Integrating Optimized Fishbone Warehouse Layout, Storage Location Assignment and Picker Routing

Yanchun Wan, Yong Liu

Abstract—**New market environments require warehouses to shorten order response time and improve operational efficiency. In this study, we combine the problems of layout optimization, storage location assignment and picker routing to optimize order picking systems. First, we propose an optimal detailed three-dimensional (3D) design of a fishbone layout for a brownfield warehouse project and ensure the minimum average distance between all storage locations to the pick-up and deposit (P&D) point. Then, according to the turnover-based and correlation storage policies, we establish storage location assignment models and apply the traveling salesman problem (TSP) to plan picker routing. Third, we design a cooperative optimization algorithm (COA) to solve multiple order picking planning problems, which can improve the average optimal travel distance by 9% with good reliability. In addition, the proposed COA has good adaptability when solving practical issues. Finally, we find that the fishbone layout can shorten the picking distance by 10-15% without considering aisle congestion.**

Index Terms—**Fishbone layout, joint optimization, picker routing, storage location assignment.**

I. Introduction

CURRENTLY, the typical warehouse retains a traditional layout, in which aisles and shelves are parallel or vertical. However, the fishbone warehouse is also an important branch of warehouse research that scholars continue to evaluate. Gue and Meller proposed the fishbone layout with a single P&D point in 2009, which can be divided into four zones (zone 1, 2, 3 and 4 in Fig. 1) according to the location of the diagonal cross aisles, horizontal cross aisles and vertical cross aisles [1]. They also found that when a fishbone layout was used in a single-instruction unit loading warehouse, the expected travel distance was reduced by 20% compared with the traditional layout.

After the fishbone layout was proposed, scholars continued to study it. Research in this field was focused on the following three key areas:

(1) Verification of fishbone layout improving warehouse operation efficiency under different conditions.

(2) Improvement of fishbone layout.

(3) Industrial application of the fishbone layout.

Manuscript received February 25, 2022; revised August 14, 2022. This work was supported by "Guangzhou Philosophy and Social Science Planning 2021 Annual Project (No. 2021GZGJ42)", "2021 Student Research Plan of South China University of Technology (No. X202010561378)", "The College Students' Innovative Entrepreneurial Training Plan Program (No.202110561010)", "2022 Key Projects of Teaching Research and Teaching Reform of South China University of Technology (No.x2ds-C9223068)", "2022 Guangdong Postgraduate Education Innovation Plan Project (No.2022ANLK006)".

Yanchun Wan is an associate professor of the Department of Electronic Commerce, South China University of Technology, Guangzhou 510006, China (e-mail: tcycwan@scut.edu.cn).

Yong Liu is an undergraduate of Department of Electronic Commerce, South China University of Technology, Guangzhou 510006, China (Corresponding author, e-mail: 15874987732@163.com).

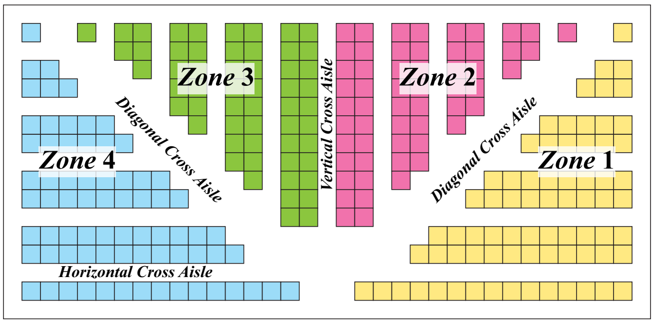

Fig. 1. Definitions of zones of a fishbone layout

Although the research focuses are different, the core purpose of most research is to improve warehouse operation efficiency, specifically the minimization of order response time, picking travel distance, picking time or other indicators. The picking process takes up more than half of the total operation time in a manual warehouse. Picking efficiency is affected by warehouse layout, storage location assignment and order batching, but there is still a lack of systematic research on layout optimization, storage location assignment and picker routing of fishbone layout warehouses.

In this paper, we present a combinatorial optimization model that jointly optimizes fishbone layouts, storage location assignment, and picker routes. We also design an optimization algorithm to solve the models, including a detailed design algorithm for a fishbone layout (DDA_FL), storage location assignment algorithm and picker routing algorithm. We can input the warehouse size to obtain an optimal design of fishbone layouts using DDA_FL. We compare the results of particle swarm optimization (PSO) and gravitational search algorithm (GSA) in solving storage location assignment and picker routing problems. Then, we also design a combinatorial optimization algorithm (COA) to solve the multiobjective optimization model proposed in this research.

The remainder of this paper is organized as follows: Section II reviews the literature on the research progress of fishbone layout, warehouse storage problems and multi-problem combinations in order picking systems. Section III shows the combinatorial optimization model for the fishbone layout, storage location assignment and picker routing. Section IV describes the detailed design of the COA. Section V analyses the advantages of the fishbone layout compared with traditional layouts and algorithm performance through a numerical example. Conclusions and future research are then proposed in section VI.

II. Literature review

After the fishbone layout was first proposed, many scholars studied whether it could improve warehouse operation efficiency with different storage and operation policies. Compared with traditional layouts, Pohl et al. found that the fishbone layout can reduce 10-15% of the expected travel distance under unit load warehouses with dual command operations [2]. Pohl et al. also found that the fishbone layout has similar improvements in warehouses with random or turnover-based storage policies [3]. Çelk et al. also found that improvement is affected by the uneven degree of demand and the order size. They also indicated that the fishbone layout might increase the expected travel distance by approximately 30% as the order size increases [4].

Scholars have proposed improvements to the fishbone layout to improve efficiency, including the warehouse size, the slope of the diagonal cross aisle, and aisle design. Gue and Meller described the design principle of the fishbone layout, in which the warehouse's width (i.e., where the P&D point is located) is twice its length, and the diagonal cross aisles end at the corner of the warehouse [1]. Pohl et al. also identified these design principles in a warehouse with random and turnover-based storage policies [2], [3]. Cardona et al. divided fishbone warehouses into greenfield warehouse projects and brownfield warehouse projects [5]. The greenfield warehouse project is built according to preferred specifications, and the brownfield warehouse project has already been built, but its best layout has to be determined. Cardona et al. also built a nonlinear optimization model that aimed to find the optimal slope of the diagonal cross aisle to minimize costs. They found these design principles highly effective in the greenfield warehouse project and proposed a common method to calculate the best slope of the brownfield warehouse project. Their experimental results showed that the operational costs would not increase markedly, even if the tilt angle deviated from the optimal value [5].

Due to the complex combination of aisles and shelves, picker routing in the fishbone warehouse layout is a research focus. Çelk et al. designed an exact algorithm for the picker routing problem of fishbone warehouse layout [4]. Kumar et al. found that mobile robots have more feasible paths in the fishbone layout warehouse than the traditional layouts, which means that the fishbone layout can reduce collisions in an order picking system [6]. To reduce the total travel distance, Öztürkoğlu et al. proposed three improved fishbone layout schemes (Chevron, Leaf and Butterfly) for a fishbone warehouse [7].

To promote the application of the fishbone layout in the industry, Cardona et al. proposed a three-dimensional detailed design method of the fishbone layout based on the warehouse cost in different regions [8]. However, this method is only applicable to greenfield warehouse projects. Because there are many brownfield warehouse projects, we provide another detailed layout as a supplement, which can generate an optimal design of fishbone layouts in given warehouse sizes.

The primary tasks of storage location assignment are determining the inventory strategy and storage location of items according to the warehouse characteristics. Van Gils et al. proposed that the decision of the order picking system can be divided into strategic, tactical, and operational decisions, and the decisions of storage location assignment influenced tactical decisions, such as batch decision, classification and picker routing [9]. Researchers must use appropriate principles and methods to locate items appropriately to improve picking efficiency when assigning storage locations. Turnover-based policy means that items with high turnover rates are allocated closer to the P&D point, which can improve the operational efficiency of warehouses [10], [11].

In recent years, with the development of e-commerce, the responsiveness of order picking has been increasing. The correlation storage policy has an important influence on the picking operation by shortening the picking travel distance by storing items with high correlation nearby [12]. The focus of the correlation storage policy is on how to measure the correlation values between items. There are two mainstream methods to determine the correlation: (1) measuring the correlation between items by the item occurrence in the same order [13]–[15]; and (2) using a data mining algorithm to analyze the demand association rules between items [16]–[19]. Therefore, we present a model for the storage location assignment according to the correlation storage policy and turnover-based policy.

The new market environment requires the warehouse to shorten order response time systematically. Van Gils et al. found that the order picking operation can be effectively managed by combining multiple order picking plan problems [9]. According to the current research results, combinations with better performances include storage location assignment and order batching [20]–[22], storage location assignment and picker routing [23]–[25], and order batching and picker routing [26]–[28]. In this study, we combine the problems of layout optimization, storage location assignment and picker routing to optimize the order picking systems. We design a novel COA with the elite retention strategy (Deb et al. [29]), and then the results of sequential and combinatorial optimization are compared.

III. Model development

This section describes three models that can be used to solve the problems of optimal design of fishbone layouts, storage location assignment and picker routing. We assume a brownfield project of a warehouse that operates under the following conditions:

(1) Only one order is picked in each picking operation; thus, order batching problems are not considered in this study.

(2) The storage location represents where items can be stored.

(3) One-to-one correspondence between storage location and items; thus, the same items can only be stored in one storage location, and only one type of item can be stored in one storage location.

(4) All storage locations have the same sizes, and all aisles have the same width a_p.

(5) Order pickers can pass through the aisles in both directions, and aisle congestion is not considered.

(6) Vertical movements are not considered as they are out of scope in this study.

The nomenclature used in this research is shown in Table I.

A. Design of the fishbone layouts

We develop a nonlinear model to obtain the optimal detailed 3D design of the fishbone layouts. This section

TABLE I
PARAMETERS AND VARIABLES

Symbol	Parameter description
Indices and sets:	
$s \in S = \{1, 2, \cdots, N\}$	Storage location index
$p \in P = \{1, 2, \cdots, M\}$	Items index
$l \in L = \{1, 2, \cdots, L\}$	Order index
$O_l, l \in L$	Item set for order l
Parameters:	
N	Total number of storage locations in a fishbone layout
C	Required storage capacity
$Dout_s$	Distance from picking point s to P&D point, calculation method is shown in Appendix A
$D_{s_1 s_2}, s_1 \in S, s_2 \in S$	Distance from picking point s_1 to s_2, calculation method is shown in Appendix A
$CorL_{p_1 p_2}, p_1 \in P, p_2 \in P$	Correlation of Item p_1 and p_2, when $CorL_{p_1 p_2} = 1$, there is a high level of correlation value between Item p_1 and Item p_2, otherwise $CorL_{p_1 p_2} = 0$
$Q_p, p \in P$	Turnover rate of Item p
s_0	P&D point
Decision variables:	
m	Slope of diagonal cross aisle
$X_{ps}, p \in P, s \in S$	When$X_{ps} = 1$, Item p is stored in storage location s, otherwise $X_{ps} = 0$
$Y_{l,p_1 p_2}, l \in L, p_1, p_2 \in \{s_0, O_l\}$	When $Y_{l,p_1 p_2} = 1$, picker picks Item p_1 at first and then picks Item p_2, otherwise $Y_{l,p_1 p_2} = 0$

will also introduce the definition, calculation and output of parameters.

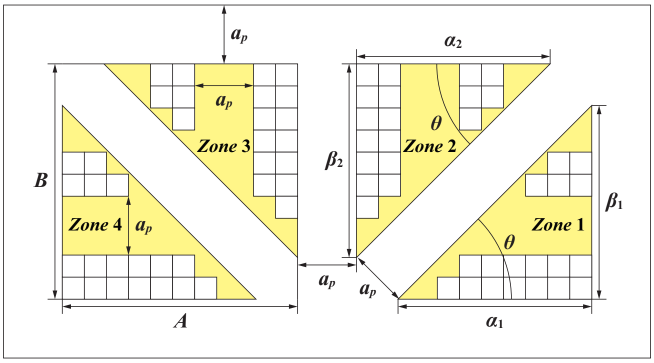

Fig. 2. Standard geometry of fishbone layouts

1) Definition of parameters: Cardona et al. [8] have defined a standard geometry for the fishbone layout, and we use their definition in this study. Parameters of the fishbone layout are shown in Fig. 2, including the sizes of the warehouse A, B, the width of all aisles a_p, the angle of diagonal cross aisles θ, the sizes of the storage locations (length, width and height: e_l, e_w, e_h), the number of shelf layers τ and the sizes of each zone $\alpha \to (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, $\beta \to (\beta_1, \beta_2, \beta_3, \beta_4)$.

2) Calculation of parameters: Determining the right angle for the diagonal cross aisles is an important issue in completing the design of a fishbone layout. Fig. 3 shows that different angles have different fishbone layouts, which can influence operation efficiency. According to the relationship between θ and $(\alpha_1, \beta_1, \alpha_2, \beta_2)$, we calculate α_1, β_1 under the condition of $m \le B/\alpha_1$ and $m > B/\alpha_1$, and α_2, β_2 under the condition of $m \le \beta_2/A$ and $m > \beta_2/A$. Methods for calculating these parameters of the model are described as follows.

Step1: Considering that each area must contain a certain number of storage locations, we calculate the slope of the diagonal crossing channel and define its domain as:

$$m = tan(\theta), m \in \left\{ m \left| \left(A - \frac{a_p}{2m}\sqrt{1+m^2} - e_l \right) m \ge e_w \right. \wedge \left(B - \frac{a_p}{2}\sqrt{1+m^2} - e_l \right) \frac{1}{m} \ge e_w \right\} \quad (1)$$

Step2: The sizes of each zone,$\alpha \to (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, $\beta \to (\beta_1, \beta_2, \beta_3, \beta_4)$are as follows:

$$\begin{cases} \alpha_1 = A - \frac{a_p}{2m}\sqrt{1+m^2}, & \beta_1 = \begin{cases} m\alpha_1, m \le \frac{B}{\alpha_1} \\ B, \frac{B}{\alpha_1} < m \end{cases} \\ \alpha_2 = \begin{cases} A, m \le \frac{\beta_2}{A} \\ \frac{\beta_2}{m}, \frac{\beta_2}{A} < m \end{cases}, & \beta_2 = B - \frac{a_p}{2}\sqrt{1+m^2} \\ \alpha_3 = \alpha_2, & \beta_3 = \beta_2 \\ \alpha_4 = \alpha_1, & \beta_4 = \beta_1 \end{cases} \quad (2)$$

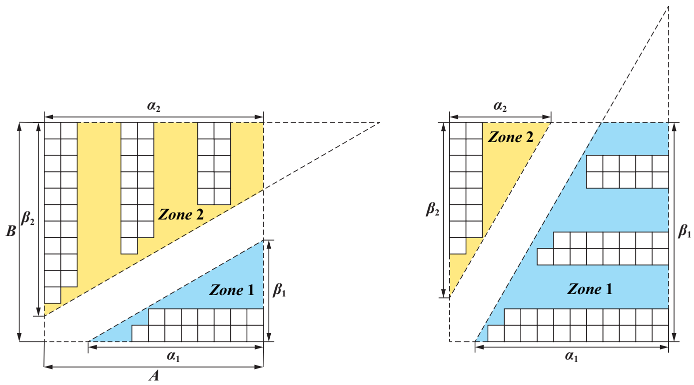

Fig. 3. Different angles of diagonal cross aisles for the fishbone layouts

Step3: The number of lines (columns) in each zone ($i =$

$1, 2, 3, 4)$, n_i is as follows:

$$n_i = \begin{cases} \max\limits_{q}(qe_w + \frac{q-1}{2}a_p \leq \beta_i), i=1 \vee i=4, \text{in zone 1 or 4} \\ \max\limits_{q}(qe_w + \frac{q-1}{2}a_p \leq \alpha_i), i=2 \vee i=3, \text{in zone 2 or 3} \end{cases} \tag{3}$$

Step4: The number of storage locations per line $j = 1, 2, \cdots, n_i$ in each zone, ω_{ij} is as follows:

$$\omega_{ij} = \begin{cases} \omega_{1j}, i = 1 \vee i = 4, \text{in zone 1 or 4} \\ \omega_{2j}, i = 2 \vee i = 3, \text{in zone 2 or 3} \end{cases} \tag{4}$$

$$\begin{cases} \omega_{1j} = \left\lfloor \frac{1}{me_l}\left[\alpha_1 m - je_w - \frac{a_p}{4}(2j-3-(-1)^j)\right]\right\rfloor \\ \omega_{2j} = \left\lfloor \frac{m}{e_l}\left[\frac{\beta_2}{m} - je_w - \frac{a_p}{4}(2j-3-(-1)^j)\right]\right\rfloor \end{cases} \tag{5}$$

Step5: Storage locations $O_{ijkt}(x_{ijkt}, y_{ijkt}, z_{ijkt})$ and picking points $Op_{ijkt}(xp_{ijkt}, yp_{ijkt}, zp_{ijkt})$ are shown in Fig. 4, and $i \in \{1, 2, 3, 4\}$, $0 \leq j \leq n_i$, $0 \leq k \leq \omega_{ij}$, $0 \leq t \leq \tau$, $j, k, t \in N^*$,

$$\begin{aligned} x_{1jkt} &= -x_{4jkt}, & y_{1jkt} &= y_{4jkt}, \\ xp_{1jkt} &= -xp_{4jkt}, & yp_{1jkt} &= yp_{4jkt}, \\ x_{2jkt} &= -x_{3jkt}, & y_{2jkt} &= y_{3jkt}, \\ xp_{2jkt} &= -xp_{3jkt}, & yp_{2jkt} &= yp_{3jkt}, \end{aligned} \tag{6}$$

$$\begin{cases} x_{1jkt} = A - (k - \frac{1}{2})e_l + \frac{a_p}{2}, xp_{1jkt} = x_{1jkt} \\ y_{1jkt} = (j - \frac{1}{2})e_w + [2j - 1 - (-1)^j]\frac{a_p}{4} \\ yp_{1jkt} = y_{1jkt} + (-1)^j(\frac{e_w}{2} + \frac{a_p}{4}) \\ x_{2jkt} = (j - \frac{1}{2})e_w + [2j - 1 - (-1)^j]\frac{a_p}{4} \\ xp_{2jkt} = x_{2jkt} + (-1)^j(\frac{e_w}{2} + \frac{a_p}{4}) \\ y_{2jkt} = B - (k - \frac{1}{2})e_l + \frac{a_p}{2}, yp_{2jkt} = y_{2jkt} \\ z_{ijkt} = e_0 + e_h(t-1), zp_{ijkt} = 0 \end{cases} \tag{7}$$

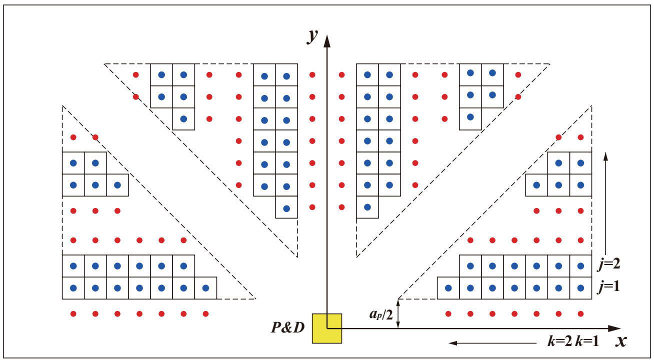

Fig. 4. Coordinate system of fishbone layouts

3) Model building: Based on the definition and parameters, we establish a layout optimization model that minimizes the average distance between all storage locations and P&D points:

- **Objective function:**

$$\min f_1 = \frac{1}{N}\sum_{s \in S} Dout_s \tag{8}$$

- **s.t.:**

$$N - C \geq 0 \tag{9}$$

$$N = \tau \sum_{i=1}^{4}\sum_{j=1}^{n_i} \omega_{ij} \tag{10}$$

The objective function (8) minimizes the average distance from storage locations to the P&D point. Constraint (9) ensures that the total number of storage locations in the fishbone warehouse exceeds the desired storage capacity. Constraint (10) calculates the fishbone warehouse's total number of storage locations.

B. Optimization model for storage location assignment

Based on the turnover-based policy and correlation storage policy, we formulate the following mathematical model for storage location assignment:

- **Objective function:**

$$\min f_2 = \beta_1 \sum_{p \in P}\sum_{s \in S} X_{ps}D_pQ_p + \beta_2 y_1 - \beta_3 y_2,$$
$$\beta_1 + \beta_2 + \beta_3 = 1 \tag{11}$$

- **s.t.:**

$$CorL_{p_1p_2}X_{p_1s_1}X_{p_2s_2}D_{s_1s_2} \leq y_1$$
$$\forall p_1, p_2 \in P, \forall s_1, s_2 \in S \tag{12}$$

$$(1 - CorL_{p_1p_2})\frac{2X_{p_1s_1}X_{p_2s_2}D_{s_1s_2}}{Q_{p_1} + Q_{p_2}} \geq y_2$$
$$\forall p_1, p_2 \in P, \forall s_1, s_2 \in S \tag{13}$$

$$\sum_{s \in S} X_{ps} = 1, \forall p \in P \tag{14}$$

$$\sum_{p \in P} X_{ps} \leq 1, \forall s \in S \tag{15}$$

$$X_{ps} \in \{0, 1\}, \forall p \in P, \forall s \in S \tag{16}$$

The objective function (11) consists of three parts: the first part ensures that items with a higher turnover rate are stored in the storage locations closer to the P&D point according to turnover-based policy. The second part makes items with higher correlation values store closer according to the correlation storage policy. To reduce aisle congestion, the third part is to make items with low correlations but high turnover rates far away from each other. Constraint (12) calculates the farthest distance between any two items whose level of correlation is 1. Constraint (13) makes items with zero correlations and high turnover rates to be far away to reduce aisle congestion. Constraint (14) ensures that only one item can be stored in one storage location. Constraint (15) ensures that a storage location stores only one item at most. Constraint (16) defines the domains of the decision variables.

C. Optimization model for picker routing

In this section, we formulate a mathematical model for the picker routing problem based on the traveling salesman problem (TSP):

- **Objective function:**

$$\min f_3 = \sum_{l \in L} \sum_{p_1 \in \{s_0, O_l\}} \sum_{p_2 \in \{s_0, O_l\}} Y_{l,p_1p_2} X_{p_1s_1} X_{p_2s_2} D_{s_1s_2} \quad (17)$$

- **s.t.:**

$$\sum_{p_1 \in \{s_0, O_l\}, p_1 \neq p_2} Y_{l,p_1p_2} = 1, \forall l \in L, p_2 \in \{s_0, O_l\} \quad (18)$$

$$\sum_{p_2 \in \{s_0, O_l\}, p_1 \neq p_2} Y_{l,p_1p_2} = 1, \forall l \in L, p_1 \in \{s_0, O_l\} \quad (19)$$

$$u_{p_1} - u_{p_2} + |O_l| \, Y_{l,p_1p_2} \leq |O_l| - 1 \quad \forall l \in L, p_1 \neq p_2, p_1, p_2 \in O_l \quad (20)$$

$$Y_{l,p_1p_2} \in \{0, 1\}, \forall l \in L, p_1, p_2 \in \{s_0, O_l\} \quad (21)$$

The objective function (17) minimizes the expected travel distance to finish order picking. Constraints (18) and (19) ensure that the order pickers pick every item only once in an assignment. Constraint (20) eliminates subloops in a route by adopting methods proposed by Miller et al. and the u_p are arbitrary real numbers [30]–[33]. Constraint (21) defines the domains of the decision variables.

IV. Heuristic algorithm design

To solve the three optimization models presented in Section III, three algorithms have been designed under the framework of gravitational search algorithm (GSA), including a detailed design algorithm of fishbone layout, storage location assignment algorithm and picker routing algorithm, which can operate independently. Then, we combine the three algorithms to obtain a COA using the cooperative optimization idea and the elite retention strategy.

According to Newton's law of motion and gravity, GSA is a heuristic search algorithm designed by Rashedi et al. [34]. GSA constructs an independent system that obeys Newton's law of motion and gravity. Every agent has its mass in the system, and they can attract each other through attraction. Therefore, all the agents travel near the heaviest agent, which uses the exploration and optimization portions of the algorithm during the iterative process. Khan et al. [35] indicate that GSA is adept at optimal global search, but its search speed decreases in the later stage of iteration. GSA can also solve a wide range of problems with a fixed small population, which can markedly reduce computational complexity [34]. This section introduces the proposed algorithms, including three that can run separately and a cooperative optimization algorithm.

A. Detailed design algorithm of fishbone layouts

First, we design Algorithm 1 to solve the optimization model proposed in Section III-A, which generates an optimal design of the fishbone layout. The input parameters are presented in Section III-A, and the output parameters include the number of lines per zone n_i, the number of storage locations per row (column) per zone ω_{ij}, the coordinates of all storage locations O_{ijkt} and picking points Op_{ijkt}, the distance matrices between picking points D and the distance matrices from the picking point to the P&D point $Dout$.

Algorithm 1 Detailed design of the fishbone layout

```
Input: θ, A, B, a_p, e_l, e_w, e_h, e_0, τ
Output: m, α, β, n_i, ω_ij, [O_ijkt, Op_ijkt], [D_out, D]
procedure DETAILED_DESIGN(θ, A, B, a_p, e_l, e_w, e_h, e_0, τ)
    m ← CalculateAngle(A, B, a_p, e_l, e_w, θ)                          ▷ Eq. (1)
    (α, β) ← CalculateSize(A, B, m, a_p)                               ▷ Eq. (2)
    n_i ← CalculateNi(α, β, a_p, e_w)                                  ▷ Eq. (3)
    ω_ij ← CalculateLocations(m, A, B, α, β, a_p, n_i, e_w, e_l)       ▷ Eq. (4)(5)
    [O_ijkt(x_ijkt, y_ijkt, z_ijkt), Op_ijkt(xp_ijkt, yp_ijkt, zp_ijkt)]
        ← CalculateCoordinate(n_i, ω_ij, A, B, a_p, e_l, e_w, e_h, e_0, τ)   ▷ Eq. (6)(7)
    [D_out, D] ← CalculateDistance(n_i, ω_ij, A, B, α, β, a_p, e_l, e_w, e_h, e_0, τ, O_ijkt, Op_ijkt)
                                                                       ▷ Appendix A
end procedure
```

Second, the optimization model presented in Section III-A is nonlinear, and the decision variable is a continuous random variable. We propose a GSA to solve the detailed design of the fishbone layout (DDA_FL_GSA). All agents have position and velocity parameters in DDA_FL_GSA. The custom part of the proposed algorithm focuses on the initialization and updating of parameters.

The initialization procedure uses a random strategy to generate agents from the solution domain, but agents created in this way may not meet the constraint of Eq. (9). Similarly, the problem also exists when agents are updated. We design Algorithm 2 to check agents that do not meet the constraint and replace them to solve the problem. For further information on GSA, refer to Rashedi et al. [34].

Algorithm 2 Check procedure

```
Input: P_k                                  ▷ P_k : position and parameters
Output: P_k
function CHECK(P_k)
    for k ← 1, K do                         ▷ Iterate over all the agents
        S ← Detailed_Design(P_k, A, B, a_p, e_l, e_w, e_h, τ)
                                            ▷ Calculate S
        while S < C do
            P_k = P_k + σ                   ▷ Replacement
            S ← Detailed_Design(P_k, A, B, a_p, e_l, e_w, e_h, τ)
        end while
    end for
return P_k
end function
```

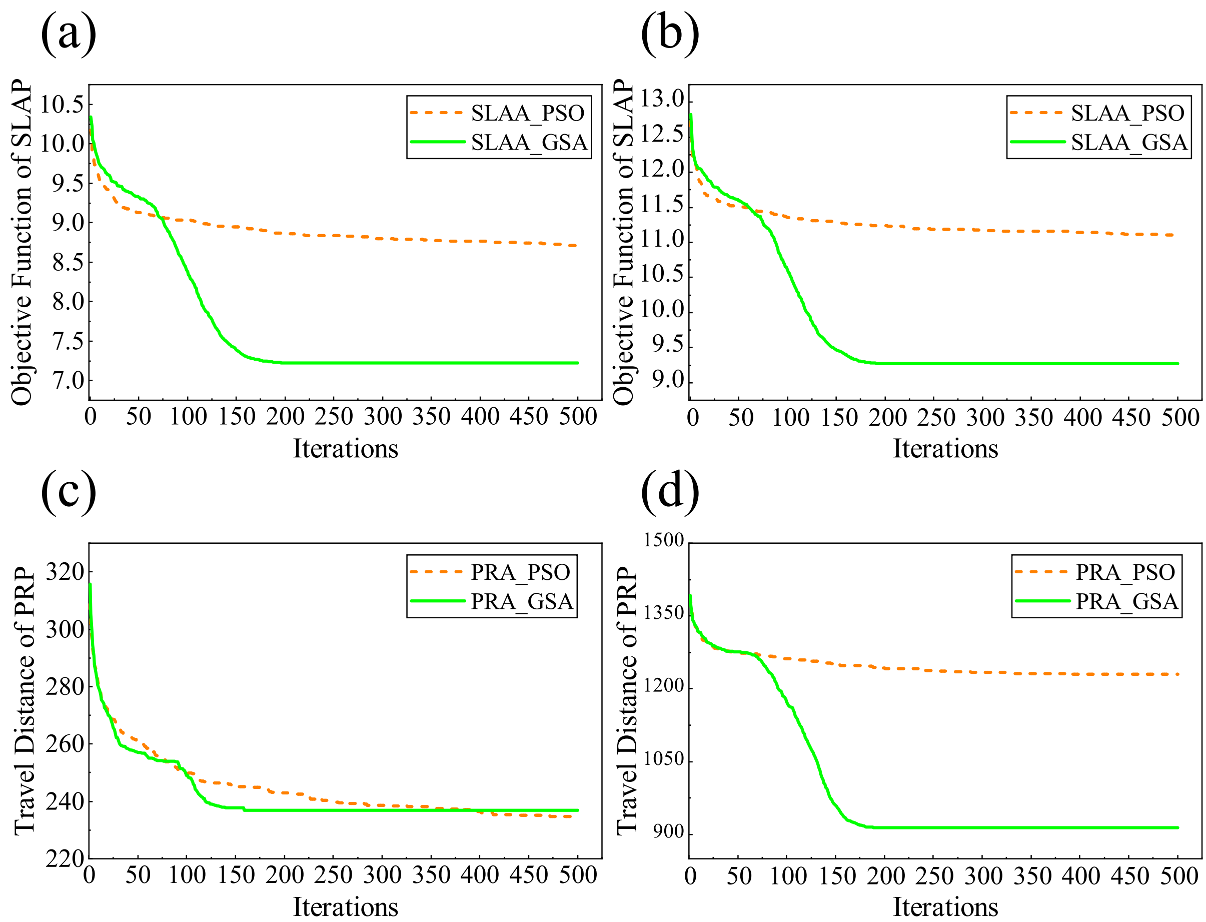

Fig. 5. Evolutionary process of the best solution. (a) Scenario 1 with 106 items in the warehouse. (b) Scenario 2 with 156 items in the warehouse. (c) Scenario 3 with 19 items in one order. (d) Scenario 4 with 73 items in one order

B. Storage location assignment and picker routing algorithms

The storage location assignment problem (SLAP) and picker routing problems (PRP) are NP-hard due to changes caused by item quantity and warehouse characteristics [36]. Therefore, we first use the PSO and GSA [35], [37] to solve the SLAP and PRP. The experimental results are shown in Fig. 5. For each instance in Fig. 5 (a, b), the objective function of SLAP computed by GSA is superior to that based on PSO at the same stage, and there is a large gap between the objective function computed by GSA and PSO. In addition, the GSA converges to the best solution within a few iterations. The experimental results in Fig. 5 (c, d) show that GSA has many advantages in finding better solutions and speeding up convergence compared with PSO as the scale of the problem grows. Therefore, we design the storage location assignment algorithm (SLAA_GSA) and picker routing algorithm (PRA_GSA) in the framework of GSA, and the flowcharts of SLAA_GSA and PRA_GSA are shown in Fig. 6.

The process of SLAA_GSA is explained as follows: we set the parameters, input a design of the fishbone layout, and initialize population position and speed parameters first. Decoding is completed using the roulette method, and the fitness values of the agents are calculated according to Eq. (11). SLAA_GSA can search for the best global and local agents based on the fitness values in the population and update the parameters and population according to GSA. The optimal scheme for storage location assignment is output when the termination criterion is met. SLAA_GSA and PRA_GSA are similar. Some of the differences between SLAA_GSA and PRA_GSA are as follows:

(1) In PRA_GSA, we must input the optimal scheme for storage location assignment generated by SLAA_GSA.

(2) We use the roulette method to obtain a feasible scheme for picker routing.

(3) The fitness values of the agents are calculated according to Eq. (17).

(4) The output of PRA_GSA is the optimal scheme for picker routing.

C. Cooperative Optimization Algorithm

In this section, we present a novel COA based on the combination of three algorithms (DDA_FL_GSA, SLAA_GSA and PRA_GSA). We found that the solution time of PRA_GSA increases markedly with increasing order size. Therefore, if the above three algorithms are combined sequentially or side by side, the computational complexity will exponentially increase exponentially during evolution. We screen out the best global solution during evolution using the elite retention strategy to reduce computational complexity. After obtaining the optimal design of the fishbone layout and the best local scheme for storage location assignment, we calculate the expected travel distance of the local optimal

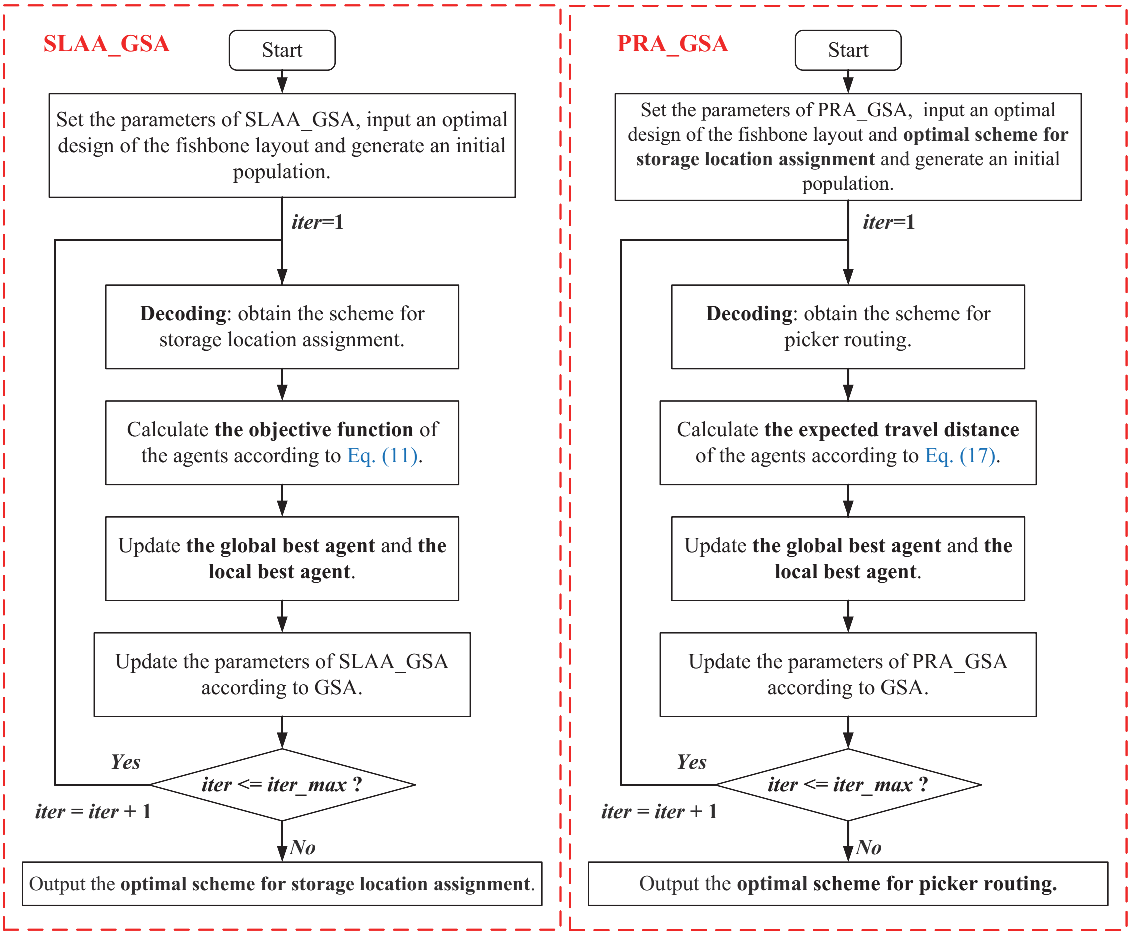

Fig. 6. Flowcharts of SLAA_GSA and PRA_GSA

scheme for picker routing as the local optimal expected travel distance by COA (Fig. 7). The proposed elite retention strategy retains the scheme for storage location assignment with the minimum expected travel distance. Then, the scheme for storage location assignment retained is inserted into an agent in the initial population of SLAA_GSA by a random strategy as a parameter. The process is repeated until termination criteria are met.

V. Numerical Example and Discussion

In this study, the numerical examples focus on a retail warehouse. The numerical example includes five datasets ((40-106), (60-126), (80-132), (100-146) and (120-156)) that include the number of orders and items in the warehouse. The analysis of the optimal slope of diagonal cross aisles for the fishbone layout is presented in Section V-A. We set the width (i.e., where the P&D point is located) of the warehouse to twice its length and the diagonal cross aisles end at the top corner of the warehouse. We performed these experiments on a computer with an Intel(R) Core (TM) I5-8300H CPU @2.30.

A. Analysis of the optimal slope in the fishbone layout

The most important characteristic of the fishbone layout is that it has two diagonal cross aisles compared with traditional layouts; thus, determining the optimal slope is key to achieving an optimal fishbone layout design. We can divide the layout design into two types, greenfield warehouse projects and brownfield warehouse projects, which are introduced in Section II. In brownfield warehouse projects, we use discrete representations to build a model that can find the optimal slope to supplement the literature [3], [5], [8].

As shown in Table II, we simulated 30 warehouses with different storage capacities to verify the reliability of the proposed method, and these warehouses met the condition of $A = B$.

We also executed GA, GSA and PSO to find the optimal value of m, and the results are shown in Fig. 8 (a, b). The optimal values of m are approximately 1 in most warehouses obtained by the three algorithms. The deviation is less than 0.041375 in more than 90% of cases and primarily appears in small scenarios. In Fig. 8 (c, d), the deviation between the optimal value of m obtained by the three algorithms and $m = 1$ on the average distance f_1 is less than 0.454%. These results indicate that the model proposed in this study is easier to solve and has good stability.

In addition to the warehouse with $A = B$, there are different types of warehouses with different B/A. We simulated 200 warehouses ($0 < B/A < 2$) with different storage capacities (in each B/A, there are ten warehouses with storage capacity from 1000 to 40000) and then compared the result with the optimal slope by Cardona et al. [5] in Fig. 9 (a). The gap between the optimal slope by DDA_FL_GSA that has a marginal advantage in solving the optimal slope, and the method of Cardona et al. [5] is small when B/A approaches 1, but the gap becomes larger as the B/A increases or decreases. The reason for this difference is due to the difference in spatial representation, considered factors and model solving method. Cardona et al. [5] established an optimization model with continuous spatial expression, in which the influence of aisles was ignored. However, we developed an optimization model with discrete spatial expression while considering the effects of the aisles. Simulation results show some differences in the optimal slope even in the same B/A with different warehouse sizes. As shown in Fig. 9 (b), the optimal slope solved by the proposed method can shorten the average distance between all storage locations and the P&D point. These results show that it is not necessary to calculate the optimal slope for different B/A. The proposed optimal model can be used to make specific decisions regarding the fishbone warehouse layout.

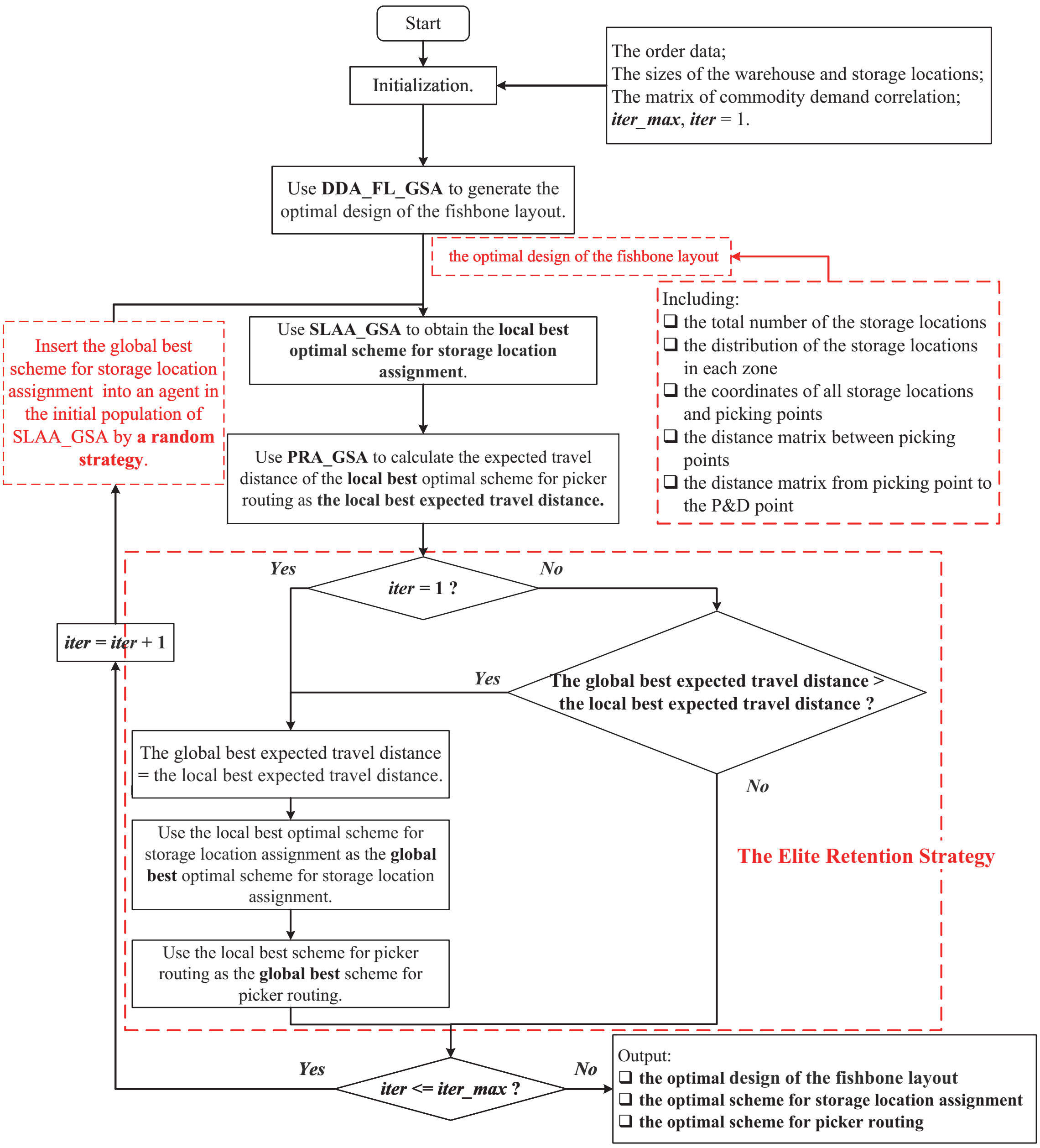

Fig. 7. Flowchart of the cooperative optimization algorithm

TABLE II
STORAGE CAPACITY OF 30 WAREHOUSES

	1	2	3	4	5	6	7	8	9	10
Small scenarios	1620	2088	2604	3168	3816	4500	5232	6048	6900	7824
Medium scenarios	8796	9828	10920	12060	13260	14520	15852	17208	18648	20148
Large scenarios	21672	23280	24948	26664	28428	30276	32160	34092	36108	38160

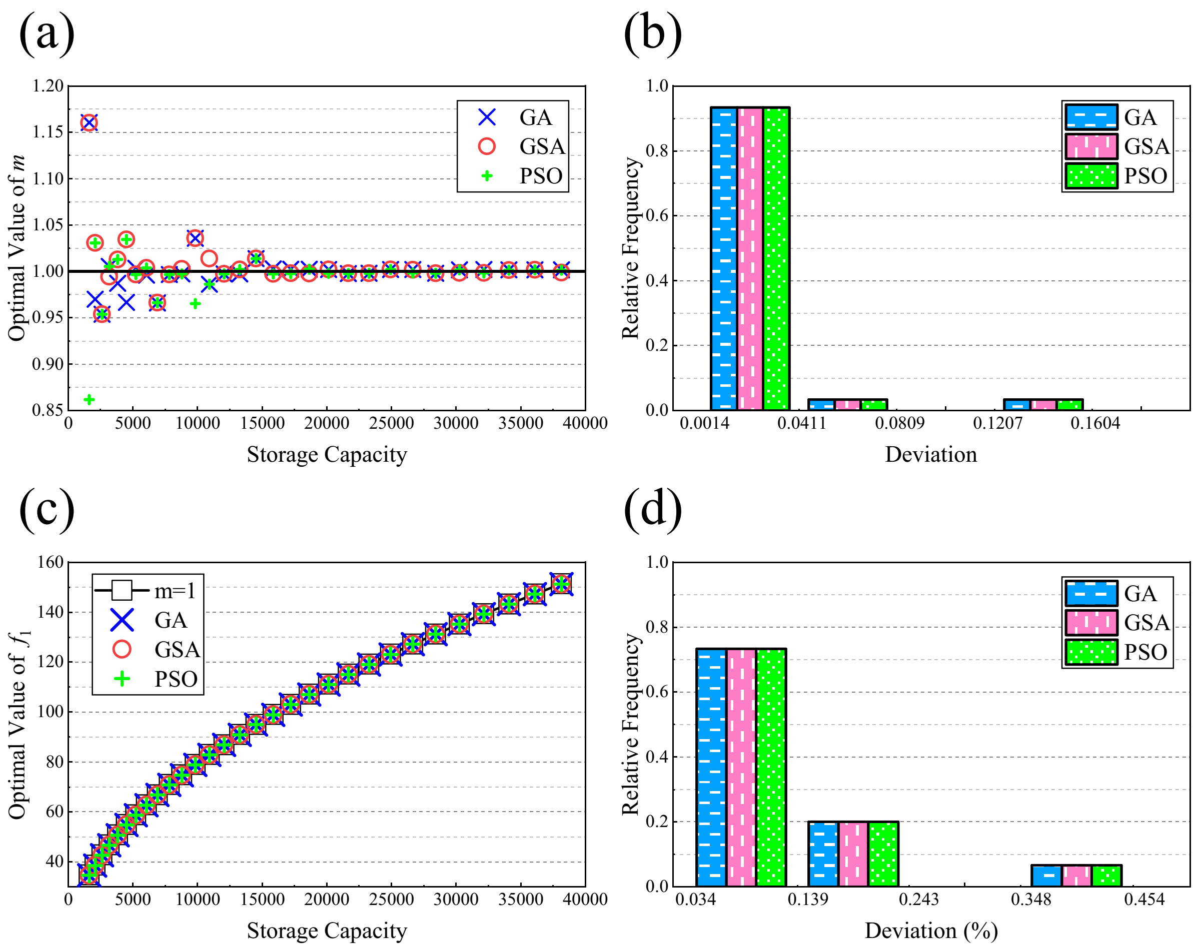

Fig. 8. Reliability verification. (a) When $A = B$, the optimal value of m. (b) Deviation between the optimal solutions obtained by the three algorithms and $m = 1$. (c) Results of f_1 under different optimal solutions. (d) Deviation of f_1 between the optimal solutions obtained by the three algorithms and $m = 1$

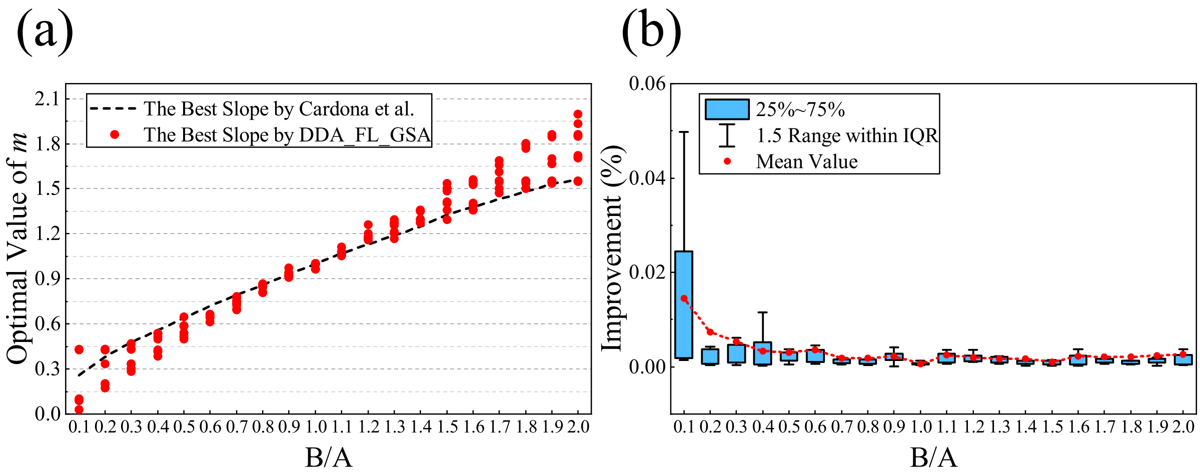

Fig. 9. Comparison of the optimal slopes and the average distance f_1 between all storage locations and the P&D point

B. Algorithm performance

This section analyzes the performance of the COA from two perspectives. First, we analyze the advantages of COA by observing the evolutionary process. Then, we compare the COA with the traditional algorithm.

We compare the evolution process of the optimal objective function value calculated by COA_GSA and COA_PSO (Fig. 10). COA_GSA and COA_PSO are combinatorial optimization algorithms based on the GSA and PSO frameworks, respectively. As shown in Fig. 10 (a), for each numerical example, COA_GSA has better global search capabilities than COA_PSO, particularly in the optimal objective function value of the storage location assignment. The fluctuation of the optimal objective function of COA_GSA means it has a wider space to search for a better solution than the current optimal solution. In addition, COA_GSA converges to the

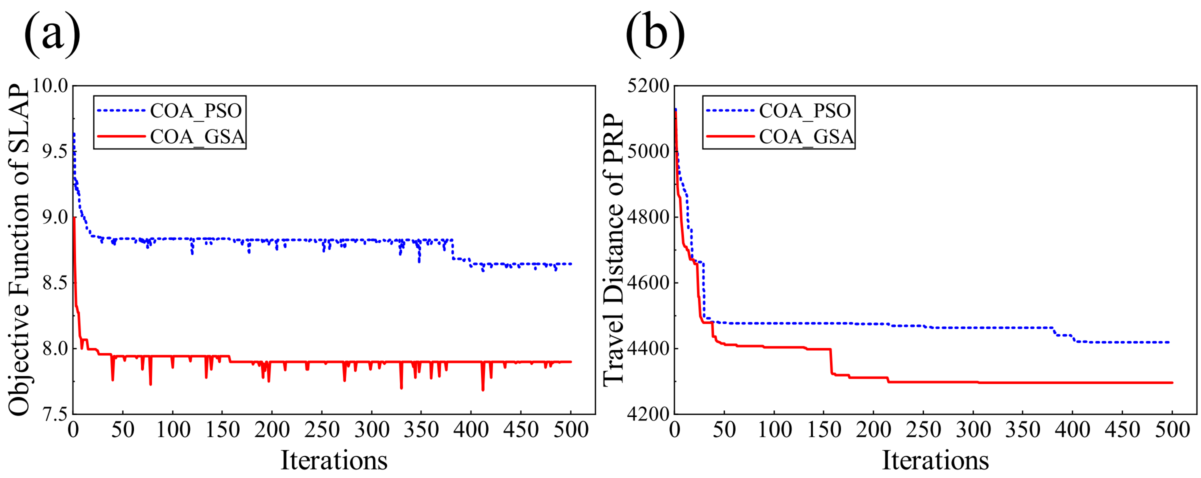

Fig. 10. Evolutionary process of the algorithms. (a) Contemporary optimal value of the objective function f_2 of storage allocation. (b) Evolutionary process of picking the optimal value of the travel distance f_3

optimal solution within a smaller number of iterations than COA_PSO. In total, the the experimental results show that COA_GSA has advantages in finding better solutions and speeding up convergence.

We also compared the results of COA_GSA with the sequential optimization algorithm (SOA_GSA), whose flowchart is shown in Fig. 11. Table III and Table IV report comparisons of the optimal objective function value of Eq. (11) and the optimal expected travel distance calculated by Eq. (17). Better results are marked in bold. $PE(\%)$ is calculated by Eq. (22) [38], which is used to evaluate the robustness of COA_GSA and SOA_GSA. $impro(\%)$ is used to evaluate the performance of COA_GSA and SOA_GSA, which is calculated by Eq. (23). From the result of the objective function value of the storage location assignment, COA_GSA has a better optimization value than SOA_GSA. The average optimal value is improved by 5-10% (Table III). In addition, COA_GSA achieves better robustness, which can ensure the stability of the solution. In the travel distance of picking, COA_GSA also has a stronger optimization ability, and the average optimal value is decreased by more than 9% compared with SOA_GSA (Table IV). The robustness of COA_GSA is weakened in picker routing problems because COA_GSA must explore the combinatorial solution of two problems. In total, COA_PSO achieves global optimization and robust capability improvement at the expense of the robust performance of picking route planning problems, particularly in large-scale cases, which indicates that the proposed method has practical significance in further improving warehouse operation efficiency. The calculation method of some relevant indicators is as follows:

$$PE(\%) = \frac{average - best}{best} \times 100\% \tag{22}$$

$$\begin{aligned} best\_impro(\%) &= \frac{SOA\_best - COA\_best}{SOA\_best} \times 100\% \\ average\_impro(\%) &= \frac{SOA\_average - COA\_average}{SOA\_average} \\ &\times 100\% \end{aligned} \tag{23}$$

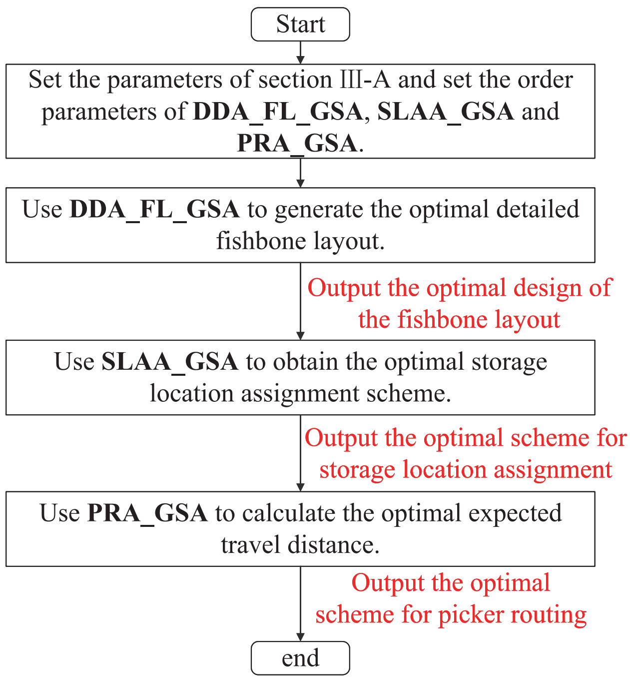

Fig. 11. Flowchart of the sequential optimization algorithm

C. Comparisons between fishbone layout and traditional layout

This section discusses the differences between the traditional layout (Fig. 12) and the fishbone layout. We simulated two layouts in the same experimental environment.

We divided the objective function of Eq. (11) into Part 1, Part 2 and Part 3, then summed them by setting $\beta_1 = 0.6$, $\beta_2 = 0.2$, $\beta_3 = 0.2$, which are shown in Table V. Part 1 ensures that items with higher turnover rates are assigned closer to the P&D point. The fishbone layout performed markedly better than the traditional layout in Part 1, which indicates that we can use the fishbone layout to reduce the operating distance markedly in unit load warehouses with single command operations. Part 2 makes items with higher correlations be stored closer according to the correlation storage policy. From the results of Table V, the fishbone

TABLE III
COMPARISON OF THE OBJECTIVE FUNCTION f_2 OF STORAGE LOCATION ASSIGNMENT BETWEEN SOA_GSA AND COA_GSA

	SOA_GSA			COA_GSA				
	best	average	$PE(\%)$	best	$impro(\%)$	average	$impro(\%)$	$PE(\%)$
40-106	7.50	8.52	13.54%	7.44	**0.76%**	7.72	**9.32%**	**3.75%**
60-126	7.74	8.56	10.56%	7.39	**4.54%**	7.78	**9.04%**	**5.34%**
80-132	8.37	9.18	9.66%	8.02	**4.24%**	8.70	**5.24%**	**8.52%**
100-146	9.18	10.00	8.93%	8.82	**3.94%**	9.07	**9.26%**	**2.90%**
120-156	9.01	10.12	12.39%	9.02	-0.19%	9.16	**9.49%**	**1.52%**

TABLE IV
COMPARISONS OF THE TRAVEL DISTANCE f_3 OF ROUTING PLANNING BETWEEN SOA_GSA AND COA_GSA

	SOA_GSA			COA_GSA				
	best	average	$PE(\%)$	best	$impro(\%)$	average	$impro(\%)$	$PE(\%)$
40-106	2958.85	2997.99	1.32%	2715.88	**8.21%**	2820.00	**5.94%**	3.83%
60-126	5021.37	5085.64	1.28%	4126.81	**17.82%**	4474.17	**12.02%**	8.42%
80-132	6651.15	6687.40	0.55%	5809.49	**12.65%**	6498.41	**2.83%**	11.86%
100-146	8962.20	9022.25	0.67%	7793.25	**13.04%**	8078.08	**10.46%**	3.65%
120-156	11632.32	11739.84	0.92%	9649.07	**17.05%**	9980.01	**14.99%**	3.43%

TABLE V
SIMULATION RESULTS OF FISHBONE LAYOUT AND TRADITIONAL LAYOUTS ($\beta_1 = 0.6$, $\beta_2 = 0.2$, $\beta_3 = 0.2$)

	Traditional layout					Fishbone layout					
	f_2 of storage location assignment				Travel distance	f_2 of storage location assignment				Travel distance	
	Part 1	Part 2	Part 3	Sum		Part 1	Part 2	Part 3	Sum		$impro(\%)$
40-106	17.90	31.85	29.07	11.29	3313.10	5.91	35.07	14.19	7.72	2820.00	**14.88%**
60-126	17.99	33.10	31.31	11.15	5284.90	6.14	35.05	14.55	7.78	4474.17	**15.34%**
80-132	17.56	32.85	29.80	11.15	7697.25	6.77	37.48	14.28	8.70	6498.41	**15.57%**
100-146	19.54	35.65	31.57	12.55	9591.40	7.11	38.92	14.89	9.07	8078.08	**15.78%**
120-156	18.82	35.75	31.66	12.11	11355.05	7.08	39.49	14.91	9.16	9980.01	**12.11%**

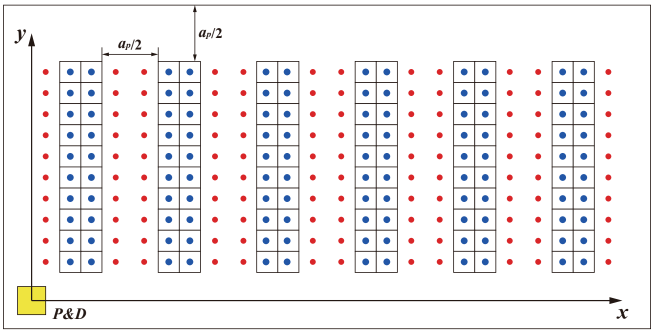

Fig. 12. Traditional layout

layout exhibits no advantage in Part 2. Part 3 is calculated by Eq. (13), and the original intention of setting this part is to reduce aisle congestion. However, results show that the fishbone layout is a risk factor for increasing aisle congestion compared with the traditional layout. If we don't consider aisle congestion, the average pickers' travel distance of fishbone layout is reduced by approximately 10-15% compared to the traditional layout.

To analyze the difference between the fishbone layout and the traditional layout of the three parts of the objective function f_2 in more detail, we simulated different weight combinations (β_1, β_2 and β_3) in Table VI. The results are shown in Table VII and Fig. 13. We can find that under different weight coefficient combinations, the comparison results of function f_2 in the two different layouts agree with those of ($\beta_1 = 0.6$, $\beta_2 = 0.2$, $\beta_3 = 0.2$).

TABLE VI
DIFFERENT WEIGHT COMBINATIONS

No.	β_1	β_2	β_3	No.	β_1	β_2	β_3
1	0.00	0.00	1.00	9	0.25	0.75	0.00
2	0.00	0.25	0.75	10	0.50	0.00	0.50
3	0.00	0.50	0.50	11	0.50	0.25	0.25
4	0.00	0.75	0.25	12	0.50	0.50	0.00
5	0.00	1.00	0.00	13	0.75	0.00	0.25
6	0.25	0.00	0.75	14	0.75	0.25	0.00
7	0.25	0.25	0.50	15	1.00	0.00	0.00
8	0.25	0.50	0.25	-	-	-	-

VI. CONCLUSIONS AND FUTURE RESEARCH

Scholars have conducted many theoretical and practical studies to improve warehouse layout schemes, operation efficiency, and industrial applications. In this study, we investigate the optimization of fishbone layout warehouses.

First, based on the comprehensive optimization of picking systems, this study extends the optimization problem of fishbone-layout warehouses to the integration optimization of layout, storage location assignment and picker routing.

TABLE VII
COMPARISON OF OBJECTIVE FUNCTION f_2 BETWEEN THE TRADITIONAL LAYOUT AND THE FISHBONE LAYOUT IN DATASET, 120-156

				Fishbone layout				Traditional layout			
No.	β_1	β_2	β_3	Sum	Part1	Part2	Part3	sum	Part1	Part2	Part3
1	0.00	0.00	1.00	-16.32	**15.81**	44.33	16.32	**-32.60**	26.56	**39.90**	**32.60**
2	0.00	0.25	0.75	-2.40	**15.72**	38.64	16.09	**-15.21**	26.36	**35.70**	**32.18**
3	0.00	0.50	0.50	10.85	**15.22**	37.67	15.97	**1.48**	26.49	**34.70**	**31.74**
4	0.00	0.75	0.25	23.91	**15.25**	37.05	15.51	**17.53**	26.07	**33.92**	**31.62**
5	0.00	1.00	0.00	36.15	**15.07**	36.15	14.02	**32.40**	26.39	**32.40**	**28.09**
6	0.10	0.10	0.80	-7.68	**12.32**	40.01	16.14	**-19.69**	24.45	**37.45**	**32.35**
7	0.10	0.25	0.65	0.47	**13.28**	38.10	15.98	**-9.46**	24.92	**35.57**	**32.06**
8	0.10	0.40	0.50	8.50	**13.13**	37.79	15.86	**0.46**	25.41	**34.53**	**31.78**
9	0.10	0.55	0.35	16.01	**12.68**	36.88	15.82	**10.31**	25.29	**34.27**	**31.61**
10	0.10	0.70	0.20	23.74	**13.13**	36.50	15.62	**19.59**	23.97	**33.55**	**31.44**
11	0.10	0.85	0.05	31.41	**13.73**	36.22	15.00	**28.55**	24.02	**32.58**	**30.96**
12	0.25	0.10	0.65	-3.74	**10.29**	40.69	15.97	**-11.46**	22.60	**38.33**	**32.21**
13	0.25	0.25	0.50	4.47	**11.09**	38.67	15.93	**-1.19**	23.40	**35.73**	**31.94**
14	0.25	0.40	0.35	12.25	**11.22**	37.41	15.76	**8.55**	23.51	**34.45**	**31.73**
15	0.25	0.55	0.20	20.10	**11.21**	37.08	15.48	**18.22**	22.42	**34.35**	**31.40**
16	0.25	0.70	0.05	27.69	**10.76**	36.75	14.57	**26.51**	21.74	**32.28**	**30.48**
17	0.40	0.10	0.50	-0.14	**9.40**	40.36	15.87	**-3.40**	21.80	**38.55**	**31.94**
18	0.40	0.25	0.35	8.00	**9.44**	38.94	15.76	**6.73**	21.62	**36.80**	**31.77**
19	0.40	0.40	0.20	**15.90**	**9.42**	37.92	15.20	16.08	21.31	**34.62**	**31.44**
20	0.40	0.55	0.05	**24.06**	**10.03**	37.81	14.89	25.30	20.74	**33.63**	**29.93**
21	0.55	0.10	0.35	**3.49**	**8.81**	41.03	15.60	4.13	20.72	**38.42**	**31.72**
22	0.55	0.25	0.20	**11.49**	**8.60**	39.32	15.33	14.04	20.15	**36.90**	**31.32**
23	0.55	0.40	0.05	**19.40**	**8.84**	38.10	14.07	23.14	20.03	**34.00**	**29.44**
24	0.70	0.10	0.20	**6.65**	**8.03**	40.73	15.24	11.31	19.61	**38.42**	**31.27**
25	0.70	0.25	0.05	**14.98**	**8.20**	39.78	14.04	20.93	19.17	**35.92**	**29.47**
26	0.85	0.10	0.05	**9.89**	**7.65**	40.71	13.61	18.31	18.81	**38.17**	**29.89**

We present a detailed 3D design optimization model of a fishbone layout for brownfield warehouse projects. This model ensures the minimum average distance between the P&D point and all storage locations and deepens the study of fishbone 3D layout optimization based on the work of Cardona et al [5], [8]. In constructing the storage location assignment model, in addition to the correlation storage policy and turnover-based storage policy, we present a storage policy in which low correlation and high turnover items are assigned farther away to reduce aisle congestion. We construct a three-objective optimization model in which three storage location assignment policies and TSP are used.

To solve a combination of multiple problems in order picking systems, we present three algorithms (DDA_FL_GSA, SLAA_GSA and PRA_GSA) that can run separately to solve three problems. We verify the reliability of DDA_FL_GSA through a simulation experiment and find that it has good adaptability and can be applied in practice. When designing SLAA and PRA, we compared the calculation performance of PSO and GSA, and the results show that GSA has more advantages in finding better solutions and speeding up convergence compared with PSO as the scale of the problem grows. Then, we design SLAA_GSA and PRA_GSA in the framework of GSA. Finally, we present COA_GSA based on DDA_FL_GSA, SLAA_GSA and PRA_GSA. The COA_GSA has the dual merits of finding better solutions and speeding up convergence. In addition, the results of this study show that COA_GSA can improve the average optimal travel distance by 9% more than SOA_GSA, but its robustness is partially weakened.

We simulate a fishbone and traditional layout in the same experimental environment. The primary advantage of the fishbone layout is a reduction in the travel distance in picker-to-order warehouses. Experimental results show that the pickers' travel distance of the fishbone-layout warehouse is reduced by 10-15% if aisle congestion is not considered. In addition, we find that the fishbone layout has the following characteristics:

(1) The fishbone layout can reduce the travel distance markedly in unit load warehouses with single command operations compared with the traditional layout.

(2) Numerical examples show that the fishbone layout has a higher risk of aisle congestion than the traditional layout.

(3) Optimal results show that the fishbone layout warehouse has marked advantages when using a turnover-based policy, but the traditional layout warehouse has advantages when using a correlation storage policy.

For future research, aisle congestion with fishbone layouts should be investigated because the risk of aisle congestion with a fishbone layout may increase, as found in this study. Concurrently, fishbone layouts are more complex than traditional layouts with regard to shelf placement, aisle location, and other aspects; thus, fishbone layout warehouse operators need more training to gain proficient operational skills. To put forward more comprehensive suggestions on the fishbone layouts, scholars can add comparing cost, human resources and other indicators with the traditional layouts. In addition,

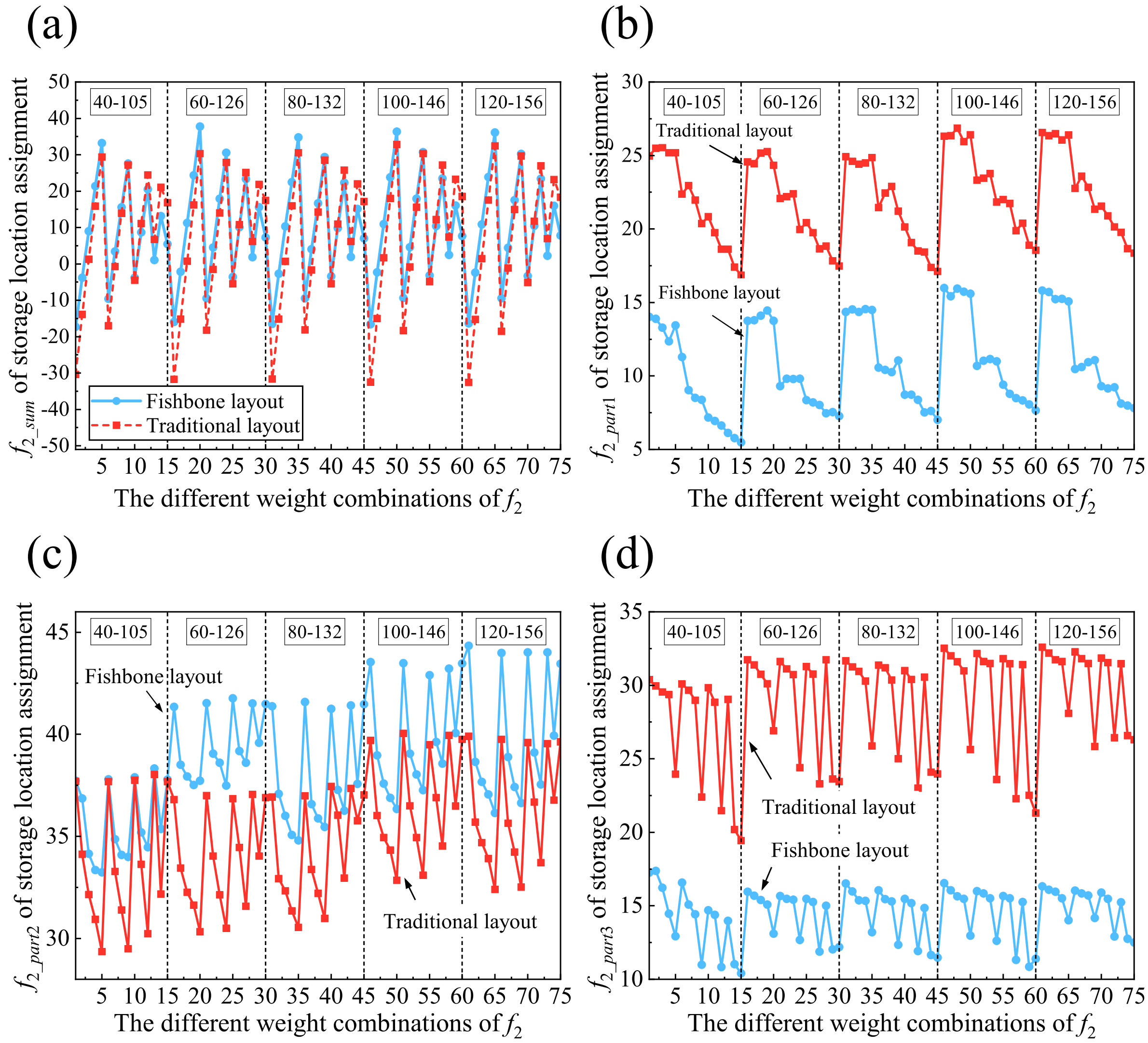

Fig. 13. Comparison of objective function f_2 between the traditional layout and the fishbone layout

order batching and time window constraints are important research fields with warehouses that require a quick response, such as E-Commerce warehouses. Therefore, introducing them into a fishbone layout warehouse will yield meaningful research results.

Appendix A
Distance Matrix

The optimization models in Section III require the parameters of the distance matrix D between picking points and the distance matrix $Dout$, which is the distance from the picking points to the P&D point. We introduce how to calculate these distance matrices in this section. Cardona et al. [5], [8] proposed a method to calculate the distance matrix from the picking points to the P&D point, and we used the method in the proposed optimization models. This study focuses on how to calculate the distance matrix between picking points [2]. The fishbone layout has four zones that are symmetric; thus, we divide it into two circumstances and define it as follows:

$$D(i_1j_1k_1t_1, i_2j_2k_2t_2) = \begin{cases} D1(i_1j_1k_1t_1, i_2j_2k_2t_2), i_1 = i_2 \\ D2(i_1j_1k_1t_1, i_2j_2k_2t_2), i_1 \neq i_2 \end{cases} \tag{A.1}$$

Case 1: $Op_{i_1j_1k_1t_1}$ and $Op_{i_2j_2k_2t_2}$ are in the same zone.

The relevant parameters are shown in Fig. A1. $D1$ is defined as follows:

$$D1(i_1j_1k_1t_1, i_2j_2k_2t_2) = \begin{cases} D_1(i_1j_1k_1t_1, i_2j_2k_2t_2), i_1 = i_2 \in \{1,4\} \\ D_2(i_1j_1k_1t_1, i_2j_2k_2t_2), i_1 = i_2 \in \{2,3\} \end{cases} \tag{A.2}$$

$$D_1(i_1j_1k_1t_1, i_2j_2k_2t_2) = \begin{cases} L_{1-a}, j_1 = j_2 \\ L_{1-b}, |j_1 - j_2| = 1 \wedge (-1)^{\min(j_1,j_2)} = 1 \\ L_{1-c}, \text{other conditions} \end{cases} \tag{A.3}$$

with:

$$L_{1-a} = |xp_{i_1j_1k_1t_1} - xp_{i_2j_2k_2t_2}| \tag{A.4}$$

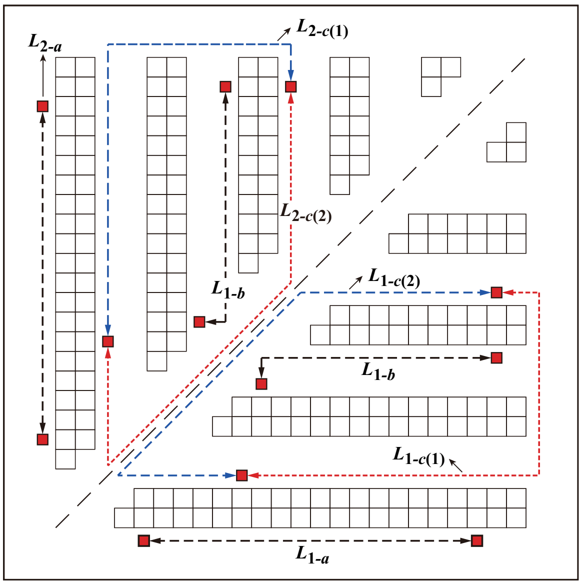

Fig. A1. Path planning of case 1

$$L_{1-b} = |yp_{i_1j_1k_1t_1} - yp_{i_2j_2k_2t_2}| + |xp_{i_1j_1k_1t_1} - xp_{i_2j_2k_2t_2}| \tag{A.5}$$

$$L_{1-c} = \min(L_{1-c(1)}, L_{1-c(2)}) \tag{A.6}$$

$$\begin{cases} L_{1-c(1)} = \left(\dfrac{a_p}{2} + A - |xp_{i_1j_1k_1t_1}|\right) \\ \quad + \dfrac{a_p}{2} + |yp_{i_1j_1k_1t_1} - yp_{i_2j_2k_2t_2}| \\ \quad + \left(\dfrac{a_p}{2} + A - |xp_{i_2j_2k_2t_2}|\right) \\ L_{1-c(2)} = \left(Am + \dfrac{a_p}{2} - yp_{i_1j_1k_1t_1}\right)\dfrac{1}{m} \\ \quad - \left(A + \dfrac{a_p}{2} - |xp_{i_1j_1k_1t_1}|\right) \\ \quad + \left(Am + \dfrac{a_p}{2} - yp_{i_2j_2k_2t_2}\right)\dfrac{1}{m} \\ \quad - \left(A + \dfrac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right) \\ \quad + |yp_{i_1j_1k_1t_1} - yp_{i_2j_2k_2t_2}|\dfrac{\sqrt{m^2+1}}{m} \end{cases} \tag{A.7}$$

and:

$$D_2(i_1j_1k_1t_1, i_2j_2k_2t_2) = \begin{cases} L_{2-a}, j_1 = j_2 \\ L_{2-b}, |j_1 - j_2| = 1 \\ \qquad \wedge(-1)^{\min(j_1,j_2)} = 1 \\ \min(L_{2-c(1)}, L_{2-c(2)}), \\ \qquad |j_1 - j_2| \neq 1 \\ \qquad \vee(-1)^{\min(j_1,j_2)} \neq 1 \end{cases} \tag{A.8}$$

with:

$$L_{2-a} = |yp_{i_1j_1k_1t_1} - yp_{i_2j_2k_2t_2}| \tag{A.9}$$

$$L_{2-b} = |yp_{i_1j_1k_1t_1} - yp_{i_2j_2k_2t_2}| + |xp_{i_1j_1k_1t_1} - xp_{i_2j_2k_2t_2}| \tag{A.10}$$

$$\begin{cases} L_{2-c(1)} = \left(B + \dfrac{a_p}{2} - yp_{i_1j_1k_1t_1}\right) \\ \quad + \dfrac{a_p}{2} + |xp_{i_1j_1k_1t_1} - xp_{i_2j_2k_2t_2}| \\ \quad + \left(\dfrac{a_p}{2} + B - yp_{i_2j_2k_2t_2}\right) \\ L_{2-c(2)} = \left(\dfrac{B}{m} + \dfrac{a_p}{2} - |xp_{i_1j_1k_1t_1}|\right)m \\ \quad - \left(B + \dfrac{a_p}{2} - yp_{i_1j_1k_1t_1}\right) \\ \quad + \left(\dfrac{B}{m} + \dfrac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right)m \\ \quad - \left(B + \dfrac{a_p}{2} - yp_{i_2j_2k_2t_2}\right) \\ \quad + |xp_{i_1j_1k_1t_1} - xp_{i_2j_2k_2t_2}|\sqrt{m^2+1} \end{cases} \tag{A.11}$$

Case 2: $Op_{i_1j_1k_1t_1}$ and $Op_{i_2j_2k_2t_2}$ are not in the same zone.

$D2$ is defined as follows:

$$D2(i_1j_1k_1t_1, i_2j_2k_2t_2) = \begin{cases} D_3(i_1j_1k_1t_1, i_2j_2k_2t_2), (i_1, i_2) \in \{(1,2)(4,3)\} \\ D_4(i_1j_1k_1t_1, i_2j_2k_2t_2), (i_1, i_2) \in \{(1,3)(4,2)\} \\ D_5(i_1j_1k_1t_1, i_2j_2k_2t_2), (i_1, i_2) \in \{(1,4),(4,1)\} \\ D_6(i_1j_1k_1t_1, i_2j_2k_2t_2), (i_1, i_2) \in \{(2,3),(3,2)\} \end{cases} \tag{A.12}$$

Situation one: One picking point is in zone 1, and the other is in zone 2; or one is in zone 4, and the other is in zone3.

The relevant parameters are shown in Fig. A2. We assume that $Op_{i_1j_1k_1t_1}$ is in zone 1 and $Op_{i_2j_2k_2t_2}$ is in zone 2, then:

$$D_3(i_1j_1k_1t_1, i_2j_2k_2t_2) = \begin{cases} \min(L_{1-2(1)-a}, L_{1-2(1)-b(1)}, L_{1-2(1)-b(1)}, L_{1-2(1)-c}), \\ \quad B + \dfrac{a_p}{2} - \left(\dfrac{B}{m} + \dfrac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right)m < yp_{i_1j_1k_1t_1} \\ L_{1-2(2)}, \\ \quad B + \dfrac{a_p}{2} - \left(\dfrac{B}{m} + \dfrac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right)m = yp_{i_1j_1k_1t_1} \\ \min(L_{1-2(3)-a}, L_{1-2(3)-b(1)}, L_{1-2(3)-b(1)}, L_{1-2(3)-c}), \\ \quad B + \dfrac{a_p}{2} - \left(\dfrac{B}{m} + \dfrac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right)m > yp_{i_1j_1k_1t_1} \end{cases} \tag{A.13}$$

with:

$$L_{1-2(1)-a} = \left(Am + \frac{a_p}{2} - yp_{i_1j_1k_1t_1}\right)\frac{1}{m} - \left(A + \frac{a_p}{2} - |xp_{i_1j_1k_1t_1}|\right) + \frac{\sqrt{1+m^2}}{m} \left|yp_{i_1j_1k_1t_1} - \left[B + \frac{a_p}{2} - \left(\frac{B}{m} + \frac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right)m\right]\right| + \left[\left(\frac{B}{m} + \frac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right)m - \left(B + \frac{a_p}{2} - yp_{i_2j_2k_2t_2}\right)\right] \tag{A.14}$$

The calculation methods of $L_{1-2(2)}$ and $L_{1-2(3)-a}$ are

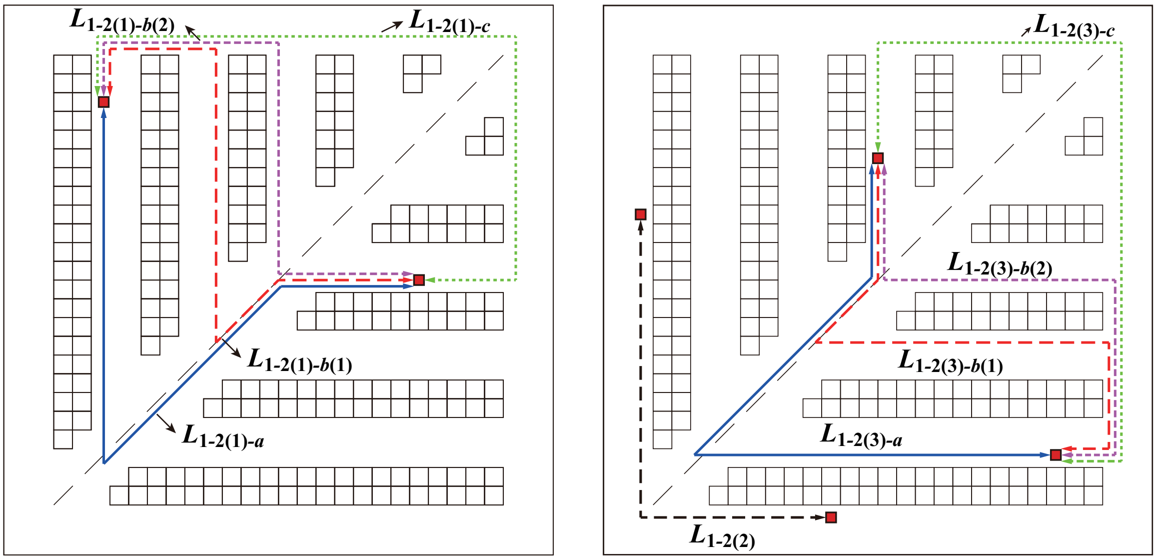

Fig. A2. Path planning of situation one in case 2

consistent with the method of $L_{1-2(1)-a}$.

$$\begin{aligned} L_{1-2(1)-c} = & \left(A + \frac{a_p}{2} - |xp_{i_1j_1k_1t_1}|\right) \\ & + \frac{a_p}{4} + \left(B + \frac{a_p}{2} - yp_{i_1j_1k_1t_1}\right) + \frac{a_p}{2} \\ & + \left(A + \frac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right) + \frac{a_p}{4} \\ & + \left(B + \frac{a_p}{2} - yp_{i_2j_2k_2t_2}\right) \end{aligned} \tag{A.15}$$

The calculation method of $L_{1-2(3)-c}$ is consistent with the method of $L_{1-2(1)-c}$.

L_1 is defined as follows:

$$\begin{aligned} L_1(x) = & \left(Am + \frac{a_p}{2} - yp_{i_1j_1k_1t_1}\right)\frac{1}{m} \\ & - \left(A + \frac{a_p}{2} - |xp_{i_1j_1k_1t_1}|\right) + \frac{\sqrt{1+m^2}}{m} \\ & \left| yp_{i_1j_1k_1t_1} - \left[B + \frac{a_p}{2} - \left(\frac{B}{m} + \frac{a_p}{2} - |x|\right)m\right]\right| \\ & + \left(\frac{B}{m} + \frac{a_p}{2} - |x|\right)m + \frac{a_p}{4} + |x - xp_{i_2j_2k_2t_2}| \\ & + \frac{a_p}{4} + \left(B + \frac{a_p}{2} - yp_{i_2j_2k_2t_2}\right) \end{aligned} \tag{A.16}$$

with:

$$\begin{cases} L_{1-2(1)-b(1)} = L_1(x_1) \\ L_{1-2(1)-b(2)} = L_1(x_2) \end{cases} \tag{A.17}$$

$$\max_{q\in Z}\left\{\left(q-\frac{1}{2}\right)e_w + [2q-1-(-1)^q]\frac{a_p}{4} + (-1)^q \left(\frac{e_w}{2} + \frac{a_p}{4}\right) < A + \frac{a_p}{2} - \left(Am + \frac{a_p}{2} - yp_{i_1j_1k_1t_1}\right)\frac{1}{m}\right\} \tag{A.18}$$

$$\begin{cases} x_1 = (q - \frac{1}{2})e_w + [2q-1-(-1)^q]\frac{a_p}{4} \\ \quad + (-1)^q(\frac{e_w}{2} + \frac{a_p}{4}) \\ x_2 = (q + \frac{1}{2})e_w + [2q+1-(-1)^{q+1}]\frac{a_p}{4} \\ \quad + (-1)^{q+1}(\frac{e_w}{2} + \frac{a_p}{4}) \end{cases} \tag{A.19}$$

L_2 is defined as follows:

$$\begin{aligned} L_2(y) = & \left(\frac{B}{m} + \frac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right)m \\ & - \left(B + \frac{a_p}{2} - yp_{i_2j_2k_2t_2}\right) + \sqrt{1+m^2} \\ & \left| |xp_{i_2j_2k_2t_2}| - \left[A + \frac{a_p}{2} - (Am + \frac{a_p}{2} - y)\frac{1}{m}\right]\right| \\ & + \left(Am + \frac{a_p}{2} - y\right)\frac{1}{m} + \frac{a_p}{4} + |y - yp_{i_1j_1k_1t_1}| \\ & + \frac{a_p}{4} + \left(A + \frac{a_p}{2} - |xp_{i_1j_1k_1t_1}|\right) \end{aligned} \tag{A.20}$$

with:

$$\begin{cases} L_{1-2(3)-b(1)} = L_2(y_1) \\ L_{1-2(3)-b(2)} = L_2(y_2) \end{cases} \tag{A.21}$$

$$\max_{q\in Z}\left\{\left(q-\frac{1}{2}\right)e_w + [2q-1-(-1)^q]\frac{a_p}{4} + (-1)^q \left(e_w + \frac{a_p}{2}\right) < B + \frac{a_p}{2} - \left(\frac{B}{m} + \frac{a_p}{2} - xp_{i_2j_2k_2t_2}\right)m\right\} \tag{A.22}$$

$$\begin{cases} y_1 = \left(q - \frac{1}{2}\right)e_w + [2q-1-(-1)^q]\frac{a_p}{4} \\ \quad + (-1)^q\left(e_w + \frac{a_p}{2}\right) \\ y_2 = \left(q + \frac{1}{2}\right)e_w + \left[2q+1-(-1)^{q+1}\right]\frac{a_p}{4} \\ \quad + (-1)^{q+1}\left(e_w + \frac{a_p}{2}\right) \end{cases} \tag{A.23}$$

Situation two: One picking point is in zone 1, and the other is in zone 3; or one is in zone 4, and the other is in zone 2.

The relevant parameters are shown in Fig. A3:

$$\begin{aligned} & D_4(i_1j_1k_1t_1, i_2j_2k_2t_2) \\ & \quad = \min(L_{1-3-a}, L_{1-3-b(1)}, L_{1-3-b(2)}, L_{1-3-c}) \end{aligned} \tag{A.24}$$

We assume that $Op_{i_1j_1k_1t_1}$ is in zone 1 and $Op_{i_2j_2k_2t_2}$ is

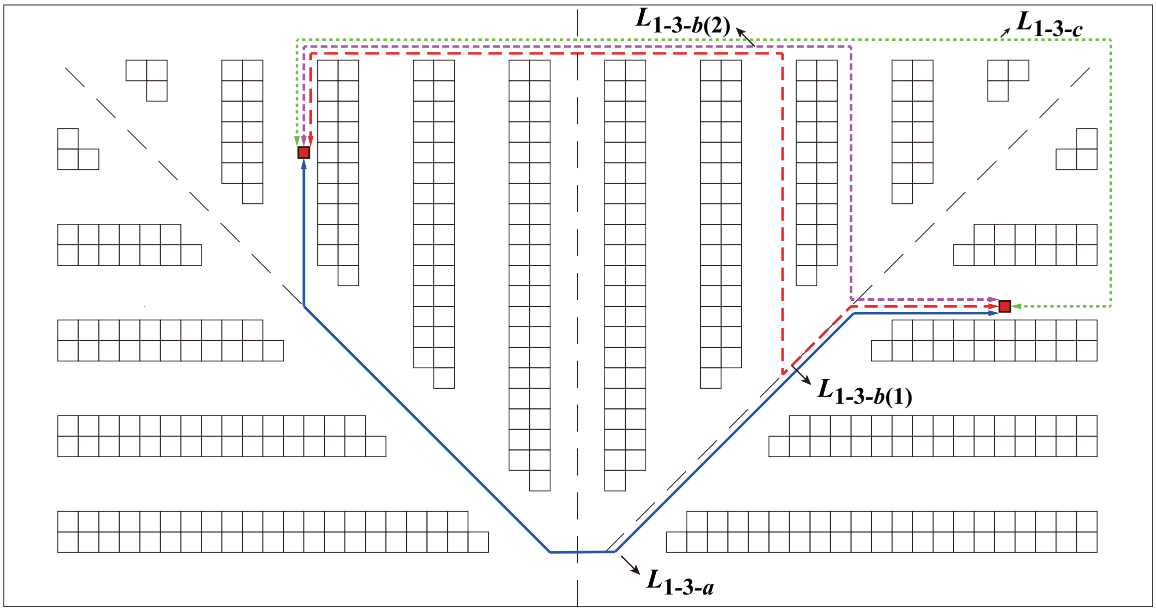

Fig. A3. Path planning of situation two in case 2

in zone 3; then:

$$\begin{aligned}L_{1-3-a} =& \left(Am + \frac{a_p}{2} - yp_{i_1j_1k_1t_1}\right)\frac{1}{m} \\ &- \left(A + \frac{a_p}{2} - |xp_{i_1j_1k_1t_1}|\right) \\ &+ \frac{\sqrt{1+m^2}}{m}\left|yp_{i_1j_1k_1t_1} - \frac{a_p}{2}\right| \\ &+ a_p + \sqrt{1+m^2}\left||xp_{i_2j_2k_2t_2}| - \frac{a_p}{2}\right| \\ &+ \left(\frac{B}{m} + \frac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right)m \\ &- \left(B + \frac{a_p}{2} - yp_{i_2j_2k_2t_2}\right)\end{aligned} \tag{A.25}$$

The calculation methods of $L_{1-3-b(1)}$, $L_{1-3-b(2)}$ and L_{1-3-c} are consistent with $L_{1-2(3)-b(1)}$ and $L_{1-2(3)-b(2)}$.

Situation three: One picking point is in zone 1, and the other is in zone 4.

The relevant parameters are shown in Fig. A4, and:

$$D_5(i_1j_1k_1t_1, i_2j_2k_2t_2) = \min(L_{1-4-a}, L_{1-4-b}) \tag{A.26}$$

We assume that $Op_{i_1j_1k_1t_1}$ is in zone 1 and $Op_{i_2j_2k_2t_2}$ is in zone 4, then:

$$\begin{aligned}L_{1-4-a} =& \left(Am + \frac{a_p}{2} - yp_{i_1j_1k_1t_1}\right)\frac{1}{m} \\ &- \left(A + \frac{a_p}{2} - |xp_{i_1j_1k_1t_1}|\right) \\ &+ \frac{\sqrt{1+m^2}}{m}\left|yp_{i_1j_1k_1t_1} - \frac{a_p}{2}\right| + a_p \\ &+ \frac{\sqrt{1+m^2}}{m}\left|yp_{i_2j_2k_2t_2} - \frac{a_p}{2}\right| \\ &+ \left(Am + \frac{a_p}{2} - yp_{i_2j_2k_2t_2}\right)\frac{1}{m} \\ &- \left(A + \frac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right)\end{aligned} \tag{A.27}$$

$$\begin{aligned}L_{1-4-b} =& \min\left[\min(a_1, b_1) + c_1, d_1\right] + e \\ &+ \min\left[\min(a_2, b_2) + c_2, d_2\right]\end{aligned} \tag{A.28}$$

The calculation method of $(a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2, e)$ is consistent with $L_{1-2(1)-b(1)}$ and $L_{1-2(1)-b(2)}$.

Situation four: One picking point is in zone 2, and the other is in zone 3.

The relevant parameters are shown in Fig. A5, and:

$$D_6(i_1j_1k_1t_1, i_2j_2k_2t_2) = \min(L_{2-3-a}, L_{2-3-b}) \tag{A.29}$$

We assume that $Op_{i_1j_1k_1t_1}$ is in zone 2 and $Op_{i_2j_2k_2t_2}$ is in zone 3, then:

$$\begin{aligned}L_{2-3-a} =& \left(\frac{B}{m} + \frac{a_p}{2} - |xp_{i_1j_1k_1t_1}|\right)m \\ &- \left(B + \frac{a_p}{2} - yp_{i_1j_1k_1t_1}\right) \\ &+ \sqrt{1+m^2}\left||xp_{i_1j_1k_1t_1}| - \frac{a_p}{2}\right| + a_p \\ &+ \sqrt{1+m^2}\left||xp_{i_2j_2k_2t_2}| - \frac{a_p}{2}\right| \\ &+ \left(\frac{B}{m} + \frac{a_p}{2} - |xp_{i_2j_2k_2t_2}|\right)m \\ &- \left(B + \frac{a_p}{2} - yp_{i_2j_2k_2t_2}\right)\end{aligned} \tag{A.30}$$

$$\begin{aligned}L_{2-3-b} =& \left(B + \frac{a_p}{2} - yp_{i_1j_1k_1t_1}\right) + \frac{a_p}{4} \\ &+ |xp_{i_1j_1k_1t_1} - xp_{i_2j_2k_2t_2}| \\ &+ \frac{a_p}{4} + \left(B + \frac{a_p}{2} - yp_{i_2j_2k_2t_2}\right)\end{aligned} \tag{A.31}$$

References

[1] K. R. Gue and R. D. Meller, "Aisle configurations for unit-load warehouses," *IIE Trans*, vol. 43, no. 10, pp. 703-720, 2011.

[2] L. M. Pohl, R. D. Meller, and K. R. Gue, "Optimizing fishbone aisles for dual-command operations in a warehouse," *Naval Research Logistics (NRL)*, vol. 56, no. 5, pp. 389-403, 2009.

[3] L. M. Pohl, R. D. Meller, and K. R. Gue, "Turnover-based storage in non-traditional unit-load warehouse designs," *IIE Trans*, vol. 43, no. 10, pp. 703-720, 2011.

[4] M. Çelk and H. Süral, "Order picking under random and turnover-based storage policies in fishbone aisle warehouses," *IIE Trans*, vol. 46, no. 3, pp. 283-300, 2013.

[5] L. F. Cardona, L. Rivera, and H. J. Martínez, "Analytical study of the Fishbone Warehouse layout," *International Journal of Logistics Research and Applications*, vol. 15, no. 6, pp. 365-388, 2012.

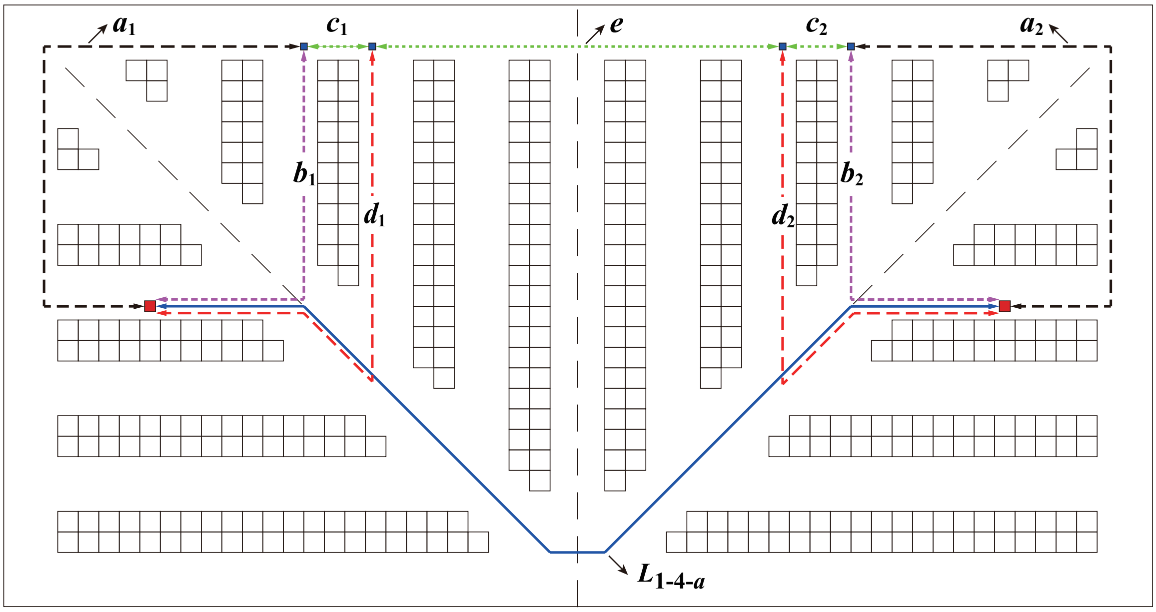

Fig. A4. Path planning of situation three in case 2

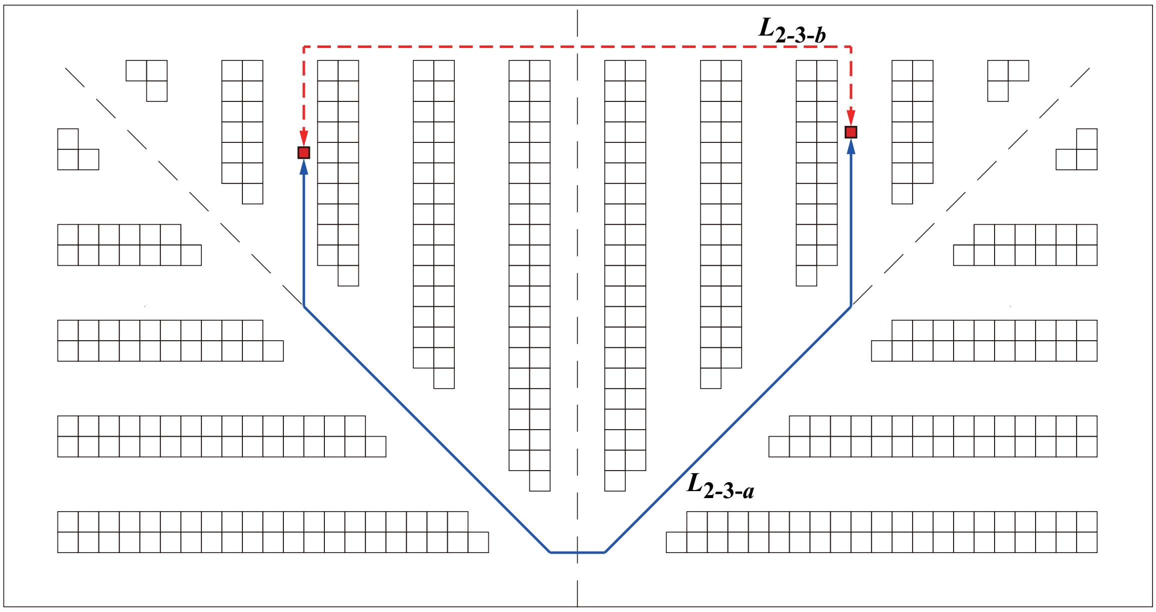

Fig. A5. Path planning of situation four in case 2

[6] N. V. Kumar and C. S. Kumar, "Development of collision free path planning algorithm for warehouse mobile robot,", *Procedia Computer Science*, vol. 133, pp. 456-463, 2018.
[7] Ö. Öztürkoğlu, K. R. Gue, and R. D. Meller, "Optimal unit-load warehouse designs for single-command operations," *IIE Trans*, vol. 44, no. 6, pp. 459-475, 2012.
[8] L. F. Cardona, D. F. Soto, L. Rivera, and H. J. Martinez, "Detailed design of fishbone warehouse layouts with vertical travel," *International Journal of Production Economics*, vol. 170, pp. 825-837, 2015.
[9] T. Van Gils, K. Ramaekers, A. Caris, and R. B. M. de Koster, "Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review," *European Journal of Operational Research*, vol. 267, no. 1, pp. 1-15, 2018.
[10] Y. G. Yu and R. B. M. De Koster, "Optimal zone boundaries for two-class-based compact three-dimensional automated storage and retrieval systems," *IIE Trans*, vol. 41, no. 3, pp. 194-208, 2009.
[11] C. L. Yang and T. P. Q. Nguyen, "Constrained clustering method for class-based storage location assignment in warehouse,", *Industrial Management & Data Systems*, vol. 116, no. 4, pp. 667-689, 2016.
[12] H. Brynzér and M. I. Johansson, "Storage location assignment: Using the product structure to reduce order picking times," *International Journal of Production Economics*, vol. 46-47, pp. 595-603, 1996.
[13] M.-K. Lee, "A storage assignment policy in a man-on-board automated storage/retrieval system," *International Journal of Production Research*, vol. 30, no. 10, pp. 2281-2292, 1992.
[14] J. Xiao and L. Zheng, "A correlated storage location assignment problem in a single-block-multi-aisles warehouse considering BOM information,", *International Journal of Production Research*, vol. 48, no. 5, pp. 1321-1338, 2010.
[15] R. Q. Zhang, M. Wang, and X. Pan, "New model of the storage location assignment problem considering demand correlation pattern,", *Computers & Industrial Engineering*, vol. 129, pp. 210-219, 2019.
[16] D. M. H. Chiang, C. P. Lin, and M. C. Chen, "The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres,", *Enterprise Information Systems*, vol. 5, no. 2, pp. 219-234, 2011.
[17] Y. F. Chuang, H. T. Lee, and Y. C. Lai, "Item-associated cluster assignment model on storage allocation problems,", *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 1171-1177, 2012.
[18] K. W. Pang and H. L. Chan, "Data mining-based algorithm for storage location assignment in a randomised warehouse,", *International Journal of Production Research*, vol. 55, no. 14, pp. 4035-4052, 2017.
[19] K. L. Keung, C. K. M. Lee, and P. Ji, "Data-driven order correlation pattern and storage location assignment in robotic mobile fulfillment

and process automation system,", *Advanced Engineering Informatics*, vol. 50, 2021.
[20] C. M. Chen, Y. M. Gong, R. B. M. de Koster, and J. A. E. E. van Nunen, "A Flexible Evaluative Framework for Order Picking Systems,", *Production and Operations Management*, vol. 19, no. 1, pp. 70-82, 2010.
[21] S. Henn and G. Wascher, "Tabu search heuristics for the order batching problem in manual order picking systems,", *European Journal of Operational Research*, vol. 222, no. 3, pp. 484-494, 1 2012.
[22] P. Kübler, C. H. Glock, and T. Bauernhansl, "A new iterative method for solving the joint dynamic storage location assignment, order batching and picker routing problem in manual picker-to-parts warehouses," *Computers & Industrial Engineering*, vol. 147, p. 106645, 2020.
[23] R. J. Mantel, P. C. Schuur, and S. S. Heragu, "Order oriented slotting: a new assignment strategy for warehouses,", *European Journal of Industrial Engineering*, vol. 1, no. 3, pp. 301-316, 2007.
[24] A. S. Dijkstra and K. J. Roodbergen, "Exact route-length formulas and a storage location assignment heuristic for picker-to-parts warehouses,", *Transportation Research Part E: Logistics and Transportation Review*, vol. 102, pp. 38-59, 2017.
[25] A. Silva, L. C. Coelho, M. Darvish, and J. Renaud, "Integrating storage location and order picking problems in warehouse planning," *Transportation Research Part E: Logistics and Transportation Review*, vol. 140, 2020.
[26] J. Won and S. Olafsson, "Joint order batching and order picking in warehouse operations,", *International Journal of Production Research*, vol. 43, no. 7, pp. 1427-1442, Apr 1 2005.
[27] C. A. Valle, J. E. Beasley, and A. S. da Cunha, "Optimally solving the joint order batching and picker routing problem,", *European Journal of Operational Research*, vol. 262, no. 3, pp. 817-834, 1 2017.
[28] O. Briant, H. Cambazard, D. Cattaruzza, N. Catusse, A.-L. Ladier, and M. Ogier, "An efficient and general approach for the joint order batching and picker routing problem," *European Journal of Operational Research*, vol. 285, no. 2, pp. 497-512, 2020.
[29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II,", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.
[30] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer Programming Formulation of Traveling Salesman Problems," *The Journal of the ACM*, vol. 7, no. 4, pp. 326–329, 1960.
[31] X. Z. Hongbo Li, Shuai Fu, and Yinyin Hu, "A Hybrid Algorithm Based on Ant Colony Optimization and Differential Evolution for Vehicle Routing Problem," *Engineering Letters*, vol. 29, no. 3, pp. 1201-1211, 2021.
[32] Y.-e. H. Lan-xue Dang, Qing-song Liu, and Yun-feng Kong, "A Hybrid Metaheuristic Algorithm for the Bi-objective School Bus Routing Problem," *IAENG International Journal of Computer Science*, vol. 46, no. 3, pp. 409-416, 2019.
[33] A. K. J. Olateju Samuel Olaniyi, Abdullahi Adinoyi Ibrahim, and Aderibigbe Felix Makanjuola, "On the Application of a Modified Genetic Algorithm for Solving Vehicle Routing Problems with Time Windows and Split Delivery," *IAENG International Journal of Applied Mathematics*, vol. 52, no. 1, pp. 101-109, 2022.
[34] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232-2248, 2009.
[35] T. A. Khan and S. H. Ling, "A novel hybrid gravitational search particle swarm optimization algorithm," *Engineering Applications of Artificial Intelligence*, vol. 102, pp. 104263, 2021.
[36] J. J. R. Reyes, E. L. Solano-Charris, and J. R. Montoya-Torres, "The storage location assignment problem: A literature review," *International Journal of Industrial Engineering Computations*, vol. 10, no. 2, pp. 199-224, 2019.
[37] Jie-Sheng Wang and J.-D. Song, "A Hybrid Algorithm Based on Gravitational Search and Particle Swarm Optimization Algorithm to Solve Function Optimization Problems," *Engineering Letters*, vol. 25, no. 1, pp. 22-29, 2017.
[38] Y. Wang and Z. Han, "Ant colony optimization for traveling salesman problem based on parameters optimization," *Applied Soft Computing*, vol. 107, pp. 107439, 2021.