Efficient Modified Bidirectional A^* Algorithm for Optimal Route-Finding*

Taeg-Keun Whangbo

Dept. of Computer Science, Kyungwon University
Seongnam-Si, Gyeonggi-Do, Korea
tkwhangbo@kyungwon.ac.kr

Abstract. A* algorithm, a kind of informed search, is widely used for finding an optimal car route, because the location of starting and ending point are known beforehand. Unidirectional A* algorithm guarantees an optimal route but requires considerable search time. On the other hand, bidirectional A* algorithm, usually known faster than unidirectional A*, does not guarantee the route found to be optimal, if the search ends when the forward and backward search meet in the middle. It may even take longer than unidirectional search to find an optimal route. In this paper, a new modified bidirectional A* algorithm which takes less search time and guarantees an optimal route is proposed. To evaluate the efficiency of the proposed algorithm, several experiments are conducted in real urban road environment and the results show that the algorithm is very effective in terms of finding an optimal route and search time.

Keywords: Heuristic search, Car Navigation System, Optimal route, A* search.

1 Introduction

Another change brought by current digital technology in our everyday life is that the printed road map for drivers is getting obsolete. The digital equipment, known as CNS (Car Navigation System), is rapidly replacing the printed maps. In CNS, one of the fundamental functions required is to present an optimal route minimizing the necessary time to the destination.

Finding an optimal route requires considering many factors such as travel distance, traffic condition, etc., and those factors can be represented as some cost function which affects the optimal route. Then the subject is generalized as the search problem for finding a path with minimal cost between source and destination.

Because this kind of problem has been occurred in many different application areas for long time, many appropriate search solutions are developed. Dijkstra method, for example, is a traditional one, and A* algorithm is well-known as an efficient method in finding an optimal route with minimal cost [1].

* This research was supported by the Ministry of Education, Seoul, Korea, under the BK21 project.

H.G. Okuno and M. Ali (Eds.): IEA/AIE 2007, LNAI 4570, pp. 344–353, 2007.

A* algorithm, which combines the advantage of uniform-cost search and greedy search, can expand the next node in either unidirectional manner or bidirectional manner. Unidirectional A* algorithm searches forward until the goal state is reached. It guarantees an optimal route, although considerable searching time is required. Bidirectional A* algorithm is to simultaneously proceed both forward from the initial state and backward from the goal, and stop searching when the two searches meet in the middle [2]. In such a condition, bidirectional A* is usually faster than unidirectional A* in finding a route, but does not guarantee the route found optimal. Although the route found in *bidirectional* may not be optimal, it is still appropriate for some application due to the reduction of time complexity. When the optimality is a critical factor, the above stop condition for bidirectional is not enough, but needs more extra steps. These extra steps make bidirectional A* take longer search time and take twice more time than unidirectional in worst case.

In this paper, we proposed a very efficient modified bidirectional A* which always guarantees the optimality and takes much less search time comparing the other modified A* algorithm. To evaluate the time complexity of the algorithm proposed, several experiments using the real urban road data are conducted and it shows the proposed algorithm in this paper is very efficient and reliable.

2 Related Research

Finding an optimal route given source (s) and destination (t) in a directed graph is a traditional graph searching problem, and many various solutions have been suggested. Dijkstra's algorithm is the classical one, with a worst-case running time of O(m+nlogn) using Fibonacci heaps. It is common practice to improve the running time of Dijkstra's algorithm heuristically while maintaining the correctness of the solution. Recent algorithms, with better running time that solve variants and special case of the shortest-path problem with better running time, are surveyed in [2], [3]. Each solution differs more or less in terms of time complexity, memory usage, etc.

If the search is guided by heuristics, it is called a heuristic search. Heuristic search normally deals with unidirectional approaches, which start from source(s) heading towards goal/destination(t). When there is one goal explicitly given, then bidirectional search is possible, which proceeds both in the forward direction from s to t and in the backward direction from t to s simultaneously. In backward searching, the node expands to the next node when there is an inward link. For example, assume there is a directed link from x_1 to x_2, but no link from x_2 to x_1, which is one-way street. In that case, if x_1- x_2 is a part of link in the optimal route, then x_2 can expand to x_1 in backward search. But in reverse case, which is no link from x_1 to x_2 but a link from x_2 to x_1. Then x_2 can not expand to x_1, although there is a link from x_2 to x_1.

A* algorithm, a traditional BFS (Best First Search), is widely used for an optimal route finding [4]. The cost function used in A* is:

$$f(n) = c(n) + h(n)$$