Research on the Connecting Path Search Algorithm for Air-Rail Integration

Tao Xu
College of Computer Science and Technology, Civil Aviation University of China, Tianjin, China
Information Technology Research Base, Civil Aviation Administration of China, Tianjin, China
Email: txu@cauc.edu.cn

Xiaolu Ding and Jianfu Li
College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China
Email: dingxiaolu88, jianfu_lili@163.com

***Abstract*—With the rapid development of high-speed railway and the changes of the tourists' requirement, it is imperative to the integrate advantages of aviation and rail. This paper constructs the air-rail integration model based on the current development situation and characteristics of Chinese railways. In allusion to the search problem of connecting path in air-rail integration network, a constrained Yen* algorithm is proposed to solve the problem in this paper. The constrained Yen* algorithm is set up by using the heuristic strategy of A* algorithm and two certain constraints by reducing running time to generate candidate paths. The experimental results show that the search problem of connecting path in air-rail integration network can be obtained fast by the constrained Yen* algorithm. Therefore, constrained Yen* algorithm is more efficient than constrained Yen algorithm in application.**

***Index Terms*—air-rail integration model; connecting path; constrained Yen* algorithm; A* algorithm**

I. INTRODUCTION

In recent years, the rapid development of high-speed railway causes a considerable impact on the air transport industry. The requirements of tourists have also changed with the continuous developments of the global route network, and they are more concerned about the travel time, cost and routes selection. In order to integrate the advantages of aviation and rail, air-rail integration network can be constructed to fully utilize their advantages, and then a number of connecting paths can be provided to meet needs of tourists. The result of this research does not only improve the traveling service quality, but also enhances the service satisfaction and the competitiveness of civil aviation of China.

At present, the official websites of some major airline companies, tourism companies in the world and domestic airlines can offer route-searching service to tourists expediently. But there is a definite gap in the usability, limitation and form of result of route-searching service [1, 2]. Besides, there is no available air-rail integration model in China, just only several airline companies can provide the special connecting path.

Due to the complexity of the tourists' mental selection, the single path cannot meet needs of tourists, the airlines, therefore, should provide a number of connecting paths. In other words, the connecting paths search algorithm for air-rail integration needs to find K connecting paths to meet the demands of tourists. Essentially, the principal problem of the connecting paths search algorithm is to seek the K multiple constrained shortest path, namely the KMCSP problem [3], in a given network topology.

KMCSP problem has attracted many attentions from researchers and become a hot topic. At present, many classical shortest algorithms [4, 5, 6] have been proposed, but only a few new algorithms based KMCSP are presented. It is well known that the KMCSP problem is NP-Complete [7, 8, 9]. A new scheme, Multi-Constraint-Pruning (MCP) algorithm to calculate the KMCSP in a given constraints network, is proposed by Zhang and Zhao [10]. In [11], a pseudo-polynomial time algorithm is presented to solve the routing problem exactly. Based on the self-adaptive multiple constraints routing algorithm (SAMCRA), Mieghem [12] proposes an improved and exact version of QoS routing algorithm, the tunable accuracy multiple constrains routing algorithm (TAMCRA). Similarly, a heuristic algorithm is proposed by Korkmaz and Krunz, namely H_MCOP algorithm [13], which attempts to minimize both the nonlinear cost function and the primary cost function. In [14], Liu and Ramakrishnan present a new algorithm, A* Prune, to list the first K multiple constrained shortest path between a given pair of nodes in a digraph. She [15] puts forward a shortest path algorithm with constraints, which is based on Dijkstra algorithm and uses adjacency matrix as data storage structure. Based on the branching procedure, Ning [16] proposes a specific algorithm for an application where resource and loopless constraints have to be respected.

K multiple constrained shortest path problem can be solved with the above typical algorithms, but those algorithms cannot be directly applied to connecting paths

Manuscript received March, 2013; revised March, 2013; accepted March, 2013.
Copyright credit, project number, Tao Xu, Xiaolu Ding etc.

doi:10.4304/jsw.8.8.1889-1896

search for air-rail integration. The reason includes three restricting factors [17]: the potential factors influencing the travel choice behavior of passengers, huge and complex searching space, and the necessary condition to the link between aviation and rail transportation. In order to solve the problem proposed above, we need to reference the design ideas of some algorithm. In this paper, we improve Yen algorithm, motivated by A*, an outstanding heuristic search algorithm, referred to as constrained Yen* algorithm.

The rest of the paper is organized as follows. In Section 2, we describe the air-rail integration pattern. Connecting path search algorithm, the constrained Yen* algorithm, is introduced in Section 3. The experimental results and evaluation are presented in Section 4, and Section 5 is about conclusions we get from result analysis.

II. CONSTRUCT THE AIR-RAIL INTEGRATION MODEL

A. Air-Rail Integration Model

Air-rail integration model refers to send visitors to reach the destination safely, fast and comfortably through air and rail transport mode. The passengers sign the contract of transportation with the unified management of integration transport enterprises or institutions, which provides high-quality, convenient and efficient transportation services throughout. In narrow sense, railway system as the regional transport can be considered as an extending service of civil aviation transport to construct air-rail integration model. The transport structure of air-rail integration model is described in Figure. 1.

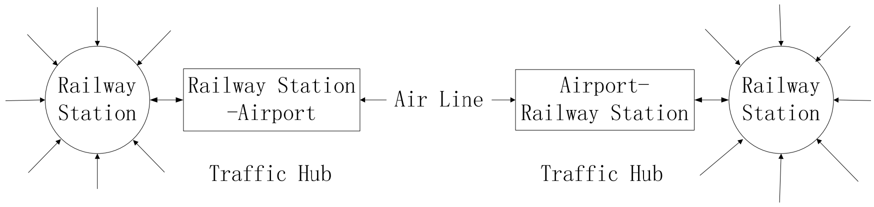

Figure.1 Transport structure of air-rail integration model

Air-rail integration model is a new traffic model combining the air transport and rail transport, therefore, how to determine the model of air-rail integration should be given primary consideration. According to current development situation and characteristics of Chinese railways, we establish three basic multimodal transport forms which are between the aviation and general rail, intercity rail and high-speed railway transport. Since the three air-rail integration forms have certain limits, we analysis them in details to maximize the transport efficiency.

The transportation organization of general railway is complex in China and the service quality between general railway and aviation still has great difference. High-speed railway and intercity railway not only have advantages of large scale and densely-distributed stations, but also fast speed, convenience and comfort. Therefore, high-speed railway and intercity railway have basic conditions on multimodal transport. As the following table shows, we compare the three transport forms from the four aspects: the technical speed, the service level, and network coverage and management system.

Through the above-mentioned analysis, such a conclusion can be drawn: implementation of the air-rail integration forms should concentrate on aviation and high-speed railway, aviation and intercity railway as its auxiliary form. Therefore, in this paper, the air-rail integration models include aviation and high-speed railway, aviation and intercity railway.

TABLE.1

COMPARISON WITH AIR-RAIL INTEGRATION FORMS

Aspect	General railway	High-speed railway	Intercity railway
Technical speed	general	high	rapid
Service level	general	good	good
Network coverage	good	general	good
Management system	bigger gap	similar	smaller gap

B. Air-Rail Integration Region

In order to fully utilize the characteristic of aviation and railway, the air-rail integration area based on the traditional economic circle becomes a new traffic way from the center to the border while taking hub cities as core. Now there are five economic circles in China, Jingjinji region, Yangtze River delta region, Pearl River delta region, Cheng-yu region and Northeast region. The distance factor should be taken into account when aviation and rail implement action of air-rail integration model. Besides, we have also considered the punctuality rate of train and plane to avoid the transfer delay.

According to the operation of the Chinese railway, the coverage of hub airports should be limited in three hundred kilometers. The air-rail integration area by the hub airports of each economic circle are described in Table 2.

TABLE.2

AIR-RAIL INTEGRATION AREA

Economic circle	Hub airport	Major city
Jingjinji region	Beijing Capital International Airport Tianjin Binhai International Airport	Baoding、Beidaihe、Cangzhou、Langfang、Shijiazhuang、Tanggu、Tangshan、Wuqing
Yangtze river delta region	Shanghai Pudong International Airport Shanghai Hongqiao International Airport Nanjing Lukou International Airport Hangzhou Xiaoshan International Airport Wuxi Shuofang International Airport	Benbu、Changzhou、Chuzhou、Jinhua、Jiaxing、Kunshan、Ningbo、Quzhou、Suzhou、Shaoxing
Pearl river delta region	Guangzhou Baiyun International Airport Shenzhen Baoan International Airport Zhuhai Sanzao Airport	Dongguan、Jiangmen、Shunde、Shaoguan、Xinhui、Xiaolan、Zhongshan
Cheng-yu region	Chengdu Shuangliu International Airport Chongqing Jiangbei International Airport	Dujiangyan、Suining、Qingchengshan、Nanchong、Dazhou、Hechuan
Northeast region	Harbin Taiping International Airport Changchun Longjia International Airport Shenyang Xiantao International Airport	Siping、Huludao、Jinzhou、Tieling、Panjin、Longjia

III. CONNECTING PATH SEARCH ALGORITHM

A. Yen Algorithm

Yen algorithm is the first one proposed and the most classical algorithm for the constrained loopless K shortest paths. It attempts to build a tree formed by K shortest paths from initial node s to terminal node t. In the tree, the root is the initial node s and the leaves are K copies of the terminal node t, and each (unique) path from s to a leaf corresponds to one of K shortest paths. Yen algorithm starts with computing the shortest path p_1 between s and t in network by Dijsktra or any other shortest path algorithm. Then, Yen algorithm computes continuously the next shortest path p_i according to p_1, p_2, ···, $p_{i\text{-}1}$ until the p_K is found.

When computing p_i, Yen algorithm considers every node v except for the terminal node in the first i-1 shortest paths as the deviation node, then computes a candidate shortest path for every deviation node. Finally, the shortest one from all the candidates is selected as p_i. As the nodes in the first i-2 shortest paths have been computed in the process of finding $p_{i\text{-}1}$, it only needs to compute the nodes along $p_{i\text{-}1}$when computing p_i. However, in order to find the next shortest path p_i, Yen algorithm needs to compute a candidate shortest path for every node in $p_{i\text{-}1}$ except for terminal node, which will cost a lot of time. Yen algorithm, therefore, is time-consuming. The pseudo-code of constrained Yen* algorithm is described in Figure 2.

```
Function: Yen algorithm
Inputs:
G= (V, E), a graph with node set V and edge set E;
(s, t): a node pair with source s and target t;
K: number of paths to be found;
Output:
K shortest paths set P_K between s and t in G
1.      k=0, max_len=999999, p ← Dijkstra(s, t)
2.      candidate_list ← { p }// set of candidate paths
3.      P_K ← {}// set of K paths
4.      while(k≤K&& candidate_list ≠Φ)
4.1         get the shortest path p from candidate_list
4.2         while( deviation node v of p)
4.2.1           get p_i begin from the node v;
4.2.2           candidate_list ←p_i ;
4.3         p as the shortest path P_K;
4.4         k++;
5.      return P_K
```

Figure.2 Yen algorithm

B. Constrained Yen* Algorithm

A* algorithm is the most prominent heuristic search algorithm [18]. It intends to firstly extend the most possible node to find target nodes faster. Therefore, it uses a heuristic function f to sort the search queue. An admissible heuristic guarantees the solution optimality of A* algorithm.

Based on analyzing the search problem of connecting path and the complicated constraints of air-rail integration network, the improvement of Yen algorithm, constrained Yen* algorithm, motivated by A*, is proposed in this paper. The air-rail integration network is defined as a weighted directed graph G. Like the basic process of Yen algorithm, constrained Yen* algorithm also starts with computing the shortest path p_1 between s and t in G by any shortest path algorithm. Then, constrained Yen* algorithm computes continuously the next shortest path p_i according to p_1, p_2, ···, $p_{i\text{-}1}$until the p_K is found. The main difference between Yen and constrained Yen* algorithm is the process to compute the next shortest path p_i and the certain constraints to choose connecting path.

Constrained Yen* algorithm first computes the evaluation function values of every node in $p_{i\text{-}1}$, and then computes continuously and preferentially the shortest path p_{vt} for better node v until p_{vt} exits. Finally, by using the certain constraints the shortest path p_i is judged. If the p_i does not meet with certain constraints, constrained

Yen* algorithm will be reapplied to other path to compute the next shortest path until the K shortest path is found or the path is empty. The pseudo-code of constrained Yen* algorithm is described in Figure 3.

```
Function: Constrained Yen* algorithm
Inputs:
G= (V, E), a graph with node set V and edge set E;
(s, t): a node pair with source s and target t;
K: number of paths to be found;
Output:
K shortest paths set P_K between s and t in G
1.      k=0, max_len=999999, p ← Dijkstra(s, t)
2.      open ← {}// set of nodes
3.      P_K ← {}// set of paths
4.      while(k≤K&& p≠Φ)
4.1        v ←successor of v in p;
4.2        open ←cost(v);
4.3        get v_min from open
4.4        get p_new begin from the node v_min
4.5        if(p_new≥max_len)
                delete node v;
                if (open≠Φ)
                     goto 4.3;
                p_new=Φ;
4.6        if ( p meets constraint conditions)
                p as the shortest path P_K;
                k++;
                p=p_new;
5       return P_K
```

Figure.3 Constrained Yen* algorithm

Although, in theory, in the worst case, constrained Yen* algorithm is in the same order of time complexity as Yen algorithm. The worst case occurs when only the short path p_{vt} of the worst node v exists, thus Yen algorithm needs to try to compute the short path for every node in open array. However, in practical applications, the possibility of the existence of this worst case is minimal. Therefore, constrained Yen* algorithm is quicker than Yen algorithm.

IV. Experiments and Results

A. Experimental Data

Aiming at the complexity in air-rail integration network, the experimental data was processed by Oracle data base. The experiments are done in the global airlines graph with 3728 nodes and 65903 edges and the Chinese railway network with 41 nodes and 254 edges in five economic circles, where each node is an airport or station, each edge is a regular flight or route between two airports or between airport and station, and the length of every edge is defined as the spherical or run distance between the two nodes. The fields in data base defined in the experiment data are shown as follow in Table 3, and the data structure in Oracle data base is described in Table 4. All algorithms are implemented using the C++ programming language and tested on randomly generated nodes.

TABLE.3

Fields Defined in the Experiment Data

File Name	File Type	Null Value	Note
DEP_LOC	VARCHAR	N	Origin airport or station
ARR_LOC	VARCHAR	N	Destination airport or station
ARR_LONGI	DOUBLE	N	Longitude of destination airport or station
ARR_LATI	DOUBLE	N	Latitude of destination airport or station
AIR_DIS	INTEGER	N	Distance of airports or stations
FIT_NUM	INTEGER	Y	Flight or train number of airport or station
CORE_NUM	INTEGER	Y	Importance degree of airport or station

TABLE.4

Data structure in Oracle data base

DEP_ LOC	ARR_ LOC	ARR_ LONGI	ARR_ LATI	AIR_ DIS	FIT_ NUM	CORE_ NUM
PEK	CAN	113.3	23.4	1883	62	294
PEK	LAX	-118.4	33.9	11122	11	582
PEK	SHA	121.4	31.1	1105	117	109
PEK	SFO	-122.4	37.8	10467	11	536
PEK	BDH	119.4	39.8	277	16	12

B. Constraint Condition and Evaluation Function

In allusion to the feature of the connecting path search in air-rail integration network, the constraints include the following two aspects. First, the number of nodes in the shortest path cannot exceed an invariable parameter. Second, the shortest path probably includes a certain transferring airport or station. Through the analysis of records about the travel choice behavior of passengers, the number of transfer is defined less than or equal to twice in experiments.

The evaluation function of constrained Yen* algorithm can be defined by Equation 1.

$$cost(v)=a_1*f(v)+a_2*f(v)/(FIT\_MUN+1)+b*CORE\_NUM \quad (1)$$

$$f(v)=g(v)+h(v)+q(v) \quad (2)$$

The *cost* (v) in Equation 1 represents the final evaluation value of node v, and the variable a_1, a_2, b are 0.7, 0.3, -1 respectively. In Equation 2, the $f(v)$ denotes the evaluation value of node v in the shortest path, where $g(v)$ means the real distance from source node s to current node v, $h(v)$ indicates the estimated distance from current node v to destination node t, q (v) represents the estimated distance from father node of current node v to destination node t.

Given the geographic coordinate values of two nodes v_1 and v_2, respectively were (x_1, y_1) and (x_2, y_2). The spherical distance function of constrained Yen* algorithm can be defined by Equation 3.

$$dis = r * a\cos(\cos(y_1 * PI/180) * \cos(y_2 * PI/180) * \cos(x_1 * PI/180 - x_2 * PI/180) + \sin(y_1 * PI/180) * \sin(y_2 * PI/180)) \quad (3)$$

The *dis* in Equation 3 represents the spherical distance value between node v_1 and node v_2. The variable r means the earth radius, and *PI* is the value of pi.

C. Experimental Results and Analysis

In order to evaluate the efficiency of constrained Yen* algorithm, we conduct experiments to compare constrained Yen* algorithm with constrained Yen algorithm. Here, the constrained Yen algorithm is the Yen algorithm with some certain constraints. In order to make it more comparable, constrained Yen* algorithm and constrained Yen algorithm are coded in C and experiments are carried out on a DELL PC with 2.9-GHZ CPU and 2-GB RAM, running over Windows XP.

1) Running time

In the first experiment, 20 Origin-Destination(O/D) pairs of node are randomly selected, and only one constraint condition is that the number of transfer is less than or equal to twice. The results of constrained Yen* algorithm and constrained Yen algorithm with different setting of K on the 20 pairs of node are shown in Figure 3. Figure 4 shows the time spent by constrained Yen* algorithm and constrained Yen algorithm in calculating the K shortest paths between every pair of node in air-rail integration network with K = 5, 10, 15, 20 and 25.

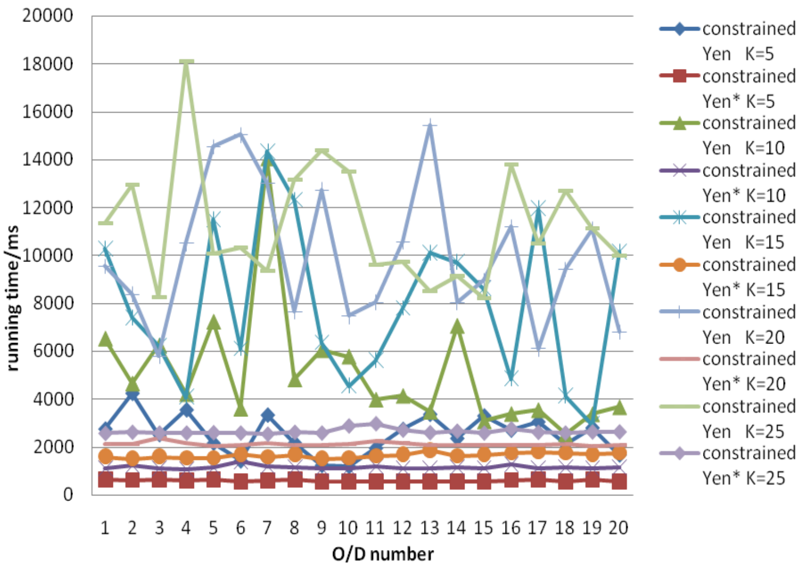

Figure.4 Running times with different K value

From Figure.4, we can get the following two points:

a) With different setting of K, for every randomly selected pairs of node, the running time of constrained Yen* algorithm is much shorter than those of constrained Yen algorithm. In terms of quantity, the total time spent by constrained Yen algorithm in computing K shortest paths of 20 pairs of node with different settings of K is 735988 milliseconds, and the time spent by constrained Yen* algorithm is 162730 milliseconds. Therefore, in terms of running time, constrained Yen* algorithm increases constrained Yen algorithm by (735988 - 162730) / 735988/100% = 77.89%.

b) For different O/D pairs, the running time of constrained Yen algorithm varies widely, however, the running time of constrained Yen* algorithm is almost the same. That is mainly due to the difference of the relative location between the source node and the destination node of different pairs in graph. That is, if the path connecting the source node and the destination node in the graph contains many nodes, when computing the next shortest path, constrained Yen algorithm needs to compute these nodes one by one, which makes constrained Yen* algorithm spend more time. On the other hand, when computing the next shortest path p_i, constrained Yen* algorithm does not need to compute all the nodes on p_{i-1}. Therefore, the relative position of different pairs of node cannot bring great influence on the running time of constrained Yen* algorithm.

2) Search scale

Like constraint condition of the first experiment, the test results of constrained Yen* algorithm and constrained Yen algorithm with different settings of K on the 8 pairs of node in second experiment are shown in Figure 5. Figure 5 shows the number of candidate path by constrained Yen* algorithm and constrained Yen algorithm in calculating the K shortest paths between every pair of node in air-rail integration network with K = 5, 10 and 15.

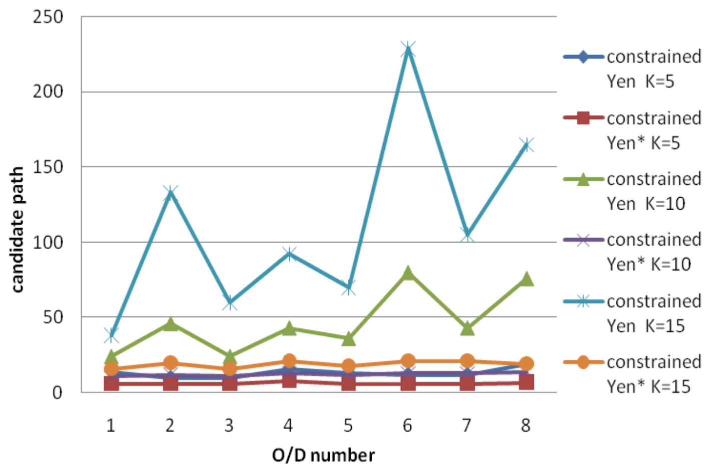

Figure.5 Number of candidate path with different K value

The Figure 5 shows that the number of candidate path of constrained Yen* algorithm is much less than that of constrained Yen algorithm, when the number of candidate path increases with the increase of K value. In other word, the searching scale of constrained Yen algorithm varies widely, however that of constrained Yen* algorithm is almost the same. The constrained Yen* algorithm, therefore, has more advantages in large-scale network.

3) Length of connecting path

In order to analyze the connecting paths of the algorithms with respect to different O/D pairs, we randomly select 5 O/D pairs of node to compare constrained Yen* algorithm and constrained Yen algorithm, and only one constraint condition is that the number of transfer is less than or equal to twice. The results of constrained Yen* algorithm and constrained Yen algorithm with same setting of K on the 5 pairs of node are shown in Table 5. Table 5 shows the different length by constrained Yen* algorithm and constrained Yen algorithm in calculating the K shortest paths between every pair of node in air-rail integration network with K = 5.

The experiment result shows that, for every O/D pair, the first shortest connecting path has the same length. For other shortest path, the length of connecting path resulted by constrained Yen algorithm is always less than the length of connecting path resulted by constrained Yen* algorithm. But the length of connecting path has little difference between constrained Yen* algorithm and constrained Yen algorithm. As indicated above, compared with constrained Yen algorithm, constrained Yen* algorithm still has high accuracy and validity.

4) Search of connecting path

There are three experiments in the following, 3 O/D pairs of node are randomly selected in experiments. The third group experiment only has one constraint condition that the number of transfer is less than or equal to twice, the others must include a specific transferring airport or station.

TABLE.5

THE SHORTEST PATH FROM AIRPORT TO STATION

O/D	constrained Yen algorithm	constrained Yen* algorithm	difference length
1	11672	11672	0
	11672	11672	0
	11676	11676	0
	11676	11693	17
	11693	11696	3
2	5738	5738	0
	5759	5761	2
	5761	5778	17
	5762	5778	16
	5778	5778	0
3	8551	8551	0
	8551	8551	0
	8551	8551	0
	8551	8552	1
	8552	8558	6
4	12812	12812	0
	12812	12813	1
	12812	12813	1
	12812	12814	2
	12813	12815	2
5	10640	10640	0
	10640	10641	1
	10640	10642	2
	10640	10643	3
	10640	10646	6

The first group experiment is searching the K shortest paths from Shanghai Hongqiao International Airport (SHA) in China to Los Angeles International Airport (LAX) in USA with the San Francisco International Airport (SFO) as the transfer airport. The second group experiment is searching the K shortest paths from PEK to Zhongshan North Station (ZSB) with the Guangzhou Baiyun International Airport (CAN) as the transferring airport. The third group experiment is searching the K shortest paths from Dongguan Station (DG) to Suzhou Station (SZ). The experiment results of constrained Yen* algorithm are shown in Table 6, Table 7 and Table 8. For constrained Yen algorithm, the first group experiment cannot be find the shortest path, the others experiments only find the first shortest path in Table 7 and Table 8.

TABLE.6

THE SHORTEST PATH FROM AIRPORT TO AIRPORT

Path number	Shortest path	Path length (km)
1	SHA→NRT→SFO→LAX	11023
2	SHA→HND→SFO→LAX	11065
3	SHA→SFO→LAX	11207
4	SHA→SFO→SJC→LAX	11207

TABLE.7

THE SHORTEST PATH FROM AIRPORT TO STATION

Path number	Shortest path	Path length (km)
1	PEK→CAN→ZSB	1983
2	PEK→WUH→CAN→ZSB	1983
3	PEK→FUG→CAN→ZSB	1986
4	PEK→CAN→XLA→ZSB	1987

TABLE.8

THE SHORTEST PATH FROM STATION TO STATION

Path number	Shortest path	Path length (km)
1	DG→CAN→HGH→SZ	1244
2	DG→SZX→HGH→SZ	1245
3	DG→CAN→WUX→SZ	1252

According to the above three experiment results, the search problem of connecting path in air-rail integration network can be found by the constrained Yen* algorithm. In table 5, NRT, HND and SJC are Nairita International Airport in Japan, Haneda International Airport in Japan and San Jose International Airport in USA respectively. In table 6, WUH, FUG and XLA are Wuhan Tianhe International Airport, Fuyang Xiguan Airport and Xiaolan Station respectively. In table 7, HGH, SZX and WUX are Hangzhou Xiaoshan International Airport, Shenzhen Bao'an International Airport and Wuxi Shuofang International Airport respectively.

Through above comparison in terms of running time, search scale and the shortest path of transfer station in air-rail integration network, we can see that constrained Yen* algorithm works better to solve the search problem of connecting path. Therefore, although with the same order of time complexity as constrained Yen algorithm, constrained Yen* algorithm is more efficient than constrained Yen algorithm in application.

V. CONCLUSION

In this paper, we describe the air-rail integration model based on the currently development situation and characteristics of Chinese railways. Then, constrained Yen* algorithm is proposed to solve the search problem of connecting path in air-rail integration network. The constrained Yen* algorithm is designed by applying the heuristic strategy of A* algorithm and the certain constraints in to Yen algorithm. The experiment results show that the search efficiency is enhanced, and the search scale is reduced in constrained Yen* algorithm. The solution of KMCSP problem for searching the connecting path in air-rail integration network can be obtained fast by the constrained Yen* algorithm.

ACKNOWLEDGMENT

I would like to thank all those who helped in the preparation of this paper. In particular, I am grateful to Prof. Xu Tao and Li Jian fu for their constructive suggestions. It is supported by the Science and Technology Projects of Civil Aviation Administration of China (No.MHRD201007) and Fundamental Research Funds for the Central Universities (No.ZXH2011B003).

REFERENCES

[1] Yusof S Z M, Asirvadam V S, Hassan M F. Flight connections multi-leg searching by adopting partial constraint satisfaction. Proceedings of International Symposium in Information Technology, vol.1, pp.1-4, 2010.

[2] Chariton C, Choi M H. Enhancing usability of flight and fare search functions for airline and travel web sites. Proceedings of the International Conference on Information Technology: Coding and Computing, vol.1, pp.320-325, 2004.

[3] Chen S, Nahrstedt K. On finding multi-constrained paths. Proceedings of IEEE International Conference on Communications, vol.2, pp.874-879, 1998.

[4] Dachuan Wei. An optimized floyd algorithm for the shortest path problem. Journal of Netwoks, vol.5, no.12, pp.1496-1504, 2010.

[5] Ke Xiong, Zhengding Qiu, Yuchun Guo. Exact algorithm for multi-constrained shortest link-disjoint paths. Journal of Software, vol.21, no.7, pp.1744-1757, 2010.

[6] Mingfu Wang. The new algorithm for finding the paths based on coding graph. Journal of Software, vol.7, no.1, pp.1-8, 2012.

[7] Yunting Lu, Zhenjun Li. An algorithm to find the *k*th shortest path in Halin Network. International Symposium on Intelligent Information Technology Application Workshops, pp.698-701, 2008.

[8] Kusetogullari H, Leeson M S, Ren W, Hines E L. K-shortest path network problem solution with a hybrid genetic algorithm: particle swarm optimization algorithm. Transparent Optical Network, pp.1-4, 2011.

[9] Berclaz J, Fleuret F, Turetken E, Fua P. Multiple object tracking using k-shortest paths optimization. Pattern Analysis and Machine Intelligence, vol.33, pp.1806-1819, 2011.

[10] Ziang Zhang, Jihong Zhao. Multi-constraint-pruning: An algorithm for finding k shortest paths subject to multiple constraints. Proceeding of the 14th Asia-Pacific Conference on Communications, vol.10, pp.1-5, 2008.

[11] Puri A, Tripakis S. Algorithms for routing with multiple constraints. Proceeding of Workshop on Planning and Scheduling using Multiple Criteria, pp.7-14, 2002:7-14.

[12] Mieghem P, Neve H D, Kuipers F A. Hop-by-hop quality of service routing. Computer Networks, vol. 37, no. 3-4, pp.407-423, 2001.

[13] Korkmaz T, Krunz M. Multi-constrained optimal path selection. Proceedings of International Conference on Computer Communications, vol. 2, pp.834-843, 2001.

[14] Liu G, Ramakrishnan K G. A* prune: an algorithm for finding k shortest paths subject to multiple constraints. Proceedings of the Conference on Computer Communications, 2001, 2:743-749.

[15] She X Y. The shortest path algorithm with constraints. Computer Applications and Software, vol. 26, no. 5, pp.236-238, 2009.

[16] Ning S. K constrained shortest path problem. IEEE Transactions on Automation Science and Engineering, vol. 1, no. 7, pp.15-23, 2010.
[17] Robinson S. Computer scientists find unexpected depths in airfare search problem. Society for Industrial and Applied Mathematics, vol. 35, no. 1, pp.1-6, 2000.
[18] Cen Zeng, Qiang Zhang, Xiaopeng Wei. GA-based global path planning for mobile robot employing A* algorithm. Journal of Computer, pp.470-474, 2012.

Tao Xu born in 1962/09/02 at Chongqing city of China, professor of the College of Computer Science and Technology in Civil Aviation University of China. Achieved M.S. in computer application technology from Nanjing University of Aeronautics and Astronautics. Research interests include theory analysis and design of the civil aviation information system, graphics and visualization technology, and software testing technology.

Xiaolu Ding born in 1988 at HeNan province of China, postgraduate of the College of Computer Science and Technology in Civil Aviation University of China.

Jianfu Li born in 1979 at HeBei province of China, lecturer of the College of Computer Science and Technology in Civil Aviation University of China.