Advances in Intelligent Systems Research, volume 134
2nd International Conference on Control, Automation, and Artificial Intelligence (CAAI 2017)

Application of Heuristic Algorithm to Computer Based Interlocking Route Search

Li Ding[1,*] and Shenghua Dai[2]
School of Electronic and Information Engineering Beijing Jiaotong University Beijing, China
[*]Corresponding author

***Abstract*—In the Computer Based Interlocking (CBI) System of railway stations, the route search sub-module is one of the most important components, since its operating efficiency, reliability and safety can directly ensure the safety of train routes and shunting routes. This paper took an actual layout of railway station signal into consideration. By means of modeling the layout into a topological structure, we abstracted it as a directed graph. Then under the direction of path search in graph theory, we applied A* heuristic search algorithm to carry out the route search and it worked efficiently.**

Keywords-computer based interlocking; route search; heuristic search algorithm

I. INTRODUCTION

The concept "Route" we use in railway system includes "train route" and "shunting route". It refers to paths that trains or shunting trains run through in the stations. Only under the condition that the route is safe and reliable, trains can enter the route. To establish a route, generally the CBI system has to go through the following process: operation phase, route selection phase, switch operation phase, route locking phase and the final step, clearing a signal. [1] The route search solution provided by this paper is mainly related with route selection phase. To be more specific, the key discussion raised in this paper is about making computer software automatically select involved signals, tracks and switches in a specified area determined by operation phase. At present, there are two commonly used methods of route search: dynamically generating route and selecting route out of route sheet. However, the second method still requires the support of a kind of route search algorithm to form the route sheet. Of course, the route sheets can also be generated completely manually, but the efficiency of this method is very low. Therefore, the best way, under the current circumstances, is to use computer-aided-design (CAD) method to generate the route sheet, which means using CAD method to design a complete, mature and universal route search program, and then to generate the route sheet. After these steps, the only thing left is manual checking, to see whether the sheet meets all interlocking requirements. [2] Without any doubts, this method can not only reduce the labor intensity, but also improve the efficiency and reliability of CBI software.

Researches on route search methods now are mainly about simplifying the layout of station signal and putting it into a general graph according to its natures and features. Then we can use the path search in graph theory to solve the problem of route search in the CBI system. The staple route search algorithm includes improved depth / breadth first search algorithm, double breadth first search algorithm, Dijkstra search algorithm and K steps algorithm combined with cost matrix diffusing along direction of minimum cost. [3]

A* algorithm is a kind of heuristic graph search algorithm. It can expand the node closer to the goal node according to its heuristic function value, so compared to the algorithms mentioned above, it has higher search efficiency. Based on the analysis, we chose A* heuristic search algorithm as the route search strategy, then with proper improvements, we designed a reasonable heuristic function which was applicable to CBI route search.

II. ANALYSIS AND TOPOLOGY MODEL OF THE LAYOUT OF STATION SIGNAL

A. Analysis of the Layout Graph of Station Signal

Signal equipment in the railway station mainly included various signals, switches, tracks, insulated joints, and insulated joints located within a clearance. Moreover, any signal equipment keeps physical contacts with those before and after it. A typical layout of station signal is shown in Figure I:

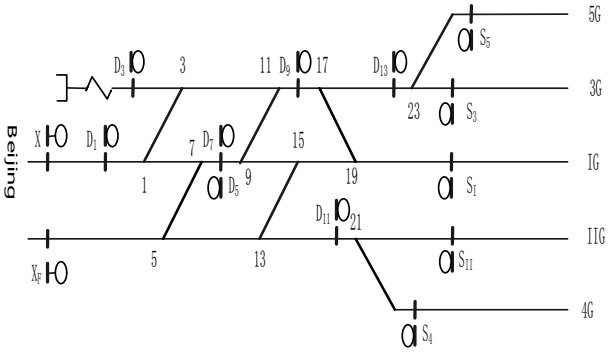

FIGURE I A TYPICAL LAYOUT GRAPH OF STATION SIGNAL

First, based on the characteristics of signal equipment in the railway station, the layout of station signal could be simplified and abstracted. Handling route was just like playing chain reaction. Once the start button and terminal button were pressed, the system will search the route automatically according to corresponding route command. A route command mainly included related signals, tracks and switches (their

positions included). Therefore, if we just retained the characteristics of the signal layout, and considered the equipments as nodes, their connections as edges, then we got the graph G. Thus the path search strategy in graph theory could be applied here in an efficient way. Due to the directionality of the route, the graph G could be abstracted as a directed graph G'. Took the main signal equipments as nodes of directed graph and the formal definition described as following: took common signal as one node, differential signal and double signal as two nodes; considered single-acting switch as one node and double-acting switch as two nodes; regarded the connections among these nodes as edges of the graph.

B. Topology Model

Based on the above definition, the typical layout of station signal could be described in the following topological graph of Figure II:

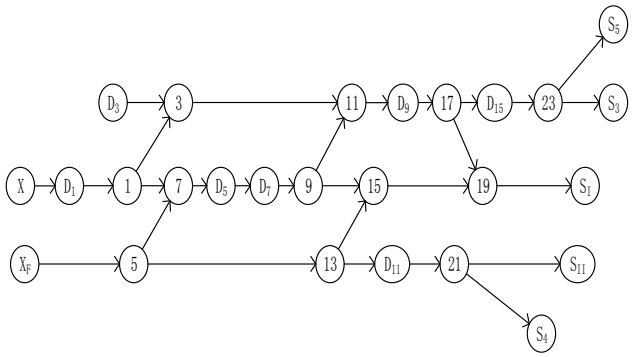

FIGURE II TOPOLOGICAL GRAPH

C. The Data Structure of Topology Graph

Obviously, the computer was unlikely to directly understand the information in FIGURE II, so it was necessary to select a kind of data structure to store the topological graph information. There were two kinds of commonly used storage structures: adjacency matrix structure and adjacency list structure. [4] After analyzing, we known that in topological graph-G', there were at most two child nodes after each node, which was classified as a kind of sparse graph. It meant that the storage rate was relatively low if we use the adjacency matrix structure. Therefore, we chose the adjacency list structure to do the job.

Next, the adjacency list structure will be described in details as follows:

The edge node class ArcNode of adjacency list is defined as follows:

```
public class ArcNode
{
    public Vertex adjvex;
    public ArcNode next;
    public ArcNode(Vertex value)
    {
        adjvex = value;
    }
}
```

The head node class Vertex of adjacency list is defined as follows:

```
public class Vertex
{
    public string name;
    public int x_ location, y_ location;
    public ArcNode firstEdge;
    public bool visited;
    public Vertex(string value, int x, Int y)
    {
        name = value;
        x_location = x;
        y_ location = y;
    }
}
```

The class Vertex corresponded to the nodes in topological graph, i.e. the signal equipment in the layout of station signal. It consisted of name, occupancy mark, relative position in coordinate (x_ location, y_ location), etc. This position coordinate was very important to the calculation of the function valuation in the later heuristic search algorithm.

III. ROUTE SEARCH STRATEGY

A. A Search Algorithm*

In 1968, the reference [5] first proposed A* algorithm. It was one of the most effective and direct search methods to solve the problem of finding the shortest path in static road network, and it was also an effective algorithm to solve many other search problems. The advantage of this algorithm was that it could firstly use evaluation function to evaluate the location of each search node in the search space and then chose the best node to expand. This approach avoided the trouble of traversing all nodes like the depth first search or breadth first search and therefore it could ignore unnecessary search nodes, so the search efficiency could be improved. The evaluation of the node required an evaluation function. Here we used the following $f(n)$ represented as:

$$f(n) = g(n) + h(n) \tag{1}$$

In the function:

$f(n)$: The actual cost of the optimal path from the start node to the goal node through the intermediate node n;

$g(n)$: The actual cost of an optimal path from start node s to intermediate node n;

$h(n)$: The actual cost of an optimal path from intermediate n to a goal node.

It was clear that $g(n)$ and $g(n)$ cannot be known in advance before finding the optimal path and $f(n)$ was not

known a priori, so we needed to use their estimates to describe the evaluation function, as follows:

$$\hat{f}(n) = \hat{g}(n) + \hat{h}(n) \tag{2}$$

in which:

$\hat{f}(n)$: An estimation of $f(n)$, also called evaluation function, i.e. the actual cost from the start node to the goal node via the node n by an optimal path;

$\hat{g}(n)$: An estimation of $g(n)$, the cost of the path from the start node to the node n having the smallest cost so far found by the algorithm. Notice that this implies

$$\hat{g}(n) \geq g(n) \tag{3}$$

$\hat{h}(n)$: An estimation of $h(n)$, also called heuristic function, any estimation of the cost of an optimal path from node n to a goal node; it depended upon the information from the problem domain.

PETER E. HART had proved that A* is admissible when the following condition was in place, that was for each intermediate node n, there is

$$\hat{h}(n) \leq h(n) \tag{4}$$

Which meant the algorithm could find an optimal path. Therefore, we could always find an optimal search path when the requirement above was met. When dealing with practical problems, we had to design an appropriate evaluation function according to their characteristics of problem.

B. Design of Evaluation Function

From the above description, we known that the key of algorithm lied in the design of heuristic function. Commonly used heuristic function usually required the assistance of real-valued function of $\hat{h}(n)$, such as the Euclidean distance, Manhattan distance, diagonal distance and so on. And then, by practical experiments, we got a reasonable heuristic function. By analyzing characteristics of the layout of signal, we found that when the search met switch nodes, there were two child nodes. It meant that we had two different path choices, the route searching along the straight track or the bending one. Obviously, the route searching direction had great relationship with the ordinate difference between intermediate node and the goal node. We marked the coordinates of start node $S(x_s, y_s)$, the coordinates of intermediate node $n(x_n, y_n)$, and the coordinates of goal node $D(x_D, y_D)$. We could apply:

$$\hat{h}(n) = | y_D - y_n | \tag{5}$$

$$h(n) = \sqrt{(x_n - x_D)^2 + (y_n - y_D)^2}$$

Obviously, the designed function met (4), i.e. it met the requirement of the algorithm;

Then, we chose:

$$\hat{g}(n) = \sqrt{(x_n - x_s)^2 + (y_n - y_s)^2} \tag{6}$$

So we got the designed evaluation function:

$$\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$$

$$\hat{f}(n) = \sqrt{(x_n - x_s)^2 + (y_n - y_s)^2} + | y_D - y_n | \tag{7}$$

The latter algorithm used (7) to sort the nodes and select the optimal expanded node.

C. Route Search Process of A Algorithm*

The functions that the route search module realized were specifically described as follows: After the operation of pressing the start button and the terminal button, the program will select one basic route. Manual pressing option button was required in order to get the alternative route. In the process of route search, the start button and terminal button corresponded to two separate signals, and the signals in turn corresponded to the start node and the goal node in the algorithm. So the start button corresponded to start node “S”, the terminal button corresponded to the goal node “D” and the signal equipments that the route passed through corresponded to arbitrary nodes “n”. Then route search process of A* algorithm could be described as follows:

a) Established two tables “Open” and “Closed”. Put the start node S in the Open; the Closed was initialized empty;

b) Judged whether node S was the same with D; if yes, the route was invalid; otherwise next;

c) Put the start node S in the Closed; at the same time all the child nodes of the node S were generated; put them in the Open to detect;

d) Used evaluation function (7) to rank all nodes in the Open, so we got the node having the smallest value. Then added it to the Closed, at the same time all the child nodes were generated. Added to the Open the child nodes which had not been added to the Closed or the Open;

e) Repeat step (d) until the Open became empty, which meant the search failed; or the Open does not became empty,

but the process had already searched the goal node D, which meant the search succeeded;

f) If the search failed, it illustrated the present route was occupied. If the search succeeded, we traced back from the goal node D to the start node S in the Closed, and the search path could be obtained.

g) Analyzed the search path combined with the actual layout of station signal, and then generated the corresponding route command.

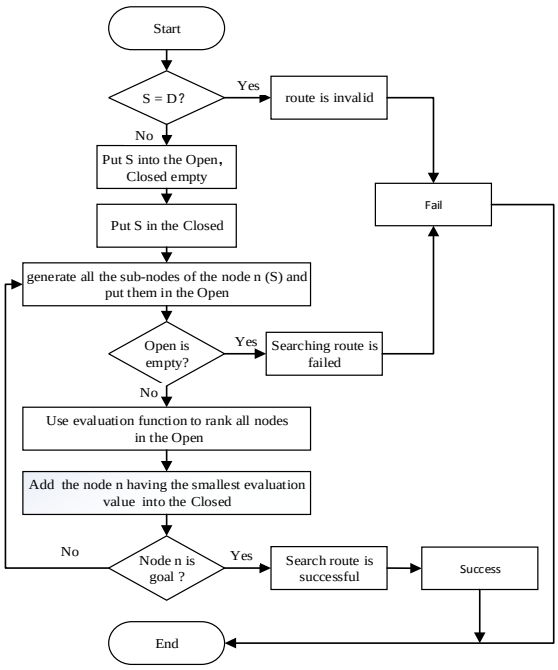

FIGURE III. A ROUTE REALIZED BY PROGRAM

IV. VERIFICATION AND ANALYSIS

We used C# language to write test software on the Visual Studio 2015 platform. Taking the figure I as an example. We chose any one of the train route or shunting route. The following was the analysis of the train route of Beijing down direction running from X to 5G.

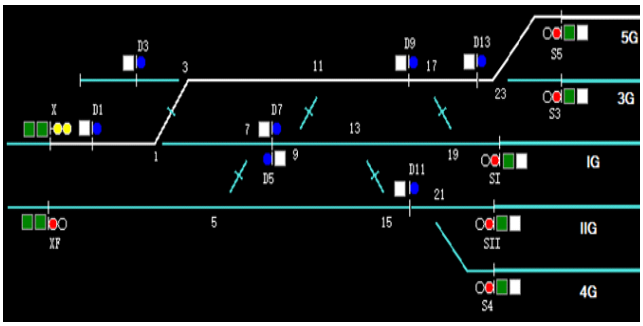

FIGURE III A ROUTE REALIZEED BY PROGRAM

Step 1: Pressed the start button (XJA) and the terminal button (S5LA) in turn, then the corresponding start node (X) and goal node (S5) in the figure 2 were identified, which meant the range of route search was also locked.

Step 2: Used A* search algorithm to search route. First, added the information of node X to the Open, and the Closed was initialized empty; then the search process was as follows:

Open = {D1}, Closed ={X};

Open = {1}, Closed = {X, D1};

Open = {3, 5}, Closed = {X, D1, 1};

Open = {5, 11}, Closed = {X, D1, 1, 3};

Open = {5, D9}, Closed = {X, D1, 1, 3, 11};

Open = {5, 17}, Closed = {X, D1, 1, 3, 11, D9};

Open = {5, D13, 19}, Closed = {X, D1, 1, 3, 11, D9, 17};

Open = {5, 19, 23}, Closed ={X, D1, 1, 3, 11, D9, 17, D13};

Open = {5, 19, 23, S5, S3}, Closed ={X, D1, 1, 3, 11, D9, 17, D13, 23};

Open = {5, 19, 23, S3}, Closed ={X, D1, 1, 3, 11, D9, 17, D13, 23, S5};

Step 3: The software program generated the corresponding route information by analyzing the nodes information generated in the Closed;

Step 4: With the assistance of the route information, the related switches were in the correct position; after confirming the switches' position, locked the switches and hostile route to ensure safety; in the end, cleared a signal and gave a sign to let train enter the route.

The realization of train route that Beijing down direction ran from X to the 5G was shown in Figure IV. Because of side track receiving train, the home signal displayed double yellow. White belt represented the lock of the train route including the switches and tracks.

V. CONCLUSION

The paper analyzed the layout graph of railway station signal, abstracted it as a graph structure and then combined it with the heuristic path search algorithm in graph theory. On this basis, A* search algorithm was used to carry out route search. By doing that, we got a reasonable route within a relatively short time, and the result was pretty good. Compared with the traditional route search algorithm, the method we applied here had a relatively higher efficiency because of its smaller number of expanded nodes and good guidance of the algorithm. To deal with the situation that tracks are occupied, the system would rule out those nodes by marking corresponding nodes, so the search did not reach them to ensure safety; for the alternative route in the process, after pressing the option button, the route search would first followed the path from start node to alternative node and then

the path from the alternative node to the goal node. After analyzing both two paths, we got the alternative route information. The algorithm can not only be used to fulfil the task of dynamically generating routes in CBI system in railway stations, but can also help design route search programs to complete the route table combined with CAD technology.

REFERENCES

[1] Guo Jin, Railway signal foundation, 2010, pp. 147-153,China Railway Press

[2] Lin Yuyun, Lv Yongchang, Computer-based interlocking, 2013, pp. 43-50, China Railway Press

[3] Liang Yifan, Application of A* algorithm in computer interlocking software, 2013, pp. 3-5, Lanzhou JiaoTong University.

[4] Li Chunbao, Data structure tutorial, 2008, pp. 196-216, Tsinghua University press

[5] PETER E. HART, NILS J. NILSSON, BERTRAM RAPHAEL, A Formal Basis for the Heuristic Determination of Minimum Cost Paths, 1968, IEEE TRANSACTIONS OF SYSTEMS SCIENCE AND CYBERNETICS