

A Heuristic Bidirectional Hierarchical Path Planning Algorithm Based on Hierarchical Partitioning

Han Zhi-Heng^{1,2}, Rui Xiaoping^{1,*}, Dong Cheng-wei², Song Xian-Feng¹ and Wang Jing¹

¹College of Resources & Environment, University of Chinese Academy of Sciences, Beijing, 100049, P.R. China;

²Beijing Institute of Surveying and Mapping, Beijing, 100049, P.R. China

Abstract: This paper proposes an improvement in the efficiency of the use of a path planning algorithm in large scale road networks. This was achieved by addressing three aspects: hierarchical layers of road networks with sub areas in each layer, selection of the layer entrance point, and the use of bidirectional search technology. This led to a new heuristic bidirectional hierarchical path planning algorithm. The algorithm was tested using Beijing road data, which showed that the time and space efficiency of the new algorithm improved significantly compared to classical algorithms. It also proved that it is reasonable to combine different kinds of path planning algorithm acceleration technologies, and that their effects can be superimposed.

Keywords: Bidirectional, heuristic function, hierarchical, path planning.

1. INTRODUCTION

The rapid development of modern cities and the large-scale popularization of the automobile have placed the traffic situation in most cities under severe stress. Intelligent transportation systems (ITSs), which aim to improve traffic conditions by making optimal use of existing road networks, have emerged as required and are being continuously developed [1]. The problem at the core of ITSs is that of vehicle location and navigation on large-scale road networks, for which a real-time path planning algorithm is a much-needed requirement. However, traditional path planning algorithms, such as the Dijkstra and A* algorithms, are not able to process large amounts of data in real-time [2, 3]. At present, improvement of the path planning algorithm would require three aspects to be addressed: restriction of the network characteristics, the use of a variety of approximate algorithms, and the use of layering technology. The advantage of hierarchical road network technology is that it is capable of significantly reducing the feasible search space of the path planning algorithm. In addition, the performance of this technology has proven superior in situations in which the network growth scales in a nonlinear manner; therefore, it is commonly used for the solution of large-scale network path planning problems [4, 5]. Attempting to further improve the efficiency of the algorithm, this paper introduces a bidirectional search technology based on a hierarchical data structure, which involves the transformation of the layer in the algorithm by introducing a heuristic function, to finally form the bidirectional hierarchical path planning algorithm of this function.

Section 2 of this paper introduces the acceleration technique that is used in path planning algorithms. Section 3 presents the data preprocessing procedure, and describes the method used to select an entrance point between different levels using a new heuristic function and the bidirectional hierarchical algorithm. Section 4 contains the results of the tests that were carried out by applying the algorithms to road data of Beijing City, following which these results are contrasted with those obtained using the classic algorithms. Section 5 summarizes the paper, and discusses ways in which the algorithm could be improved in future.

2. ACCELERATION TECHNIQUE USED IN PAPER

2.1. Hierarchical Technology

When solving complex problems, the fundamental approach followed by hierarchical technology is to first address the key issue and to ignore the details, after which the approach is refined to include the details [5-7]. This approach is consistent with hierarchical spatial reasoning [8]. By some standard [4], the division of a road network into levels can significantly reduce the number of network nodes of a path planning algorithm when processing large-scale road network conditions, thereby improving the efficiency of the algorithm.

At present, domestic and foreign scholars have proposed a variety of hierarchical modes to divide large scale road networks into different levels. Lu and Zhou suggested an optimal path algorithm based on hierarchical spatial reasoning [9], which divides the road network into three levels: main roads, general streets, and lanes or alleys. Each consecutive lower layer contains all the roads in the higher layer. A different approach was adopted by Fu, who divided all roads into two layers [10], of which the main layer contained highways, main roads, and expressways, and the minor layer

*Address correspondence to this author at the College of Resources & Environment, University of Chinese Academy of Sciences, Beijing, 100049, P.R. China; E-mail: ruixpsz@163.com

contained the remaining roads. Connectivity of the higher layer was ensured by artificially adding those roads that do not by default belong to the main layer.

Considering the variety of methods used in existing work, it is clear that there is no recognized standard for processing hierarchical road networks, although generally the roads are divided into two to four layers. The use of too many or too few layers is not reasonable, because using too many layers complicates transfers from one layer to another, whereas using too few layers prevents the loaded data from being significantly reduced; hence, the number of layers should be determined based on the actual situation.

2.2. Bidirectional Search Technology

The classical path planning algorithm is a Breadth-First Search (BFS) algorithm that performs a search by starting from the start point and expanding to the end point and by doing so, generates a tree with the starting point as its root and the expansion points as children. A bidirectional search decomposes the initial problem into two equal sub problems, with the searching process starting from both the start point and the end point at the same time [11]. When either of the two search processes extends from a node that has already been searched in the other direction, the searches meet and then end.

Assuming that the average node connectivity in a road network is m and the height of the tree is n when the searches meet, then the number of expanded nodes will be $\sim m(m^n - 1) / (m - 1)$. Further, assuming that the bidirectional searches meet at the i^{th} layer, the number of nodes expanded in the forward search will be $\sim m(m^i - 1) / (m - 1)$, the number of nodes expanded in backward search will be $\sim m(m^{n-1} - 1) / (m - 1)$, and then the overall number of nodes expanded will be $\sim m(m^i + m^{n-1} - 2) / (m - 1)$. The ratio of the number of nodes expanded in the bidirectional search to the percentage of BFS nodes will be $\sim 2/m^{(n/2)} * 100\%$. As the road network expands, n gradually increases, and the time efficiency of bidirectional search promotion becomes more obvious.

Other researchers have introduced bidirectional search technology in the path planning algorithm [12-14], but found that the efficiency and precision of the bidirectional shortest path algorithm was affected by the transformation method used by the forward and backward search processes as well as by the termination condition. Although bidirectional search technology is generally capable of improving the efficiency of the algorithm, the accelerating effect of this technology alone is not obvious; in fact, it may be worse than that of BFS in some cases. This suggests that a combination of bidirectional search technology and layering techniques may serve to enhance the performance of the algorithm in practice.

3. DATA PREPROCESSING AND BIDIRECTIONAL HIERARCHICAL ALGORITHM

3.1. Preprocessing of Road Network

In general, in a hierarchical road network, a higher level layer is a subset of a lower level layer. This means that the data scale of the lower level layer is close to that of the entire road network, and requires the algorithm to load the data of

the entire road network into the lower level layer. Some researchers would address this problem by using a hierarchical data partition to further reduce the amount of data that is loaded. However, these processing methods focus on achieving partition equilibrium, which, in most cases, will destroy the road network topology connectivity, and are not conducive to the expansion of the search process when using the path planning algorithm. To address this problem, this paper proposes a partitioning approach based on the "hierarchical" strategy, in which the new method no longer emphasizes the balanced partition size, but instead focuses on maintaining the existing road network connectivity.

Each road is assigned a grade feature, with the road number gradually decreasing as the grade level increases. The core idea of the new approach is that, whereas sub regions are generated, the higher-grade road network is used as border demarcation for the second grade, and this process can either be applied recursively right down to the lowest level in the region in which only the lowest grade roads occur, or until the number of nodes reaches the threshold. The specific steps of the method are as follows:

1. Establish a road network topology table, named the Vertex-Edge table, of which the table structure is shown in Table 1;

Table 1. Vertex-Edge relation table.

Vertex Id	Vertex Id
Adj_count	The number of adjacent vertexes or edges
Vertex_adj_id	Ids of adjacent vertexes
Line_adj_id	Ids of adjacent edges
Point (Geometry)	Coordinates of point
Kind	Grade of point

2. Take the entire road network as an input, set the partition boundary road grade K , initialize all nodes in the road network as empty and initialize the variable area Num to mark the sub area number;
3. Select a vertex randomly of which area Num is empty to start, and if the grade of this vertex is greater than K , push this vertex onto the queue Q . If Q is not empty, the loop starts and pops a vertex from Q ; then set the sub area number of this vertex as area Num, and join all the vertexes adjacent to this vertex with Q . When Q is empty, the sub area of which the number equals area Num is completed;
4. Set area Num +=1, select a vertex of which area Num is empty and repeat the above process until all the vertexes have its sub area number;
5. Increase K , repeat the above process until K equals the second lowest grade.

A flow chart of the partition algorithm based on a hierarchical policy is shown in Fig. (1).

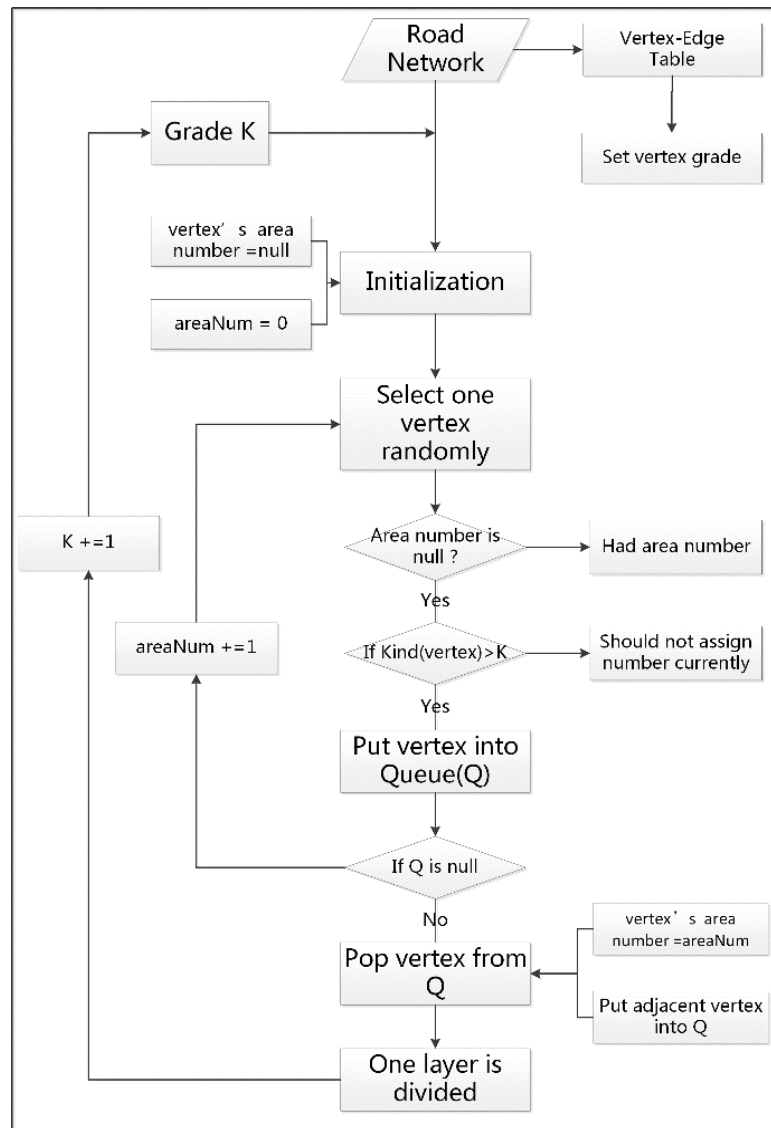


Fig. (1). Flow chart of partition algorithm based on hierarchical policy.

3.2. A New Heuristic Function to Select an Entrance Point

After preprocessing, the road network is divided into many layers, with each layer containing many sub areas. A point of transition between different layers is referred to as the “Entrance Point” [15], selection of which determines the rationality of the planning results. The heuristic function is used to search for points through which the path planning function could make a transition between two different layers.

Li and Shi [16] proposed to restrict the areas from which the entrance point may be chosen to improve the rationality of choice of a suitable entrance point. The introduction of these restricted areas to choose a better entrance point is reasonable, but it does not necessarily improve directionality. This paper proposes the following function to estimate the heuristic distance of each alternate entry point, the heuristic distance represents the probability that each alternate entry point is selected. The smaller the value, the greater the likelihood of being selected from the alternatives, the entrance

point will be the alternative which has the minimum heuristic distance.

$$F_i = D_{(s,i)} + L_{(l,t)}$$

where F_i represents the estimated heuristic distance of alternate entry point i , $D_{(s,i)}$ represents the shortest distance between the starting point and i (the shortest distance is computed by the Dijkstra or A* algorithms) and ensures there is a shortest path between the starting point and i , and $L_{(l,t)}$ represents the Euclidean distance between i and the target point. F_i reflects whether the alternate entrance point is located near the virtual connection of the starting point and the target point. The computation of $D_{(s,i)}$ and $L_{(l,t)}$ is shown in Fig. (2). In addition, it also reflects the possibility of the existence of an alternative entrance point that falls on the optimal path.

The path planning algorithm enters each different layer through its entrance point, and there are many entrance points in one sub area. As the heuristic function that was defined above, requires the shortest path from the starting point to an alternative entrance point, this process will in-

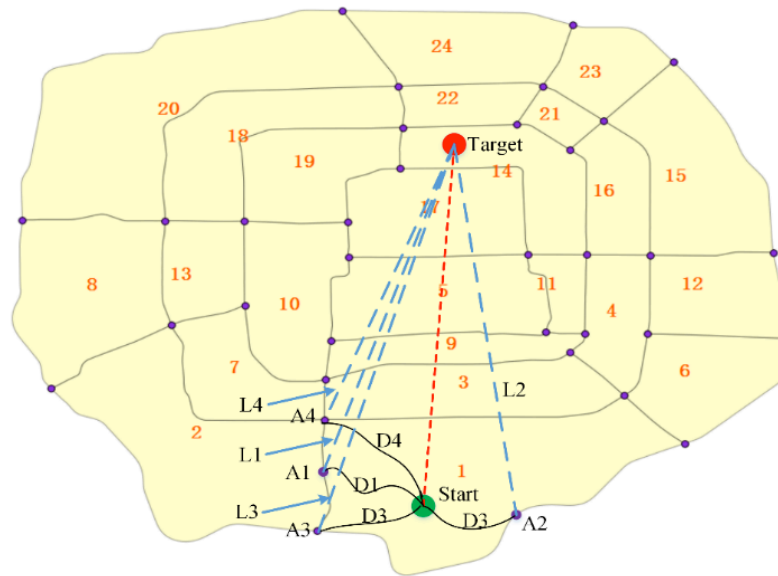


Fig. (2). Compute the heuristic distance of each alternative entrance point.

crease the time cost of the algorithm. However, considering that a sub area is usually small and does not contain too many vertexes and edges, the time cost could be expected to be negligible. Besides, the shortest path would be used in the final result.

3.3. Bidirectional Hierarchical Algorithm

The method based on the alternate implementation of forward and backward searching affects the efficiency of the bidirectional search procedure. Optimal efficiency requires the algorithm to ensure that the number of forward and backward searches is equal, as shown in the analysis below. There are two indicators that can be used for this purpose: the current distance of the shortest path and the search expanded number. Taking the current distance of the shortest path as indicator of transformation and the distances covered during the forward and backward searches are D_f and D_b , respectively, if D_b is greater than D_f , the algorithm executes the forward search, else, the algorithm executes the backward search. However, the search trees generated by using this method may have a large difference in height, which would reduce the efficiency of the algorithm. On the other hand, taking the execution counter as indicator, and the counters of the forward and backward searches are F_{count} and B_{count} , respectively, if B_{count} is greater than F_{count} , the algorithm executes the forward search, else, the algorithm executes the backward search. This approach guarantees alternation of the search direction, thereby avoiding the continuous repetitive execution of either forward or backward searching. The heights of trees will therefore be equal.

The code framework of the bidirectional hierarchical algorithm is as below:

```
//compute the finished layer, this layer is just between the
layer which contains start point and the layer which contains
target point
```

```
GetFinalLayer(startPoint, endPoint)
```

```
//compute the entrance point of start point and target
point
```

```
GetBestEntryPoint(startPoint, endPoint, areaNum)
//initializes the forward search and backward search
queue, counter
Open.Enqueue(startPoint)
_Open.Enqueue(endPoint)
isForward=true
If(reachFinalLayer())
{
    ConstructResultPath()
}
Else
{
    While(Open.Count>0
        _Open.Count>0isSearchFinishes())
    {
        If(isForward)
        {
            forwardSearchExpand()
            isSearchFinishes()
        }
        Else
        {
            backSearchExpand()
            isSearchFinishes()
        }
    }
}
If(count <= _count)
{
    count++
}
```

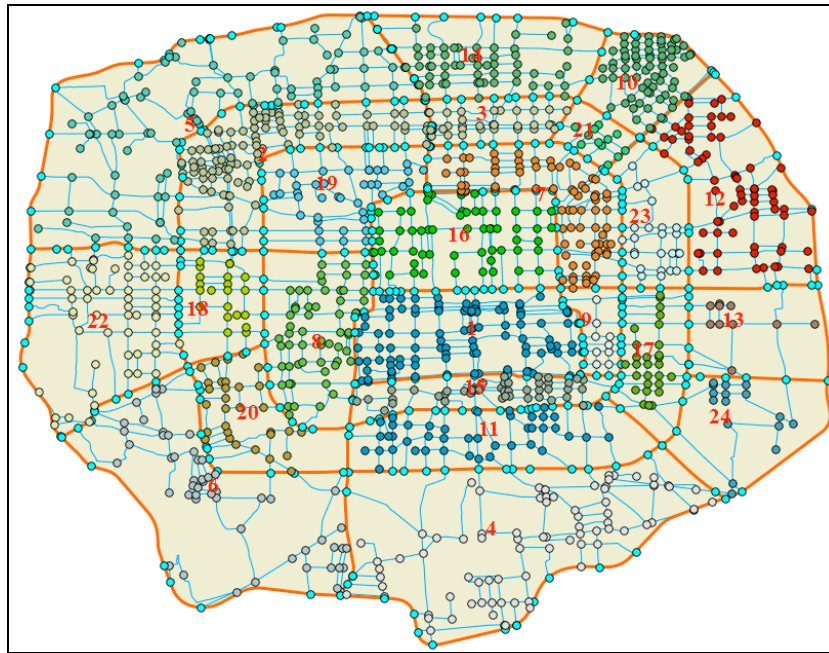


Fig. (3). Partition result with road network.

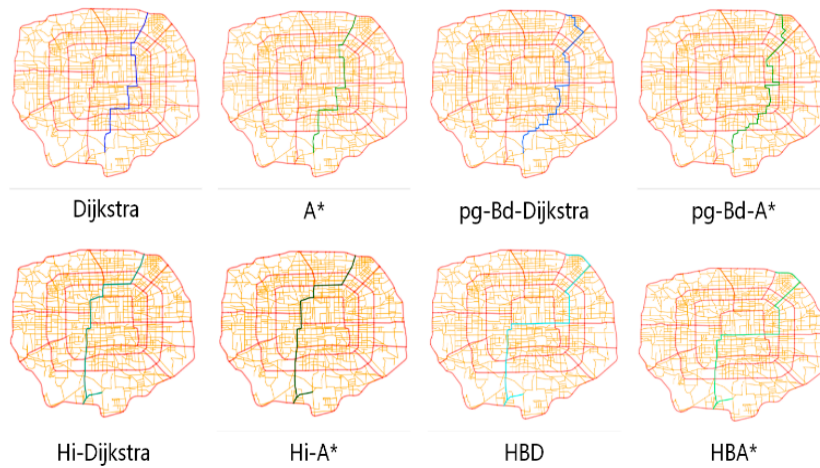


Fig. (4). Comparison of path plan algorithms.

```

    isForward=true
}
Else
{
    _count++
    isForward=false
}
}
}
}

```

4. ALGORITHM TEST RESULTS AND ANALYSIS

The test data consists of all the roads in the Beijing fifth ring. To simplify the problem, the road data is divided into two layers consisting of Level1, which contains highways, main roads, and expressways, and Level 2, which contains

the remainder. Level2 is additionally divided into 25 sub areas. The result is shown in Fig. (3).

The algorithms were tested by selecting the vertex pair randomly. Subsequently, four kinds of algorithms, namely, classical, bidirectional search, hierarchical, and bidirectional hierarchical algorithms, each of which contains a Dijkstra and an A* algorithm, were tested. The results are shown in Fig. (4) and listed in Table 2. For each test, the time cost (column Time), the ratio of loaded vertexes to all vertexes (column Ratio), the length of the shortest path (column Length), the length of the Level1 roads (column H-length), and the estimated time of arrival (column E-time) were recorded. The estimated time of arrival was obtained by using statistics of the road speed that was released by the Beijing Municipal Traffic Commission [17], upon which it was assigned to the relevant road.

Analyzing the Table 2, we can find that, compared with classical algorithms, the time cost and ratio of hierarchical

Table 2. Comparison of path plan algorithms on statistical indexes.

Algorithms		Time (ms)	Ratio	Length (km)	H-length (km)	E-time (min)
classical algorithms	Dijkstra	3211	84.7	30.867	6.26	81.333
	A*	1823	42.8	30.867	6.26	81.333
bidirectional search algorithms	Bd-Dijkstra	1198	/	35.953	15.654	79.6818
	Bd-A*	1184	/	35.953	11.104	89.5398
hierarchical algorithms	Hi-Dijkstra	3110	57.5	45.898	43.033	60.2346
	Hi-A*	2965	40.2	45.898	43.033	60.2346
bidirectional hierarchical algorithms	HBD	332	32.3	44.207	41.342	58.2054
	HBA	227	22.6	44.207	41.342	58.2054

algorithms reduced significantly, it means that layering technology is valuable; compared with classical algorithms, the time cost of bidirectional search algorithms reduced significantly, it means that bidirectional search technology is useful; compared with hierarchical algorithms, the time cost and ratio of bidirectional hierarchical algorithms further reduced, compared with bidirectional search algorithms, the time cost is similar while the ratio falls significantly, which proves that the policy to combine layering technology and bidirectional search technology can improve the algorithm dramatically.

CONCLUSION

This paper describes the application of a variety of combinations of acceleration technologies for use with a path planning algorithm. The destruction of road network topology connectivity resulting from the division of road networks into layers poses a major problem. This paper proposes a solution that uses a partition approach based on a "hierarchy" strategy to maintain existing road network connectivity. This was achieved by proposing the use of a new heuristic function for selecting a reasonable entrance point for the path planning algorithm. The alternate implementation of forward and backward (bidirectional) searching, together with the use of hierarchical algorithms, was shown to be a simple but effective way of solving the problem. The test based on Beijing road data showed that the new algorithm offered a significant improvement in time and space efficiency. However, all the impedance in the test is static, whereas in reality the road conditions are continuously fluctuating. It should be possible to reflect actual road conditions by constructing a model that allows the road impedance to vary with time, which will form the subject of further research.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work is supported by the National Science and Technology support program of China (No. 2012 BAC25B01), the National Science and Technology Major

Project of China (2011ZX05039-004) and the Knowledge Innovation Program of the Chinese Academy of Sciences (KZCX2-EW-QN605).

REFERENCES

- [1] A. Jarašūnienė, "Research into intelligent transport systems (ITS) technologies and efficiency", *Transport*, vol. 22, pp. 61-64, 2007.
- [2] Y.W. Huang, N. Jing, and E. A. Rundensteiner, "A hierarchical path view model for path finding in intelligent transportation systems", *Geo Informatica*, vol. 1, pp. 125-159, 1997.
- [3] E. Kohler, R.H. Mohring, and H. Schilling, "Acceleration of shortest path and constrained shortest path computation," In: *WEA'05 Proceedings of the 4th International Conference on Experimental and Efficient Algorithms*, Grace, 2005, pp. 126-138.
- [4] Q. Song, and X. F. Wang, "Efficient routing on large road networks using hierarchical transactions on intelligent transportation systems", *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 132-140, 2011.
- [5] Q. Song, and X. WANG, "Survey of speedup techniques for shortest path algorithms", *Journal of University of Electronic Science and Technology of China*, vol. 41, pp. 176-184, 2012.
- [6] R. rajagopalan, K. G. Mehrotra, C. K. Mohan, and P. K. Varshney, "Hierarchical path computation approach for large graphs", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, pp. 427-440, 2008.
- [7] F. B. Zhan, "Three fastest shortest path algorithms on real road networks: data structures and procedures", *Journal of Geographic Information and Decision Analysis*, vol. 1, pp. 70-82, 1997.
- [8] A. Car, and A. U. Frank, "General principles of hierarchical spatial reasoning-the case of way finding," In: *Proceedings of the Conference on Spatial Data Handling (SDH 94)*, 1994, pp. 646-664.
- [9] F. Lu, C.H. Zhou, and Q. Wan, "An optimum vehicular path algorithm for traffic network based on hierarchical spatial reasoning", *Journal of Wuhan Technical University of Surveying and Mapping*, vol. 25, pp. 226-232, 2000.
- [10] M.Y. Fu, J. Li, and Z. H. Deng, "A new route planning algorithm based on the hierarchical road network", *Journal of Computer-Aided Design & Computer Graphics*, vol. 17, pp. 719-722, 2005.
- [11] H. Kaindl, and G. Kainz, "Bidirectional heuristic search reconsidered", *Journal of Artificial Intelligence Research*, vol. 7, pp. 283-317, 1997.
- [12] D. Tsatcha, É. Saux, and C. Claramunt, "A bidirectional path-finding algorithm and data structure for maritime routing", *International Journal of Geographical Information Science*, vol. 28, pp. 1355-1377, 2014.
- [13] G. Nannicini, D. Delling, L. Liberti, and D. Schultes, "Bidirectional A* search for time-dependent fast paths," In: *Wea'08: Proceedings of the 7th International Conference on Experimental Algorithms*, Massachusetts, 2008, pp. 334-346.

- [14] H. Ma, and R. Liang, "Using bidirectional search to compute optimal shortest paths over multi-weight graphs," In: *ISCC-C '13 Proceedings of the 2013 International Conference on Information Science and Cloud Computing Companion*, Guang Zhou, 2013, pp. 66-71.
- [15] Q. Li, and Z. Zeng, "A voronoi-based hierarchical graph model of road network for route planning," In: *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*, Beijing, 2008, pp. 599-604.
- [16] J. Yuan Li, and J. Shi, "Optimum path algorithm in traffic network based on model of hierarchical spatial reasoning", *Computer Engineering*, vol. 32, pp. 207-209, 2006.
- [17] *Beijing Traffic Development Annual Report*, Beijing Transportation Research Center, Beijing, 2012, pp. 207-20.

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© Zhi-Heng et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.