Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

The Application of Graphic Theory on Railway Yard Interlocking Control System

Xiaoli SHE, Yun SHA, Qiang CHEN, Jian YANG, *Senior Member, IEEE*

***Abstract*—A most important function of railway yard interlocking system is to supply accessible and safe routes for train dispatching, namely route search. A railway yard can be modeled as a directed weighted graph according to the network of infrastructure and rule of train movement. Then the problem turns to a graph search problem. In this paper we presented two style of interlocking system both existed with different characteristics and supply areas. Then two graph search approaches, based on revised Dijkstra algorithm and heuristic search algorithm respectively, are adopted to satisfy the different demand for each system. This associated solution is presented to be well fitted for railway yard interlocking system. Experiment results based on real yard graph demonstrate its feasibility and effectiveness.**

I. INTRODUCTION

THE computer interlocking system used on railway yard is a key system to ensure the safety of train dispatch [2-4]. Since the earliest computer interlocking system supplied in railway at Goteborg, Switerland in 1978, and China National Railway Signal & Communication Corp. (RSC) developed the first computer interlocking system fitted in Chinese railway system in 1984, it has been improved and extended considerably in the latest 20 years [4]. With the fast development of high-speed railway lines and the extending of scales on railway transport hub yards, more advanced interlock control system is heavily needed with high safe level and improved performance [2].

A basic function of railway interlocking system is to find strictly correct and available route for train dispatch, to avoid train collisions and derailments [1]. Recent interlocking systems mainly adopt one of the implementation styles: called as route-list based style and route-search based style. The former one contains a set of static data consisted by all possible routes which are generated previously [2]. The latter one, contrarily, is embedded by real-time route-search process other than data copies [2, 3]. These two styles are both employed by different interlocking systems. The latter one holds stronger effectiveness, while the former one is still in much utilization as particular terrain and yard configurations in Chinese railway system.

For route-list based system, we proposed a new multi-route search approach based on revised Dijkstra algorithm to previously generate route-list data [5,6]. For a connected graph, Dijkstra algorithm offers an effective approach to search the shortest route. Moreover, many works have been done to improve the efficiency of Dijkstra algorithm significantly [7]. When used on route-list data build, one problem is that not only the basic route, but all alternative routes are all needed to be saved on interlocking system to defense the case that sometimes the basic route is busy, hence enhancing yard passing capability [4]. We find that there are some special features of the graph formed by railway station, in which a multi-route search method based on revised Dijkstra algorithm can be applied for searching all possible routes without extra searching cost. It can keep the high efficiency of Dijkstra algorithm, meanwhile satisfying the demand of railway interlocking system.

A more effective and accurate search method is needed for route-search based system as safety requirement and real-time implementation. In this paper, we employed a heuristic search approach, namely A* algorithm [5,8,9]. Effective heuristic function is defined by combining useful relative location of infrastructures, yard configurations and train movement rules. The framework of A* algorithm associated with appointed heuristic function can be proved theoretically that it always finds the destination fast, and even in worst cases it is capable to find destination node finally. This guarantees its correctness of this approach.

Finally, experiment results shows the feasibility and high effectiveness of our proposed approaches compared with other searching methods applied on interlocking systems.

II. ROUTE LIST AND ROUTE SEARCH

A railway yard is formed by infrastructures, including signals, sections, level crossing et al. Each infrastructure has local circuit to output its real-time status (busy or idle) and accept the operation command. We model the infrastructure network to graph, additionally considering the train's prescribed regulation such as single-direction rule and inertial-direction-first rule [1], then a directed weighted graph can be obtained, based on which route search naturally turns to be a shortest path searching problem.

Figure 1 shows a typical graph formed by a real station. Note that the direction and weight of arcs are not labeled.

Figure 2 shows the detail of certain area marked on Figure 1. Now we discuss how we change the route search problem as a short path problem on a simple case. In Figure 2, for example, we suppose a train needs to dispatch from D1 to D2. There are two alternative pathways as Figure 2 shows: D1->D3->D2 (Route A), and D1->D4->D2 (Route B). Based on the general "train moving rule" on the yard, the optimal

1-4244-1068-1/07/$25.00 ©2007 IEEE.

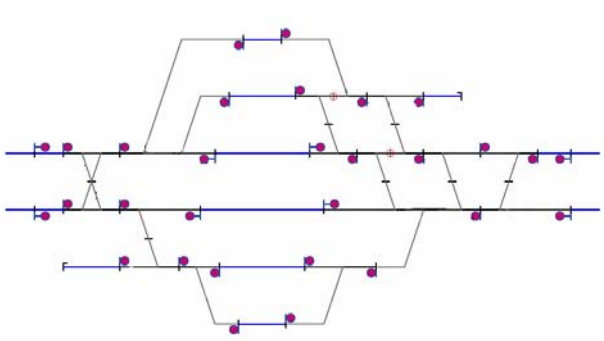


Figure.1 A Typical Graph of Railway Yard
Note: The red circles represent signal lamps used as the start (end) notes of train dispatching route; the blue lines represent main lines or end lines of the station. The marked area by yellow circle will be discussed further as an example.

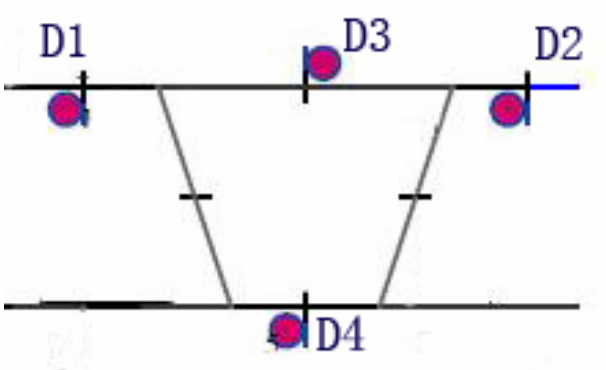


Figure.2 A Detailed Graph of Railway Yard
Dn denotes the name of signal lamps.

route would be straightly forward. So we expect our approach choose Route A as first choice. We achieve this goal by setting proper "cost" for each line to make the total "cost" of Route A less than Route B. Then we can solve route search problem by solving shortest path problem.

III. REVISED DIJKSTRA ALGORITHM ON ROUTE-LISTED BASED SYSTEM

Route-list based systems are now used on earlier interlocking systems and for some particular yards with anomalous structures. When system receives a command of train dispatching, it selects the appropriate route from route list, which contains all infrastructures ID with expected status along this route. The static database takes convenience of execution, while increasing the risk of data error as the large amount of total route-list data, and causing low efficiency by time cost of data correction check.

Obviously, the accuracy is most important factor that decides whether a searching method can be applied for generating route-list data. Then the efficiency is pursued based on high accuracy. The section below shows that revised Dijkstra algorithm would be fitted to this issue.

Dijkstra algorithm is the application of Greedy Algorithm on shortest path search [5,6]. It adjusts search strategy dynamically. On each loop of its searching course, it deals with the node with the minimum cost ever obtained, referred as "best-first-search". The searching process of Dijkstra algorithm shows as Figure 2. When a search process for one single source s is completed, we can trace back one by one to get the shortest path from s to each v according to the array F, which saves the father node along the shortest path. We called it as trace-back process. For a graph $G=(V,E)$ with $|V|=n$, $|E|=m$, and $n/m=[1,2]$, the time cost for complete search using Dijkstra algorithm is $O(n^2 \log n)$. Compared with Depth-First Search approach (DFS), which is generally used on route search [2], Dijkstra algorithm is of significant higher efficiency.

```
Procedure Dijkstra(G, s)
   d(s)←0;
   for (each v ∈ V - {s}) do
      d(v)← ∞;
   repeat
   S←∅; Q←V;
   while Q ≠ ∅
      do u←Extract-Min(Q)                  // extract-key
      S←S∪{u}
      for (each v connected with u) do
         if    d(v)>d(u)+w(u, v)
         then  d(v)←d(u)+w(u, v);F(v)←u;   // decrease-key
         endif
      repeat
   repeat
   output d(1:n),F(1:n)
```

Figure 2. Pseudo Code of Basic Dijkstra Algorithm

Here we introduce the proposed approach, the multi-route search method based on revised Dijkstra algorithm. In the "decrease-key" of basic Dijkstra algorithm shown as Figure 2, it replaces the prior route once finding a current route with less cost in every loop. In fact, these replaced ones contain information of alternative routes. In multi-route search approach, we keep the framework of Dijkstra algorithm while adding a "backup-key" to keep the useful information. The process of proposed algorithm is shown as Figure 3.

```
Procedure Dijkstra(G, s)
   d(s)←0;
   for (each v ∈ V - {s}) do
      d(v)← ∞;
   repeat
   S←∅; Q←V;
   while Q ≠ ∅
      do u←Extract-Min(Q)                  // extract-key
      S←S∪ {u}
      for (each v connected with u) do
         if    d_min(v)>d(u)+w(u, v)
         then  d_back(v)←d_min(v);F_back(v)←F_min(v);   // decrease-key
               d_min(v)←d(u)+w(u, v);  F_min(v)←u;
         else  d_back(v)←d(u)+w(u, v);F_back(v)←u;  // backup-key
         endif
      repeat
   repeat
   output d(1:n),F(1:n)
```

Figure.3 Pseudo Code of Multi-route search approach

After search process completed for a single source s, we

trace back along the father nodes recorded on each node. Other than one single route, we can get a "complete minimum graph", which only include all the nodes and arcs with available routes pass by. Figure 4 shows a complete minimum graph obtained from trace-back process. On this tree along with each father nodes recorded, we can get all alternative routes.

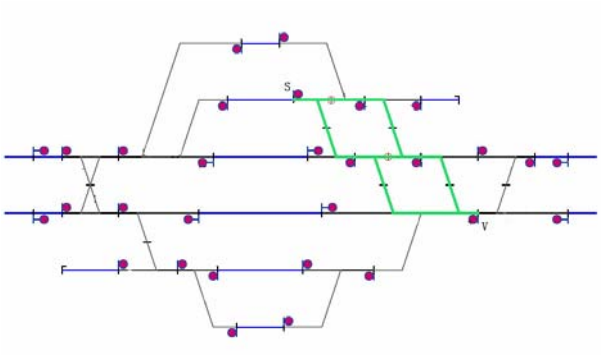


Figure.4 A Complete Minimum Graph:
The green bold line denotes the complete minimum graph form s to v. The searching direction denotes the direction of train movement.

Note that the multi-route search approach can not guarantee a complete searching for all available routes on a general graph. However, as the arrangement of infrastructures and the movement rules of train, there are some special characteristics for the station graph. Each note pair in a station, if connected, is joined by single infrastructure. In other words, each note pair has no more than one arc. Besides, train's movement follows single-direction rule, then the station graph can be a single-direction graph. On such graphs, the proposed approach is feasible to obtain all accessible routes.

IV. A* ALGORITHM WITH APPLICATION TO ROUTE-SEARCH BASED SYSTEM

For route-search based system, route data is generated and released dynamically. There are only infrastructure information and their connect relationship saved on interlocking system. As its high efficiency and flexible data structure, this style shows more potential than route-list based system. However, a key problem is to find an effective, accurate and robust searching approach embedded on interlocking system.

Here we proposed an effective heuristic search algorithm, namely A* algorithm for this issue. General heuristic search theory and A* algorithm are reviewed as follows.

Given a graph $G=(V,E)$, with Source node s and destination node v. $k^*(n_1,n_2)$ denote the cost from node n_1 to node n_2. Set $g^*(n)=k^*(s,n)$, denoting the actual cost from source s to node n; and $h^*(n)=k^*(n,v)$, denoting the actual cost from n to v, which can not be easily known until the end of searching process.

In the searching process, we estimate $g^*(n)$ and $h^*(n)$ with $g(n)$ and $h(n)$ respectively. Set $f(n)=g(n)+h(n)$ to estimate the actual cost $f^*(n)$. Then we put estimate cost function $f(n)$ in the framework of Dijkstra algorithm to replace $d(n)$ in Figure 2.

Note that $h^*(n)$ is hardly to know, then the heuristic function $h(n)$, which is used to estimate $h^*(n)$, is the key factor to determine whether a heuristic algorithm can work effectively. It can be proved that when a heuristic algorithm satisfies the follow conditions [5,6]:

a. There exists one route with minimum cost route;
b. The problem domain is finite;
c. All sub-cost between any nodes are positive;
d. $h(n)<h^*(n)$

Then such heuristic algorithm, which is called A* algorithm, can find optimized solution guaranteed theoretically.

To apply for the railway interlocking system, we define $h(n)$ as

$$h(n)=bFlag\times\sqrt{(pS.x-pD.x)^2+(pS.y-pD.y)^2}+(!bFlag)\times fCostMax$$

and $bFlag$ is determined by

$$bFlag = Dir \;\&\; ((pS.x-pD.x)\times Slope<(-|pS.y-pD.y|))$$

Note that $bFlag$ denotes whether point pS locates at inaccessible areas to pD, which is co-determined by the searching direction Dir and switch slope of certain yard $Slope$.

Figure 5 shows how this heuristic function works. $h(n)$ calculates the straight distance between pS and pD, which is definitely smaller than its actual cost , say $pS->pN->pD$, which is guaranteed by triangle inequality. Additional judgment of $bFlag$ decreases the searching scales more over to improve searching efficiency.

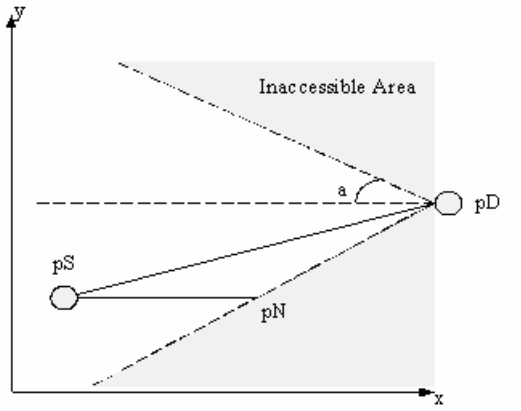


Figure.5 Sketch Graph of heuristic function

To sum up, A* algorithm with the heuristic function referred is well fitted for route search based interlocking system, as its accuracy by theoretical guarantee and its high efficiency by employing heuristic information.

V. EXPERIMENT AND FUTURE WORK

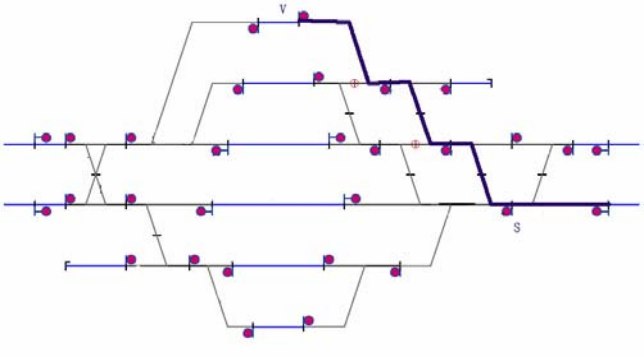
We choose the station ZhongJiaCun along railway line between Xi'an and Yan'an to demonstrate the feasibility and efficiency of the proposed approach. The route list now used on the real station, which has proved to be correct, is adopted as criteria. The station graph is shown as Figure 1. Results using DFS, Dijkstra algorithm, multi-route search approach and A* algorithm to generate route list respectively are shown as Table 1.

TABLE.1 EXPERIMENT RESULT ON ZHONGJIACUN STATION
Station Condition: Node: 41; Arc: 49; Available Routes: 143

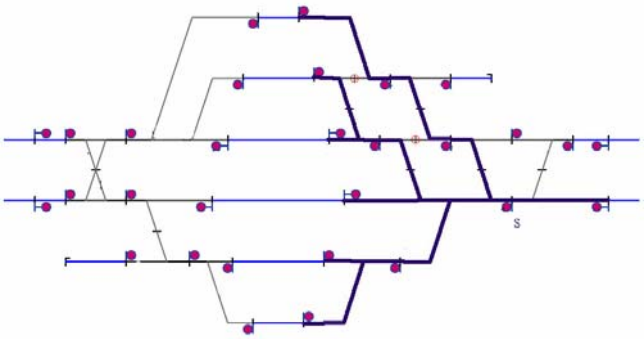
	Available Routes	Missing Routes	Node Settle Time
Criteria	143	0	
DFS	143	0	3206
Dijkstra	116	27	1057
Revised Dijkstra	143	0	1057
A*	143	0	1836

Dijkstra search approach is presented to be feasible to cover all available routes with smallest searching cost, and it's a proper method to generate the route-list data. The results well demonstrated its completeness and high efficiency.

Another example will shows the superiority of A* algorithm on the real-time route search environment, in which one source-destination node pair is appointed. Figure 6 shows the searching trace of Dijkstra algorithm and A* algorithm respectively. We can see that A* algorithm achieves the destination note very fast, while Dijkstra algorithm comes through more irrelative nodes. A* algorithm has its superiority on real-time route searching system as its high efficiency, while revised Dijkstra algorithm offers all routes started from S by once executing searching process.



(a) Searching trace from S to V by A* algorithm



(b) Searching trace of from S by Revised Dijkstra Algorithm
Figure 6. Sketch Graph of the searching track

Further work will be focused on robustness of the proposed approaches against various station structure and different graph size. For A* algorithm, an auto-test system is now being build to simulate various local yard cases and generate types of command, especially some illegal ones. Then A* algorithm is prepared to be tested based on the auto-test system to verify its robustness.

VI. SUMMARY

In this paper, two graph search approaches are employed to railway yard interlocking system. The accuracy of both approaches is theoretically assured, and their characteristics are well fitted to the particular requirement of different styles of interlocking system respectively. Experiment results show their high potential for the application on railway interlocking system.

ACKNOWLEDGMENT

This work was supported by the by Fundamental Research Foundation of Tsinghua University and Beijing Institute of Petrochemical Technology.

REFERENCE

.

[1] Railway Ministry Standard of P.R.China: *"Railway Signal design criteria"*. Beijing: China Railway Publishing House. 2006.

[2] Z.X. Zhao, *"Computer Interlocking system"*. Beijing: China Railway Publishing House. 1999.

[3] H.Z. Xu and Q. Yue, *"Computer Interlocking system and Signal Design"*. Beijing: China Railway Publishing House.2005.

[4] F. Wang, *"Discussion of Some technologies of Railway Yard Interlocking system"*. 2005

[5] Leiserson, C.E., *Introductions to Algorithms*. 2004

[6] Buckley, F. and M. Lewinter, *"A Friendly Introduction of Graphics"* 2004: Tsinghua Univ. Publishing House

[7] GOLDBERG, A. V. & TARJAN, R. E. *"Expected performance of Dijkstra's shortest path algorithm.*" Princeton, NEC Research Institute.1996.

[8] Hart, P.E., N.J. Nilsson, and B. Raphael, "*A formal basis for the heuristic determination of minimum cost paths in graphs.*" IEEE Transactions on Systems Science and Cybernetics, 1968

[9] Lester, P., *A* Pathfinding for Beginners*, 2003. http://www.gamedev.net/reference/articles/article2003.asp