Real-time Transportation Route Selection With Traffic Considerations

Ke Huang  
University of Massachusetts Lowell  
Computer Science  
Lowell, MA 01854  
khuang@cs.uml.edu

ABSTRACT  
In this project, we describe a real-time transportation route selection policy with traffic considerations. It is a dynamic policy that has a global perspective of the real-time traffic status. The A* search algorithm[4] is used to find the optimal route and the traffic status is the heuristic during the search. We created a map to simulate a real-word transportation network and a traffic file to simulate the traffic congestions. The route compared with the shortest path shows that our algorithm can provide us an optimal route selection, which costs less time and has a higher average speed, at most of the time if there are traffic congestions along the shortest path.

Author Keywords  
Transportation route, A* search, heuristic, traffic considerations

INTRODUCTION  
Nowadays, transportation has become more and more important in our modern lives. Especially for a commercial society, it is the vessel of the economy. The demands for traffic routing selection and driving navigation have increased dramatically. People not only care about the time, but also concerns about how much money they need to spend. It is a big challenge for a routing selection approach that whether the route it selects can save us time and/or money.

There are many traffic routing policies such as fastest or the shortest routing policy with local traffic optimization. However, the lack of global perspective might lead to a worst choice or even lead to traffic congestion. Since the traffic is dynamic, and any static policy is inherently inadequate. It follows that we need a dynamic policy with real-time traffic considerations. This dynamic policy has a global perspective of the real-time traffic status and it will always select a currently global optimal path from origin place to the destination. We can consider the route selection problem as a search problem, and our goal is to find a path from the origin to the destination which costs us less time and/or money. We will introduce the A* search into this project that use the traffic status as a heuristic.

PROJECT DESCRIPTION  
If we regard a map as a graph that each place is a vertex and all roads are edges in the graph, then this problem can be seen as a graph search problem that we need to find an optimal path form one vertex to the other. There are two things we need to do. The first thing is to find all feasible paths from the origin to the destination. Secondly, to select an optimal path among all feasible paths considering with the real time traffic situations. Each time the traffic status changes, we will have a new graph search problem that can provide us an optimal sub-routing. Hence, the approach is dynamic during the entire searching procedure until we have arrived the destination.

As we know, Google Maps provides transportation navigation service on line, which called Google directions, and real-time traffic services on line. Before we start our detailed analysis, let us see two figures that captured from Google Maps. The Figure 1 showed below is a route selected by Google Maps from place A to place B which the distance is 0.6 mile. And the real traffic flow colored by red indicates there was a traffic congestion on road 110. The route provided by Google Maps needs to go through road 101. While Figure 2 is an alternative route from A to B that the distance is also 0.6 mile but avoided the traffic jam on road 110. Obviously, the second route is a better choice for us. From this case we can see that if we can be informed of the traffic, we can alternatively select another route with traffic consideration and this route can provide us a better navigation.

Problem Analysis  
We have mentioned that a route selection problem is a graph search problem. And we can model the real-time transportation route selection with traffic considerations problem as an informed search problem, specifically an A* search problem[4]. Since we are not only trying to find a shortest path between two vertices in the graph, but a path that will cost us less time and money. Intuitively, the traffic situation is the heuristic in our search. In order to model this problem, we need to setup data structures for each road and for traffic congestions. We can abstract a road with four properties which are Node A, Node B at the two ends of the road, length of the road s and the speed limitation or average speed \( v_a \) of the road. We denote these four properties as \( \{A, B, l, v_a\} \). In a same sense, a traffic congestion has six properties and it can be denoted as \( \{A, B, l, v_a, s, v_r\} \). In the traffic jam denotation, the first four entities are used to specify the different roads that there may exist more than one road between node A and B. And s denotes the length of the traffic jam, while \( v_r \) indicates the real-time average speed of the traffic flow.
Usually, we have two options to do the route search, one is to find a shortest path between two places[1] and the other is to find a fastest route between two places. In this project, we choose the shortest path strategy. Since in most of the time, a shortest path may also be a fastest path. Like all shortest path searching problems, we want to find a shortest path between two places, but we also need to arrive the destination as fast as possible. Hence, the length between two nodes can be used as the cost function.

we define the heuristic function as follow.

\[ h = \left( \frac{s}{v_r} - \frac{s}{v_a} \right) \times v_r \]

\[ = \frac{v_a - v_r}{v_a} \times s \]

In this function, \((s/v_r - s/v_a)\) indicates how much extra time we need to spend if there is a traffic jam of length \(s\). And the heuristic function converts this extra time to an extra distance by multiply speed \(v_r\). Then we successfully transformed the time consideration into a distance consideration. Intuitively, if there is no traffic congestion, and we can save time \(t\) compare to a traffic jam, then we can drive \(v_a t\) distance that closer to the destination. Look at it in reverse, if we have to wait time \(t\) because of the jam, then it equals that we have to drive \(v_r t\) more to go to the destination. Since \(v_a > v_r\) in all cases, we are never over estimated. Especially, if \(v_r = 0\), the waiting time is infinity but the heuristic function gives us only \(s\). It follows that the heuristic function is admissible. Thus, the total cost in the A* search could be:

\[ c = l + h \]

\[ = l + \frac{v_a - v_r}{v_a} \times s \]

Since traffic status is dynamic and we need to update the real-time traffic periodically, we need to do the A* search periodically as well. Basically, it is a greedy algorithm that it always finds an optimal path based on current position if the traffic changes rapidly. But if the traffic changing is not rapid, it is optimal in global since it always selects the best choice in each search procedure guarantees it to be optimal.

It seems so good till now. But this approach is not perfect. Since we cannot predict the traffic for a long time and it only consider the current traffic status, if the traffic changes dramatically each time, the route selected may be a very bad selection that it may leads us haunting around the destination. But this probability is very low in real word that a traffic congestion cannot disappear in a very short time and even if we choose the traditional shortest route, we have to take the risk to wait a long time as well. The analysis of this circumstance is out of this project.

Data Set
As a web application, the data of Google Maps is not available for us to download it to local. We cannot implement the project by using real map and traffic data from Google. So, we created a map by using the definition of four properties in the previous section. It is tentative and ambitious that it only focuses on this project. And there have some function to interpret the data and can convert it into internal data structure.

On the other hand, we also need to simulate the traffic jam. We created another file to store the real traffic information. Since the traffic is dynamic and update periodically, we need to implement a mechanism to simulate the changes of each period in a single file. There are two dashes followed by a space and a number to indicate the real-time traffic during time period of that number. For example, "– 3" means the followed traffic status is only valid during the 3rd time period till we find another double-dash. The time period starts from 1. If there is a line only specified the number of time period, it means the traffic has not changed that it’s the same status as the previous time interval. While if there missed some number, it means there is no traffic jam in that time period.

Deliverable And Demonstration
There are two components of the delivery. One is the abstracted map information and the simulated real-time traffic
the delivery should be versatile that it can be designated with different maps and different traffic situations. The file name of the map and traffic can be assigned as parameters when calling the simulator. It provides an interface to deal with different scenarios. The interval of updating the real-time traffic can also be changed. When there is a traffic updating, the A* searching engine update the route selection as well.

In order to test whether the route selected by our simulator is optimal, we print out the traditional route’s time cost and length that select by the shortest path approach and the route selected with traffic considerations. The length, the time costs of the route and average speed can provide us enough information about the route we have selected. We want the route selected to be as short as possible and the time cost should be as few as possible. It implies a higher average speed is a better selection.

ANALYSIS OF RESULTS
The application can provide us a total time cost and the total distance from the origin to the destination of both the shortest path search and real-time search with traffic considerations. An average running speed will be calculated as well. We can compare the time cost and average speed between two algorithms. A higher average speed and less time cost can be a good guideline of evaluation.

Shortest Path Route
We choose the shortest path route as the control route to evaluate the real-time route. Since the shortest path strategy is the most used navigation policy. Once the origin and destination are specified, it always provides us a shortest path between two places. But the time cost varies in different traffic conditions. A traffic jam can affect the time cost along the entire route. In our tests, if the shortest route has no traffic congestion, then it is the optimal route which has a less time cost and a higher average speed. And it may do better than the real-time route which we will discuss this later.

Real-time Route With Traffic Considerations
A lot of tests have been done and the test results showed that the dynamic route selection algorithm can provide us a better route most of the time. But if the traffic status changes rapidly it may give us a lower average speed and higher time cost route. This is the case that the control route do better than the real-time dynamic route.

There are two reasons for this case. This first reason is the traffic file cannot give us a real simulation of real traffic congestions. In our model, we simplified a traffic jam by a certain length of road and a car can only run on it with a low speed. It means the length of the jam and speed are both constants. But in real word, they are dynamic and the changing interval is dynamic as well. In our module we just simply calculate the distance that has no traffic jam and then evaluate the traffic jam. If the traffic changes too rapidly, then the influence of the traffic jam may do little about the heuristic. Since the heuristic cannot provide a plausible hint of the route selection.

The second reason is that we cannot predict the traffic that a current optimal route may have a severe traffic jam. In some specific case, the real-time dynamic algorithm provides an optimal route in this traffic interval, but next traffic interval there may have a severe traffic congestion in the route. So, the algorithm needs to re-calculate the route and may have to go round the destination. At the same time, the traffic jam may disappeared in the shortest path and along this route we can arrive the destination much quicker. This is the worst case that may happen and lead the real-time dynamic route to be not optimal.

DISCUSSION
The Choose of Control Route
Obviously, for a route search problem we have two major options to evaluate the route, one is the total distance and the other is the total time cost. We chose total distance to be a evaluation function in this project. There are two reasons that we do not choose the total time cost to be the evaluation function. First, the time cost depends on the length of the route and the speed that both of them could be dynamic. And second, if there is a traffic congestion, we cannot evaluate how long will the congestion be. Then it is implausible to use the time cost as the evaluation variable.

On the other hand, the total distance is a static variable once we have selected a route. In most of time, a shorter distance implies a less time cost. We use the expected speed of a road and a traffic jam to calculate the expected time cost, and this expected time cost becomes a heuristic during the route selection. This approach successfully considered both of the total distance and total time cost. Hence, the average speed can always higher than the shortest-path route.
Broader Impacts
Though this project is only simulates to find an optimal route with traffic considerations, the idea can be applied to different areas of real problems. Its impact could be broad. We will illustrate some applications that can be used in our real lives.

- Driving navigation. The driving navigation services that most available for us only considering the distance. The lack of global perspective of traffic conditions fails to provide a more realistic optimal route.
- Avoiding traffic jam and traffic load balance. One of the most significant drawback of current direction service is it always provides a route that contains main roads or highways. It makes the traffic on those roads are very busy. And if there were an accident, the traffic jam is unavoidable. While the real-time transportation route selection with traffic considerations will load balance the traffic flow. It will also be useful for transportation department to schedule traffic lights dynamically.
- This approach can be applied to other networks like telephone or wireless network and Internet network. Especially for Internet network, the backbone routers are always highly loaded. For example, most of people are watching a basketball game online and all routers on the critical route to the server are highly loaded. Because the route protocols does not consider the current package flow and it will always select a critical route to the server. If we can apply this idea to this scenario, we can load balance the package flow and every one can have a good time to watch the game.

Related Work
The A* search is widely used in the graph or tree search. In this project, it evaluates nodes by combining the cost to reach the node and the cost to get from the node to the goal. Hence, if we can provide a reasonable heuristic function, then the A* search is both complete and optimal. The test results showed that our heuristic function is admissible but not consistent all the time. Because we used traffic as our heuristic and it is dynamically changing, it could not guarantee to be consistent all the time. As long as the traffic condition does not become worse along the real-time route, the A* search can give us a globally optimal selection. Since it satisfies the optimal structure[1]. Except the A* search, there are some other algorithms and approaches in the area of route selection. And these approaches solved the problems that the traffic changes frequently and we are not informed about the traffic condition in the future.

One most used algorithm is called "Genetic Algorithms". Hitoshi Kanoh was working on the dynamic route planning for car navigation using virus genetic algorithms[2]. His solution is using a genetic algorithm adopting viral infection. The method is to use viruses as domain specific knowledge. A part of an arterial road is regarded as a virus. A population of viruses is generated in addition to a population of routes. Crossover and infection determine the near-optimal combination of viruses. When traffic congestion frequently changes during driving, an alternative route can be selected using viruses and other routes in the population in a real time. This approach solved the problem of A* search that if the traffic congestion frequently changes, the route is not optimal.

Another approach formulate the problem as a dynamic multi-objective problem[3]. There are three objective functions to optimize simultaneously in the problem: route length, travel time that changes rapidly with time, and ease of driving. The method gives the Pareto-optimal set by using both the predicted traffic and a hybrid multi-objective GA (GA + Dijkstra algorithm) so that a driver can choose a favorite route after looking at feasible ones. This hybrid method works very well according to the test results. Since the uncertainty of traffic is the main problem of route selection, if we can predict traffic, the search is well informed and the result can be much optimal.

CONCLUSIONS
In this project, I demonstrate a real-time transportation route selection approach by using A* search algorithm. Experiments in dynamic environments using a simulated map and traffic show that the proposed method is superior to the shortest path algorithm in car navigation. But this method still has some disadvantages. It is not optimal if the traffic changes frequently or the traffic becomes worse along the selected route.

ACKNOWLEDGMENT
The work described in this paper was conducted as part of a Fall 2010 Artificial Intelligence course, taught in the Computer Science department of the University of Massachusetts Lowell by Prof. Fred Martin.

REFERENCES