Route computation in large road networks: A hierarchical approach

Jagadeesh, G. R.; Srikanthan, T.

2008

Jagadeesh, G.R., & Srikanthan, T. (2008). Route computation in large road networks: a hierarchical approach. IET Intelligent Transport Systems, 2(3), 219 - 227.

<https://hdl.handle.net/10356/79622>

<https://doi.org/10.1049/iet-its:20080012>

© 2008 The Institution of Engineering and Technology. This is the author created version of a work that has been peer reviewed and accepted for publication by IET Intelligent Transport Systems, Institution of Engineering and Technology. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [<http://dx.doi.org/10.1049/iet-its:20080012>].

Downloaded on 26 Apr 2023 23:10:09 SGT

ROUTE COMPUTATION IN LARGE ROAD NETWORKS: A HIERARCHICAL APPROACH

G.R. Jagadeesh and T.Srikanthan

[asgeorge@ntu.edu.sg, astsrikan@ntu.edu.sg]

Centre for High Performance Embedded Systems

School of Computer Engineering

Nanyang Technological University

Singapore 639798

ABSTRACT

The optimal route between a given origin and destination needs to be computed in a fast and efficient manner in dynamic route guidance systems. Conventional routing algorithms have been found to be inadequate when applied directly to large road networks. Algorithms based on the concept of hierarchical abstraction make use of the knowledge about the road network to reduce search and provide near-optimal solutions. In this paper, we present a generic procedure for organising a given road network as a multiple-layer hierarchy. We propose an efficient hierarchical routing algorithm, which breaks down the route search into a number of individual searches in small sub-networks. The algorithm incorporates a heuristic layer-switching technique to improve its performance without compromising the accuracy. The hierarchical routing algorithm was tested on the road network of Singapore and the solutions were found to be comparable to the optimal least-cost paths.

Key words: Hierarchical routing, Dynamic route guidance, Road networks, Heuristic search.

1. INTRODUCTION

Dynamic route guidance systems help to tackle many of the transportation problems by minimising congestion and ensuring uniform utilisation of the road network. In the computer literature, finding an optimal route between a given origin and destination is known as the shortest path problem. In the context of vehicle navigation, the optimal route could be computed based on a variety of factors such as travel distance, travel cost or any other criteria preferred by the driver. The shortest path problem remains a well-researched area and there are a number of algorithms to solve it such as Dijkstra's algorithm, A* algorithm and their many variants [1].

In applications such as dynamic route guidance, the optimal route between two points needs to be computed in a fast and efficient manner. However, the response time of the shortest path algorithms is a function of the network size. Empirical results indicate that the computation time becomes unacceptable as the network size becomes large [2]. This is particularly true for in-vehicle route guidance systems that are typically embedded systems with limited computation power and memory resources. Shortest path algorithms are not designed to utilise knowledge about the road network to improve the search efficiency. Also, the solutions provided by these algorithms may not match the preferences of drivers such as travelling on major roads.

In this paper, we propose an approach based on hierarchical abstraction to substantially reduce the complexity of route computation in large road networks. In problems such as route finding, abstraction breaks down the route search into a number of smaller searches. The total complexity is reduced to the sum of the complexities of the individual searches rather than the product of complexities [3].

2. RELATED RESEARCH

The road network is represented as a graph consisting of a set of nodes (junctions) and links (roads). It is common knowledge that in any road network, the roads are classified into various classes such as highways, expressways and residential streets. Most researchers who have proposed hierarchical routing algorithms recommend the formation of hierarchy on the basis of road types [4], [5], [6], [7], [8], [9]. This is chiefly based on the observation that a major portion of all journeys lie on major roads such as highways and expressways, which permit faster travel. Our experiments with the road network of Singapore validate the above observation.

An alternate approach to forming multiple levels of abstraction for route finding was proposed by Timpf *et al* [10]. They proposed a conceptual model with three levels of abstraction named planning, instructional and driver levels based on human spatial reasoning. In this approach an object may be present in all the three levels but with different levels of detail. For instance, while in the planning level a highway is considered as an abstract edge that connects two nodes, the driver level includes information about the lanes present in the highway.

Hierarchical routing algorithms tend to restrict the route computation to small networks, which are subsets of the original road network. The road network could be divided into various sub-networks either by geographically partitioning it [11], [12] or by taking advantage of an interesting property of the road network. In a large road network, it could be observed that the major roads form a fully connected network among them and naturally partition the whole network into many smaller grids [5]. A typical hierarchically organised road network would contain multiple layers of abstraction with each layer other than the

topmost one divided into many regions or grids. The topmost layer consists of only the most important roads in the network and the lowest layer includes all roads.

When hierarchical routing algorithms are employed to find the best routes in a hierarchically abstracted road network, the results are typically not guaranteed to be optimal. Some hierarchical routing algorithms that aim to provide optimal routes require a large number of shortest paths in the network to be pre-computed and stored [11], [12], [13]. However, pre-computation usually requires prohibitively large storage space [14]. Most hierarchical routing algorithms proposed in the literature produce near-optimal routes and do not involve pre-computation of shortest paths.

Liu [5] proposed a hierarchical routing algorithm wherein the grid sub-networks containing the origin and the destination are linked to the major road network and a shortest path algorithm is applied to this augmented major road network. The routes found using the above method were found to be about 9% longer than the optimal routes. However, this algorithm does not break down the search into multiple searches. Instead, a single search is carried out to find the best route between the origin and the destination on the augmented major road network. The size of this augmented network could be large especially when large road networks with more than two layers are involved.

Assuming a two-layer hierarchy, the algorithms described in [4], [6] operate as follows. The optimal route between the origin in the bottom layer and the nearest node that belongs to the top layer is computed using a conventional shortest path algorithm. The same procedure is repeated for the destination. Next, the optimal route between the two top layer nodes that are nearest to the origin and destination is computed. Although the above method offers performance gain over non-hierarchical algorithms, choosing the nearest top layer node to connect to the higher layer results in substantial loss of accuracy [15]. For instance, assume

that the destination lies to the right of the origin and the nearest top layer node lies to its left. The hierarchical algorithm would require the vehicle to travel to the node on the left and then turn back and travel towards the destination in the right. Such a route would not be acceptable to most drivers.

A more accurate but computationally expensive approach would be to exhaustively invoke the hierarchical algorithm to compute the routes passing through each of the top layer nodes in the origin and destination sub-networks and subsequently select the pair of top layer nodes that yield the least-cost path [16]. We shall refer to this method as the best-node approach. It was established in [16] through evaluations on a real road network that the best-node approach achieved an error rate that was less than one-fourth of that of the nearest-node approach. However, the best-node approach involves a very large number of route computations and was found to require approximately seven times more computation time compared to the nearest-node approach.

A hierarchical routing algorithm that seeks to achieve the same level of accuracy as that of the best-node approach while considerably reducing the computation time was proposed by Quek and Srikanthan [8]. In this approach, the optimal routes connecting the origin / destination to multiple nodes in the top layer are computed. This information is used to subsequently promote and represent the origin and destination in the top layer. Bhagavatula [7] proposed a similar hierarchical routing algorithm, where the optimal routes between the origin / destination and all the top layer nodes within a circle of certain radius are computed. However, it is obvious that only one of these routes will form part of the final route between the origin and the destination. Therefore, though this approach offers better accuracy, it involves superfluous route computations. The number of such superfluous route computations becomes large when the hierarchy consists of more than two layers.

In our previous work [17], we have proposed a heuristic layer-switching technique to eliminate all unwanted route computations. However, that solution is applicable only to a two-layer hierarchy, which is inadequate to represent very large road networks. In this paper, we present a generic and formal methodology for partitioning a given road network into more than two layers of abstraction as well as a hierarchical routing algorithm that operates on such a hierarchy. The hierarchical routing algorithm employs the heuristic layer-switching technique to avoid superfluous route computations. Thus it addresses one of the most important challenges encountered by hierarchical routing algorithms for road networks: How to efficiently switch to the abstracted higher layer without compromising accuracy?

3. APPROACH

In the proposed approach, the route computation is performed only within small sub-networks, which we refer to as routable networks. It is assumed that an efficient shortest path algorithm exists for finding the optimal route in a routable network. In order to maintain the runtime of the algorithm within acceptable limits, it is ensured that the routable networks do not have more than a certain number of nodes. We denote this maximum limit as $N_{\max}$. The value of $N_{\max}$ is determined based on many factors such as the number of nodes in the entire network and the computational resources available. While estimating the size of a sub-network, the terminal nodes (dead ends) present in it should not be considered. This is because a terminal node need not be explored during the search for the best route unless the node happens to be the origin or the destination.

The roads in the network could be classified into various classes. For instance, in a typical country network, the residential streets could be classified as class 0, the semi-major roads as class 1, the highways as class 2 and so on. If there are k classes, then k different road

collections need to be formed with varying degrees of abstraction such that collection i will contain all roads of class i or higher. The collections for a typical country road network form a pyramid structure as the example illustrated in Figure 1. The lowest collection includes all roads in the network. The one that is immediately above it includes only a subset of the lowest collection. The highest collection consists of only the most vital roads in the network. As it could be seen, a road may belong to more than one collection.

In the proposed approach, the road network is represented as a hierarchy consisting of multiple layers. Each layer is an abstraction of the layer immediately below it. For any layer other than the highest layer, we define the abstraction ratio as the ratio of the number of non-terminal nodes in that layer to the number of non-terminal nodes in the layer immediately above it. The roads in each layer form a fully connected network. In other words, each node in a layer is reachable from every other node in that layer. Connections to a higher layer occur via entry / exit nodes, which we refer to as E-nodes. Every node in a layer is mapped to a few nodes in the next higher layer, which act as its E-nodes. Typically, for a given node in a lower layer fully or partially bounded by higher-layer roads, the nodes that lie at the intersections of those bounding higher-layer roads with other higher-layer roads serve as its E-nodes. Each layer other than the topmost one is partitioned into a number of grids as described later. The roads within each grid form a fully connected network.

Before we proceed to describe the procedure for constructing a multiple-layer hierarchy, it is necessary to have a brief understanding of how route computation is carried out in the proposed hierarchical routing algorithm. The algorithm starts with the lowest layer. If the origin and the destination are present within the same grid, then the shortest path algorithm is applied to the grid. Else if the origin and the destination belong to two adjacent grids, then the shortest path algorithm is applied to the union of the two grids. If both the

above conditions are not satisfied, then the origin and the destination are promoted to the next higher layer by connecting them to their E-nodes. If a solution is not obtainable in that layer too, then the origin, destination and their E-nodes are promoted to the next higher layer. This is achieved by connecting the E-nodes of the origin and destination with their respective E-nodes in the next higher layer. This process is repeated till a solution is found or the highest layer is reached. The highest layer could be considered as a single grid and therefore, a solution is always obtainable. The concept of promoting nodes from the lower layers to a higher layer is illustrated in Figure 2, where grid lines of different thickness represent roads of different types.

For any layer other than the lowest layer, the maximum number of nodes that could be promoted from the layers below it needs to be determined. We refer to this number as the promotion limit. The promotion limit depends on the maximum number of E-nodes per node. For example, let us assume that each node has up to 4 E-nodes. In the case of layer 1, only the origin and the destination could be promoted from layer 0. Hence, the promotion limit of layer 1 is given by $P_1 = 2$. In the case of layer 2, in addition to the origin and the destination, their E-nodes in layer 1 are also promoted. Therefore, the promotion limit of layer 2 is given by $P_2 = 2 + 4 + 4 = 10$. In general, for $x \neq 0$, the promotion limit of layer x is given by:

$$P_x = 2 \sum_{i=0}^{x-1} n_e^i \quad (1)$$

where n_e is the maximum number of E-nodes per node.

3.1 Constructing a Multiple-Layer Hierarchy

We present a methodology for organising a given road network into multiple layers of abstraction. The layer formation is a one-time manual process and the same set of layers could be used repeatedly to find the best route between any two points in the network. The

number of layers depends mainly on the total number of nodes in the entire road network and the maximum number of nodes in the routable network, N_{max} . Prior to layer formation, the roads in the network need to be arranged into various collections as explained in the previous section. The steps involved in layer formation are as follows.

1. Let x denote the current layer. To start with, $x = 0$ (Layer 0). Layer 0 includes all the roads in the network (i.e., Collection 0). Partition the layer into various grids such that each grid does not contain more than $((N_{max} - 2) / 2)$ non-terminal nodes.
2. The roads belonging to the next higher collection are selected to form the next layer. $x = x + 1$. (Next higher layer).
3. If the total number of non-terminal nodes in Layer x is less than or equal to $(N_{max} - P_x)$, where P_x is the promotion limit of layer x , proceed to Step 5.
4. Partition the Layer x into grids such that each grid does not contain more than $((N_{max} - P_x) / 2)$ non-terminal nodes. Go to Step 2.
5. Identify low-level roads from Collection $(x - 1)$ that can function as vital shortcuts in Layer x and add them to Layer x till the total number of non-terminal nodes in Layer x becomes $(N_{max} - P_x)$. The layer formation ends because the highest layer is reached.

In Step 1, we choose $((N_{max} - 2) / 2)$ as the maximum limit for the number of non-terminal nodes in a grid in the lowest layer. This is due to the fact that in the event of the origin and the destination falling on two adjacent grids, the shortest path algorithm needs to be applied to the union of the two grids. Therefore, we ensure that the size of a grid does not exceed half the maximum size of the routable network. $(N_{max} - 2)$ is chosen as the limit

instead of N_{max} because two nodes have to be reserved for the origin and the destination, in case they happen to be terminal nodes. Similarly, in Step 4, we fix $((N_{max} - P_x) / 2)$ as the limit for the number of non-terminal nodes in a grid in layer x because the nodes that would be promoted from the lower layers have to be accommodated. When a layer with $(N_{max} - P_x)$ non-terminal nodes is reached, the layer formation ends because a routable network is realised and hence no further abstraction is necessary.

3.1.1 Grid Formation

The procedure for layer formation explained earlier requires each layer other than the topmost one to be partitioned into multiple grids. The road network in each layer is naturally partitioned into many grids by the roads belonging to the next higher layer as illustrated in Figure 3. However, the grids thus formed are likely to significantly vary in size. It has been established that when decomposing a network into multiple subnetworks to facilitate high-speed route computation, the favourable decomposition schemes are ones in which the number of subnetworks is relatively small and subnetworks are of equal size [18]. Hence, there is a need to manually regulate the size of the grids such that the grids are of roughly equal size and the number of grids is maintained small.

If the number of non-terminal nodes in a naturally formed grid is more than the prescribed limit laid down by the layer formation procedure, the grid needs to be broken into smaller grids by promoting some important low-level roads as high-level roads. It is preferable to promote those low-level roads that when elevated to the higher layer, can serve as shortcuts in that layer. On the other hand, if the number of non-terminal nodes in the grid is lesser than the limit, the possibility of combining two or more natural grids into a single grid should be explored. For example, in Figure 4, three natural grids are combined into one grid, which is represented by the shaded region.

For each grid in a given layer, information regarding the adjacent grids needs to be stored. Two grids are considered as adjacent if they share a common border. A node that lies on the common border between two grids should be represented in both the grids. Roads that form the boundary of a grid are also considered as part of the grid.

3.1.1 E-node Mapping

In the proposed hierarchical routing algorithm, switching between different layers occurs through the E-nodes. Therefore, for each node, information about its E-nodes in the higher layer needs to be stored. The simplest way of doing this is to directly map each node to a set of E-nodes surrounding it. We refer to this method as node-based E-node mapping. While this method is suitable for layers with less number of nodes, applying it to a layer with a large number of nodes requires large storage resources. It is likely that all the nodes in a particular region or grid will have the same set of E-nodes in the higher layer. Therefore, it is advantageous to map each grid to a set of E-nodes, which is common for all the nodes in the grid. Typically, for a grid fully or partially bounded by higher-layer roads, the nodes that lie at the intersections of those bounding higher-layer roads with other higher-layer roads are chosen as the E-nodes. This method is referred to as grid-based E-node mapping.

4. HIERARCHICAL ROUTING ALGORITHM

In this section, we present a hierarchical routing algorithm to efficiently compute a near-optimal route in a road network organised as a multiple-layer hierarchy. Avoiding overhead during layer switching and determining when to switch to a different layer play important roles in the efficiency of hierarchical routing algorithms [19]. As discussed earlier, the methods proposed in the literature for layer switching are either inaccurate or involve

computational overhead. In the following, we present a novel heuristic technique to efficiently switch between layers without compromising the accuracy.

4.1 Heuristic Layer-Switching Technique

In the proposed algorithm, the origin and the destination are linked to the next higher layer by connecting them to all of their respective E-nodes. The connection between a given node and its E-node is established using a pseudo link, which represents the optimal route between them. Therefore, it follows that the cost (such as travel distance or travel time) of traversing the pseudo link should be equal to the travel cost of the optimal route between the nodes connected by it. This implies that the optimal routes between a given node and all its E-nodes have to be computed before promoting it to the higher layer. However, it is wasteful to compute the optimal routes between a node and all its E-nodes because only one of these routes can be part of the final route between the origin and the destination and the rest are rendered superfluous.

In the proposed heuristic layer-switching technique, the optimal routes between a node and its E-nodes are not computed in advance. Instead, a heuristic estimate of the travel cost of the optimal route between the node and each of its E-nodes is assigned as the travel cost of the pseudo link that connects them. After the route between the origin and the destination in the higher layer is found, the pseudo links that form part of the route are identified. Subsequently, each of the pseudo links in the route is replaced by the optimal route between the two nodes connected by it. This concept is illustrated in Figure 5 using a two-layer hierarchy.

Extending the above method to a hierarchy with more than two layers, the origin and the destination are iteratively promoted to the higher layers until they fall within a routable

network. The origin and the destination are deemed to be part of a routable network if any of the following three conditions is met.

1. The origin and the destination belong to the same grid.
2. The origin and the destination belong to two grids adjacent to each other.
3. The origin and the destination are part of the topmost layer.

In the worst case, the origin and the destination are promoted all the way up to the topmost layer. The origin and the destination are connected to the nodes in the topmost layer through multiple chains of pseudo links. However, only two of these chains would eventually form part of the final route. A tentative route between the origin and the destination is found in the topmost layer. The tentative route consists of the following 3 portions.

1. A chain of pseudo links connecting the origin to the topmost layer.
2. A chain of real links in the topmost layer.
3. A chain of pseudo links connecting the topmost layer to the destination.

In order to obtain the final route, each pseudo link in the tentative route is identified and replaced by the optimal route between the two nodes connected by the pseudo link.

If we regard the optimal routes as synonymous with routes having shortest lengths, then ideally, the optimal route between two nodes is a straight road connecting them. Although such routes rarely exist in practice, our experiments with the road network of Singapore suggest that in a well-connected city road network the straight-line (Euclidean) distance is a reasonable estimate of the actual travel distance. However, computing the Euclidean distance involves square and square root computations, which slow down the response of the algorithm. In our work, we make use of an efficiently computable distance

metric, which closely approximates the Euclidean distance. The heuristic estimate of the travel distance between two nodes located at co-ordinates (x_1, y_1) and (x_2, y_2) is given by:

$$D_h = \max (|x_1 - x_2|, |y_1 - y_2|) + \min (|x_1 - x_2|, |y_1 - y_2|) / 2 \quad (2)$$

Our evaluations have shown that there is no marked difference in the quality of the computed routes when the approximate distance shown in Equation (2) was used as a heuristic in the place of the real Euclidean distance. Out of 100 routes computed using the approximate distance heuristic, only one route differed from the corresponding route computed with the Euclidean distance heuristic [20].

4.2 The Algorithm

The algorithm makes use of three lists namely PROMOTED, ENODES and CURRENT to store different sets of nodes. We make use of a function *layer* (n), which denotes the highest layer at which a given node n is present. It is assumed that an efficient shortest path algorithm exists for finding the optimal route within the routable networks. The complete hierarchical routing algorithm is presented below.

1. Let x denote the current layer. To start with, $x = 0$. (Layer 0). Let the initial PROMOTED list contain only the origin and the destination. The ENODES and CURRENT lists are initially empty.
2. For every node i in the PROMOTED list, check if *layer* (i) = $x - 1$. If so, retrieve its E-nodes and add them to the ENODES list. Connect node i with each of its E-nodes through a pseudo link, whose travel cost is heuristically estimated as follows.

- (a) When the route computation is based on the minimum travel distance criteria, the heuristic estimate of the travel distance between two nodes located at coordinates (x_1, y_1) and (x_2, y_2) is given by:

$$D_h = \max (|x_1 - x_2|, |y_1 - y_2|) + \min (|x_1 - x_2|, |y_1 - y_2|) / 2 \quad (3)$$

- (b) When the route computation is based on the minimum travel time criteria, the heuristic estimate of the travel time is given by:

$$T_h = D_h / V_l \quad (4)$$

where D_h is the heuristic estimate of the travel distance and V_l is the average travel speed for the roads in the layer l , such that $layer(i) = l$.

3. Add the nodes in the ENODES list to the PROMOTED list and empty the ENODES list.
4. For every node i in the PROMOTED list, check if $layer(i) \geq x$. If so, add node i to the CURRENT list.
5. Check if x is the highest layer. If so, find the tentative route between the origin and the destination by applying the shortest path algorithm to the highest layer and proceed to Step 9.
6. Check if all the nodes in the CURRENT list fall in the same grid in the layer x . If so, find the tentative route between the origin and the destination by applying the shortest path algorithm to the grid and proceed to Step 9.

7. Check if all the nodes in the CURRENT list fall within any two adjacent grids in the layer x . If so, find the tentative route between the origin and the destination by applying the shortest path algorithm to the union of the two grids. Go to Step 9.
8. Empty the CURRENT list. $x = x + 1$. (Next higher layer). Go to Step 2.
9. For every pseudo link in the tentative route, do the following:
 - (a) Identify its start node n_1 and end node n_2 .
 - (b) Determine the highest layer at which both n_1 and n_2 are present. Both the nodes will be present in the same grid because either n_2 is the E-node for n_1 or vice-versa.
 - (c) Apply the shortest path algorithm to the grid to find the optimal route between n_1 and n_2 .
 - (d) Replace the pseudo link with the optimal route between its start and end nodes.
10. Report the final route between the origin and the destination.

4.3 An Optional Enhancement

In the hierarchical routing algorithm explained above, the cost of travelling between a node in the lower layer and its E-node in the higher layer is represented using a heuristic estimate. The rationale for doing so is that the choice of the final route between the origin and the destination is mostly influenced by the actual travel costs of the links in the higher layer,

where the routing takes place. The heuristic estimate does not take into account the dynamic nature of the traffic condition between a given node and its E-node. Therefore, it is possible that in some cases, the algorithm will grossly underestimate the actual travel cost of the optimal route represented by the pseudo link. This will result in a route other than the best possible route being offered as the solution. For instance, the final route provided by the algorithm may require the vehicle to reach an E-node through a congested route.

In order to overcome this problem, an optional improvement step could be incorporated into the algorithm after Step 9(c). After computing the actual travel cost between the two nodes connected by a pseudo link, it could be compared with the heuristic estimate. If the difference between the two is unacceptable (an appropriate threshold value should be assigned) then the tentative route in the higher layer should be recomputed after replacing the heuristically estimated travel cost of the pseudo link with the actual travel cost. This ensures that a better route is found if such a route exists.

5. EVALUATION

The road network of Singapore was used as a test bed to empirically validate the hierarchical routing methodology. The roads in the network are grouped by the Land Transport Authority (LTA) into a number of categories ranging from Category A to Category E. Category A refers to the expressways that typically have an average travel speed of 95 km/hr and Category E denotes minor roads with average travel speed of 20 km/hr. The expressways serve as the major transportation arteries that connect the various towns and regional centres in the island. The other roads in the network connect to the expressways through a number of entry and exit ramps without impeding the traffic on the expressways.

All evaluations were carried out using the Smallworld Geographical Information System (GIS) and its development language, Magik. After removing all the redundant information, the network contained 11742 nodes and 30108 links. The layer formation procedure described earlier was applied to the road network of Singapore resulting in three layers of abstraction. The value of N_{max} was chosen as 128. The maximum number of E-nodes for any node was chosen to be 4. The details of the Singapore road network after layer formation are presented in Table 1. It can be seen that the number of nodes in the highest layer remains below the chosen value of N_{max} . While the lowest layer (Layer 0) included all the roads in the network, the highest layer (Layer 2) consisted of the expressways (Category A) and a few other major roads belonging to Category B. The views representing the three layers for a portion of the Singapore network are presented in Figure 6.

5.1 Accuracy of the Hierarchical Routing Algorithm

The hierarchical routing algorithm was employed to find the best routes between 100 randomly chosen origin-and-destination pairs in the Singapore road network. The 100 samples were subdivided into four distance categories of 25 samples each based on the distance between the origin and the destination. The route computation was carried out based on the minimum travel-distance criteria and the routes found by the hierarchical routing algorithm were compared with the optimal routes. The A* algorithm was used to compute the optimal routes for each origin-and-destination pair. The results are tabulated in Table 2. On average, the hierarchical routing algorithm finds near-optimal routes, which are only 3.5% longer than the optimal routes.

5.2 Performance of the Hierarchical Routing Algorithm

Hierarchical routing algorithms do not guarantee the optimal solution and therefore, a larger gain in performance must be achieved in order to compensate the loss of accuracy. In the following, we compare the worst-case behaviour of the proposed hierarchical routing algorithm with a well-known shortest path algorithm.

Taking the Dijkstra's algorithm as a benchmark, the worst-case runtime complexity of the non-hierarchical algorithm is $O(n^2)$, where n is the total number of nodes in the network. In the hierarchical routing algorithm, route computations are performed only within the routable networks containing not more than (N_{max}) nodes. Considering the worst case, in a two-layer hierarchy, 3 route computations are required and 5 route computations are required in a three-layer hierarchy. In general, if there are x layers in the network, then $(2x - 1)$ route computations are required. Since we consider Dijkstra's algorithm as the benchmark, the determination of the optimal route in the routable network takes $O(N_{max}^2)$ time, which is constant. Therefore, the runtime complexity of the hierarchical routing algorithm is $O(x)$, where x is the number of layers. Assuming a constant abstraction ratio r for all the layers, it could be shown that the number of layers x grows logarithmically with the number of nodes in the network [3]. Thus the proposed hierarchical routing algorithm reduces the worst-case runtime complexity of the route computation process from $O(n^2)$ to $O(\log_r n)$ where r is the abstraction ratio and n is the total number of nodes in the network.

The hierarchical routing algorithm has been prototyped on a Field Programmable Gate Array (FPGA). Details of this implementation can be found in [20]. It was determined that on average, the hierarchical routing algorithm required 1.28 ms to compute a near-optimal route in the Singapore road network organised as a three-layer hierarchy as shown in Table 1. When compared with a fast implementation of an optimal shortest path algorithm (Dijkstra's

algorithm with Fibonacci heaps and node priority queue), the hierarchical routing algorithm was found to be about 70 times faster [20].

6. CONCLUSIONS

There are many factors that render the conventional shortest path algorithms unsuitable for route computation in large road networks. Algorithms based on the concept of hierarchical abstraction offer definite advantages when applied to road networks. In this paper, we have proposed a formal methodology for organising a given road network as a multiple-layer hierarchy. The procedures are generic in nature because they are not tailored towards any particular type of road network and could be applied to any road network irrespective of its size. We have presented a hierarchical routing algorithm that makes use of a novel heuristic layer-switching technique to eliminate superfluous route computations.

The procedure for constructing a multiple layer hierarchy was applied to a real road network of a large city resulting in a three-layer hierarchy. The proposed hierarchical routing algorithm was verified on this three-layer hierarchy. The algorithm finds near-optimal routes, which are only 3.5% longer than the optimal least-cost routes found by the non-hierarchical algorithm. The performance of the hierarchical routing algorithm compares favourably with the conventional shortest path algorithm. While the shortest path algorithm finds the optimal route in quadratic time, the hierarchical routing algorithm finds a near-optimal route in logarithmic time.

REFERENCES

1. Jacob, R., Marathe, M.V., and Nigel, K.: 'A computational study of routing algorithms for realistic transportation networks', Proceedings of the Second Workshop on Algorithmic Engineering, 1998
2. Zhan, F.B., and Noon, C.E.: 'Shortest path algorithms: An evaluation using real road networks', Transportation Science, 32, (1), pp. 65-73, 1998
3. Korf, R.E.: 'Planning as search: A quantitative approach', Artificial Intelligence, 1989, 33, pp. 65-88
4. Car, A., and Frank, A.U.: 'General principles of hierarchical spatial reasoning – The case of wayfinding', Proceedings of the Sixth International Symposium on Spatial Data Handling, 1994, 2, pp. 646-664
5. Liu, B.: 'Intelligent route finding: Combining knowledge, cases and an efficient search algorithm', Proceedings of the 12th International Conference on Artificial Intelligence, 1996, pp. 380-384
6. Yang, T.A., Shekhar, S., Hamidzadeh, B., and Hancock, P.A.: 'Path planning and evaluation in IVHS databases', Proceedings of the IEEE Vehicle Navigation and Information Systems Conference, 1991, pp. 283-290
7. Bhagavatula, K.K.: 'An approximate shortest path algorithm for hierarchical networks', Master of Science Thesis, MIT, 1997
8. Quek, K.H., and Srikanthan, T., 'Improving hierarchical route computations for roadway networks', Proceedings of the IEEE Intelligent Transportation Systems Conference, 2001, pp. 112-117
9. Tatomir, B., Dibowsky, H., and Rothkrantz, L.J.M.: 'Hierarchical routing in traffic networks', Proceedings of the 16th Belgian-Dutch Conference on Artificial Intelligence, October 2004, pp. 75-82

10. Timpf, S., Volta, G.S., and Pollock, D.W.: 'A conceptual model of wayfinding using multiple levels of abstraction', *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Springer-Verlag, 1992, pp. 348-367
11. Jing, N., Huang, Y., and Rundensteiner, E.A.: 'Hierarchical encoded path views for path query processing: An optimal model and its performance evaluation', *IEEE Transactions on Knowledge and Data Engineering*, 1998, 10, (3), pp. 409–432
12. Jung, S., and Pramanik, S.: 'An efficient path computation model for hierarchically structured topographical road maps', *IEEE Transactions on Knowledge and Data Engineering*, 2002, 14, (5), pp. 1029–1046
13. Wang, Z., Che, C., Chen, L., and Lim, A.: 'An efficient shortest path computation system for real road networks', *Proceedings of the 19th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*, 2006, pp. 711-720
14. Shekhar, S., Fetterer, A., and Goyal, B.: 'Materialization trade-offs in hierarchical shortest path algorithms', *Proceedings of the 5th International Symposium on Advances in Spatial Databases*, 1997, pp. 94–111
15. Park, C. K., Sung K., Doh S., and Park, S.: 'Finding a path in the hierarchical road networks', *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 2001, pp. 25-29
16. Chou, Y. L., Romeijn, H. E., and Smith, R. L.: 'Approximating shortest paths in large-scale networks with an application to intelligent transportation systems' *INFORMS Journal on Computing*, Vol.10, No.2, Spring 1998, pp.163-179
17. Jagadeesh, G.R., Srikanthan, T., and Quek, K.H.: 'Heuristic techniques for accelerating hierarchical routing on road networks', *IEEE Transactions on Intelligent Transportation Systems*, 2002, 3, (4), pp. 301 -309

18. Habbal, M.B, Koutsopoulos, H.N., and Lerman, S.R.: ‘A decomposition algorithm for the all-pairs shortest path problem on massively parallel computer architectures’, *Transportation Science*, 1994, 28, (4), pp. 292–308
19. Zhao, Y.: ‘Vehicle location and navigation systems’ (Artech House, 1997)
20. Jagadeesh, G.R.: ‘VLSI architecture for hierarchical routing in dynamic route guidance systems’, Master of Engineering Thesis, Nanyang Technological University, 2002

Figure 1: An example of road collections in a country road network

Figure 2: Promotion of nodes to higher layer

Figure 3: Natural grids formed by roads in the higher layer

Figure 4: Combining multiple natural grids into one

(a) Origin (O) and destination (D) on the road network.

(b) O and D are linked to their E-nodes in the top layer.

(c) Two pseudo links form part of the route between O and D.

(d) Pseudo links replaced with optimal routes

Figure 5: Heuristic layer-switching technique

(a) Layer 2

(b) Layer 1

(c) Layer 0

Figure 6: Three-layer views for a part of the Singapore road network

Layer	Number of non-terminal nodes	Promotion limit	Number of grids
Layer 0	8475	-	194
Layer 1	293	2	8
Layer 2	118	10	-

Table 1: Layer-wise details of the road network of Singapore

S. No.	Distance category	Average travel distance of the optimal routes (m)	Average travel distance of the routes found by the hierarchical algorithm (m)	Excess distance. (Average) (m)	% of excess distance. (Average)
1	Very short	4440	4581	141	3.18
2	Short	9886	10371	486	4.91
3	Medium	19810	20606	797	4.02
4	Long	33256	34187	931	2.80
	All	16848	17436	589	3.49

Table 2: Accuracy of the hierarchical routing algorithm

Caption Sheet

Figure 1: An example of road collections in a country road network

Figure 2: Promotion of nodes to higher layer

Figure 3: Natural grids method formed by roads in the higher layer

Figure 4: Combining multiple natural grids into one

Figure 5: Heuristic layer-switching technique

Figure 6: Three-layer views for a part of the Singapore road network

Table 1: Layer-wise details of the road network of Singapore

Table 2: Accuracy of the hierarchical routing algorithm