© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

http://penerbit.uthm.edu.my/ojs/index.php/ijie

The International Journal of Integrated Engineering

A Smart Guidance Indoor Parking System Based on Dijkstra's Algorithm and Ant Colony Algorithm

K. Ibrahim. Ata, A. Che Soh*, A. J. Ishak, H. Jaafar

Control System & Signal Processing (CSSP) Research Centre,
Department of Electrical and Electronic Engineering, Faculty of Engineering,
University Putra Malaysia, Serdang, Selangor, 43400, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2020.12.02.001
Received 8 January 2019; Accepted 2 January 2020; Available online 28 February 2020

Abstract: One of the more apparent problems associated with the growing number of parking spaces in shopping complexes, office buildings, and other types of building is the lack of notification to drivers of vacant and occupied parking bays in a large parking space. Given the heavy traffic commonly seen during the weekends, most drivers might spend at least 20 to 30 minutes just to find an empty parking bay, which leads to wasteful fuel consumption. Currently, most shopping malls have conventional parking systems, where heavy traffic can lead to an increase in the number of accidents occurring. This is a serious problem that requires a solution. The existing parking systems are complex and have poor performance due to low-speed processing and inefficiency. Thus, to overcome these problems, a smart parking system should be designed and implemented. This paper introduces a smart guidance indoor parking system based on embedded system integrated with both the Dijkstra's algorithm and Ant Colony algorithm (ACO) to provide drivers with an efficient path to the nearest parking bay. The smart guidance indoor parking comprises two parts, namely the hardware and the software. The hardware part explains the components used in the system and the software part explains the algorithms which have been used in the system. The proposed smart guidance indoor parking has achieved 37.50%, 10.81%, and 34.88% improvement compared with the conventional Dijkstra's algorithm. The smart guidance indoor parking system was successfully developed in an adaptable structure, convenient cost and easy handling procedure.

Keywords: Smart parking system, Dijkstra's algorithm, shortest path problem, embedded system, Ant Colony algorithm (ACO).

1. Introduction

In today's world, people are mainly concerned about time and cost. Air pollution is also a worry for those who are conscious about the environment. People enjoy driving around and spending time exploring the city and visiting shopping malls. It has been found by many researchers that traffic congestion is among the main factors causing inefficient consumption of time and fuel (Litman, 2018). Thus, the places that are commonly frequented by a large number of visitors have the need for functional and reliable parking systems. However, current parking lots operate based only on the guidance of managers or sightless search for empty parking bays by drivers. The growing number of vehicles indicate a worsening of the traffic jam problem which in turn increases fuel consumption (Abidi *et al.*, 2015). These problems

**Corresponding author: azuracs@upm.edu.my*

become much bigger in a closed area such as indoor parking areas of the shopping complexes (Alija, 2015). During weekends, the malls are very crowded and usually packed with visitors, which makes it very challenging for the drivers to find an empty and suitable parking space to park their cars. Thus, an optimal solution should be proposed for the abovementioned problems. In this paper, a new smart guidance system is introduced based on Dijkstra's and ACO to generate a path guidance for a driver to find the nearest parking bay.

The most common solution for the shortest path problem is Dijkstra's algorithm. The traditional Dijkstra's algorithm is an algorithm to find the shortest paths between nodes in a weighted graph, for example, networks of the road. It was formulated by computer scientist Edsger W. Dijkstra in 1956 (Wang, 2008). Many researchers have tried to prepare an optimal path for parking system by making use of Dijkstra's algorithm. Jaafar *et al*. (2014) have applied Dijkstra's algorithm by calculating the distance between the car parking bay and the specified entrance mall, and subsequently suggested the shortest distance for arrival. (Cai, Zhang, and Yongjie , 2015) have optimized the traditional Dijkstra's algorithm with a constrained optimization condition. Thus, the optimization Dijkstra produced results which are easier and higher in efficiency as compared to the traditional Dijkstra (Han *et al*., 2017), which has been applied to a large parking area in Beijing. The system is based on the use of Wi-Fi due to its advantages over the GPS. The system allows the user to interact by using his cell phone. All of these proposed solutions have contributed to improvement of the parking system management.

In addition, the ACO is used to optimize the shortest path significantly vide ACO, based on the foraging behavior of ant colonies (Zhu and Kwong, 2010). Although the applications of ACO on the parking systems are less significant, it is able to provide efficiency in path planning. There are studies that have successfully applied ACO in the parking system in order to find the shortest path (Wang, Zhang and Chui, 2017) has been applied for the smart cities by using the internet of things technology. The proposed solution does not only take into account the shortest path but also the fewest routes. In addition, another study on integrated ACO with local search to optimize the path has been proposed by Abdulkader, Gajpal, and Elmekkawy (2015). Simulations for both studies found that ACO showed better results with less computational time. Thus, this paper introduces a new method based on both ACO as well as Dijkstra's algorithm. The proposed solution in this paper is made on a 'first come first serve' basis. Moreover, the system generates the optimal path taking into consideration the walking distance between a car parking bay and the entrance of the mall, as well as the driving distance from the main gate.

2. Methodology

The proposed solution basically consists of two parts, namely hardware and software. In order to achieve more reliability and efficiency in parking management, the system serves both visitors and employees who work in the building. Visitors can select a particular mall entrance, and the system will then calculate and show the path to the vacant parking bay nearest to the selected entrance. The system will reserve the parking bay for the said visitor for three minutes; if the driver doesn't park in this space within this time period, the system will unreserve this parking bay and it will be available to another driver. Meanwhile, for employees who work in the building, they should each be given a radio-frequency identification (RFID) card to obtain access to their reserved parking bays. Fig.1 illustrates the proposed system design.

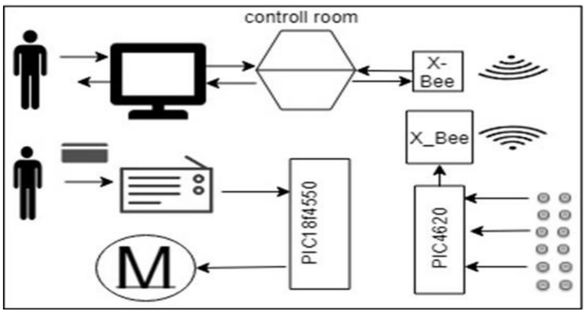

Fig. 1 - The proposed system block diagram

2.1 Hardware Development

The system contains two main circuits. The first circuit works with car bays to record the status of the bay. The second circuit will be on the main gate to interact with the user. Both circuits were tested and designed by Proteus 8 professional. The first circuit is on the parking bays which does the task of detecting the parking status and sending them to the PC by some of the integrated circuit components as illustrated in **Error! Reference source not found.** 2.

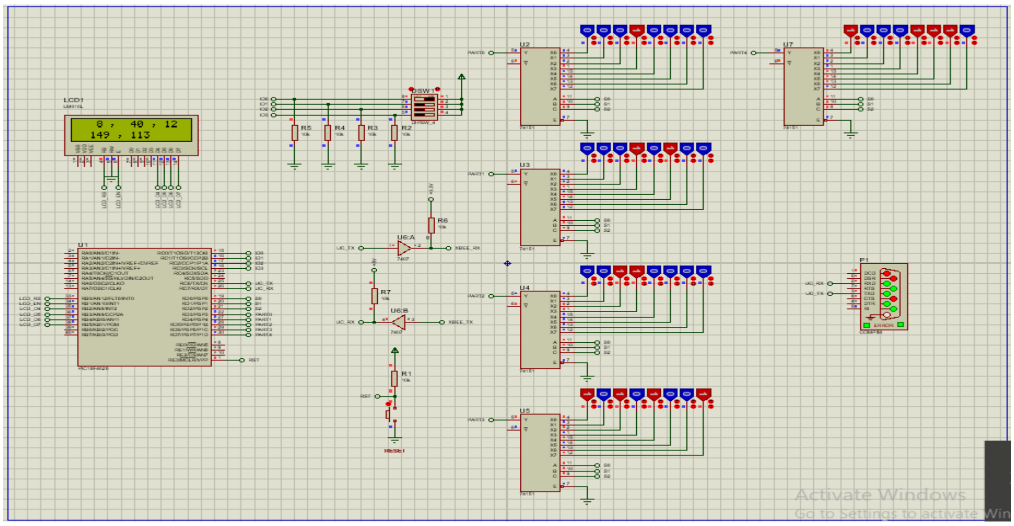

Fig. 2 - The bays circuit simulation

This circuit contains 40 toggles/ dual in-line package (DIP) switches on every parking bay to detect its status. In the case that the parking bay is vacant, the toggle switch is 0, and in the full parking case, the toggle is 1. The last switch is chosen as another kind of swathes which is a contact touch switch for simulation purposes only. This type of switch is designed to be used on a printed circuit board along with other electronic components and is commonly used to customize the behavior of an electronic device for specific situations, it can be used to determine cars exiting. These switches are connected to multiplexers (MUX TTL/74151) 8 to 1. This MUX will select one of the eight inputs signals and forwards the selected input into a single line. Five MUX's have been used to multiplex 40 parking switches for every eight switches in one multiplexer.

The outputs and selectors are connected with port RD (RD0-RD7) in the microcontroller. The microcontroller will output the parking status into two devices i.e. the LCD a liquid-crystal display **(**JHD162A) (16×2 LCD) to show the availability for the parking and the XBee to send them to the PC.

The PIC4620 is interfaced with Zigbee (XBee pro S2B) by USART "Addressable Universal Synchronous Asynchronous Receiver and Transmitter". USART is a two-lines communication system (RX for sending and TX for receiving) in which the data flow serially. The sending line in the PIC (TX) is connected with the receiving line (RX) in the XBee and the receiving line in the PIC(RX) is connected to the sending line (TX). The parking status will be sent form PIC TX to XBee RX. This XBee will send these data to another master XBee which interfaced with the PC by XBee adapter. The PC connection can be used to configure the XBee Module through Digi's X-CTU software. XCTU is a free, multi-platform application compatible with Windows, MacOS, and Linux. XCTU is more like the XBee control center. It has a serial terminal, settings option and device cloud among many other components, see **Error! Reference source not found.** 3.

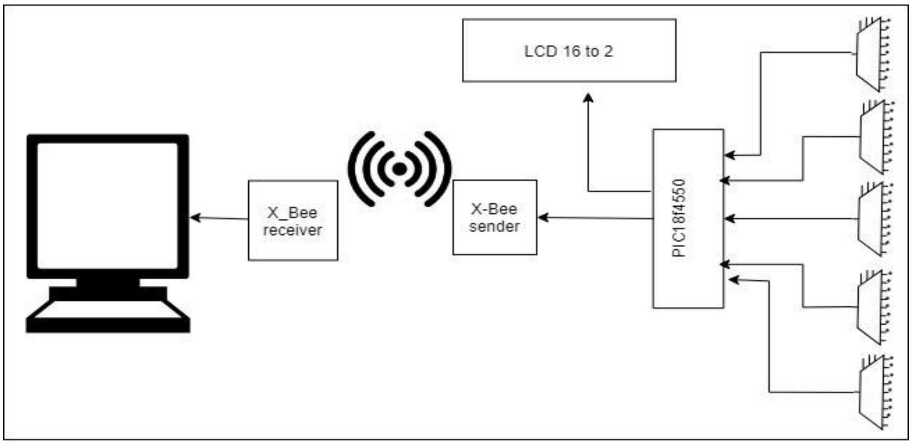

Fig. 3 - The proposed system block diagram

The second circuit is located on the main gate from which the cars enter to the parking. Because the system has two kinds of users, this circuit is designed to meet the system requirements properly as shown in **Error! Reference source not found.** 4. This circuit is designed to control the motor for opening and closing the gate and to identify the employees by the RFID card and reader as illustrated in **Error! Reference source not found.**5.

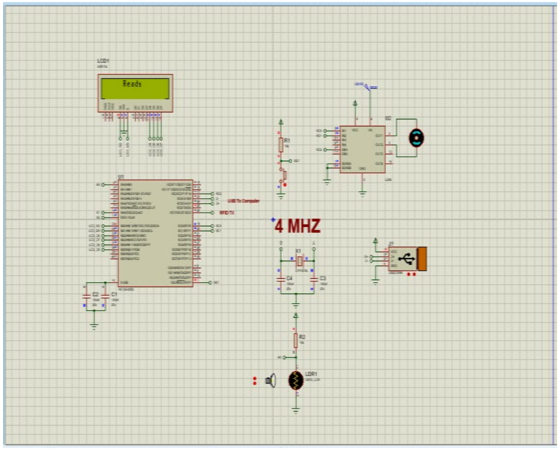

Fig. 4 - The gate circuit simulation

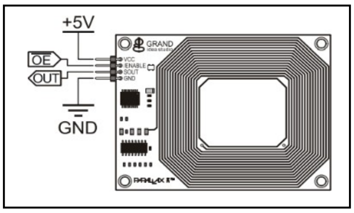

Fig. 5 - RFID reader

For the employees, the RFID Reader Module identification is used in which there are specific parking bays are reserved previously for them. This RFID serial reader will send radio frequency waves 125 kHz to read the RFID card. It then sends it to the PIC, and the PIC will send it to the PC by USB. The employees' ID cards are stored in the program previously. If the received ID is previously defined, the system will show the acceptance message on GUI and the gate will be opened by the DC (direct current) motor. The working concept of this motor is to convert a direct current electrical energy into mechanical energy. DC Motors requires high current and high voltage than a Microcontroller can handle. Moreover, to reverse a DC motor, the circuit needs to be able to reverse the direction of the current in the motor. The easiest way to do this is using an H-bridge circuit.

The direction of the motor is controlled by the H-bridge L298 which can control two motors. The first pair of drivers is enabled by connecting EN1 to Logic HIGH. IN1 and IN2 are connected to RD0 and RD1 of PIC Microcontroller respectively which are used to provide a control signal to the DC Motor to reverse the direction. DC Motor is connected to OUT1 and OUT2 of the H-bridge.

When the motor is open to allow a car to pass, the light dependent resistor LDR will detect whether the car has arrived to the planned destination or not. This LDR is a component whose resistance changes when the amount of light falling on it changes. Thus, when a car is passing above the LDR, the falling light will be blocked so the resistance will be increase then decrease when the car leaves. Once the resistor changed, the microcontroller will send the H -Bridge to send a high volt to the motor then the motor will close. Fig. **6** 6 shows the gate circuit communication.

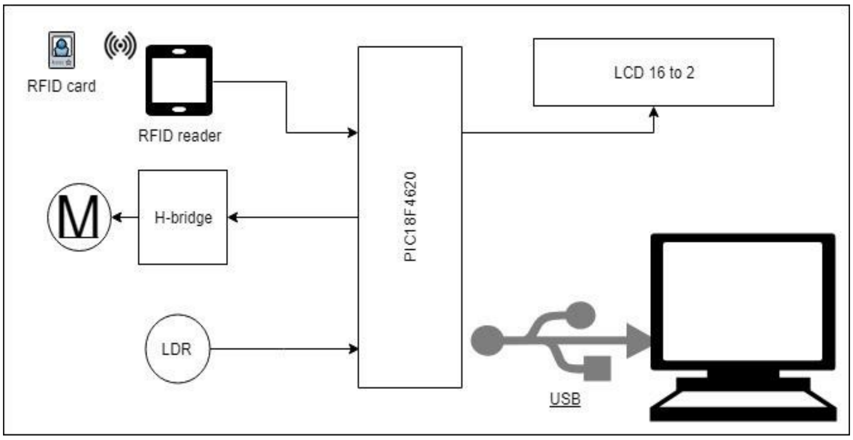

Fig. 6 - The block diagram of circuit 2 (gate circuit)

2.2. Software Development

The software development basically consists of two algorithms, which are Dijkstra's and ACO. Dijkstra's algorithm is a common algorithm applied to solve the shortest path problem (SPP). It can get the shortest path from a node to every other node. Its time complexity is O(n2). The *n* represents a count of a node in the graph. This minimum distance cannot only be measured with pure length, but can also be measured in other ways, such as time overhead, economic cost, and throughput (Deng *et al.*, 2012).

Moreover, the ACO is inspired by the behaviour of ant colonies and other insects. The real ants forage for the food source by making use of specific algorithm. The secret behind this fascinating behavior is that the ants produce a chemical called pheromone, which is deposited while the ants walk between the nest and the food sources. The ants will choose to follow the path with higher pheromone levels with a higher probability than trails with lower pheromone levels. As more and more ants follow the trail, the trail becomes increasingly attractive (Walia, Banasal and Harleen , 2014). To find the shortest path from a source *s* vertex or location to a destination location *d*, the Dijkstra's algorithm maintains a set *s* of vertices which final shortest-path weights from the sources have already been determined.

The proposed Path Guidance Algorithm (PGA) will be designed to find an optimized path for a car driver, starting from an entrance gate to a parking bay in a parking lot. In the implementation of the proposed PGA, the data will be forwarded to the personal computer with a USB slot and the path will be generated by the Dijkstra's Algorithm. Then the ACO will optimize it by performing certain tasks suitable for the parking lot. Each task will contribute towards solving the problem of finding the optimized guidance path in the parking lot. Moreover, the Binary Search Tree (BST) concept has been applied in the proposed algorithm in order to store information about the parking bays, to specify the car size, and from which gate the car enters the parking. Fig. 7 shows the flow chart of the procedure for the proposed PGA.

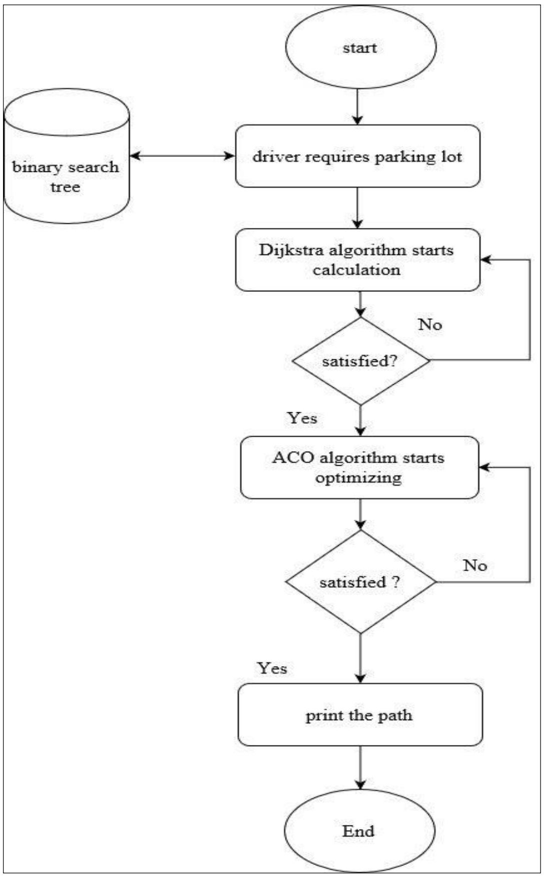

Fig. 7 - The Path Guidance Algorithm procedure

The problems of finding the optimized guidance path can be broken down and categorized into Opt1, Opt2, and Opt3. The problem of finding the shortest path between the gate and the parking bay is termed as Opt1. The problem of finding the nearest available bay to the selected entrance is termed as Opt 2. The problem of finding the optimal solution by combining the shortest path and the nearest available bay is termed as Opt3. It is important to consider both Opt1 and Opt2 in order to find the optimized guidance path. For instance, if only Opt1 has been considered, the path between the termed Opt1 and the gate will be the shortest. However, the termed Opt1 might not be the nearest one to the termed Opt2. Thus, only taking into consideration Opt1 will not result in finding the optimal guidance path.

However, if Opt1 and Opt2 have both been considered in the system, the generated path will be the nearest to the termed Opt2. However, there might be a different termed Opt1with similar distance from the termed Opt2 with shorter guidance path from the gate to it. Thus, Opt3 must be applied in the proposed algorithm to ensure that the generated guidance path is optimal.

2.3 The Hardware-Software Integration

The software has been integrated with hardware by importing the data from all the circuits using the PC Universal Serial Bus USB ports as shown in

Fig.8.

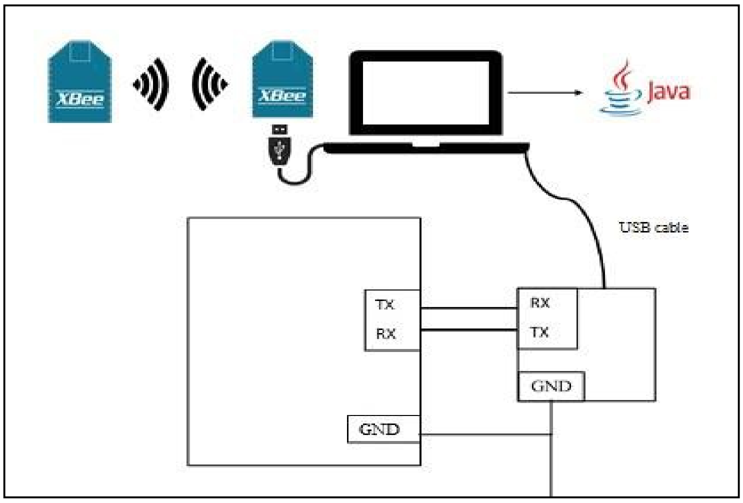

Fig. 8 - The block diagram of the hardware and software integration

The first port is on the XBee circuit side in which the parking bay status will be sanded to the program serially by the XBee adapter. The data will be sanded in Hexadecimal then converted to binary bits as a bit 0 and 1. Each one of the parking bays will take a value of zero and one. For example, if the bits are 00001111 have been obtained and the circuit switch ID is zero, the first parking in the port zero which is B0 in the program will take the first bit which is zero. This circuit has 40 parking bays switches.

The communication between the XBee sender and the XBee receiver has been fixed by X-CTU settings. When the correct working of the PCBs has been confirmed, the circuits have been integrated by the program in Java. The bays circuit has been connected with the PC by the XBee adapter, and the bays status transmission has been configured. The gate circuit is connected to the PC by USB. This USB transfers the RFID number to the PC in order to control its access. The integration between the hardware and software is shown in Fig.9.

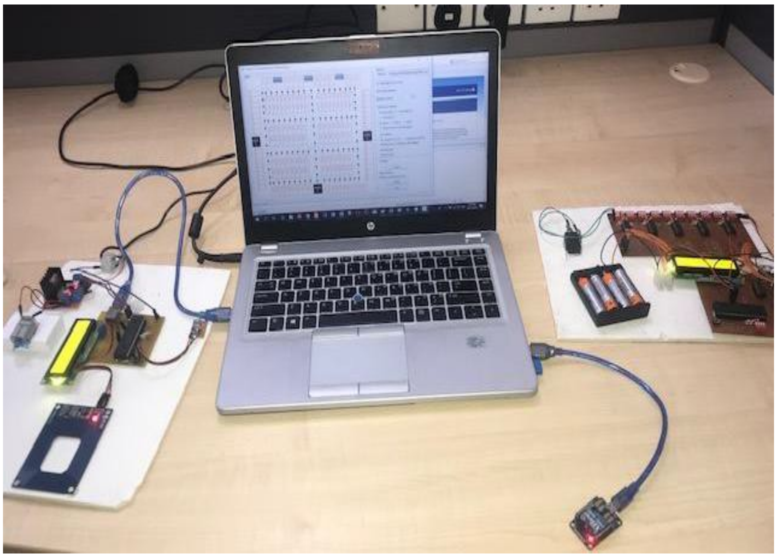

Fig. 9 - The system of hardware and software integration

3. Results and Discussion

To test and evaluate the proposed algorithms, a parking lot simulation was designed. The simulation parking lot is made up of three entrances, three parking gates, one hundred and sixty-two small parking bays, and fifty-four big parking bays. The simulation parking area is 3944 m^2 (58-meter × 68 meters) as shown in Fig. 10. The simulation was carried out using Java programming language.

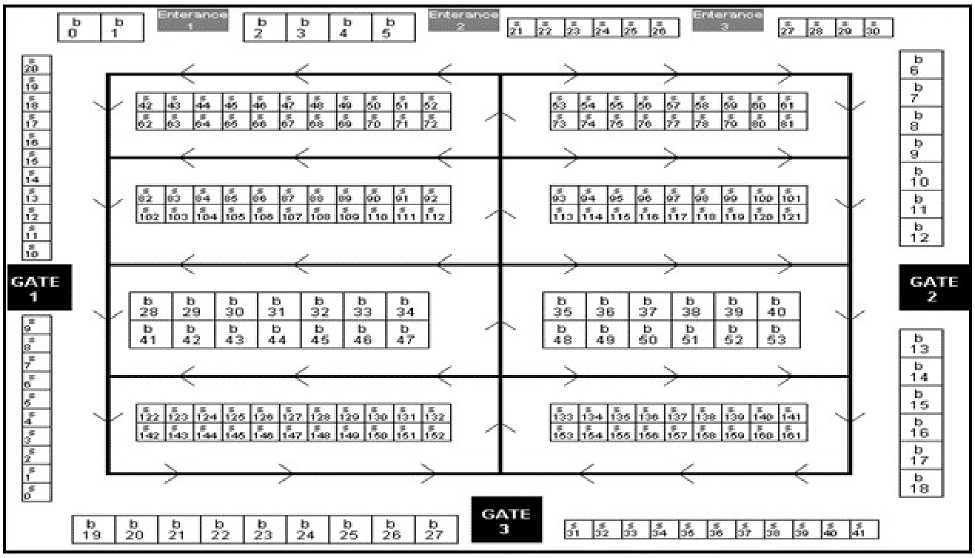

Fig. 10 - Layout of parking lot simulation

Additionally, three different scenarios were tested in the system in terms of the entrance and gate selected. Fig. 11 below shows the three different scenarios generated by the graphical user interface (GUI).

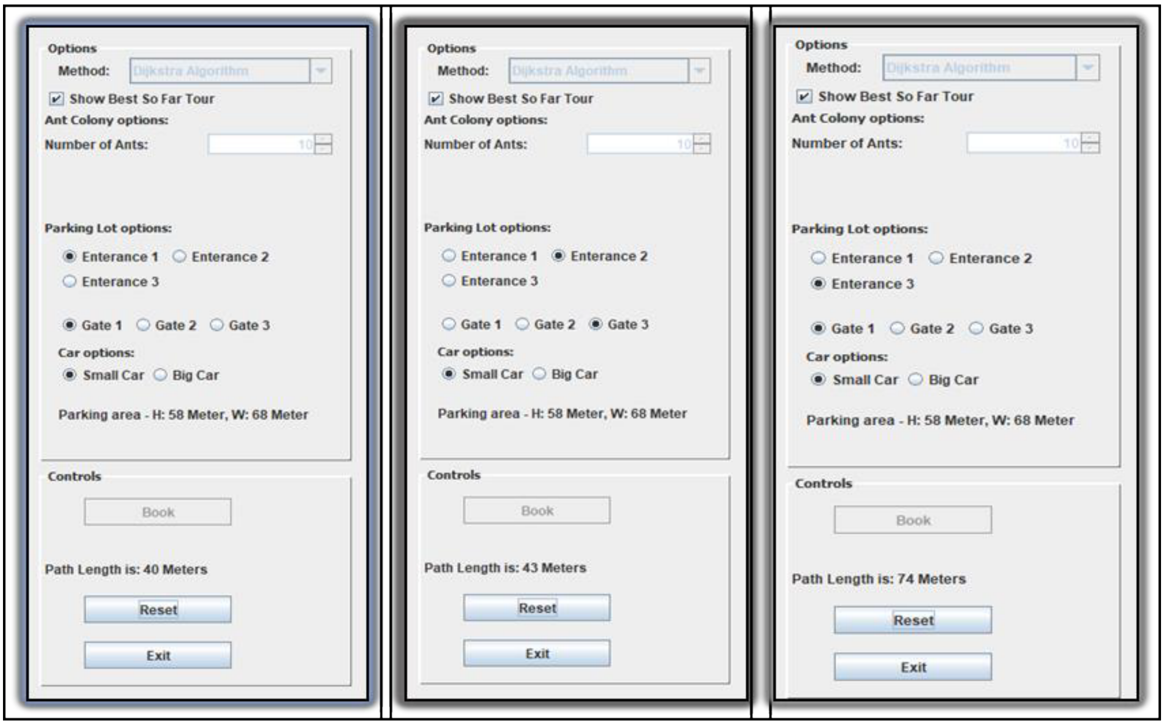

Fig. 11 - Graphical user interface (GUI)

The outcome of the proposed solution was compared with the result obtained from applying the traditional Dijkstra's algorithm. In terms of shortest path and accuracy, betterperformance isseenwith the application of the traditional Dijkstra's algorithm as shown in Fig. 12. The results show that the guidance path algorithm is more efficient and reliable compared to the traditional Dijkstra's algorithm. The time taken for drivers to reach the nearest parking bay is reduced, which would in turn reduce fuel consumption. Each scenario shows a clear improvement. In the first scenario, the improvement in the guidance path algorithm is 37.50%. The second scenario has achieved 10.81% in the shortest path while the third scenario has achieved 34.88% compared with the traditional Dijkstra's algorithm. These results are set out in Fig. 13 where the improvement brought about by the proposed algorithm can be clearly seen.

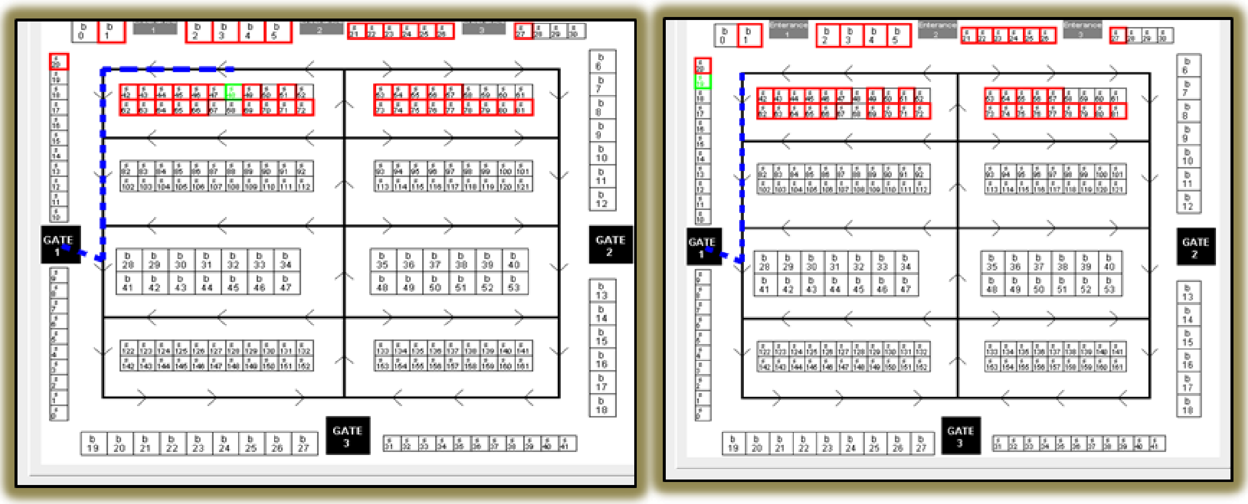

Fig. 12 - The comparison traveled path by Dijkstra's algorithm and the proposed algorithm

Scenario	Dijkstra algorithm	Proposed solution
Case 1	40	25
Case 2	74	66
Case 3	43	28

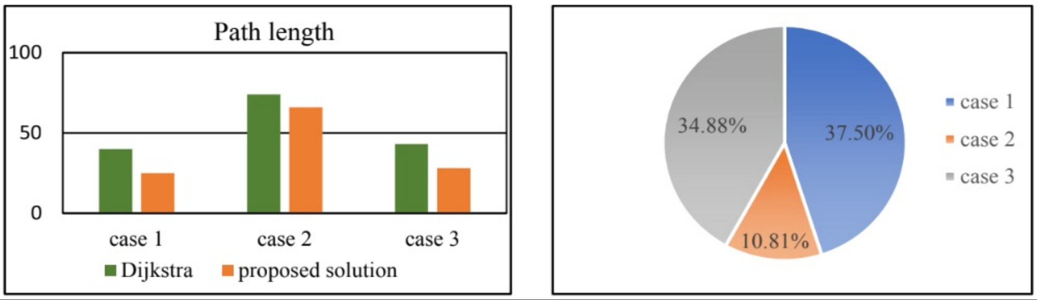

Fig. 13 - Improvements that can be seen in the three cases

4. Conclusion

This paper has sought to introduce an effective algorithm based on smart parking guidance algorithm by considering how representative decision factors are in parking environments. The first factor is the walking distance between a parking bay and the mall entrance. The second factor is the driving distance between the main gate and the parking lot. An effective layout has been proposed to provide a reliable parking system and to manage a car parking procedure that is more effective and practical as compared with a conventional layout. As can be seen from the simulation results, the system achieved the task of finding the optimal path compared to the proposed layout, hence proving its effectiveness.

Acknowledgments

The authors wish to acknowledge and thank the Department of Electrical and Electronic Engineering, University Putra Malaysia, for supporting this research.

References

[1] Abdulkader, M. M.S., Yuvraj G., & Tarek Y. E.(2015). Hybridized ant colony algorithm for the multi compartment vehicle routing problem. Applied Soft Computing, 37, 196-203.
[2] Abidi, S., Krichen, S. Alba, E. & Miguel, J. (2015). A new heuristic for solving the parking assignment problem. Procedia Computer Science, 60(2015), 312-21.
[3] Saleh Alija, A. (2015). Analysis of Djikstra's and A* algorithm to find the shortest path.Master Thesis, Universiti Tun Hussein Onn Malaysia.
[4] Cai, W., Dong Z., and Yongjie P. (2015). Implementation of Smart parking guidance system based on parking lots sensors networks. 2015 IEEE 16th International Conference on Communication Technology (ICCT), Hangzhou, China,19-24.
[5] Han, Y., Jiawen S., Meng W. & Guang Y. (2017). Optimization design and evaluation of parking route based on automatic assignment mechanism of parking lot. Advances in Mechanical Engineering, 9(7): 1-9.
[6] Jaafar, H, Zabidi, M. H., Che Soh, A. & Hoong, T. P. (2014). Intelligent guidance parking system using modified Dijkstra ' s algorithm. Journal of Engineering Science and Technology,August(2014), 131-139.
[7] Litman, T. (2018). Evaluating public transit benefits and costs. Best Practices Guidebook, Victoria Transport Policy Institute.
[8] Mahajan, K. K, & Agrawal, R.A.(2017). Keenly Intellective and automated parking technology." International Journal of Scientific Research in Computer Science, Engineering and Information Technology,2(2): 219-22.
[9] Walia, P., Banasal, B.N. & Harleen K. (2014). Comparison on the performance of Genetic algorithm and Ant Colony 0ptimization for solving the travelling salesman problem. International Journal of Advanced Research in Computer Science and Software Engineering 4(2), 891-96.
[10] Wang, H. , Zhang, F. & Chui, P. (2017). A parking lot induction method based on Dijkstra algorithm. 2017 Chinese Automation Congress (CAC), Jinan, China, 20-22 Oct. 2017.
[11] Wang, M., Dong, H., Li, X. & Song, L.(2017). Novel parking system designed for smart cities. 2017 Chinese Automation Congress (CAC), Jinan, China, 20-22 Oct. 2017.
[12] Zhu, G., & Sam K. (2010). Gbest-guided artificial Bee Colony algorithm for numerical function optimization. Applied Mathematics and Computation, 217(7), 3166-3173.