doi:10.3772/j.issn.1006-6748.2023.04.006

Research on dual-command operation path optimization based on Flying-V warehouse layout①

LIU Jiansheng(刘建胜), * YUAN Bin *, YANG Zan② *, ZHONG RAY Y **
(* School of Advanced Manufacturing, Nanchang University, Nanchang 330031, P. R. China)
(** Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Hong Kong 999077, P. R. China)

Abstract

To enhance the efficiency of warehouse order management, this study investigates a dual-command operation mode in the Flying-V non-traditional warehouse layout. Three dual-command operation strategies are designed, and a dual-command operation path optimization model is established with the shortest path as the optimization goal. Furthermore, a genetic algorithm based on a dynamic decoding strategy is proposed. Simulation results demonstrate that the Flying-V layout warehouse management and access cooperation operation can reduce the operation time by an average of 25% – 35% compared with the single access operation path, and by an average of 13% – 23% compared with the 'deposit first and then pick' operation path. These findings provide evidence for the effectiveness of the optimization model and algorithm.

Key words: Flying-V, access collaboration, path optimization, dynamic decoding, genetic algorithm

0 Introduction

Logistics optimization in warehouse management can effectively reduce the operating costs for enterprises. Among all logistics processes, access operation is the most labor-intensive and costly, with costs accounting for up to 55% of the total operating expenses of a warehouse [1]. Previous studies have indicated that optimizing access operation is crucial for improving warehousing efficiency [2]. The time spent on access operation is a key indicator for measuring operational efficiency, and it is closely associated with the selection of access operation path. Therefore, reducing the travel distance of access operation is capable of enhancing the efficiency of warehousing management operations.

In recent years, researchers have studied access operation paths in warehouse by taking into account different warehouse layouts and order distributions under the assistance of heuristic algorithms such as genetic algorithms[3-4], ant colony algorithms[5-8], and particle swarm algorithms [9-12]. To reduce the access cost of goods, Ref. [13] proposed the Flying-V layout mode as an innovative warehouse layout, proving that this non-traditional layout can shorten travel distance by 10% – 20% compared with traditional layout in terms of picking efficiency. However, most current studies focus on single-command operation mode[14-19] by maximizing their respective operational efficiency without considering the association of access operations, where only deposit or picking operations are conducted during a single operation trip. Although this single-command operation mode is simple and easy to execute, it leads to problems such as idle trips and resource waste, indicating the need for improving overall operation efficiency. Therefore, this paper focuses on optimizing the dual-command operation path of Flying-V layout warehouse.

1 Problem description and mathematical model

1.1 Problem description

In this study, a batch of ordered goods required depositing while another batch required picking, and the objective is to complete the order operations with the shortest total operation path. The warehouse layout adopted Flying-V type layout, and the plane layout of the entire warehouse is shown in Fig. 1. The P&D

① Supported by the National Natural Science Foundation of China (51565036).
② To whom correspondence should be addressed. E-mail: yangzan@ncu.edu.cn.
Received on Jan. 18, 2023

(pick and deposit) point is the entrance and exit of the warehouse. To facilitate the study of warehouse management path optimization, certain assumptions have been made.

(1) During the operation, the freight vehicle has a load limit of Q, allowing for multiple operations to be carried out.

(2) It is assumed that walking distance on both the left and right sides of the passage are negligible.

(3) In addition, turning back and walking in the passage is permitted.

(4) The demand for goods in every order is less than that of the freight vehicle's load capacity, and the freight vehicle can only access each location once.

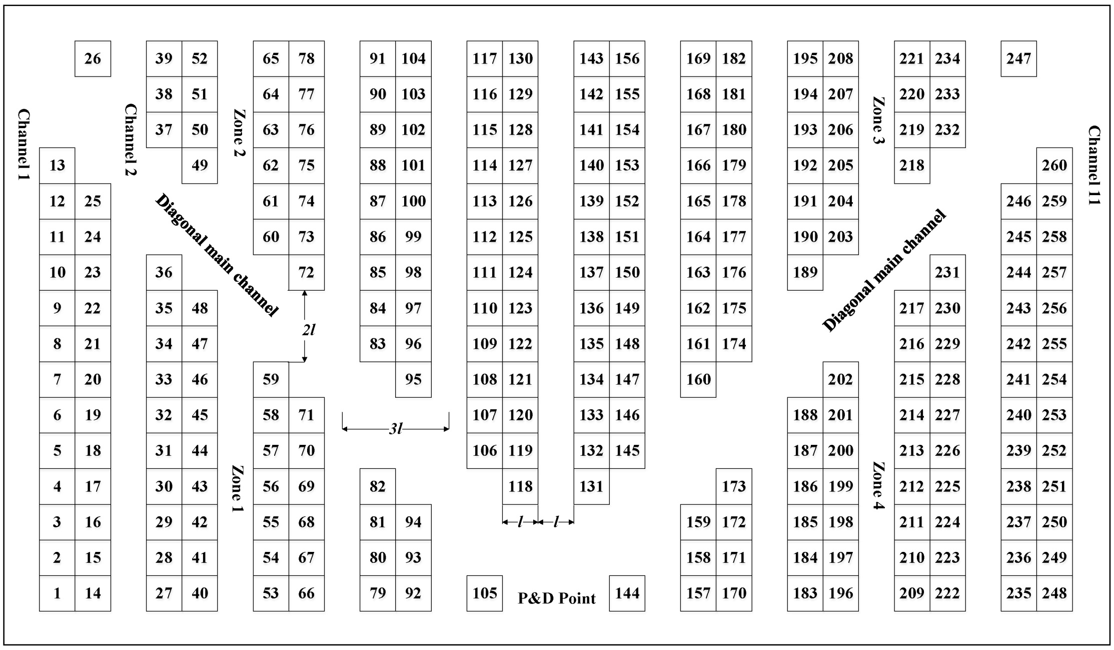

Fig. 1 Flying-V warehouse layout

1.2 Parameter design

Fig. 1 shows the picking area number and cargo space number. The cargo space number ranges from 1 to 260, from left to right and bottom to top, with the P&D point number being 0. The warehouse layout is divided into four picking areas, starting clockwise from the lower left corner of the warehouse, and is divided into Zone 1, Zone 2, Zone 3 and Zone 4. Regardless of the height of the cargo space, the length and width of the shelf are l, the width of the picking channel is l, and the width of the diagonal main channel is $2l$.

To represent the corresponding cargo space number, a virtual coordinate system is utilized in the plane layout. The array $\{k, x, y, z\}$ is employed, where $k(k = 1,2,3,4)$ indicates the cargo area number, $x(x = 1,2,3,...,11)$ represents the number of channels, $y(y = 1,2,3,...,y_{\max})$ depicts the number of rows of shelves starting from the diagonal main channel, and $z(z = 1,2)$ indicates the left and right sides of the channel; specifically, $z = 1$ denotes the left side of the channel, and $z = 2$ represents the right side of the channel. For example, $\{2,5,10,1\}$ represents Zone 2, the 5 th channel, the 10 th space from the diagonal main channel upward, the shelf on the left, i.e., cargo space number 102 in Fig. 1.

1.3 Distance matrix calculation

To optimize the distance to complete the order access operation, it is necessary to calculate the distances between any two points, including the distance between the P&D point and the cargo space point, as well as the distance between two cargo space points.

(1) Distance between the P&D point and the cargo space point i.

1) When the cargo space is located in Zone 1 (the same for Zone 4), that is, $k_i = 1$:

$$d_{0i} = \min\begin{cases} 2 \times |6 - x_i| \times 3l - y_i \times l \\ \sqrt{2} \times |6 - x_i| \times 3l + y_i \times l - 2\sqrt{2}l \end{cases} \tag{1}$$

2) When the cargo space point is located in Zone 2 (the same for Zone 3), that is, $k_i = 2$:

$$d_{0i} = \sqrt{2} \times |6 - x_i| \times 3l + y_i \times l \tag{2}$$

(2) Distance between any two cargo space points:

1) When two cargo space points are in Zone 1 (the same for Zone 4), $k_i = k_j = 1$:

$$d_{ij} = \begin{cases} |y_i - y_j| \times l, x_i = x_j \\ \min\begin{cases} 2 \times (6 - x_i) \times 3l - (y_i + y_j - 2) \times l \\ \sqrt{2} \times |x_j - x_i| \times 3l + (y_i + y_j - 2\sqrt{2}) \times l \end{cases} & x_i \neq x_j \end{cases} \tag{3}$$

2) When two cargo space points are in Zone 2 (the same for Zone 3), $k_i = k_j = 2$:

$$d_{ij} = \begin{cases} |y_i - y_j| \times l, x_i = x_j \\ \min\begin{cases} 2 \times (x_i - 1) \times 3l - (y_i + y_j - 2) \times l \\ \sqrt{2} \times |x_j - x_i| \times 3l + (y_i + y_j - 2\sqrt{2}) \times l \end{cases} & x_i \neq x_j \end{cases} \tag{4}$$

3) When two cargo space points are located in Zone 1 and Zone 2 respectively (the same for Zone 3 and Zone 4), $k_i = 1, k_j = 2$:

$$d_{ij} = \begin{cases} (y_i + y_j) \times l,\ x_i = x_j \\ \min\begin{cases} (x_i + x_j - 2 + |x_j - x_i|) \times 3l + (y_i - y_j + 2) \times l \\ (12 - x_i - x_j + |x_j - x_i|) \times 3l - (y_i - y_j - 2) \times l \\ \sqrt{2} \times |x_j - x_i| \times 3l + (y_i + y_j - 4\sqrt{2}) \times l,\ x_i \neq x_j \end{cases} \end{cases} \tag{5}$$

4) When two cargo space points are located in Zone 1 and Zone 3, respectively (the same for Zone 2 and Zone 4), $k_i = 1, k_j = 3$:

$$d_{ij} = \min\begin{cases} 2 \times |x_j - x_i| \times 3l - (y_i - y_j) \times l + 2l \\ 10 \times 3l + (y_i - y_j) \times l + 2l \\ \sqrt{2} \times |x_j - x_i| \times 3l + (y_i + y_j) \times l - (6\sqrt{2} - 4)l \end{cases} \tag{6}$$

5) When two cargo space points are located in Zone 1 and Zone 4, respectively, $k_i = 1, k_j = 4$:

$$d_{ij} = \min\begin{cases} 2 \times |x_j - x_i| \times 3l - (y_i + y_j) \times l + 2l \\ (2 \times |6 - x_i| + \sqrt{2} \times |x_j - 6|) \times 3l - (y_i - y_j) \times l - 2\sqrt{2}l \\ (\sqrt{2} \times |6 - x_i| + 2 \times |x_j - 6|) \times 3l + (y_i - y_j) \times l - 2\sqrt{2}l \\ \sqrt{2} \times |x_j - x_i| \times 3l + (y_i + y_j) \times l - (8\sqrt{2} - 4)l \end{cases} \tag{7}$$

6) When two cargo space points are located in Zone 2 and Zone 3, respectively, $k_i = 2, k_j = 3$:

$$d_{ij} = \begin{cases} |y_i - y_j| \times l, x_i = x_j = 6 \\ \min\begin{cases} 10 \times 3l - (y_i + y_j) \times l + 2l \\ \sqrt{2} \times |x_j - x_i| \times 3l + (y_i + y_j) \times l - (4\sqrt{2} - 4)l \end{cases} & x_j \neq x_i \neq 6 \end{cases} \tag{8}$$

1.4 Modeling

The goal of optimization is to minimize the distance to complete the order access process while returning to the entrance for multiple operations. The mathematical model for the path problem can be designed as follows.

Objective function is

$$S = \min \sum_{i \neq j \in \Omega} x_{ij} d_{ij} \tag{9}$$

Constraints:

$$\sum_{i \in \Omega} x_{ij} = 1, \forall j \in \Omega \tag{10}$$

$$\sum_{j \in \Omega} x_{ij} = 1, \forall i \in \Omega \tag{11}$$

$$x_{ij} \in \{0,1\}, \forall i,j \in \Omega \tag{12}$$

$$Qi \leqslant Q, \forall i \in \Omega \tag{13}$$

$$Q_i = Q_{i-1} + f_i q_i \tag{14}$$

$$f_i = \begin{cases} -1 & \text{deposit} \\ 1 & \text{pick} \end{cases} \tag{15}$$

Decision variables:

$$x_{ij} = \begin{cases} 1 & \text{go to } j \text{ after completing point } i \text{ task} \\ 0 & \text{do not go to } j \text{ after completing point } i \text{ task} \end{cases} \tag{16}$$

where,

S: total traveling distance when all order operations are completed;

$i,j \in \Omega$: all cargo spaces to be picked and the starting point; and $i = 0$ indicates the P&D point;

d_{ij}: the shortest distance between cargo space i and cargo space j, calculated according to Eqs (1) - (8);

Q_i: load when starting from point i;

Q_0: initial load from P&D point;

Q: maximum load;

q_i: required weight at cargo space point i;

The objective Eq. (9) seeks to minimize the distance required to complete all orders; Eq. (10) and Eq. (11) guarantee that each picking point has one and only one previous and subsequent task; Eq. (12) defines the range of values for the decision variables; Eq. (13) and Eq. (14) prohibit overloading during the operation.

2 Algorithm solution

To solve the aforementioned model, a dynamic decoding genetic algorithm is implemented. Algorithm 1 provides the corresponding pseudo-code, and the corresponding elaboration for the following steps are shown in subsections 2.1 - 2.6.

Algorithm 1 The dynamic decoding-based genetic algorithm	
Input:	Population size: N, Crossover probability: Pc, Mutation probability: Pm, Number of orders: *Num_orders*, Required weight at each point: q, Operation type: *label*, Maximum load: Q
Output:	Optimal individual: *xbest*
1.	Initialize population with random candidate solutions, shown in subsection 2.1.
2.	Decode (using Algorithm 2) and evaluate each candidate solution shown in subsection 2.2.
3.	$g = 0$
4.	While terminate condition is not satisfied do
5.	Select parents shown in subsection 2.3.
6.	Crossover operation shown in subsection 2.4.
7.	Mutation operation shown in subsection 2.5.
8.	Decode (using Algorithm 2) and evaluate new candidate solution shown in subsection 2.2.
9.	Select individuals for the next generation shown in subsection 2.3.
10.	$g = g + 1$
11.	End while

2.1 Initialization

To initiate the optimization process, the value for the population size N, cross probability Pc and mutation probability Pm are defined. The chromosome code is randomly generated as $1 \times No$, where No refers to the order quantity. This process is repeated N times to generate an $N \times No$ population.

2.2 Decoding

The natural number code is used, with numbers ranging from 1 to No and 0 for the P&D point number. The sequence of codes indicate the access sequence of the cargo space points.

If there is no load limit, the problem could be simplified into a standard TSP problem, which only requires visiting each cargo space point in sequence and returning to the starting P&D point without the need for additional decoding. However, due to the load limit, it is necessary to go back and forth to the starting point during the access operation. Therefore, 0 is inserted into the code sequence and the load is dynamically calculated to determine the position where 0 is inserted. The dynamic decoding steps are as follows.

(1) Considering the limit state, at a certain time during the access operation, all goods ordered in all cycles are on the freight vehicle and are decoded according to the load limit of the freight vehicle. If the freight vehicle carrying $1 - i$ orders is not overweight, the first cycle of decoding is $(0, p1, p2, \ldots, pi, 0)$.

(2) Cargo space point $i + 1$ is added to the decoding cycle to simulate the load of the previous $i + 1$ cargo space points. The order weight to be warehoused is taken as the initial load to simulate the access operation of each cargo space point and calculate the load of each cargo space point. If a middle point is overweight, it means that the decoding fails, and the first decoding cycle is still $(0, p1, p2, \ldots, pi, 0)$. If no overweight occurs during the intermediate process, it means that the decoding is successful, and the decoding is $(0, p1, p2, \ldots, pi, pi + 1, 0)$. Then cargo space point $i + 2$ is added to the decoding cycle, and the above steps are repeated until point $i + n$ becomes overweight in the simulation process. At this point, the decoding is considered as failure, and the next cycle of decoding starts.

(3) Steps (1) and (2) are repeated until all order points are decoded successfully.

Algorithm 2 presents the process of dynamic decoding.

Algorithm 2 The dynamic decoding algorithm

```
Input:  Individual: x, Number of orders: Num_orders, Required weight at each point: q, Operation type: label, Maximum load: Q
Output: Decoded individual: x_d
For i = 1 to Num_orders
Calculate the initial load of the first i orders Q_0 .
x_d = 0
For j = 1 to i
x_d = [x_d, x(i)]
If label(j) == 1
Q_i = Q_{i-1} + q(j)
Else
Q_i = Q_{i-1} - q(j)
End if
If Q_i <= Q
Continue
Else
   x_d = [x_d, x(i - 1)]
   save decode fragment x_d
break
End if
End for
End for
Restores the decoded fragment to a one-dimensional array x_d
```

2.3 Selection operation

The fitness value is the value of the objective function. To optimize the population and improve the fitness of individuals, the principles of 'survival of the fittest' in nature are followed. Inspired by the replication operation in bacterial foraging algorithms, half of the individuals with poor fitness value are directly eliminated, while the other half individuals with good fitness values are copied. To prevent the subsequent crossover and mutation operations from degrading the individuals with the best fitness value, an elite retention strategy is adopted. This strategy ensured that the fittest individuals are preserved in the population and not lost during the optimization process.

2.4 Crossover operation

To increase the diversity of the population and improve the global search ability, double-point crossover is adopted. This allows the same chromosome crossover operation to generate new chromosomes, which further enhances the optimization process. In double-point crossover, two crossing points are randomly selected on the two parent chromosomes. The chromosome between the two points is copied to the corresponding chromosome of the other parent, and the previous duplicated code is removed. This process increased the diversity of the population and allowed for a more efficient search for optimal solutions.

For example:

Parent 1: 1 2 | 3 4 5 6 7 8 | 9

Parent 2: 9 8 | 7 6 5 4 3 2 | 1

Copy intermediate codes and delete duplicate codes:

1 ~~2 3 4 5 6 7~~ 8 9 7 6 5 4 3 2

9 ~~8 7 6 5 4 3~~ 2 1 3 4 5 6 7 8

Offspring 1: 1 8 9 7 6 5 4 3 2

Offspring 2: 9 2 1 3 4 5 6 7 8

2.5 Mutation operation

The mutation operation uses double-point exchange mutation, which further increases the diversity of the population. In this operation, two point are randomly generated in the chromosome. The codes of the two points are then exchanged to complete the mutation operation. This approach allowes for the exploration of new solutions and prevents the population from getting trapped in a local optimum. By introducing random changes to the chromosomes, the algorithm is able to search for more optimal solutions across the solution space.

For example:

Before : 1 2 | 3 4 5 6 7 8 | 9

After : 1 2 | 8 4 5 6 7 3 | 9

3 Simulation experiment

Experimental environment: Windows 10 operating system, Intel (R) Core (TM) i5-10400 CPU @ 2.9 GHz processor, 32.0 GB RAM, developed with Matlab R2018a.

To demonstrate the effectiveness of the proposed algorithm, a randomized example with 20 orders and their corresponding demands are created, which is shown in Table 1. The cargo space number is designed according to the model parameters, and the maximum load (Q) of the freight vehicle is set to 20 kg. For example, the second order requires picking 4 kg of goods from the No. 32 cargo space. To ensure a sufficient population size to 200, which is 10 times the number of orders, set the evolution times to 500. The selection of crossover and mutation probability is determined through experiments. An orthogonal table is used to select the crossover probability ranging from 0.1 to 0.9 with an interval of 0.1 and the variation probability from 0.01 to 0.1 with an interval of 0.01. The optimal values are obtained by changing the parameters and running the process 20 times. Calculate the average value and find that the optimal crossover probability is 0.8, and the optimal mutation probability is 0.1.

The reason for selecting a large probability of crossover and mutation is analyzed. Since the replication is used for selection, the average fitness value of the population, i. e., the objective value, would decrease rapidly, but at the same time, the diversity of the population and the global search ability decrease rapidly. Therefore, selecting a large probability of crossover and mutation can effectively increase the diversity of the population and the global search ability, leading to better optimization results.

Table 1 Order demand

SN	Cargo space No.	Demanded weight/kg	Operation type
1	22	2	Deposit
2	32	4	Pick
3	34	5	Deposit
4	40	1	Pick
5	91	6	Pick
6	92	2	Pick
7	99	4	Pick
8	112	6	Pick
9	119	3	Pick
10	123	3	Pick
11	127	6	Deposit
12	146	3	Pick
13	152	8	Deposit
14	165	10	Deposit
15	170	7	Deposit
16	174	3	Pick
17	175	6	Pick
18	190	1	Pick
19	241	9	Deposit
20	244	9	Deposit

Three different access operation schemes are adopted, and genetic algorithms are used to solve the optimal path.

Mode 1: separated deposit and picking operations, with the deposit order operation and picking order operation conducted separately. The shortest operation distance calculated is 277.55.

Mode 2: 'deposit first and then pick' operation. After all the goods on the freight vehicle are deposited, the freight vehicle does not return to the entrance and exit but continues with the picking operation. The shortest operation distance calculated is 234.68.

Mode 3: the deposit and picking operations are completed simultaneously in an access collaboration operation. The shortest operation distance calculated is 190.40.

Calculation results are presented in Table 2 and Figs 2 – 4.

Table 2 Optimization results of three different modes

Mode 1		Mode 2		Mode 3	
Path	Load/kg	Path	Load/kg	Path	Load/kg
0		0		0	
34	13	241	18	123	6
22	8	244	9	127	9
127	6	175	0	91	3
0	0	190	6	99	9
241	18	174	7	112	13
244	9	112	10	0	19
0	0	99	16	92	7
152	18	0	20	40	9
165	10	127	13	32	10
0	0	22	7	34	14
170	7	34	5	22	9
0	0	32	0	0	7
146	0	40	4	165	18
190	3	92	5	152	8
175	4	91	7	146	0
174	10	123	13	0	3
119	13	119	16	241	18
0	16	0	19	244	9
112	0	165	18	175	0
99	6	152	8	190	6
91	10	146	0	174	7
123	16	0	3	119	10
0	19	170	7	0	13
92	0	0	0	170	7
40	2			0	0
32	3				
0	7				

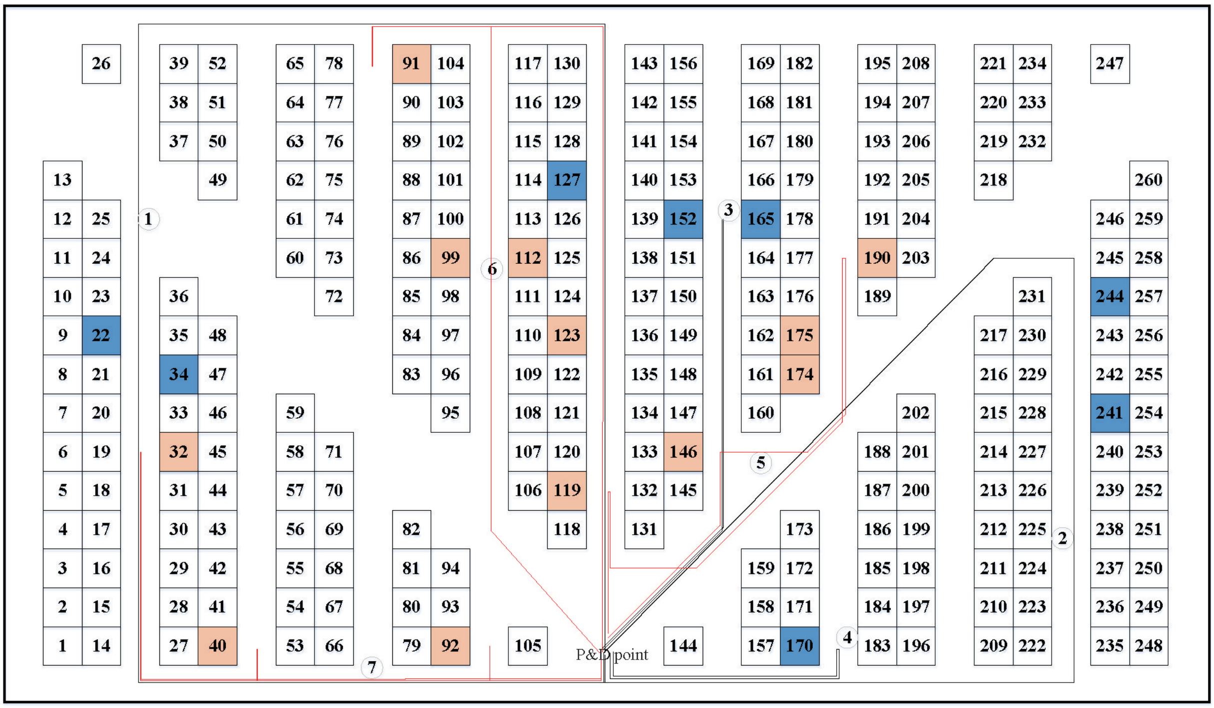

Fig. 2 Optimization path of Mode 1

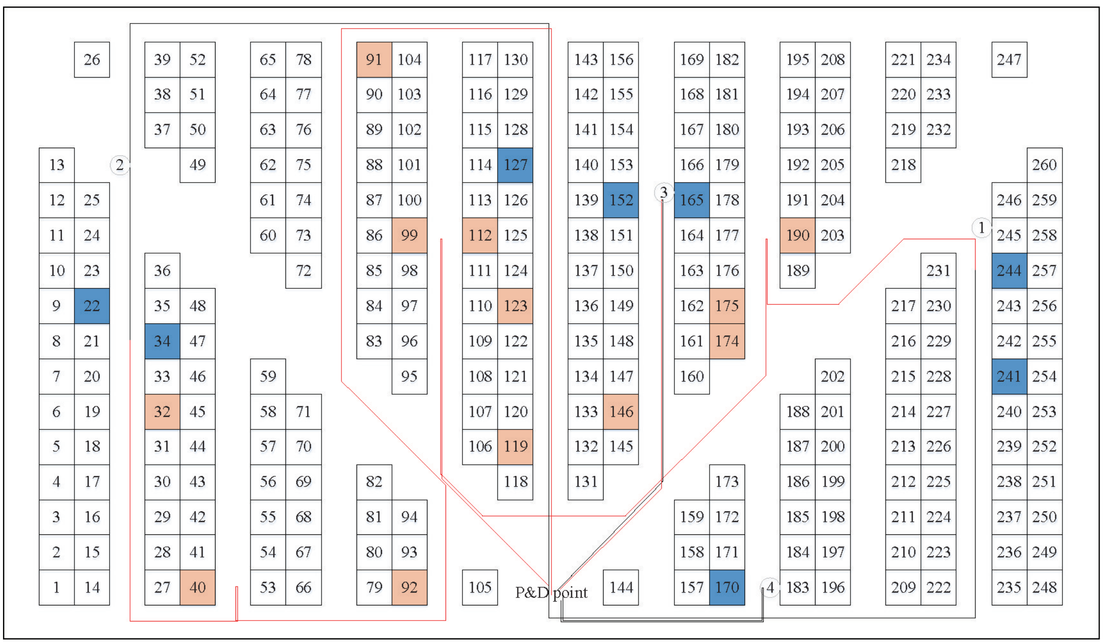

Fig. 3 Optimization path of Mode 2

Note: in the table, the freight vehicle follows the path 0→34→22→127→0, starting from the entrance and carrying 13 kg of cargo. When arriving at cargo space No. 34, 5 kg of cargo is deposited, and when reaching cargo space No. 22, 2 kg of cargo is deposited. Finally, 6 kg of cargo is deposited in cargo space No. 127 before returning to the entrance and exit to load cargo. In the figures, the dotted line indicates picking cargo for stock out, while the solid line indicates depositing of cargo.

The calculation results, Table 2, and the simulation path diagrams show that all three modes are operated with maximum load to reduce trips to and from the entrance and exit and shorten the operation path. Mode 1 has the longest path, while Mode 2 is slightly shorter with some optimization. Mode 3 has the shortest path, which is 31.4% shorter than Mode 1 and 18.8% shorter than Mode 2. The reason is that Mode 1 has no-load when travelling to and from the entrance and exit. After deposit, it returns to the entrance and exit without any load, and when picking the cargo, it also goes to the cargo spaces with no load. Although Mode 2 can avoid no-load at the entrance and exit, it is not optimized as a whole. Mode 3 is optimized as a whole while avoiding no-load, resulting in the most optimal outcome.

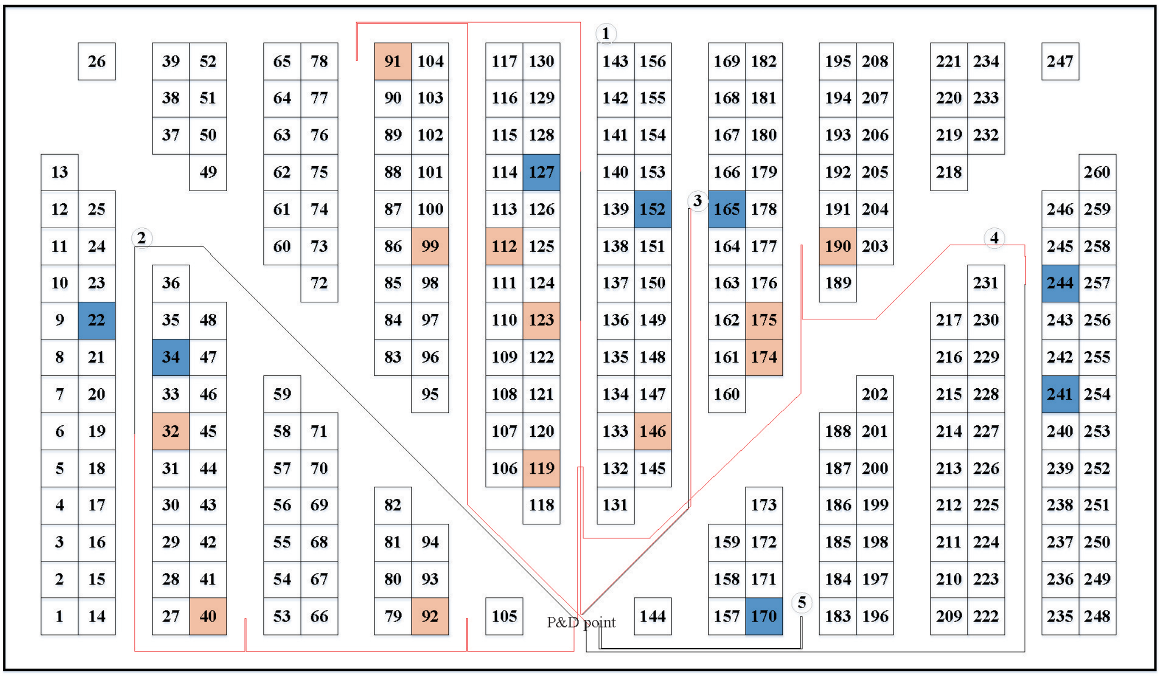

Fig. 4 Optimization path of Mode 3

To prove the effectiveness of the algorithm for access cooperative operation, numerous experiments have been conducted. The orders between 20 and 100 are randomly generated for calculation. For each example of different order quantity, the calculation is repeated 100 times, and the average value is calculated, as shown in Table 3 and Fig. 5. The experimental results show that in the non-traditional Flying-V warehouse layout mode, the operation in Mode 3 can be shortened by an average of 25% – 35% compared with the operation path in Mode 1, and 13% – 23% on average compared with the operation path in Mode 2. With an increase in order size, the optimization effect of Mode 3 becomes better.

4 Conclusion

This paper establishes a Flying-V layout warehouse path optimization model for dual-command operation path optimization of Flying-V layout warehouse management and proposes a dynamic decoding genetic algorithm. The simulation optimization experiment is conducted by randomly generating orders, and the optimization paths of three solutions, namely, separated operation of deposit and picking, ‘deposit first and then pick’ operation, and access collaboration operation, are calculated. The experimental results show that the access collaboration of dual-command operation can effectively reduce no-load, shorten the path, and improve efficiency.

Table 3 The average of 20 independent runs of three modes for different number of orders

Number of orders	Mode 1	Mode 2	Mode 3
20	294.55	238.53	195.43
30	415.47	342.46	283.23
40	528.34	467.51	387.63
50	658.49	537.46	483.14
60	780.15	716.72	571.09
70	957.47	890.08	720.88
80	1116.91	1051.29	819.65
90	1282.01	1188.09	955.50
100	1450.14	1386.69	1100.46

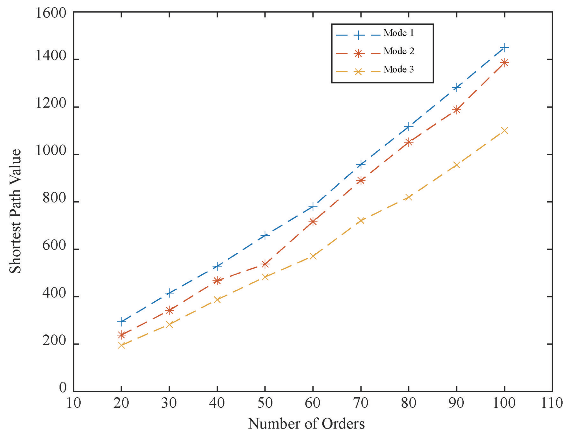

Fig. 5 Average optimization results of three modes for different number of orders

References

[1] RICCARDO A, MARCO B, MAURO G, et al. Multi-objective warehouse building design to optimize the cycle time, total cost, and carbon footprint[J]. The International Journal of Advanced Manufacturing Technology, 2017, 92(1-4): 839-854.

[2] BHAVIN S, VIVEK K. A comprehensive review of warehouse operational issues[J]. International Journal of Logistics Systems and Management, 2017, 26(3)1-33.

[3] TSAI C Y, LIOUJ J H, HUANG T M. Using a multiple-GA method to solve the batch picking problem: considering travel distance and order due time[J]. International Journal of Production Research, 2008, 46(22): 6533-6555.

[4] ZHAI L, FENG S. A novel evacuation path planning method based on improved genetic algorithm[J]. Journal of Intelligent and Fuzzy Systems, 2022, 42(3): 1813-1823.

[5] FANGYU C, HONGWEI W, CHAO Q, et al. An ant colony optimization routing algorithm for two order pickers with congestion consideration [J]. Computers and Industrial Engineering, 2013, 66(1): 77-85.

[6] ROBERTA D S, ROBERTO M, GIUSEPPE V, et al. An adapted ant colony optimization algorithm for the minimization of the travel distance of pickers in manual warehouses[J]. European Journal of Operational Research, 2018, 267(1): 120-137.

[7] DI CAPRIO D, EBRAHIMNEJAD A, ALREZAAMIRI H, et al. A novel ant colony algorithm for solving shortest path problems with fuzzy arc weights[J]. Alexandria Engineering Journal, 2022, 61(5): 3403-3415.

[8] MASOUMI Z, VAN GENDEREN J, SADEGHI N A. An improved ant colony optimization-based algorithm for user-centric multi-objective path planning for ubiquitous environments[J]. Geocarto International, 2021, 36(2): 137-154.

[9] YAO B, HU P, YANG S. An improved particle swarm optimization for the automobile spare part warehouse location problem[J]. Mathematical Problems in Engineering, 2013(15): 1-6.

[10] YANG L, ZHENG Y, XU Y, et al. Research on location assignment model of intelligent warehouse with RFID and improved particle swarm optimization algorithm[C]// International Conference on Computer Systems, Electronics and Control. Dalian: IEEE, 2017: 1262-1266.

[11] BHUNIA A K, SHAIKH A A. Investigation of two-warehouse inventory problems in interval environment under inflation via particle swarm optimization[J]. Mathematical and Computer Modelling of Dynamical Systems, 2016, 22(2): 160-179.

[12] WEI C, JI Z, CAI B. Particle swarm optimization for cooperative multi-robot task allocation: a multi-objective approach [J]. IEEE Robotics and Automation Letters, 2020, 5(2): 2530-2537.

[13] KEVIN R G, RUSSELL D M. Aisle configurations for unit-load warehouses[J]. IIE Transactions, 2009, 41(3): 171-182.

[14] ALEJANDRO CANO J, ALBERTO CORREA-ESPINAL A, ANDRES GOMEZ-MONTOYA R, et al. Genetic algorithms for the picker routing problem in multi-block warehouses[J]. Business Information Systems, 2019, 353: 313-322.

[15] MARCO B, FRANCESCO G G, MAURO G, et al. Integration of single and dual command operations in non-traditional warehouse design[J]. The International Journal of Advanced Manufacturing Technology, 2020, 111(9): 2461-2473.

[16] PAN J C, SHIH P, WU M. Storage assignment problem with travel distance and blocking considerations for a picker-to-part order picking system[J]. Computers and Industrial Engineering, 2012, 62(2): 527-535.

[17] SUBBIAH S, SCHOPPMEYER C, VALDÈS J M D L, et al. Optimal management of shuttle robots in a high-rise warehouse using timed automata models[J]. IFAC Proceedings Volumes, 2013, 46(9): 1358-1363.

[18] LIAN Y, XIE W, ZHANG L. A probabilistic time-constrained based heuristic path planning algorithm in warehouse multi-AGV systems [J]. IFAC PapersOnLine, 2020, 53(2): 2538-2543.

[19] ZHOU L, FAN X, WANG J, et al. A random classified-storage picking path model for V-type storage layout[J]. Complexity, 2020(26): 1-12.

LIU Jiansheng, born in 1978. He received his Ph. D degree from Nanchang University in 2009. He studied in Auburn University in USA as a visiting scholar granted by Chinese Scholarship Council. His research interests including digital and intelligent manufacturing, facility layout and logistics management and optimization.