sustainability

Article

Optimization of Storage Location Assignment for Non-Traditional Layout Warehouses Based on the Firework Algorithm

Xuan Zhang [1], Tiantian Mo [2,*] and Yougong Zhang [3]

[1] School of Management, Zhejiang University, Hangzhou 310058, China; xuan_zhang@zju.edu.cn
[2] College of Business, Jiaxing University, Jiaxing 314001, China
[3] Ocean College, Zhejiang University, Zhoushan 316021, China; zhangyougong123@gmail.com
* Correspondence: tiantianmo@zju.edu.cn

Abstract: With the development of logistics, sustainable warehousing has become increasingly important. To promote the warehousing efficiency, non-traditional layout warehouses and storage location assignments have been proposed separately. However, they are rarely combined. Taking inspiration from the advantages of non-traditional layout warehouses and storage location assignments, a storage location assignment optimization algorithm for non-traditional layout warehouses is proposed to further improve the efficiency and sustainability of warehousing. By reducing the picking distance and picking time, this algorithm further boosts the warehouse efficiency and sustainability, saving energy in the process and resulting in positive effects on the environment and the economy. In the process of establishing the model, taking the order-picking efficiency and shelf stability as optimizing objectives, a multi-objective optimization model is derived. Then, a storage location assignment optimization algorithm based on the firework algorithm is developed using adaptive strategies for explosion and selection to enhance the convergence rate and optimization performance of the algorithm. With this approach, the storage location assignment optimization for non-traditional layout warehouses can be handled well. Finally, a set of comparative simulations is carried out with MATLAB, and the results show various positive effects for sustainable warehouse management, such as a higher order-picking efficiency, better shelf stability, time and resource savings, and so on.

Keywords: sustainable warehousing management; storage location assignment; non-traditional layout warehouse; firework algorithm

Citation: Zhang, X.; Mo, T.; Zhang, Y. Optimization of Storage Location Assignment for Non-Traditional Layout Warehouses Based on the Firework Algorithm. *Sustainability* **2023**, *15*, 10242. https://doi.org/10.3390/su151310242

Academic Editors: Riccardo Patriarca, Giulio Di Gravio, Francesco Costantino and Ripon Kumar Chakrabortty

Received: 14 March 2023
Revised: 25 May 2023
Accepted: 26 June 2023
Published: 28 June 2023

1. Introduction

The warehouse plays a critical role in logistics and is considered one of its most significant components [1–3]. Ensuring its sustainability is important, as it impacts both economic and social factors and thus the overall sustainability of logistics [4,5]. The sustainability of a warehouse is crucial for its long-term viability. The resources that are utilized within the warehouse, such as space, equipment, and the workforce, are usually limited [6,7]. Without efficient resource utilization, order picking becomes unsustainable, resulting in increased energy consumption, capital expenditure, and human resources depletion.

The core activities of warehousing include receiving, storage location assignment, order picking, and shipping [8,9]. Several optimization strategies for making warehouses sustainable have been developed, such as warehouse layouts, storage location assignment, etc. [10,11]. Based on traditional layout warehouses, non-traditional layout warehouses were developed to decrease the pathways traveled to store and retrieve cargoes and reduce the energy cost [12,13]. The Flying-V warehouse layout [14,15], the Fishbone warehouse layout [16], the chevron, leaf, and butterfly warehouse layouts [17,18], and the straight diagonal cross-aisle non-traditional warehouse [19] are typical non-traditional warehouse layouts. The expected traveling distances for the Flying-V warehouse layout and the

Fishbone warehouse layout are up to 20% shorter than those of traditional warehouses, which contributes to energy saving [12]. Therefore, non-traditional layout warehouses are useful for improving the efficiency [13]. Moreover, storage location assignment is another practical strategy to improve the warehousing efficiency. Storage location assignment refers to the sustainable management of warehousing by reasonably optimizing the placement of cargoes, improving the order-picking efficiency, and reducing energy loss and resource waste [20]. Suitable storage location assignment can reduce the travel time and distance of picking robots, a practical strategy for improving the efficiency [21,22]. Therefore, this paper aims to answer the following research question:

How can a storage location assignment optimization algorithm be established for non-traditional layout warehouses to improve their efficiency and sustainability?

Although the advantages of non-traditional warehouse layouts and storage location assignments have been elaborated separately, they have rarely been combined. Additionally, storage location assignment for non-traditional layout warehouses has not been extensively considered. To address this gap, inspired by the superiority of non-traditional warehouse layouts and storage location assignments, these factors are integrated to provide a storage location allocation optimization algorithm for warehouses with non-traditional layouts to improve the storage efficiency. When assigning storage locations for non-traditional layout warehouses, the establishment of an optimization model and the design of an optimization algorithm are the main challenges. Firstly, to overcome these challenges, a multi-objective optimization model that considers both the order-picking efficiency and shelf stability as optimizing objectives is established. Subsequently, a storage location assignment optimization algorithm based on the Firework algorithm is proposed. Specifically, adaptive strategies are adopted in the explosion and selection stages, which enhance the convergence rate and optimization performance of the algorithm. Therefore, storage location assignment optimization for non-traditional layout warehouses can be effectively handled.

1.1. Literature Review

To make warehouses more sustainable, numerous studies have been conducted to optimize resource utilization, including routing, scheduling, storage location assignment, and other methods. Prior research developed Key Performance Indicators (KPIs) for assessing the sustainability of the warehousing performance, including economic, environmental, and social variables [5,7]. Specifically, economic variables include the warehouse operation performance and economic performance, while environmental variables include the resource allocation, emissions waste, and environmental commitments. Social variables include labor practices, decent work, and product responsibility. Chiang et al. [23] developed a picking-list assignment strategy that groups similar items together to reduce the traveling distance and time for picking robots. This leads to an increased efficiency and a reduction in carbon emissions, contributing to a more sustainable supply chain. On the other hand, Popovic et al. [7] focused on workforce scheduling problems to decrease the labor costs. In addition, Burinskiene et al. [24] increased the efficiency of warehouse procedures by identifying wasteful warehouse processes and reducing the replenishment and order-picking costs. This paper improves warehousing sustainability by using a novel strategy. Non-traditional storage layouts and storage location assignments are comprehensively considered to achieve sustainable warehousing management. Specifically, a storage location assignment optimization algorithm for non-traditional layout warehouses is proposed, which improves the picking efficiency and increases the shelf stability. This enhances energy conservation in warehousing, promoting both environmental and economic benefits.

The design of a storage location assignment optimization algorithm must consider both the shelf stability and the picking efficiency, making it a multi-objective model [25]. To address multi-objective optimization, several algorithms have been developed, such as the firework algorithm (FWA), the genetic algorithm (GA), the particle swarm optimization algorithm (PSO), and the polynomial algorithm [26–29]. Zhang et al. [30] proposed a GA with a two-stage iterative approach to develop a layout that considers the adjacency and other constraints with the lowest transportation cost. For storage location formation, Li et al. [31] proposed a multi-objective model and an improved GA considering the order-picking frequency and shelf stability based on the class storage policy. Chen et al. [32] presented an established neighborhood structure for storage location assignment problems and created a tabu search algorithm. Zhang et al. [33] expressed this as an integer programming model and created the simulated annealing algorithm. In view of the storage location assignment problem with a Flying-V layout, an approach to the storage location assignment problem based on the Flying-V layout was proposed by Liu et al. [34]. Hu et al. [35] formulated an optimization model for the storage location assignment, considering the inventory efficiency and shelf stability as optimizing objectives based on Fishbone layout characteristics. Soheyl et al. [36] proposed the Multi-Objective Stochastic Fractal Search (MOSFS) to solve complex multi-objective optimization problems. With the consideration of uncertain parameters, objective functions, and constraints, a mathematical model was designed by Soheyl et al. [25]. Additionally, several artificial-intelligence-based solution techniques have been formulated to solve the complex nonlinear problem. In this paper, a practical multi-objective optimization model for quantifying the warehousing sustainability is proposed by considering the characteristics of the storage location assignment, order-picking efficiency, and shelf stability as optimizing objectives.

1.2. Main Contributions

Although several algorithms have been employed to solve the storage location assignment problem, few of them consider the modeling of non-traditional layouts and multi-objective optimization as an integrated challenge. Therefore, storage location assignment for non-traditional layout warehouses remains a challenging task. In this paper, a storage location assignment for non-traditional warehouse layouts based on the FWA is proposed. The contributions are listed below:

(a) Establishing a model for non-traditional layout warehouses can be challenging. In this paper, a model of non-traditional layout warehouses is established in detail, which consists of a Flying-V layout and a Fishbone layout.
(b) A practical multi-objective optimization model is proposed to quantify the sustainability of warehousing. Specifically, the characteristics of the storage location assignment, order-picking efficiency, and shelf stability are taken as optimizing objectives, and a multi-objective optimization model is proposed.
(c) To address the multi-objective optimization model described above, a storage location assignment optimization algorithm based on the FWA is developed. Adaptive strategies are adopted for explosion and selection to improve the convergence rate and optimization performance of the algorithm.

Therefore, the storage location assignment optimization of non-traditional layout warehouses can be handled well. Furthermore, to verify the effectiveness and priority of the proposed algorithm, comparative simulations are implemented, which indicate a faster convergence rate and better optimization performance.

The structure of this paper is as follows: Section 2 describes the modeling of non-traditional warehouse layouts, including the Flying-V layout and Fishbone layout. Section 3 describes the modeling of storage location assignment optimization with integrated consideration of multiple optimizing objectives. Next, Section 4 describes the design of the storage location assignment algorithm based on the FWA. To prove the priority of the proposed algorithm, the comparative GA is described in Section 5. Moreover, Section 6 presents comparative simulations of different storage location assignment algorithms for different non-traditional warehouse layouts, which verifies the significant priority of the proposed algorithm. Finally, the contributions and future research directions are summarized in Section 7.

2. The Modeling of Non-Traditional Warehouse Layouts

The object of this research was to optimize storage location assignment for non-traditional warehouse layouts. In this section, the modeling of non-traditional warehouse layouts is derived, including the Flying-V layout and Fishbone layout, as shown in Figures 1 and 2.

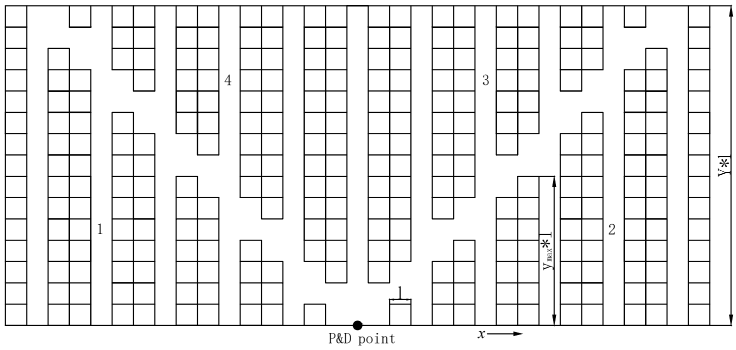

Figure 1. Flying-V layout.

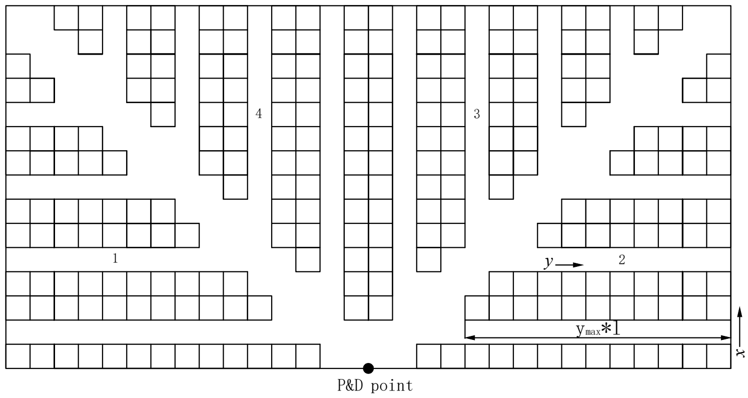

Figure 2. Fishbone layout.

Before carrying out the modeling of non-traditional warehouse layouts, we assume that [34,35]:

(a) The numbering, weight, and access frequency of cargoes are known;
(b) The same kind of cargo can be stored in different storage locations.
(c) The horizontal speed and vertical speed of the picking robot are known, and its starting and braking processes can be ignored;
(d) During the picking process, the picking robot can only access one storage location every time;
(e) The width of the picking roadway is equal to the width of a shelf.

Moreover, some related parameters can be described as follows: The notations in this paper are defined in Table 1. The length of the storage space is l, the height of each shelf layer is h, and the storage area is k $(k = 1, 2, 3, 4)$. Starting from the lower left corner, the area is divided into area 1, area 2, area 3, and area 4 in a counterclockwise direction. Area 3 and area 4 are the middle parts of Figures 1 and 2. x $(x = 1, 2, \ldots, x_{\max})$ is the row number of the storage location, y $(y = 1, 2, \ldots, y_{\max})$ is the column number of the storage location, z $(z = 1, 2, \ldots, z_{\max})$ is the number of layers in the storage location, and i is the number of cargoes. The cargo located in row x, column y, and floor z in zone k is marked as $(kxyz)$, and r_i is the access frequency of the cargoes. v_1 is the horizontal speed of the picking robot, and v_2 is its vertical speed.

Table 1. The definitions of the notations.

Notations	Definitions
l	The length of the storage space
h	The height of each shelf layer
k	The storage area
x	The row number of the storage location
y	The column number of the storage location
z	The number of layers in the storage location
i	The number of cargoes
r_i	The access frequency of the cargoes
v_1	The horizontal speed of the picking robot
v_2	The vertical speed of the picking robot
L_x	The distance from the $P\&D$ point to the storage location
f_1	Objective function 1
f_2	Objective function 2
f	The overall objective function
g	The fitness function
F_1,F_2	The sub-objective function
w_1	The weight of sub-objective function F_1
w_2	The weight of sub-objective function F_2
$\hat{f}_1$	The optimal value of the efficiency of warehouses for storage and retrieval (7)
$\hat{f}_2$	The optimal value of the center of gravity of all cargoes (8)
m_i	The weight of cargoes numbered i
j	The number of fireworks
s_j	The number of sparks numbered j
n	The maximum amount of fireworks
$f_{\max}$	The maximum value of the objective function among n fireworks
N_0	A parameter controlling the total number of sparks generated by n fireworks
ξ	The smallest constant in the computer
$\hat{s}_j$	The bounds for s_j
a,b	The constant parameters
A_j	The amplitude of explosion for each firework
$\hat{A}$	The maximum explosion amplitude
$f_{\min}$	The minimum value of the objective function among n fireworks
q_e^u	The location of each spark
w	The random dimensions of sparks
d	The dimensionality of firework q_j
q_j	The location of the firework
x^*	The current best location
$p\left(x_j\right)$	The selection probability of each firework location
K	The set of all current locations of both fireworks and sparks
$R\left(q_j\right)$	The distance between a location q_j and other locations q_e
i_{max}	The number of cargoes
T	The maximum evolutionary generation
N	The population of the GA

2.1. Model of the Flying-V Warehouse Layout

As shown in Figure 1, the entire warehouse has four equal distribution areas, one $P\&D$ point, and two diagonal cross-aisles, and the shelves are arranged in the Flying-V layout.

The maximum columns $y_{\max}$ of the shelf change continuously with x, and can be derived as

(a) When $k = 1$ or $k = 2$:

$$y_{\max} = \begin{cases} 1.5x - 0.5, x \text{ is odd} \\ 1.5x, x \text{ is even} \end{cases} \tag{1}$$

(b) When $k = 3$ or $k = 4$

$$y_{\max} = \begin{cases} Y - 1.5x - 0.5,\ x \text{ is odd} \\ Y - 1.5x,\ x \text{ is even} \end{cases} \tag{2}$$

where Y represents the maximum number of rows of shelves in the warehouse.

L_x is the travel distance of the picking robot from the $P\&D$ point to the shelf where the cargoes are located, and its expression is

(a) When $k = 1$ or $k = 2$:

$$L_x = \begin{cases} \sqrt{2} * (1 + 1.5(x-1)) * l,\ x \text{ is odd} \\ \left(\sqrt{2} * 1.5x + 1\right) * l,\ x \text{ is even} \end{cases} \tag{3}$$

(b) When $k = 3$ or $k = 4$

$$L_x = \begin{cases} \left(\sqrt{2} * 1.5(x-1) + 2\right) * l,\ x \text{ is odd} \\ \left(\sqrt{2} * 1.5(x-1) + 1\right) * l,\ x \text{ is even} \end{cases} \tag{4}$$

2.2. Model of the Fishbone Warehouse Layout

As shown in Figure 2, similar to the Flying-V warehouse layout, the model of the Fishbone warehouse layout can be derived as follows:

The maximum number of columns $y_{\max}$ for the shelf changes continuously with x, and can be derived as

$$y_{\max} = \begin{cases} Y - 1.5(x-1),\ x \text{ is odd} \\ Y - 1.5x + 1,\ x \text{ is even} \end{cases} \tag{5}$$

L_x is the travel distance of the picking robot from the $P\&D$ point to the shelf where the cargoes are located, and it can be derived as

$$L_x = \begin{cases} \sqrt{2} * (1 + 1.5(x-1)) * l + l,\ x \text{ is odd} \\ \sqrt{2} * (2 + 1.5(x-2)) * l + 2l,\ x \text{ is even} \end{cases} \tag{6}$$

3. Model of the Storage Location Assignment Optimization

The optimization of storage locations is conducted to assign suitable storage locations for cargoes based on their characteristics, i.e., weight and picking frequency [19], which is helpful to sustainable warehousing management. The picking frequency varies among different types of cargo. To improve the picking efficiency, the picking time for all cargoes should be minimized, which can be achieved by calculating the sum of the product of the picking efficiency and the picking time of each cargo. Although existing research has considered the warehouse efficiency for storage and retrieval [37,38], the shelf stability, which is influenced by the weight of each cargo, has not been sufficiently considered. To promote shelf stability, the overall center of gravity of the cargo should be maintained as low as possible. Therefore, multiple optimizing objectives for non-traditional warehouse layouts are fully considered in this paper, such as the stability of the shelf and the efficiency of storing and retrieving cargoes. The multi-optimization model for storage location assignment optimization can be derived as follows:

Objective function:

$$f_1 = \min \sum_{i=1}^{i_{\max}} r_i * \left(\frac{L_x}{v_1} + \frac{(y-1) * l}{v_1} + \frac{(z-1) * h}{v_2}\right) \tag{7}$$

$$f_2 = \min \frac{\sum_{i=1}^{i_{\max}} m_i * z * h}{\sum_{i=1}^{i_{\max}} m_i} \tag{8}$$

where

$$\begin{cases} x \leq x_{\max} \\ y \leq y_{\max} \\ z \leq z_{\max} \end{cases} \tag{9}$$

Equation (7) represents the objective function established through the efficiency of warehouses for storage and retrieval, and Equation (8) represents the objective function established using the center of gravity of all cargoes. Equation (9) represents the constraints of storage location assignment in non-traditional warehouse layouts, and m_i is the weight of cargoes numbered i.

For multi-objective optimization problems, many solutions have been proposed, in which the weight method is a widely utilized one [39]. For storage location assignment, the dimensions and ranges of the objective function (7) and (8) are quite different. Therefore, the weight method cannot be directly used, which results in certain objective values being weakened. To solve this issue, the dimension of each single objective function is normalized in this paper, using the optimal value of each single objective function. Thus, the multi-objective problem is transformed into a single-objective problem. The overall objective function f and the fitness function g are derived as follows:

$$f = w_1 F_1 + w_2 F_2 \tag{10}$$

$$g = \frac{1}{f} \tag{11}$$

where w_1 and w_2 represent the weights of two sub-objective functions. Sub-objective functions F_1 and F_2 can be obtained by dimensional normalization:

$$F_1 = \frac{\hat{f}_2}{\hat{f}_1 + \hat{f}_2} f_1 \tag{12}$$

$$F_2 = \frac{\hat{f}_1}{\hat{f}_1 + \hat{f}_2} f_2 \tag{13}$$

where $\hat{f}_1$ represents the optimal value for the efficiency of warehouses for storage and retrieval (7), and $\hat{f}_2$ represents the optimal value for the center of gravity of all cargoes (8).

4. Algorithm Design with the Firework Algorithm

Proposed by Tan and Zhu [40], the FWA has been widely applied for optimization due to its advantages. For example, it has been used successfully to optimize the local-concentration model's parameters for spam detection [41], and for a Gaussian process regression model for determining the WiFi indoor location [42]. As shown in Figure 3, a storage location assignment algorithm for non-traditional warehouse layouts based on the FWA is proposed in this paper. The algorithm mainly consists of four steps: explosion, mutation, evaluation, and selection. In particular, the location of a firework represents a candidate solution to the storage location assignment for non-traditional warehouse layouts, and an explosion represents a random search operation in the solution space around the firework. The main steps of the proposed algorithm are described as follows:

(a) Firstly, inspired by the phenomenon of firework explosion, a certain number of firework locations are generated in the search space, which will generate a set of sparks by exploding.
(b) Secondly, the location of sparks is obtained by explosion and mutation. A firework with higher fitness can explode with a greater number of sparks with a smaller amplitude, while a firework with lower fitness can explode with fewer sparks with a larger amplitude.
(c) Thirdly, the quality of each firework location is derived with the fitness function (11).

(d) Then, the fireworks and sparks with high fitness are selected as the locations (candidate solutions) for the next generation's fireworks.
(e) Finally, optimization ends when the maximum number of evaluations is reached.

Moreover, to better illustrate the design process of the proposed algorithm, some key parts are described in detail below.

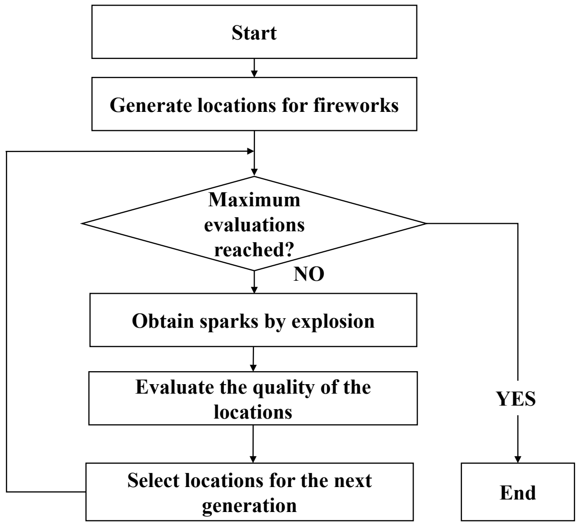

Figure 3. The framework of the proposed algorithm based on the FWA.

4.1. Number of Sparks

The number of sparks depends on the quality of each firework and can be derived as follows.

$$s_j = N_0 \cdot \frac{f_{\max} - f + \xi}{\sum_{j=1}^{n} (f_{\max} - f) + \xi} \tag{14}$$

where j is the number of fireworks. f is the overall objective function (10). $f_{\max}$ is the maximum value of the objective function among n fireworks. N_0 is a parameter controlling the total number of sparks generated by n fireworks. ξ denotes the smallest constant in the computer, which is utilized to avoid a zero-division error. To avoid the overwhelming effects of splendid fireworks, bounds for s_j are designed as shown in (15).

$$\hat{s}_j = \begin{cases} round(a \cdot N_0) & s_j < aN_0 \\ round(b \cdot N_0) & s_j > bN_0, a < b < 1 \\ round(s_j) & otherwise \end{cases} \tag{15}$$

where a and b are constant parameters.

4.2. Amplitude of Explosion

The amplitude of explosion for each firework can be derived as follows: In contrast to the design of the spark number, the amplitude of a good firework explosion is smaller than that of a bad one.

$$A_j = \hat{A} \cdot \frac{f - f_{\min} + \xi}{\sum_{j=1}^{n} (f - f_{\min}) + \xi} \tag{16}$$

where $\hat{A}$ denotes the maximum explosion amplitude, and $f_{\min}$ is the minimum value of the objective function among n fireworks.

4.3. Obtaining Sparks by Explosion

The location of each spark q_e^u generated by q_j^u can be obtained by randomly setting w dimensions ($1 \le e \le s_j, 1 \le u \le w$), which is calculated by

$$q_e^u = q_j^u + A_j \cdot rand(-1, 1) \tag{17}$$

where w represents the random dimensions of sparks, $w = round(d \cdot rand(0, 1))$, and d is the dimensionality of firework q_j.

Moreover, to maintain the diversity of the sparks, a Gaussian distribution with a mean of 1 and standard deviation of 1 is utilized to define the coefficient of the explosion. A certain number of sparks are generated in each explosion generation.

4.4. Selection of Locations

At the beginning of each explosion generation, the current best location x^* is always kept for the next explosion generation. After that, $n - 1$ locations are selected based on their distances to other locations to maintain the diversity of the sparks. The next generation of fireworks is selected using the roulette method with the selection probability [43,44]. The selection probability of each firework location q_j can be derived as follows:

$$p(x_j) = \frac{R(q_j)}{\sum_{e \in K} R(q_e)} \tag{18}$$

where K is the set of all current locations of both fireworks and sparks. $R(q_j)$ represents the distance between a location q_j and other locations q_e, which can be derived as follows:

$$R(q_j) = \sum_{e \in K} d(q_j, q_e) = \sum_{e \in K} \|q_j - q_e\| \tag{19}$$

As the evaluations reach the desired evaluation point, the optimal storage location assignment can be obtained.

5. Genetic Algorithm

To make the performance superiority of the proposed storage-location-assignment-based algorithm on the FWA more convincing, the genetic algorithm (GA) was selected as a comparative object. The GA is widely used to solve combinatorial optimization problems [28]. However, in the actual application process of the traditional GA, the phenomenon of prematurity often occurs in the early stage of evolution, and the phenomenon of slow convergence often occurs in the later stage of evolution [45–47]. The deficiencies can be effectively solved and the optimization performance can be improved by the adaptive mechanism. Therefore, an adaptive strategy is implemented among the selection, crossover, and mutation operations of the genetic algorithm. The framework of GA is shown in Figure 4.

Inputs: $i_{\max}$ (number of cargoes), m_i (weight of cargoes), r_i (picking frequency of cargoes), v_1 (horizontal speed of the picking robot), v_2 (vertical speed of the picking robot), l (length of the storage location), h (height of each shelf layer), w_1, w_2 (weight of the two sub-objective functions), and N (population of the GA).

Output: optimal assignment of storage locations for non-traditional warehouse layouts.

Step 1. Input the parameters of the storage location assignment $i_{\max}$, m_i, r_i, v_1, v_2, l, h, w_1, w_2.
Step 2. Initialize the adaptive genetic algorithm parameters.
Step 3. Start the algorithm and initialize the population.
Step 4. Determine whether the number of iterations has been reached. If so, go to Step 5; otherwise, continue.

Step 4.1. Calculate the objective function value and the fitness of the individuals in the population.
Step 4.2. Select: Adaptively transform the fitness value.
Step 4.3. Retain the optimal individual.
Step 4.4. Crossover: Carry out an adaptive transformation of the crossover rate.
Step 4.5. Mutation: Carry out an adaptive transformation of the mutation rate.

Step 5. End of the algorithm: The optimal assignment of storage locations for non-traditional warehouse layouts can be obtained.

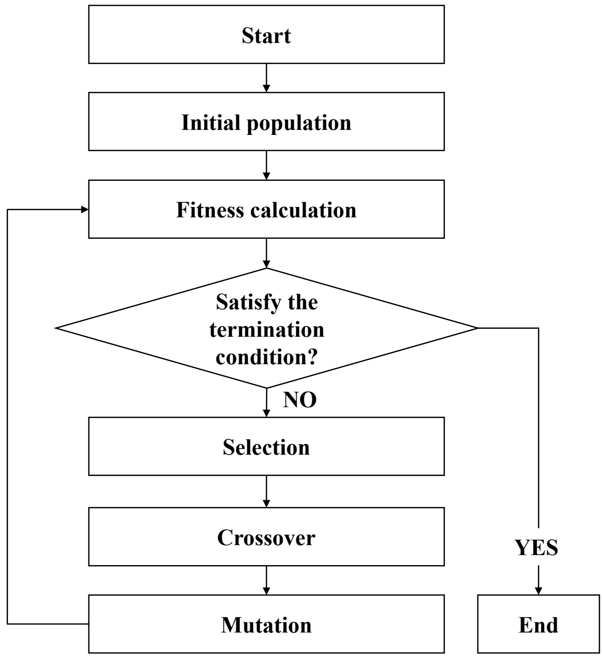

Figure 4. The framework of the GA.

6. Simulation

6.1. Simulation Setup

To describe and verify the optimized performance of the proposed algorithm, two typical non-traditional warehouse layouts, i.e., the Flying-V layout and Fishbone layout, were selected as the research objects for the comparative simulation. The information about the cargoes is shown in Table 2. It was provided by an automobile parts manufacturer. All parameters in the simulation are expressed according to the International System of Units (SI).

To make the performance comparison of the optimized algorithm more convincing, GA and FWA were selected as comparative objects for this simulation. The parameters of these algorithms were selected with the overall consideration of the operating frequency range of the storage location assignment and the response time of the algorithms.

$A1$: GA. The framework of the GA is shown in Figure 4. The primary parameters were specified as follows:

The maximum evolutionary generation was set to $T = 1000$, and the population was set to $N = 100$.

$A2$: For the proposed algorithm based on the FWA, the primary parameters were specified as follows:

The maximum evolutionary generation was set to $T = 100$, the initial firework number was $n = 200$, $a = 0.001$, $b = 0.999$, $N_0 = 20$, and $\hat{A} = 20$.

To verify the performance levels of these comparative algorithms with different non-traditional warehouse layouts, two simulations were designed to reflect the storage location assignment performance to a certain extent.

SET1: Flying-V warehouse layout.

SET2: Fishbone warehouse layout.

Table 2. Information about the cargoes.

Number	Weight	Frequency	Location Used
1	13	2	2
2	27	19	1
3	29	15	1
4	15	5	3
5	28	10	3
6	37	19	3
7	17	15	1
8	40	6	1
9	23	13	4
10	18	8	1
11	29	5	1
12	13	11	4
13	22	1	1
14	36	20	3
15	14	4	2
16	21	4	3
17	19	12	1
18	39	20	3
19	20	16	3
20	37	1	1
21	32	5	4
22	20	7	3
23	34	1	2
24	18	8	1
25	31	15	2
26	30	19	1
27	28	7	4
28	35	4	3
29	27	4	3
30	29	12	2
31	34	12	4
32	33	8	2
33	33	15	4
34	19	13	1
35	37	15	2
36	19	18	1
37	36	17	1
38	40	4	3
39	22	18	1
40	11	3	2

6.2. Simulation of SET1

To compare the optimization performances of these storage location assignment algorithms for the Flying-V warehouse layout, simulation SET1 was designed as shown in Figure 1. The maximum number of rows of storage locations in the 1st and 2nd areas is $x_{\max} = 10$, and that in the 3rd and 4th areas is $x_{\max} = 9$. The length of the storage space l is 1 m, and the height of each shelf layer h is 0.8 m; the maximum number of shelf layers is $z_{\max} = 4$, the horizontal speed of the picking robot is $v_1 = 2$ m/s, and the vertical speed is $v_2 = 0.5$ m/s. The weight of two sub-objective functions, w_1 and w_2, is 50.0%.

For the comparative simulation conducted in SET1, the simulation results are shown in Figures 5 and 6. The average and optimal values of the optimizing objectives for each generation in the iterative process of the FWA are shown in Figure 5. The average and optimal values of the optimizing objectives for each generation in the iterative process of the GA are shown in Figure 6. Moreover, a performance comparison of the two algorithms is shown in Table 3. Optimal solutions for the proposed algorithm and GA are shown in Tables 4 and 5.

Table 3. Performance comparison of these algorithms for SET1.

Algorithm	Init	Converge	Promote	Generation	Time
GA	382.7	253.7	33.7%	558	76.89
FWA	336.4	189.6	43.6%	178	4.55

Table 4. The optimal solution of the proposed algorithm for SET1.

Number	Location	Number	Location
1	3152, 1441	21	4641, 2212, 2411, 2312
2	1821	22	1211, 1444, 1322
3	1511	23	2511, 1641
4	3131, 2211, 1512	24	1651
5	1451, 2631, 1611	25	1231, 3212
6	1111, 4731, 2611	26	2221
7	1622	27	1731, 3222, 1221, 3411
8	1621	28	3221, 1811, 1411
9	1921, 1112, 1911, 2111	29	1214, 1881, 1213
10	2711	30	3311, 1561
11	2721	31	2051, 3111, 2431, 1851
12	2421, 3191, 1541, 2412	32	3211, 2112
13	3271,	33	2112, 1531, 3321, 1412
14	3122, 4151, 1612	34	2311
15	1721, 3642	35	1711, 3611
16	3321, 1513, 1912	36	1521
17	1631	37	1341
18	1313, 3511, 1321	38	1883, 1801, 1311
19	1671, 1113, 1431	39	1312
20	1212	40	2521, 2113

Table 5. The optimal solution of the GA for SET1.

Number	Location	Number	Location
1	2831, 2213	21	2922, 3531, 3232, 3491
2	3071	22	2341, 3411, 4423
3	1322	23	1931, 3571
4	3624, 2963, 1213	24	4741
5	1791, 2682, 2732	25	3171, 1734
6	2111, 1731, 3111	26	2211
7	4651	27	1962, 3331, 4211, 1332
8	3651	28	4481, 2661, 4281
9	4211, 3433, 3231, 4633	29	3322, 2054, 1523
10	3032	30	1573,2121
11	3612	31	3212, 3371,1331, 4392
12	1221, 4722, 3493,3531	32	1912, 1771
13	2231	33	2611, 1512, 3274, 4531
14	4231, 2331, 4431	34	3452
15	1513, 3831	35	4173, 3202
16	3311, 2741,4712	36	1422
17	1431	37	1671
18	4631, 2312, 3121	38	4351, 2903, 3411
19	1892, 4163, 2112	39	3281
20	1422	40	4161, 3451

As can be seen from Figures 5 and 6 and Table 3, the optimal and the average values of the objective function show a gradual downward trend in the iterative process. According to the simulation results of the GA algorithm presented in Figure 6, when the iteration exceeds 558 generations, the optimal value of the objective function tends to converge, the average objective function value of the initial population is 382.7, and the average objective function value after algorithm optimization and convergence is 253.7. The optimization effect increases by 33.7%. According to the simulation results of the FWA shown in Figure 5, when the number of iterations exceeds 178, the optimal value of the objective function tends to converge, the average objective function value of the initial population is 336.4, and the average objective function value after algorithm optimization and convergence is 189.6. The optimization effect increases by 43.6%. To make a comparison of the computational complexity, the convergence times of these algorithms were calculated. The convergence time of the proposed algorithm was 4.55 s, and the convergence time of the GA was 76.89 s. Moreover, the optimization performance of the proposed algorithm for at least 30 different size (small, medium, and large) instances is provided in Tables 6–8, which verifies the applicability of the proposed model and its solution procedure.

Table 6. The optimization performance of the proposed algorithm for SET1 at a small scale.

Number	Init	Converge	Promote	Generation
1	169.5	74.6	56.0%	161
2	195.6	84.4	56.8%	262
3	176.0	78.2	55.6%	229
4	165.6	75.9	54.2%	112
5	170.1	77.1	54.6%	139
6	177.8	78.1	56.1%	177
7	169.7	79.8	53.0%	152
8	171.5	79.5	53.7%	137
9	171.6	74.9	56.3%	249
10	176.6	80.6	54.3%	144

Table 7. The optimization performance of the proposed algorithm for SET1 at a medium scale.

Number	Init	Converge	Promote	Generation
1	317.7	185.6	41.6%	138
2	359.5	202.1	43.8%	184
3	347.1	191.7	44.8%	217
4	351.1	201.6	42.3%	166
5	349.3	194.1	44.4%	185
6	339.8	202.1	40.5%	157
7	346.2	198.5	42.7%	194
8	353.2	197.2	44.2%	205
9	318.1	181.3	43.0%	132
10	358.2	212.7	40.6%	118

Table 8. The optimization performance of the proposed algorithm for SET1 at a large scale.

Number	Init	Converge	Promote	Generation
1	719.2	484.6	32.6%	174
2	723.3	460.5	36.3%	403
3	682.3	454.0	33.5%	176
4	723.3	477.4	34.0%	236
5	697.3	468.3	32.8%	148
6	746.7	469.2	37.2%	421
7	722.2	476.6	34.0%	214
8	702.3	448.0	36.2%	334
9	725.1	464.1	36.0%	323
10	763.2	514.6	32.6%	133

To verify the performance of the proposed algorithm, the well-known commercial software CPLEX 12.10 was used to solve this model for different instances of varying scales, as presented in Table 9. The results show that, for small- and medium-scale instances, the FWA can obtain reasonable solutions compared to those generated by CPLEX, with an average gap of less than 8.51%. In terms of the calculating time, the FWA requires significantly less time to calculate instances compared to CPLEX. For large-scale instances, the FWA can rapidly find a solution, whereas CPLEX cannot find a feasible solution within a reasonable time window. As such, the proposed algorithm can effectively improve the solving efficiency of complex models and ensure the solution quality.

Table 9. Comparison between the proposed algorithm and CPLEX for SET1.

Scale	FWA	FWA Time	CPLEX	CPLEX Time	Error
small	78.31	2.36 s	74.58	798.4 s	5.00%
medium	196.69	4.55 s	181.27	3485.3 s	8.51%
large	471.73	13.59 s	NAN	NAN	NAN

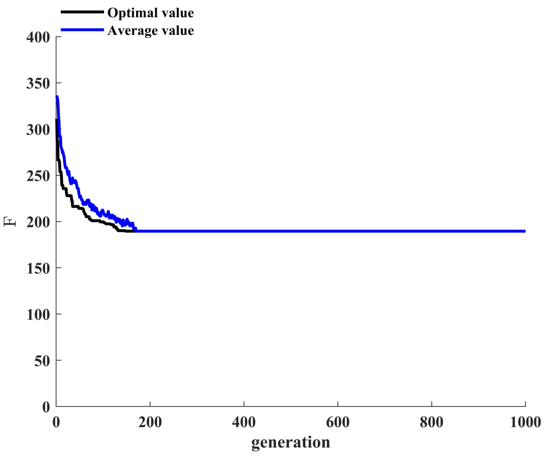

Figure 5. The object value of the proposed FWA algorithm for SET1.

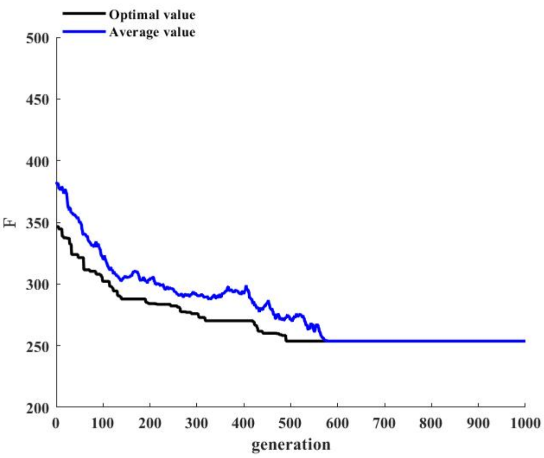

Figure 6. The object value of the proposed GA algorithm for SET1.

Through the performance comparison mentioned above, the response speed of the proposed algorithm was shown to be quicker than that of GA, and the convergence performance was better than that of the GA. Therefore, the proposed algorithm is more suitable for storage location assignment for Flying-V layout warehouses. The results demonstrate that the proposed optimizing algorithm effectively increased the sustainability of the warehouses. Specifically, the energy consumption needed for picking robots decreased by lowering the center of gravity of the cargo storage location assignment and increasing the picking efficiency. For the Flying-V layout, it is a practical storage location assignment optimizing algorithm.

6.3. Simulation of SET2

To compare the optimization performances of these storage location assignment algorithms with the Fishbone warehouse layout, the SET2 simulation was designed as shown in Figure 2. The maximum number of rows of storage space is $x_{\max} = 9$. The length of the storage space is $l = 1$ m. The height of each shelf layer is $h = 0.8$ m. The maximum number of shelf layers is $z_{\max} = 4$. The horizontal speed of the picking robot is $v_1 = 2$ m/s, the vertical speed is $v_2 = 0.5$ m/s, and the weight of two sub-objective functions, w_1 and w_2, is 50.0%.

For the comparative simulation performed with SET2, the simulation results are shown in Figures 7 and 8. The average and optimal values of the optimizing objective of each generation in the iterative process of the FWA are shown in Figure 7. The average and optimal values of the optimizing objectives for each generation in the iterative process of the GA are shown in Figure 8. Moreover, a performance comparison of the two algorithms is shown in Table 10. The optimal solutions for the proposed algorithm and the GA are shown in Tables 11 and 12.

As can be seen from Figures 7 and 8 and Table 10, the optimal and the average values of the objective function show a gradual downward trend in the iterative process. According to the simulation results of the GA presented in Figure 8, when the number of iterations exceeds 647 generations, the optimal value of the objective function tends to converge, the average objective function value of the initial population is 308.2, and the average objective function value after optimization and convergence is 217.1. The optimization effect increases by 29.6%. According to the simulation results of the FWA shown in Figure 7, when the number of iterations exceeds 176, the optimal value of the objective function tends to converge, the average objective function value of the initial population is 278.8, and the average objective function value after optimization and convergence is 143.3. The optimization effect increases by 48.6%. The convergence time of the proposed algorithms is 3.41 s, and the convergence time of the GA is 41.17 s. Moreover, the optimization performance of the proposed algorithm for at least 30 different size (small, medium, and large) instances is provided in Tables 13–15, which verifies the applicability of the proposed model and its solution procedure.

Futhermore, CPLEX was utilized to solve this model, as shown in Table 16. The results show that, for small- and medium-scale instances, the FWA can obtain reasonable solutions compared to those generated by CPLEX, with an average gap of less than 8.19%. In terms of the calculating time, the FWA requires less computing time compared to CPLEX. For larger-scale instances, the FWA can rapidly find a solution, whereas CPLEX cannot find a feasible solution. As such, the proposed algorithm can effectively improve the solving efficiency of complex models while ensuring the solution quality.

Through the performance comparison mentioned above, the response speed of the proposed algorithm was shown to be quicker than that of the GA, and the convergence performance was better than that of the GA. Therefore, the proposed algorithm is more suitable for storage location assignment for Fishbone layout warehouses. The results show that the proposed optimizing algorithm improves the sustainability of warehouses. Specifically, by lowering the center of gravity of the cargo distribution and improving the picking efficiency, the energy consumption required for picking robots is effectively reduced. It is a sustainable storage location assignment algorithm for the Fishbone layout.

According to the above analysis of the simulation results for SET1 and SET2, compared with the GA, the priority and effectiveness of the proposed storage location assignment algorithm for non-traditional warehouse layouts were verified. Theoretically, this paper contributes to sustainable warehousing by combining the superiority of non-traditional warehouse layouts and storage location assignments. In this way, a storage location assignment optimization algorithm for non-traditional layout warehouses is provided.

Table 10. Performance comparison of these algorithms for SET2.

Algorithm	Init	Converge	Promote	Generation	Time
GA	308.2	217.1	29.6%	647	41.17 s
FWA	278.8	143.3	48.6%	176	3.41 s

Table 11. The optimal solution of the proposed algorithm for SET2.

Number	Location	Number	Location
1	3151, 1131	21	2341, 1113, 1312, 1511
2	2122	22	1141, 4141, 1152
3	1151	23	4111, 2161
4	1213, 2212, 3112	24	3211
5	2131, 1411, 1241	25	1172, 2331
6	1281, 1181, 2171	26	2311
7	2241	27	1212, 3131, 1421, 1711
8	1271	28	2221, 1611, 2141
9	2151, 1231, 1612, 1171	29	1811, 2231, 1221
10	1381	30	1261, 1112
11	1162	31	1161, 1123, 1251, 1512
12	2113, 1114, 1322, 3311	32	2611, 2411
13	3631	33	4121, 1122, 3111, 1311
14	1132, 3231, 1121	34	2121
15	1153, 1431	35	1211, 2111
16	3121, 2201, 1412	36	2112
17	1222	37	1341
18	2211, 1211, 1142	38	1321, 2321, 2511
19	1191, 1331, 3411	39	1111
20	3152	40	1232, 2114

Table 12. The optimal solution of the GA for SET2.

Number	Location	Number	Location
1	2234, 1142	21	3511, 1291, 4292, 1641
2	2241	22	1293, 3623, 4232
3	2112	23	4552, 3552
4	2192, 2173, 1621	24	3272
5	4451, 1321, 3724	25	3261, 2742
6	1221, 1342, 2471	26	4551
7	2311	27	2124, 3191, 2212, 3721
8	2511	28	2532, 2282, 2392
9	3531, 1121, 3822, 1201	29	4251, 2423, 3412
10	4181	30	3321, 3222
11	3353	31	1531, 4231, 3622, 2132
12	2412, 1411, 1292, 1373	32	4371, 1412
13	2213	33	4221, 4242, 2281, 4131
14	1551, 4261, 2351	34	1343
15	1344, 1441	35	4421, 4162
16	1151, 4172, 2131	36	1721
17	4212	37	1301
18	2222, 2651, 2442	38	1341, 2161, 3193
19	3401, 3532, 4381	39	3231
20	1712	40	3371, 3421

Table 13. The optimization performance of the proposed algorithm for SET2 at a small scale.

Number	Init	Converge	Promote	Generation
1	173.9	60.2	65.4%	80
2	142.0	61.1	57.0%	148
3	139.9	65.9	52.8%	229
4	142.2	65.2	54.1%	219
5	141.2	65.2	53.8%	95
6	144.5	65.2	54.9%	141
7	148.6	72.0	51.6%	164
8	148.2	68.4	53.8%	143
9	141.0	65.3	53.7%	229
10	137.8	63.1	54.2%	134

Table 14. The optimization performance of the proposed algorithm for SET2 at a medium scale.

Number	Init	Converge	Promote	Generation
1	275.1	152.4	44.6%	164
2	277.5	154.9	44.2%	144
3	269.1	147.9	45.0%	175
4	280.1	156.8	44.0%	251
5	281.3	155.0	44.9%	196
6	290.3	151.3	47.9%	299
7	292.0	154.9	47.0%	286
8	282.2	152.5	45.9%	175
9	275.1	152.4	44.6%	164
10	277.5	154.9	44.2%	144

Table 15. The optimization performance of the proposed algorithm for SET2 at a large scale.

Number	Init	Converge	Promote	Generation
1	580.3	363.9	37.3%	273
2	558.2	341.6	38.8%	215
3	572.4	366.0	36.1%	191
4	566.9	357.1	37.0%	173
5	568.9	350.9	38.3%	178
6	569.0	364.5	37.7%	174
7	557.1	336.1	39.7%	273
8	566.4	360.0	36.4%	230
9	573.0	348.0	39.3%	204
10	561.8	344.5	38.7%	221

Table 16. The comparison between the proposed algorithm and CPLEX for SET2.

Scale	FWA	FWA Time	CPLEX	CPLEX Time	Error
small	65.16	1.86 s	60.18	600.7 s	8.19%
medium	153.3	3.41 s	143.76	2504.8 s	6.64%
large	353.26	10.37 s	NAN	NAN	NAN

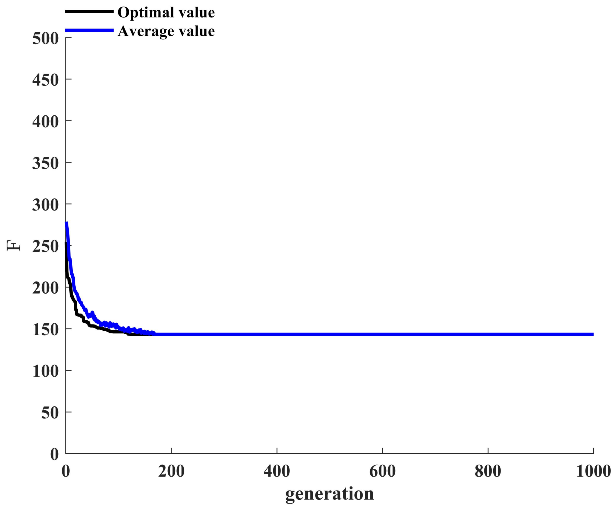

Figure 7. The object value of the proposed FWA algorithm for SET2.

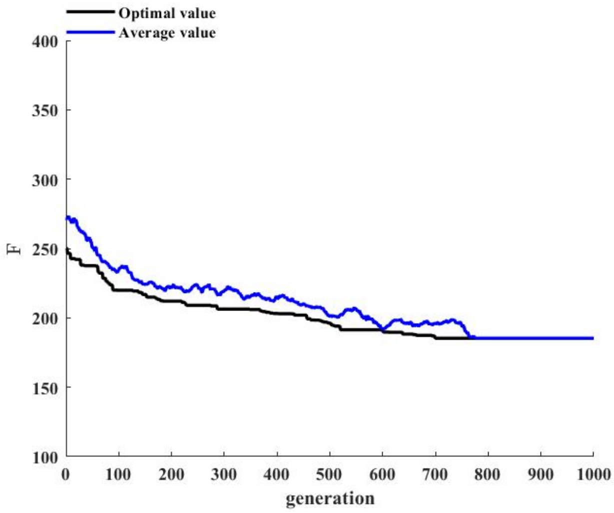

Figure 8. The object value of the proposed GA algorithm for SET2.

7. Conclusions

As the resources utilized within warehouses, such as space, equipment, and the workforce, are often limited, achieving warehousing sustainability within these constraints is crucial [48]. The objective of this study was to improve the warehousing efficiency and sustainability by establishing a storage location assignment optimization algorithm for non-traditional layout warehouses. The contributions are threefold. First, establishing a model for non-traditional layout warehouses can be challenging. This was addressed by establishing Flying-V layout and Fishbone layout models for non-traditional layout warehouses in detail. Second, to quantify warehousing sustainability, a practical multi-objective optimization model that considers the storage location assignment, order-picking efficiency, and shelf stability as optimizing objectives was proposed. Third, a storage location assignment optimization algorithm based on the FWA was proposed. The proposed algorithm leverages adaptive techniques in the explosion and selection stages, thereby improving the convergence rate and optimization performance. The results show that the proposed algorithm has a faster convergence rate and a better optimization performance. After optimization, there is greater potential for promoting the warehousing efficiency and increasing the warehouse sustainability.

This research presented a strategy to enable efficient and sustainable operations while cutting costs within warehouses. By employing a storage location assignment optimization algorithm for non-traditional layout warehouses, the limited space and workforce resources can handle more cargo. The proposed algorithm improves the shelf stability and reduces the travel distance of picking robots by lowering the center of gravity of cargo storage and optimizing cargo storage location assignment. All of this creates the potential to increase the warehousing sustainability.

The application of the proposed optimizing algorithm is not limited to the sustainable warehousing management mentioned in this paper. It can also be applied to the sustainable management of equipment resources, supermarket management, library management, and any type of management that uses a sustainable storage location assignment system. By utilizing these systems, organizations can achieve optimal storage location assignment with limited space and equipment resources. This results in increased efficiency and reduced resource waste.

The limitations of this paper are as follows: First, although the two typical non-traditional layout warehouses, Flying-V and Fishbone, were chosen as the research objectives, more non-traditional warehouse layout designs can be explored, such as leaf and butterfly and chevron ones. Second, the optimization of order-picking tracking and the integration of different algorithms can be considered. Lastly, the proposed algorithm has not been applied to practical warehouses. In the future, it will be applied to practical warehouses, and the corresponding experimental performance will be analyzed.

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

Funding: This work was supported by the Scientific Research Fund of the Zhejiang Provincial Education Department, China (No. Y201941984).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

FWA	Firework Algorithm
GA	Genetic Algorithm

References

1. Alnahhal, M.; Salah, B.; Ahmad, R. Increasing throughput in warehouses: The effect of storage reallocation and the location of input/output station. *Sustainability* **2022**, *14*, 4611. [CrossRef]
2. Gao, J.; Siddik, A.B.; Khawar Abbas, S.; Hamayun, M.; Masukujjaman, M.; Alam, S.S. Impact of E-commerce and digital marketing adoption on the financial and sustainability performance of MSMEs during the COVID-19 pandemic: An empirical study. *Sustainability* **2023**, *15*, 1594.
3. Li, Z.; Gao, G.; Xiao, X.; Zuo, H. Factors and formation path of cross-border E-commerce logistics mode selection. *Sustainability* **2023**, *15*, 3685.
4. Cai, J.; Li, X.; Liang, Y.; Ouyang, S. Collaborative optimization of storage location assignment and path planning in robotic mobile fulfillment systems. *Sustainability* **2021**, *13*, 5644.
5. Torabizadeh, M.; Yusof, N.M.; Ma'aram, A.; Shaharoun, A.M. Identifying sustainable warehouse management system indicators and proposing new weighting method. *J. Clean. Prod.* **2020**, *248*, 119190. [CrossRef]
6. Alnahhal, M.; Salah, B.; Ruzayqat, M. An efficient approach to investigate the tradeoff between double handling and needed capacity in automated distribution centers. *Sustainability* **2022**, *14*, 7678. [CrossRef]
7. Popović,V.; Kilibarda, M.; Andrejić, M.; Jereb, B.; Dragan, D. A new sustainable warehouse management approach for workforce and activities scheduling. *Sustainability* **2021**, *13*, 2021.

8. Bartholdi, J.J.; Hackman, S.T. *Warehouse & Distribution Science: Release 0.96*; The Supply Chain and Logistics Institute: 2016. Available online: https://www.warehouse-science.com/ (accessed on 13 March 2023).
9. Gu, J.; Goetschalckx, M.; McGinnis, L.F. Research on warehouse operation: A comprehensive review. *Eur. J. Oper. Res.* **2007**, *177*, 1–21.
10. Larco, J.A.; De Koster, R.; Roodbergen, K.J.; Dul, J. Managing warehouse efficiency and worker discomfort through enhanced storage assignment decisions. *Int. J. Prod. Res.* **2017**, *55*, 6407–6422. [CrossRef]
11. Roodbergen, K.J.; Vis, I.F.; Taylor, G.D., Jr. Simultaneous determination of warehouse layout and control policies. *Int. J. Prod. Res.* **2015**, *53*, 3306–3326.
12. Gue, K.R.; Meller, R.D. Aisle configurations for unit-load warehouses. *IIE Trans.* **2009**, *41*, 171–182. [CrossRef]
13. Bortolini, M.; Faccio, M.; Ferrari, E.; Gamberi, M.; Pilati, F. Design of diagonal cross-aisle warehouses with class-based storage assignment strategy. *Int. J. Adv. Manuf. Technol.* **2019**, *100*, 2521–2536.
14. Gue, K.R.; Ivanović, G.; Meller, R.D. A unit-load warehouse with multiple pickup and deposit points and non-traditional aisles. *Transp. Res. E-Log.* **2012**, *48*, 795–806.
15. Yang, X.; Liu, X.; Feng, L.; Zhang, J.; Qi, M. Non-Traditional layout design for robotic mobile fulfillment system with multiple workstations. *Algorithms* **2021**, *14*, 203. [CrossRef]
16. Zhou, L.; Liu, J.; Fan, X.; Zhu, D.; Wu, P.; Cao, N. Design of v-type warehouse layout and picking path model based on internet of things. *IEEE Access* **2019**, *7*, 58419–58428. [CrossRef]
17. Zhou, L.; Zhao, J.; Liu, H.; Wang, F.; Yang, J.; Wang, S. Stochastic models of routing strategies under the class-based storage policy in fishbone layout warehouses. *Sci. Rep.* **2022**, *12*, 12876.
18. Cardona, L.F.; Soto, D.F.; Rivera, L.; Martínez, H.J. Detailed design of fishbone warehouse layouts with vertical travel. *Int. J. Prod. Econ.* **2015**, *170*, 825–837. [CrossRef]
19. Bortolini, M.; Botti, L.; Cascini, A.; Gamberi, M.; Mora, C.; Pilati, F. Unit-load storage assignment strategy for warehouses in seismic areas. *Comput. Ind. Eng.* **2015**, *87*, 481–490.
20. Keung, K.L.; Lee, C.K.M.; Ji, P. Industrial internet of things-driven storage location assignment and order picking in a resource synchronization and sharing-based robotic mobile fulfillment system. *Adv. Eng. Inform.* **2022**, *52*, 101540.
21. Chiang, D.M.H.; Lin, C.P.; Chen, M.C. The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres. *Enterp. Inf. Syst.* **2011**, *5*, 219–234. [CrossRef]
22. Kim, J.; Méndez, F.; Jimenez, J. Storage location assignment heuristics based on slot selection and frequent itemset grouping for large distribution centers. *IEEE Access* **2020**, *8*, 189025–189035. [CrossRef]
23. Chiang, T.-A.; Che, Z.-H.; Hung, C.-W. A K-means clustering and the Prim's minimum spanning tree-based optimal picking-List consolidation and assignment methodology for achieving the sustainable warehouse operations. *Sustainability* **2023**, *15*, 3544. [CrossRef]
24. Burinskiene, A.; Lorenc, A.; Lerher, T. A simulation study for the sustainability and reduction of waste in warehouse logistics. *Int. J. Simul. Model.* **2018**, *17*, 485–497. [CrossRef]
25. Soheyl, K.; Sahman, K. Designing energy-efficient high-precision multi-pass turning processes via robust optimization and artificial intelligence. *J. Intell. Manuf.* **2021**, *32*, 1621–1647.
26. Soheyl, K.; Doulabi, H. A flexible robust model for blood supply chain network design problem. *Ann. Oper. Res.* **2022**. [CrossRef]
27. Soheyl, K.; Doulab, H. Designing emergency flood evacuation plans using robust optimization and artificial intelligence. *J. Comb. Optim.* **2021**, *41*, 640–677.
28. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [CrossRef]
29. Agarwal, R.; Agarwal, D.; Upadhyaya, S.; Ahmad, I. Optimization of a stochastic model having erratic server with immediate or delayed repair. *Ann. Oper. Res.* **2022**. [CrossRef]
30. Zhang, G.Q.; Xue, J.; Lai, K.K. A genetic algorithm based heuristic for adjacent paper-reel layout problem. *Int. J. Prod. Res.* **2000**, *38*, 3343–3356. [CrossRef]
31. Li, M.; Chen X.; Liu, C. Pareto and Niche Genetic Algorithm for storage location assignment optimization problem. In Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control, Dalian, China, 18–20 June 2008; pp. 465–4655.
32. Chen, L.; Langevin, A.; Riopel, D. A tabu search algorithm for the relocation problem in a warehousing system. *Int. J. Prod. Econ.* **2011**, *129*, 147–156. [CrossRef]
33. Zhang, R.Q.; Wang, M.; Pan, X. New model of the storage location assignment problem considering demand correlation pattern. *Comput. Ind. Eng.* **2019**, *129*, 210–219. [CrossRef]
34. Liu, J.; Zhang, Y.; Xiong, F.; Hu, Y. An approach to storage location assignment problem based on Flying-V layout. *Oper. Res. Manag. Sci.* **2019**, *28*, 27–33.
35. Hu, Y.; Liu, J.; Zhang, Y. Storage location assignment optimization in non-traditional warehouse base on AGA and MPSO. *High Technol. Lett.* **2018**, *28*, 980–990.
36. Soheyl, K.; Bahman, N. Multi-objective stochastic fractal search: A powerful algorithm for solving complex multi-objective optimization problems. *Methodol. Appl.* **2020**, *24*, 3037–3066.

37. Chou, Y.C.; Chen, Y.H.; Chen, H.M. Recency-based storage assignment and warehouse configuration for recurrent demands. *Comput. Ind. Eng.* **2012**, *62*, 880–889. [CrossRef]
38. Guerriero, F.; Musmanno, R.; Pisacane, O.; Rende, F. A mathematical model for the multi-level product allocation problem in a warehouse with compatibility constraints. *Appl. Math. Model.* **2013**, *37*, 4385–4389. [CrossRef]
39. Deb, K. Multi-objective optimization. In *Search Methodologies*; Springer: Boston, MA, USA, 2014; pp. 403–449.
40. Tan, Y.; Zhu, Y. Fireworks algorithm for optimization. In *Advances in Swarm Intelligence. ICSI 2010. Lecture Notes in Computer Science*; Tan, Y., Shi, Y., Tan, K.C., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6145.
41. He, W.; Mi, G.; Tan, Y. Parameter optimization of local-concentration model for spam detection by using fireworks algorithm. In *International Conference in Swarm Intelligence*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 439–450
42. Tao, Y.; Zhao, L. A novel system for WiFi radio map automatic adaptation and indoor positioning. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10683–10692. [CrossRef]
43. Qian, W.; Chai, J.; Xu, Z.; Zhang, Z. Differential evolution algorithm with multiple mutation strategies based on roulette wheel selection. *Appl. Intell.* **2018**, *48*, 3612–3629. [CrossRef]
44. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
45. Sun, N.; Lu, Y. A self-adaptive genetic algorithm with improved mutation mode based on measurement of population diversity. *Neural Comput. Appl.* **2019**, *31*, 1435–1443. [CrossRef]
46. Yan, C.; Li, M.; Liu, W.; Qi, M. Improved adaptive genetic algorithm for the vehicle Insurance Fraud Identification Model based on a BP Neural Network. *Theor. Comput. Sci.* **2020**, *817*, 12–23. [CrossRef]
47. Hamian, M.; Darvishan, A.; Hosseinzadeh, M.; Lariche, M.J.; Ghadimi, N.; Nouri, A. A framework to expedite joint energy-reserve payment cost minimization using a custom-designed method based on mixed integer genetic algorithm. *Eng. Appl. Artif. Intel.* **2018**, *72*, 203–212. [CrossRef]
48. Cantini, A.; De Carlo, F.; Tucci, M. Towards forklift safety in a warehouse: An approach based on the automatic analysis of resource flows. *Sustainability* **2020**, *12*, 8949. [CrossRef]