

## Research Article

# Research on Large-Scale Road Network Partition and Route Search Method Combined with Traveler Preferences

De-Xin Yu,<sup>1</sup> Zhao-Sheng Yang,<sup>1</sup> Yao Yu,<sup>2</sup> and Xiu-Rong Jiang<sup>2</sup>

<sup>1</sup> State Key Laboratory of Automobile Simulation and Control, School of Traffic, Jilin University, Changchun 130025, China

<sup>2</sup> School of Traffic, Jilin University, Changchun 130025, China

Correspondence should be addressed to Yao Yu; [yuy.jlu@foxmail.com](mailto:yuy.jlu@foxmail.com)

Received 5 September 2013; Accepted 24 October 2013

Academic Editor: Wuhong Wang

Copyright © 2013 De-Xin Yu 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.

Combined with improved Pallottino parallel algorithm, this paper proposes a large-scale route search method, which considers travelers' route choice preferences. And urban road network is decomposed into multilayers effectively. Utilizing generalized travel time as road impedance function, the method builds a new multilayer and multitasking road network data storage structure with object-oriented class definition. Then, the proposed path search algorithm is verified by using the real road network of Guangzhou city as an example. By the sensitive experiments, we make a comparative analysis of the proposed path search method with the current advanced optimal path algorithms. The results demonstrate that the proposed method can increase the road network search efficiency by more than 16% under different search proportion requests, node numbers, and computing process numbers, respectively. Therefore, this method is a great breakthrough in the guidance field of urban road network.

## 1. Introduction

Path optimization is one of the most key issues in traffic flow guidance filed, and it is mainly used for vehicle positioning and system navigation. As we know, travelers need some route information to assist them to get to their destinations. So path guidance technology has been developed which not only can provide travel routes but also relieve the traffic congestion.

The optimal path search theory [1] is the core of the path optimization. There are a lot of relevant researches on the path search algorithms and classic calculation methods around the world. For example, Dijkstra algorithm, A\* algorithm, and ant colony algorithm are mature algorithms. But all the above algorithms are generally applied to small-scale road network. Currently, the advanced search path algorithms were improved via parallel computing [2] in order to solve the large network path problem (LNPP). So some optimized algorithms such as Dijkstra algorithm with approximation barrel structure, Pallottino algorithm, and Dijkstra algorithm [3] with quad heap structure have been proposed. These mentioned improved path search algorithms can meet the needs of search speed of large-scale road network path

optimization. However, it rarely considers the travelers' drive habits and preferences, which leads to the path optimization results which are often not accepted by travelers. That is, the travelers' obey rate will drop, and guidance effects cannot be presented.

In view of above-mentioned insufficiency, we consider the drive habits and preferences in order to ensure implementation of the real-time path optimization of large-scale road network. And we construct a traffic network simulation system and a multilayer and multitasking network data storage structure in which parallel data is shared according to the traffic network data [4, 5]. We also adequately analyze the interactional relationship between road network decomposition and shortest path calculation. And then a Hierarchical Pallottino parallel search algorithm model has been proposed in accordance with the data storage and parallel computing technology. The proposed model can dynamically adjust the search level in terms of travel distance. And the multilayer Pallottino search algorithm is run separately in subnets, achieving real-time interaction within boundary nodes via information transmission among the adjacent subnets. Consequently, a global optimal solution can be obtained.

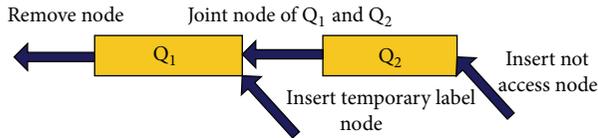


FIGURE 1: Schematic diagram of deque data structure.

## 2. A Hierarchical Pallottino Parallel Search Algorithm Model Based on the Data Storage

Pallottino algorithm adopts priority deque structure to calculate the optimal path. Deque algorithm [6–8] is a combination of two first-in first-out (FIFO) queues,  $Q_1$  and  $Q_2$ .  $Q_2$  has a higher priority than  $Q_1$ . As for a node, if its state numeral is not accessed, then this indicates it has a higher priority and it will enter the candidate node set from the end of  $Q_2$ ; if its state numeral is temporary, then this indicates it has a lower priority and the candidate node will enter from the end of  $Q_1$ . During the process of route search, the nodes of  $Q_2$  are always firstly searched for achieving the two-queue search mechanism and the fast iterative correction of node labeling [9]. When removing nodes from the candidate node set, the deque needs to be detected whether it is empty or not. If the queue  $Q_1$  is not empty, then it will delete nodes from the front of  $Q_1$ ; if the queue  $Q_1$  is empty, it will continue to detect whether the queue  $Q_2$  is empty or not. If the queue  $Q_2$  is not empty, then it will delete nodes from the front of  $Q_2$ . Operate the above cycle after deleting a node. The operation does not end until the queue  $Q_2$  is empty. The schematic diagram of deque data structure is shown in Figure 1.

In daily life, a sizable number of drivers prefer to select the main sections to travel. However, when driving for a near distance, they may not select the main road [10]. Therefore, a multilayer road network data storage means is utilized to improve the Pallottino search algorithm, and a Hierarchical Pallottino parallel search algorithm is put forward. When the driver travels for a far distance, the proposed algorithm should be used to search the main road node layer. When the driver travels for a near distance (when the number of the cells between the origin and destination is less than or equal to 1, it is regarded as “the travel distance is near”), it needs to search both the main road node layer and secondary road node layer.

### 2.1. Urban Road Network Partition and Optimization

**2.1.1. Urban Road Parallel Data Structure.** The data structure of urban road network [11] is a key point of solving path problem, which directly determines the difficulty and timeliness of path algorithm procedures. When organizing network data, it not only invokes conveniently and fast, but also reduces the occupation of memory resources and expresses clearly in order to facilitate data calls and program checks. In terms of urban road traffic network, its corresponding data structure has the following three characteristics: (a) the data is so large, and the program needs to call and access it frequently; (b) the topological relations of road network are clear and the

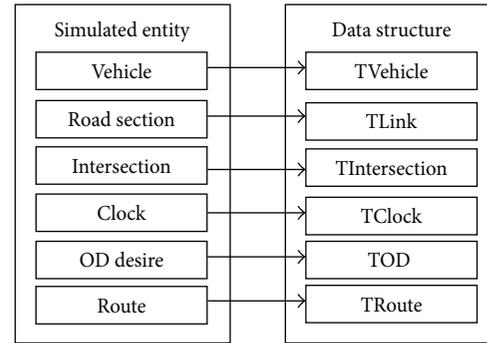


FIGURE 2: Traffic network simulation systems and shared parallel data structure.

relationship between the nodes and the sections is fixed; (c) there is concurrency when the data structure is accessed by the program.

The basic element of traffic network data system is composed of car, section, route, intersection, clock, and so forth. Furthermore, every traffic network element and road network structure all can be considered as a single object. This paper will adopt C++ to set up traffic network simulation system and shared parallel data structure through object-oriented class definition (shown in Figure 2), to meet the demand of the following Hierarchical Pallottino parallel search algorithm.

The above different types of member variables and member properties are shown in Algorithm 1.

After defining various types of entities in network structure, for every type of entity, a single object is just one unit of traffic network objects set. Therefore, multitasking concurrent accessing data could be implemented through establishing shared data structures so as to improve the speed of accessing data, satisfying the timeliness requirement of large-scale road network guidance.

**2.1.2. Network Structure Optimization Considering Drivers' Characteristics.** As we know, drivers often choose the road they are familiar with or the urban trunk roads as their priority travel routes. In view of the characteristics, we should give priority to search and calculate this layer of the road network to ensure the speed of calculation, which is particularly important for the large-scale road network.

In this paper, urban road network is divided into multiple levels [12, 13]. The roads preferred by drivers are taken as main roads to constitute the main road network which divides the urban road network into small zones. In addition, a series of minor road networks are constituted by the road network of every zone (like zone K, M, and L in Figure 3). With regard to doing route optimization, the final route optimization network is made up of the minor road network within the scope of drivers' origin zone or destination zone and the main road network. Thus searching road network will be simplified and the speed of searching improved.

As for the road impedance function, there are five popular items including traveling time, driving distance, congestion degree, road quality, and comprehensive cost [14–16]. In view

```

Deque<TVehicle>vehiclist // Define the vehicle queues;
Deque<TLink>linklist // Define the road section queues;
Deque<TIntersection>intersectionlist // Define the intersection queues;
Deque<TOD>ODlist // Define the OD queues;
Deque<TRoute>Routelist // Define the Route queues;
    
```

ALGORITHM 1: Definitions of road network structure and queue structure.

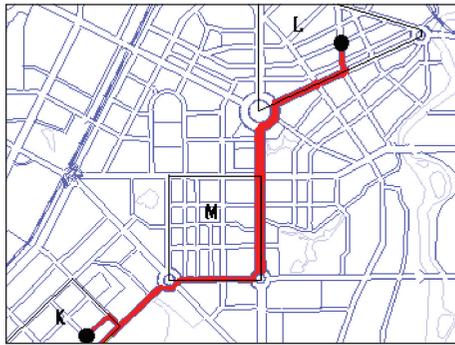


FIGURE 3: Multilevels of urban road network.

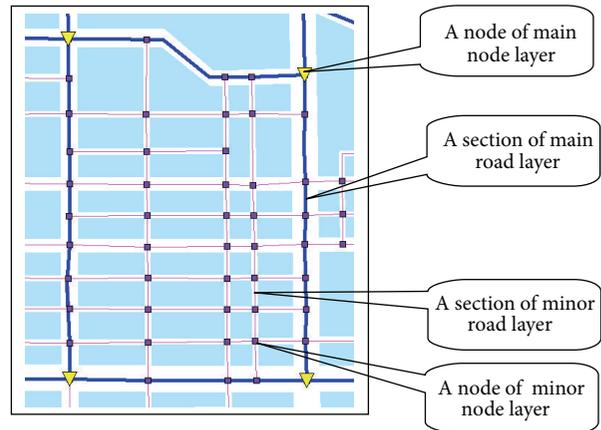


FIGURE 4: Layered search thought diagram.

of above ideas, real-time dynamic generalized travel time could be defined as road impedance (1) in this paper. There are several ways to determine the road impedance data, which reflects different user requirements and system control strategies. Considering the circumstances, the final route is not the absolute optimal one in accordance with generalized road impedance in the whole road network, but the result of quasi-drivers changing directions based on optimization principle.

Consider the following:

$$C_i(k) = t_i(k). \tag{1}$$

The formula above is the most common dynamic road impedance form, which indicates the average time that vehicles take on the  $k$ th road section, containing travel time and parking delay time (including intersection delay).

In order to avoid common problems caused by the road network partition, this paper adopts the thought of layered optimization. Firstly, the main roads and the minor roads are separated into two layers. The intersections of the main roads are defined as main nodes and the intersections of the minor roads are defined as minor nodes (Figure 4). The triangle symbol indicates the node of the main nodes layer and its number is equal to the number of intersections of the main roads in the whole map. According to the network layered perspective, removing the intersections of the main roads and the minor roads will greatly reduce the time complexity in the process of route calculation, thereby the calculation speed will be improved.

In order to solve the problem that drivers may not choose the main roads when the driving distance is relatively short, this paper adopts the calculating method of neighboring zones union. If the origin and the destination are in the same

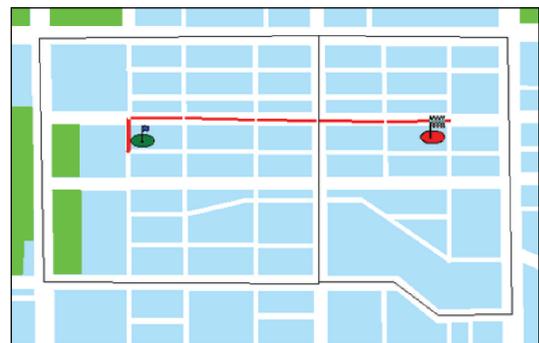


FIGURE 5: Neighboring zones incorporated principle diagram.

or the neighboring zones, this paper will take the nodes of the same or the neighboring zones as a union and search the union during the process of calculation. In principle, if there are excessive nodes in zone, it will affect the speed of route calculation. Thus in the process of zone division, the scope of zones should be reduced as much as possible in order to reduce the number of nodes. Figure 5 shows the route calculation results incorporated into the principle.

**2.2. The Hierarchical Pallottino Parallel Search Algorithm Model.** According to the above mentioned road network data storage and network decomposition method, a large-scale network will be decomposed into several subnetworks [17] whose number is the same as that of the processor. We use the Hierarchical Pallottino search algorithm in every processor. At this moment, it cannot obtain the information of adjacent

nodes of boundary nodes, so the global optimal solution cannot be obtained. Therefore, while using the Hierarchical Pallottino search algorithm for parallel computing, the paper uses the way of information transfer to send the boundary node's real-time information to the processor. And the processor receives the information sent from the adjacent subnet so as to complete the boundary node label iterative correction and obtain global optimal solution.

The specific calculation steps are as follows.

*Step 1.* According to the starting point  $O$  and the terminal point  $D$  position that user inputs, processor could analyze the road node layer and generate a road barrier matrix  $W$ . Define the node collection of cell  $i$  as  $P(i)$ , the node collection of the main road as  $PZ$ , and the road network node collection as  $P$ . Identify the starting point  $O$  and the terminal point  $D$  that user inputs and determine their cells  $k$ , and  $l$ , respectively. If the starting point and the terminal point are located in the same cell, then set  $P = P(k)$ ; if they are not in the same cell, then set  $P = PZ$  and incorporate the nodes of the cells  $k$  and  $l$  into  $P$ . If there is only one cell  $m$  between the starting point and the terminal point, then the nodes of cell  $m$  will be incorporated into  $P$ .

*Step 2.* Label each node  $i$  in the search network node collection. There are three numerals which are length numeral  $d_i$ , predecessor node numeral  $p_i$ , and current node status label  $s_i$ . Length numeral  $d_i$  indicates the accumulated weights from the starting point to the point  $i$ . Predecessor node numeral  $p_i$  is the previous direct point when the shortest distance between the starting point  $q$  and the point  $i$  obtains the maximum value. Current node status label  $s_i$  contains unvisited temporary labels and permanent labels.

*Step 3.* Define a message queue about the boundary node information for each subnetwork:  $M_x[h]$  ( $x = 1, 2, \dots, k, h < n/k$ ).

*Step 4.* Initialize the queues  $Q_1$  and  $Q_2$  when calculating the shortest path from arbitrary starting point  $q$  to all other nodes  $i$ , and we can set

$$d_q = 0 \quad d_i = +\infty \quad i \neq q. \quad (2)$$

*Step 5.* Judging the condition of queue  $Q$ , if it is empty, then go to Step 7, and if not, go to Step 6.

*Step 6.* Judge whether the formula (3) is established or not for each adjacent node  $j$  of node  $i$  in the nonempty queue  $Q$

$$d_j > d_i + \omega_{i,j}, \quad (3)$$

where  $\omega_{i,j}$  is the section weight between nodes  $i$  and  $j$ . If the formula (3) is established, the following formula will be set:

$$d_j = d_i + \omega_{i,j}. \quad (4)$$

If node  $j$  is not in the  $Q$  and  $s_j$  is not accessed,  $j$  will be inserted into the end of the  $Q_1$  and  $s_j$  will be updated to a temporary label. If node  $j$  is not in the  $Q$  and  $s_j$  is temporary mark state,  $j$  will be inserted into the end of the  $Q_2$ .

*Step 7.* In current subnet, observing each adjacent node  $r$  of node  $l$  and judging whether it is the shortest path, we can write

$$d_r > d_l + \omega_{l,r}. \quad (5)$$

If (5) is established, the node  $r$  will be added to its subnet message queue, and message queue will be sent to all adjacent subnetworks.

*Step 8.* Judge the condition of message queue. If it is empty, go to Step 10, and otherwise, go to Step 9.

*Step 9.* Each subnet sends a message queue to adjacent subnets and accepts feedback from other subnets, and each received adjacent node  $a$  which is adjacent to node  $t$  in the current process is calculated as follows:

$$d_a > d_t + \omega_{t,a}. \quad (6)$$

If (6) is established, then the following formula will be set:

$$d_a = d_t + \omega_{t,a}. \quad (7)$$

If node  $a$  is not in the  $Q$  and  $s_a$  is not accessed, then  $a$  will be inserted into the end of the  $Q_1$  and  $s_a$  will be updated to a temporary label. If node  $a$  is not in the  $Q$  and  $s_a$  is temporary mark state,  $a$  will be inserted into the end of the  $Q_2$ .

*Step 10.* Each processor sends the results of shortest path calculation to the host processor, and the algorithm ends.

### 3. Experimental Analysis

In order to verify the optimization results of the proposed Hierarchical Pallottino parallel search algorithm based on data storage, the paper uses C++ and OpenMP parallel programming method to design and develop parallel computing program of large-scale road network, utilizing client/server/(C/S) mode for setting up parallel computing platform. We test the proposed method based on real road network data of Guangzhou. This traffic network includes 11,489 nodes and 18,364 paths.

Experimental platform: Cluster of Workstations; CPU: Intel(R) Core(TM)2 Duo CPU E8600@3.33 GHZ Dual-Core Processor; Memory: 2 GB, each node Gigabit Switch: 4 sets.

*3.1. Road Network Decomposition.* Combined with the characteristics of drivers route choice, Guangzhou road network (central urban area as shown in Figure 6(a)) is divided into the following areas, and the dynamic topological relations of each subinterval (the corresponding dual connectivity graph) are shown in Figure 6(b).

According to the corresponding dual connectivity characteristics, it can be illustrated that the selected network, to some extent, possesses the small-world property, and road network has both smaller characteristic path length and larger clustering coefficient. Furthermore, fewer obstacles of road network are reflected from the small-world property, which has a good accessibility. According to the calculation

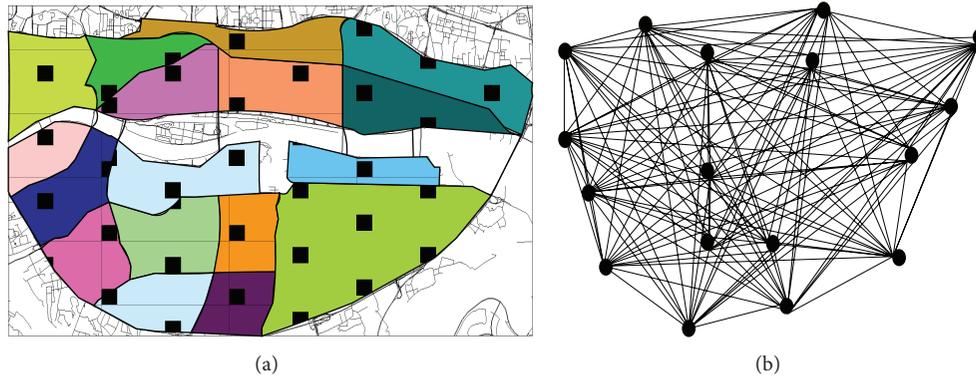


FIGURE 6: (a) Subarea of road network; (b) the dynamic topological relations.

results of network characteristic value and characteristic judgment conditions of small-world network, combined with the characteristics of the network and the connectivity of the intersection, we are convinced that it is the basis for evaluation of road network structure and the optimization strategy. Due to limited space, this paper is not to research this part. Decomposition of Guangzhou actual road network is shown in Figure 7.

**3.2. Shortest Path Parallel Computing Experiment.** On the traffic network, the paper separately applies the proposed Hierarchical Pallottino parallel search algorithm (HTWO\_Q), Dijkstra algorithm based on approximate barrel structure (DIKBA), Pallottino algorithm (TWO\_Q), and Dijkstra algorithm based on improved quad-stack structure (DIKQH) to calculate shortest path search time within all pairs based on the weights of generalized travel time.

In order to ensure the executive effectiveness of algorithms, the paper adopts both undecomposed and decomposed networks to analyze algorithms. In the undecomposed network conditions, we separately compare them in different conditions (i.e., 1:1, 1:N, and N:N) and search results are shown in Table 1.

Table 1 shows that the search speeds of the proposed HTWO\_Q algorithm and TWO\_Q algorithm are basically equivalent in the standard network. In 1:1 search, the computation times of DIKQH and DIKBA algorithms are superior to the proposed algorithm, but their calculation times are inferior to the HTWO\_Q algorithm and TWO\_Q algorithm in N:N search. Due to the advantages of the deque data extraction and storage, the path search times of HTWO\_Q and TWO\_Q algorithms are superior to the other two search algorithms by more than 5% in 1:N search.

Under the decomposed network conditions, we use the parallelization algorithms to calculate the search time by computing cluster under the processes 8 and 10. The calculative results are shown in Table 2.

From Table 2 we can see that the search time of the proposed HTWO\_Q algorithm is obviously superior to those of the other three algorithms in 1:N and N:N search, but it is inferior to DIKQH and DIKBA algorithms in 1:1 search. In the actual urban guidance system, 1:1 path search has

limitations, while 1:N and N:N search could satisfy the dynamic requirements of traveler and have more extensive application value.

During the path search, it is obvious that different numbers of network nodes will affect executive time of algorithm. In order to better illustrate the calculation time of these algorithms under different vertices, the algorithms' executive efficiency is compared and analyzed, respectively, in 4000, 8000, and the whole road network nodes. Results are shown in Figure 8. When road network data is stored in the way we have mentioned, the time of multiprocessor parallel computing optimal path (or global optimal solution) is closely related to the number of processes. Then, we compare the algorithms, respectively, using 6, 8, and 10 processes, and the results are shown in Figure 9.

Figure 8 demonstrates that the search speeds of HTWO\_Q and TWO\_Q are similar within 4000 nodes' network, which are lower than those of DIKBA and DIKQH. However, when searching in 8000 nodes and in the whole network, the proposed HTWO\_Q algorithm is superior to the other three algorithms, and its operating efficiency is 16.32% faster than that of the other three ones.

As shown in Figure 9, with the increasing of processes, the path-searching efficiency is not upward trend. By contrast, it firstly increases and then decreases, which is concerned with consumption of communication time when parallel algorithms are running. The result indicates that when process number is 8, the search efficiency of road network can be greatly enhanced.

## 4. Conclusions

Considering travelers' preferences, the purpose of this paper is to solve large-scale road network path optimization problem. With the way of object-oriented classified definition, a simulated traffic network system and parallel data structures have been constructed. To analyze the relationship between data storage and network decomposition, a multilayer Pallottino parallel search algorithm is proposed. Then, series of sensitive experiments are conducted to verify the algorithm under actual road network environment. Finally, the results show that the search time of the proposed optimization

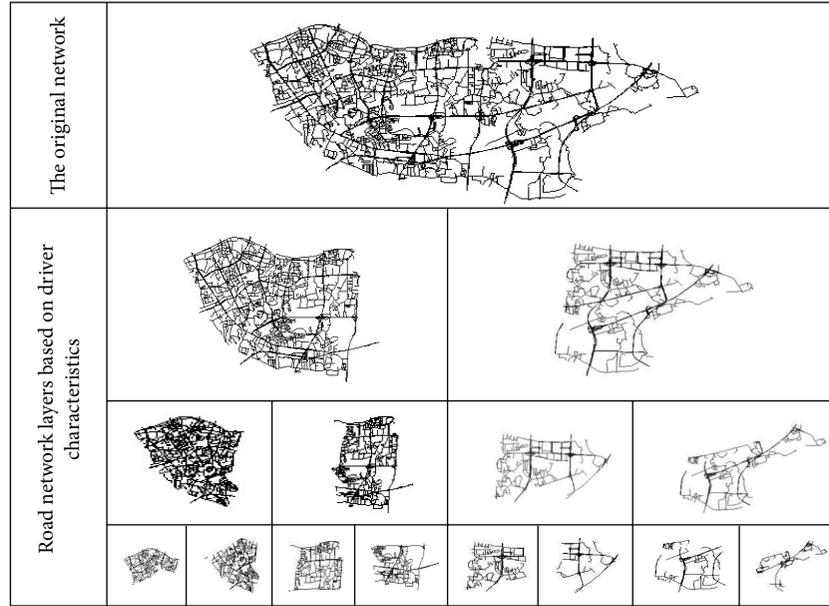


FIGURE 7: Decomposition of actual road network of Guangzhou.

TABLE 1: Comparison of path search times under initial network environment.

Computing time (s)	1:1 route searching	1:N route searching	N:N route searching
HTWO_Q	0.0097331000	0.0097331000	94.2922860000
DIKBA	0.0051053340	0.0103312830	100.4570000000
TWO_Q	0.0098270450	0.0098270450	95.3031400000
DIKQH	0.0050473020	0.0110842745	103.4130000000

TABLE 2: Comparison of path search times under Hierarchical network environment.

Computing time ( $10^{-3}$ s)	1:1		1:N		N:N	
	8	10	8	10	8	10
HTWO_Q	0.94271	1.00325	0.94263	1.00399	80.3311	83.6300
DIKBA	0.71433	0.91125	1.30764	1.38241	153.296	168.117
TWO_Q	1.15031	1.20301	1.14998	1.21334	100.775	119.068
DIKQH	0.807979	0.91966	1.58446	1.78677	174.210	195.429

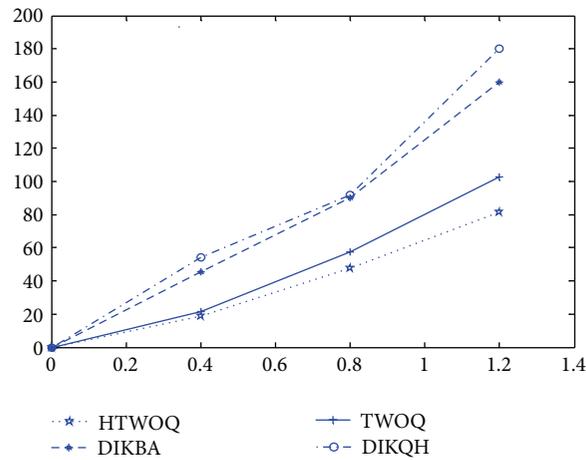


FIGURE 8: Computation times comparison of the four algorithms under N:N search pattern. The x-axis represents numbers of vertices (unit: thousand) and the y-axis represents computed times (unit: ms).

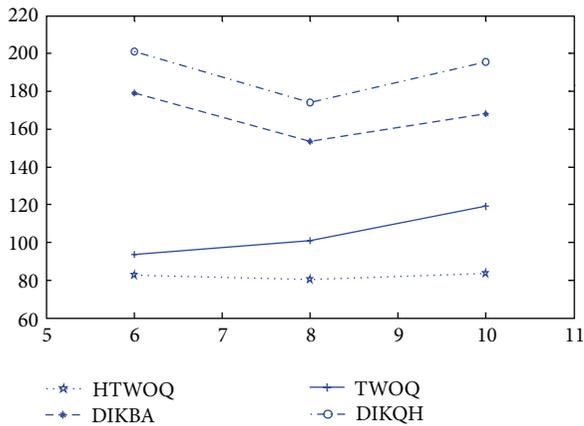


FIGURE 9: Computation times comparison of the four algorithms under multiple processes. The  $x$ -axis represents numbers of processes and the  $y$ -axis represents computed times (unit: ms).

method is significantly better than those of DIKBA, TWO\_Q, and DIKQH during the time of running 1:N and N:N patterns. Calculated results can not only meet the real-time demand but also satisfy the preferences of travelers, which can effectively improve the obey rate of guidance service and have positive impacts on development of intelligent guidance systems.

## Conflict of Interests

The authors claim that there is no conflict of interests in their research.

## Acknowledgments

The authors express their sincere appreciation to the Chinese National High Technology Research and Development Program Committee for the financial supports for this research under Grants no. 2012AA112307 and National Natural Science Foundation no. 61074137.

## References

- [1] M. Mesbah, M. Sarvi, I. Ouveysi, and G. Currie, "Optimization of transit priority in the transportation network using a decomposition methodology," *Transportation Research C*, vol. 19, no. 2, pp. 363–373, 2011.
- [2] S. Cicerone, G. D'Angelo, G. di Stefano, D. Frigioni, and V. Maurizio, "Engineering a new algorithm for distributed shortest paths on dynamic networks," *Algorithmica*, vol. 66, no. 1, pp. 51–86, 2013.
- [3] M. R. Khouadjia, E. Talbi, L. Jourdan, B. Sarasola, and E. Alba, "Multi-environmental cooperative parallel metaheuristics for solving dynamic optimization problems," *The Journal of Supercomputing*, vol. 63, no. 3, pp. 836–853, 2013.
- [4] C. Artigues, M.-J. Huguet, F. Gueye, F. Schettini, and L. Dezou, "State-based accelerations and bidirectional search for bi-objective multi-modal shortest paths," *Transportation Research C*, vol. 27, pp. 233–259, 2013.
- [5] L. Fu, "An adaptive routing algorithm for in-vehicle route guidance systems with real-time information," *Transportation Research B*, vol. 35, no. 8, pp. 749–765, 2001.
- [6] Y. Nagasaka, K. Okada, A. Nakamura et al., "Method combining focus-shift and angle multiplexing method for holographic data storage," *Japanese Journal of Applied Physics*, vol. 47, no. 7, pp. 5904–5908, 2008.
- [7] M. A. A. Martínez, J. Cordeau, M. Dell'Amico, and M. Iori, "A branch-and-cut algorithm for the double traveling salesman problem with multiple stacks," *INFORMS Journal on Computing*, vol. 25, no. 1, pp. 41–55, 2013.
- [8] H. Sundell and P. Tsigas, "Lock-free dequeues and doubly linked lists," *Journal of Parallel and Distributed Computing*, vol. 68, no. 7, pp. 1008–1020, 2008.
- [9] A. Arvind and C. P. Rangan, "Symmetric Min-Max heap: a simpler data structure for double-ended priority queue," *Information Processing Letters*, vol. 69, no. 4, pp. 197–199, 1999.
- [10] D. Luxen and P. Sanders, "Hierarchy decomposition for faster user equilibria on road networks," in *Experimental Algorithms*, vol. 6630 of *Lecture Notes in Computer Science*, pp. 242–253, Springer, Berlin, Germany, 2011.
- [11] J. Gilles and Y. Meyer, "Properties of BV-G structures + textures decomposition models. Application to road detection in satellite images," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2793–2800, 2010.
- [12] D. Rosas, J. Castro, and L. Montero, "Using ACCPM in a simplicial decomposition algorithm for the traffic assignment problem," *Computational Optimization and Applications*, vol. 44, no. 2, pp. 289–313, 2009.
- [13] M. Holzer, F. Schulz, and D. Wagner, "Engineering multilevel overlay graphs for shortest-path queries," *Journal of Experimental Algorithmics*, vol. 13, article 5, pp. 1–26, 2008.
- [14] M. Yoshikawa and H. Terai, "Hardware-oriented ant colony optimization and its application to dynamic route guidance problem," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 6, no. 3, pp. 221–228, 2011.
- [15] C. Ren, "Study on improved tabu search algorithm for Min-Max vehicle routing problem," *Applied Mechanics and Materials*, vol. 87, pp. 178–181, 2011.
- [16] L. Roditty, "On the  $k$  shortest simple paths problem in weighted directed graphs," *SIAM Journal on Computing*, vol. 39, no. 6, pp. 2363–2376, 2010.
- [17] H. Meyerhenke, B. Monien, and T. Sauerwald, "A new diffusion-based multilevel algorithm for computing graph partitions," *Journal of Parallel and Distributed Computing*, vol. 69, no. 9, pp. 750–761, 2009.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

