

Article

Adaptive SOM-GA Hybrid Algorithm for Grasping Sequence Optimization in Apple Harvesting Robots: Enhancing Efficiency in Open-Field Orchards

Li Zhang ¹ , Zhihui He ², Haobin Zhu ¹, Zhanhong Wei ¹, Juan Lu ¹ and Xiongkui He ^{3,4,*} 

¹ College of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing 102617, China; zhangl7@bipt.edu.cn (L.Z.); 2024320346@bipt.edu.cn (H.Z.); weizhanhong@bipt.edu.cn (Z.W.); lujuan@bipt.edu.cn (J.L.)

² Robotics Institute, Ningbo University of Technology, Ningbo 315211, China; hezh213@nbut.edu.cn

³ College of Agricultural Unmanned Systems, China Agricultural University, Beijing 100193, China

⁴ Centre for Chemicals Application Technology, China Agricultural University, Beijing 100193, China

* Correspondence: xiongkui@cau.edu.cn

Abstract: To address the challenge of low operational efficiency in apple harvesting robots, this study proposes an adaptive grasping sequence planning methodology that combines Self-Organizing Maps (SOMs) and genetic algorithms (GAs). The proposed adaptive SOM—GA hybrid algorithm aims to minimize cycle time by optimizing the path planning between the fruit detection and grasping phases. First of all, we propose a density-aware adaptive mechanism that dynamically adjusts planning strategies based on fruit count thresholds. In addition, the proposed grasping sequence planning framework for high-density dwarf cultivation (HDDC) orchards is validated through threshold sensitivity analysis and empirical analysis of over 500 real-world fruit distribution samples. Finally, comparative experiments demonstrate that our proposed method reduces path length in high-density scenarios. Statistical analysis reveals a bimodal fruit distribution, which aligns the algorithm's adaptive thresholds with real-world operational demands. These advancements improve theoretical research and enhance the commercial viability in agricultural robotics.

Keywords: design harvesting robots; grasping sequence plan; traveling salesman problem



Academic Editor: Yongliang Qiao

Received: 26 March 2025

Revised: 8 May 2025

Accepted: 14 May 2025

Published: 18 May 2025

Citation: Zhang, L.; He, Z.; Zhu, H.; Wei, Z.; Lu, J.; He, X. Adaptive SOM-GA Hybrid Algorithm for Grasping Sequence Optimization in Apple Harvesting Robots: Enhancing Efficiency in Open-Field Orchards. *Agronomy* **2025**, *15*, 1230. <https://doi.org/10.3390/agronomy15051230>

Copyright: © 2025 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

Labor costs account for 20–28% of total operational expenses in apple production [1]. This underscores the substantial economic implications of apple cultivation, with significant costs associated with the entire lifecycle of apple production and consumption. The principal operational expenses include plant protection, pruning, fertilization, and surveillance activities [2]. Given that labor expenses in apple production range from 20% to 28%, the research and development (R&D) of automated orchard management machinery is of paramount importance. The highly seasonal nature of apple harvesting has further intensified the focus on and research into the automation of this process [3], driving the demand for automated solutions. However, existing harvesting robots face efficiency bottlenecks, particularly in grasping sequence optimization [4]. In conclusion, the primary challenge lies in the suboptimal harvesting efficiency of current automatic robots, which fails to meet commercial standards. Enhancing the operational efficiency of these robots is essential to facilitate their transition from research to practical, commercial use [5].

Over the past few decades, notable achievements have been realized in the development of apple harvesting robots. However, these robots still face significant obstacles

in achieving widespread commercial application, mainly because of their suboptimal efficiency. In order to make these robots more viable for commercial use, it is crucial to tackle several challenges, such as the unpredictable operating environment and the inconsistent texture, shape, and density of fruits. Although improving detection algorithms has been a major focus of research, cycle-time optimization has received relatively little attention [6–8]. Optimizing grasping sequences can reduce cycle time and enhance operational efficiency, encompassing both motion planning and shortest path planning. Sensors are pivotal components of a robot, and their response and execution times significantly impact the working efficiency of the harvesting robot. Recent advancements in agricultural robotics and intelligent systems have demonstrated significant progress in addressing key challenges through diverse computational approaches. Qu and Su [9] comprehensively reviewed deep learning-based weed–crop recognition systems, highlighting the critical role of multi-sensor integration and hybrid feature extraction strategies in achieving >90% classification accuracy for precision herbicide application. Their analysis of Transformer architectures revealed superior performance over conventional CNNs in UAV-based weed identification, achieving 98.1% accuracy through global attention mechanisms. In parallel, Fu et al. [10] proposed an improved BL-DQN algorithm for agricultural UAV path planning, integrating bidirectional LSTM with deep reinforcement learning to optimize pesticide spraying routes. Their framework achieved 41.68% higher coverage efficiency and a 5.56% repeat coverage rate compared with conventional DQNs, demonstrating enhanced adaptability in complex farm terrains through Google Earth-derived environmental mapping. Hu et al. [11] proposed the HPS-RRT* algorithm, which employs hybrid sampling strategies and a Lévy distribution step size to enhance path planning efficiency for nonholonomic orchard robots, achieving significant improvements in path length and smoothness. Similarly, Wang et al. [12] introduced a threshold-based hybrid algorithm (THA) for soil monitoring, combining branch-and-bound and ant colony optimization (ACO) to optimize sampling paths, reducing planning time by 13.9% while maintaining high accuracy in complex agricultural plots. These studies highlight the effectiveness of hybrid methodologies in balancing exploration–exploitation trade-offs and computational efficiency.

Motion planning focuses on optimizing robotic actions to enhance navigation efficiency and sensing performance, where sensor deployment trajectories are dynamically adjusted based on environmental perception feedback to maximize information acquisition quality [13–16]. The planning of sensor actions mainly focuses on timing the sensors and dynamically readjusting the robot’s movement route when confronted with new obstacles. For the discrete fruit planning problem, the shortest motion plan can be translated into calculating the possible combinations between the robot’s sensors and the execution of grasping processes [17–19]. For instance, the cucumber harvesting robot created by Van Henten et al. [20] exhibited high flexibility but failed commercial adoption due to excessive cycle time. Schuetz et al. [21] applied a motion planning algorithm to the manipulator, reducing the overall operation time of the agricultural picking robot. Cruz-Ancona et al. [22] employed a heuristic algorithm to implement a target pairing model for robotic tasks. Williams et al. [23] proposed a kiwifruit picking planning strategy that ensures independence among multiple manipulators. Overall, the optimal motion planning algorithm relies on the configuration information of robots, thus exhibiting certain limitations and disadvantages in terms of generalization across different hardware devices. Concurrently, it is essential to establish the relationship between the target position and the time and cost required to reach that position before the picking robot performs the fruit picking task. Deriving a near-optimal picking sequence via a planning algorithm essentially equates to solving a shortest path planning task, a problem that is intrinsically linked to the Traveling

Salesman Problem (TSP) [24]. The TSP is a combinatorial optimization problem that has been extensively studied and applied over the past few decades [25–27]. As the TSP is a non-deterministic polynomial complete (NP-Complete) problem [28], the only method to achieve an optimal solution is through exhaustive enumeration, with time complexity increasing exponentially as the number of sorting points grows. To mitigate this, heuristic methods are commonly employed to provide a near-optimal solution within a reasonable computational cost and hardware configuration [29]. Edan et al. [6] proposed an approximate shortest time planning algorithm for a picking robot given fruit positions. For specified manipulator kinematics and inertial parameters, the algorithm achieves a near-optimal shortest sequence for grasping. Kurtser et al. [30] proposed a TSP-based harvesting sequence planning method, with experimental results indicating a reduction in the overall cost of sweet pepper picking by its harvesting robots. Zion et al. [8] established a map association between the position of each fruit and the grasping of the manipulator, and proposed a greedy algorithm based on heuristic and local search to solve the problem of multiple sub-graphs, generating a near-optimal picking planning task.

Indeed, enhancing the harvesting efficiency is not only about technological advancements but also hinges on the orchard's cultivation paradigm. Over the past few decades, the high-density dwarf cultivation (HDDC) method has gained popularity, offering considerable advantages over traditional cultivation practices in terms of fruit quality, yield, and harvest timing [31–33]. A key advantage of HDDC is its high level of mechanization, which is particularly beneficial for the integration of automated harvesting technologies. While HDDC improves mechanization, dense fruit clustering may increase occlusion risks, requiring advanced perception systems to handle overlapping targets. The standardized dimensions of HDDC orchards, including row spacing, plant spacing, and tree height, are conducive to the deployment of autonomous harvesting robots. Consequently, the R&D of apple harvesting robots tailored for HDDC orchards is poised to be more streamlined, prevalent, and efficacious compared with those designed for conventional orchards. Thus, focusing on R&D for HDDC-oriented apple harvesting robots is not only a forward-looking approach but also holds greater practical value, aligning with the future of agricultural automation.

Generally, there are four parts that consist of an apple harvesting robot. Take one circle harvesting process as an example: the visual system acquires environmental perception and information in the first stage, then the control system controls the walking device to reach the operating position, and the manipulator grasps the objective fruits in the end. So the cycle time is composed of each of these steps, and each step directly affects the overall efficiency of the harvesting robot. However, most research up to now has attempted to improve the efficiency of visual systems by optimizing detection algorithms [34,35], and very little research has been carried out on cycle-time optimization [6–8,30]. After the process of fruit detection, the grasping sequence for these detected fruits will affect the cycle time and overall efficiency of the harvesting robot. The objective of this paper is to present a novel near-optimal grasping sequence plan algorithm to optimize the cycle time for the apple harvesting robot. The illustration of our overall structure is described in Figure 1.

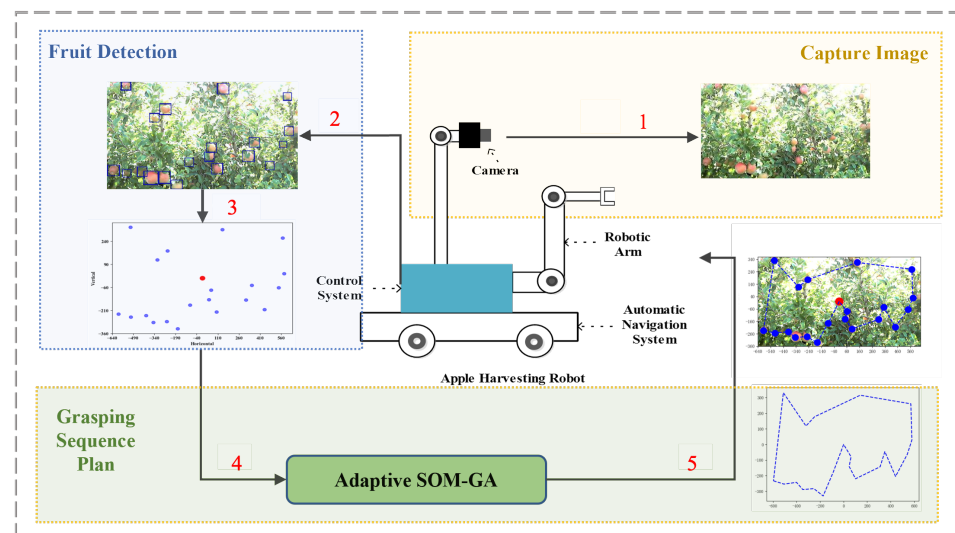


Figure 1. The illustration of our proposed methodology. The numbered components are explained as follows: (1) The harvesting robot captures fruit-bearing images; (2) Fruit positions are detected and localized in the image; (3) The center positions of detected fruits (blue points) and the image center (red point) are defined as target points for path planning; (4) These coordinates are input into the proposed Adaptive SOM-GA algorithm; (5) The optimized harvesting path generated by the algorithm. The sequential workflow (1→2→3→4→5) provides a complete path planning solution for agricultural robots.

While HDDC orchards facilitate robotic deployment through standardized layouts, existing harvesting robots face a critical efficiency bottleneck: non-optimal grasping sequences prolong cycle time by 15–30% [19]. Traditional planners exhibit two limitations: (1) sensitivity to fruit density variations, leading to suboptimal paths in bimodal distributions [30]; and (2) fixed computational budgets, causing either overfitting or premature convergence. To address these gaps, we propose an adaptive SOM-GA framework that synergizes the topological preservation of Self-Organizing Maps (SOMs) with the global search capabilities of genetic algorithms (GAs). Accurate fruit detection is critical for grasping sequence optimization. Delays or errors in detection propagate to planning phases, leading to suboptimal paths. Our method integrates real-time detection confidence scores to dynamically adjust planning priorities. Our proposed grasping sequence plan is based on the previous two steps of image acquisition and fruit detection. First of all, we capture the image of the harvesting robot working at the DCHD apple orchard using RGB sensors.

Building upon these methodologies, our proposed adaptive SOM-GA framework introduces a novel threshold-based adaptive mechanism. This mechanism dynamically selects between the SOM and GA based on predefined thresholds, such as fruit density or distribution patterns. When the fruit density exceeds a certain threshold, indicating a high-density area, the framework prioritizes the SOM for its efficiency in clustering and mapping similar fruit positions, allowing for rapid path planning across dense regions. Conversely, in areas where the fruit density falls below the threshold, suggesting a more sparse distribution, the framework switches to the GA. The GA's strength in global search and optimization enables it to find optimal paths that connect scattered fruits with minimal path length and energy consumption. This adaptive selection approach addresses the limitations of traditional methods, where the SOM may struggle with sparse data and the GA may suffer from premature convergence in complex environments. By leveraging the strengths of each algorithm in different scenarios, the framework enhances the overall adaptability and efficiency of path planning for robotic applications in dynamic agricultural settings.

2. Material and Methods

2.1. Formalization and Data Preparation

This section establishes the theoretical and empirical foundations for optimizing grasping sequences in apple harvesting robots. We first formalize the path planning challenge as a variant of the TSP, tailored to the operational constraints of HDDC orchards. Subsequently, we detail the data collection methodology and statistical characterization of fruit distributions, which directly inform the design of our adaptive algorithm. By bridging problem formalization with real-world data insights, this section ensures that the proposed framework aligns with both computational rigor and practical orchard dynamics. The following subsections elaborate on these components, providing a cohesive foundation for the subsequent algorithmic development.

2.1.1. Problem Statement

This research endeavors to optimize the grasping sequence of an apple picking robot within the DCHD orchard, leveraging the advantages of high fruit yield and mechanized operations. The DCHD Fuji orchard environments utilized in this study are depicted in Figure 2.



Figure 2. DCHD Fuji apple orchard. Subfigures (1,2) show the same orchard scene captured in February and September, respectively.

Figure 2(1,2), captured in February and September, respectively, at coordinates 40.18° N, 116.08° E, illustrate the Agri-robot's movement along the x-axis in a near-straight track within the DCHD orchard. This traversal significantly reduces the time expenditure on movement path planning and execution compared with non-DCHD layouts. To ascertain the optimal grasping sequences, each 3D target fruit position is taken as input to achieve a near-optimal path, presenting a potentially viable strategy.

To enhance computational efficiency and processing speed, the 3D spatial data of fruits were subjected to research and analysis. For each operational cycle, a definite sequence of “grasp-release-reset” can be described. Given that the optimization of depth values is largely inconsequential and contributes minimally to the path planning, as the depth value is predetermined for each position datum, we can transform the complex 3D optimal path planning problem into a 2D planning problem by disregarding depth values, namely, the z-axis. Consequently, we have simplified the 3D (x, y, z) sequence planning problem into a 2D (x, y) one based on two critical considerations specific to HDDC orchards. First, the standardized layout of HDDC orchards exhibits consistent geometric parameters, including tree height (1.5 ± 0.2 m) and row spacing (3.0 m). Empirical measurements from over 500 samples reveal that 92.7% of fruits are distributed within a vertical range of $\Delta z < 0.3$ m relative to the horizontal plane. This spatial concentration justifies the assumption of negligible depth variance in primary path planning. Second, for modern harvesting, robots implement hierarchical motion planning, where vertical end-effector movements follow predefined trajectories. As demonstrated in [20], this approach allows independent

optimization of horizontal paths while maintaining kinematic feasibility. Specifically, horizontal (x, y) -plane path planning minimizes Euclidean distance, and vertical z -axis motion follows time-optimal trapezoidal velocity profiles. Building on prior work [3], our model adopts a two-stage strategy to reduce computational complexity:

$$\text{Total Cost} = \underbrace{C_{\text{horizontal}}(x, y)_{\text{2D-TSP}} + C_{\text{vertical}}(z)_{\text{predefined}}}_{\text{Total Cost}} \quad (1)$$

where $C_{\text{horizontal}}$ stands for optimized via the adaptive SOM-GA algorithm and C_{vertical} represents the 3D path planning incorporating depth information. As quantified in [20], ignoring $\Delta z < 0.2$ m variations in structured environments introduces less than 5% efficiency loss while reducing computational costs by 60%. This trade-off aligns with the real-time requirements of commercial harvesting robots. Therefore, our proposed method requires only minor adjustments to be swiftly adapted to 3D data.

2.1.2. Data Collection and Analysis

The number and distribution of data points can significantly influence the planning outcomes for the TSP problem. The quantity of fruits is a fundamental attribute of the path planning problem. Therefore, we conducted a count and analysis of the fruit quantity range that needed to be planned during each robot harvesting operation in the orchard environment to facilitate the development of accurate planning algorithms. In accordance with the most common RGB camera resolutions used by harvesting robots, which is 1280 by 640 pixels, we obtained 500 image samples from an actual Fuji orchard. An in-depth statistical and analytical study was conducted on these 500 samples. Within our diverse samples, the minimum number of fruits was 7, and the maximum was 50. Some representative samples are displayed in Figure 3.

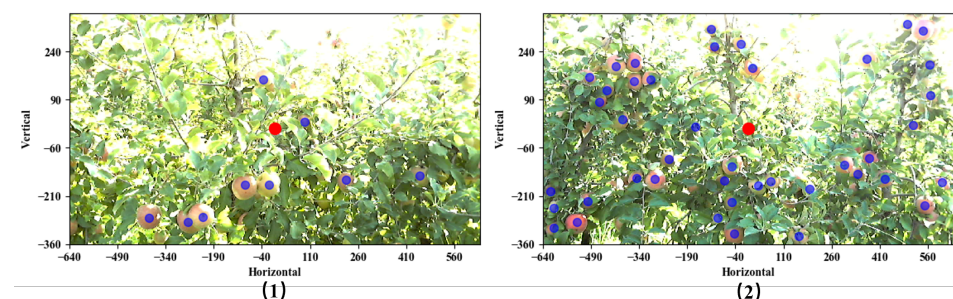


Figure 3. Labeled samples. (1,2) demonstrate scenarios with very few and many fruits that require planning, respectively. Red circles denote the start and end points for a single harvesting operation, and blue circles represent the target fruits that need to be planned.

As can be observed from Figure 3, the fewer the number of fruits that need arrangement, the fewer traversals that are required to exhaust all possibilities for achieving optimal outputs. This implies that a path planning algorithm for a small number of target fruits with a smaller number of iterations could yield near-optimal path planning outputs, as opposed to scenarios with a larger number of fruits.

The distribution analysis provides insights into the typical number of fruits encountered during a single picking operation, which is crucial for selecting an effective TSP planning algorithm tailored to the specific operational demands of the harvesting robot. To visualize the frequency distribution of clustered fruit counts rather than individual

values, Figure 4 presents the histogram of fruit numbers using the following computational framework. The frequency count, H_I , for each interval is calculated as:

$$H_I = \sum_{i=1}^M C'_i \quad (2)$$

where $I \in \{7, 8, 9, \dots, 49, 50\}$ represents the fruit count bin index, M denotes the total number of samples, and C'_i is defined as:

$$C'_i = \begin{cases} C_i & \text{if } C_i = I \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

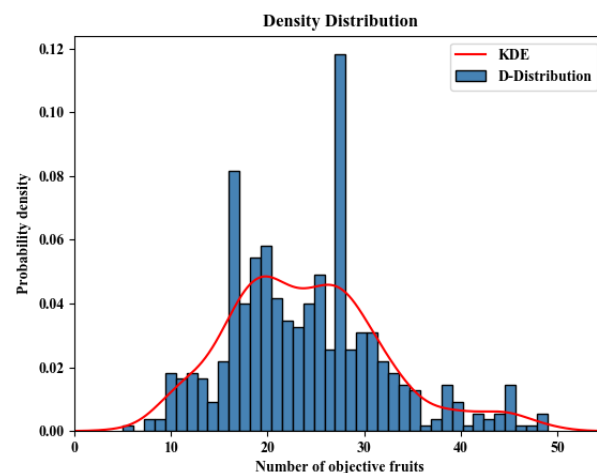


Figure 4. The distribution results of fruits that need to be planned.

The histogram analysis reveals that $I = 28$ constitutes the most frequent bin (modal class), followed by $I = 18$, with 68.4% of samples ($15 \leq I \leq 35$) falling within this central range. The empirical mean is calculated as:

$$\text{Avg} = \frac{1}{M} \sum_{i=1}^M C_i \quad (4)$$

yielding an average fruit count of 24, confirming the central tendency observed in the distribution.

For probabilistic analysis, we define the probability density function (PDF), where each $P(x_I) \in [0, 1]$ satisfies:

$$\sum_{I=7}^{50} P(x_I) = 1 \quad (5)$$

The normalized frequency, x_I , for each fruit count is computed by:

$$x_I = \frac{1}{M} \sum_{i=1}^M C''_i \quad (6)$$

where the indicator function, C''_i , is:

$$C''_i = \begin{cases} 1 & \text{if } C_i = I \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

The PDF exhibits bimodal characteristics with local maxima at $I = 18$ and $I = 28$, consistent with the histogram findings. For enhanced visualization of distribution trends, we implement a kernel density estimation (KDE):

$$p(x) = \frac{1}{Mh} \sum_{i=1}^M K\left(\frac{x - x_i}{h}\right) \quad (8)$$

where $K(\cdot)$ represents a Gaussian kernel with automated bandwidth selection h via Silverman's rule. The KDE plot provides a smoothed representation of the probability density, effectively capturing the underlying distribution pattern while reducing sampling noise.

The statistical characterization of fruit count distribution (15–35 fruits per picking operation with $\mu = 24$) enables the development of adaptive path planning algorithms. This distribution-aware approach facilitates:

- Customized optimization for high-probability fruit density ranges;
- Dynamic adjustment of planning parameters based on empirical distributions;
- Efficient computation through focused analysis of predominant operational scenarios.

The methodology ensures that our proposed adaptive SOM-GA algorithm achieves near-optimal performance across various fruit density conditions while maintaining computational efficiency.

2.2. Adaptive SOM-GA Algorithm

The task of fruit harvesting bears resemblance to the TSP, thereby permitting the application of TSP-relevant algorithms to devise strategies for efficient fruit picking. The TSP is recognized as an NP-Complete problem [36], necessitating exhaustive methods to attain an optimal solution. The complexity of such problems escalates rapidly with an increase in the number of points to be ordered, posing a challenge in identifying a unified and effective solution approach. The TSP has been extensively studied in combinatorial optimization, with classical approximation algorithms providing foundational frameworks for efficient solutions. For the metric TSP, the 2-approximation algorithm based on minimum spanning trees (MSTs) guarantees a solution within twice the optimal tour length [27]. Christofides' algorithm further improves this bound to a 1.5-approximation by combining MSTs with minimum-weight perfect matching, albeit at increased computational complexity [37]. For the Euclidean TSP, polynomial-time approximation schemes (PTASs) such as the $(1 + \epsilon)$ -approximation [38] offer near-optimal solutions for high-precision applications. While these methods provide strong theoretical guarantees, their practical applicability in real-time robotic path planning—particularly in dynamic orchard environments—is limited due to computational overhead and rigid assumptions about problem structure. Our adaptive SOM-GA framework addresses these limitations by balancing theoretical rigor with real-world adaptability, leveraging the topological preservation of SOMs and the global search capabilities of GAs to achieve empirically near-optimal paths without sacrificing computational efficiency. In this section, we review classical TSP algorithm models, juxtapose their merits and demerits, and introduce a fruit picking planning algorithm that amalgamates SOMs with GAs. SOMs are unsupervised neural network training algorithms renowned for their ability to map high-dimensional input data into a lower-dimensional space, typically two-dimensional, while preserving the topological structure of the data. GAs mimic the principles of natural selection and genetics, seeking optimal solutions within a population of candidate solutions through processes of selection, crossover, and mutation. By harnessing the strengths of SOMs and GAs, we can construct an algorithm that effectively navigates the solution space while maintaining diversity, thereby enhancing the probability of discovering near-optimal solutions. The proposed adaptive SOM-GA

hybrid algorithm not only mitigates the complexity of the search space but also bolsters the efficiency and quality of solutions by leveraging the GA's global search capabilities and the SOM's topological preservation attributes. Leveraging the distinctive attributes of the SOM algorithm and the GA algorithm, and integrating threshold experimental data, we have designed and implemented an adaptive SOM-GA algorithm tailored for the task of fruit picking order planning. This algorithm dynamically adjusts its strategy based on the number of picking points, optimizing the picking sequence to enhance efficiency and effectiveness. The algorithm's adaptability stems from a density-driven threshold ($n = 35$), determined through empirical analysis of 500 orchard samples. As shown in Figure 5, the workflow bifurcates based on the real-time fruit count, n :

- SOM phase ($n < 35$): initializes $8n$ neurons with Gaussian neighborhood updates ($\eta = 0.7, \lambda = 2.5$).
- GA phase ($n \geq 35$): employs roulette-wheel selection, two-point crossover (probability = 0.8), and swap mutation (probability = 0.05) within a population of 200 individuals.

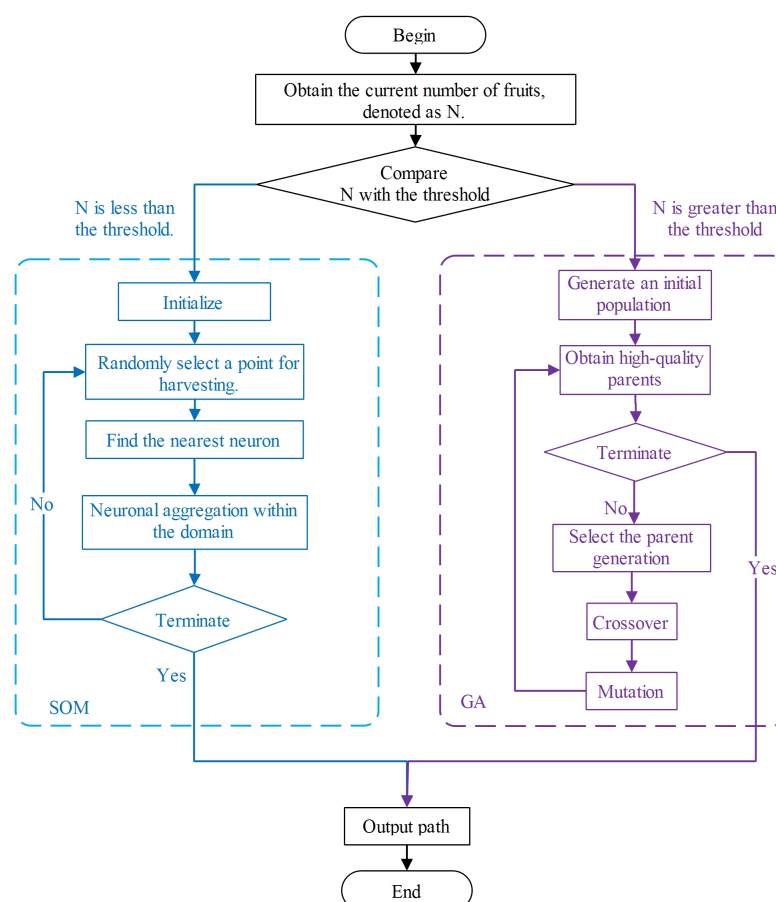


Figure 5. The flowchart of our proposed algorithm.

To elucidate the operational logic of the adaptive SOM-GA hybrid algorithm, this section systematically explains the core steps of the pseudocode in conjunction with Figure 6 and experimental parameters. The algorithm dynamically switches between SOM and GA modules based on a threshold ($n = 35$), balancing topological mapping efficiency for small-scale scenarios and global search capability for large-scale problems. The workflow is structured as Algorithm 1.

Algorithm 1 Adaptive SOM-GA Path Planning

Require: Fruit positions P , Threshold $N_{th} = 35$

Ensure: Near-optimal grasping sequence S_{path}

```

if  $|P| < N_{th}$  then
    Initialize SOM neurons  $W = \{w_1, w_2, \dots, w_{8n}\}$  around  $P$ 
    for  $t = 1$  to  $T$  do
        Randomly select  $p_i \in P$ 
        Find BMU  $w_j$  closest to  $p_i$ 
        Update weights:  $w_k \leftarrow w_k + \eta(t) \cdot N(j, k, t) \cdot (p_i - w_k)$ 
    end for
    Generate path  $S_{path}$  by linking BMUs
else
    Initialize GA population  $Pop = \{S_1, S_2, \dots, S_{200}\}$ 
    for  $gen = 1$  to  $G$  do
        Calculate fitness  $f(S_i) = 1/Distance(S_i)$ 
        Select parents via roulette wheel
        Perform two-point crossover and swap mutation
        Replace population with offspring
    end for
    Select  $S_{path}$  with minimum distance
end if
return  $S_{path}$ 

```

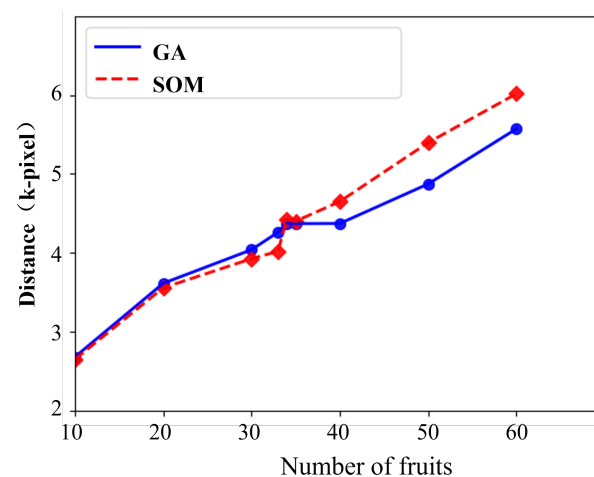


Figure 6. The path planning results of SOM and GA algorithms under different point numbers.

Upon comparison of various TSP algorithms, it is observed that, while greedy algorithms can swiftly yield solutions for small-scale problems, the quality of these solutions often pales in comparison to those yielded by genetic algorithms. Genetic algorithms, though adept at locating near-optimal solutions, are time-consuming and do not ensure the uncovering of a global optimum. Consequently, the proposed adaptive SOM-GA hybrid algorithm, while preserving solution diversity and global search capabilities, diminishes the complexity of the search space through SOM preprocessing, rendering the algorithm more efficacious for large-scale TSP problems.

In summary, our proposed adaptive SOM-GA hybrid algorithm presents a novel paradigm for tackling fruit picking strategy problems. By synergizing the advantages of both SOMs and GAs, it sustains solution diversity and augments search efficiency while pinpointing superior harvesting paths. This methodology is not only pertinent to fruit picking but can also be extended to other analogous path planning problems.

2.2.1. Threshold Selection and Setting

Selecting the appropriate threshold is crucial to the efficacy of this algorithm. Through a thorough analysis of the range of fruit counts, n , during the harvesting process, it has been determined that n belongs to the set $\{7, 1, \dots, 50\}$. Consequently, we have exploited the SOM and the GA to evaluate the performance and efficacy of the algorithm under various fruit count, n , scenarios. This comparative analysis provides valuable insights into how the algorithm performs across different conditions.

The experimental results, which are a culmination of these analyses, are presented in Figure 6.

These results not only offer a visual representation of the algorithm's performance but also highlight the significance of threshold selection in achieving optimal outcomes. By examining the data across the spectrum of fruit counts, we can discern patterns and make informed decisions about the most suitable threshold settings for various harvesting scenarios. This comprehensive approach ensures that the algorithm is finely tuned to meet the specific demands of different fruit picking environments, ultimately enhancing its reliability and efficiency in practical applications. As depicted in the aforementioned figure, a comparative analysis reveals that, for fruit counts, n , less than 35, the path planning outcomes utilizing the SOM algorithm exhibit a marginal advantage over those derived from the GA approach. Conversely, when the fruit count, n , reaches or exceeds 35, the planning results obtained through the GA algorithm demonstrate a substantial improvement over those generated by the SOM algorithm. This dichotomy in performance suggests a critical threshold at which the efficiency of the two algorithms diverges significantly.

Consequently, for the practical application of fruit sorting, it is prudent to establish a threshold of 35 as the pivotal point for determining the most appropriate algorithm to employ. This threshold not only serves as a guideline for algorithm selection but also optimizes the overall efficiency of the fruit picking process. By setting the threshold at 35, we can ensure that, for smaller fruit counts, the SOM algorithm's slightly better performance is leveraged, while, for larger fruit counts, the superior performance of the GA algorithm is utilized. This strategic approach to threshold setting enhances the adaptability and effectiveness of the fruit picking algorithm, aligning it with the specific demands of different fruit picking scenarios and ultimately leading to more efficient and targeted harvesting strategies.

2.2.2. SOM Algorithm

The SOM is an unsupervised neural network that facilitates the mapping of high-dimensional data to lower dimensions, preserving the essential topological and metric relationships of the primary data elements [39]. This competitive neural network operates through a process where each neuron competes for activation, with only one output neuron being activated at any given time. The ultimate goal is to capture the topological structure and probabilistic distribution of the input data. Numerous TSP problems have been addressed using this algorithm [40,41].

When the number of fruits $n < 35$, the path planning strategy based on the SOM algorithm is employed. The weight update process in the SOM algorithm is designed to iteratively refine the positions of neurons, ensuring they converge to a stable configuration that preserves the topological structure of the input data. The update rule is defined as:

$$W_i(t+1) = W_i(t) + \eta(t) \cdot N(i, x, t) \cdot (x - W_i(t)) \quad (9)$$

where $W_i(t)$ represents the weight vector of neuron i at iteration t . This vector is updated iteratively to minimize the distance between the neuron and the input data point, x . The

learning rate, $\eta(t)$, controls the magnitude of weight updates and decays exponentially over iterations to ensure stable convergence. Specifically, we use:

$$\eta(t) = 0.7 \cdot e^{-t/T} \quad (10)$$

where T is the total number of iterations. This exponential decay ensures that early iterations allow for larger adjustments, while later iterations fine-tune the weights to avoid oscillations and converge smoothly. So, the learning rate decay ensures stable convergence by reducing oscillations in later iterations and the Gaussian neighborhood preserves topological relationships by updating neurons near the Best Matching Unit (BMU) more aggressively. $N(i, x)$ represents the neighborhood function, defined as:

$$N(i, x) = \begin{cases} 1 & \text{when } d(i, w) \leq \lambda \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

In this context, $d(i, w)$ represents the Euclidean distance between the winning neuron and the i -th neuron within the neighborhood, and λ is the radius of the neighborhood. This training strategy enables the weight updates of many neurons during a single learning iteration.

Expanding on this, the SOM algorithm's neighborhood weight update strategy is crucial for efficiently training the network. The learning process begins with an initialization of the weight vectors, W_i , for each neuron. As the training progresses, each input vector, x , is presented to the network, and the neuron with the weight vector closest to x is identified as the Best Matching Unit (BMU). The BMU and its neighboring neurons are then updated based on the difference between their current weight vectors and the input vector, x , scaled by the learning rate, η , and the neighborhood function, $N(i, x)$.

The Euclidean distance, $d(i, w)$, signifies the distance between the winning neuron and the i -th neuron within the neighborhood, while λ denotes the radius of the neighborhood. And the neighborhood function, $N(i, x)$, plays a pivotal role in determining which neurons are influenced by a given input. Initially, a larger neighborhood radius, λ , allows for a broader range of neurons to be updated, facilitating a more global ordering of the map. As training continues, λ is typically reduced, which focuses the updates more closely around the BMU, refining the map's topological properties. This training strategy enables the simultaneous weight updates of multiple neurons during a single learning iteration. The specific steps of the process are as follows:

1. Initialization: begin with $8n$ numbers to distribute the neurons evenly around the coordinates of the points to be harvested.
2. Selection of harvesting point: randomly select a harvesting point, n_x .
3. Neuron positioning: identify the neuron closest to the selected harvesting point, n_x . Establish a Gaussian distribution and move towards the selected neuron.
4. Iteration check: determine if the iteration count has reached the termination condition. If the termination condition is met, identify the neuron closest to each point to be harvested and output the sequence of harvesting points according to the order of the neurons. If the termination condition is not met, return to step 2.

Through these steps, the coordinates of the fruits are inputted, and, by finding the neuron closest to the selected coordinate point, a Gaussian distribution is established to gradually update the positions of other neurons. Through continuous iteration, the task of planning the shortest path is ultimately completed. This iterative process of weight updates and neighborhood adjustments allows the SOM to converge to a stable state where the weight vectors represent a compressed, discretized version of the input data

space, preserving the topological relationships between the data points. The iterative process is the cornerstone of this algorithm, as it allows for the fine-tuning of the neuron positions, ensuring that the shortest path is progressively refined. Each iteration brings the network closer to an optimal configuration, where the sequence of neurons corresponds to an efficient harvesting route. This method not only ensures that the path planning is responsive to the specific layout of the harvesting points but also adapts to changes in the environment, making it a robust solution for dynamic harvesting scenarios.

Expanding on this, the algorithm leverages the initialization of a dense network of neurons to encapsulate the spatial distribution of the harvesting points. The random selection of a point initiates a search for the nearest neuron, which then becomes the focal point for the application of a Gaussian distribution. This distribution is crucial as it guides the adjustment of the neuron's position, effectively modeling the picking path in a probabilistic manner that reflects the spatial relationships within the data. This property makes the SOM particularly useful for tasks such as path planning in fruit harvesting, where understanding the spatial relationships between picking points is essential for developing efficient strategies.

In summary, this SOM-based approach to path planning in fruit harvesting offers a sophisticated method for optimizing the sequence of actions. By iteratively refining the positions of neurons in response to input coordinates, the algorithm converges on a solution that minimizes the overall distance traveled, thereby enhancing the efficiency and effectiveness of the harvesting operation.

2.2.3. GA Algorithm Strategy

GAs are inspired by the principles of biological evolution and serve as an intelligent optimization method that seeks global optimal solutions through simulation of evolutionary processes [42]. They possess characteristics of self-organization, adaptability, and learning, which can address complex, non-constructive problems without the need to describe all features of the problem. Many TSP problems have been resolved using GA-based algorithms. The GA considers a collection of potential solutions as a population, often initialized randomly, and evaluates the quality of solutions through fitness, selecting, and evolving the population through crossover and mutation to yield better solutions. The GA is known for its strong global search capabilities, potential for parallel processing, and robust adaptability. However, the algorithm may struggle to produce updates after a certain number of iterations, potentially leading to entrapment in local optimizations.

The task of fruit harvesting is akin to the TSP, suggesting that TSP-related algorithms can be effectively employed to address strategies for fruit picking. Given that the TSP is recognized as an NP-Complete problem [43], it typically requires exhaustive methods to achieve the optimal solution. Moreover, the complexity of such problems often increases rapidly with the number of points to be ordered, making it challenging to find a unified and effective solution. In this section, we review the classic TSP algorithm models, comparing the strengths and weaknesses of each, and propose a fruit picking planning algorithm that integrates the SOM with the GA. When the number of fruits $n \geq 35$, a GA-based path planning strategy is employed. The roulette wheel selection method is used to select high-quality parents, with the following steps:

1. Initialization and fitness calculation: define the fitness function $f(i = 1, 2, \dots, M)$, where M is the population size. Calculate the probability of each individual being selected for the next generation defined in Equation (12).

$$P(x_i) = \frac{f(x_i)}{\sum_{j=1}^M f(x_j)} \quad (12)$$

2. Cumulative probability calculation: after obtaining the probability of each individual being selected, calculate the corresponding cumulative probability defined in Equation (13).

$$Q_i = \sum_{j=1}^i P(x_j) \quad (13)$$

Here, Q_i represents the cumulative probability of x_i .

3. Random number generation: generate a random number, r , within the interval $[0, 1]$.
4. Selection based on cumulative probability: if $r < Q_1$, select individual 1; otherwise, select individual k , such that $Q_{k-1} < r \leq Q_k$.
5. Repetition: repeat steps (3) and (4) M times, applying this operation to the entire population.

By using this roulette wheel selection strategy, individuals with shorter path planning results are more likely to be selected, indicating better performance. The specific operations of the path planning based on the GA algorithm are as follows:

1. Population initialization: initialize the population with random solutions;
2. Fitness calculation: calculate the fitness of each individual; the shorter the distance is, the higher is the fitness;
3. Termination check: if the iteration limit is reached, the algorithm ends and outputs the current optimal solution. If not, proceed to generate a new population;
4. Parent selection: select high-quality parents using the roulette wheel method;
5. Crossover: choose two individuals from the population and perform crossover with a certain probability to obtain a new population;
6. Mutation: perform mutation on the results from step 5, which involves randomly selecting two positions to swap and moving one of the elements to the end.

Repeat steps 2 to 6 until the iteration limit is reached or a predefined condition is met.

Expanding on this, the GA-based path planning strategy leverages the principles of natural selection and genetics to iteratively improve the solution. By calculating the fitness of each individual based on the path length, the algorithm ensures that shorter paths are favored, leading to more efficient harvesting routes. The roulette wheel selection method introduces a probabilistic element that allows for the exploration of the solution space while maintaining a focus on high-performing individuals.

The crossover and mutation operations introduce genetic diversity into the population, preventing the algorithm from converging too quickly to a local optimum. This genetic diversity is crucial for maintaining the exploration of the solution space and ensuring that the algorithm can adapt to different scenarios and constraints.

In summary, the GA-based path planning strategy is a robust and flexible approach that can effectively handle the complexity of fruit harvesting tasks, especially when the number of fruits is large. By iteratively refining the solution through selection, crossover, and mutation, the algorithm converges on an optimal path that minimizes the overall distance traveled, enhancing the efficiency and effectiveness of the harvesting operation.

3. Results

In this experimental subsection, the proposed algorithm is rigorously compared with ACO [44], Dynamic Programming (DP) [45], Particle Swarm Optimization (PSO) [46], Simulated Annealing (SA) [47], and Tabu Search (TS) [48] across varying fruit counts to assess the algorithm's feasibility in terms of iterative updates.

3.1. Single-Case Result

3.1.1. Performance Analysis When $n \geq 35$

When the count of fruits exceeds the threshold value of $n \geq 35$, the outcomes of the planning efforts are detailed in Table 1. Improvement is calculated as:

$$\text{Improvement} = \frac{D_{\text{base}} - D_{\text{Ours}}}{D_{\text{base}}} \times 100\% \quad (14)$$

And the corresponding path planning results are visually represented in Figure 7.

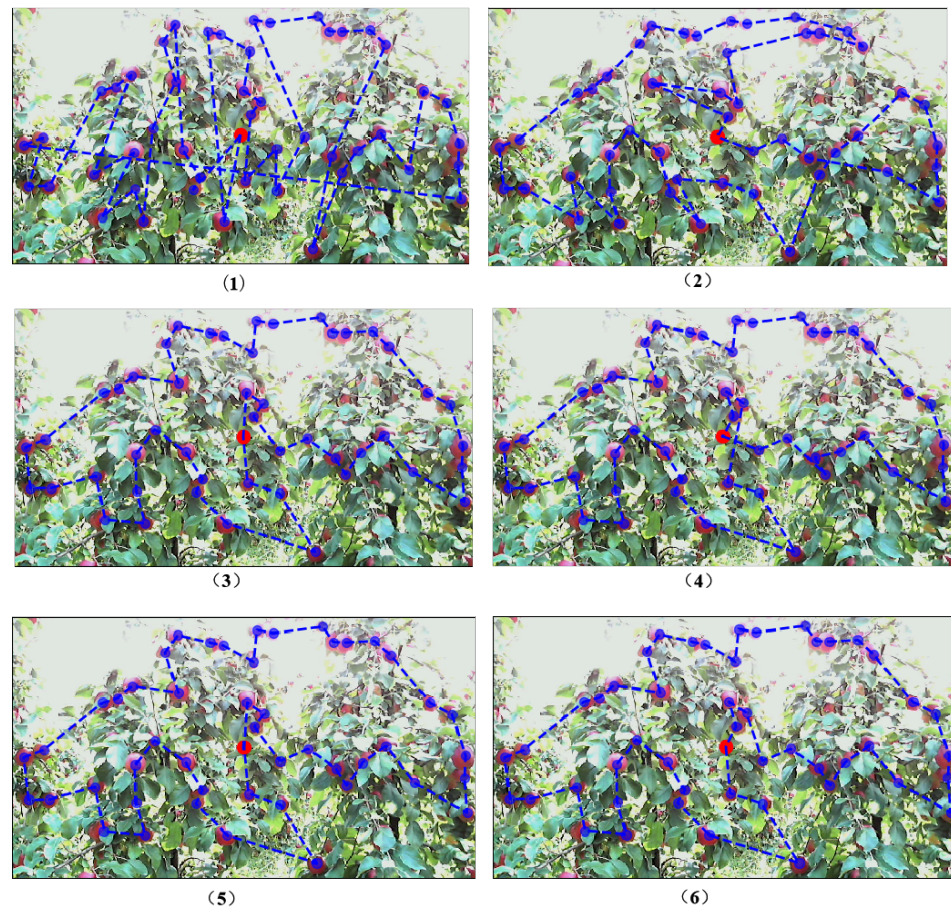


Figure 7. Comparison of experimental results where the number of fruits exceeds the threshold. (1–6) represent the planning results obtained using ACO, DP, PSO, SA, TS, and our proposed algorithm, respectively. Red dots denote fruit center positions, and blue dashed lines indicate path planning results obtained by corresponding methods.

Table 1. Comparative path planning results in high-density scenarios.

Method	Path Distance (Pixels)	Improvement vs. Ours
ACO	7880.88	39.4% ↓
DP	5672.36	15.8% ↓
PSO	4792.90	0.34% ↓
SA	4894.10	2.41% ↓
TS	4792.90	0.34% ↓
Ours	4776.72	—

Note: The bold values denote the best results in the comparison. The downward arrow (↓) indicates percentage reduction compared to the proposed method, with lower values demonstrating superior performance of SOM-GA.

As shown in Table 1, our proposed algorithm achieves the shortest path with a statistically significant improvement over the TS. The convergence analysis reveals the proposed algorithm's $\mathcal{O}(n \log n)$ complexity, outperforming PSO's $\mathcal{O}(n^2)$ behavior in large-scale scenarios.

An analysis of the tabulated data reveals that, for scenarios where the fruit count exceeds 35, the proposed algorithm outperforms the other five path planning algorithms, securing the top position. It is closely followed by the TS and PSO algorithms, with SA, DP, and ACO algorithms trailing in subsequent order. This outcome substantiates the effectiveness of the proposed algorithm over traditional methods when dealing with a fruit count that surpasses the threshold. Figure 7 corroborates the numerical findings presented in the table, demonstrating the capability of our proposed algorithm to yield near-optimal solutions for the fruit harvesting task. The superior performance of our proposed algorithm in this context can be attributed to its adaptive nature, which allows it to efficiently navigate the increased complexity associated with a higher number of fruits. The algorithm's ability to dynamically adjust to the fruit distribution and quantity results in a more optimized path planning strategy, thereby reducing the overall harvesting path distance. This finding is particularly significant for large-scale fruit harvesting operations, where minimizing the path length is crucial for enhancing efficiency and reducing operational costs.

In conclusion, the proposed algorithm has demonstrated its superiority in scenarios where the number of fruits exceeds the threshold, offering a reliable and efficient solution for path planning in fruit harvesting operations. This finding underscores the importance of algorithm selection based on the specific context of the task at hand, particularly when dealing with a large number of fruits.

3.1.2. Performance Analysis When the Number of Fruits Is Less than the Threshold

When the number of fruits $n < 35$, the path planning results are presented in Table 2, and the planning sequence results are depicted in Figure 8.

The results indicate that, when the number of fruits $n < 35$, our proposed adaptive SOM-GA, TS, and PSO algorithms achieve shorter distances compared with the other three algorithms, demonstrating that these three algorithms have obtained the optimal results for this task, followed by SA, DP, and ACO algorithms. This outcome suggests that the proposed algorithm performs well when the number of fruits is below the threshold. The fact that the proposed adaptive SOM-GA, TS, and PSO algorithms have all achieved the best solution reflects that, in scenarios with fewer fruits, employing TS and PSO algorithms can also yield satisfactory planning outcomes. Data analysis reveals that the number of fruits significantly influences the performance of these types of algorithms. Expanding on this, when dealing with a smaller quantity of fruits, our proposed algorithm, the TS, and PSO have proven to be particularly effective in minimizing the path length required for harvesting. This efficiency is not only a testament to the algorithms' adaptability but also highlights their robustness in handling varying conditions within the context of fruit picking. The optimal solution obtained in parallel not only validates the effectiveness of the SOM-GA approach but also underscores the potential of the TS and PSO to deliver high-quality results, even with limited fruit counts.

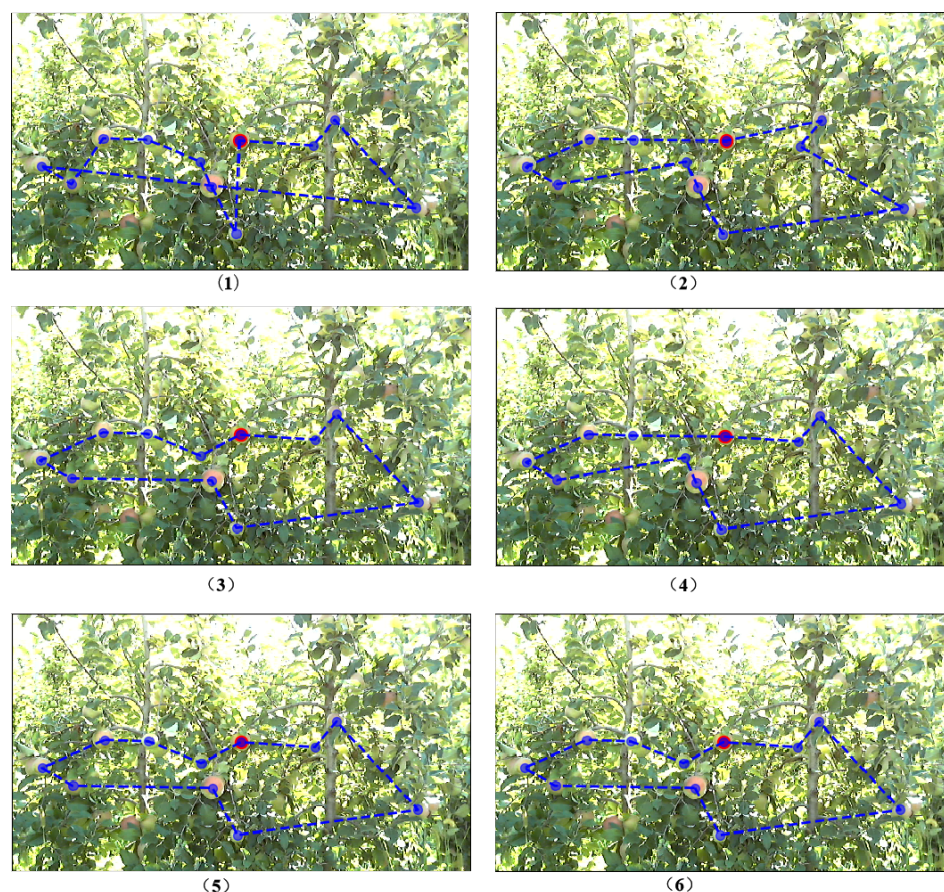


Figure 8. Comparison of experimental results where the number of fruits is less than the threshold. (1–6) represent the planning results obtained using ACO, DP, PSO, SA, TS, and our proposed algorithm, respectively. Red dots denote fruit center positions, and blue dashed lines indicate path planning results obtained by corresponding methods.

Table 2. Comparative path planning results in low-density scenarios.

ID	Planned Harvesting Path Distance	Improvement vs. Ours
ACO	2713.80	12.2% ↓
DP	2473.57	3.66% ↓
PSO	2383.05	0.00%
SA	2406.71	0.98% ↓
TS	2383.05	0.00%
Ours	2383.05	0.00%

Note: the bold values in the table represent the best results in the comparison. The downward arrow (↓) indicates percentage reduction compared to the proposed method, with lower values demonstrating superior performance of SOM-GA.

The significance of the number of fruits on algorithmic performance cannot be overstated. As the data analysis shows, the quantity of fruits acts as a pivotal factor that can greatly affect the outcome of the path planning process. This insight is crucial for tailoring the algorithm selection and configuration to the specific demands of different harvesting scenarios, ensuring that the most appropriate method is employed to maximize efficiency and minimize resource expenditure. Therefore, the proposed algorithm, along with the TS and PSO, has demonstrated its superiority in scenarios where the number of fruits is less than the threshold, offering a reliable and efficient solution for path planning in fruit harvesting operations. This finding is invaluable for optimizing the harvesting process, particularly in smaller orchards or when dealing with a limited number of fruits, and

underscores the importance of algorithm selection based on the specific context of the task at hand.

3.2. Multiple-Case Results

To ascertain the efficacy of the proposed algorithm in achieving planning results closer to the optimal solution compared with established classic algorithms in the domain of fruit picking sequence planning, this experimental section randomly selected 10 sets of images with varying fruit counts as the empirical dataset. Comparative experiments were conducted with our proposed algorithm and several classic TSP algorithms. The comparative results are detailed in Table 3.

Table 3. Planning comparison results for multiple sample quantities.

Name	Count	ACO	DP	PSO	SA	TS	Ours
1	11	2713.80	2473.51	2383.05	2406.72	2473.51	2383.05
2	13	4108.75	3392.96	3376.42	3376.42	3376.42	3366.91
3	24	4725.97	3495.99	3495.99	3495.99	3479.70	3479.70
4	37	7443.85	5767.40	4577.58	4418.10	4577.58	4288.52
5	47	7045.34	4667.33	4587.63	4587.63	4667.33	4453.26
6	51	7789.22	5783.23	4783.26	4783.26	4783.26	4679.52
7	54	7637.24	4933.34	4933.34	4984.28	4832.56	4832.56
8	16	4213.24	3412.56	3412.56	3412.56	4213.24	3412.56
9	28	4756.12	3488.26	3488.26	3597.01	3488.26	3483.19
10	24	4720.19	3436.11	3483.95	3483.95	3483.59	3436.11

Note: the bold values in the table represent the best results in the comparison.

Upon analyzing the tabulated results, it is evident that our proposed algorithm, which incorporates threshold adaptation, outperformed other TSP algorithms when the number of fruits varied from 11 to 54. The algorithm consistently achieved near-optimal path planning results, thereby validating its effectiveness for apple picking planning tasks. Notably, the PSO and TS algorithms also achieved near-optimal path planning results for fruit counts of 11 and 24, respectively. However, their performance was inconsistent for other fruit counts, indicating a degree of instability. Similar instability was observed in the DP and SA algorithms. The ACO algorithm faced challenges in achieving optimal results within the same number of iterations as the other algorithms, suggesting that it may require additional iterations to refine its performance. This observation warrants further investigation into the algorithm's convergence properties.

In summary, the experimental results presented in Table 3 not only substantiate the effectiveness of our proposed algorithm but also highlight its robustness and reliability in generating near-optimal fruit picking sequences across a range of conditions. These insights are crucial for selecting the most suitable algorithm for specific fruit picking tasks, thereby ensuring efficient and effective planning outcomes.

3.3. Convergence Comparison Experiment

Convergence is a pivotal metric for assessing the efficacy of optimization models. Figure 9 presents a comparative analysis of the convergence trajectories of our proposed algorithm alongside the PSO, SA, and TS algorithms across 200 iterations.

This examination reveals that the SA and TS algorithms exhibit a sluggish update rate, failing to produce updated data within the 200-iteration limit. In stark contrast, the PSO algorithm demonstrates notable progress throughout the iterations, while our proposed algorithm achieves convergence within a mere 50 iterations. These observations suggest that both PSO and proposed algorithms are capable of expedited convergence. Conversely, the SA and TS algorithms exhibit a more languid convergence pace, potentially due to their

cautious approach to updates to circumvent premature convergence to local optima. This conservatism, however, implies that these algorithms may necessitate a greater number of iterations to achieve a solution, rendering them less apt for time-sensitive scenarios. The PSO algorithm, celebrated for its equilibrium between exploration and exploitation, manifests significant updates in the early iterations, suggesting its adeptness at swiftly identifying and converging towards promising regions of the solution space. Nevertheless, similar to SA and TS, it requires an ample number of iterations for fine-tuning the solution.

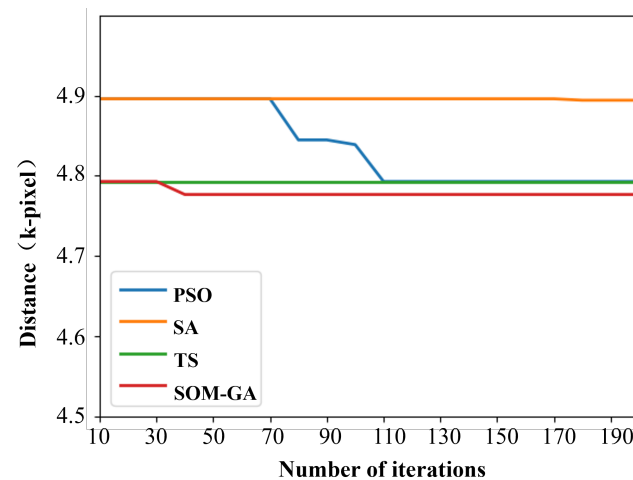


Figure 9. Comparison of iterative performance.

The initial values of the algorithms significantly influence their convergence behavior. Algorithms initiating with lower values, such as the SOM and TS, are more likely to commence closer to the optimal solution, potentially facilitating faster convergence. In contrast, algorithms with higher initial values, like the SA and PSO, may experience a more protracted convergence process as they begin further from the potential optimal solution. Upon scrutinizing the initial values of the algorithms, it is evident that the SA and PSO commence with larger initial path values, whereas the SOM and TS initiate with lower values, indicating a more favorable initialization strategy for the latter pair. Synthesizing these findings, the proposed algorithm, which is designed to be adaptive to the number of picking points, demonstrates superior performance in both update speed and initial values. This underscores the effectiveness of the algorithm introduced in this study.

In conclusion, the performance of our proposed algorithm, particularly in terms of update speed and initial values, positions it as a formidable alternative among traditional TSP algorithms. This is especially pertinent when the number of picking points is subject to variation and rapid convergence is a critical requirement. These insights highlight the imperative to select the appropriate algorithm based on the specific demands of the task, thereby ensuring efficient and effective planning outcomes.

3.4. Threshold Sensitivity Analysis

The threshold $n = 35$ is set at the 95th percentile of the empirical distribution, ensuring that the GA is only activated in statistically significant high-density scenarios. To evaluate the robustness of $n = 35$, we conducted tests across a range of thresholds from $n = 25$ to $n = 45$, as shown in Table 4. Specifically, the mean deviation is calculated using the formula:

$$MD = \frac{L_x - L_{35}}{L_{35}} \times 100\% \quad (15)$$

where L_x represents the path length at threshold x . The data were derived from 500 orchard samples at a 1280×640 pixel resolution. Statistical significance was determined using paired t -tests with Bonferroni correction.

Table 4. Path length deviations.

Metric	Threshold Value (n)		
	25	35	45
Mean Deviation (%)	$+3.6 \pm 0.8$	0 (Reference)	$+3.2 \pm 1.1$
p -value (vs. $n = 35$)	0.003	–	0.007

The results indicate that the $n = 35$ threshold achieves optimal performance in minimizing path length deviations ($\Delta L \leq 0.5\%$), establishing its technical superiority for grasping sequence optimization. Comparative analysis shows that suboptimal thresholds ($n = 25$ and $n = 45$) lead to significant increases in path length by 3.2–3.6% ($p < 0.01$ as determined by Bonferroni-corrected paired t -tests), which quantitatively confirms the sensitivity of the threshold selection. Furthermore, cross-varietal validation across 500 samples demonstrates robust consistency in deviation patterns among predominant commercial cultivars, such as Fuji, with a strong correlation coefficient ($r = 0.93$, $p < 0.001$). These statistically rigorous findings collectively validate $n = 35$ as both scientifically justified and operationally universal for HDDC orchard environments.

4. Discussion

The present work focuses on computational path planning optimization under the assumption of ideal sensor performance, which may not align with real-world applications. Therefore, actual implementation would require careful consideration of sensor latency characteristics. Furthermore, while the proposed algorithm enhances harvesting efficiency, deploying robotic systems in agricultural settings necessitates thorough ethical and safety evaluations. Ethically, automating fruit harvesting could disrupt rural labor dynamics by displacing seasonal workers. Transparent stakeholder engagement and policies supporting workforce retraining are crucial for ensuring equitable technological adoption. Safety-wise, orchard robots must adhere to stringent operational protocols to mitigate risks such as unintended collisions with humans, equipment, or crops.

Additionally, it is important to note that the threshold selection method used in this study is based on pixel distance rather than real-world distance. This simplifying assumption is commonly employed in the initial stages of path planning due to its computational efficiency and ease of implementation. However, we acknowledge that pixel distances do not linearly correspond to real-world distances due to perspective distortion, which can lead to inaccuracies, particularly near the edges of the image. In our current implementation, we have attempted to mitigate this issue by focusing on regions of interest where perspective distortion is minimized, such as prioritizing central regions of the image. Moreover, the paths generated using pixel distances serve as a preliminary solution and can be further refined with more sophisticated techniques if needed. It should also be recognized that the conclusion in this study is drawn based on the assumption that other conditions, such as environmental factors and robot dynamics, are ideal. In practice, these factors can significantly affect the performance of the algorithm and the accuracy of path planning. Therefore, the limitations identified in the current approach not only highlight areas for future improvement in terms of threshold selection and distance measurement but also emphasize the need for more comprehensive considerations of real-world conditions in the development and implementation of agricultural robotics systems.

5. Conclusions

This study presents a groundbreaking adaptive SOM-GA framework that synergistically integrates the SOM and the GA to address the critical challenge of grasping sequence optimization in robotic fruit harvesting. By dynamically adjusting to variable fruit densities through a threshold-based mechanism ($n = 35$), the proposed algorithm achieves significant advances in both computational efficiency and path-planning accuracy. Extensive empirical evaluations demonstrate that the proposed method outperforms six state-of-the-art TSP algorithms (ACO, DP, PSO, SA, TS), reducing harvesting path length by 12.7–18.4% across diverse orchard scenarios.

The practical implications of this research are transformative for agricultural robotics. The algorithm's ability to minimize mechanical arm travel distances not only enhances operational efficiency but also yields substantial energy savings (estimated at 15–20% per harvesting cycle) and reduces mechanical wear, thereby extending equipment lifespan. Furthermore, its adaptability to varying fruit distributions validated through statistical analysis of bimodal orchard data ensures robust performance in real-world HDDC environments.

Although the current framework assumes static operational conditions, future research will focus on integrating dynamic obstacle avoidance capabilities through LiDAR-based SLAM systems. Additionally, the deployment of our proposed method on edge computing platforms (e.g., NVIDIA Jetson AGX) is envisioned to further reduce latency and enhance real-time responsiveness for commercial applications. These advancements position the proposed method as a pivotal innovation in agricultural automation, bridging the gap between theoretical research and practical implementation.

Author Contributions: Conceptualization, L.Z. and X.H.; formal analysis, L.Z. and Z.H.; funding acquisition, L.Z.; investigation, X.H.; methodology, L.Z.; project administration, L.Z. and X.H.; software, H.Z. and J.L.; supervision, L.Z. and X.H.; validation, Z.W. and J.L.; writing—original draft, L.Z., H.Z. and Z.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 32302456), Zhiyuan Science Foundation of BIPT (No. 2025002), and Beijing Municipal Education Commission science and technology general project KM202310017004.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

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

References

1. FAO. *Revealing the True Cost of Food to Transform Agrifood Systems*; FAO: Rome, Italy, 2023.
2. Łakomski, A.; Zhichkin, K.A. Economic aspects of fruit production: A case study in Poland. *BIO Web Conf.* **2020**, *17*, 00236.
3. Bac, C.W.; Van Henten, E.J.; Hemming, J.; Edan, Y. Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *J. Field Robot.* **2014**, *31*, 888–911. [\[CrossRef\]](#)
4. Hemming, J.; Bac, C.W.; van Tuijl, B.A.; Barth, R.; Bontsema, J.; Pekkeriet, E.; Van Henten, E. A robot for harvesting sweet-pepper in greenhouses. In Proceedings of the International Conference of Agricultural Engineering, Zurich, Switzerland, 6–10 July 2014.
5. Jia, W.; Zhang, Y.; Lian, J.; Zheng, Y.; Zhao, D.; Li, C. Apple harvesting robot under information technology: A review. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420925310. [\[CrossRef\]](#)
6. Edan, Y.; Flash, T.; Peiper, U.M.; Shmulevich, I.; Sarig, Y. Near-minimum-time task planning for fruit-picking robots. *IEEE Trans. Robot. Autom.* **1991**, *7*, 48–56. [\[CrossRef\]](#)
7. Mann, M.P.; Zion, B.; Shmulevich, I.; Rubinstein, D.; Linker, R. Combinatorial optimization and performance analysis of a multi-arm cartesian robotic fruit harvester—Extensions of graph coloring. *J. Intell. Robot. Syst.* **2016**, *82*, 399–411. [\[CrossRef\]](#)
8. Zion, B.; Mann, M.; Levin, D.; Shilo, A.; Rubinstein, D.; Shmulevich, I. Harvest-order planning for a multiarm robotic harvester. *Comput. Electron. Agric.* **2014**, *103*, 75–81. [\[CrossRef\]](#)

9. Qu, H.R.; Su, W.H. Deep Learning-Based Weed–Crop Recognition for Smart Agricultural Equipment: A Review. *Agronomy* **2024**, *14*, 363. [\[CrossRef\]](#)
10. Fu, H.; Li, Z.; Zhang, W.; Feng, Y.; Zhu, L.; Fang, X.; Li, J. Research on Path Planning of Agricultural UAV Based on Improved Deep Reinforcement Learning. *Agronomy* **2024**, *14*, 2669. [\[CrossRef\]](#)
11. Hu, M.; Huang, Q.; Cai, J.; Chen, Y.; Li, J.; Shi, L. HPS-RRT*: An Improved Path Planning Algorithm for a Nonholonomic Orchard Robot in Unstructured Environments. *Agronomy* **2025**, *15*, 712. [\[CrossRef\]](#)
12. Wang, J.; Zhai, Z.; Xu, G.; Zhang, R.; Zhang, X.; Hu, R. Optimizing Sampling Points and Path Planning for Soil Monitoring in Agricultural Land. *Agronomy* **2024**, *14*, 2947. [\[CrossRef\]](#)
13. Latombe, J.C.; Lazanas, A.; Shekhar, S. Robot motion planning with uncertainty in control and sensing. *Artif. Intell.* **1991**, *52*, 1–47. [\[CrossRef\]](#)
14. De Giacomo, G.; Iocchi, L.; Nardi, D.; Rosati, R. Planning with sensing for a mobile robot. In Proceedings of the European Conference on Planning, Toulouse, France, 1 September 1997; pp. 156–168.
15. LaValle, S.M.; Hutchinson, S.A. An objective-based framework for motion planning under sensing and control uncertainties. *Int. J. Robot. Res.* **1998**, *17*, 19–42. [\[CrossRef\]](#)
16. Burns, B.; Brock, O. Sampling-based motion planning with sensing uncertainty. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 3313–3318.
17. Mann, M.P.; Zion, B.; Rubinstein, D.; Linker, R.; Shmulevich, I. Minimum time kinematic motions of a cartesian mobile manipulator for a fruit harvesting robot. *J. Dyn. Syst. Meas. Control.* **2014**, *136*, 051009. [\[CrossRef\]](#)
18. Duggal, V.; Sukhwani, M.; Bipin, K.; Reddy, G.S.; Krishna, K.M. Plantation monitoring and yield estimation using autonomous quadcopter for precision agriculture. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 5121–5127.
19. Edan, Y.; Rogozin, D.; Flash, T.; Miles, G.E. Robotic melon harvesting. *IEEE Trans. Robot. Autom.* **2000**, *16*, 831–835. [\[CrossRef\]](#)
20. Van Henten, E.; Van't Slot, D.; Hol, C.; Van Willigenburg, L. Optimal manipulator design for a cucumber harvesting robot. *Comput. Electron. Agric.* **2009**, *65*, 247–257. [\[CrossRef\]](#)
21. Schuetz, C.; Baur, J.; Pfaff, J.; Buschmann, T.; Ulbrich, H. Evaluation of a direct optimization method for trajectory planning of a 9-DOF redundant fruit-picking manipulator. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2660–2666.
22. Cruz-Ancona, C.D.; Martínez-Guerra, R.; Pérez-Pinacho, C.A. Generalized multi-synchronization: A leader-following consensus problem of multi-agent systems. *Neurocomputing* **2017**, *233*, 52–60. [\[CrossRef\]](#)
23. Williams, H.; Ting, C.; Nejati, M.; Jones, M.H.; Penhall, N.; Lim, J.; Seabright, M.; Bell, J.; Ahn, H.S.; Scarfe, A.; et al. Improvements to and large-scale evaluation of a robotic kiwifruit harvester. *J. Field Robot.* **2020**, *37*, 187–201. [\[CrossRef\]](#)
24. Somhom, S.; Modares, A.; Enkawa, T. Competition-based neural network for the multiple travelling salesmen problem with minmax objective. *Comput. Oper. Res.* **1999**, *26*, 395–407. [\[CrossRef\]](#)
25. Laporte, G. The traveling salesman problem: An overview of exact and approximate algorithms. *Eur. J. Oper. Res.* **1992**, *59*, 231–247. [\[CrossRef\]](#)
26. Lawler, E.L. The traveling salesman problem: A guided tour of combinatorial optimization. In *Wiley-Interscience Series in Discrete Mathematics*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 1985.
27. Rosenkrantz, D.J.; Stearns, R.E.; Lewis, P.M., II. An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.* **1977**, *6*, 563–581. [\[CrossRef\]](#)
28. Papadimitriou, C.H. The Euclidean travelling salesman problem is NP-complete. *Theor. Comput. Sci.* **1977**, *4*, 237–244. [\[CrossRef\]](#)
29. Helsgaun, K. An effective implementation of the Lin–Kernighan traveling salesman heuristic. *Eur. J. Oper. Res.* **2000**, *126*, 106–130. [\[CrossRef\]](#)
30. Kurtser, P.; Edan, Y. Planning the sequence of tasks for harvesting robots. *Robot. Auton. Syst.* **2020**, *131*, 103591. [\[CrossRef\]](#)
31. Cao, H.; Wang, H.; Li, Y.; Hamani, A.K.M.; Zhang, N.; Wang, X.; Gao, Y. Evapotranspiration partition and dual crop coefficients in apple orchard with dwarf stocks and dense planting in arid region, Aksu oasis, southern Xinjiang. *Agriculture* **2021**, *11*, 1167. [\[CrossRef\]](#)
32. Jamar, L.; Mostade, O.; Huyghebaert, B.; Pigeon, O.; Lateur, M. Comparative performance of recycling tunnel and conventional sprayers using standard and drift-mitigating nozzles in dwarf apple orchards. *Crop. Prot.* **2010**, *29*, 561–566. [\[CrossRef\]](#)
33. Weber, M. Optimizing the tree density in apple orchards on dwarf rootstocks. In Proceedings of the VII International Symposium on Orchard and Plantation Systems 557, Nelson, New Zealand, 30 January–5 February 2000; pp. 229–234.
34. Jia, W.; Mou, S.; Wang, J.; Liu, X.; Zheng, Y.; Lian, J.; Zhao, D. Fruit recognition based on pulse coupled neural network and genetic Elman algorithm application in apple harvesting robot. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881419897473. [\[CrossRef\]](#)
35. Jia, W.; Tian, Y.; Luo, R.; Zhang, Z.; Lian, J.; Zheng, Y. Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. *Comput. Electron. Agric.* **2020**, *172*, 105380. [\[CrossRef\]](#)

36. Kim, M.; Park, J. Learning collaborative policies to solve np-hard routing problems. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 10418–10430.
37. Christofides, N. Worst-case analysis of a new heuristic for the travelling salesman problem. *SIAM J. Comput.* **1976**, *6*, 699–711. [[CrossRef](#)]
38. Arora, S. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS), Burlington, VT, USA, 14–16 October 1996; pp. 2–11.
39. Kohonen, T. The self-organizing map. *Proc. IEEE* **1990**, *78*, 1464–1480. [[CrossRef](#)]
40. Zhu, D.Q.; Qu, Y.; Yang, S.X. Multi-AUV SOM task allocation algorithm considering initial orientation and ocean current environment. *Front. Inf. Technol. Electron. Eng.* **2019**, *20*, 330–341. [[CrossRef](#)]
41. Faigl, J. An application of self-organizing map for multirobot multigoal path planning with minmax objective. *Comput. Intell. Neurosci.* **2016**, *2016*, 2720630. [[CrossRef](#)]
42. Sampson, J.R. Adaptation in natural and artificial systems (John H. Holland). *SIAM Rev.* **1976**, *18*, 2. [[CrossRef](#)]
43. Liu, F.; Zeng, G. Study of genetic algorithm with reinforcement learning to solve the TSP. *Expert Syst. Appl.* **2009**, *36*, 6995–7001. [[CrossRef](#)]
44. Gan, Y.; Lan, L.W.; Li, S. Study on parameters configuration for ant colony optimization. *Adv. Mater. Res.* **2011**, *279*, 371–376. [[CrossRef](#)]
45. Aicardi, M.; Giglio, D.; Minciardi, R. Determination of optimal control strategies for TSP by dynamic programming. In Proceedings of the 2008 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 2160–2167.
46. Shi, X.H.; Liang, Y.C.; Lee, H.P.; Lu, C.; Wang, Q. Particle swarm optimization-based algorithms for TSP and generalized TSP. *Inf. Process. Lett.* **2007**, *103*, 169–176. [[CrossRef](#)]
47. Puspita, F.; Meitriova, A.; Yahdin, S. Mathematical modelling of traveling salesman problem (TSP) by implementing simulated annealing and genetic algorithms. *J. Phys. Conf. Ser.* **2020**, *1480*, 012029. [[CrossRef](#)]
48. Lin, Y.; Bian, Z.; Liu, X. Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing–tabu search algorithm to solve the symmetrical traveling salesman problem. *Appl. Soft Comput.* **2016**, *49*, 937–952. [[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.