Netw Spat Econ (2011) 11:193–205
DOI 10.1007/s11067-009-9098-7

Optimizing Train Network Routing Using Deterministic Search

KePing Li · ZiYou Gao ·
BaoHua Mao · ChengXuan Cao

Published online: 27 January 2009

Abstract In this paper, we use walk search strategy to solve the optimization problem of train routing on railway network. The proposed approach is a local search algorithm which explores the railway network by walker's navigating through the network. Using some selection rules, walker can dynamically determine the optimal route of trains. In order to analyze and evaluate the proposed approach, we present two computational studies in which the search algorithm is tested on a part of railway network. The results demonstrate that the proposed approach is an effective tool for optimizing the train routing problem on railway network. Moreover, it can be executed with shorter computation time.

Keywords Train routing · Deterministic random walk · Traffic demand

1 Introduction

The problem of train routing concerns the choices of routes for scheduling trains on railway network. Train routing problem is a highly constrained network optimization problem that has confounded many traditional and modern optimization methods. The objective is to minimize the cost from a

The project is supported by National Natural Science Foundation of China under Grant Nos 60634010 and 60776829, New Century Excellent Talents in University under Grant No NCET-06-0074 and the Key Project of Chinese Ministry of Education under Grant No 107007.

K. Li (✉) · Z. Gao · B. Mao · C. Cao
State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University,
Beijing 100044, People's Republic of China
e-mail: rtkpli@188.com

specified origin station to its destination station. It has been shown that the optimal route can be found with essentially local strategies, i.e. with strategies that do not require precise global information of the network (Kleinberg 2000; Watts et al. 2002). Random walk is one of them.

The theory of random walk has a long history and has been applied to solve numerous theoretical and practical problems (Spitzer 1976; Jespersen et al. 2000). The dynamics of random walk on network represents a powerful tool, it can be used to explore the connection between the network topology and functional properties of the network (Jespersen et al. 2000). The random walk is also interesting since it could be a mechanism of transport and search on network (Guimerà et al. 2002; Adamic et al. 2001).

Based on random walk, we propose an algorithm for solving the problem of optimal route. We improve the random walk search strategy, and make it suitable for searching the optimal route on railway network. The paper is organized as follows. We introduce the background in Section 2. In Section 3, we outline the principle of the design of new transportation plan. The proposed algorithm and numerical computation results are respectively presented in Sections 5 and 6. Finally, conclusions of the proposed approach are presented.

2 Background

The optimization model for train routing problem is defined over a network, where the nodes represent stations and the edges represent existing and possible train connections among these stations. Assad proposed a multi-commodity network flow model for train routing problem, where some levels of interactions between routing and yard activities are incorporated (Assad 1980). Haghani proposed a formulation and a solution method for a combined train routing and makeup (Haghani 1989). Rodriguez presented a programming model for the route of trains travelling through a junction (Rodriguez 2007). Zwaneveld considered the problem of routing trains through railway stations (Zwaneveld et al. 1996). Carey et al. considered the train routing problem for large, busy, complex train stations, which includes heuristics analogous to those used in practice by train planners and managers (Carey and Carville 2003). In the engineering and design, the route of trains is under the constraints of network structure and infrastructure systems. In Zhang et al. (2005), Zhang et al. presented a preliminary network flow equilibrium model to identify and understand the interactions among infrastructure systems. O'Kelly outlined a simple notation and analytical framework for optimizing flows within a hub (O'Kelly 2009). Traffic incident also impacts on the train routing. Daamen et al. described a tool that identifies route conflicts and the train numbers involved (Daamen et al. 2009).

Many optimization models use iterative algorithms to improve the performance of train routing. The first model proposed by Szpigel (1972). Later, Carey and Lockwood developed a basic model to serve as the foundation for further research (Carey and Lockwood 1995). Other authors developed the computer-aided dispatch systems (Petersen et al. 1986; Jovanovic and Harker 1991). In order to solve some complex problems, researchers have devised various algorithms for optimizing the train routing, such as the heuristic method based on Lagrangian relaxation (Keaton 1989), the combination of genetic and tabu search algorithms (Gorman 1998a), expert system (Lozano et al. 2002) and neural network (Martinelli and Teng 1996). However, with the time constraints for real-time optimization and the enormous size of the most railway networks, none of the existing methods provide ideal results (Blum and Eskandarian 2002).

The method for train routing can be divided into two groups: computer-aided simulation and mathematical programming. In practice, mathematical programming methodologies were not widely employed. For example, the most basic formulation of train routing is the line-planning problem, but it is still done by hand (Kleinberg 2000). Instead, computer-aided simulation methodologies were mostly used (Ghoseiri et al. 2004). However, Most computer-aided dispatch systems either attempt little or no optimization (Blum and Eskandarian 2002). An evident instance is that the Santa Fe railway was unable to implement the iterative methods because they were too time-consuming. Instead, their train routing relies on a simple heuristic (Gorman 1998b).

In this paper, an improved walk search strategy is proposed to solve the train routing problem. The proposed strategy is a computer-aided simulation algorithm, where the optimization for train routing is considered. The proposed algorithm has the following advantages: (1) Since the time is discrete, numerical implementation of the proposed search algorithm is relatively simple. Using some updated rules, the proposed algorithm can be executed with shorter computation time. (2) The objective function can be written as a flexible formula, such as some rules and conditions. This feature means that the proposed algorithm can be used to solve many practical optimization problems. (3) To our knowledge, this work explicitly shows this effect for random walk for the first time.

3 The design of new transportation plan

In general, a railway network is divided into many small areas. From one area to another area, there is a passenger flux per day. The departure station is called the origin station, and the arrival station is called the destination station. The origin-destination (OD) flux of passengers represents the number of passengers who depart from the origin station and arrive the destination station per day. Demand for railway transportation is usually expressed in terms of the OD flux. Given these demands, the planner must establish a set of operations that will govern the routes of trains. For every OD pair of traffic demand, the train dispatcher assign each train to a route which includes distinct combinations of the track sections and routes. Usually, OD flow is not only under the travel time restriction, but also is under the constraints of population and employment. But, in railway transportation, when workers design the train schedule and plan, train routing is mainly assigned by the travel time. In the proposed algorithm, we regard the travel time as the optimization object.

In practical railway transportation system, the design of new passenger transportation plan is usually divided into several steps (Assad 1980; Bussieck et al. 1997; Ghoseiri et al. 2004). At first, the passenger demand has to be assessed and analyzed on a given railway network. According to the passenger demand, planners determine the departure and arrival stations of trains, the types of trains and the number of trains etc. Second, planners determine the routes of trains on such a given network. In this step, the main problem is how to choose the routes of trains from specified departure stations to specified arrival stations, which is called train routing problem. Then, according to the routes of trains, planners determine the plan of train travelling. In this step, the main problem is how to determine the departure and arrival time of trains at each station.

From a specified departure station to a specified arrival station, there are many possible routes. The chosen route of trains is the optimal route which has the smallest transportation cost among the possible routes. The physical layout of the track, such as control signaling and switch, imposes constraints on train travelling. These constraints mainly influence the departure and arrival time of trains at each station. In most case, these constrains are not considered in train routing problem.

4 The search algorithm based on deterministic walk

All railway networks can be represented in graph data structures. Let $G = (V, E)$ be an undirected graph, where V is the set of nodes and E is the set of edges. Each edge e_m has a weight c_m which is defined as the cost that unit train flow passes through the edge e_m. A route between two nodes is an alternating sequence of nodes and edges. The set of routes can be represented as

$$R_{ij} = \{\rho_{ij}^{l} : l = 1, 2, ..., \lambda(i, j)\}. \tag{1}$$

Where ρ_{ij}^{l} is the lth route between node i and node j, and $\lambda(i, j)$ is the number of the alternative routes between node i and node j. The edge correlation matrix can be defined as

$$a_{ij}^{ml} = \begin{cases} 1 & \text{if } e_m \in \rho_{ij}^{l} \\ 0 & \text{other}. \end{cases} \tag{2}$$

The cost carrying out the unit train flow from node i to node j is as follows.

$$c_{ij}^{l} = \sum_{m \in E} a_{ij}^{ml} c_m. \tag{3}$$

Let N_{ij} be the train flow from node i to node j. Since there can be more than one route between any two nodes, a binary decision variable is introduced for each allowed combination of a train and a route. The decision variable is defined as the follows.

$$x_{ij}^{l}=\begin{cases}1 & \text{if } N_{ij} \text{ selects the route } \rho_{ij}^{l}\\ 0 & \text{other.}\end{cases} \tag{4}$$

The problem of finding a route with the minimum cost between two specified nodes can be expressed as a mathematical formula, which has the following form

$$min \sum\nolimits_{i,j} N_{ij} \sum\nolimits_{l} c_{ij}^{l} x_{ij}^{l}, \tag{5}$$

$$s.t. \sum\nolimits_{l} x_{ij}^{l} = 1, \quad \forall i,\ j \tag{6}$$

$$x_{ij}^{l} \in \{0, 1\}, \quad \forall i,\ j, l. \tag{7}$$

When a train travels from an origin station to its destination station, only the default route is chosen. Conventionally, the train routing problem is solved by listing out all the default routes. However, since the number of the stations is large, this approach leads to that the amount of work and its need of memory increase. A common approach to solve this problem is to use the graph search algorithm. In this paper, we use the deterministic walk search strategy to solve the train routing problem.

In principle, the flow between an OD pair can be loaded to multiple available routes. When the travel time is related to the flow, the shortest route could vary with time. In this case, the flow could be assigned to different route. However, when the travel time is not related to the flow, the shortest route does not vary with time. In this case, the flow is assigned to the shortest route. This means that larger OD flow is assigned to the shortest route. In railway traffic, the travel time is not related to the flow. When passengers travel form one station to another station, they always choose a shortest route where they can spend smaller travelling time. This means that on the shortest route, the OD flux of passengers from specified origin station to specified destination station is larger than that on other possible routes. Based on above ideas, in our method, walker chooses its neighbor node which has a larger OD flux from the chosen nodes to specified destination node with a high probability. Here the neighborhood node is the node which has a direct connection to the current node.

Using purely local information, such as the number of nodes and edges, we implement the walk search strategy that monitors the neighbor nodes in search of its destination node. In general, from a given origin node to a given destination node, there are a number of possible routes. The search process for walker will be repeated many times. In the search process, the walker moves according to the following deterministic rules: (a) Starting from node r, the walker is only restricted to move to one of its neighbor nodes. Moreover, the edge connecting the node r and the chosen neighbor node should not be visited in preceding time steps. If all such edges have been visited in the preceding time steps, the walker randomly choose one of them; (b) The probability the walker moving from node r to its sth neighbor node is $P_s = f_{sj}/(\sum_k f_{kj})$, where f_{sj} represents the OD flux between the node s and the destination node j, and k represents the kth neighbor node of node r. The search cost for this step is calculated by $c_m = d_{rs}/v$. Here d_{rs} is the length of the edge connecting node r and node s, and v is the speed that walker moves. In fact, the search cost c_m represents the time that walker moves from node r to node s.

The probability P_s ensures that the proposed algorithm can find optimal route with shorter search time. As discussed above, OD flow is larger on optimal route. When OD flow is larger, this probability is high. If this probability is high, the proposed algorithm can find the optimal route with shorter search time. This means that the probability P_s reduces the search time. In addition, this probability mainly affects the search process. It ensures that the optimal solution can be found. The found optimal solution usually has smaller value of objective function. In other words, this probability ensures that the proposed algorithm can find the route which has smaller value of objective function.

The objective function is defined as that from a specified origin node to a specified destination node, walker has the smallest travelling time (or search cost). The executed process of the proposed algorithm can be described as follows.

Step 1: initialization parameters. Circle time k_c=0. k_{cmax} is the maximum circle time.
Step 2: displace one walker on the origin node i.
Step 3: circle time k_c=k_c+1.
Step 4: according to the deterministic walk rules described above, the walker moves from the origin node i to its destination node j. As the walker arrives its destination node j, a chosen route is found, or a solution is obtained.
Step 5: if $k_c < k_{cmax}$, go to Step 2.
Step 6: record the best solution which has the smallest search cost among the obtained solutions.

In step 4, only one route is obtained among all possible routes. This means that the search process is under the constraint $\sum_l x_{ij}^l$=1. In step 6, we can range all the obtained solutions. Using this step, one (or more) optimization solution(s) can be found from a specified origin node to a specified destination node.

When we use the proposed algorithm to solve the optimization problem of train routing, from a specified origin station i to a specified destination station j, we need find one (or two) optimal route(s). Here a walker represents the train flow N_{ij}. For the entire railway network, we usually need to determine many optimal routes among many specified stations. In this case, we can repeatedly execute the proposed algorithm many times to find all optimal routes for all specified OD pairs. The total search cost for the network is calculated by

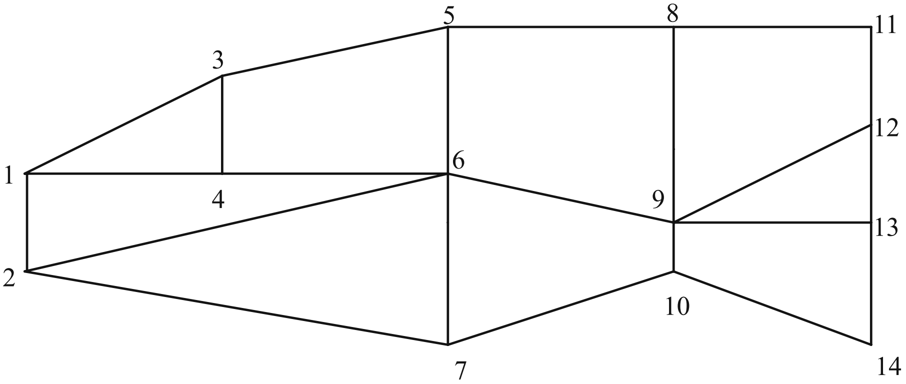


Fig. 1 An example of railway network

$\sum_{i,j} N_{ij} \sum_l c_{ij}^l = \sum_{i,j} N_{ij} \sum_l \sum_m a_{ij}^{ml} c_m$. When a special event occurs, such as the accident, confliction or the visitation of Providence, rerouting for trains will be considered. In this case, we use at least two optimal routes between two specified stations.

5 Computation results

In order to test the efficiency of the proposed search strategy, we apply the proposed algorithm to find optimal routes of trains on a part of railway network. Here the speed v for walkers is set to be $v = 200$ km/h. Figure 1 depicts the part of the railway network. This part of the railway network includes 22 track sections and 14 stations. For the sake of convenience, these stations are respectively labelled by the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 and 14. The lengths of the track sections are given in Table 1.

On the part of the railway network, the OD flux of passengers is given in Table 2. Here the number, such as 81 in column 8, means that there are 81 passengers travelling from station 1 to station 8 per day. On the basis of the data given in Table 2, we determine the departure and arrival stations of the routes of trains on the railway network. Between two stations, if the number of passengers who directly depart from one station to other station is larger than

Table 1 Lengths of track sections

Track	Length (km)	Track	Length (km)	Track	Length (km)	Track	Length (km)
d_{12}	110	d_{35}	260	d_{710}	260	d_{913}	220
d_{13}	240	d_{46}	250	d_{89}	210	d_{1014}	240
d_{14}	220	d_{56}	160	d_{811}	220	d_{1112}	110
d_{26}	490	d_{58}	250	d_{910}	50	d_{1213}	100
d_{27}	480	d_{67}	190	d_{912}	250	d_{1314}	130
d_{34}	100	d_{69}	260				

Table 2 OD flux of passengers

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	194	133	96	322	275	68	81	223	343	69	31	274	47
2	88	0	59	19	273	64	191	136	22	301	57	23	240	104
3	123	37	0	42	57	107	290	104	8	26	243	310	19	81
4	111	93	57	0	61	144	370	23	147	32	51	25	293	273
5	297	223	36	9	0	38	173	32	63	190	287	27	23	49
6	215	51	97	157	24	0	62	216	3	24	19	270	25	52
7	106	242	274	290	123	51	0	39	52	192	26	47	356	103
8	273	57	256	47	83	120	48	0	27	115	32	55	371	71
9	197	36	18	133	66	39	22	54	0	267	48	14	126	106
10	391	226	63	39	148	47	105	400	180	0	247	218	103	201
11	22	59	246	55	290	17	43	82	49	107	0	118	203	171
12	36	31	230	52	86	263	15	70	51	148	68	0	160	151
13	322	117	49	107	18	31	300	67	143	237	88	140	0	121
14	91	126	36	293	48	74	150	50	181	173	148	130	91	0

300 per day, we will design a route of trains between such two stations. This means that trains will directly travel between the two stations. This criterion usually is promulgated by government according to practical demand. For example, the OD flux of passengers between station 1 and station 5 is larger than 300, we will assign a route of trains between station 1 and station 5. The assigned route between station 1 and station 5 is expressed by $1 \leftrightarrow 5$. The final determined results are as follows. $1 \leftrightarrow 5$, $1 \leftrightarrow 10$, $2 \leftrightarrow 10$, $3 \leftrightarrow 12$, $4 \leftrightarrow 7$, $7 \leftrightarrow 13$, $8 \leftrightarrow 13$, $10 \leftrightarrow 1$, $10 \leftrightarrow 8$, $13 \leftrightarrow 1$.

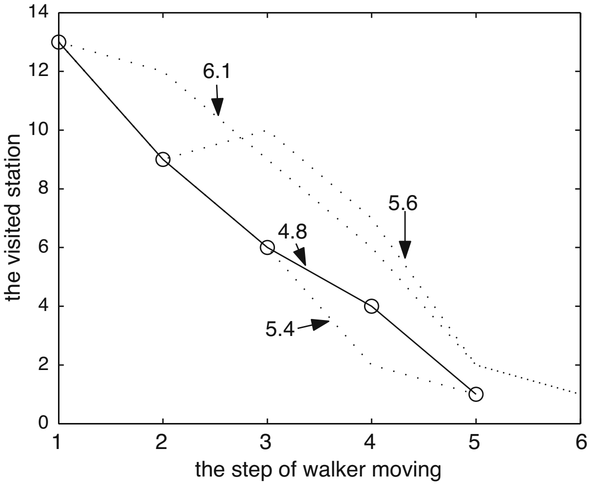
The proposed algorithm is suitable for computer programming. Using such an algorithm, we obtain the optimal routes of trains on the railway network. Table 3 gives the obtained results. The routes listed in column 3 represent the optimal routes of trains, and the costs listed in column 4 represent corresponding minimum search costs. For example, the route $1 \rightarrow 3 \rightarrow 5$ listed in column 3 is the optimal route. Such a route has the smallest search cost among all possible routes from the origin station 1 to the destination station 5. In Table 3, most of routes of trains have the smallest search costs, but the route, i.e. $1 \rightarrow 2 \rightarrow 7 \rightarrow 10$, only has a smaller search cost. In the case study, from the node 1 to its neighbor nodes, the OD flow between the nodes 1 and 2 is much larger than that between the node 1 and other neighbor nodes. Using the proposed search algorithm, walker starting from the node 1 selects its neighbor node 2 with a higher probability. In this case, the obtained optimal solution is $1 \rightarrow 2 \rightarrow 7 \rightarrow 10$, which is a local optimal solution among all possible solutions. Moreover, it is a second optimal solution, which is only smaller than the global optimal solution $1 \rightarrow 4 \rightarrow 6 \rightarrow 9 \rightarrow 10$.

Since the probability for the walker to step in one direction is greater than that in other directions, the walker propagates in such a direction with higher probability and eventually visit the nodes in this direction. Figure 2 presents the search process for walker. Here, the horizontal axis indicates the step that walker moves, and the vertical axis indicates the visited station. The chosen route for the walker is between the stations 13 and 1. In Fig. 2, the optimal route between stations 13 and 1 is a solid line with symbol "o", and other dash lines represent different chosen routes. Here the numbers represent the search

Table 3 Routes of trains and search costs

Origin	Destination	Route of train	Search cost (h)
1	5	$1 \rightarrow 3 \rightarrow 5$	2.5
1	10	$1 \rightarrow 2 \rightarrow 7 \rightarrow 10$	4.3
2	10	$2 \rightarrow 7 \rightarrow 10$	3.7
3	12	$3 \rightarrow 5 \rightarrow 8 \rightarrow 11 \rightarrow 12$	4.2
4	7	$4 \rightarrow 6 \rightarrow 7$	2.2
7	13	$7 \rightarrow 10 \rightarrow 9 \rightarrow 13$	2.7
8	13	$8 \rightarrow 9 \rightarrow 13$	2.2
10	1	$10 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 1$	3.9
10	8	$10 \rightarrow 9 \rightarrow 8$	1.3
13	1	$13 \rightarrow 9 \rightarrow 6 \rightarrow 4 \rightarrow 1$	4.8

Fig. 2 The search processes for walkers



costs for each lines. From Fig. 2, it is obvious that the optimal route has the smallest cost among the chosen routes.

We programme the proposed algorithm on a personal computer which has one processor. When the algorithm is executed, we record the CPU time (finding the optimal solution). Table 4 gives the average CPU time for finding optimal routes of trains. The computation experiment is made 10 times, the final result is obtained by averaging them. From Table 4, it is obvious that the optimal routes of trains are obtained with shorter computation time. This means that the proposed search algorithm is an effective tool for optimizing train routing.

In practical applications, k shortest paths algorithm is a common used algorithm (Eppstein 1998). Using the k shortest paths algorithm, the Dijkstra's algorithm which is the best known algorithm up to now (Dijkstra 1959), is used to find the shortest route between any two specified nodes. For the purpose of investigating the performance of the proposed algorithm, we compare the proposed algorithm with the k shortest paths algorithm. In Fig. 1, there are totally 182 possible OD pairs. For each OD pair, we made one test respectively using the k shortest paths algorithm and the proposed algorithm. Among 182

Table 4 CPU time for optimizing train routing

Route of train	CPU time (s)
1 → 3 → 5	0.094
1 → 2 → 7 → 10	0.018
2 → 7 → 10	0.032
4 → 6 → 7	0.104
7 → 10 → 9 → 13	0.094
8 → 9 → 13	0.036
10 → 9 → 6 → 4 → 1	0.054
10 → 9 → 8	0.032
13 → 9 → 6 → 4 → 1	0.112

Table 5 Different optimal routes

Origin	Destination	The *k* shortest paths algorithm (h)	The proposed algorithm (h)
1	10	1 → 4 → 6 → 9 → 10 (3.9)	1 → 2 → 7 → 10 (4.25)
2	9	2 → 6 → 9 (3.75)	2 → 7 → 10 → 9 (3.95)
3	10	3 → 4 → 6 → 9 → 10 (3.3)	3 → 5 → 6 → 9 → 10 (3.65)
5	14	5 → 6 → 9 → 10 → 14 (3.55)	5 → 8 → 9 → 10 → 14 (3.75)
6	11	6 → 9 → 12 → 11 (3.1)	6 → 5 → 8 → 11 (3.15)
10	3	10 → 9 → 6 → 4 → 3 (3.3)	10 → 7 → 6 → 4 → 3 (4)
11	4	11 → 8 → 5 → 3 → 4 (4.15)	11 → 12 → 9 → 6 → 4 (4.35)
13	3	13 → 9 → 6 → 4 → 3 (4.15)	13 → 12 → 11 → 8 → 5 → 3 (4.7)

measurements, we finally find 8 different optimal routes. The comparison results are listed in Table 5. Here we only list the different routes. From Table 5, we can see that the routes listed in column 3 are all the global optimal solutions, but the routes listed in column 4 are all the local optimal solutions. The costs of these local optimal solutions are only smaller than that of the global optimal solutions, i.e. these local optimal solutions are the second optimal solutions. The comparisons indicate that the performance of the proposed algorithm is very close to the classical k shortest paths algorithm. However, unlike the k shortest paths algorithm, the proposed algorithm can be used when weights have negative values.

In order to further test the proposed algorithm, we compare the optimal routes obtained by two different search algorithms on a larger railway network which has 47 stations. The numerical computation results indicate that: (1) when we only need to find one optimal route between two specified stations, the CPU time using the proposed algorithm is close to that using the k shortest paths algorithm; (2) when we need to find two optimal routes between two specified stations, the CPU time using the proposed algorithm is obviously smaller than that using the k shortest paths algorithm. Table 6 gives part of the comparison results. Here each optimal route is called the second shortest route, which is only smaller than the shortest route. From Table 6, we can see that the CPU time listed in column 4 is obviously smaller than that listed

Table 6 CPU time for optimizing train routing

Origin	Destination	The *k* shortest paths algorithm (s)	The proposed algorithm (s)
18	24	0.071	0.027
27	41	0.044	0.026
22	40	0.052	0.040
11	23	0.053	0.031
20	30	0.043	0.030
12	37	0.063	0.050
18	31	0.060	0.030
7	26	0.033	0.020
18	8	0.043	0.020
27	31	0.045	0.040

in column 3. Moreover, using the proposed algorithm, the objective function can be written as a flexible formula. This feature ensures that the proposed algorithm can be applied extensively.

6 Conclusions

In conclusions, we propose a new search strategy to solve the problem of train routing on railway network. The proposed algorithm is based on deterministic walk. Since the proposed algorithm consists of some simple rules, it can be executed with shorter computation time. Remarkable, it can be integrated into a decision support system used by operators who make decisions to determine the optimal routes of trains.

It should be pointed out that the proposed algorithm depends on the amount and quality of OD flow data. On a large network, since the stations and routes are larger, a great amount of OD flow data is needed. In addition, the good performance of the proposed algorithm needs the OD flow data with high quality. In the proposed algorithm, the capacity constraint of each link (or route) is not considered. This limits the practical applications of the proposed algorithm. In our future work, we will extend the proposed algorithm where the capacity constraint is considered.

The proposed search algorithm is a heuristic algorithm. As a heuristic algorithm, its search process is to select a solution as optimal solution from possible solutions. Such an optimal solution possibly is not a globally optimal solution, but it surely is a locally optimal solution. In other words, there is no guarantee that such an algorithm will provide a global optimal solution over all time. In fact, it is an improved heuristic search strategy, and we believe that the proposed search strategy will gain new perspectives for route choice modelling. The proposed algorithm is also used to solve the optimal route problem in other fields, such as the city traffic and internet.

References

Adamic LA, Lukose RM, Puniyani AR, Huberman BA (2001) Search in power-law networks. Phys Rev E 64(4):046135, Sept

Assad AA (1980) Modelling of rail networks: toward a routing/makeup model. Transp Res Part B 14(1–2):101–114, March–June

Blum J, Eskandarian A (2002) Enhancing intelligent agent collaboration for flow optimization of railroad traffic. Transp Res Part A 36(10):919–930, Dec

Bussieck MR, Winter T, Zimmermann UT (1997) Discrete optimization in public rail transport. Math Program 79(1–3):415C444, Oct

Carey M, Carville S (2003) Scheduling and plat forming trains at busy complex stations. Transp Res Part A 37(3):195–224, March

Carey M, Lockwood D (1995) A model, algorithms and strategy for train pathing. J Oper Res Soc 46(8):988–1005, Aug

Daamen W, Goverde RMP, Hansen IA (2009) Non-discriminatory automatic registration of knock-on train delays. Netw Spatial Econ. doi:10.1007/s11067-008-9087-2

Dijkstra EW (1959) A note on two problems in connexion with graphs. Numerische Matematik 1(1):269–271, Dec

Eppstein D (1998) Finding the k shortest paths. SIAM J Comput 28(2):652–673

Ghoseiri K, Szidarovszky F, Asgharpour MJ (2004) A multi-objective train scheduling model and solution. Transp Res Part B 38(10):927–952, Dec

Gorman MF (1998) An application of genetic and tabu searches to the freight railroad operating plan problem. Ann Oper Res 78(1):51–69, Jan

Gorman MF (1998) Santa Fe Railway uses an operating-plan model to improve its service design. Interfaces 28(4):1–12, Apr

Guimerà R, Díaz Guilera A, Vega-Redondo F, Cabrales A, Arenas A (2002) Optimal network topologies for local search with congestion. Phys Rev Lett 89(24), 248701, Dec

Haghani AE (1989) Formulation and solution of a combined train routing and makeup, and empty car distribution model. Transp Res B 23B(6):433–452, Dec

Jespersen S, Sokolov IM, Blumen A (2000) Relaxation properties of small-world networks. Phys Rev E 62(3):4405–4408, Sept

Jovanovic D, Harker PT (1991) Tactical scheduling of rail operations: the SCAN I system. Trans Sci 25(1):46–64, Sept

Keaton MH (1989) Designing optimal railroad operating plans: Lagrangian relaxation and heuristic approaches. Transp Res B 23(6):415–431, Dec

Kleinberg JM (2000) Navigation in a small world. Nature 406(6798):845–847, Aug

Lozano ER, Macias ER, Laita LM (2002) A computer algebra approach to the design of routes and the study of their compatibility in a railway interlocking. Math Comput Simul 58(3):203–214, Feb

Martinelli DR, Teng H (1996) Optimization of railway operations using neural net-works. Transp Res Part C 4(1):33–49, Feb

O'Kelly ME (2009) Routing traffic at hub facilities. Netw Spatial Econ. doi:10.1007/s11067-008-9061-z

Petersen ER, Taylor AJ, Martland CD (1986) An introduction to computer assisted train dispatch. J Adv Trans 20(1):63–72

Rodriguez J (2007) A constraint programming model for real-time train scheduling at junctions. Transp Res Part B 41(2):231–245, Feb

Spitzer F (1976) Principles of random walk. Springer, New York

Szpigel B (1972) Optimal train scheduling on a single-track railway. In: Ross M (ed) Operational Research '72, OR'72. North-Holland, Amsterdam, pp 343–351

Watts DJ, Dodds PS, Newman ME (2002) Identity and search in social networks. Science 296(5571):1302–1305, May

Zhang PC, Peeta S, Friesz T (2005) Dynamic game theoretic model of multi-Layer infrastructure networks. Netw Spatial Econ 5(2):147–178

Zwaneveld P, Kroon JL, Romeijn HE, Salomon M, et al (1996) Routing trains through railway stations: model formulation and algorithms. Trans Science 30(3):181–194