An automated storage and retrieval system optimization with MILP methods

Yixin Chen [1, †], Boning Fan [2, †], Jiaming Li [3, *, †], Jianfeng Lin [4, †] and Mengmeng Lin [5, †]
[1]School of Automotive and Transportation Engineering, Hefei University of Technology, Hefei, Anhui 230009, China;
[2]Xuteli School, Beijing Institute of Technology, Beijing, 100081, China;
[3]School of Automation, Southeast University, Nanjing, Jiangsu 211100, China;
[4]School of Power and Mechanical Engineering, Wuhan University, Wuhan 430072, China;
[5]College of Engineering, Northeast Agricultural University, Harbin, Heilongjiang 150030, China.
[*]Corresponding author email: 213191258@seu.edu.cn
[†]These authors contributed equally.

Abstract

In this paper, we consider the problem of scheduling an automatic guided vehicle (AGV) in an automated storage and retrieval system (AS/RS). In this system, the AGV acts as a transfer station and performs the task of depositing and taking out goods. It is assumed that the number and location of shelves are known, as well as the number and type of goods on each shelf. The AGV performs transport tasks with decisions that include finding the optimal transport sequence between different shelves and the optimal sequence of goods within each shelf. Therefore, these decisions jointly minimize the total transport distance. Using mixed-integer linear programming (MILP), the AGV can not only complete the task with the shortest working route but also return to the sorting station with the shortest path after completing the task. Using MATLAB software to simulate the two cases, the results of obtaining the optimal solution verified the correctness of the algorithm. The results show that the algorithm has high performance for optimizing AS/RSs.

Keywords: automated storage,retrieval system

1. Introduction

Nowadays, with the need for selection and storage rising day by day, robots are required to behave well in a practical assignment like [1]. We will import some successes in systems like line tracking with a sliding-mode control method in [2] and path routing with distribution planning, and Cargo flows modelling in [3] to make an effective retrieval system. In a problem to minimize the moving distance of an automatic guided vehicle (AGV) when completing products from shelves, we decide to further applications of the AGV problem by referring to [4] for the construction of the database and [5] for the task scheduling to make an optimal route.

To better solve the Automated Storage and Retrieval System (AS/RS), we prefer mixed-integer linear programming (MILP), which represents an effective mathematical modelling approach to solve complex optimization tasks by identifying the potential trade-offs between conflicting solutions objectives. In MILP program, we will encounter a vehicle routing problem (VRP), a combinatorial optimization, and an integer programming problem, which depends on the modules we choose to solve. Besides, AS should include battery problems to plan for the best use of left power. And the transport efficiency in transferring goods should also be considered well.

This paper aims to present a solution towards erecting a MILP to Linear Programming and finding an easier way to define the parameter of indicating moving tacks for MILP method to get the best selection and optimal path. [6] points out a way to use an optimization approach for sizing port rail networks and planning railway shunting operations, size the maximum number of trains. However, [6] did not give the optimal solution for the minimum cost. In addition, [7] uses the conventional VRP to calculate the route program, which can report the minimum route after solutions in the training part. Moreover, in the UAV path planning, [8] noted a way to form routes by detecting a set of targets, but detecting targets takes a tremendous computing capability. To tackle the potential problems of excessive waste of calculation, we'll refer to [9] to construct the distributed database. The robot will proceed to a designated principle of shelves of goods in a timely manner. This will allow a robot to consume the lowest energy for the best efficiency. And to ensure the robotic safe works, we will import the Rapid Visible Tree by referring [10] to decide the emergent situation in noisy points from the outside

Fifth International Conference on Computer Information Science and Artificial Intelligence (CISAI 2022)
edited by Yuanchang Zhong, Proc. of SPIE Vol. 12566, 125662V · © 2023 SPIE
0277-786X · doi: 10.1117/12.2667780

circumstance. Besides, it is crucial to ensure the correct match targets by separate target and decoy database searches. [11] point out a way of individual target and decoy database searches in estimated errors. According to [12], we can easily drive the system against the interruption, like noise or being exposed to the environment by using Bacterial Potential Field (BPF) for path planning in MRs. In the following, we present the mechanic structure and the mathematical module. Finally, we summarize the paper with experiments highlighting the overall system performance.

2. Problem Formulation

2.1 Problem description and definition

An automated storage and retrieval system consists of multiple shelves and multiple sorting stations. Set $S \in \{s_1, s_2 \dots s_m\}$ consists of all the shelves. Set $Sort \in \{sort_1, sort_2 \dots sort_m\}$ consists of all the sorting stations. Set $P \in \{p_1, p_2 \dots p_l\}$ consists of all types of products. The distance between s_i and s_j $(i \neq j)$ or s_i and $sort_j$ are known constants. The storage capacity cap_i for s_i is fixed, and we assume that cap_i is a constant. We also assume that the quantity of product i on shelf j, which is called O_{ij}, is known. When the product in-out task arrives, our planning system needs to determine the next action for each product. For example, p_k is to be transported from $sort_i$ to s_j and p_{k+1} is to be transported from s_j to $sort_i$. Shipping a set of products from $sort_i$ to s_j and then from s_j to $sort_i$ is a transportation task completed by an automatic guided vehicle (AGV). This problem aims to minimize the movement distance D of the AGV while completing the product in-out task. The constraints include: First, the in-out tasks must be perfectly completed. Second, the number of products on s_i shall not exceed cap_i. Third, the number of products taken from the shelf must not exceed its inventory level.

2.2 Problem analysis and simplification

Firstly, there are generally multiple sorting stations and multiple shelves for an automated storage system. The sorting station is usually used to temporarily store goods during the in-out tasks, as a transfer station, not for storing goods, and the shelves are used as the final actual storage for the goods. Therefore, after completing each input and output task, $sort_1, sort_2 \dots sort_m$ should not possess any goods, and the goods stored in the warehouse should only be held on $s_1, s_2 \dots s_n$.

An automated storage and retrieval system with multiple sorting stations and multiple shelves will complicate the problem, and the income is too low. The multiple sorting station management does not conform to the actual situation. In a real-time case, an enormous warehouse is usually divided into areas and divisions. It is impossible to place $sort_1, sort_2 \dots sort_m$ on the same level. The system is also combined with the actual economic benefits. So generally, a sorting station should be in charge of the in-out task of multiple shelves. Based on the above real situation, we suppose that the storage system has been divided into several levels. On the top level is a superior sorting station which distributes tasks to the sorting stations on the secondary level. The sorting stations on the secondary level can continue to have jurisdiction over the sorting stations on the lower level. According to this method, a perfect storage and retrieval system can be divided into multiple levels. The levels finally form a tree structure. In this case, only the sorting station on the bottom level can directly manage the in-out tasks of the shelves. At the same time, the upper stations can directly distribute tasks to the shelves. Each sorting station follows the same principle. Each station in a lower level can be regarded by the upper-level station as a shelf. So the principle of managing the task is the same as the principle for the bottom level station. If we solve the principle for the sorting station on the bottom level, we can generalize it to the whole system.

For example, in figure 1, a storage system is divided into three levels: shelves 1,2 and 3 directly belong to sorting station 1.1.1, which is the lowest level; sorting stations 1.1.1 and 1.1.2 are directly subordinate to sorting station 1.1, which is the second level; finally sorting station 1.1, 1.2 and 1.3 are directly subordinate to sorting station 1, which is the highest level. The parent sorting station at each level can treat its substation as a shelf regardless of the specific structure contained. Each level focuses only on the members of that level without the upper or lower ones. When product input (product output is the inverse process of product input, the principle is the same), from the total sorting station 1 distribution products to the sub sorting workstation 1.1, 1.2, 1.3. Each sub-sorting station further distributes products to the next level sub-stations. The lowest level of sorting station 1.1.1 will directly face the shelves and directly assign products to each shelf. Therefore, according to this rule, a complex storage system containing multiple sorting stations and shelves can be decomposed into various levels. Each level can be seen as a simple subsystem. After all this simplification, the large and complex system has been decomposed into the simplest subsystem. The subsystem discussed here ultimately only contains one sorting station $sort_1$ and shelves $s_1, s_2 \dots s_n$.

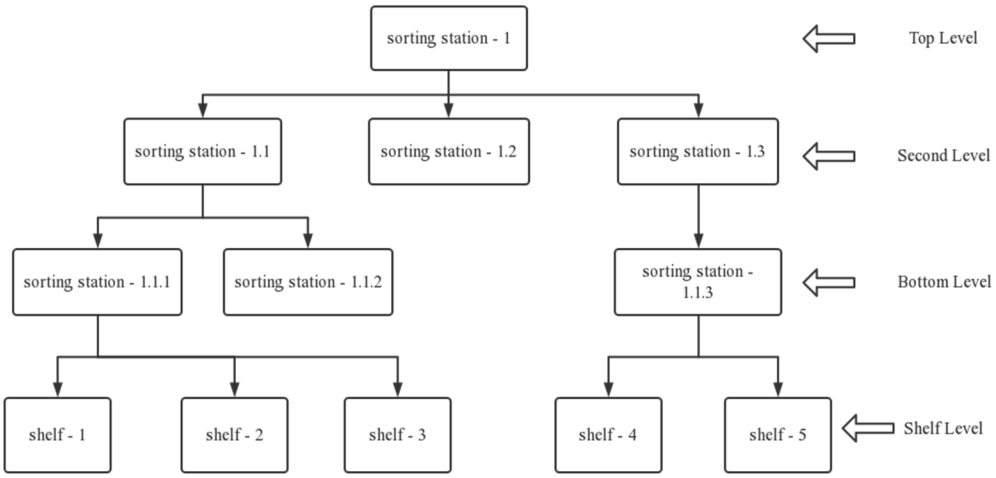

Figure 1. Tree form of the system.

2.3 Model's methodology

Firstly, we can collect the data on the shortest distance from the shelves $S \in \{s_1, s_2 \dots s_n\}$ to its subordinated sorting station $sort$, which is already known by technical measurements. The shortest distance between the shelves is also known. $sort$ always has the current number of each type of goods on each shelf. The shelves have multiple cameras to monitor the current amount of the goods and timely update the information to the station. A wide variety of goods can be stored on each shelf. The storage area for each cargo on s_k ($k \in \{1,2..n\}$) is fixed. That means different types of cargo cannot share their storage space. The model is also effective when different types of cargo share their storage space. The problem-solving procedure and methods are the same. The additional part is to set a different amount of remaining storage space on each shelf when entering the problem's parameters. Only one kind of cargo is accessed for each task, the accessing to other types of goods will be started only after completion of the last accessing. The same goods are either deposited or taken out. Taking and saving goods at the same time is not meaningful. Each cargo access task in this minimum area handles the deposit and removal process separately. The rule is to take out first and then deposit. This rule has no impact on the partition storage of goods on the shelf. However, if the goods share the storage space, taking out the goods first can improve the efficiency of the following storage process. While depositing, we consider that all goods have been placed in $sort$ in the initial state. The only AGV in the area is ready at the $sort$ waiting for command. Since only one cargo is deposited at a time, an AGV carries only one cargo at a time, the load of an AGV is set as limited or unlimited. A limited situation means that the AGV pickup or deposits the cargo on only one shelf at each time. An unlimited situation implies that the AGV is always large enough to carry all goods transported in the area at one time. Here we only consider the limited case, which means the AGV can only pick or deposit goods to or from one shelf at a time. Then it will return to the sorting station in the shortest path immediately after exchanging goods with the shelf.

Each time the AGV takes $sort$ as the starting point. When the AGV receives the task to take out a kind of goods from S, it will check out the quantity of these goods on S to narrow down the searching area. The shelf, which has the shortest distance from the sorting station, will be reached first. However, it is not always the most efficient way to take out goods from the nearest shelf. There may be only a tiny amount of desirable goods on the nearest shelf, while another slightly farther shelf has plenty of desirable goods. The problem will be explained and solved later. The AGV will keep on searching until the sum of such goods available on each shelf meets the total amount of goods to be taken out. After that, the AGV will follow the shortest path to the selected shelves to collect the desirable goods. After finishing the taking-out procedure, it will directly return to $sort$ to unload the goods and then continue to take out goods from the next shelf, which has been planned before. On the contrary, when goods are waiting to be stored on S, the procedure is consistent with taking out. The AGV will check the available space for the goods on all the shelves to narrow the searching area. The shelf, which has the shortest distance from the sorting station, will be reached first. The AGV will keep on searching until the shelves total remaining inventory space enough to store all the goods at the sorting station, and then follow the same way to put goods into each shelf.

3. Model Formulation

3.1 Model constraints

There are several constraints to this problem. First, there is only one sorting station in this area, and the shortest distance l_i between s_i and $sort_1$ is known. The quantity of goods j on s_i are always known as O_{ij}. Second, each shelf can store a variety of goods. Different kinds of goods can be stored in different zones or mixed. Third, each task only accesses one type of goods. There will be only one task, either depositing or withdrawing. Only then will the task for another kind of goods begin. The strategy will generally be taking out first and then depositing. Fourth, if the goods are to be stored at the initial state, all the goods to be stored have been placed in the sorting station, and the AGV starts at $sort_1$. Fifth, there is only one AGV in this area, and each delivery only reaches one shelf. The car only travels between $sort_1$ and one shelf, and the single cargo capacity of the car is always greater than the capacity of a single shelf. Sixth, the AGV first looks through S to find s_k ($k \in \{1,2..n\}$) with the available place

Table 1. Variables explanation.

M	The number of the shelves.
N	The quantity of the goods to be taken out or deposited.
i	The mark of each shelf from 1 to n
j	The mark of each kind of goods from 1 to m
l_i	The shortest distance from each shelf to the sorting station
X_i	A binary variable which determines that whether the shelf i is the target for the AGV
l_{ij}	The available place on shelf i for goods j
O_{ij}	The quantity of goods j on shelf i

(when depositing) or available goods taken by the car (when withdrawing). Then the AGV uses the shortest path algorithm to find the shortest distance to s_k. Then the AGV arranges the route based on the distance data or available place of s_k. The AGV will choose the shortest route of s_k, and then it will start from $sort_1$ and follow the shortest path to exchange goods with s_k. Then the AGV will immediately return to $sort_1$ in the same shortest path. Seventh, in each case, the AGV may need to go to multiple shelves for exchanging to complete the total storage or depositing task. Note that the variables used in our model will be shown in the table 1.

3.2 Model definition

Here we define the objective function and the constrains. The objective function is to minimize the total distance covered by the AGV. As we mentioned in the operation mode part, the AGV will go back straightly to the sorting station to unload the goods, so the distance should be doubled each time. The objective function is:

$$min\ 2\times\sum_{i=1}^{M} X_i l_i\ . \tag{1}$$

The constraints include:

$$\sum_{i=1}^{M} X_i O_{ij} \geq N\ , \tag{2}$$

$$\sum_{i=1}^{N} X_i l_{ij} \geq N\ . \tag{3}$$

Equation (2) stands for all the shelves in S have more goods than the quantity of the goods to be taken out. Equation (3) stands for all the shelves in S have more available place for the quantity of the goods to be deposited.

4. Method and Algorithm

The situation is now being simplified as a MILP. Here we use interior-point method to solve the problem. Interior-point method is a type of algorithm used to solve both linear and nonlinear convex optimization problems containing inequalities as constraints. The LP (Linear-programming) Interior-Point method relies on having a linear programming model with objective function and all constraints being continuous and twice continuously differentiable. A problem is assumed to be strictly feasible and will have a dual optimal that will satisfy Karush-Kuhn-Tucker (KKT) constraint. The problem is solved (assuming there IS a solution) either by iteratively solving for KKT conditions or to the original problem with equality instead of inequality constraints, and then applying Newton's method to these conditions.

Interior-point methods came about from a desire for algorithms with better theoretical bases than the simplex method. While the two strategies are similar in a few ways, the interior point methods involve relatively expensive (in terms of computing) iterations that quickly close in on a solution, while the simplex method involved usually requires many more inexpensive iterations. From a geometric standpoint, interior point methods approach a solution from the interior or exterior of the feasible region but are never on the boundary.

There are two important interior point algorithms: the barrier method and the primal-dual IP method. The primal-dual method is usually preferred due to its efficiency and accuracy. Major differences between the two methods are as follows. There is only one loop/iteration in primal-dual because there is no distinction between outer and inner iterations as with the barrier method. In primal-dual, the primal and dual iterates do not have to be feasible.

For our problem, the barrier method is enough for us to solve the problem because there is only one loop in the method, which can be easily applied to a MILP. The barrier method's procedure is illustrated as: for the problem:

$$\begin{aligned} &min\ \mathrm{f(x)} \\ &\text{s.t. } g_i(x) \le 0, j = 1,2,3...,m \end{aligned}. \tag{4}$$

We define a barrier function either in two different ways:

$$Q(x,r) = f(x) + r\sum_{i=1}^{m}(g_i(x))^{-1} \tag{5}$$

or

$$Q(x,r) = f(x) - r\sum_{i=1}^{m}\ln(-g_i(x)). \tag{6}$$

Then derivate the function $Q(x,r)$ with respect to $x_1, x_2 \ldots x_n$:

$$\frac{\partial Q}{\partial x_k} = \frac{\partial f}{\partial x_k} - \frac{\partial r\sum_{i=1}^{m}(g_i(x))^{-1}}{\partial x_k} \quad k = 1,2,3 \ldots n \tag{7}$$

or

$$\frac{\partial Q}{\partial x_k} = \frac{\partial f}{\partial x_k} - \frac{\partial r\sum_{i=1}^{m}\ln(-g_i(x))}{\partial x_k} \quad k = 1,2,3 \ldots n. \tag{8}$$

Then we get all the derivation functions. These functions are respected to the variable. We use Newton Method to follow what is called a central path, which is a series of point we iterate through that all satisfy the equality constraints from the original problem, but give increasingly more optimized values for the objective function, with the inequality constraints not necessarily equal to 0. We can also let directly to 0 and calculate the limitation for all the derivation functions to work out the values of $x_1, x_2 \ldots x_n$. However, it is difficult to put it into actual use because of the difficulty of calculating a limitation by a program.

In our problem we notice that all the constraints are in equation 9:

$$\begin{aligned} &min\ \mathrm{f(x)} \\ &\text{s.t. } \mathrm{g}_i(x) \geq 0, j = 1,2,3...,m \end{aligned} \tag{9}$$

So, we need to add a minus sign for all the constraints beforehand to match the method. This will not make any change to the optimization.

We first take the initial state for our problem by giving strictly feasible x to our objective function. However, it is not the optimal solution. We also set μ as a small but positive value to decide whether the procedure is reaching our target. Then we repeated the following procedure. First, we compute x to our barrier function as the initial minimizing direction x_1. Then we use the Newton method to move a small step to a optimal direction and update the x_1 to x_2. Then we calculate the barrier function (equation 5 or equation 6) with x_1 and x_2. Then compare the difference between Q_1 and $Q_2$with μ. If the difference is smaller then μ, the procedure should be stop. Otherwise, we should continue the procedure by updating the current direction with the new direction calculated by newton method. The pseudo-code is described in detail in algorithm 1.

Algorithm 1. Barrier method for solving our problem.

1: Given strictly feasible $x, r^{(0)} > 0, \mu > 1$, and tolerance $\epsilon > 0$.

2: Set $r := r^{(0)}/\mu$.

3: **repeat**

4: Set $t := \mu t$.

5: Centering step. Compute $x^*(r)$ by minimizing $Q(x, r)$, subject to $Ax = b$, starting at x

6: Update. $x := x^*(r)$

7: **until** Duality gap $\frac{m}{r} < \epsilon$.

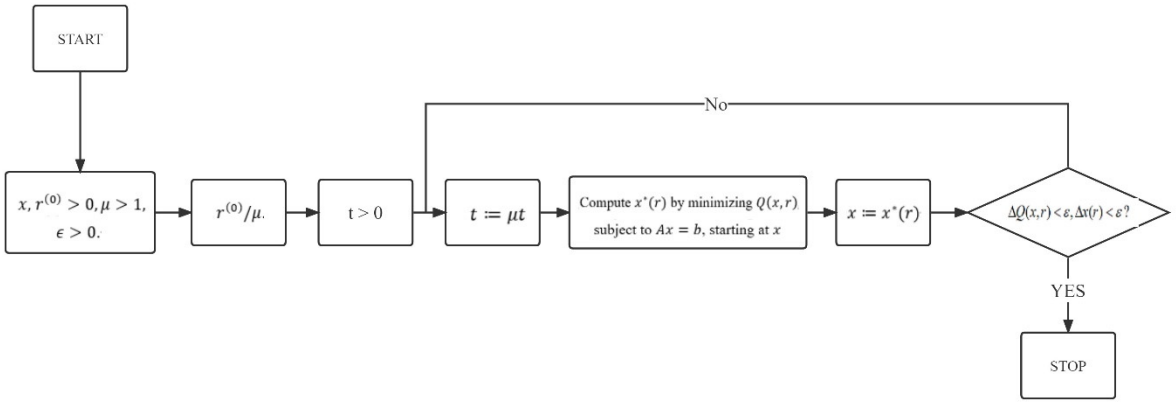

Figure 2. Flow chart of our algorithm

5. Simulation Results and Proof of reliability

To verify the algorithm we proposed, the simulations for a specific scene are conducted in 2 cases. 3 types of goods are stored on 5 shelves in case 1. The goods are marked as good A, good B, and good C, respectively. And each shelf can store a maximum of 30 goods A, 20 goods B, and 15 goods C. The main goal of case 1 is to retrieve 22 goods A from this area with a minimized moving distance. In case 2, the number of shelves is added to 15. The capacity of shelf for each type of good remains unchanged. The main goal in this case is to fetch 50 goods A.

The locations of the shelves in two cases are randomly generated. Then all the corresponding parameters were input to the model, the simulation results can be derived as the figure 2 and figure 3 show. In figure 2 and figure 3, node 1 means that it is $sort_1$. The vectors between the nodes represent the distances between different shelves.

We note the number of the distance on each vector to show the cost for each move. The results of case 1 is shown in table 1. Shelf 2 and shelf 4 were chosen as the targeted shelves and the minimization distance is 84. After the delivery, the goods A stored on shelf 2 and shelf 4 changed from 12, and 17 to 0 and 7, respectively. In case 2, the shelf 2, shelf 4 and shelf 11 were the final chosen shelves. The minimization distance is 178. The goods A stored on shelf 2 and shelf 4 verified from 12 and 17 to 0 and 7, respectively. The results of two cases were certified through exhaustive enumeration. All other ways to fetch the goods is longer than the way we have chosen. Here we let the AGV to fetch goods in a nearest principle. The results are shown in table 4 and table 5. The AGV needs to cover 92 units of distance in case 1 and 237 units of distance in case 2. Compared with our algorithm (84 units in case 1 and 178 units in case 2), the total distance is longer and the shelves passing by is more. Therefore, the correctness of this algorithm is proved. As figure 5 shows, the derivatives of the barrier function $Q(x, r)$ have reached a global minimum, which indicates that we have achieved the optimal solution of the problem.

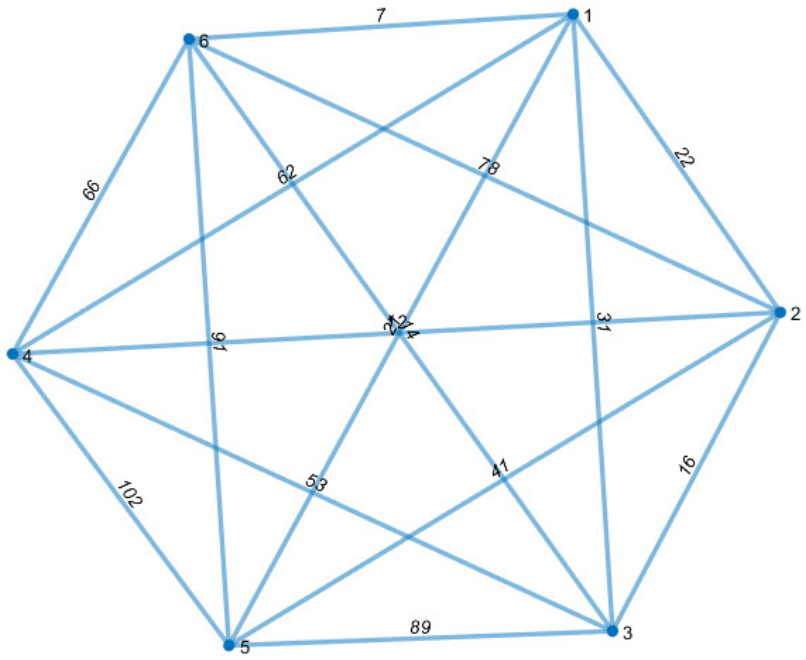

Figure 3. Picture shows the graph of case 1.

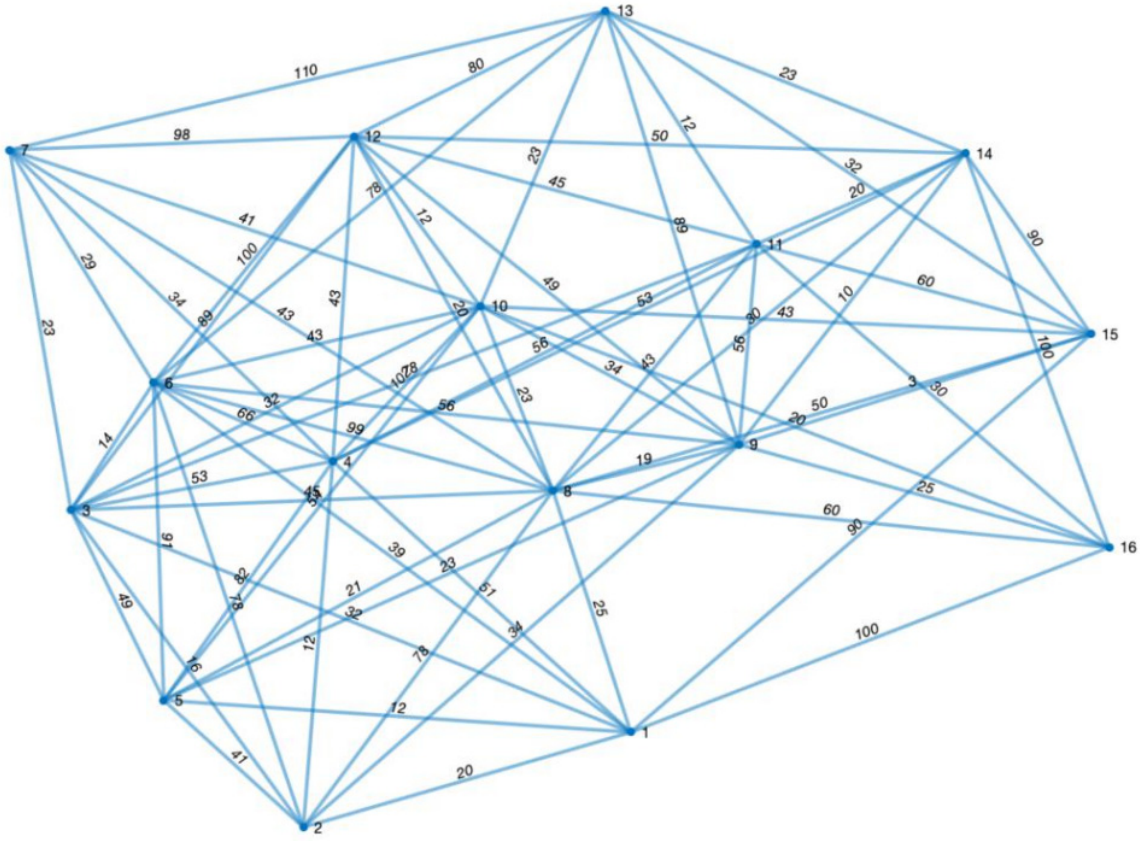

Figure 4. Picture shows the graph of case 2.

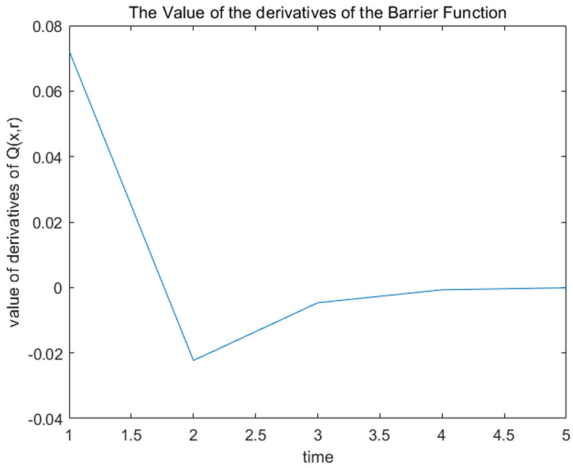

Figure 5. The value of the derivatives of the barrier function

Table 2. Result for case 1.

The shelves' number we choose	s_2, s_4
The total minimum distance	84

Table 3. Result for case 2.	
The shelves' number we choose	s_2, s_4, s_{11}
The total minimum distance	178

Table 4. Result for case 1 using nearest principle.	
The shelves' number we choose	s_2, s_3, s_5, s_6
The total minimum distance	92

Table 5. Result for case 2 using nearest principle.	
The shelves' number we choose	s_2, s_3, s_5, s_7, s_8
The total minimum distance	237

6. Conclusion

This paper has discussed the problem of scheduling an AGV in an automated storage and retrieval system. The system is set to be optimized with simplification and MILP. The result shows that our model can perform well in most cases to schedule the shortest distance for the AGV. In our model, the strategy can be used not only on the AGV but also between the sorting stations at different levels. The whole system with multiple levels can import our strategy to each level to optimize the whole system.

References

[1] Mehdi F, Asghar M, Michael H and Kate S 2018 A cross-entropy method for optimising robotic automated storage and retrieval systems *Int. J. Prod. R.* **56** 6450-72

[2] Dongkyoung C 2004 Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar Coordinates *IEEE T. Contr. Syst. T.* **12** 637-44

[3] Jun Der L, Andre K, Yi Ping L, Larry Jung Hsing L and Yi Wei H 2018 A green vehicle routing method for the regional logistics center *IEEE Int. Conf. Ind. Eng. Eng. Manage.* 71-5

[4] Stacia K W Robert K J and Jeffrey L B 2004 Automatic annotation of organellar genomes with DOGMA Bioinformatics **20** 3252-5

[5] Kelen V, Lu´ıs F R, Nadia Junqueira M, Marcelo B, A Paulo M 2016 Integrated tasks assignment and routing for the estimation of the optimal number of AGVS *Int. J. Adv. Manuf. Technol.* **82** 719-36

[6] Claudia C, Simone F, Simona S and Silvia S 2016 An MILP optimization problem for sizing port rail networks and planning shunting operations in container terminals IEEE T. Autom. Sci. Eng. **13** 1492-503

[7] Qing W, Shuai P and Shuan L 2020 *Chinese Control and Decision Conf.* (Shenyang) pp 2220-2224

[8] Satyanarayana G M, David W C and Kaarthik S 2016 Path planning for cooperative routing of air-ground vehicles *American Control Conf.* (Boston: Boston Marriott Copley Place) pp 4630-4635

[9] Qasim A, Hammad S, Imran A and Sridevi T 2016 Concurrency control in distributed database system *International Conf. on Computer Communication and Informatics* (Coimbatore)

[10] Wen X, Aiguo S and Lifeng Z 2021 *IEEE International Conf. on Robotics and Automation* (Xi'an) pp 11182-11188

[11] Joshua E E and Steven P G 2007 Target-decoy search strategy for increased confidence in large- scale protein identifications by mass spectrometry *Nat. Methods* **4** 207-14

[12] Oscar M, Ulises O and Roberto S 2015 Path planning for mobile robots using Bacterial Potential Field for avoiding static and dynamic obstacles *Expert Syst. Appl.* **42** 5177-91