves (Figure 9), it can be seen that the EPDE-GWO algorithm outperforms the other classical optimisation algorithms in all metrics. The EPDE-GWO algorithm can find better solutions and shorter travelling paths, thus saving more energy. The length of the unoptimized path is 1076 m, so the EPDE-GWO algorithm can save 24.6% of the path length (energy consumption).

The innovative design of the EPDE-GWO algorithm is a key factor in its superior performance in terms of energy consumption and path optimisation. The integration of DE and EP strategies into the GWO framework serves to enhance the algorithm's global search capabilities, whilst simultaneously preventing premature convergence to local optima. The DE strategy plays a pivotal role in augmenting the global search capabilities of the algorithm by fostering diversity within the population and forestalling premature convergence to local optima. This broader exploration of the search space enables the EPDE-GWO algorithm to identify shorter paths with greater consistency, even when presented with varying conditions.

This is reflected in its lower worst-case and mean path lengths. Furthermore, the EP strategy enhances this performance by ensuring that the optimal solutions are preserved across generations, thereby facilitating faster convergence and more reliable optimisation results. The integrated DE and EP strategies not only enhance the exploratory and exploitative capabilities of the algorithm, but also ensure that these capabilities are efficiently applied to solve complex, real-world problems such as path planning in vertical farms. The reduction in path length achieved by EPDE-GWO is a direct consequence of its capacity to balance exploration and exploitation, thereby avoiding the common pitfalls of local optima and ensuring the identification of the most efficient paths. This makes EPDE-GWO a powerful tool for optimising energy consumption in environments where efficiency is critical.

3.2.2. Comparison of Convergence Efficiency

In Table 6, the data indicate that the optimal number of convergence iterations for the EPDE-GWO, SA, and GWO algorithms is one iteration, indicating that under optimal conditions, these algorithms can converge to the optimal solution in a single iteration. The GA and PSO algorithms have an optimal number of convergence iterations of two iterations, which is slightly more than that of EPDE-GWO, SA, DBO, and GWO. The worst number of convergence iterations for the EPDE-GWO algorithm is 30 iterations, which is lower than those of the other algorithms, suggesting that EPDE-GWO converges faster even in the worst-case scenario. The worst numbers of convergence iterations for the other algorithms are GA (50 iterations), PSO (43 iterations), SA (50 iterations), DBO (49 iterations), and GWO (36 iterations). Clearly, EPDE-GWO performs better in this regard.

Table 6. Experimental convergence count results comparison.

Algorithm	Min Convergence Iteration	Max Convergence Iteration	Mean Convergence Iteration	Standard Deviation	Average Iteration Time (s)
GA	2	50	23.9	14.2	1.22
EPDE-GWO	1 *	30 *	11.7 *	6.8 *	2.45
PSO	2	43	11.7	9.1	1.15
SA	1	50	22.4	13.9	1.11
GWO	1	36	12.4	9.7	1.23
DBO	1	49	22.8	14.8	1.20

Note: * Best in class.

The average number of convergence iterations for both EPDE-GWO and PSO algorithms is 11.7 iterations, which is significantly better than the other algorithms, suggesting that these two algorithms perform the best in terms of overall convergence efficiency. Although PSO has the same average number of convergence iterations as EPDE-GWO, EPDE-GWO has a better worst-case convergence iteration number and standard deviation, indicating that it performs more consistently across all scenarios. Therefore, the EPDE-GWO algorithm demonstrates superior performance when compared to the PSO algorithm.

In addition to comparing the convergence performance of the different algorithms, we also evaluated their computational cost, which is an important factor in practical applications. The computational cost is measured in terms of the average iteration time for each algorithm, as shown in Table 5. The EPDE-GWO algorithm, while demonstrating the fastest convergence and highest performance in path optimization, incurs the highest computational cost with an average iteration time of 2.45 s. This is notably higher than the other algorithms, where the average iteration time ranges from 1.11 to 1.23 s. The increased computational cost for EPDE-GWO is primarily due to the additional complexity introduced by the DE and EP strategies, which enhance the algorithm's optimization capabilities.

Figure 9 depicts the distribution of the top three best and worst solutions. The horizontal axis represents the different algorithms, while the vertical axis indicates the number of occurrences. The blue colour represents the top three best solutions, and the red colour represents the top three worst solutions. It shows that EPDE-GWO achieves 45 counts of the top three best solutions, with a standing ratio of 90%. This demonstrates that EPDE-GWO has higher convergence efficiency and a greater probability of converging to the optimal solution.

The superior convergence performance of the EPDE-GWO algorithm is a direct consequence of its algorithmic improvements, as evidenced by its capacity to reach optimal solutions in fewer iterations and its consistency across different scenarios. The capacity of the EPDE-GWO algorithm to achieve convergence in a single iteration under optimal conditions serves to illustrate the efficiency of its algorithmic design. The incorporation of DE into the GWO framework is of significant importance. The DE component of the algorithm enhances its global search capability, allowing it to rapidly explore a diverse set of potential solutions and thereby increasing the likelihood of identifying the optimal solution at an early stage of the process. Furthermore, the EP strategy facilitates the retention of optimal solutions, thereby reducing the number of iterations required to refine and reach the optimal solution, and thus minimising the worst-case convergence time. The high frequency with which EPDE-GWO achieves optimal solutions, as illustrated in Figure 10, can be attributed to the strategic integration of DE and EP within the algorithm. The combined effect of these strategies not only increases convergence efficiency, but also improves the reliability of the algorithm in consistently identifying the optimal solution.

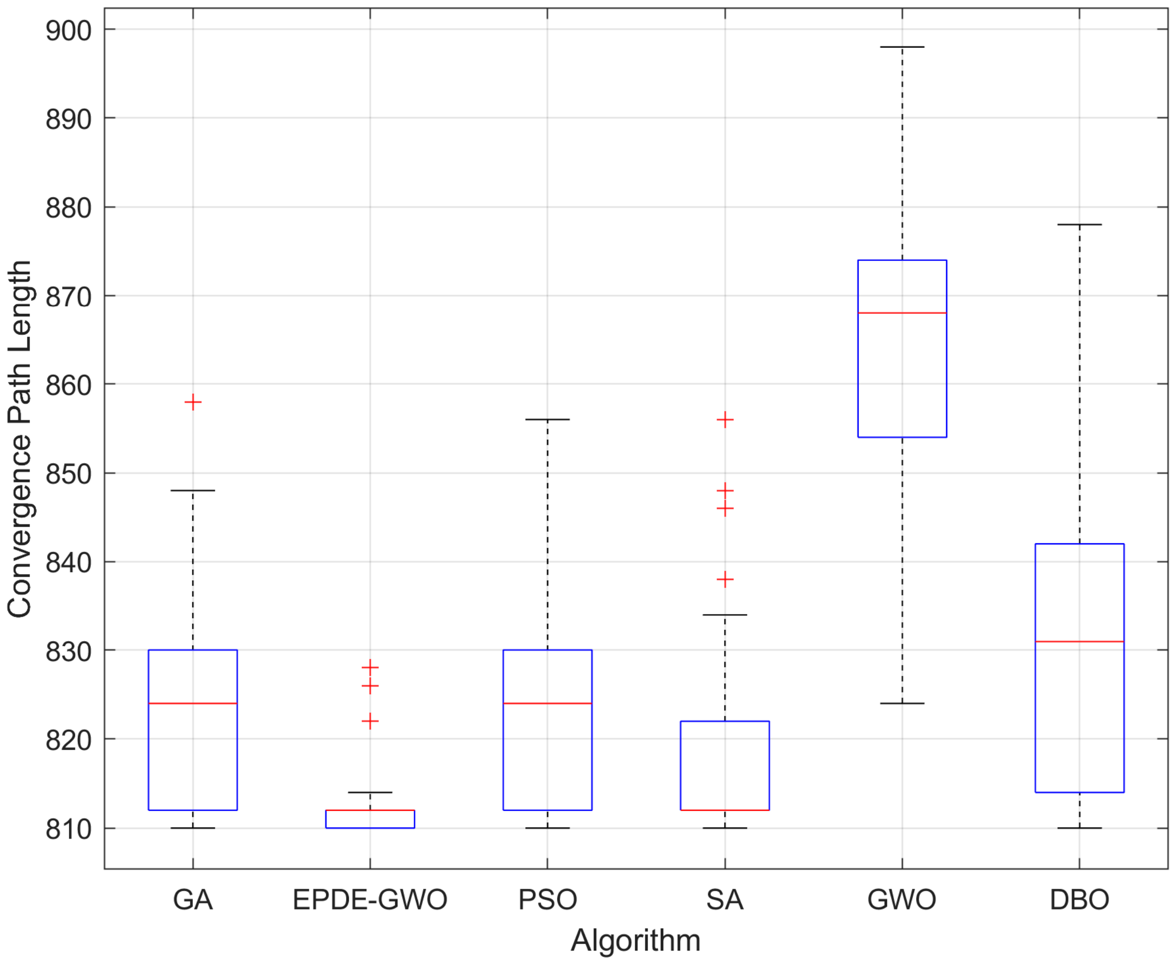

Figure 10. Box plots of convergence path lengths for different algorithms.

3.2.3. Comparison of Algorithm Stability and Robustness

As shown in the path length box plot in Figure 10, the x-axis lists the six different algorithms tested and the y-axis represents the convergence path length. The EPDE-GWO algorithm exhibits a more concentrated path length distribution and a smaller interquartile range (IQR), indicating enhanced stability. The GA, PSO, and SA algorithms demonstrate comparable performance and a certain degree of stability; however, SA displays a higher incidence of outliers. The GWO and DBO algorithms exhibit larger IQRs and a higher median, indicating suboptimal performance in terms of both stability and optimisation outcomes.

As shown in Figure 11, the x-axis lists the six different algorithms tested and the y-axis represents the number of iterations required for each algorithm to converge. The EPDE-GWO and PSO algorithms exhibit a more concentrated distribution of iteration counts and smaller interquartile ranges (IQRs), suggesting that their convergence speeds are consistent and stable across multiple experiments. In contrast, the interquartile ranges (IQRs) of GA, SA and DBO are considerable, indicating that their stability is poor. The median number of iterations is higher for GA, SA and DBO, indicating that these three algorithms require more iterations to reach a stable solution.

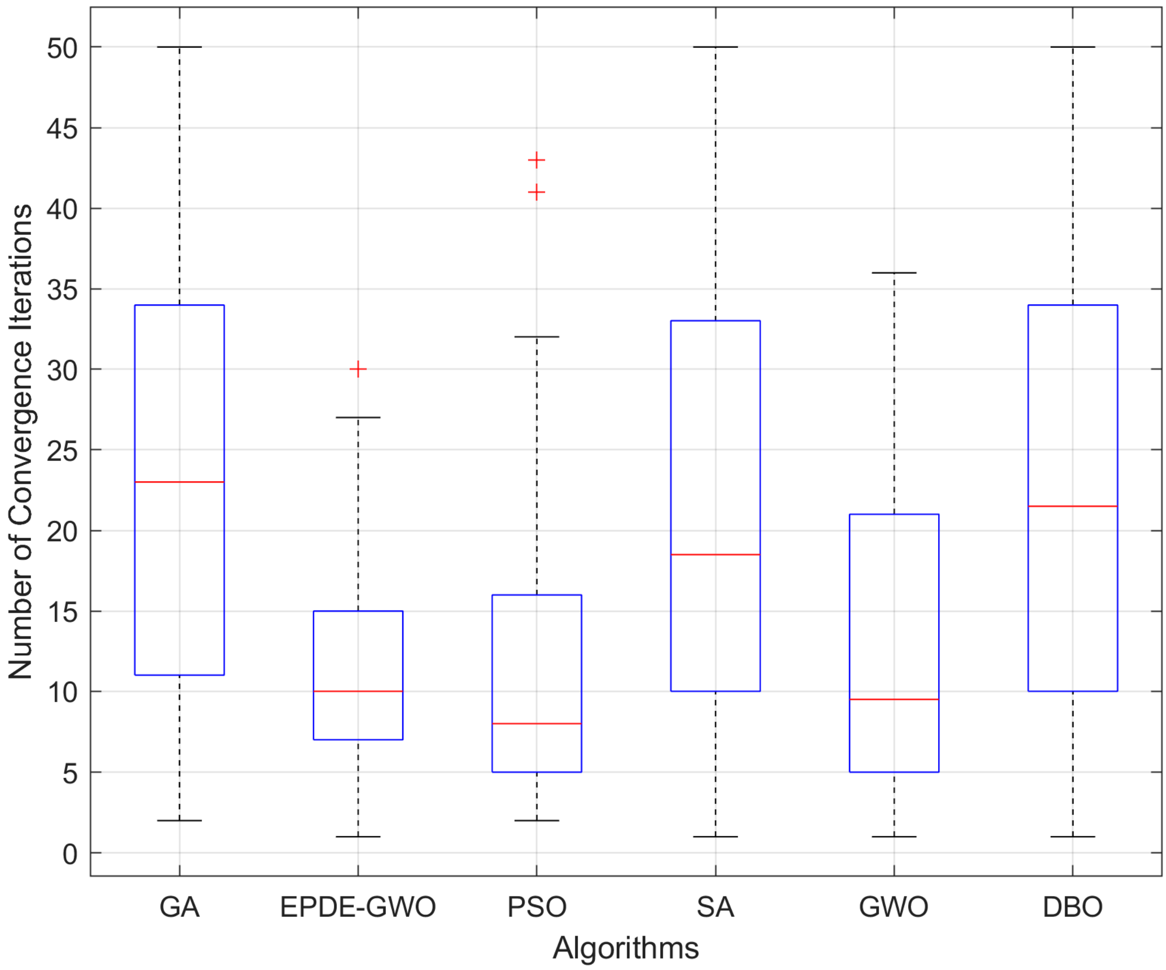

Figure 11. Box plots of the number of convergence iterations for different algorithms.

Table 7 presents a comparison of the performance of different algorithms in optimizing the path length for varying numbers of agri-robots. It shows that the EPDE-GWO algorithm performs best in terms of optimal path length, average path length, average number of iterations, and standard deviation for any quantity of robots, indicating an advantage in path planning. EPDE-GWO not only finds shorter paths but also converges faster, with more stable results. In contrast, the other algorithms do not perform as well as EPDE-GWO in terms of path length and number of iterations, and their results are less stable. This suggests that EPDE-GWO is more efficient in path optimization, particularly as the number of agri-robots increases.

Figure 12 describes the average number of iterations with varying numbers of agri-robots. GA's average number of iterations is relatively stable, remaining around 25 iterations throughout the process, showing some robustness. For the SA and DBO algorithms, the average number of iterations initially increases and then decreases, with large fluctuations, indicating relatively poor robustness. The EPDE-GWO algorithm has a relatively smooth trend in the average number of iterations, remaining stable, especially when the number of robots is high. The average number of iterations is always low, demonstrating good robustness. With PSO, as the number of robots increases from 6 to 10, the average number of iterations decreases significantly, showing some robustness, but it fluctuates considerably in the early stages. For the GWO algorithm, the average number of iterations keeps increasing as the number of agri-robots increases, showing poor robustness.

Table 7. Experimental results of varying numbers of robots.

Number of Agri-Robots	Algorithm	Optimal Path Length (m)	Mean Path Length (m)	Mean Number of Iterations
4	GA	1740	1775.9	25.5
	EPDE-GWO	1694	1718.3	20.1
	SA	1710	1767.5	24.5
	PSO	1740	1744.7	28.3
	GWO	1754	1806.3	16.6
	DBO	1742	1780.5	27.2
6	GA	1724	1751.7	24.7
	EPDE-GWO	1676	1701.4	18.2
	SA	1694	1735.6	38.1
	PSO	1692	1721.9	31.2
	GWO	1752	1787.9	16.5
	DBO	1732	1758.4	34.1
8	GA	1600	1634.5	27.1
	EPDE-GWO	1578	1599.1	20.8
	SA	1598	1632.6	35.1
	PSO	1578	1657.3	32.4
	GWO	1622	1654.2	19.8
	DBO	1612	1640.8	30.4
10	GA	1516	1542.2	28.7
	EPDE-GWO	1478	1493.4	21.4
	SA	1500	1528.8	27
	PSO	1478	1529.9	22.3
	GWO	1538	1598.1	29.1
	DBO	1524	1555.3	24.4

Figure 13 describes the variation of the average path length with varying numbers of robots. The GA and DBO algorithms demonstrate stable performance with shorter path lengths. For the GWO algorithm, the average path length decreases with the increase in the number of robots but fluctuates significantly in some phases, indicating some instability. The EPDE-GWO algorithm's average path length decreases continuously with a smoother trend, demonstrating good overall performance and robustness with shorter path lengths. The PSO algorithm's average path length decreases gradually with increasing numbers of robots. Although there are some fluctuations in the early stages, the overall performance is better. The SA algorithm's average path length is longer than that of the other algorithms but decreases gradually with increasing numbers of robots, indicating more stable performance.

The superior performance of the EPDE-GWO algorithm can be attributed to its innovative design, which integrates DE and EP strategies. This is evidenced by its centralised distribution of path lengths, consistent convergence speeds, and lower variability in results. The DE strategy enhances the global search capability by promoting diversity in the population, thereby enabling the algorithm to explore the search space thoroughly and avoid premature convergence to local optima. This robust exploration enables the EPDE-GWO algorithm to consistently identify shorter paths, even in complex scenarios involving a larger number of robots. Concurrently, the EP strategy guarantees that the optimal solutions are retained and refined throughout the optimisation process, thereby facilitating a more rapid convergence and greater stability across iterations. The combination of these strategies results in a more predictable, efficient, and scalable optimisation process compared to traditional algorithms, which may lack such mechanisms and thus exhibit greater variability and instability.

Furthermore, the algorithm's capacity to maintain a stable and low average number of iterations, particularly as the problem scale increases, underscores its robustness and adaptability. The smooth trend in performance, as evidenced by the continuous decrease in average path length with increasing robot numbers, serves as a testament to EPDE-GWO's ability to effectively balance exploration and exploitation. In contrast to alternative algorithms that may exhibit pronounced fluctuations or instability due to less effective search and preservation mechanisms, the design of EPDE-GWO ensures consistent efficiency and robustness, rendering it particularly well-suited for complex path planning tasks. Although the computational cost is higher due to the added complexity of DE and EP, this is justified by the superior optimisation results delivered by the algorithm, which demonstrates that the design choices in EPDE-GWO lead to significant practical advantages.

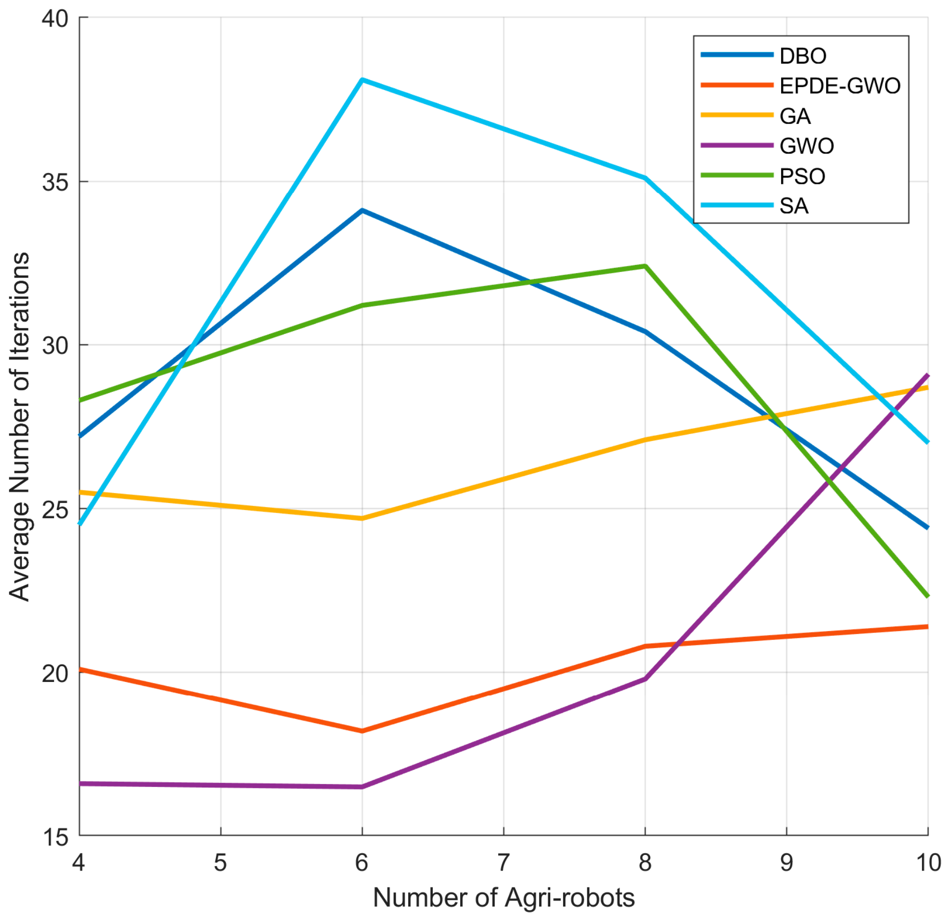

Figure 12. Average number of convergences with varying numbers of agri-robots.

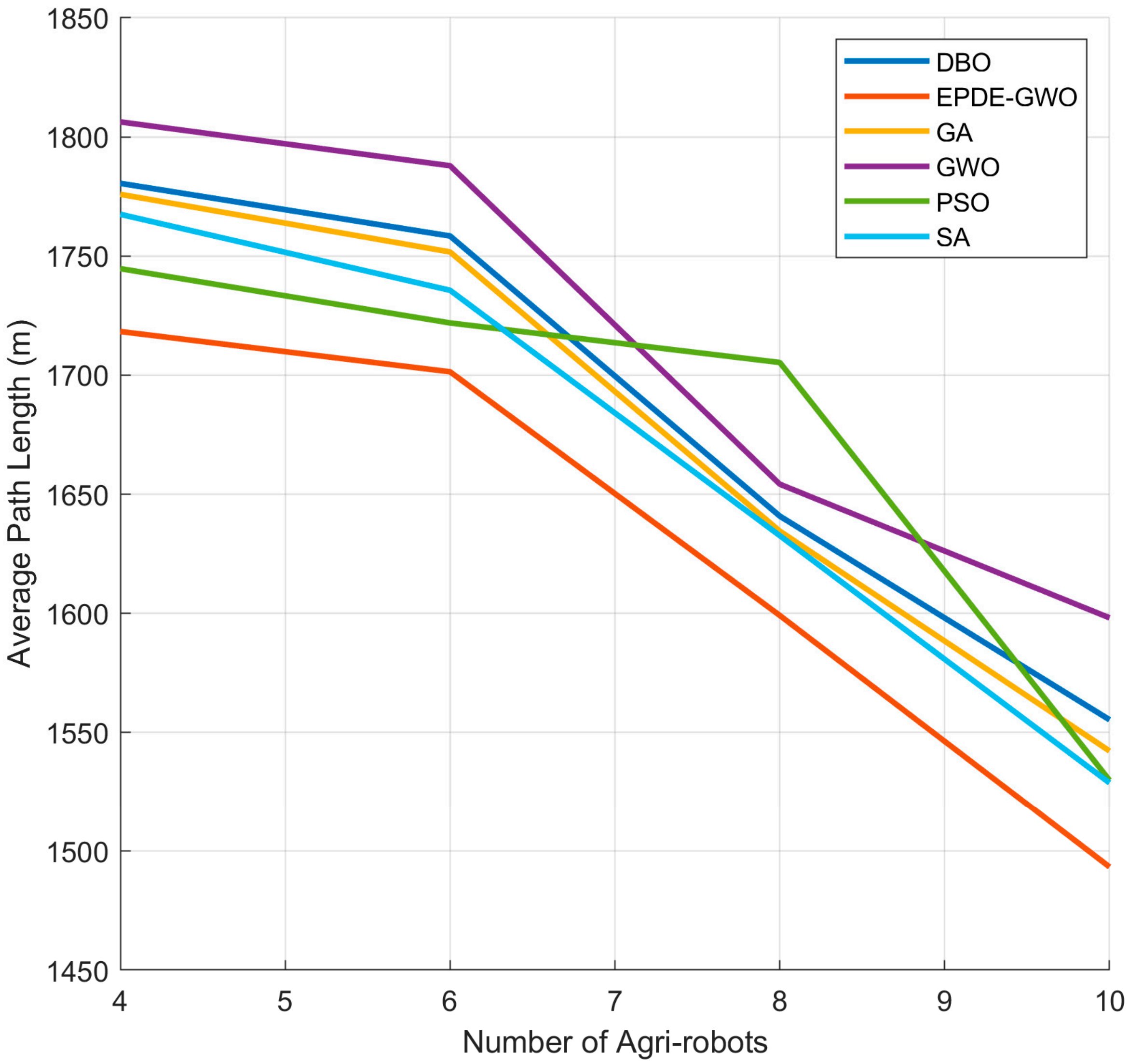

Figure 13. Average path with varying numbers of agri-robots.

4. Discussion

The results of the ablation experiments show that the DE-GWO algorithm finds the global optimal solution more quickly by introducing a population updating approach based on a differential evolution strategy. This is because the mutation and crossover operations in DE facilitate the exploration and utilization of the search space by combining existing individuals to introduce new ones. This improves the efficiency of the search process and allows the algorithm to converge to a high-quality solution more quickly. However, this process of pursuing globally optimal solutions also imposes certain limitations on the DE-GWO algorithm. As shown in Tables 3 and 4, the standard deviation of DE-GWO is larger than that of GWO. This indicates that the DE-GWO algorithm is less stable than the GWO algorithm. The improved stability of the EP-GWO algorithm is attributed to its elitist preservation strategy, which retains the optimal individuals in each generation. This approach ensures that the best genes are passed on to subsequent generations, thereby enhancing the overall quality of individuals in the population. Consequently, this mechanism prevents the algorithm from converging to local optima and minimizes significant fluctuations. In contrast, the basic GWO algorithm lacks this mechanism and is more susceptible to random factors, leading to substantial variations in path length and reduced stability.

Secondly, the EPDE-GWO algorithm, which combines both EP and DE strategies into the basic GWO, is proposed. By setting the same experimental parameters in the VFSMRTP model, a comprehensive comparison with GA, SA, PSO, DBO, and GWO algorithms was made. The results show that EPDE-GWO has the shortest average path length (815.1 m), the fastest convergence (11.7 iterations), the smallest IQR, and a stable convergence curve. Therefore, the improved algorithm effectively addresses the shortcomings of the GWO algorithm, such as poor global optimisation-seeking ability and poor stability.

The increased computational complexity of the EPDE-GWO algorithm, resulting from the integration of DE and EP strategies, is substantiated by its superior convergence speed, enhanced global optimisation, and increased stability. These enhancements result in more efficient path planning, which in turn reduces energy consumption and enhances economic efficiency in vertical farms. The experimental results corroborate the algorithm's resilience and scalability across varying numbers of robots, thereby attesting to its efficacy in both small- and large-scale operations. Although the computational expense is greater, it is offset by substantial energy savings and enhanced economic outcomes, rendering the trade-off justifiable.

5. Conclusions

In this paper, a multi-tasking path planning model for multi-agri-robots is investigated in the complex environment of vertical farms and the EPDE-GWO algorithm is proposed to solve it. Finally, it is compared with various other algorithms in the same environment. The main conclusions are as follows. First, the VFSMRTP model is built based on the environmental constraints, task constraints, and constraints between agri-robots in vertical farms. Second, the GWO algorithm is improved using DE and elite retention strategies, and the EPDE-GWO algorithm is proposed, and ablation experiments are conducted to verify the impact of each module on the overall performance of the algorithm. Finally, the performance of the EPDE-GWO algorithm is compared with other algorithms in the VFSMRTP model, and the results show that the EPDE-GWO algorithm excels in global optimisation capability, convergence speed, algorithmic stability, and robustness.

While the EPDE-GWO algorithm has demonstrated superior performance in optimizing path planning for vertical farm robots, it is important to acknowledge certain limitations of the study. A key limitation is the increase in computational cost associated with the algorithm. The enhanced complexity introduced by the DE and EP strategies results in a higher average iteration time of 2.45 s, which is greater than that of the other algorithms compared in this study. This increased computational cost could pose challenges in scenarios where computational resources are limited or where real-time processing is required. Future research could explore ways to optimize the algorithm further to reduce its computational burden while maintaining its performance benefits.

In the future, we will focus our research on the collaboration of different types of agri-robots to perform complex agricultural tasks. Another avenue is the study of adaptive strategies of the algorithm when facing uncertainty or dynamic changes of the environment in vertical farms. Moreover, future research will explore integrating advanced machine learning techniques such as Reinforcement Learning and deep learning-based predictive models, along with more advanced optimization algorithms, including recent developments in swarm intelligence, hybrid methods, and dynamic optimization techniques. Additionally, the scalability of the algorithm to larger robot clusters will be investigated, addressing the challenges of increased complexity in path planning for large-scale multi-robot systems. These techniques hold significant potential for enhancing the efficiency and effectiveness of path planning in complex, dynamic environments like vertical farms.

Author Contributions: Conceptualization, J.S., T.S.H., M.K.A.b.M.A., A.b.A., L.F. and R.Z.; methodology, J.S.; software, J.S.; validation, T.S.H., M.K.A.b.M.A. and R.Z.; formal analysis, J.S. and L.F.; investigation, L.F.; resources, J.S., T.S.H., M.K.A.b.M.A. and A.b.A.; data curation, J.S. and L.F.; writing—original draft preparation, J.S.; writing—review and editing, R.Z.; visualization, J.S.; supervision, T.S.H., M.K.A.b.M.A. and A.b.A.; project administration, T.S.H.; funding acquisition, J.S. and T.S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

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

Acknowledgments: The works in the paper were performed at the faculty of engineering, Universiti Putra Malaysia, Selangor 43400, Malaysia, when Jiazheng Shen was a PhD degree student at Universiti Putra Malaysia.

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

References

1. United Nations. Transforming Our World: The 2030 Agenda for Sustainable Development. Available online: https://sdgs.un.org/2030agenda (accessed on 10 May 2021).
2. O'Hara, J.K.; Lin, J. Population Density and Local Food Market Channels. *Appl. Econ. Perspect. Policy* **2020**, *42*, 477–496. [CrossRef]
3. United Nations. 2022 Revision of World Population Prospects. Available online: https://population.un.org/wpp/ (accessed on 10 May 2021).
4. McClements, D.J.; Barrangou, R.; Hill, C.; Kokini, J.L.; Lila, M.A.; Meyer, A.S.; Yu, L. Building a Resilient, Sustainable, and Healthier Food Supply Through Innovation and Technology. *Annu. Rev. Food Sci. Technol.* **2021**, *12*, 1–28. [CrossRef] [PubMed]
5. Van Delden, S.H.; SharathKumar, M.; Butturini, M.; Graamans, L.J.A.; Heuvelink, E.; Kacira, M.; Kaiser, E.; Klamer, R.S.; Klerkx, L.; Kootstra, G.; et al. Current Status and Future Challenges in Implementing and Upscaling Vertical Farming Systems. *Nat. Food* **2021**, *2*, 944–956. [CrossRef] [PubMed]
6. Saad, M.H.M.; Hamdan, N.M.; Sarker, M.R. State of the Art of Urban Smart Vertical Farming Automation System: Advanced Topologies, Issues and Recommendations. *Electronics* **2021**, *10*, 1422. [CrossRef]
7. Halgamuge, M.N.; Bojovschi, A.; Fisher, P.M.; Le, T.C.; Adeloju, S.; Murphy, S. Internet of Things and Autonomous Control for Vertical Cultivation Walls towards Smart Food Growing: A Review. *Urban For. Urban Green.* **2021**, *61*, 127094. [CrossRef]
8. Asiminari, G.; Moysiadis, V.; Kateris, D.; Busato, P.; Wu, C.; Achillas, C.; Sørensen, C.G.; Pearson, S.; Bochtis, D. Integrated Route-Planning System for Agricultural Robots. *AgriEngineering* **2024**, *6*, 657–677. [CrossRef]
9. Zhivkov, T.; Sklar, E.I.; Botting, D.; Pearson, S. 5G on the Farm: Evaluating Wireless Network Capabilities and Needs for Agricultural Robotics. *Machines* **2023**, *11*, 1064. [CrossRef]
10. Wang, N.; Yang, X.; Wang, T.; Xiao, J.; Zhang, M.; Wang, H.; Li, H. Collaborative Path Planning and Task Allocation for Multiple Agricultural Machines. *Comput. Electron. Agric.* **2023**, *213*, 108218. [CrossRef]
11. Kiani, F.; Seyyedabbasi, A.; Nematzadeh, S.; Candan, F.; Çevik, T.; Anka, F.A.; Randazzo, G.; Lanza, S.; Muzirafuti, A. Adaptive Metaheuristic-Based Methods for Autonomous Robot Path Planning: Sustainable Agricultural Applications. *Appl. Sci.* **2022**, *12*, 943. [CrossRef]
12. Guo, H.; Miao, Z.; Ji, J.C.; Pan, Q. An Effective Collaboration Evolutionary Algorithm for Multi-Robot Task Allocation and Scheduling in a Smart Farm. *Knowl. -Based Syst.* **2024**, *289*, 111474. [CrossRef]
13. Eiffert, S.; Wallace, N.D.; Kong, H.; Pirmarzdashti, N.; Sukkarieh, S. Resource and Response Aware Path Planning for Long-Term Autonomy of Ground Robots in Agriculture. *arXiv* **2021**, arXiv:2105.10690. [CrossRef]
14. Ouyang, C.; Qiu, Y.; Zhu, D. Adaptive Spiral Flying Sparrow Search Algorithm. *Sci. Program.* **2021**, *2021*, 6505253. [CrossRef]
15. Pehlivanoglu, Y.V. An Enhanced Genetic Algorithm for Path Planning of Autonomous UAV in Target Coverage Problems. *Appl. Soft Comput.* **2021**, *112*, 107796. [CrossRef]
16. Li, L.; Liu, L.; Shao, Y.; Zhang, X.; Chen, Y.; Guo, C.; Nian, H. Enhancing Swarm Intelligence for Obstacle Avoidance with Multi-Strategy and Improved Dung Beetle Optimization Algorithm in Mobile Robot Navigation. *Electronics* **2023**, *12*, 4462. [CrossRef]
17. Yuan, Q.; Sun, R.; Du, X. Path Planning of Mobile Robots Based on an Improved Particle Swarm Optimization Algorithm. *Processes* **2022**, *11*, 26. [CrossRef]
18. Emmi, L.; Fernández, R.; Gonzalez-de-Santos, P.; Francia, M.; Golfarelli, M.; Vitali, G.; Sandmann, H.; Hustedt, M.; Wollweber, M. Exploiting the Internet Resources for Autonomous Robots in Agriculture. *Agriculture* **2023**, *13*, 1005. [CrossRef]
19. Candra, A.; Budiman, M.A.; Hartanto, K. Dijkstra's and A-Star in Finding the Shortest Path: A Tutorial. In Proceedings of the 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA), Medan, Indonesia, 16–17 July 2020; pp. 28–32.
20. Tian, X.; Afrin, M.; Mistry, S.; Mahmud, R.; Krishna, A.; Li, Y. MURE: Multi-Layer Real-Time Livestock Management Architecture with Unmanned Aerial Vehicles Using Deep Reinforcement Learning. *Future Gener. Comput. Syst.* **2024**, *161*, 454–466. [CrossRef]
21. Varga, B.; Kulcsár, B.; Chehreghani, M.H. Deep Q-Learning: A Robust Control Approach. *Int. J. Robust Nonlinear Control* **2023**, *33*, 526–544. [CrossRef]
22. Lee, W.; Kim, T. Multiagent Reinforcement Learning in Controlling Offloading Ratio and Trajectory for Multi-UAV Mobile-Edge Computing. *IEEE Internet Things J.* **2024**, *11*, 3417–3429. [CrossRef]
23. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]

24. Zafar, M.N.; Mohanta, J.C.; Keshari, A. GWO-Potential Field Method for Mobile Robot Path Planning and Navigation Control. *Arab. J. Sci. Eng.* **2021**, *46*, 8087–8104. [CrossRef]
25. Liu, S.; Liu, S.; Xiao, H. Improved Gray Wolf Optimization Algorithm Integrating A* Algorithm for Path Planning of Mobile Charging Robots. *Robotica* **2024**, *42*, 536–559. [CrossRef]
26. Makhadmeh, S.N.; Al-Betar, M.A.; Doush, I.A.; Awadallah, M.A.; Kassaymeh, S.; Mirjalili, S.; Zitar, R.A. Recent Advances in Grey Wolf Optimizer, Its Versions and Applications: Review. *IEEE Access* **2024**, *12*, 22991–23028. [CrossRef]
27. Liu, X.; Li, G.; Yang, H.; Zhang, N.; Wang, L.; Shao, P. Agricultural UAV Trajectory Planning by Incorporating Multi-Mechanism Improved Grey Wolf Optimization Algorithm. *Expert Syst. Appl.* **2023**, *233*, 120946. [CrossRef]
28. Nadimi-Shahraki, M.H.; Zamani, H.; Asghari Varzaneh, Z.; Sadiq, A.S.; Mirjalili, S. A Systematic Review of Applying Grey Wolf Optimizer, Its Variants, and Its Developments in Different Internet of Things Applications. *Internet Things* **2024**, *26*, 101135. [CrossRef]
29. Deng, W.; Shang, S.; Cai, X.; Zhao, H.; Song, Y.; Xu, J. An Improved Differential Evolution Algorithm and Its Application in Optimization Problem. *Soft Comput.* **2021**, *25*, 5277–5298. [CrossRef]
30. Abdulsaheb, J.A.; Kadhim, D.J. Classical and Heuristic Approaches for Mobile Robot Path Planning: A Survey. *Robotics* **2023**, *12*, 93. [CrossRef]
31. Lim, H.S.; Fan, S.; Chin, C.; Chai, S.; Bose, N. Particle Swarm Optimization Algorithms with Selective Differential Evolution for AUV Path Planning. *Int. J. Robot. Autom. (IJRA)* **2020**, *9*, 94–112. [CrossRef]
32. Mousa, M.H.; Hussein, M.K. Efficient UAV-Based Mobile Edge Computing Using Differential Evolution and Ant Colony Optimization. *PeerJ Comput. Sci.* **2022**, *8*, e870. [CrossRef] [PubMed]
33. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
34. Zhao, Z.; Liu, S.; Wei, J.; Qin, F. Improved Biological Neural Network Approach for Path Planning of Differential Drive Agricultural Robots with Arbitrary Shape. *Comput. Electron. Agric.* **2024**, *216*, 108525. [CrossRef]
35. Chen, X.; Chen, J.; Du, C.; Xu, Y. Region Coverage Path Planning of Multiple Disconnected Convex Polygons Based on Simulated Annealing Algorithm. In Proceedings of the 2021 IEEE 4th International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, 13–15 August 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 238–242.
36. Rozenstein, O.; Cohen, Y.; Alchanatis, V.; Behrendt, K.; Bonfil, D.J.; Eshel, G.; Harari, A.; Harris, W.E.; Klapp, I.; Laor, Y. Data-Driven Agriculture and Sustainable Farming: Friends or Foes? *Precis. Agric.* **2023**, *25*, 520–531. [CrossRef]
37. Liang, Z.; Zeng, J.; Liu, L.; Zhu, Z. A Many-Objective Optimization Algorithm with Mutation Strategy Based on Variable Classification and Elite Individual. *Swarm Evol. Comput.* **2021**, *60*, 100769. [CrossRef]
38. Huo, L.; Zhu, J.; Wu, G.; Li, Z. A Novel Simulated Annealing Based Strategy for Balanced UAV Task Assignment and Path Planning. *Sensors* **2020**, *20*, 4769. [CrossRef] [PubMed]
39. Wang, N.; Zhang, Y.; Ahn, C.K.; Xu, Q. Autonomous Pilot of Unmanned Surface Vehicles: Bridging Path Planning and Tracking. *IEEE Trans. Veh. Technol.* **2021**, *71*, 2358–2374. [CrossRef]