

Article

A Full-Coverage Path Planning Method for an Orchard Mower Based on the Dung Beetle Optimization Algorithm

Lixing Liu ¹ , Xu Wang ¹, Hongjie Liu ^{1,2}, Jianping Li ^{1,2}, Pengfei Wang ^{1,2} and Xin Yang ^{1,2,*}

¹ College of Mechanical and Electrical Engineering, Hebei Agricultural University, Baoding 071000, China; 20211090024@pgs.hebau.edu.cn (L.L.); 20221090026@pgs.hebau.edu.cn (X.W.); liuhj@hebau.edu.cn (H.L.); ljpd527@hebau.edu.cn (J.L.); jdwpf@hebau.edu.cn (P.W.)

² Hebei Province Smart Agriculture Equipment Technology Innovation Center, Baoding 071001, China

* Correspondence: yangxin@hebau.edu.cn; Tel.: +86-1593-345-9619

Abstract: In order to optimize the operating path of orchard mowers and improve their efficiency, we propose an MI-DBO (multi-strategy improved dung beetle optimization algorithm) to solve the problem of full-coverage path planning for mowers in standardized quadrilateral orchard environments. First, we analyzed the operation scenario of lawn mowers in standardized orchards, transformed the full-coverage path planning problem into a TSP (traveling salesman problem), and mathematically modeled the U-turn and T-turn strategies based on the characteristics of lawn mowers in orchards. Furthermore, in order to overcome the issue of uneven distribution of individual positions in the DBO (dung beetle optimization) algorithm and the tendency to fall into local optimal solutions, we incorporated Bernoulli mapping and the convex lens reverse-learning strategy in the initialization stage of DBO to ensure a uniform distribution of the initial population. During the algorithm iteration stage, we incorporated the Levy flight strategy into the position update formulas of breeding beetles, foraging beetles, and stealing beetles in the DBO algorithm, allowing them to escape from local optimal solutions. Simulation experiments show that for 18 types of orchards with different parameters, MI-DBO can find the mowing machine's operation paths. Compared with other common swarm intelligence algorithms, MI-DBO has the shortest average path length of 456.36 m and can ensure faster optimization efficiency. Field experiments indicate that the algorithm-optimized paths do not effectively reduce the mowing machine's missed mowing rate, but the overall missed mowing rate is controlled below 0.8%, allowing for the completion of mowing operations effectively. Compared with other algorithms, MI-DBO has the least time and fuel consumption for operations. Compared to the row-by-row operation method, using paths generated by MI-DBO reduces the operation time by an average of 1193.67 s and the fuel consumption rate by an average of 9.99%. Compared to paths generated by DBO, the operation time is reduced by an average of 314.33 s and the fuel consumption rate by an average of 2.79%.

Keywords: orchard lawn mower; route planning; turning strategy; swarm intelligence algorithm



Citation: Liu, L.; Wang, X.; Liu, H.; Li, J.; Wang, P.; Yang, X. A Full-Coverage Path Planning Method for an Orchard Mower Based on the Dung Beetle Optimization Algorithm. *Agriculture* **2024**, *14*, 865. <https://doi.org/10.3390/agriculture14060865>

Academic Editor: Massimo Cecchini

Received: 9 May 2024

Revised: 23 May 2024

Accepted: 29 May 2024

Published: 30 May 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

With the trend in land transfer, large-scale planting has created conditions for the intelligent development of agricultural machinery [1–3]. Positioning and navigation systems based on GPS (Global Positioning System), Glonass, Galileo, and the BDS (the BeiDou Navigation Satellite System) are increasingly being applied in precision agriculture [4–6]. By using computer technology to pre-plan the operation paths of large-scale farmland with known boundary information, essential path references can be provided for both autonomous and manual agricultural machinery operation [7].

In 2008, Bochtis introduced “B-patterns”, representing agricultural machinery’s coverage of farmland as a directed-graph traversal problem. The optimization for finding the best path can be transformed into a problem of finding the shortest path in the graph [8].

Zhou et al. utilized grid-based methods to establish an environmental map model and employed the inward spiral full-coverage algorithm, resulting in high coverage efficiency and low repetition rates in robotic operations [9]. Chen et al., in terms of the traversal sequence of agricultural machinery, used the SA (simulated annealing) algorithm to obtain an optimal path set. They solved the full-coverage traversal sequence through unit decomposition and synthesis, addressing the issues of poor adaptability of traditional traversal methods and the tendency of classical SA to get stuck in local optimal solutions during large-scale agricultural machinery operations [10]. Jeon et al. developed a comprehensive full-coverage tractor path planning scheme, dividing any quadrilateral into internal and external path planning. The internal path planning uses reciprocating coverage, while the external path planning uses spiral coverage [11]. For regular quadrilaterals, once the agricultural machinery and field parameters are determined, the effective operation path remains fixed. During actual operations, the main challenge lies in determining the operating sequence between different operation rows, i.e., the problem of scheduling and sequencing. Thus, if we consider the operation rows as cities to be traversed and the agricultural machinery as a traveling salesman, this problem can be viewed as a TSP (traveling salesman problem) seeking the shortest path to traverse all cities [12–14]. Yao et al. used SA (simulated annealing) for optimizing agricultural machinery operation paths, and proposed a method based on SADG (Doppler and the greedy strategy simulated-annealing algorithm) to optimize agricultural machinery operation paths from three aspects: rectangular farmland, trapezoidal farmland, and different types of agricultural machinery [15]. The traditional “B-pattern” model is mainly used for fields with equal inter-row spacing (where the operation line refers to the route the agricultural machinery takes from one end of the orchard to the other, such as transplanting and harvesting). When agricultural machinery performs orchard management operations like mowing and trenching, the inter-row spacing is usually uneven. Therefore, the path planning for orchards cannot directly apply “B-patterns” [16]. In 2015, D. Bochtis et al. applied the improved “B-patterns” to orchards, significantly enhancing the efficiency of agricultural machinery operations in orchards [17]. However, there has been no research conducted from the perspective of the TSP regarding orchard environments.

As the number of rows in the orchard operation increases, the combination of operation row sequences grows exponentially. Selecting a path with a shorter total turning distance becomes increasingly difficult. The TSP belongs to the group of classic combinatorial optimization problems, and currently there is no perfect algorithm that can obtain an exact solution. Researchers have adopted a series of swarm intelligence algorithms to solve this problem, such as GA (the genetic algorithm) [18], PSO (particle swarm optimization) [19], ACO (ant colony optimization) [20], and SA [21].

Inspired by the dung beetle’s rolling, dancing, foraging, breeding, and stealing behaviors, the DBO (dung beetle optimizer algorithm) was proposed [22]. DBO fully simulates various physical relationships among members within the dung beetle population, including the attraction between individuals and targets, as well as the close-range repulsion and long-range attraction between individuals. DBO simulates the flight behavior of dung beetles by continuously updating the position of each dung beetle, maintaining the overall structure of the population, and driving individuals towards better solutions. Compared to other swarm intelligence algorithms, DBO more realistically reflects the dynamic processes within the dung beetle population and converges faster. However, like other swarm intelligence optimization algorithms, DBO also suffers from imbalances in global exploration and local exploitation capabilities, leading to the risk of local optima and weak global exploration abilities.

In summary, this article categorizes the path planning problem of orchard lawn mowers as a TSP, and constructs a mathematical model. In response to the disadvantage of weak global exploration ability caused by uneven initial solution distribution in traditional DBO, Bernoulli mapping and convex lens reverse-learning strategy are introduced in the initialization stage. Subsequently, in the search phase, in order to prevent the algorithm

from falling into local optima, we introduce the Levy flight strategy. We use MI-DBO to solve the problem of full-coverage path planning for orchard lawn mowers. To verify the effectiveness of MI-DBO, we compared it with other swarm intelligence algorithms. The remaining sections of this paper are organized as follows: Section 2 constructs a mathematical turning model for grass-cutting machines in a novel orchard environment; Section 3 introduces the materials and methods used in this article; Section 4 compares MI-DBO with other algorithms and analyzes the results; Section 5 concludes the paper and outlines our future research plans.

2. Problem Description

2.1. Characteristics and Path Requirements of Orchard Mowing

The boundaries of modern orchards are predominantly in the shape of quadrangles, with row spacing generally set at 3.5 to 4 m. Considering that ground-fabric mulching, with a width of 1.5 to 2 m, is commonly laid during orchard establishment to conserve soil moisture and temperature rise, and that mowing is unnecessary in areas with mulching materials, therefore, such areas are designated as non-working zones in our model construction. The width of natural or artificial grass between rows is approximately 1.5 to 2 m; such areas, which need to be mown, are defined as the working areas. Common orchard lawn mowers are predominantly four-wheeled self-propelled, equipped with the reverse function, and the cutting width is usually between 1 and 1.2 m. Therefore, the mower can complete the inter-row mowing task by traversing each fruit-tree row twice. Figure 1 illustrates the standardized orchard environment intended for the optimization operation proposed in this paper. An ideal working path should enable the mower to operate in all working areas with minimum costs (operation time and fuel (electricity) consumption). The non-working paths, primarily in the field edge areas, are the main factors that affect the mower's efficiency. Consequently, this study neglects the length of inter-row paths and focuses on optimizing the final working path of the mower by reducing the turning path in the field edge. Currently, there are three commonly used turning methods in production, namely U-turn, T-turn, and Ω -turn. For mowers, an efficient turning strategy should possess the following three fundamental characteristics:

1. Distance Feature: minimize the length of turning path to reduce turning consumption (time, fuel) as much as possible;
2. Field Edge Space Feature: occupy the smallest field edge turning space to reduce preparation work for the field edge area;
3. Reverse Feature: compatible with the machine's reversing capability; for example, T-turn requires the mower to reverse.

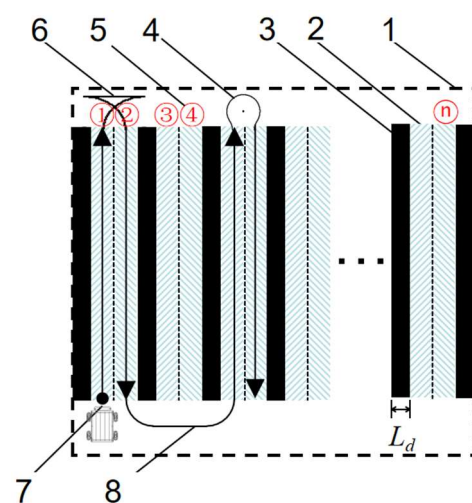


Figure 1. Standardized orchard environment: 1. Orchard boundary. 2. Working area. 3. Non-working area. 4. Ω -turn. 5. Numbers of working rows. 6. T-turn. 7. Starting point of operation. 8. U-turn.

In conclusion, for quadrilateral orchards and four-wheeled self-propelled mowers, two turning modes of U-turn and T-turn (as shown in Figure 2) are focused on in this study.

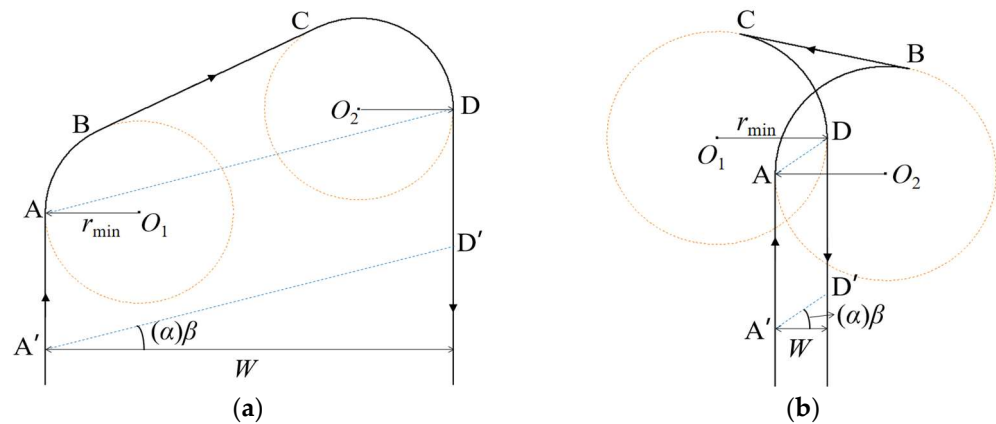


Figure 2. Turning strategies: (a) U-turn; (b) T-turn.

2.2. Mathematical Model

By taking the total length of working paths turning between connected rows as the optimized objective function and the arrangement order of working paths in all rows as the independent variable, a mathematical model describing this problem can be established. The mower needs to move from row i to row j after completing its operation, and the turning distances for U-turn and T-turn can be obtained from Figure 2:

$$U(d_{ij}) = \pi r_{\min} + 2 \left\{ \frac{(w \cdot d_{ij} + m \cdot L_d)^2}{4} \left[\tan^2\left(\theta - \frac{\pi}{2}\right) + 1 \right] - r_{\min} \cdot (w \cdot d_{ij} + m \cdot L_d) + r_{\min}^2 \right\}^{\frac{1}{2}} \quad (1)$$

$$T(d_{ij}) = \pi r_{\min} + 2 \left\{ \frac{(w \cdot d_{ij} + m \cdot L_d)^2}{4 \cdot \sin^2 \theta} - r_{\min} \cdot (w \cdot d_{ij} + m \cdot L_d) + r_{\min}^2 \right\}^{\frac{1}{2}} \quad (2)$$

In the equation, $U(d_{ij})$ represents the distance for the U-turn, $T(d_{ij})$ represents the distance for the T-turn, i represents the row number where the mower completes its operation, j represents the row number where the mower is about to enter, d_{ij} represents the difference in working rows before and after the mower's turn ($d_{ij} = |j - i|$, where $i \neq j$), $r_{\min}$ represents the minimum turning radius of the mower, w represents the mower's working width, L_d represents the width of ground-fabric mulching, θ represents the working angle α or β , and m represents the number of non-working areas crossed by the mower before and after the turn. The illustration is as follows:

$$m = |e - t| \quad (3)$$

$$e = \begin{cases} \frac{i}{2}, i = 2n \\ \frac{i+1}{2}, i = 2n+1 \end{cases}, n \in \mathbb{Z} \quad (4)$$

$$t = \begin{cases} \frac{j}{2}, j = 2n \\ \frac{j+1}{2}, j = 2n+1 \end{cases}, n \in \mathbb{Z} \quad (5)$$

The method for determining the type of turning in the field edge:

$$L_{\min}(d_{ij}) = \begin{cases} U(d_{ij}), |d_{ij}| \geq 2r_{\min}/w, i \neq j \\ T(d_{ij}), |d_{ij}| < 2r_{\min}/w, i \neq j \end{cases} \quad (6)$$

If n represents the total number of working rows, a represents the starting row number, b represents the ending row number, N represents the set $\{1, 2, 3, \dots, n\}$, the decision variable $x_{ij} = 1$ represents the inclusion of a turning path from row i to row j in the turn path, and $x_{ij} = 0$ represents the exclusion of such a turning path in the turn path, and $|S|$ represents the number of elements in the set S , then the optimization model can be expressed as follows:

$$\min \sum_{i \neq j} L_{\min}(d_{ij}) \cdot x_{ij} \quad (7)$$

$$\sum_{j=1}^n x_{ij} = \begin{cases} 1, & i \in N, i \neq b \\ 0, & i \in N, i = b \end{cases} \quad (8)$$

$$\sum_{i=1}^n x_{ij} = \begin{cases} 1, & i \in N, j \neq a \\ 0, & i \in N, j = a \end{cases} \quad (9)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, 2 \leq |S| \leq n - 1, S \subset N \quad (10)$$

$$x_{ij} \in \{1, 0\}, i, j \in N, i \neq j \quad (11)$$

In the model, Equation (7) represents the optimization objective function, ensuring that the total length of turning paths is minimized, and Equations (8)–(11) ensure that each working row is traversed once and does not eventually return to the starting row.

3. Materials and Methods

3.1. Dung Beetle Optimization Algorithm

Based on the dung beetle's rolling, dancing, breeding, foraging, and stealing behaviors, the DBO (Dung beetle optimization) algorithm incorporates five different update rules. Each dung beetle group consists of five different types of agent dung beetles, namely, rolling dung beetles, dancing dung beetles, breeding dung beetles, foraging dung beetles, and stealing dung beetles. Rolling dung beetles refers to the behavior of dung beetles rolling feces into a ball and rolling it to a secure location for storage. Dung beetles can navigate by using celestial cues such as the Sun, Moon, and polarized light to roll the dung ball in a straight line. When the environment changes, the position of the dung beetle changes accordingly, and the rolling behavior can be represented as

$$x_i(t+1) = x_i(t) + \sigma \cdot k \cdot x_i(t-1) + q \cdot \Delta x \quad (12)$$

$$\Delta x = |x_i(t) - X^W| \quad (13)$$

In the equation, t represents the current iteration number, and $x_i(t)$ represents the position of the i -th dung beetle in the t -th iteration. $k \in (0, 0.2]$ is a constant, representing the deviation coefficient, and $q \in (0, 1]$ is a random number. $\sigma = \pm 1$, where 1 indicates no deviation and -1 indicates deviation from the original direction. In this paper, to simulate the complex environment of the real world, a probability method is used to set it as 1 or -1 . Δx represents the change in the environment. X^W represents the worst position in the current population.

When a dung beetle encounters an obstacle and cannot move forward, it will regain a new direction through dancing to find a new path. The position update at this time is determined by the tangent function, and the formula is as follows:

$$x_i(t+1) = x_i(t) + \tan(\theta) |x_i(t) - x_i(t-1)| \quad (14)$$

In the equation, $\tan(\theta)$ represents the deviation angle. Since the tangent function is a periodic function, only the values of the tangent function defined in the interval $[0, \pi]$ need to be considered. $|x_i(t) - x_i(t-1)|$ represents the difference between the position of the i -th dung beetle in the t -th iteration and its position in the $t-1$ iteration. Therefore, the

position update of the dancing dung beetle is closely related to both current and historical information. It is important to note that when $\theta = 0, \pi/2$, or π , $\tan(\theta)$ is meaningless, and the dung beetle's position remains unchanged.

In nature, dung beetles roll dung balls to a safe area to provide a secure environment for their offspring. The boundary selection strategy for this area is

$$\begin{cases} Lb^* = \max(X^* \cdot (1 - R), Lb) \\ Ub^* = \min(X^* \cdot (1 + R), Ub) \end{cases} \quad (15)$$

$$R = 1 - t/T_{\max} \quad (16)$$

In the equation, X^* represents the current population's best position. Lb represents the lower bound of the search space, and Ub represents the upper bound of the search space. Lb^* and Ub^* represent the lower and upper bounds of the oviposition area, respectively. R represents the inertia weight. $T_{\max}$ is the maximum number of iterations.

Once a safe position is found, female dung beetles will choose this area to lay their breeding balls. In the DBO, each female dung beetle will only generate one breeding ball in each iteration. The dynamic updating process of the breeding ball's position iteration is

$$x_i(t+1) = X^* + b_1 \cdot (x_i(t) - Lb^*) + b_2 \cdot (x_i(t) - Ub^*) \quad (17)$$

In the equation, $x_i(t)$ represents the position of the i -th dung beetle in the t -th iteration, and b_1 and b_2 are two independent random vectors of size $1 \times D$, where D is the dimension of the optimization problem. The position of the breeding ball is strictly confined within a certain range, that is, the oviposition area.

After the breeding ball successfully hatches, it will turn into a young dung beetle and will search for food. Therefore, it is necessary to establish an optimal foraging area to guide the dung beetles in their search for food. The foraging area also utilizes a dynamic boundary strategy, which is

$$\begin{cases} Lb^b = \max(X^b \cdot (1 - R), Lb) \\ Ub^b = \min(X^b \cdot (1 + R), Ub) \end{cases} \quad (18)$$

In the equation, X^b represents the global optimal position, and Lb^b and Ub^b , respectively, represent the lower and upper bounds of the best foraging area. Once this area is determined, the position update method for foraging dung beetles is defined as

$$x_i(t+1) = x_i(t) + C_1(x_i(t) - Lb^b) + C_2(x_i(t) - Ub^b) \quad (19)$$

In the equation, $x_i(t)$ represents the position of the i -th foraging dung beetle in the t -th iteration. C_1 is a random number following a normal distribution, and C_2 is a random number belonging to the interval (0,1).

Within the population, there are dung beetles that steal dung balls from other dung beetles. The position update formula for the stealing dung beetles is as follows:

$$x_i(t+1) = X^b + F \cdot g \cdot (|x_i(t) - X^*| + |x_i(t) - X^b|) \quad (20)$$

In the equation, $x_i(t)$ represents the position of the i -th stealing dung beetle in the t -th iteration. g is a random vector of size $1 \times D$ following a normal distribution, and F is a constant.

3.2. Multi-Strategy Improved DBO

As an intelligent optimization algorithm, DBO is a stochastic search algorithm that essentially generates an initial population through random initialization. However, in this process, it is common to encounter an uneven distribution of dung beetles' individual positions, a weak global exploration capability, a low population diversity, and the tendency to

get stuck in local optima. To address these issues, chaotic mapping is used in the initialization stage of the DBO population to generate a highly diversified initial population. Chaotic mapping is used to generate chaotic sequences, which are random sequences produced by simple deterministic systems, possessing characteristics such as randomness, ergodicity, and regularity [23]. Introducing chaotic mapping during population initialization allows individuals to make the most of the information in the solution space, thereby improving global search capability. Common chaotic mappings include Logistic, Cubic, Bernoulli, and Singer, among others. We have compiled 9 types of chaotic mappings, and Figure 3 shows the histograms of 105 sequence values generated by these chaotic mappings within the range of 0 to 1.

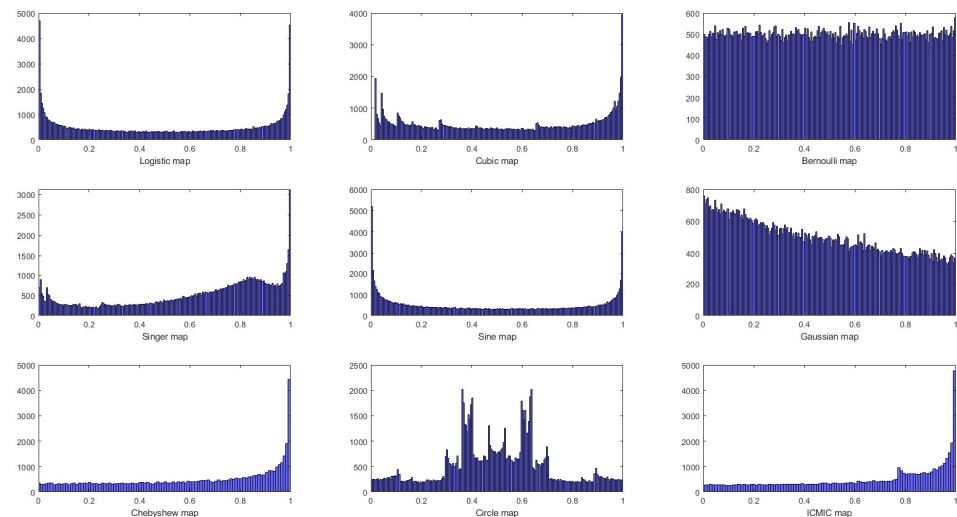


Figure 3. Comparison of chaotic mapping for population initialization.

From Figure 3, it can be observed that the sequence values generated by the Bernoulli mapping are more uniform compared to other methods, which helps expand the search range of the dung beetle population in spatial terms, increase the diversity of population positions, and to some extent, alleviate the algorithm's tendency to fall into local optima. Therefore, we choose the Bernoulli mapping to generate the initial population, with its mathematical model defined as

$$x_{i+1} = \begin{cases} x_i / (1 - \rho), & x_i \in (0, 1 - \rho) \\ (x_i - 1 + \rho) / \rho, & x_i \in (1 - \rho, 1) \end{cases} \quad (21)$$

In the equation, x_i represents the current value of the chaotic sequence generated in the i -th generation, and ρ is the control parameter. Through preliminary experiments, we found that when $\rho = 0.5$, Bernoulli can achieve better traversal.

The initial population formed through the Bernoulli mapping is relatively uniform. In order to enhance the global exploration capability of the DBO algorithm, we perform lens-imaging reverse learning on the population after chaotic mapping to generate the reverse population of the chaotic mapping population. We select superior individuals from the chaotic mapping population and the reverse population as the initial population of the algorithm, thereby improving the quality of the initial solutions and increasing the likelihood of DBO finding the optimal solution. The principle of the lens reverse-learning strategy is shown in Figure 4 [24].

The equation for convex lens imaging is

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \quad (22)$$

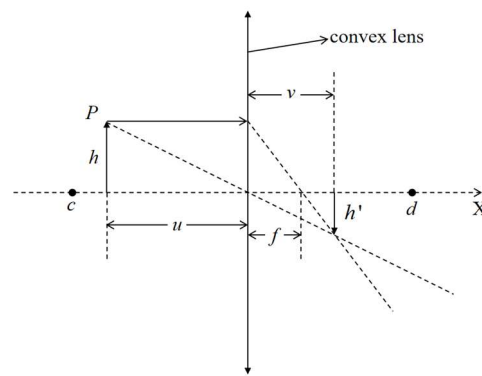


Figure 4. Schematic diagram of reverse-learning strategy based on convex lens imaging.

In the equation, u is the object distance, v is the image distance, and f is the focal length of the convex lens. Combining this with Figure 4, it can be understood that

$$\frac{(c+d)/2 - x}{x' - (c+d)/2} = \frac{h}{h'} \quad (23)$$

In the equation, (c, d) represents the range of the X-axis, the projection of object P on the X-axis is x , and h is the height of the object. After convex lens imaging, the projection on the X-axis becomes x' , and the height of the object becomes h' .

Let $K = h/h'$, then we have

$$x' = \frac{c+d}{2} + \frac{c+d}{2K} - \frac{x}{K} \quad (24)$$

From this, the model of convex lens reverse-learning strategy can be derived as

$$x_{n+1} = \frac{c_j + d_j}{2} + \frac{c_j + d_j}{2K} - \frac{x_n}{K} \quad (25)$$

In the equation, c_j and d_j are the upper and lower bounds of the j -th dimension, and K is the scaling factor of the lens.

In addition to optimizing the initialization stage, it is evident from Equations (17), (19) and (20) that the position updates of breeding dung beetles, foraging dung beetles, and stealing dung beetles are closely related to the current individual optimal values. This characteristic can lead to premature convergence and becoming trapped in local optima during the iterative process. To address this, we apply the Levy flight [25] strategy to the position updates of breeding dung beetles, foraging dung beetles, and stealing dung beetles. The Levy flight strategy is a probability distribution with a heavy-tailed characteristic. By introducing the step generated by the Levy flight, dung beetles can randomly roam in the search space, enhancing the algorithm's exploration in space. That is, after the original position update, the Levy flight strategy is applied again to update the individual positions, ensuring diversity in the dung beetle population in the later stages of the algorithm, breaking out of local optima, and expanding search capability. The Levy flight strategy is as follows:

$$x_i(t+1) = x_i(t) \cdot \text{Levy}(\lambda) \quad (26)$$

In the equation, $\text{Levy}(\lambda)$ represents the Levy distribution with a parameter of λ . We define it as

$$\text{Levy}(\lambda) = \frac{\gamma \cdot E}{|F|^{0.5}} \quad (27)$$

$$\gamma = \frac{\Gamma(1+\lambda) \cdot \sin(\pi\lambda/2)}{\Gamma((1+\lambda)/2) \cdot \lambda \cdot 2^{((\lambda-1)/2)}} \quad (28)$$

In the equation, E and F follow normal distributions, and $\Gamma(x) = (x-1)!$. Through preliminary experiments, we found that the algorithm converges faster when $\lambda = 1.5$.

The updated position-updating method for breeding dung beetles, foraging dung beetles, and stealing dung beetles is as follows:

$$x_i(t+1) = \text{Levy}(\lambda) \cdot (X^* + b_1 \cdot (x_i(t) - Lb^*) + b_2 \cdot (x_i(t) - Ub^*)) \quad (29)$$

$$x_i(t+1) = \text{Levy}(\lambda) \cdot (x_i(t) + C_1(x_i(t) - Lb^b) + C_2(x_i(t) - Ub^b)) \quad (30)$$

$$x_i(t+1) = \text{Levy}(\lambda) \cdot (X^b + F \cdot g \cdot (|x_i(t) - X^*| + |x_i(t) - X^b|)) \quad (31)$$

In conclusion, we refer to the optimized algorithm as MI-DBO. The pseudocode for MI-DBO is as follows (Algorithm 1):

Algorithm 1. MI-DBO.

```

Initialize population and parameters
num_dung_beetle = N
MaxIter = T
minBounds = Lb
maxBounds = Ub
Initialize population position utilization Equations (21) and (25)
Initialize global fitness values Fitness utilization Equation (7)
While ( $t < T$ ) do
  For  $i = 1:N$  do
    If  $i \in \text{rolling\_dung\_beetle}$  then
      If  $\tau < 0.9$  then      #  $\tau \in (0,1)$ 
        Update the location of dung beetle  $X(i)$  utilization Equation (12)
        Update fitness values  $\text{NewFit}[i]$  utilization Equation (7)
      Else
        Update the location of dung beetle  $X(i)$  utilization Equation (14)
        Update fitness values  $\text{NewFit}[i]$  utilization Equation (7)
      End If
    End If
    If  $i \in \text{breeding\_dung\_beetle}$  then
      Update the location of dung beetle  $X(i)$  utilization Equation (29)
      Update fitness values  $\text{NewFit}[i]$  utilization Equation (7)
    End If
    If  $i \in \text{foraging\_dung\_beetle}$  then
      Update the location of dung beetle  $X(i)$  utilization Equation (30)
      Update fitness values  $\text{NewFit}[i]$  utilization Equation (7)
    End If
    If  $i \in \text{stealing\_dung\_beetle}$  then
      Update the location of dung beetle  $X(i)$  utilization Equation (31)
      Update fitness values  $\text{NewFit}[i]$  utilization Equation (7)
    End If
    Greedy choice
    If  $\text{NewFit}[i] < \text{Fitness}$  then
       $\text{Fit}[i] = \text{NewFit}[i]$ 
    End If
  End For
   $t = t+1$ 
End While
Output the global optimal position  $\text{BestPosition}$  and its fitness value  $\text{BestFitness}$ 

```

3.3. Computing Environment

To compare the optimization effects and patterns of job scheduling, we conducted comparative experiments on path optimization using traditional GA, PSO, SA, ACO, SA, and DBO, as well as SADG proposed in reference [15], and the MI-DBO proposed in this paper. We mainly compare these from two aspects: optimization performance and search

efficiency. The comparison of optimization performance is based on the reduction rate of the optimization objective value, denoted as DR, expressed as

$$DR = \frac{(\bar{X} - \bar{Y})}{\bar{X}} \times 100\% \quad (32)$$

In the equation, $\bar{X}$ represents the average job path optimization value of classical algorithms. It is worth noting that the classical algorithm for SADG is SA, and for MI-DBO, the classical algorithm is DBO. $\bar{Y}$ represents the average job path optimization value for SADG and MI-DBO.

To determine the optimal parameter combinations for each algorithm, we first conducted preliminary experiments. The optimization objective of the preliminary experiments was a unimodal benchmark function. Each algorithm ran independently, and the optimal parameter combinations were recorded. In the end, we determined the following parameters: for GA, the population size is 200, crossover probability is 0.8, and mutation probability is 0.1; for PSO, the individual learning factor is 0.2, the social learning factor is 0.25, the inertia factor is 5, and the number of particles is 500; for SA and SADG, the initial temperature is 100 °C, and the cooling coefficient is 0.95; for ACO, the number of ants is 200, the importance factor of pheromones is 0.2, the importance factor of the heuristic function is 2.5, the evaporation factor of pheromones is 0.25, the minimum pheromone concentration is 0.1, the maximum pheromone concentration is 0.3, and the turning heuristic factor is 0.4; for DBO and MI-DBO, the population size is 90, and the ratio of rolling dung beetles, breeding dung beetles, foraging dung beetles, and stealing dung beetles is 6:6:7:11. All algorithms were run for 1000 iterations.

The above algorithms are implemented in MATLAB R2022a programming on an Intel Core i7—12700H CPU, 2.30 GHz frequency, 32 GB memory, in Windows 11 operating system environment. Considering the characteristics of Chinese orchards and lawn mowers, the parameters for the lawn mower and orchard are selected as follows: $w = 1$ m, $r_{\min} = 2$ m, and $L_d = 1.8$ m. The number of operating rows chosen are 10 rows, 18 rows, 40 rows, 56 rows, 86 rows, and 110 rows. The operating angle combinations are $\alpha = \beta = 90^\circ$, $\alpha = 90^\circ$ and $\beta = 45^\circ$, and $\alpha = 120^\circ$ and $\beta = 60^\circ$ for 18 types of orchards (as shown in Figure 5), with the optimization of the operating paths for each type of orchard averaged over 10 runs.

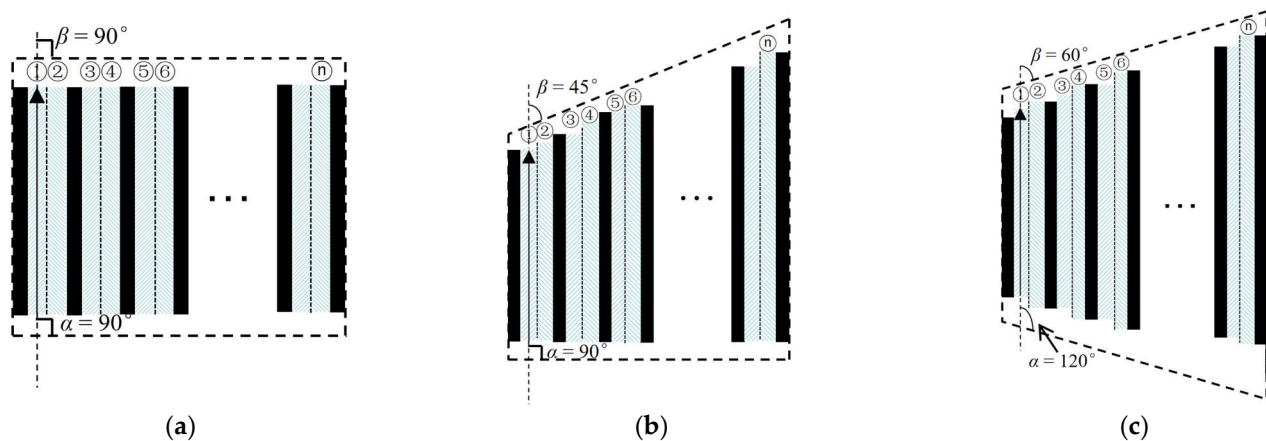


Figure 5. Illustration of orchard angle variation: (a) $\alpha = \beta = 90^\circ$; (b) $\alpha = 90^\circ$ and $\beta = 45^\circ$; (c) $\alpha = 120^\circ$ and $\beta = 60^\circ$.

3.4. Orchard Environment

To further verify the practicality of MI-DBO in optimizing the lawn mower operation path, we conducted comparative field experiments using SADG, DBO, MI-DBO, and operational row-by-row-based methods.

Experimental Locations: the tapered cherry-orchard area in the Shunnong Fruit Modern Agricultural Park, Shunping County, Baoding City, Hebei Province (referred to as

Orchard 1), the tapered apple-orchard area in Shunnong Fruit Modern Agricultural Park, Shunping County, Baoding City, Hebei Province (referred to as Orchard 2), and the rectangular apple-orchard area in Fruit Tree Research Institute of Shijiazhuang City, Hebei Province (referred to as Orchard 3). The actual plot boundaries were obtained from Google Earth, as shown in Figures 6–8. The parameters of each orchard area were obtained through on-site measurements, as shown in Table 1.



Figure 6. Park No. 1.

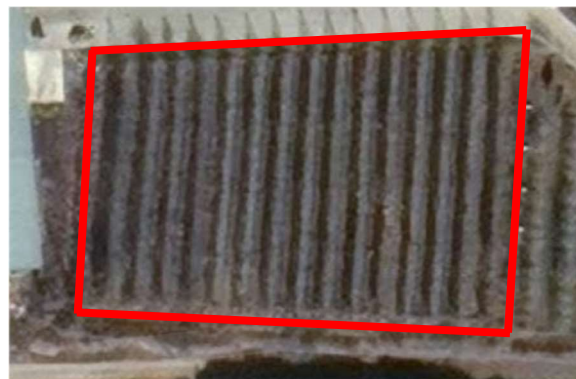


Figure 7. Park No. 2.



Figure 8. Park No. 3. The red square represents the shape of the orchard.

Table 1. Parameter table of each park.

Park Number	Fruit Tree Line	Mowing Line	L_d/m	$\alpha/^\circ$	$\beta/^\circ$	Total Area of Working Area/ m^2
1	6	12	2	90	70	335.54
2	15	30	2	90	80	796.13
3	27	54	2	90	90	4050.00

Test Equipment: G301 mower produced by Wuxi Kalman Navigation Technology Co., LTD. (Wuxi, China). The parameters of G301 are $w = 1.2$ m, $r_{\min} = 2$ m.

Evaluation Indicators: since the test process is affected by many factors such as equipment performance, orchard road conditions and signal quality, the actual traveling track of the lawn mower has a large deviation from the theoretical calculation, so the path length in this section is not used as an index to evaluate the operation efficiency. In addition to the path length, the total operating time (T), mower fuel consumption rate (G) and missed cutting rate (M) are also intuitive efficiency evaluation indicators. Among them, T is timed by a timer, and G and M are calculated as follows:

$$G = \frac{H_1 - H_2}{H_3} \times 100\% \quad (33)$$

$$M = \frac{\sum_{i=1}^n N_i}{N_y} \times 100\% \quad (34)$$

In the equation, H_1 is the height of the mower's fuel tank before the test; H_2 is the height of the mower's fuel tank after the test; H_3 is the height of the mower's fuel tank when it is full of fuel, which is measured as 0.32 m; N_i is the missed cut area of job line i (we estimate N_i by scanning it using the "Area Measurement" function of the "Pocket Recognition King V4.11.4.0" software); and N_y is the total area of the work area. It should be noted that in order to collect the missed cut rate of each algorithm, different cutter heights were used for different algorithm tests in the same park.

Experimental Process: (1) measure and record H_1 ; (2) set the G301 lawn mower to mow the orchard in row-by-row operation order; (3) measure and record the missed mowing areas and H_2 ; (4) repeat steps (1)–(3), sequentially conducting experiments on the operation paths generated by SADG, DBO, and MI-DBO. The experimental site is shown in Figure 9.



Figure 9. Orchard testing site: 1. Job line. 2. G301. 3. Job line number. 4. Ground cloth. The red line represents the dividing line of the homework line.

Experimental Time: the experiments in Park 1 and Park 2 were conducted on 22 March 2024, while the experiment in Park 3 was conducted on 25 March 2024.

4. Results and Discussion

4.1. Optimization Performance Test Results and Discussion

The results of the optimization performance test are shown in Table 2.

Table 2. Optimization results of mower's working path.

n	α	β	GA/m	PSO/m	ACO/m	SA/m	SADG/m	DR/%	DBO/m	MI-DBO/m	DR/%
10	90°	90°	81.88	81.88	81.88	81.88	77.70	5.11	81.88	77.70	5.11
	90°	45°	87.88	87.88	89.68	88.41	87.88	0.60	89.68	87.88	2.00
	120°	60°	86.60	86.60	87.95	86.88	86.60	0	87.95	79.08	10.09
18	90°	90°	181.69	173.81	168.09	175.16	152.92	12.70	153.04	146.41	4.33
	90°	45°	191.66	198.60	187.21	193.85	161.04	16.93	173.86	156.30	10.10
	120°	60°	188.98	176.64	170.33	182.83	160.49	12.22	163.08	149.12	8.56
40	90°	90°	430.12	472.81	391.23	415.52	351.02	15.52	376.85	322.76	14.35
	90°	45°	442.74	472.81	437.21	452.43	373.21	17.51	397.25	349.08	12.13
	120°	60°	485.52	491.45	444.13	467.40	383.12	18.03	392.65	344.15	12.35
56	90°	90°	755.64	770.00	662.09	734.64	529.02	27.99	600.52	472.76	21.27
	90°	45°	806.56	820.01	700.61	771.53	557.40	27.75	626.52	518.79	17.19
	120°	60°	760.63	788.61	699.52	749.63	531.89	29.05	687.13	507.43	26.15
86	90°	90°	1304.67	1259.72	1072.78	1196.08	880.44	26.39	901.11	700.42	22.27
	90°	45°	1602.92	1431.54	1311.43	1596.61	941.50	41.03	1221.78	730.40	40.25
	120°	60°	1667.08	1999.76	1261.93	1416.79	932.57	34.18	1005.66	744.51	25.99
110	90°	90°	1951.51	1850.21	1632.30	1684.88	1007.88	40.18	1410.60	936.51	33.61
	90°	45°	2114.84	2118.23	2117.56	2202.09	1116.39	49.30	1619.72	936.51	42.18
	120°	60°	2100.16	2163.22	2079.41	2102.21	1105.35	47.42	2767.38	954.64	65.50
	90°	90°	81.88	81.88	81.88	81.88	77.70	5.11	708.70	456.36	20.75
Average			846.73	857.99	755.30	811.05	524.25	23.44	81.88	77.70	5.11

According to the analysis of Table 1, for the 18 types of orchards with different parameters, GA, PSO, ACO, SA, SADG, DBO, and MI-DBO can all find the operation paths for the lawn mower. GA, PSO, ACO, SA, and DBO do not have the ability to jump out of local optima, resulting in their paths always being longer than those in SADG and MI-DBO. Among them, SADG and MI-DBO have improved the performance of finding the optimal path based on traditional algorithms. SADG is optimized by SA, while MI-DBO is optimized by DBO. As the performance of DBO improves with the increase in job rows, the performance of MI-DBO also improves with the increase in job rows. The variation in orchard operation angles does not affect the algorithms' ability to find the optimal path. This conclusion differs from that of Reference [15] because this study did not divide the orchard into rectangles and trapezoids when calculating the turning distance, but instead provided a formula applicable to all quadrilateral orchards. The variation in the number of rows for the operation is an important factor affecting the algorithms' optimization ability. The more rows for operation, the more obvious the optimization effect, and the reduction rate of the optimization objective increases with the increase in the number of rows for operation. When all other parameters are the same, MI-DBO always has the shortest planned path.

4.2. Optimization Efficiency Test Results and Discussion

Three types of orchards with operation parameters $n = 40$ and $\alpha = \beta = 90^\circ$, $n = 56$ and $\alpha = 90^\circ$ and $\beta = 45^\circ$, and $n = 86$ and $\alpha = 120^\circ$ and $\beta = 60^\circ$ were selected from Table 1. A comparison of the path optimization efficiency of GA, PSO, ACO, SA, SADG, DBO, and MI-DBO was conducted (each algorithm was run 10 times and the average value was taken), and the comparison results are shown in Figures 10–12.

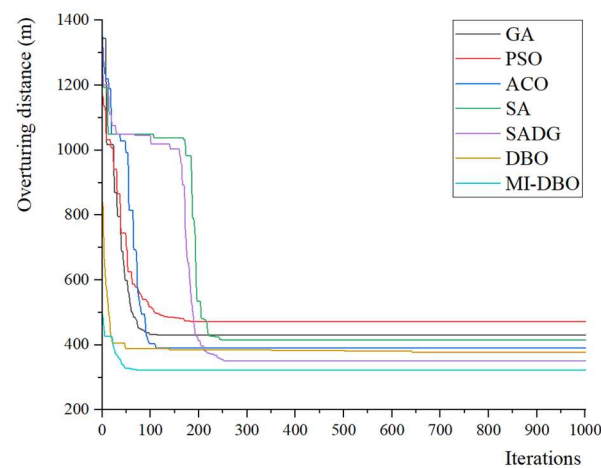


Figure 10. Efficiency comparison of path optimization for $n = 40$ and $\alpha = \beta = 90^\circ$.

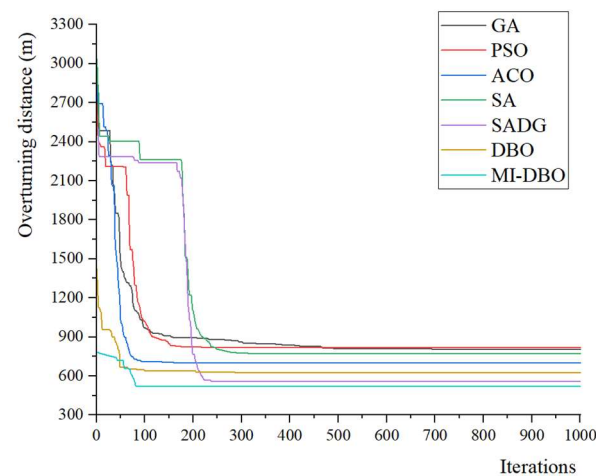


Figure 11. Efficiency comparison of path optimization for $n = 56$ and $\alpha = 90^\circ$ and $\beta = 45^\circ$.

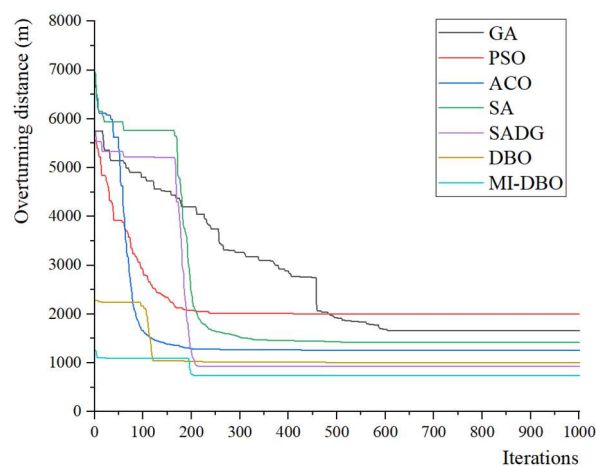


Figure 12. Efficiency comparison of path optimization for $n = 86$ and $\alpha = 120^\circ$ and $\beta = 60^\circ$.

Based on Figures 10–12, it can be seen that GA, PSO, ACO, SA, SADG, DBO, and MI-DBO can all find the optimal path within 1000 iterations, and tend to stabilize. The performance of GA is unstable. In the first two experiments, GA had a fast convergence speed in the early stage. However, in the third experiment, the convergence speed of GA became very slow. However, PSO and ACO have lower overall optimization efficiency

due to uneven population distribution during the initialization stage, although their convergence speed is slower. SA and SADG showed relatively stable performance in three trials. Their common flaw is that the algorithm converges slowly in the early stages. The characteristic of MI-DBO and DBO is their high optimization efficiency. In the three types of orchards with different parameters, MI-DBO always finds a shorter path and ensures faster optimization efficiency. From the convergence curve of MI-DBO, it can be seen that the algorithm optimized using chaotic mapping and that the concave lens reverse-learning strategy can generate better fitness values in the first iteration. In the subsequent iterations, the Levy flight strategy allows the algorithm to escape local optima, resulting in better final results than those of the DBO algorithm.

4.3. Orchard Test Results and Discussion

The results of the orchard test are shown in Table 3.

Table 3. Results of field experiment.

Park Number	Method	Working Line Sequence	T/s	G/%	M/%
1	Row-by-row	1→2→3→4→5→6→7→8→9→10→11→12	1093	8.30	0.36
	SADG	6→4→2→1→3→5→7→9→12→10→11→8	642	4.88	0.69
	DBO	2→5→8→11→12→10→7→9→6→3→1→4	694	5.27	0.57
	MI-DBO	3→1→2→4→6→8→11→9→12→10→7→5	558	4.46	0.56
2	Row-by-row	1→2→3→4→5→6→7→8→9→10→11→12→13→14→15→16→17→18→19→20→21→22→23→24→25→26→27→28→29→30	3496	26.56	0.38
	SADG	23→26→29→27→30→28→25→24→21→18→15→17→20→22→19→16→13→11→9→7→5→2→4→1→3→6→8→10→12→14	2711	19.21	0.50
	DBO	22→24→27→25→28→30→29→26→23→20→17→14→11→13→16→19→21→18→15→10→8→6→4→2→1→3→5→7→9→12	2798	20.16	0.72
	MI-DBO	29→26→24→22→20→18→16→14→12→10→8→6→3→1→4→2→5→7→9→11→13→15→17→19→21→23→25→28→30→27	2596	17.67	0.64
3	Row-by-row	1→2→3→4→5→6→7→8→9→10→11→12→13→14→15→16→17→18→19→20→22→23→24→25→26→27→28→29→30→31→32→33→34→35→36→37→38→39→40→41→42→43→44→45→46→47→48→49→50→51→52→53→54	8064	61.26	0.41
	SADG	35→32→29→26→23→24→27→30→33→31→28→25→22→20→19→17→14→16→13→11→9→7→5→2→4→1→3→6→8→10→12→15→18→21→34→36→38→40→42→44→46→48→50→53→51→54→52→49→47→45→43→41→39→37	6332	48.69	0.57
	DBO	4→2→5→7→9→11→17→31→34→37→38→40→42→44→46→48→50→53→51→54→52→49→47→45→43→41→39→36→33→35→32→29→26→23→20→15→12→14→16→18→21→24→27→30→28→25→22→19→13→10→8→6→3→1	6523	49.11	0.72
	MI-DBO	36→34→32→30→28→26→24→22→20→18→16→14→12→10→8→6→3→1→4→2→5→7→9→11→13→15→17→19→21→23→25→27→29→31→33→35→37→39→41→43→45→47→49→52→54→51→53→50→48→46→44→42→40→38	5918	44.03	0.61

From Table 3, it can be observed that in the three orchard areas with different parameters, the operation paths of SADG, DBO, and MI-DBO effectively reduce operation time and fuel consumption compared to traditional row-by-row operation methods. However, the missed mowing rate did not decrease significantly. This is because most of the missed mowing areas are concentrated at the ends of the operation rows. The row-by-row operation method involves a T-shaped turning approach, resulting in longer stops at the ends and hence a lower missed mowing rate. Overall, the missed mowing rates for all methods are less than 0.8%, indicating efficient completion of inter-row mowing tasks. Compared with SADG and DBO, MI-DBO has the lowest operating time and fuel consumption in all three orchards. Compared to the traditional row-by-row operation method, MI-DBO

reduces operation time by an average of 1193.67 s and fuel consumption by 9.99%, on average. Compared to the basic DBO method, MI-DBO reduces operation time by an average of 314.33 s and fuel consumption by 2.79%, on average. From the experiments in the three orchards, it can be seen that as the number of rows increases, the time and fuel consumption of MI-DBO operations decrease significantly. This validates the practicality of MI-DBO in orchard experiments, showing that as the number of rows increases, more time and fuel can be saved.

5. Conclusions

This study transforms the path planning problem of orchard lawn mowers into a TSP and constructs a mathematical model suitable for practical situations. On the basis of this DBO, we used Bernoulli mapping in chaotic mapping and the convex lens reverse-learning strategy to generate uniformly distributed initialization sequences. Then, during the iteration process, a Levy flight strategy was added to prevent the algorithm from becoming stuck in local optima, and an MI-DBO algorithm suitable for quadrilateral orchard environments was proposed. The experimental results show that MI-DBO performs the best in optimizing performance, optimization efficiency, and actual orchard experiments. When operating the orchard lawn mower, it can effectively optimize the operation route and save operation time and fuel consumption, thereby achieving the goal of improving work efficiency. In the future, our team will study the implementation of multiple lawn mowers operating simultaneously in multiple parks within a standardized orchard environment.

Author Contributions: Conceptualization, L.L. and X.W.; methodology, L.L. and X.W.; software, L.L., X.W. and H.L.; validation, L.L., X.W. and H.L.; formal analysis, L.L.; investigation, X.W. resources, H.L.; data curation, L.L. and X.W.; writing—original draft preparation, L.L.; writing—review and editing, L.L. and X.W.; visualization, L.L. and X.W.; supervision, H.L., P.W. and J.L.; project administration, P.W. and X.Y.; funding acquisition, J.L. and X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the earmarked fund for CARS (CARS-27) and supported by the Earmarked Fund for the Hebei Apple Innovation Team of the Modern Agro-industry Technology Research System (HBCT2024150202).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Qian, L.; Lu, H.; Gao, Q.; Lu, H. Household-owned farm machinery vs. outsourced machinery services: The impact of agricultural mechanization on the land leasing behavior of relatively large-scale farmers in China. *Land Use Policy* **2022**, *115*, 106008. [\[CrossRef\]](#)
2. Chakraborty, S.; Elangovan, D.; Govindarajan, P.L.; ELnaggar, M.F.; Alrashed, M.M.; Kamel, S. A comprehensive review of path planning for agricultural ground robots. *Sustainability* **2022**, *14*, 9156. [\[CrossRef\]](#)
3. Liu, L.; Wang, X.; Wang, X.; Xie, J.; Liu, H.; Li, J.; Wang, P.; Yang, X. Path Planning and Tracking Control of Tracked Agricultural Machinery Based on Improved A* and Fuzzy Control. *Electronics* **2024**, *13*, 188. [\[CrossRef\]](#)
4. Sharma, V.; Tripathi, A.K.; Mittal, H. Technological revolutions in smart farming: Current trends, challenges & future directions. *Comput. Electron. Agric.* **2022**, *201*, 107217.
5. Albiero, D.; Garcia, A.P.; Umez, C.K.; Paulo, R.L. Swarm robots in mechanized agricultural operations: A review about challenges for research. *Comput. Electron. Agric.* **2022**, *193*, 106608. [\[CrossRef\]](#)
6. Zhang, M.; Ji, H.Y.; Li, S.C.; Cao, R.Y.; Xu, H.Z.; Zhang, Z.Q. Research Progress of Agricultural Machinery Navigation Technology. *Trans. Chin. Soc. Agric. Mach.* **2020**, *51*, 1–18.
7. Huang, X.M.; Ding, Y.C.; Zong, Y.W.; Liao, Q.X. *Research on the Turning Mode and Path Generation Algorithm of Agricultural Machinery Operation Headland*; Annual Academic Conference of China Agricultural Engineering Society: Chongqing, China, 2011; pp. 1–6.
8. Bochtis, D. Planning and Control of a Fleet of Agricultural Machines for Optimal Management of Field Operations. Ph.D. Thesis, Aristotle University of Thessaloniki, Department of Agricultural Engineering, Thessaloniki, Greece, 2008.
9. Zhou, X.; Zhang, L.; Ma, Y. The Application of Environment Modeling Based on Grid Map in Path Planning of Cleaning Robot. *Manag. Technol. SME* **2021**, *7*, 183–184.

10. Chen, K.; Xie, Y.; Li, Y.; Liu, C.; Mo, J. Full Coverage Path Planning Method of Agricultural Machinery under Multiple Constraints. *Trans. Chin. Soc. Agric. Mach.* **2022**, *5*, 17–26+43.
11. Jeon, C.W.; Kim, H.J.; Yun, C.; Han, X.; Kim, J.H. Design and validation testing of a complete paddy field-coverage path planner for a fully autonomous tillage tractor. *Biosyst. Eng.* **2021**, *208*, 79–97. [[CrossRef](#)]
12. Huang, X.M.; Fu, Z.K.; Ding, Y.C.; Wang, J.Y.; Li, X. Sequence optimization of direction parallel path for the agricultural vehicles in field of rectangle boundary. *J. Huazhong Agric. Univ.* **2014**, *33*, 130–136.
13. Huang, X.M.; Zhang, L.; Wang, S.S.; Luo, C.M. Path Planning of Rapeseed Combine Seeder in Field of Convex Boundary. *Trans. Chin. Soc. Agric. Mach.* **2022**, *53*, 33–40+150.
14. Sethanan, K.; Neungmatcha, W. Multi-objective particle swarm optimization for mechanical harvester route planning of sugarcane field operations. *Eur. J. Oper. Res.* **2016**, *252*, 969–984. [[CrossRef](#)]
15. Yao, J.F.; Liu, J.; Zhang, P.; Teng, G.F. Optimization of agricultural machinery operation path based on doppler and greedy strategy. *J. Chin. Agric. Mech.* **2020**, *41*, 130–137.
16. Xu, J.; Lai, J.; Guo, R.; Lu, X.; Xu, L. Efficiency-Oriented MPC Algorithm for Path Tracking in Autonomous Agricultural Machinery. *Agronomy* **2022**, *12*, 1662. [[CrossRef](#)]
17. Bochtis, D.; Griepentrog, H.W.; Vougioukas, S.; Busato, P.; Berruto, R.; Zhou, K. Route planning for orchard operations. *Comput. Electron. Agric.* **2015**, *113*, 51–60. [[CrossRef](#)]
18. Matei, O.; Pop, P. An efficient genetic algorithm for solving the generalized traveling salesman problem. In Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, 26–28 August 2010; pp. 87–92.
19. Jain, M.; Saihpal, V.; Singh, N.; Singh, S.B. An overview of variants and advancements of PSO algorithm. *Appl. Sci.* **2022**, *12*, 8392. [[CrossRef](#)]
20. Dorigo, M.; Maniezzo, V.; Colnari, A. *The Ant System: An Autocatalytic Optimizing Process*; Academia: Cambridge, MA, USA, 1991.
21. Karabulut, K.; Tasgetiren, M.F. A variable iterated greedy algorithm for the traveling salesman problem with time windows. *Inf. Sci.* **2014**, *279*, 383–395. [[CrossRef](#)]
22. Xue, J.; Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* **2023**, *79*, 7305–7336. [[CrossRef](#)]
23. Arora, S.; Anand, P. Chaotic grasshopper optimization algorithm for global optimization. *Neural Comput. Appl.* **2019**, *31*, 4385–4405. [[CrossRef](#)]
24. Long, W.; Wu, T.; Tang, M.; Xu, M.; Cai, S. Grey Wolf Optimizer Algorithm Based on Lens Imaging Learning Strategy. *Acta Autom. Sin.* **2020**, *46*, 2148–2164.
25. Liu, L.X.; Wang, X.; Yang, X.; Liu, H.J.; Li, J.P.; Wang, P.F. Path planning techniques for mobile robots: Review and prospect. *Expert Syst. Appl.* **2023**, 120254. [[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.