Neighbor Search with Hash Map Indexing Technique for Complex Networks

Wai Mar Hlaing[1+] and Myint Myint Sein[2]

[1, 2]Geographic Information System, University of Computer Studies, Yangon, Myanmar

Abstract. Neighbor Search with Hash Map Indexing Technique is used to get the high performance when the optimal path is searched in the complex networks. This system can also give advice the public bus passengers about the travel route depend on the travel time and cost. Moreover, the proposed technique is highly performance one if it compares about the response time of many other popular cited shortest path algorithms. Especially it contains two main parts for finding the optimal path, the first one is dividing the complex large tree into small sub-trees using divide and conquer at an optimal threshold value. The second one is using heuristic neighbor search instead of searching the heuristic values of all expanded nodes at current level. Heuristic neighbor search and hash-map indexing technique is used together to reduce the time complexity when the heuristic values are searched dynamically depend on the user query to reach the target. The proposed system is faster than the popular bi-directional heuristic search A* algorithm, previously proposed combined forward and backward heuristic search algorithm and modified heuristic search algorithm. Road network and bus network in Yangon Region is used as the case study for spatial database.

Keywords: Divider and Conquer, Heuristic Search A*, Combined Forward and Backward Heuristic Search, Modified Heuristic Search

1. Introduction

Transportation contains as a major sector in all countries because it is related all other important sectors such as Education, Health and Business. Especially, most of the developing countries depend on the public transportation system. Shortest path finding is a considerable factor as the challenge for geographical information systems because geographical network is very complex. Shortest path finding algorithms adopt Heuristic Search and Classical Algorithms to find the optimal travel route. However, the problem of time complexity is still facing when the network structure is complex and large. Popular heuristic search algorithm A* is applied to find the shortest path during the short time because it does not need to expand all nodes while traversing the network to reach the target. A* is the high efficiency algorithm when the time complexity is compared among other Classical shortest path algorithms such as mature Dijkstra. However, A* algorithm needs to calculate all nodes in the open list repeatedly. In addition, A* need to pre-store the heuristic values between all nodes of the whole network database. Therefore, pre-processing time may be very long. For these problems, W. M. Hlaing proposed Combined forward and backward heuristic search algorithm [1]. This algorithm does not need to pre-store the heuristic values. It needs to calculate only the heuristic values of the neighbour nodes depend on the dynamically user query. CFBHS algorithm reduces the calculation time about finding the heuristic values of unnecessary nodes to find the optimal route. This algorithm is faster than the popular A* algorithm.

Some short path algorithms work in two directions to get the optimal route. It meets the short path result at a same point in forward and backward directions. It reduces the time complexity by working two directions during the same time when the short result is searched but it still needs to traverse all nodes in the whole graph. F. Islam uses A* algorithm with bi-directional search to meet the best efficiency [2].

CFBHS algorithm is faster than the A* algorithm and bi-directional Dijkstra. However, when the heuristic search A* algorithm is used together with bi-directional search, it is faster than the previous proposed CFBHS algorithm but the optimal result cannot be searched at sometimes because the path does not meet at the same point when the graph is traversed using A* in bi-directional. W. M. Hlaing proposed modified heuristic search algorithm [3]. It is faster than the CFBHS algorithm and A* algorithm in bi-directional search by working in one direction for finding the optimal route. At sometimes, modified heuristic algorithm cannot find the shortest path because it removes unexpectly some necessary nodes while searching the shortest path. However, the time efficiency is so fast if it compares the other algorithms such as CFBHS, heuristic search A*. Our work modifies the previously proposed CFBHS algorithm. It is faster than the CFBHS algorithm and the accuracy is better than the previously proposed modified heuristic search algorithm. And the main contributions for this work include:

- The short path finding in each subtree of a large tree network structure uses the hash map indexing approach to be easy and fast for retrieving the dynamically heuristic values.
- To increase the efficiency by reducing the calculation of unnecessary nodes repeatedly.
- To reduce the time complexity by studying the relations among the subtrees.

Indexing technique is used to be easy and fast for retrieving the required information from big data [4]. Hash map indexing is applied together with many techniques and algorithms such as Hash Map Indexing based on online querying, Approximate Nearest Neighbour Search and Content based Image Retrieval [5].

The rest of this paper is organized as follows. The newly proposed shortest path approach (i.e., the CFBHS with Hash Map Indexing) will be described in Section 2. Section 3 describes the comparison of node reduction, performance and accuracy among of different algorithms using statistical analysis. Section 4 concludes the paper.

2. Proposed Neighbour Search with Hash Map Indexing Technique

Retrieving the required information from Big Data is necessary for every sector. To be fast and easy for extracting the useful information from Geospatial Big Data becomes a challenge in Nowadays. Our proposed work will modify the previously proposed CFBHS algorithm by combining the mature approach known Hash Map Indexing Technique.

2.1. Hash Map Indexing Technique

Most of the research works when the data size is huge, these use the indexing technique to get high efficiency for data retrieving. Otherwise, at sometimes it is used as a caching technique and the recently used data are stored in hash table instead of database, file and other storage areas. Because, the time complexity about data accessing of hash map indexing is constant time. In our system, hash table is used as a memory cache. Hash table contains two parts: keys and values as shown in Fig. 1. Current Node and Target Node is saved together as a key. The respective values of the keys are stored by finding the heuristic distance values between these two nodes. These heuristic values are searched dynamically depend on the user query. The proposed system uses the number of hash table depend on the maximum degree of the vertex in the graph. The system uses only one hash table for sharing the heuristic values in this table among of the subtrees. All subtrees store their estimated heuristic values in the same hash table. Other hash tables are used to store the node that has the minimum value among of the current neighbor nodes.

2.2. Overview of the Proposed System

The proposed system is an extension of the work that previously published in the Global Conference on Consumer Electronics and already accepted the Journal of Internet Technology. When the size of the geospatial data is large, finding the optimal path in the short time becomes a challenge. The modified heuristic search algorithm is based on the CFBHS algorithm [3]. However, it works the short path finding in one direction. At first, this modified heuristic search algorithm splits the large tree into small subtrees. Each small subtree finds the local optimal shortest paths until the number of predefined threshold value is reached.

When the short path is searched from each subtree, it expands the tree only the neighbor nodes of the current node as the partial paths for the next level. In this way, it traverses the tree to reach the target. Euclidean Distance method is used for finding the distance values between the two nodes. For this case, the heuristic values between the current node and target may be duplicated among of the subtrees.

Our system solved about these problems using hash map indexing technique. Using the hash map indexing technique, the proposed system can reduce the unnecessary calculation times among of the subtrees. Fig. 1 describes the overview of the proposed system. The detailed works of the proposed system have described in the Journal of Internet Technology. Firstly, the system constructs the spatial database into graph database using Haversine Distance Method. Then, the tree network is separated into small subtrees using degree-based threshold calculation method. This pre-processing step needs to find the suitable threshold value only for the first time if the same database is used.

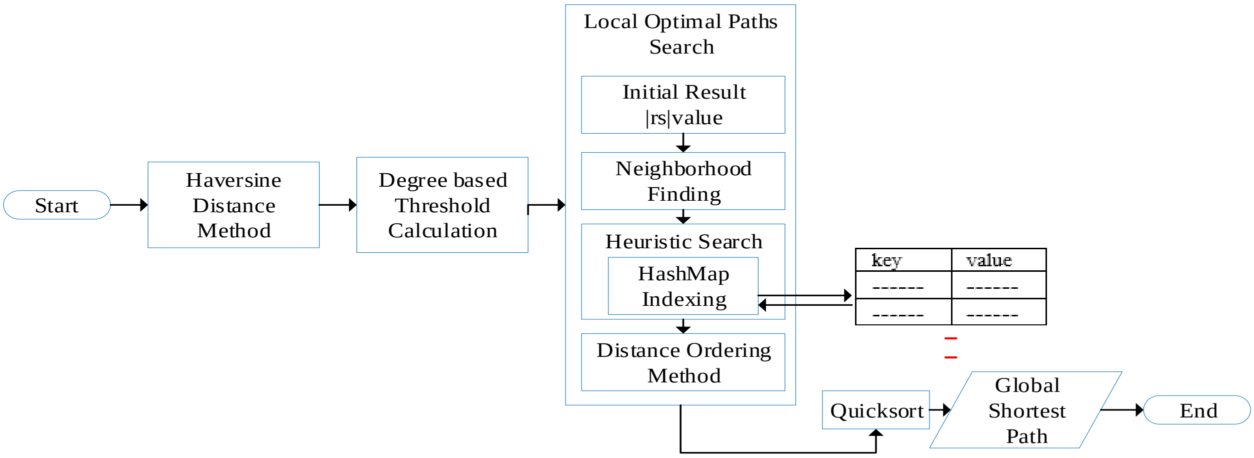


Fig. 1: System Flow

When the threshold value for dividing the tree has been defined, the system will define the number of results |rs| that will need to find from each subtree. When the shortest path is searched, the system considers the neighbor nodes of the current node to reach the target node. It does not expand all nodes because the time complexity may be high. In the previous proposed CFBHS algorithm, we need to find the paths in both forward and backward direction because it removed many nodes while going to target by considering only on the neighbors. However, it is faster than the popular heuristic A* algorithm. When the modified heuristic search is used, it is faster than CFBHS algorithm. This modified heuristic search algorithm's accuracy may be low because of finding the optimal paths in forward direction.

In this proposed system, we use hash map indexing technique when the heuristic value is searched to satisfy high accuracy and efficiency. The number of hash map table depends on the nature of the graph. If the maximum degree of a graph contains n, the number of hash map table also contains from Hash map table $H1, H2.., Hn$. The first table $H1$ is storing the heuristic values of expanded nodes for current query. The second table $H2$ is storing the comparison results of two nodes. For example, if node 'a' and node 'b' are compared, the system will store hash map key (a, b) and the node that possesses the smallest distance to reach the target among of these two nodes will be stored as the hash map value. Hash map table H3 will also store the comparison results among of three sub-nodes in current parent node. The remaining tables are saving the comparison results in this way. The heuristic estimated values are searched using straight line Euclidean distance. The smallest heuristic distance is choosing as the node that need for expanding next level. In this way, this partial path goes until the target node is reached. If the number of current results is less than the predefined result number |rs|, the system finds second local optimum path by removing, target node and a node before target occurs. After finding the local optimum results from each subtree, Quicksort algorithm is used to order the local optimal paths according to their total distance cost. Finally, the system will give the global shortest path.

2.3. Combining the Hash Map Indexing Technique in the Proposed System

When the current node of the tree is expanded into the neighbor nodes for the next level, the system must choose one node among of the neighbor nodes to go the target. The system needs to calculate the heuristic values between these neighbor nodes and the target node. The modified steps for the proposed system contain as follows:

1) Before calculating these heuristic values, the modified system will check the Hash Map table.
2) The system will check whether the current key value of the subtree already exists in the Hash Map Table. Key Name is the storing current node and target node together.

a. If the current key value already exists in the table, estimated heuristic value will not calculate again.
b. Otherwise, heuristic value will be calculated and stored together with key name in the Hash Map Table.

3) The system will compare the heuristic distance values among the expanded nodes of current level and target node.

a. Before comparison is made, the system will check whether this comparison already finish in their respective hash map table. If the number of expanded nodes of current level is four, hash map table H4 will be used. This table contains four nodes in a key. If the key already exists, the comparison result does not need to calculate again.
b. Otherwise, smallest distance values will be searched among of these nodes in current level of the tree.

Whenever the system decides to choose a node for the next level from each subtree, the modified steps that mentioned in above are used. When the network structure is complex and huge, these duplicate nodes calculation problems among of these subtrees are more evident. By combining the hash map indexing technique, previous nodes calculation problems among of the subtrees can be reduced.

3. Experimental Results

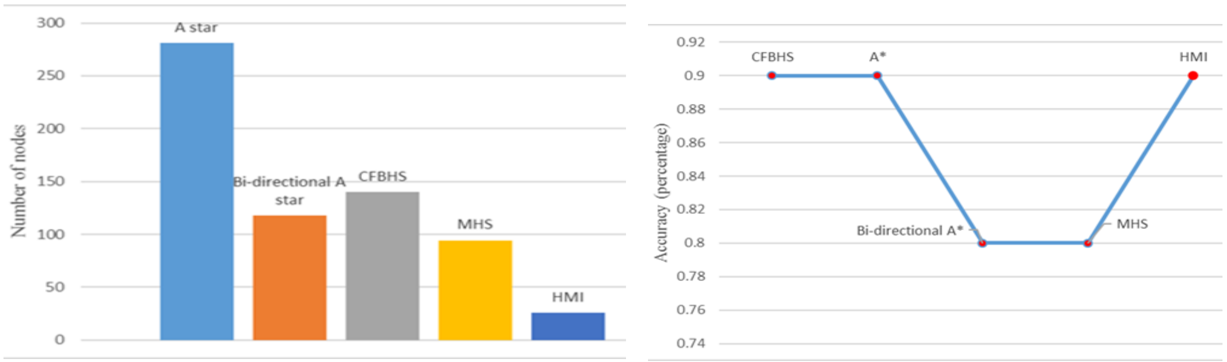
The spatial data of Yangon city in Myanmar are used for evaluating the proposed method. This section especially contains two main parts. The first part contains checking the number of compared nodes at each level of the network among of the algorithms using Yangon Downtown Network that contiains 44 nodes. It considers the junction points as the nodes of the Road network. Moreover, it compares the accuarcy among of different algorithms. The second part describes the statistical results about the accuracy of the algorithms such as A*, Bi-directional A*, Combined Forward and Backward Heuristic Search, Modified Heuristic Search and Hash Map Indexing with Heuristic Method using Yangon Bus Network. It is a complex hypergraph network and it contains about 4000 nodes and many edges.

3.1. Reducing Unnecessary Node Calculation and Accuracy Among of the Different Algorithms

Fig. 2 (a) describes the required nodes among of the short path finding algorithms that need to compare while finding the optimal route. Hash Map Indexing technique (HMI) need to compare only the heuristic values among of the few nodes. HMI is better than the previous proposed CFBHS and MHS algorithms according to the node reduction comparison. HMI calculates heuristic values dynamically depend on the user query and temporarily stored in the hash map tables. By adding hash map indexing technique in previous proposed CFBHS algorithm, it is significantly reducing the heuristic values calculation and nodes comparison jobs.

CFBHS algorithm is better than popular heuristic search calculation A* because CFBHS algorithm depends only on the neighbor nodes to choose a node for the next level. A* considers not only the expanded nodes of the current level but also all nodes in the open list. Modified Heuristic Search algorithm is better than Bi-directional A* algorithm because it works on the neighbor nodes in one directional search.

The accuracy of five short path finding algorithms using Yangon Downtown Network is compared in Fig. 2(b). The accuracy of these algorithms is searched using Eq.1. The percentage of accuracy is dividing the number of correct rules by the total number of rules. According to the comparisons, CFBHS, A*, CFBHS with hash map indexing is more accurate than the bi-directional A* and Modified Heuristic Search.



(a) (b)

Fig. 2: Number of comparison nodes and Accuracy among of Different Algorithms

$$\text{ACCURACY} = \frac{\text{CORRECTNESS}}{\text{TOTAL}} \times 100\% \tag{1}$$

3.2. Comparison About the Performance Among CFBHS, Modified Heuristic Search and Hash Map Indexing with Heuristic Method for the Yangon Region

Our system checks the accuracy and performance using different networks. Section 3.1 describes the performance and accuracy about Yangon Downtown Network. Section 3.2 in Current section also describes the comparison among of the algorithms using Complex Yangon Bus Network. HMI is the fastest heuristic search algorithm among of the other algorithms described in Fig. 3.

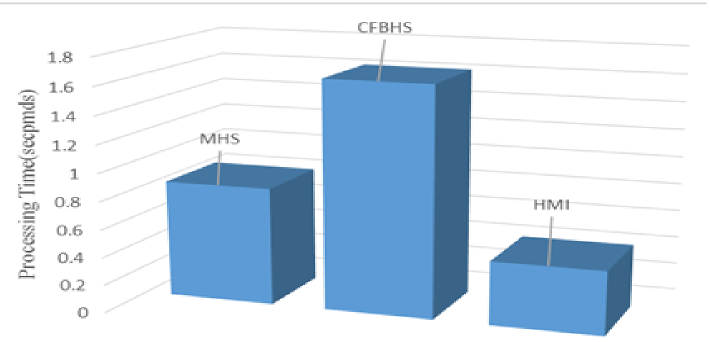


Fig. 3: The Processing Time among of Different Algorithms

4. Conclusion

Combined Forward and Backward heuristic search with hash map indexing technique is used to find the optimal route. This algorithm works only on the neighbor nodes while searching the optimal route. It divides the large tree into small subtrees. Therefore, it is faster than other popular heuristic search algorithms. This paper combines hash map indexing technique in CFBHS algorithm. This reduces unnecessary heuristic calculation and comparison time among of the subtrees. According to the statistical analysis and comparisons, by combining hash map indexing technique in combined forward and backward heuristic search algorithm, it outperforms to satisfy both conditions about accuracy and performance for different networks.

5. References

[1] W. M. Hlaing, S. Liu, and J. Pan. A Novel Solution for Simultaneously Finding the Shortest and Possible Paths in Complex Networks. J. Internet Technol., Sci. vol. 20, no. 6, pp. 1693–1708, 2019.

[2] F. Islam. A * -Connect: Bounded Suboptimal Bidirectional Heuristic Search. *Proc. of International Conference on Robotics and Automation (ICRA)*, pp. 2752–2758.

[3] W. M. Hlaing, M. M. Sein. K-means Nearest Point Search Algorithm and Heuristic Search for Transportation. 2018 IEEE 7th Glob. Conf. Consum. Electron., pp.779–780.

[4] J. Wang, W. Liu, S. Kumar, and S. Chang. Learning to Hash for Indexing Big Data - A Survey. 2015.

[5] P. Sadeghi-tehran. Scalable Database Indexing and Fast Image Retrieval Based on Deep Learning and Hierarchically Nested Structure Applied to Remote Sensing and Plant Biology, 2019.