Applied Science and Innovative Research
ISSN 2474-4972 (Print) ISSN 2474-4980 (Online)
Vol. 6, No. 2, 2022
www.scholink.org/ojs/index.php/asir

Original Paper

Optimization of Warehouse Picking Based on TSP

Yuhan Wang[1], Yanshan Liao[1], Junhong Kang[1], Yu Dang[1] & Zixin Jiang[1]

[1] Queen Mary University of London Engineering School, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

Received: March 9, 2022 Accepted: May 3, 2022 Online Published: May 23, 2022

doi:10.22158/asir.v6n2p85 URL: http://doi.org/10.22158/asir.v6n2p85

Abstract

Modern warehouse management has entered the era of information intelligence, which requires the establishment of a set of accurate, reasonable and efficient warehouse management methods, to improve the efficiency of workers.

In this paper, a series of TSP (Traveling salesman problem) algorithms are used to further optimize the process efficiency of goods removal from the warehouse. Based on Freudian algorithm, the distance problem between cargo grid and recheck station was solved in the first problem. Founded on the genetic algorithm in TSP, specific application problems were solved in two, three and four questions.

For problem 1, by observing the relation between the coordinates given and the actual route, all the coordinates of cargo grid were divided into two kinds: odd and even columns, and the two kinds of coordinates were processed, respectively. After coordinate processing, the distance matrix between each lattice can be obtained by using Freudian algorithm. At the same time, due to the small number of recheck stations and their regular distribution, the distance matrix between each recheck station and each cargo grid can be obtained by manually classifying the coordinates of the recheck stations and applying Freud algorithm again. Output all the matrices in the same EXCEL table, the distance matrix between the 3013 elements can be obtained. (See Appendix 1 for the distance matrix and Appendix 1 for the algorithm).

For problem 2, this problem is a unidirectional TSP problem by macro analysis. Firstly, the distance matrix of the required point was called from the solution of problem 1, which was imported into LINGO program to establish 0-1 decision variables, and the objective function is established to solve according to the principles of single-direction connection and loop-breaking, to obtain the connection sequence and loop distance. Select the cargo lattice back to the initial recheck station and connect it with the nearest recheck station, to complete the task of breaking the ring, solving the connection sequence and the total distance length. The outbound time can be divided into three parts: (1) journey *time; (2) Pick up time; (3) Packing time, group calculation, sum can get the result. The total distance is 382.5m (about 1254.92 ft) and the total minimum time is 462 seconds (about 7 and a half minutes).*

For problem 3, distance matrix of the required point was called from the solution of problem 1, and imported into LINGO program, afterwards, 0-1 decision variable and objective function were established according to the principles of single-direction connection and loop-breaking to solve, and then the connection sequence and loop distance was obtained. Select the cargo lattice back to the initial recheck station and connect it with the nearest recheck station, to complete the task of breaking the ring, solving the connection sequence and the total distance length. Compared with Question 2, Question 3 specifies the available recheck station and the total time needed to complete the shipment. During the calculation of each task order, it was classified and discussed (double-starting point operation), and the overall optimization was carried out according to the starting point and end point of each task order, and the total shortest time was 2288.6 seconds (about 38 minutes) (See the attachment for the results of this question).

For question 4, queuing theory is needed. To reduce the total outbound time, it is necessary to give priority to orders with abbreviated time. Firstly, the shortest outbound time of 49 task orders should be calculated, and the time should be arranged in order from short to long. After that, according to the order of the task list, nine pickers pick goods in a certain order. Hence, a variable should be introduced to preserve the status of the recheck table and the remaining time currently occupied.

Keywords

Freud algorithm, TSP (Traveling salesman problem), 0-1 linear dynamic programming problem

1. The Proposal and Restatement of the Problem

1.1 Proposal of the Problem

Modern warehouse management has entered the era of information and intelligence, which requires the establishment of a set of accurate, reasonable and efficient management and operation of warehouse methods. By optimizing the goods delivery process, it can effectively improve workers' work efficiency and save time. It is also of great significance for enterprises to distribute posts, formulate work performance and carry out scientific and reasonable warehouse layout.

1.2 Restatement of the Problem

After a customer order of an e-commerce company is placed into the warehouse, the goods are taken off the shelves and sent out of the warehouse, which includes five processes: positioning, grouping, picking, rechecking, packing.

Question 1: Figure 1 shows an example of a picker's route in a warehouse (L1-L7). When picking up the goods in the warehouse, the picker needs to walk between the goods and recheck station. A method should be designed to calculate the distance between 3000 grids and 3013 elements of 13 recheck stations according to the distance marked in the figure.

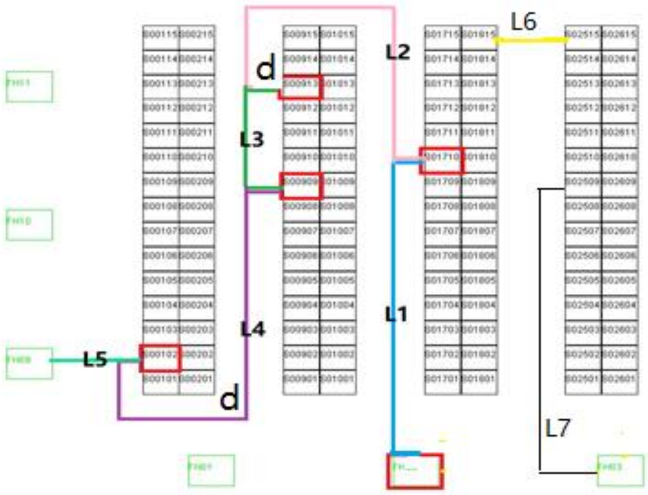

Figure 1. Distance Diagram in the Lower-left Corner of the Warehouse

Question 2: When all recheck stations work normally, the ideal picking route for P should be designed, including the access sequence of the grid and the return recheck station, and the time required for the exit of the warehouse should be calculated.

Question 3: It is assumed that two recheck stations (FH03 and FH11) work normally and five task orders (T0002-T0006) are waiting to be received. The pick-up clerk P is still responsible for picking up the goods, and the initial position of P is fh03. Through modeling and optimization, P’s task receiving sequence and planning ideal sorting route need to be specified so that these tasks can be transported out of the warehouse as soon as possible. In addition, the time required to complete the shipment also needs to be calculated.

Question 4: If 4 recheck stations (FH01, FH03, FH10, and FH12) work normally, 49 task orders (T0001-T0049) are waiting for picking, and 9 pickers (P1-P9) is responsible for picking, Each sorter should be assigned task orders and initial sort recheck stations and ideal sort routes. After 49 task orders are made, the time required to complete the warehouse shall also be calculated.

Note: In questions 3 and 4, when a picker has multiple tasks, the task can only be completed one by one, and cannot pick up goods from another task in the process of completing another task.

2. Analysis of the Problem

2.1 Analysis of Problem 1

For the first question, according to the requirements of the question, it was found that the coordinates of the pallet and the recheck table could not be used directly, so coordinate processing should be carried out according to the actual connection situation. As for freight grid coordinates, by observing the relationship between the given coordinates and the actual route, all freight grid coordinates can be divided into two categories: odd column and even column (macroscopically, that is, the left and right freight grids of the same shelf), as shown in Figure 2. The coordinates of these two categories are treated in diverse ways.

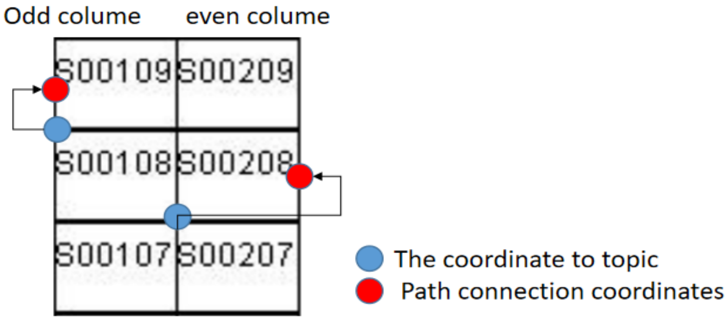

Figure 2. Processing of Odd/even Column Coordinates

Following the process shown in Figure 2, the new coordinates of all cargo grids can be obtained. Due to the topic request, the geometric distance between coordinates cannot be calculated directly here, but similar to "walk straight, turn right", so Freud algorithm was utilized (see Appendix 2), the cargo case further classification was discussed and the distance between the matrix was obtained, which would meet the topic request spontaneously.

For the recheck table, its treatment is similar to the cargo case, but slightly different. According to the requirements of the topic, as shown in Figure 3, it can be seen that four coordinate points are meeting the requirements on the same recheck table.

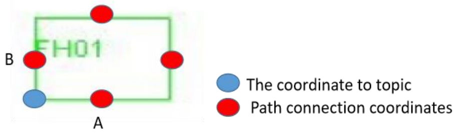

Figure 3. Coordinates of Recheck Table Which Meet the Requirement

Of course, it is unnecessary to consider 4 path connection coordinates for each recheck station. For FH01 recheck station, it is located below all cargo compartments. The connection of the midpoint in Figure 3 can be directly ignored. Similarly, for the FH09 recheck station (not marked in the figure), which is located to the left of all cargo compartments, the connection of the midpoint in Figure 3b can be directly ignored. However, for the whole program writing, because of the small size of the recheck table lattice, it is not necessary to carry out the above discussion. After this step is completed, the Freudian algorithm (see Appendix 2 for details) is used to classify and discuss the connection between cargo grids and recheck stations, and between recheck stations and recheck stations (see 5.1.2 for details), and the distance matrix between all cargo grids and recheck stations and each recheck station can be obtained. Output all the above distance matrices in the same EXCEL table, the answer to the question can be obtained, which is asymmetric matrix of 3013*3013.

2.2 Analysis of Question 2

The second problem, generally, this problem is a basic single-direction TSP problem, so it can be solved by establishing a linear programming model through LINGO. To begin with, according to the initial recheck table and the pick-up grid in the task list, the sequence number of its elements was obtained from the first result, to facilitate the establishment of a small-scale distance matrix (for example, in question 2, the distance matrix of 24*24 should be established). After the distance matrix was obtained, it was then imported into the LINGO program. For the LINGO program, the corresponding 0-1 decision variable should be established (1 for connection, 0 for non-connection). According to the principle of one-way connection and loop-breaking, the objective function should be established to solve. After the calculation results, the order of the cargo grid connection was determined, and the destination of the cargo grid position, at the same time can be calculated to focus on the cargo grid and its nearest distance, and finally, the total distance can be calculated. To solve the total outbound time, the outbound time is divided into three parts: (1) The total outbound time has been calculated in the previous step, and the speed is known, this part of the time can be calculated; (2) Pickup time (the number of goods in each case is known, and this part of the time can be directly calculated by calling the function in EXCEL.); (3) Packing time (the time is fixed, and the packing time of each order is 30 seconds), calculate the total time respectively.

MATLAB software was utilized to realize the ant colony algorithm in the subsequent solving process. Compared with LINGO programming, this method is slightly complicated, but the solving speed is relatively fast, and the data in question 1 can be directly used as the algorithm supplement of question 2 (Question 3 and question 4). However, it also has significant limitations, which will be briefly described in 6.1.3.

2.3 Analysis of Question 3

For question 3, compared with question 2, specified the available recheck station but needs to calculate the shortest time to complete the warehouse exit. Since there is only one picker, and each picker can only execute one task order at a time, from the perspective of calculation time, Question 3 is the repetition of Question 2, and the process required for calculating and executing each task order is the same as question 2. Firstly, according to the sequence number of elements in the selected task list, a small-scale distance matrix is established and imported into the LINGO program. In The LINGO program, the corresponding 0-1 decision variables (1 for connection, 0 for non-connection) are established. According to the principle of one-way link and loop-breaking, the objective function is established to solve the problem. After obtaining the calculation results, determine the location of the terminal lattice, calculate the distance between the terminal lattice and the two available recheck tables, and finally calculate the total distance. When solving the total outbound time, the required method is the same as that in question 2.

The queuing theory should be considered to solve the shortest total outbound time. According to the procedure above, how long it takes to execute each order can be figured out and sorted. At the same time, the starting point and endpoint of each order execution should be considered (the distance between the starting point and endpoint is long, so it takes a long time), and the final sorting result and total time can be obtained after reasonable sorting. (The packing time is relatively short compared to the time it takes to execute each task order. During the packing time, the same picker cannot complete the picking task in the next task order.)

2.4 Analysis of Question 4

For question 4, the question is a further expansion of question 3, which can be divided into two parts. In the first part, the single distance matrix of each task was calculated based on the existing data, and the shortest time required by each task was calculated. In the second part, the queuing theory was used to sort the task order from short to long according to the shortest time required to complete the task order and assign it to each picker according to the task order time and connection execution order. In addition, a variable should be introduced to preserve the status of the recheck table and the remaining time currently occupied. Every time the picker arrives at the recheck table, a judgment needs to be operated.

3. Assumptions of the Model

1. When walking around the broken line of the obstacle, take d=750mm for both transverse and vertical offset;

2. The distance between the two recheck stations is simplified as the sum of the absolute values of the coordinate differences between the two check stations. For example, coordinate A (x_1, y_1) of the recheck station and coordinate B (x_2, y_2) of the recheck station, the distance between the two check stations is $|x_2 - x_1| + |y_2 - y_1|$;

3. The distance between the cargo lattice and the recheck station is simplified as the distance between the midpoint of the cargo lattice and the midpoint of the nearest edge of the recheck station;

4. The shelves and the recheck table are obstructions and are not passable, while the rest are passable. Do not consider the size of the truck, shelves, and the height of the recheck table.

4. Symbol Description

Symbol	Symbol description	Unit
$(\boldsymbol{i}, \boldsymbol{j})$	Matrix coordinates row I and column J	/
$\boldsymbol{x_{ij}}$	0-1 Decision variables	/
$\boldsymbol{minZ}\&\boldsymbol{Z}$	Objective function distance & actual total distance	**mm**
$\boldsymbol{d_{ij}}$	The distance from lattice I to j	**mm**
$\boldsymbol{u_i}$	Circle analysis	/
$\boldsymbol{T}\&\boldsymbol{t_i}$	Total time & segment time	**s**
$\boldsymbol{v}$	Picker’s walking speed	$\mathbf{m/s}$

5. The Establishment and Solution of the Model

5.1 Problem 1: Model Establishment and Solution

5.1.1 Modeling and Solving of Problem 1

For the first question, according to the requirements of the question, it was found that the coordinates of the pallet and the check table could not be used directly, so coordinate processing should be carried out according to the actual connection situation.

The following is the first step to dealing with the coordinates of the cargo lattice and solving the path length between cargo lattice and cargo lattice:

As for freight grid coordinates, by observing the relationship between the given coordinates and the actual route, all freight grid coordinates can be divided into two categories: odd column and even column (macroscopically, that is, the left and right freight grids of the same shelf), as shown in Figure 4. The coordinates of these two categories are treated in different ways.

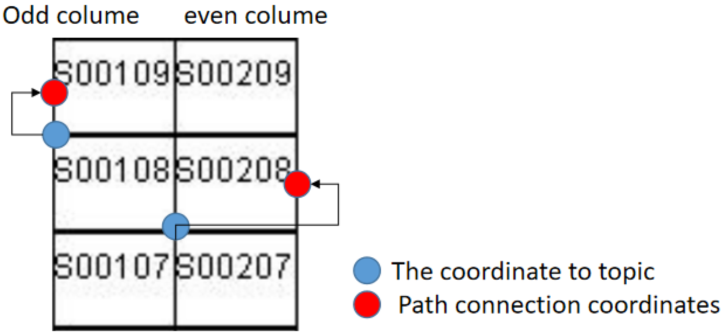

Figure 4. Processing of Odd/even Column Coordinates

Following the process shown in Figure 4, the new coordinates of all cargo compartments can be obtained. Due to the topic request, here is the geometric distance between the coordinates directly, but is similar to "walk straight, turn right", so the use of the Freud algorithm (see Appendix 2), and the goods classification are discussed further, by analyzing the geometrical relationship between the goods, it can be divided into three categories (A point coordinates of (x1, y1), point coordinates (x2, y2) as B):

(1) When the cargo lattice is located in the same row and column (that is, it connects itself), as shown in Figure 5:

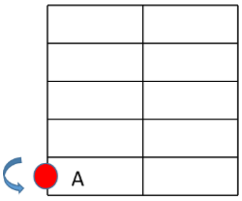

Figure 5. Cargo Case in the Same Row and Column

With Path length:

$$d(i,j) = 0 \tag{1}$$

(2) When the cargo lattice is in the same column and not in the same row, as shown in Figure 6:

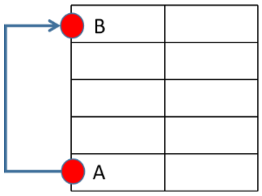

Figure 6. Cargo Rows in the Same Column

With Path length:

$$d(i,j) = 750 \times 2 + |y_1 - y_2| \tag{2}$$

(3) When the goods are not in the same column, as shown in Figure 7:

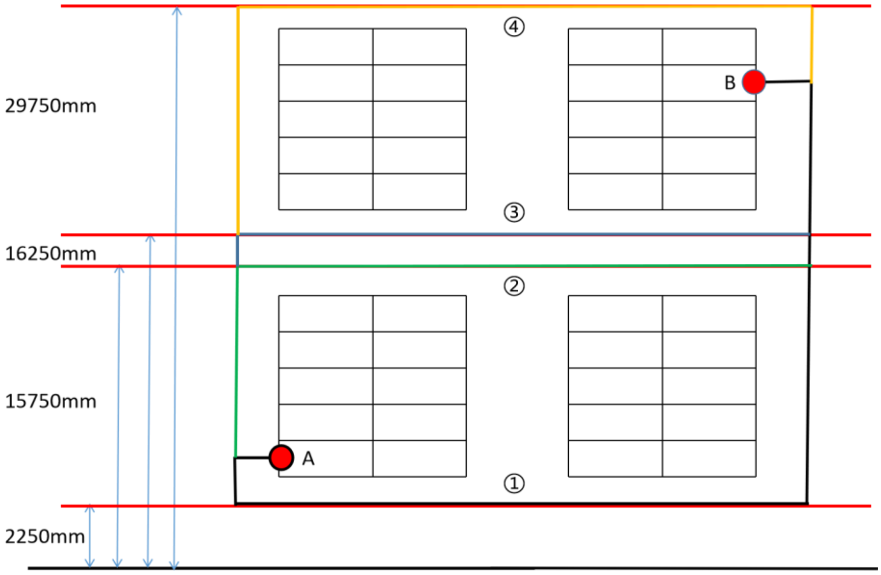

Figure 7. Schematic Diagram of Goods in Different Columns

It can be seen from Figure 7 that there are four paths from A to B in the figure. In this case, there are eight "aisles between shelves" represented by the red line in Figure 7, and their ordinates are 2250mm, 15750mm, 16250mm, 29750mm, 30250mm, 43750mm, 44250mm, and 57750mm respectively.

Taking the aisle between shelves with ordinate of 2250mm as an example, the path length between two shelves meeting any condition ③ is:

$$\mathrm{d(i,j)} = 750 \times 2 + |\mathrm{x}_1 - \mathrm{x}_2| + |2250 - y_1| + |2250 - y_2| \tag{3}$$

Set the ordinate of the aisle between shelves as Y, then:

$$d(i,j) = 750 \times 2 + |x_1 - x_2| + |Y - y_1| + |Y - y_2| \tag{4}$$

It can be known from this that the shortest path d between any two cargo grids meeting condition ③ is:

$$d = \min[750 \times 2 + |x_1 - x_2| + |Y_i - y_1| + |Y_i - y_2|] \quad i = 1,2,\dots,8 \tag{5}$$

It can be seen from the above process that, in programming calculation, only relevant if statements are written to complete the judgment of the relation between cargo cells, and the accurate distance between two cargo cells can be calculated, and then the distance matrix between cargo cells of 3000*3000 can be generated.

The following is to solve the path length between the recheck station and the cargo grid, and between the recheck station and the recheck station:

For the recheck table, its treatment is slightly different from the cargo case. According to the requirements of the topic, as shown in Figure 8, it can be seen that four coordinate points are meeting the requirements on the same check table.

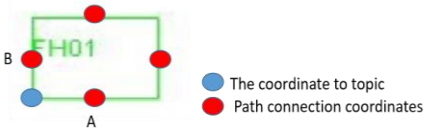

Figure 8. The Check Table Meets the Required Coordinates

Similarly, the coordinates of the check table were transformed by the program first. Similar to the cargo grid that meets the above condition ③, the path distance between the generating coordinates of each check station and the generating coordinates of the cargo grid (completed) should be solved here, and its minimum value should be taken.

If the coordinate of the cargo grid is (x1,y1), the initial coordinate of the check table is (x0,y0), and its four generation coordinates are (x3,y3), (x4,y4), (x5,y5), and (x6,y6), the shortest path is expressed as follows:

$$d = 750 + \min[|x_1 - x_i| + |y_1 - y_i|] \quad i = 3,4,5,6 \tag{6}$$

Simply, the path distance between any two check stations can be expressed as:

$$d = \left|x_i - x_j\right| + \left|x_i - x_j\right| \; i,j = 1,2,\ldots,13 \tag{7}$$

It can be seen from the above process that the above modeling algorithm can be directly implemented to directly generate the corresponding distance matrix during the programming calculation (the distance matrix between the pallet and the check table is 3000*13 and 13*3000, and the distance matrix between the check table is 13*13 symmetric matrix).

The answer can be obtained by outputting all the above distance matrices through an Excel table (see attachment for distance matrix).

5.2 Establishment and Solution of the Model in Question 2

5.2.1 Modeling and Solving Process of Question 2

The second problem is the basic unidirectional TSP problem, so the general linear programming model can be established by LINGO to solve it.

First of all, according to the initial check station and the pick-up grid in the task list, the element serial number is obtained from the first result to facilitate the establishment of a small-scale distance matrix (in this question, the starting check station and 23 cargo grids are known in the single task, so a 24*24 distance matrix should be established). After the distance matrix was obtained, it was imported into the LINGO program, and the modeling process is as follows:

For any goods grid (or check table), establish 0-1 decision variables to determine whether they are connected:

$$x_{ij} = \begin{cases} 0, & \text{If cargo I and cargo J are not connected} \\ 1, & \text{If cargo I and cargo J are connected} \end{cases} \tag{8}$$

Here, if there is only one cargo grid (or check grid) after each cargo grid (or check grid), then:\

$$\sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij} = 1, \quad i = 1, \dots, n \tag{9}$$

Considering that there is only one pallet (or check table) in front of each pallet, then:

$$\sum_{\substack{i=1 \\ i\neq j}}^{n} x_{ij} = 1, \quad j = 1, \dots, n \tag{10}$$

However, the above constraints alone cannot avoid the generation of multiple disconnected loops. Therefore, additional variables were introduced and the following sufficient constraints were added:

$$u_i - u_j + nx_{ij} \leq n - 1, \quad 1 < i \neq j \leq n \tag{11}$$

The objective function is:

$$minZ = \sum_{i,j=1}^{n} d_{ij}x_{ij} \tag{12}$$

The model is as follows:

$$s.t. \begin{cases} minZ = \sum_{i,j=1}^{n} d_{ij}x_{ij} \\ \sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij} = 1, \quad i = 1, \dots, n \\ \sum_{\substack{i=1 \\ i\neq j}}^{n} x_{ij} = 1, \quad j = 1, \dots, n \\ u_i - u_j + nx_{ij} \leq n - 1, \quad 1 < i \neq j \leq n \\ x_{ij} = 0 \; or \; 1， \quad i,j = 1, \dots, n \\ u_i \in R, \quad i = 1, \dots, n \end{cases} \tag{13}$$

Substitute the obtained small-scale matrix into the operation, it can be obtained that:

$$minZ = 406600 \; mm \tag{14}$$

And the terminal grid is S07305 (element number: 1085), and the nearest recheck station is FH07, with a distance of 7700mm. The distance between the terminal pallet and the starting check station is 31800mm, and the final distance is:

$$Z = 382500 \; mm = 382.5m \tag{15}$$

Please see the attachment for the walking route (connection mode).

Time required for picker to travel:

$$t_1 = \frac{Z}{v} = 255s \tag{16}$$

Picking time:

$$\mathrm{t}_2 = 177\mathrm{s} \tag{17}$$

Packing time:

$$\mathrm{t}_3 = 30\mathrm{s} \tag{18}$$

Total time:

$$T = t_1 + t_2 + t_3 = 462s \tag{19}$$

5.3 Establishment and Solution of the Model in Question 3

5.3.1 Modeling and Solving Process of Question 3

In the solving process of problem 3, its modeling programming calculation process is completely consistent with that of problem 2. First of all, according to the initial check station and pickup grid in each task order, the element serial number was obtained from the first result, to establish a small-scale distance matrix, and the corresponding distance matrix of each task order is imported into LINGO program for solving:

In this case, there are five task lists. For each task list, there are two possible starting points. Therefore, a comparison is required.

For any goods grid (or check table), establish 0-1 decision variables to determine whether they are connected:

$$x_{ij} = \begin{cases} 0, & \text{If cargo I and cargo J are not connected} \\ 1, & \text{If cargo I and cargo J are connected} \end{cases} \tag{20}$$

Here, if there is only one cargo grid (or check grid) after each cargo grid (or check grid), then:

$$\sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij} = 1, \quad i = 1, \ldots, n \tag{21}$$

Considering that there is only one pallet (or check table) in front of each pallet, then:

$$\sum_{\substack{i=1 \\ i\neq j}}^{n} x_{ij} = 1, \quad j = 1, \ldots, n \tag{22}$$

However, the above constraints alone cannot avoid the generation of multiple disconnected loops. Therefore, additional variables were introduced and the following sufficient constraints were added:

$$u_i - u_j + nx_{ij} \le n - 1, \quad 1 < i \neq j \le n \tag{23}$$

The objective function is:

$$minZ=\sum_{i,j=1}^{n}d_{ij}x_{ij} \tag{24}$$

The model is as follows:

$$s.t.\begin{cases} minZ=\sum_{i,j=1}^{n}d_{ij}x_{ij} \\ \sum_{\substack{j=1\\ j\neq i}}^{n}x_{ij}=1,\quad i=1,\dots,n \\ \sum_{\substack{i=1\\ i\neq j}}^{n}x_{ij}=1,\quad j=1,\dots,n \\ u_i-u_j+nx_{ij}\leq n-1,\quad 1<i\neq j\leq n \\ x_{ij}=0\ or1,\ \ i,j=1,\dots,n \\ u_i\in R,\quad i=1,\dots,n \end{cases} \tag{25}$$

Repeat the above process, and a total of 10 groups of distance data can be obtained. The two cargo grids connected to the starting check table may be the terminal cargo grids in the TSP loop. The following is a calculation method of terminal lattice:

Here, the calculated distance for each order is Z1 based on FH03 and Z2 based on FH11. When the starting check station is FH03, the distance between the end check station and the starting check station is D1, and the distance between the end check station and another check platform station (FH11) is D2; When the starting check station is FH11, suppose that the distance between the end check station and the starting check station is D3, and the distance between the end check station and another check station (FH03) is D4, then the shortest path Z of each order is:

$$Z=min\{Z_1,Z_2,(Z_1-d_1+d_2),(Z_2-d_3+d_4)\} \tag{26}$$

Theoretically, there are four terminal compartments, so formula (26) needs to be calculated four times to get the minimum total distance of an order.

For five orders from Order 2 to Order 6, the starting point recheck station, endpoint recheck station, path distance, and required time (including packaging time) are shown in Table 1:

Table 1. Order Information

The order number	Starting point check station	Terminal check station	Path distance divided by m	Total time $\mathrm{It_i}/s$
2	FH11	FH11	445.4	508.9333
3	FH11	FH03	390.14	461.0933
4	FH03	FH11	434.3	505.5333
5	FH03	FH03	363	437
6	FH03	FH11	447.1	496.0667

The order of the picker is as follows (see the attachment for the specific execution path): Order 4 (FH03) → (FH11) Order 2 → (FH11) Order 3 → (FH03) Order 5 → (FH03) Order 6 → (FH11)

The time required for the picker to complete all orders is:

$$T = \left(\sum_{i=1}^{5} t_i \right) - 4 \times 30 = 2288.6s \tag{27}$$

5.4 Establishment and Solution of the Model in Question 4

5.4.1 Modeling and Solving Process of Question 4

It can be seen that in the solving process of question 4, its modeling programming calculation process is completely consistent with that of Question 3. First of all, according to the initial check station and pickup grid in each task order, the element serial number is obtained from the first result, to establish a small-scale distance matrix, and the corresponding distance matrix of each task order is imported into LINGO program for solving:

In this case, there are 49 task orders in total. For each task order, there are four possible starting points. Therefore, a comparison is required.

For any goods grid (or check table), establish 0-1 decision variables to determine whether they are connected:

For any goods grid (or check table), establish 0-1 decision variables to determine whether they are connected:

$$x_{ij} = \begin{cases} 0, & \text{If cargo I and cargo J are not connected} \\ 1, & \text{If cargo I and cargo J are connected} \end{cases} \tag{28}$$

Here, if there is only one cargo grid (or check grid) after each cargo grid (or check grid), then:

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1, \quad i = 1, \dots, n \tag{29}$$

Considering that there is only one pallet (or check table) in front of each pallet, then:

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} x_{ij} = 1, \quad j = 1, \dots, n \tag{30}$$

However, the above constraints alone cannot avoid the generation of multiple disconnected loops. Therefore, additional variables were introduced and the following sufficient constraints were added:

$$u_i - u_j + n x_{ij} \leq n - 1, \quad 1 < i \neq j \leq n \tag{31}$$

The objective function is:

$$minZ = \sum_{i,j=1}^{n} d_{ij} x_{ij} \tag{32}$$

The model is as follows:

$$\text{s.t.}\begin{cases} \min Z = \sum_{i,j=1}^{n} d_{ij}x_{ij} \\ \sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij} = 1, \quad i = 1,\dots,n \\ \sum_{\substack{i=1 \\ i\neq j}}^{n} x_{ij} = 1, \quad j = 1,\dots,n \\ u_i - u_j + nx_{ij} \le n-1, \quad 1 < i \neq j \le n \\ x_{ij} = 0 \text{ or } 1 \ , \ i,j = 1,\dots,n \\ u_i \in R, \quad i = 1,\dots,n \end{cases} \tag{33}$$

A total of 196 groups of distance data can be obtained by repeating the above process. The two cargo grids connected to the starting check table may be the terminal cargo grids in the TSP looped. The following is a calculation method of terminal lattice:

Here, the calculated distance for each order is Z1 based on FH01, Z2 based on FH03, Z3 based on FH10, and Z4 based on FH10. When the starting check station is FH01, the distance between the terminal check station and the starting check station is d1, and the distance between the terminal check station and the other three check stations is D2, D3, d4. Each recheck station shall perform the above process once, and the path D of the comparison process is:

$$Z = \min\{Z_1, (Z_1 - d_1 + d_2), (Z_1 - d_1 + d_3), (Z_1 - d_1 + d_4)\} \tag{34}$$

Theoretically, there are four terminal grids, so formula (21) needs to be calculated 16 times to get the minimum total distance Z of an order.

At this time, the LINGO program is used to output the order's access order and input the starting point and endpoint of the order into Excel.

According to Formula 16~19, for a task order, the total execution time T is:

$$T = t_1 + t_2 + t_3 \tag{35}$$

Queuing theory: if the total delivery time needs to be reduced, then short time orders need to to be given priority.

Use an Excel spreadsheet to sort the time. After sorting, all orders are divided into 9 pieces according to the min-max principle, and the starting and ending points of orders are fully considered.

Due to the operation speed of the LINGO program, this problem cannot calculate all the results efficiently.

6. Evaluation of the Model

6.1 This Topic Applies TSP Model Evaluation

6.1.1 Advantages

LINGO program operation is effective, compared with the ant colony algorithm dot-matrix direct connection method, the algorithm can complete the broken line path according to the requirements, and can directly output the path connection method and the total distance.

6.1.2 Shortcomings

The LINGO program itself cannot calculate large matrices, which leads to the completion of the modeling of question 4, but there are certain problems in the programming operation, and the operation process cannot be completed within the specified time.

The results obtained by the TSP program are a loop closed-loop, which does not apply to models with different starting points and ending points. Therefore, further operation processing is required after obtaining LINGO operation results.

6.1.3 Attempt Optimization

In the second question, we try to use the ant colony algorithm. However, after the test of the small matrix, it is found that the program is difficult to solve the problem of the "walk straight line, turn right Angle" method, so in the subsequent programming implementation, the LINGO genetic algorithm was still used to solve.

References

Liu Wenjun, Gao Wei, & Deng Senyuan. (2019). *Journal of hanshan normal university, 40*(03), 17-24+108.

Wang Haiying, Huang Qiang, Li Chuantao, et al. (2010). *Graph theory algorithm, and MATLAB implementation*. Beijing University of Aeronautics and Astronautics Press.

Xiao Huayong. (Eds.). (2014). *Practical Mathematical Modeling and Software Application*. Xi'an: Northwestern Polytechnical University Press. ISBMN 978-7-5612-4144-8