

Research Article

Design and Implementation of the Optimization Algorithm in the Layout of Parking Lot Guidance

Zhendong Liu , Dongyan Li, Yurong Yang, Xi Chen, Xinrong Lv, and Xiaofeng Li

School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China

Correspondence should be addressed to Zhendong Liu; liuzd2000@126.com

Received 31 December 2020; Revised 11 February 2021; Accepted 27 March 2021; Published 12 April 2021

Academic Editor: Xingsi Xue

Copyright © 2021 Zhendong Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The information guidance system for parking spaces in large- and medium-sized parking lots is not efficient at present. It tends to be difficult to find an empty parking space in parking lots in big cities. One of the problems is the large amount of calculation in the traditional Dijkstra algorithm. In this paper, an improved Dijkstra algorithm is presented and optimized to find the best parking path with the purpose of looking for the nearest free parking space based on the layout model in parking lot parking guidance. The experiments show that the improved Dijkstra algorithm can find the optimal parking space and the optimal parking path and improve the parking efficiency.

1. Introduction

With the continuous expansion of the scale of cities, the problem of parking difficulties has become increasingly prominent and traffic accidents frequently occur. The main reasons for the problem of “difficult parking” are as follows: the current urban parking spaces are in short supply; more importantly, in the process of searching for parking spaces, people can get less valuable parking space information; and there is a lack in parking space information management platform to guide vehicle drivers to park reasonably [1]. Solutions that help people to find the nearest parking lot have been put forward by researchers with the application of GPS technology/BeiDou technology becomes mature. The research focuses on outdoor parking guidance, but the indoor navigation technology has not been developed as it should be [2]. In fact, the indoor navigation technology and the outdoor navigation technology are essentially the same and they all need three kinds of technical support, namely, indoor positioning technology, indoor map, and path planning technology [3]. The research of indoor navigation and positioning technology has been paid more attention, and its application has gradually become popular. In the future, the construction of smart cities will be inseparable from indoor navigation and positioning technology. The develop-

ment of indoor navigation and positioning technology with high precision, low cost, and universality and the realization of indoor and outdoor seamless navigation and positioning have always been the hot and difficult research topics at home and abroad [4]. Now, the development of indoor navigation has become a new direction for major enterprises to break through. One of the main applications of indoor navigation is to guide cars to park in parking lots. When the car owners arrives at the destination parking lot, they need to rely on the guidance of the managers in the parking lot or drive blindly to find the parking space, which wastes time and energy and also brings new problems: parade parking, resulting in congestion in the parking lot. It may take a long time for car owners to find appropriate parking spaces. It will greatly affect the parking of car owners when they encounter traffic congestion in the parking lot. Therefore, we need a parking space information management platform, which will be better if it has the function of guiding parking. It is an urgent problem that the guidance of parking spaces is imperfect and needs to be solved [5].

To address this challenge, we established an abstract data model of parking guidance based on the layout model in parking lot parking guidance, optimized this data model by using Dijkstra’s improved algorithm, and finally found an optimal parking path with the lowest cost. With this method,

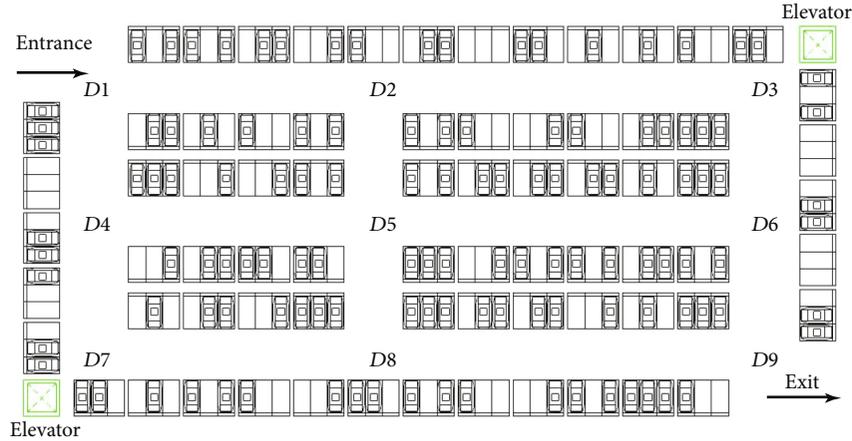


FIGURE 1: Parking space guidance layout diagram.

the car owners can find an idle parking space in the shortest time and parking efficiency and users' parking experience is greatly improved; the efficiency of the parking lot will also be improved.

2. System Structure and Parking Space Model Establishment

2.1. Parking Garage Information Guidance Management System. We designed and implemented an parking garage information guidance management system for intelligent parking. The main functions of the system includes parking space information collection, transmission, guidance control, and display statistics. Its main function is to monitor the occupancy situation of parking spaces in the parking lot and parking specifications in real time. There are many methods for monitoring the parking space status. For example, GPP and PGS2 systems adopt ultrasonic waves, while Siemens systems use parking sensors placed on the ground [6]. Monitoring vehicles by infrared sensors is one of the most sensible methods while considering that the system is mainly aimed at large- and medium-sized parking lots, and ultrasonic sensors are very sensitive to temperature changes and extreme air [7]. Therefore the system adopts infrared photoelectric switches. By arranging four infrared photoelectric switches in the parking space and adjusting the positions of infrared radio and television switches, the system can judge the state of vehicle parking irregularly if all four infrared photoelectric switches are covered by vehicles; it is standard. The working process of the parking garage information guidance management system is as follows: the infrared photoelectric switch on the parking space collects the specific information of the parking space, transmits the collected information, obtains the parking space status in the background, and selects all the empty parking spaces. When the owner enters the parking lot, there is a demand for finding the best empty parking space. The system finds the shortest path of the best empty parking space for the vehicle according to the guidance algorithm and publishes the information to the vehicle owner and guides him to park his vehicle as soon as possible. At the same time, the system will also detect the parking

specification after the owner has parked the vehicle, which is mainly realized by setting reasonable infrared photoelectric switches. If the parking is standard, the system will feedback to the owner that the parking is successful. If the parking is not standardized, the system will prompt the owner of the vehicle to park the vehicle in a standardized way through the mobile phone. Intelligent parking solutions and intelligent systems provide a way to obtain parking lot information [8] and guide car owners to park faster and better and improve parking efficiency.

2.2. Model of the Parking Space Guidance Layout. Figure 1 is a typical parking space guidance layout diagram. The parking lot has only one exit and one entrance, and they are set up separately without affecting each other. The two pedestrian elevators are located in the middle of the entrance and exit, which are symmetrical as a whole. We discuss the optimization of the optimal parking path in this paper, taking this typical parking layout as an example. It can be regarded as a weighted graph; the cars to be parked, the intersections of lanes, and all free parking spaces are regarded as nodes; the paths in each driving direction are regarded as an edge; and the length of the driving road can be regarded as the weight of the edge. According to the path weights in the parking space guidance layout, the path optimization algorithm is used to calculate the weights of parking spaces and the best parking space and the corresponding parking path are selected to improve parking efficiency.

Many factors need to be considered in the selection of an empty parking space. The problems of the parking space itself are as follows: too close to the exit or entrance may cause safety problems and the empty parking space without cars parked next to it is better than the empty parking space with cars parked next to it [9]. There is also the issue of the owner's angle: the distance between the empty parking space and the pedestrian elevator and the distance between the empty parking space and the exit or entrance [10]. However, considering that the parking garage information guidance management system also involves guiding the car to the parking lot in the early stage and the owner has been driving the car for a period of time and the owner may be tired after

// P is the set of free parking spaces, S is the weight (path distance) of free parking spaces

- 1 calculate the distance between longitude and latitude of free parking spaces and longitude and latitude of the car M to be parked, the set of 5 free parking Spaces with the shortest distance is denoted as P and the free parking Spaces are marked as $P_i (i \in [0, 4])$.
- 2 Supposing that the shortest time shortest path set is $T (P \subseteq T)$, the corresponding weights (path distance) is $S(i)$, the set T is initialized to null and the weights $S(i)$ is initialized to 0, $i \in [0, 4]$;
- 3 While not all the elements in the set P enter the set T do
- 4 Select $P_k, P_k = \{P_i \in P - T \mid \min S(i)\}$, P_k is the end point of the shortest path from the current car M , $T = T \cup \{P_k\}$;
- 5 Update the weight of the shortest path from car M to P_k , $S(k) = \min\{S(k), S(i) + S(i, k)\}$ // $S(i, k)$ indicates the path distance between node i and node k .
- 6 End while
- 7 Output the final weights of all empty parking nodes in the empty parking space set P , and the berth P_i corresponding to $\min\{S(i)\}$ is the optimal berth.
- 8 The path corresponding to $W = \{M, \dots, Dn, \dots, P_i\}$ is the optimal path from the car M to the optimal P_i .

ALGORITHM 1: ImDA(P, M).

arriving at the parking lot, so it is better to let the owner park the car in the shortest time. In this way, the parking efficiency of the parking lot is improved and the situation of car congestion is avoided. Set the free parking space in the parking lot as P_i , set the path distance from the parked vehicle to the free parking space as $S(i)$, and then, set the corresponding optimal parking path as $\min\{S(i)\}$.

3. Optimization Algorithm in the Layout of Parking Lot Guidance

In life, we are faced with many problems related to the shortest path. For example, it can reduce the cost of transporting unit materials and save freight expenses if we chose a reasonable transportation route under the condition of the existing traffic network. Thus, the problem of path optimization is crucial to our life.

3.1. Classical Shortest Path Algorithm and Its Disadvantages. The shortest path problem is a classical algorithm problem in graph theory, which is aimed at finding the shortest path between two nodes in a graph (composed of nodes and paths). Shortest path algorithms which are commonly used include the Dijkstra algorithm, Floyd algorithm, A* (A Star) algorithm, BFS algorithm, and Johnson algorithm [11]. Among them, the Dijkstra algorithm is used to solve the shortest distance between a certain source point and other end points [12]; simply put that it is a typical single-source shortest path algorithm, which is neither DFS search nor BFS search, DFS occupies less memory but is slower, and BFS occupies more memory but is faster. The Dijkstra algorithm avoids these two problems. The algorithm is simple and easy to implement and has a high practical value. The Floyd algorithm is used to find the distance between any two points, which is different from the Dijkstra algorithm. In short, the Floyd algorithm is a multisource shortest path algorithm, which adopts the method of dynamic programming. The Floyd algorithm is simple and easy to implement, but by comparing their time complexity, Dijkstra algorithm's time complexity is generally $O(n^2)$, while Floyd algorithm's time complexity is generally $O(n^3)$. Therefore, the Dijkstra algorithm is faster than the Floyd algorithm, that is, the Dijk-

stra algorithm has more advantages in finding the shortest path of a single source. As we all know, finding free parking spaces in parking lots is a typical example of finding the shortest path of a single source. In addition, the Dijkstra algorithm is used to determine the shortest path and configure other conditions such as parking lane intersections, parking spaces, and their occupancy rates. The free parking space resources can be used more efficiently [13]. Therefore, this paper mainly improves the Dijkstra algorithm.

In the parking space guidance layout diagram, it will have a large amount of calculation when using the Dijkstra algorithm to find the shortest parking path and the efficiency of finding the best parking space will be very low, resulting in the problem of low parking efficiency. Therefore, the algorithm needs to be improved.

3.2. Improved Dijkstra Algorithm. There will be a sea of nodes and path when the traditional Dijkstra algorithm is applied to large parking lots, which makes the weighted graph more complicated. When the traditional Dijkstra algorithm is used to find the best parking path, the calculation is heavy and the parking efficiency is low. It is the simplest way to choose the nearest free parking space to the vehicle itself when we choose the optimal free parking space. Based on this idea, we design an optimization algorithm based on the Dijkstra algorithm in this paper. The basic idea of this optimization algorithm is as follows: obtain the latitude and longitude of free parking spaces and vehicles to be parked and calculate the distance between free parking spaces and vehicles to be parked by using the obtained latitude and longitude information; Euclidean distance is used here because it can better represent the true distance between two nodes. We select the nearest five free parking spaces which can be appropriately increased when the parking lot is as large as nodes and add them to the weighted graph, thus greatly reducing the number of nodes in the weighted graph. Then, the weights of the five free parking spaces are calculated (the path distance is taken as the weight). Finally, the parking space with the minimum value is selected to determine the best parking path. To facilitate the representation of parking paths, the lane intersections $D1-D9$ are introduced. Here is the improved Dijkstra algorithm:

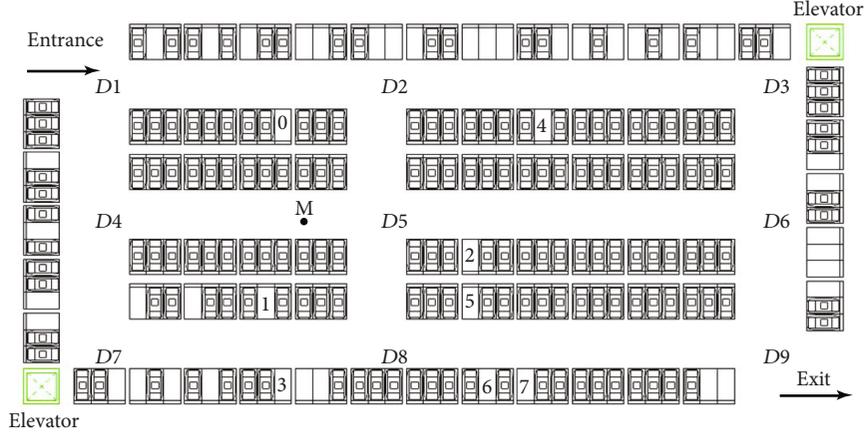


FIGURE 2: Parking space guidance layout at a certain time.

TABLE 1: Path distance in the layout.

| Path | Path distance | Path | Path distance |
|---------|---------------|----------|---------------|
| $D1-D4$ | 7 | P_0-D2 | 6 |
| $D4-D7$ | 8 | P_1-D8 | 7 |
| $D4-D5$ | 15 | P_2-D5 | 5 |
| $D5-D6$ | 18 | P_3-D8 | 6 |
| $M-D5$ | 5 | P_4-D2 | 8 |

TABLE 2: Parking space guidance weight and path.

| Parking space | Path | Path distance (weight) |
|---------------|---|------------------------|
| P_0 | $M \rightarrow D5 \rightarrow D2 \rightarrow P_0$ | 18 |
| P_1 | $M \rightarrow D5 \rightarrow D8 \rightarrow P_1$ | 20 |
| P_2 | $M \rightarrow D5 \rightarrow P_2$ | 10 |
| P_3 | $M \rightarrow D5 \rightarrow D8 \rightarrow P_3$ | 19 |
| P_4 | $M \rightarrow D5 \rightarrow D2 \rightarrow P_4$ | 20 |

4. Experimental Analysis

Combined with the above improved Dijkstra algorithm, a specific example is selected for analysis. Figure 2 is a parking space guidance layout diagram of the parking lot at a certain time, which can clearly see the parking space occupation situation in the parking lot at this time. The following example takes Figure 2 as an example; assume that $P_0 \dots P_7$ shown in the figure is the free parking space in the parking lot at this time (assuming that all but these seven parking spaces are occupied) and M represents the car to be parked.

According to the abovementioned optimization algorithm, firstly, the free parking spaces $P_0 \dots P_7$ (represented by 0, 1, 2, 3, 4, 5, 6, and 7, respectively, in the figure) and the longitude and latitude of the car to be parked are obtained and the free parking spaces (the center points of rectangular parking spaces) are taken to obtain $C_0 \dots C_7$; calculate the distance from M to $C_0 \dots C_7$ and select the five points with the shortest straight line distance. The five points screened out in this example are P_0, P_1, P_2, P_3, P_4 (represented by 0, 1, 2, 3, and 4, respectively, in the figure), and these five points are the nodes that need to be added to the weighted graph.

Assuming that the measured distance (taking the width of a parking space as the unit distance) is as shown in Table 1 below, assume that each road is as wide as three parking widths, points $D1-D9$: two parking spaces from the far left of the road and one parking space from the right side of the road.

According to the abovementioned optimization algorithm for the parking space guidance layout diagram, the five points $P_0, P_1, P_2, P_3,$ and P_4 closest to the car to be parked are

screened out and the weights of these five points are calculated, namely, the path distance, which can also be said to be the driving distance of the route. The smaller the weight, the smaller the driving distance from the driving to the parking space, the simpler the parking, and the shorter the required time. The optimal parking path and weights from the car M to the free parking space are shown in Table 2.

It is not difficult to see that the parking guidance weight of parking space P_2 is the lowest, that is, the best parking space for car M is P_2 and the corresponding best parking path is $M \rightarrow D5 \rightarrow P_2$. After preliminary screening, the optional idle parking spaces are reduced and the number of nodes is greatly reduced. The shortest straight line distance between the source point and the end point of the car does not mean the shortest driving route. There may be the following special circumstances: the straight line distance is short, but it cannot be reached directly, so it takes a long distance to reach the parking space and the actual driving route is longer than other idle parking spaces. Therefore, parking spaces should be screened first, then, the shortest path of the screened parking spaces should be further calculated, and finally, the smallest weight should be selected as the best parking space. In this example, P_0 is the closest straight line distance from the source point (car M) to the free parking space; the corresponding parking path is $M \rightarrow D5 \rightarrow D2 \rightarrow P_0$, and the weight (path distance) is 18. However, according to the above algorithm, the shortest parking path is $M \rightarrow D5 \rightarrow P_2$ and P_2 is the best parking space with a weight (path distance) of 10. By comparing the parking space P_0 , obtained by the shortest straight line distance with the parking space P_2 obtained by the algorithm, it can be seen

that the optimized Dijkstra algorithm reduces the path distance, reduces the driving distance, and improves the parking efficiency, which further illustrates the superiority of the algorithm. A parking garage information guidance management system can guide owners to park the car faster and detect the parking conditions of vehicle owners through diffuse infrared photoelectric switches deployed in parking spaces. If it is detected that the car owners are parking irregularly, it can push the parking irregularity information to the user's mobile phone to remind the user to park the car regularly.

5. Conclusion

The traditional Dijkstra algorithm takes all empty parking spaces, parked cars, and lane intersections as nodes and adds all possible paths into the weighted graph, which is computationally intensive and inefficient. In view of this shortcoming of the traditional Dijkstra algorithm, we design an improved Dijkstra algorithm by limiting conditions, the number of nodes and the number of paths are reduced, the amount of computation is reduced, the computing efficiency is improved, and thus, the parking efficiency and parking experience are improved.

The main innovation of this paper is as follows: when looking for the best parking space, not considering the walking distance of the owner or the distance from the entrance and exit, then, the appropriate weight calculation method is selected to choose the parking space, which is different from other studies. We choose the car as the search center in this paper, which can ensure that the parking space is relatively close and optimal and ensure that the car is parked in the shortest time. In addition, it can also ensure that cars can search parking spaces anytime and anywhere in the parking lot, so as to find parking spaces that are close to themselves and convenient, with relatively few limitations.

However, there are many shortcomings in this paper that need to be further improved:

- (1) It is necessary to strictly prove the improved Dijkstra algorithm to ensure that it is effective for graphs of large order, which is mainly aimed at the case of large parking lots
- (2) Finding an unoccupied parking slot by the interested vehicle owners with the least overhead becomes an NP-hard problem bounded by various constraints [14]. Therefore, it is necessary to continue to improve the Dijkstra algorithm and improve the time complexity of the algorithm
- (3) The calculation method of weights is relatively simple, which needs to be further improved and strictly compared with other methods, such as the neural network-based predictive control approach [15, 16]
- (4) The parking space library information guidance management system is mentioned above, but this paper mainly introduces the Dijkstra algorithm for finding the best parking space [17, 18]. The descrip-

tion of the system is relatively few, and the system still needs further improvement. For example, after finding the best parking space, you can rely on the display at the intersection to indicate the driving direction or that the roadside indicator lights up to directly guide the car to the parking space, so as to realize a more intelligent parking system

Data Availability

All data are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Our work was supported by the NSFC under Grant nos. 61672328 and 61672323, and the research is also supported by the Science and Research Plan of the Luoyang branch of Henan Tobacco Company, no. 2020410300270078.

References

- [1] Y. Yang, *Design and Implementation of Intelligent Parking Guidance System*, Jilin University, 2018.
- [2] S. Liao, K. Du, Y. Zhao et al., "Scheme design of parking navigation system in indoor parking lot," *Value Engineering*, vol. 38, no. 7, pp. 162–164, 2019.
- [3] T. Lei and Y. Chu, "Analysis on the current development of indoor navigation technology," *Modern Business Trade Industry*, vol. 41, no. 24, p. 153, 2020.
- [4] W. Gao, C. Hou, W. Xu, and X. Chen, "Research progress and prospect of indoor navigation and positioning technology," *Journal of Navigation and Positioning*, vol. 41, no. 24, p. 153, 2020.
- [5] W. Wang and Z. Gai, "Design of parking guidance path in parking lot based on Dijkstra algorithm," *Network Security Technology & Application*, vol. 9, pp. 52–53, 2018.
- [6] J. Hanzl, "Parking information guidance systems and smart technologies application used in urban areas and multi-storey car parks [J]," *Transportation Research Procedia*, vol. 44, pp. 361–368, 2020.
- [7] M. Bachani, U. M. Qureshi, and F. K. Shaikh, "Performance analysis of proximity and light sensors for smart parking [J]," *Procedia Computer Science*, vol. 83, pp. 385–392, 2016.
- [8] R. Singh, C. Dutta, N. Singhal, and T. Choudhury, "An improved vehicle parking mechanism to reduce parking space searching time using firefly algorithm and feed forward back propagation method [J]," *Procedia Computer Science*, vol. 167, pp. 952–961, 2020.
- [9] W. Li, Q. Li, and S. Yu, "Parking guidance system of parking lot using internet of things technology," *Automation & Instrumentation*, vol. 4, pp. 234–235, 2015.
- [10] Y. Zhang and S. Tian, "Application of Dijkstra optimization algorithm in parking space guidance system," *Computer Measurement and Control*, vol. 22, no. 1, pp. 191–193, 2014.
- [11] S. Wang and Z. Wu, "Improved Dijkstra shortest path algorithm and its application," *Computer Science*, vol. 39, no. 5, pp. 223–228, 2012.

- [12] W. Yan and W. Wu, *Data Structure (C Language Version)*, Tsinghua University Press, 2012.
- [13] J. Fan, "Predicting vacant parking space availability: an SVR method with fruit fly optimisation," *IET Intelligent Transport Systems*, vol. 12, no. 10, pp. 1414–1420, 2018.
- [14] S. Saharan, N. Kumar, and S. Bawa, "An efficient smart parking pricing system for smart city environment: a machine-learning based approach [J]," *Future Generation Computer Systems*, vol. 106, pp. 622–640, 2020.
- [15] J.-H. Shin, J.-G. Kim, and H.-B. Jun, "Dynamic control of intelligent parking guidance using neural network predictive control," *Computers & Industrial Engineering*, vol. 120, pp. 15–30, 2018.
- [16] H. Liu, Y. Wang, and N. Fan, "A hybrid deep grouping algorithm for large scale global optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1112–1124, 2020.
- [17] S. Liu, Y. Wang, W. Tong, and S. Wei, "A fast and memory efficient MLCS algorithm by character merging for DNA sequences alignment," *Bioinformatics*, vol. 36, no. 4, pp. 1066–1073, 2020.
- [18] S. Wei, Y. Wang, Y. Yang, and S. Liu, "A path recorder algorithm for multiple longest common subsequences (MLCS) problems," *Bioinformatics*, vol. 36, no. 10, pp. 3035–3042, 2020.