Research on Dynamic Energy Saving Path Planning Algorithm based on Iov

Yang Xu [a], Yibin Zhang [b]

School of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[a]xuyang@cqupt.edu.cn, [b]304385733@qq.com

Abstract

Path planning is an important way to alleviate traffic congestion, but the traditional path planning algorithm can only calculate the shortest distance path. Meanwhile, the traffic information is dynamic and changes over time, so the initial optimal path is likely to be invalid when traffic conditions change. In this paper, a dynamic energy saving path planning algorithm (DESPP) based on IoV (Internet of Vehicles) is proposed, which can plan the energy saving path for driver dynamically. An extended Dijkstra algorithm is elaborated, by setting the fuel consumption for the calculation of weight, changing the storage structure of road data, and using the improved quick sort algorithm to sort the weights, so that fast search to the adjacent node can be realized, then the dynamic energy saving path planning algorithm (DESPP) will be obtained. The simulation results show that DESPP algorithm can reduce about 15.19% of the total fuel consumption compared with the traditional Dijkstra algorithm. The DESPP algorithm can be applied in finding energy saving path for navigation system, which is more suitable for the actual need of drivers.

Keywords

Path Planning, Energy Saving Path, Internet of Vehicles, Dijkstra Algorithm.

1. Introduction

With the quick development of urban traffic, traffic congestion has become a serious social problem, which directly leads to increasing fuel consumption and serious environmental pollution. Path planning is to find a path from the starting point to the terminal point in a particular network according to some evaluation standards (such as the shortest or the fastest, etc), which is an important way to solve traffic congestion and save fuel consumption. Traditional path planning algorithm can only solve the static shortest path problem [1,2], due to not considering the real-time road conditions such as traffic jams, accidents, etc., the shortest path is not the optimal path. Therefore, many scholars have carried out extensive research on the path planning problem [3~6], and made some progress, but mainly focused on solving the shortest path or the shortest time, the research on energy saving path planning is very little. With the consideration of reducing fuel consumption, this paper based on Internet of Vehicles (IoV) environment, access to the real-time traffic information, changes the traditional problem that find the shortest path into find the most efficient path. Taking into account the two conditions of the vehicle: Travel on the road segments and delay at intersections, we improved the traditional Dijkstra algorithm, and proposed the dynamic energy saving path planning algorithm(DESPP), simulation experiment is designed to verify the effectiveness of the algorithm.

2. Real-time traffic information collection

DSRC is regarded as the communication protocol between OBU (On Board Unit) and RSU (Road Side Unit) in this paper. Through the data fusion about vehicles by RSU, we can obtain the road segment’s condition. The RSU at intersections can read the time delay of traffic lights. By 3G/4G, RSU can update the traffic information to ISC(Information Service Center) periodically.

2.1 Vehicle-to-Infrastructure cooperation and calculation of road segment average speed

While the vehicle enters the range of RSU, the OBU transfers the vehicles' information to RSU. Thus, the data interaction of vehicles and road can be achieved. The message format of the vehicle's speed is illustrated in Table 1.

Table 1. The message format of the vehicle's speed

road segment ID	vehicle ID	vehicle speed	time stamp	message type

In above, the *road segment ID* is the road segment which the vehicle runs. The vehicle can only be identified by the *vehicle ID*. The *vehicle speed* is the vehicle's instantaneous velocity. The message type is *vehicle_speed*. A two-way road contains information of two sections' ID, the single road only has information of one section' ID. The RSU can storage information like road segment ID, road segment's length, road segment's coordinate and so on. OBU can get the road segment ID according to the running direction of vehicle and information of location. The RSU at the intersection can read the traffic lights' information directly through connecting whit the traffic lights control system.

2.2 OBU/RSU-to-ISC cooperation

RSU can update the traffic information to ISC (Information Service Center) periodically through 3G/4G. The traffic information includes rode segment information and the time delay at intersections.

Thus, the real-time traffic information can be storage in ISC. In particular, all the data of RSU are updated synchronically in order to insure the effectiveness of the real-time traffic information. The format of the traffic data between RSU and ISC are illustrated in Table 2 and Table 3.

Table 2. The message format of road segment state

RSU ID	road segment ID	average speed	passable	time stamp	message type

Table 3. The message format of intersection's time delay

RSU ID	intersection ID	direction 1	…	direction N	time stamp	message type

In above, *RSU ID* is the RSU's unique number at the road segment or intersection. The *road segment ID* is the number of road segment where the RSU at. For single road, set the corresponding direction of the road segment ID is unpassable. The *average speed* is the average speed of the road segment. The *passable* indicates whether the vehicle is allowed to pass through. There are two conditions: passable or unpassable. The *time stamp* records the send time of the message. The message type is *road_speed or intersection_delay*. The *intersection ID* is the intersection's unique number. The *direction 1*,…, *direction N* means the time delay of all directions at intersection. If there is turning restriction, then the corresponding time delay can be set as ∞.

OBU uses 3G/4G to connect to ISC, by using Socket as the data transfer communication. The lightweight data transfer format JSON (JavaScript object notation) [7] is used in this paper to realize the traffic information transmission between OBU and ISC. A JSON file can contain multiple similar format traffic information.

The JSON data of a road segment's traffic condition:

```
{
 “road_Id”:12                    //road segment ID is 12
 “ave_speed”:43                  //the average speed of the road segment is 43km/h
 “passable”:“1”                  //0 means unpassable, 1 means passable
}
```

The JSON data of an intersection's traffic condition:

```
{
     “ver_Id”:    3                          // intersection ID is 3
     “latlong”：106.517828,29.604941   //the latitude and longitude of the intersection
```

"lastTime1":10 //the delay time of direction 1 (s)

"lastTime2":10 //the delay time of direction 2 (s)

"lastTime3":10 //the delay time of direction 3 (s)

……

}

2.3 The functions of ISC

The ISC has two main functions: (1) store the real-time traffic information; (2) send the latest traffic information to OBU. When ISC receives the information sent by RSU, it will updates the road network information database. The information send from RSU are as follows:

road segment ID, intersection ID;

average speed of road segment, delay time of intersection;

whether the road can be passed (passable, unpassable);

time stamp of the current traffic information data.

Then, OBU can send a request to ISC to get the latest traffic information through 3G/4G wireless communication technology.

3. Algorithm design

In this paper, the dynamic energy saving path planning problem is resolved based on real time traffic information. The corresponding algorithm is designed by modifying the calculation weight of traditional Dijkstra algorithm.

3.1 Set fuel consumption as the calculation weight

In this paper, the fuel consumption is used as the weight. Comprehensive consideration of the two main aspects: Vehicles running on the road segment and idling at the intersection. Assume that the vehicle is traveling at the average speed of the road segment, the amount of fuel consumed in road segment $<i, j>$ can be calculated by its length L_i and the and the oil consumption per kilometer $O(vs_{ij}^{t})$, which is a function of its average driving speed vs_{ij}^{t} for time zone *t*, as the shown in Fig.1[7]. The idling fuel consumption at the intersection of the vehicle can be calculated by idle time and idle fuel consumption rate.

The fuel consumption in road segment $< v_i, v_j >$:

$$O_{ij}(t) = L_{ij} \times O(vs_{ij}^{t}) \tag{1}$$

Where L_{ij} is the length of road segment $< v_i, v_j >$, vs_{ij}^{t} is the average speed of road segment $< v_i, v_j >$ at time zone *t*.

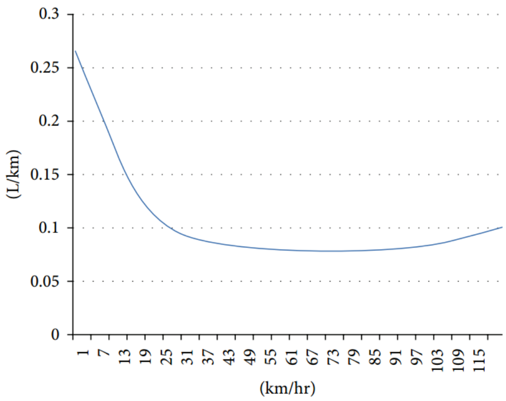

Fig. 1 An example of the relationship between the driving speed and the corresponding fuel consumption per kilometer.

Idle fuel consumption at the intersection i is

$$O_{di}^{t} = T_i^t \times \tau \quad (2)$$

Where T_i^t is the time delay of the intersection i; τ is the idle fuel consumption rate of the vehicle, it can be measured experimentally.

At time zone t, the total fuel consumption of vehicle passing through road segment$< v_i, v_j >$ is

$$W_{ij} = O_{di}^{t} + O_{ij}(t) = T_i^t \times \tau + L_{ij} \times O(vs_{ij}^{t}) \quad (3)$$

Obviously, by using the fuel consumption as the weight can meet the consistency principle.

3.2 Storage of road network topology relation

Traditional Dijkstra algorithm uses adjacency matrix store road network's topology. For the urban road network, the nodes is huge, it needs to open N×N (N is node number) the storage space. The space complexity is $O(N^2)$. There will be a large number of 0 and ∞ elements in the adjacency matrix, computational efficiency and storage efficiency will be affected. Thus, we uses the adjacency list to store the network topology, at the same time, use an auxiliary bidirectional circulation linked list to store the adjacent node that remains to be sorted.

3.3 Fast realization of weight sorting

The key of Dijkstra algorithm is to select the smallest weight segmental arc from the nodes. In the traditional Dijkstra algorithm, adjacency node disorderly storage in an unordered list, if the weight of its adjacent node is not made any treatment, it needs to scan cycle several times to get the minimum weight arcs, this will reduce the efficiency of the algorithm. If the related adjacent node are sorted according to the weight, we can obtain the node of conform requirements in one cycle, which can improve the efficiency of the algorithm. In this paper, the weights are sorted by the improved quick sorting algorithm, then the optimization of related adjacent node is achieved, and the search time of the algorithm is reduced.

The basic idea of improved quick sort algorithm: supposed to sort the weight array is *W*[0]……*W*[*n*-1]，the mean value of the maximum element and the smallest element of the array to be sorted is *Ave*. Set *Ave* as the middle element, *k* and *l* are the first and last elements, *k=0, l=N-1*. Search from *l* to the front (*l=l-1*), find the first value *W*[*l*] that less than *Ave*, meanwhile, search from *k* to the back (*k=k+1*), find the first value *W*[*k*] that greater than *Ave*, exchange value of *W*[*l*] and *W*[*k*]. Repeat the above steps until k=l. Then, use recursive method to quickly sort the sub sequences, when the length of the array is less than a certain value, the recursion is stopped. Although such a quick sort after the end did not achieve the desired goal, but it has a certain initial sequence, can be used direct insertion sort algorithm. At the same time, using the method of set up signs to have ordered or is the same data sorting judgments. If in the process of scanning, there is no exchange of data occurred, indicating that the sequence has been ordered, so that the completion of the order.

Based on the above, the improve steps are as follows: Let *Z* be the permanent mark node collection, *M* is the adjacency node collection of the nodes that has marked; *N* stores the sorting nodes; *dist*[*t*] is the shortest path length from source *s* to the node *t*; w_t is the shortest path's forerunner node from *s* to the *t*; d_{ij} is the distance from node *i* to node *j*.

1) Initialization. *Z*=$\{s\}$, *N*=*M*[*s*], $dist[t] = d_{st}, (t \in M)$, otherwise $dist[t] = \infty, (t \notin M)$; $w_t = s$.

The node in *M*[*s*] stores *N*[] and carry on sorting, takes the heap top node *p*, if $d_{sp} = \min_{t\in M[s]} d_{st}$, then $Z = Z \cup \{p\}$.

Renew the adjacency node collection near *p*, *N*[]=*M*[*p*]-*Z*; $dist[t] = \min_{t\in M[p]-Z}\{d_{st}, d_{sp} + d_{pt}\}$, if $d_{st} > d_{sp} + d_{pt}$, then the forerunner node of w_t is *p*.

Search *M*[*Z*]-*Z*, and store in *N*[], designate the minimum value of *dist*[*t*] in *M*[*Z*]-*Z*, and store in *Z*, $dist[p] = \min_{t\in M[z]-z} dist[t]$， $Z = Z \cup \{p\}$.

If all nodes have marked, then the algorithm terminate, otherwise changes over to the second step and continues to carry out, until all nodes gather in *Z*.

Based on the above content, the dynamic energy saving path planning algorithm (DESPP) based on real-time traffic information is obtained, as shown in Fig.2.

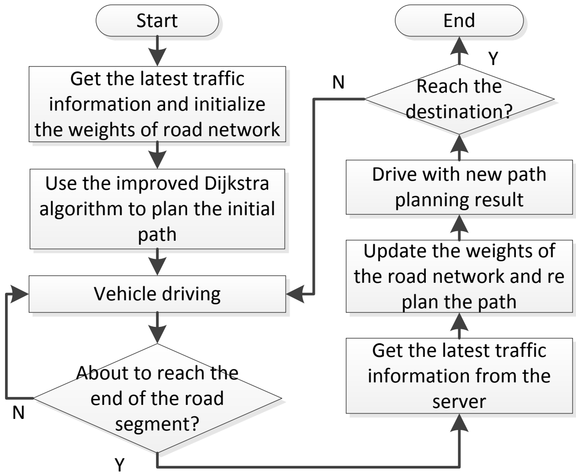

Fig. 2 The flow chart of dynamic energy saving path planning algorithm

4. Simulation experiment

4.1 Experimental design

The experimental hardware environment: Lenovo Core(TM) i5-4590, 3.30GHz, RAM capacity 4GB, Windows10 64 bit operating system. All programs are written in Java, the Java virtual machine version is JDK1.8.

In order to verify the correctness and validity of the algorithm, according to a city road network, a part of the road network is selected and simplified, which contains 72 nodes, 249 edges. The dynamic road network structure diagram is shown in Fig.3. Due to the limitation of the experimental environment, the real-time traffic information is generated by the program simulation, the number of time interval is set as 500, T = {1, 2, ... , 500}, and each time interval is 30s. Considering that the road segment is stable in most time, we only choose 100 intervals to change the road impedance. Meanwhile, the road resistance coefficient is 1% chance becomes infinite, indicating that this road impassable. Set the maximum speed of the road network is 70km/h, the idle fuel consumption rate of vehicle is 0.5L/h, and the intersection delay time is 5s. 30 random node pairs are selected to carry out the experiment, in order to make the experimental results more scientific, we will keep the change process of traffic information in the files, after each test road network will be in accordance with the same process changes.

4.2 Simulation Results

We compare the effect of traditional Dijkstra (TD) algorithm and the dynamic energy saving path planning (DESPP) algorithm by fuel consumption and travel distance. Fig.4 shows that, compared with the traditional path planning algorithm, the DESPP algorithm can effectively reduce the fuel consumption of vehicles. In the 30 groups, 28 groups of vehicle fuel consumption were significantly reduced, after the use of DESPP algorithm, the total fuel consumption savings of 15.19%. Meanwhile, it is found that the group 15 and the group 27 are not reduced, but increased, this is because the DESPP algorithm can only obtain the optimal results at the present time. For global terms, according to the real-time traffic information frequently changing route leads to the increase of the distance of the vehicle, as shown in Fig.5, after adopting DESPP algorithm, the total distance of the vehicle increased 7.44% than traditional Dijkstra algorithm, there may be a case of fuel consumption does not fall, the solution to this problem is to predict the road conditions in a short time, which can be used as the next step in the research work. Overall, the experimental results show that after adopting DESPP algorithm, 93.3% of vehicle driving fuel consumption decreased, this shows that the path planning method proposed in this article is feasible, which can effectively reduce the waste of fuel consumption and pollution emissions, and is more suitable for the actual needs of users.

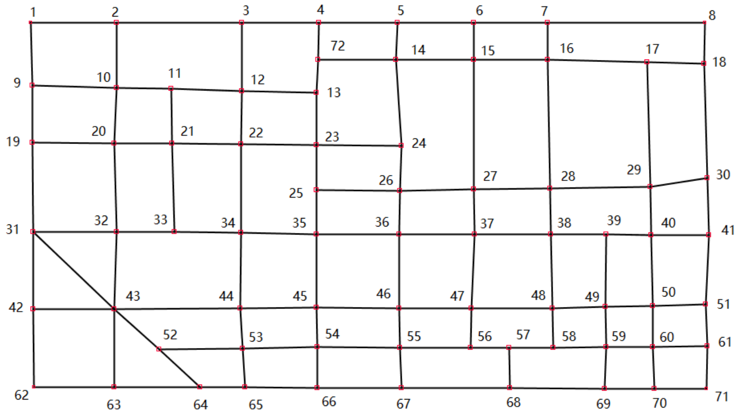

Fig. 3 The dynamic road network structure

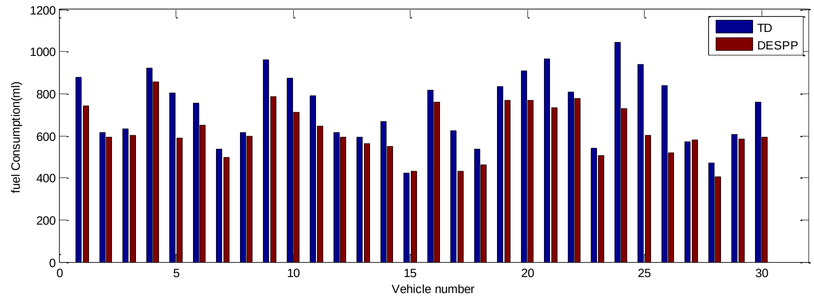

Fig. 4 The fuel consumption using TD algorithm vs. DESPP algorithm

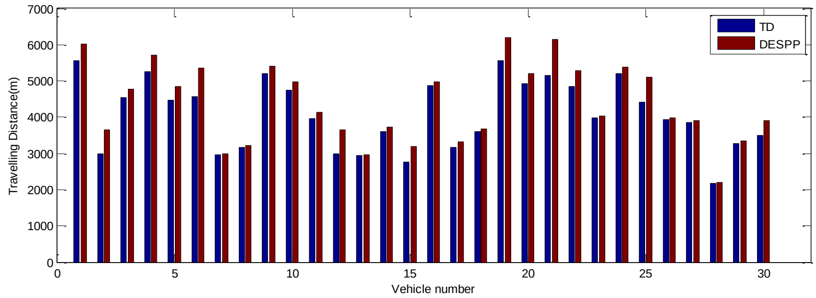

Fig. 5 The travelling distance using TD algorithm vs. DESPP algorithm

4.3 Application implementation

The above algorithm is implemented in IoVs platform. The dynamic energy saving navigation app is designed and implemented on the Google Android 4.1 operation system by using the open source map OpenStreetMap, which a regional map of Chongqing, China is selected. After setting the start point and end point, then the initial path is obtained, the vehicle drive on this path, as shown in Fig.6. During the driving process, the vehicle receives the real-time traffic information, and the current position is the start point, then re plan path, as shown in Fig.7.

Fig. 6 Initial plan path Fig. 7 Replan path after receiving the traffic information

5. Conclusion

This paper presents a dynamic energy saving path planning algorithm based on real-time traffic information, a dynamic road network model is designed to reflect the actual traffic situation, and the traditional Dijkstra algorithm is improved. The experimental results show that the proposed algorithm can provide a more energy efficient driving route, and effectively reduce the fuel consumption.

References

[1] SUN Li-jun, HU Xiang-pei, WANG Zheng. Reviews on Vehicle Routing Problem and Its Solution Methods[J]. Systems Engineering, 2006, 24(11):31-37.

[2] Dijkstra E W. A Note on Two Problems in Connection with Graphs[J]. Numerische Mathematics, 1959, 1(1):269--271.

[3] MA Renli, GUAN Zhengxi. Summation for Present Situation and Future Development of Path Planning Technology[J]. Modern Machinery, 2008(3):22-24.

[4] XIANG Dong-mei, CHEN Shu-hui. Research on the Shortest Time Path Planning Based on Dynamic Traffic[J]. Microcomputer Information, 2012,28(9):317-319.

[5] Chu K, Kim J, Jo K, et al. Real-time path planning of autonomous vehicles for unstructured road navigation[J]. International Journal of Automotive Technology, 2015, 16(4):653-668.

[6] Kong X, Yang J. Path Planning with Information on Travel Time Reliability[C]// Cota International Conference of Transportation Professionals. 2016.

[7] Chang I C, Tai H T, Yeh F H, et al. A VANET-Based Route Planning Algorithm for Travelling Time- and Energy-Efficient GPS Navigation App[J]. International Journal of Distributed Sensor Networks, 2013, 9(3):310-325.