An Optimization Algorithm for TSP-like Problem in Warehouse Picking

Wan Quan[1], Suhui Ge[2] and Shulian Zhang[1]

[1] College of Information Engineering, Qingdao Institute of Technology, Qingdao 202204, Shandong, China

[2] Qingdao Engineering Vocational College, Qingdao Institute of Technology, Qingdao 202204, Shandong, China

Email: wanlibaidu@foxmail.com; suhuige@126.com; 958007736@qq.com

Abstract

In order to solve the problem of the order picking optimization in warehouse line, this study puts forward the concept of class TSP (Travelling Salesman Problem) based on a model using graph theory knowledge modelling. With the aid of the thought of classical TSP, it puts forward three principles, based on a fixed starting point (not repeatable) integer programming model; The solution of the model is also based on a simple optimization algorithm (vertex translocation method) of the classical TSP, and the appropriate modification is made. For the problem of unfixed starting point, the global traversal algorithm is added. Through the reconstruction of the line sequence, a TSP-like problem is successfully transformed into a TSP, and a model is finally established, which is a prominent innovation point of this paper. The calculation method of adjacency matrix W is given in the postscript.

Keywords. TSP-like; integer programming; vertex translocation method.

1. Introduction

The problem stems from the tenth MathorCup National Mathematical Modeling Challenge C problem, a warehouse with x shelves, each shelf with y shelves, each shelf with Z shelves, the existing w review table, now the picker sets out from one of the check stations, takes the goods from different prices, and returns to the nearest check station. Q: How to arrange the order of picking so that the total journey of the pickers is the shortest. With 2 horizontal rows of shelves, each row has 7 shelves, each shelf has 215 shelves, 10 check for example, see figure 1 warehouse layout, of which the left side of five and the following five check for the table.

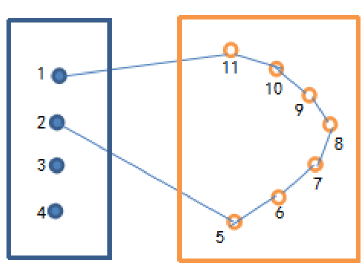

Figure 1. Warehouse shelf layout.

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 125060K

doi: 10.1117/12.2661781

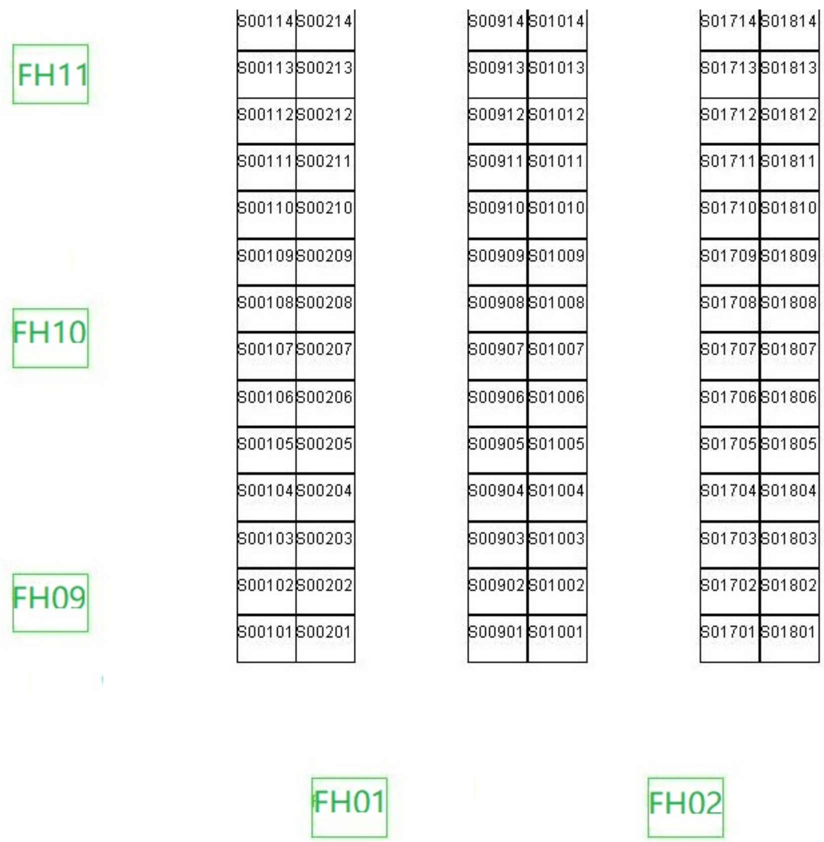

Figure 2. 4+7 model diagram.

Let G be a connected weighted undirected graph with $M + N$ vertices, where M vertices are the fixed starting and ending points, and n vertices are the process vertices, we start from a point in M, pass N vertices exactly once, and finally get back to the shortest distance from any of the M vertices. Taking 4 + 7 vertices as an example, as shown in figure 2, we start from Point 1, and finally return to a line at Point 2.

In the case of $M + N$ vertices, W_{ij} is used to represent the weight of any two vertices, which can refer to the distance between two cities. From this we can construct the adjacency matrix W as follows.

$$W = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1(m+n)} \\ W_{21} & W_{22} & \cdots & W_{2(m+n)} \\ \cdots & \cdots & \cdots & \cdots \\ W_{(m+n)1} & W_{(m+n)2} & \cdots & W_{(m+n)(m+n)} \end{bmatrix} \quad (1)$$

From the undirected graph, it can be seen that the W matrix is a symmetric matrix, and the weight on the diagonal is 0, which means the following relationship holds.

$$W_{ij} = W_{ji} (i = 1, 2, \ldots, m+n) \quad (2)$$

$$W_{ii} = 0 \ (i = 1, 2, \ldots, m+n) \quad (3)$$

2. Problem Analysis

It is easy to see from figures 1 and 2 that if the starting point and the ending point are the same, it is a classic TSP (travel agent) problem, but now the starting point and the ending point may or may not be the same. This paper calls it a TSP-like problem, and it is obvious that the TSP-like problem is a special case of the TSP-like problem [1].

In order to grasp the essence of the problem more intuitively, a typical case is studied, which is explained as follows:

First, the starting point is fixed, so let's say we start at point 1.

Second, the endpoint is fixed, and vertex Number 2 is assumed to be the endpoint.

Third, it goes through n points exactly once.

Fourth, the best route and the shortest journey are found.

The idea of solving this new problem is to transform it into a classical TSP problem, for which the following three transformation rules are made:

Rule 1, after we start at vertex number one, we go to vertex number two.

Rule 2, we start at vertex number two and choose the right path, go through n vertices and return to vertex number one.

Rule 3, the injunction $W_{12}=W_{21}=0$.

With the above three rules, the original problem is equivalent to: starting from vertex No. 1, passing through vertex No. 2, and preferably returning to the TSP problem at No. 1. At this point, the adjacency matrix is W_2 (indicating the final return of No. 2).

$$W2 = \begin{bmatrix} 0 & 0 & \cdots & W_{1(2+n)} \\ 0 & 0 & \cdots & W_{2(2+n)} \\ \cdots & \cdots & \cdots & \cdots \\ W_{(2+n)1} & W_{(2+n)2} & \cdots & 0 \end{bmatrix} \tag{4}$$

3. Establishment of TSP Problem Model

The 0/1 variable X_{ij} is defined as the decision variable of the model, indicating whether the ith vertex in an effective line is directly connected to the JTH vertex, which is 1, and otherwise is 0. Based on the above analysis, a TSP-like linear programming model is established [2].

Objective function:

$$\min \sum_{j=1}^{2+n} \sum_{i=1}^{2+n} X_{ij} W_{ij} \tag{5}$$

$$S.T. \begin{cases} \sum_{i=1}^{2+n} X_{ij} = 2(j = 1, 2, 3, \ldots, 2+n) & (6) \\ w_{12} = w_{21} = 0 & (7) \\ X_{12} = X_{21} = 1 & (8) \end{cases}$$

It is necessary to explain the meaning of constraint condition equation (6). Since TSP problem forms a circle, any node must be connected to two edges, so there are only two edges connected to a point. The starting point, the end point, plus n process points, a total of *n*+2 points. Equations (7) and (8) are based on rules 1 and 3, respectively.

4. Model Solution

This problem is a typical integer programming problem of linear programming, and belongs to the transportation problem in linear programming, about which there are many ways of solving the problem, branch and bound, simulated annealing, particle swarm optimization (pso) algorithm and so on, also have a function in Matlab to solve the integer programming problem, see equation (9), there are eight parameters, which can check the documentation that came with the Matlab manual.

$$X = intlinprog\left(c,\ intcon,\ A,\ b,\ A1,\ b1,\ L,\ U\right) \tag{9}$$

The problem has been basically solved, but this problem is a classical problem in graph theory [3]. From the point of graph theory, this paper presents a concise algorithm.

4.1. The Optimization Algorithm of Classical TSP Problem

In order to illustrate the idea of the TSP-like problem algorithm, let's look at the solution to the classic TSP problem.

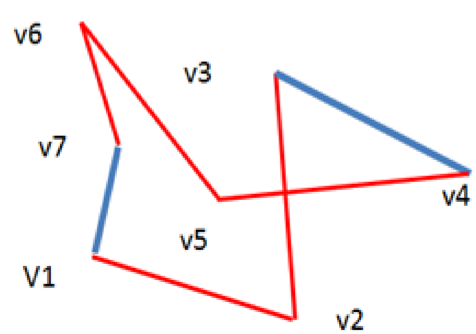

Figure 3. Initial line starting from Point V1.

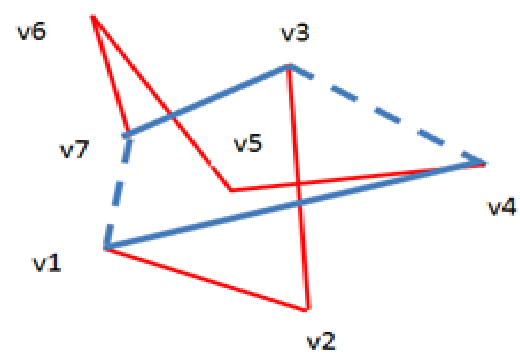

Figure 4. Transform the line from point V1.

Figure 3 is an initial line of any TSP problem that accesses each vertex in sequence from point V1 and returns to point V1 (v1->v2->v3->v4->v5->v6->v7->v1). Obviously, the optimal route cannot be guaranteed. For this reason, (v1, V7) and (v3, v4) edges are replaced by (v3, V7) and (v1, v4), as shown in figure 4. The result is still a loop through each vertex (v1->v2->v3->v7->v6->v5->v4->v1). If the total distance of the new circuit is less than that of the original circuit, then replace the original circuit with the new circuit, so as to continuously improve the circuit, and finally choose an optimal circuit, the classic TSP problem is solved.

4.2. An Optimization Algorithm for TSP-like Problems

For the integer programming model corresponding to equations (5)-(8), the following algorithm is obtained under the constraint of the three rules by borrowing the algorithm idea in Section 4.1. This is further illustrated visually through a flow chart [4].

The first step is to enter the adjacency matrix W2 (endpoint 2).

The second step is to set up the initial solution vector L1: [V1, V2, V3... V(2+n), V1], start from V1 and return to the closed loop of V1, and calculate the total travel S1.

The third step is to replace (vi, vi+1) and (vj, vj+1) with (vi, vj) and (vi+1, vj+1), and calculate the sum of the two sides before and after the conversion. The Si (Among them i = 2, 3, ..., n. j = 4, 5, ..., n+2; vn+3=v1) is updated by a smaller solution of the sum of the lengths

The fourth step is to output the final solution vector and the value of Sn.

So far, the TSP-like problem with a fixed starting point has been solved. For the TSP-like problem with a fixed starting point (the starting point and the ending point can be any of M points, allowing the same starting point), only the starting point needs to be traversed, for specific reference to the following algorithm flow [5][6].

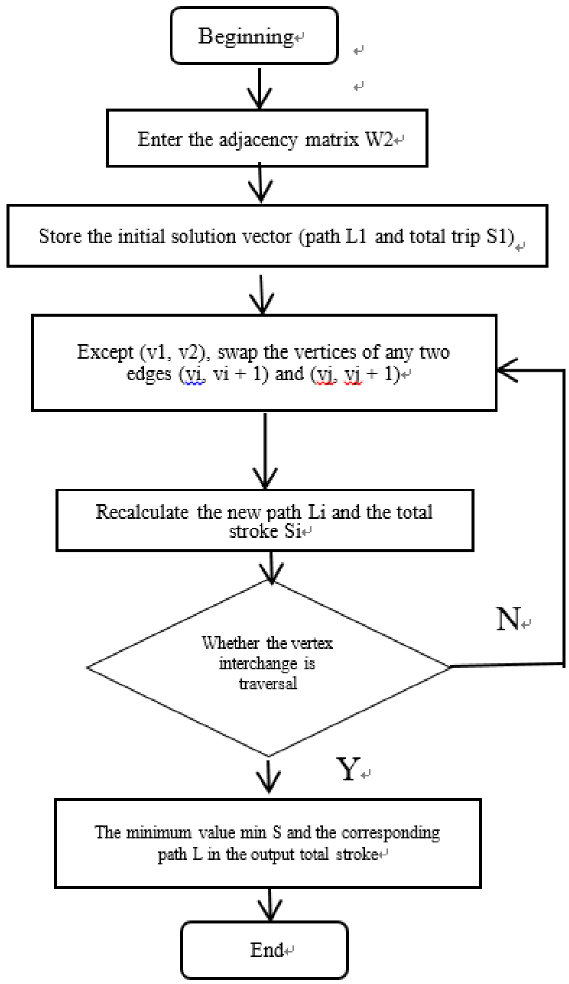

Figure 5. Algorithm flow chart of TSP problem with fixed starting point.

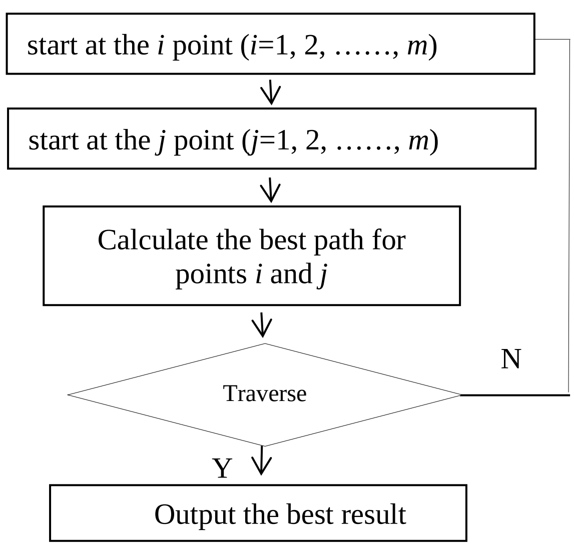

Figure 6. Algorithm flow chart of TSP problem with non-fixed starting point.

After the analysis of such TSP problems, matlab can be used to write the corresponding program to solve.

5. Complexity Analysis of the Algorithm

Here refers to the time complexity of the algorithm, which is the time measurement of the algorithm, denoted as: $T(n)$=O($f(n)$). It means that with the increase of problem size N, the growth rate of algorithm execution time is the same as that of $F(n)$, which is called asymptotic time complexity of algorithm, or time complexity for short [7][8].

The size function $F(n)$ of this problem comes from figures 5 and 6, denoted as $f1$ and $f2$, respectively. The following values of $f1$ and $f2$ are calculated respectively.

$$f1 = (n-1) \times (n-2+1) = (n-1)^2 \tag{10}$$

$$f2 = m \times m \times f1 = m^2 (n-1)^2 \tag{11}$$

In order to use big O notation, equation (11) can be further expanded into the following expression:

$$\begin{aligned} f2 &= m^2 \times (n^2 - 2n + 1) \\ &= (mn)^2 - 2m^2 n + m^2 \end{aligned} \tag{12}$$

Thus, the time complexity of the algorithm can be obtained:

$$T = O\left((mn)^2\right) \tag{13}$$

It can be seen that this algorithm has no advantage in execution efficiency and is even relatively clumsy. The advantage is that it has clear thinking and is relatively easy to understand [9, 10].

6. Conclusion

The model can be applied to many aspects of life, the classical application is mainly embodied in the aspects of warehouse picking, supermarket shopping and tour route arrangement, etc. , many times it is not an easy thing, especially in the warehouse picking problem, the number of goods is relatively large, the establishment of the adjacency matrix is very large, MATLAB calculation efficiency is very low, python is recommended for data processing.

Acknowledgment

Authors would like to appreciate their research team for the approved fund project, Shandong province undergraduate university teaching reform research key project (2018X072), and Chairman of Qingdao institute of technology (2020JY009). Their team offers them the confidence and give them useful literature knowledge and information in this paper.

References

[1] Jedrzejowicz P and Wierzbowska I 2020 Parallelized swarm intelligence approach for solving TSP and JSSP Problems *Algorithms* **13** (6) 142.

[2] Wu Q, Chen Z, Wang L, Lin H, Jiang Z, Li S and Chen D 2019 Real-time dynamic path planning of mobile robots: A novel hybrid heuristic optimization algorithm *Sensors* **20** (1).

[3] Shamsaldin A S, Rashid T A, Al-Rashid A R A, Al-Salihi N K and Mokhtar M 2019 Donkey and smuggler optimization algorithm: A collaborative working approach to path finding *Journal of Computational Design and Engineering* **6** (4).

[4] Yousri D, Allam D and Eteiba M B 2018 Chaotic whale optimizer variants for parameters estimation of the chaotic behavior in Permanent Magnet Synchronous Motor *Applied Soft Computing Journal* **74** 479-503.

[5] Ghahremani-Nahr J, Kian R and Sabet E 2018 A robust fuzzy mathematical programming model for the closed-loop supply chain network design and a whale optimization solution algorithm *Expert Systems with Applications* **116** 454-471.
[6] Jain M, Singh V and Rani A 2018 A novel nature-inspired algorithm for optimization: Squirrel search algorithm *Swarm and Evolutionary Computation* **44** 148-175.
[7] Gálvez J, Cuevas E and Avalos O 2017 Flower pollination algorithm for multimodal optimization *International Journal of Computational Intelligence Systems* **10** (1).
[8] Mostafa A, Hassanien A E, Houseni M and Hefny H 2017 Liver segmentation in MRI images based on whale optimization algorithm *Multimedia Tools and Applications* (23).
[9] A computationally efficient Branch-and-Bound algorithm for the permutation flow-shop scheduling problem[J].Jan Gmys, Mohand Mezmaz,Nouredine Melab,Daniel Tuyttens. European Journal of Operational Research . 2020 (3).
[10] Planning for tourism routes using social networks[J] . Isabel Cenamor,Tomás de la Rosa,Sergio Nú?ez,Daniel Borrajo. Expert Systems With Applications . 2017.