http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (11): 2153-2159, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.2153.2159

A Two-stage Heuristic Algorithm for Locomotive Scheduling

Jie Zhang, Shaoquan Ni, Lulu Ge, Yuanyuan Wang
School of Transportation and Logistics, Southwest Jiaotong University,
Chengdu, Sichuan, China

Abstract: Locomotive scheduling problem is to assign locomotive to all the trains on the train working diagram with the fewest costs. Locomotive operation optimization is of great significance to the benefits of railway transportation. The study proves that the a feasible solution of locomotive scheduling can be transformed into the form of the locomotive routing's sequential connection by analyzing its characteristics. Accordingly, a two-stage heuristic algorithm is designed. The first stage is to solve the optimum locomotive routing connection on one station and the second stage to deal with locomotive deadheading and optimize the locomotive routing on all the stations synthetically. Compared with the actual locomotive scheduling in the railway operation practice, the results point out that the two-stage heuristic algorithm is an effective and efficient method to solve the locomotive scheduling problem.

Key words: Railway transportation, locomotive scheduling, train working diagram, heuristic algorithm

INTRODUCTION

Locomotive scheduling, determined how to assign sufficient locomotive for trains according to train working diagram, is a key problem of railway transportation tactical plan. An optimized locomotive schedule will reduce the equipped locomotive amount and decrease transportation costs.

On the base of train working diagram, locomotive schedule influenced by the locomotive route, locomotive type, locomotive operation time standard at station, etc. About the Locomotive Scheduling Problem (LSP), many researchers have obtained many significant results. Booler (1980) formulated a multi-commodity flow model and then presents a heuristic method for solving. Wright (1989) designed three assignment algorithms to solve this problem. Ziarati *et al*. (1997, 1999) proposed a Dantzig-Wolfe decomposition method for operational strategic locomotive assignment and then designed a feedback neural network based on Ising Mean Field Approach for LSP of CN railway system. Cordeau *et al*. (2001) assigned the locomotive together with car assignment by a branch-and-bound method. Ahuja *et al*. (2002) and Vaidyanathan *et al*. (2008) solved the real-life LSP of CSX in USA. For locomotive schedule in China (Yang, 1990) proposed a tabu method and with the help of signing the residence time matrix, a optimized solution was quickly found. Based on Yang (1990), Lu *et al*. (1998), Xiao (1999) and Xie (2010) presented some algorithms for formulating unpaired locomotive diagram, respectively. Shi and Hu (1996) designed a linear algorithm for locomotive schedule, which can reduce the locomotive allocation amount. Yan and Cui (2006) reviewed the latest locomotive scheduling model.

For locomotive scheduling problem, the literature mentioned above have designed sophisticated methods such as linear programming, intelligent algorithm etc for solving the problem. But by exploring the characteristics of the problem, solutions of the problem can also be easily found. This study aims to solve LSP with a new method based on the locomotive routing's sequential connection characteristics.

PROBLEM DESCRIPTION

In China, locomotives go around between districts stations in the fixed railway line according to a certain route, which is called as locomotive route. There are one or several locomotive types in a locomotive route but they all can replace each other. Based on transportation planning stage, locomotive schedule has two types, which are the basic schedule and daily-shift locomotive schedule. The basic schedule is a cyclical locomotive working plan for a period of time and the daily-shift

Corresponding Author: Jie Zhang, School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, Sichuan, China

locomotive schedule is for daily locomotive dispatching. The study only discusses cyclic locomotive scheduling in the tactical planning level.

Locomotive schedule is locomotive working plan and it is designed according to timetable, locomotive route and other restrictions. The objective of locomotive scheduling is to allocate all trains with the minimum cost, namely, minimize the Locomotive Turnaround Time (LTT), or minimize the number of locomotives. At the same time, locomotive scheduling must meet the following requirements:

- As the train working diagram is given, the suitable number of locomotives must be assigned for trains to meet the requirement of train traction power and train speed
- While the locomotives do servicing work in the service depot and locomotive turnaround depot, it must meet the requirement of the technical operation time according to the certain locomotive route
- According to axle load requirements, one train is allowed to attach 4 locomotives at most
- Dead-hauling in two opposition direction at the same time is not allowed

In addition, the bi-locomotive set is a group of locomotives assembled to pull train-segment. When the bi-locomotive set is disassembled into two single locomotives, an extra technical operation time is required. Thus, once the train-segment taken by the bi-locomotive set is late, the other trains which are assigned to these two locomotives will be influenced.

MATHEMATICAL MODEL

The process of locomotive scheduling is to match the locomotive of inbound trains with outbound trains at the locomotive turnaround station, which is labeled as Locomotive Routing Connection (LRC). LRC tell us the connection relationship between the inbound locomotive and outbound train-segment. Therefore, the inbound train and the outbound train at a turnaround station is to be split into two disjoint vertex sets A and B and view locomotive route as edge, thus, the locomotive schedule problem is transformed into bipartite graph G, which is shown in Fig. 1. By solving bipartite graph matching of each station, a complete locomotive scheduling is finally formulated. The model is shown as follows:

As is shown above, for a locomotive turnaround section, let S_k (k = 1..P) represent the locomotive turnaround station. Let L_i (k = 1..M) represent train-segment. Let v_{ik} represent the starting station of train-segment L_i and T_{di} as departure time, Let u_{ik} represent the section arriving station of train-segment L_i and T_{ai} as arrival time. For the train-segment L_i, t_i represent the train travel time, n_i represent the number of locomotive number and f_i as the number of locomotives will be attached:

$$u_{ik} = \begin{cases} 1 & \text{if the train } L_i \text{ arrives at the station k} \\ & \text{and hangs up the locomotives} \\ 0 & \text{otherwise} \end{cases}$$

$$v_{ik} = \begin{cases} 1 & \text{if the train } L_i \text{ departs from the station k} \\ & \text{and hangs up the locomotives} \\ 0 & \text{otherwise} \end{cases}$$

Let x_{ij} represent the locomotive routing connection variables:

$$x_{ij} = \begin{cases} 1 & \text{if the locomotive pulls the train } L_i \\ & \text{and continues the train } L_j \\ 0 & \text{otherwise} \end{cases}$$

After the locomotive pulls the train-segment L_i and continues L_j, the time of the locomotive dwell at the station is:

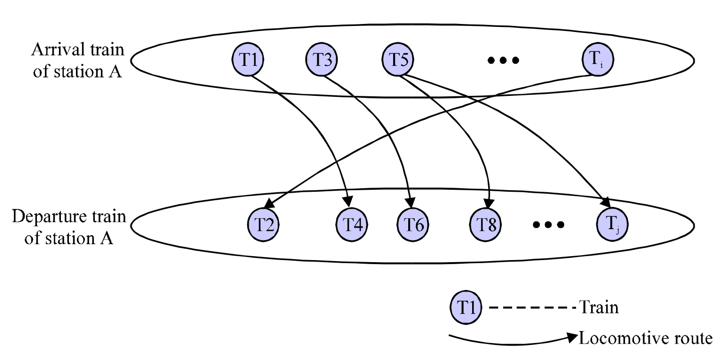

Fig. 1: Bipartite graph model for LSP

$$t_{ij} = \begin{cases} T_{dj} - T_{ai} & \text{if } T_{dj} > T_{ai} \\ T_{dj} - T_{ai} + 1440 & \text{if } T_{dj} < T_{ai} \end{cases}$$

where, T_k is the shortest locomotive service time standard at the turnaround station k.

According to the description of locomotive scheduling, the objective is to minimized the locomotive turnaround time, which is shown as follows:

$$Z = \text{Min}\left(\sum_{i=1}^{M}\sum_{j=1}^{M} n_{ij}t_{ij}x_{ij} + \sum_{i=1}^{M} t_i(n_i + f_i)\right) \quad (1)$$

Subject to :

$$\forall i \in M, \quad \sum_{j=1}^{M} x_{ij} = 1 \quad (2)$$

$$\forall j \in M, \quad \sum_{i=1}^{M} x_{ij} = 1 \quad (3)$$

$$\forall t_{ij} > 0, \quad \sum_{k=1}^{P} v_{jk}t_{ij} > T_k \quad (4)$$

$$\forall x_{ij} = 1, \quad \sum_{k=1}^{P} u_{ik}v_{jk} = 1 \quad (5)$$

Constraints (2) indicates that an inbound locomotive could pull one and only one outbound train-segment; Constraints (3) indicates that an outbound locomotive could pull one and only one inbound train-segment once; Constraints (4) indicates that the locomotive routing connection must satisfy the locomotive service time standard at the station; Constraints (5) indicates that the train-segments in one locomotives routing connection should be at the same station.

ALGORITHM

Since the locomotive routing at the station of LSP is a bipartite graph matching problem, Hungarian algorithm is the traditional algorithm for this kind of problem. Since the locomotive scheduling involves mass data and large coefficient matrix, the computational efficiency will be very slow. However, it is a special assignment problem, the study is to design appropriate algorithm so as to get the result quickly by analyzing the routing connection characteristics of locomotive schedule and then proposes a two-stage heuristic algorithm for locomotive scheduling.

Sequential characteristic of the locomotive routing connection: The LSP is to work out the locomotive routing connections at each turnaround station, namely, the locomotive routing connections between inbound trains and outbound trains. The result is the matching sets of inbound trains and outbound trains and the objective is to minimize the total locomotive turnaround time. For the locomotive routing connection problem, any solution can be transformed into the solution called as " first to come and first to go", while the total locomotive turnaround time is fixed.

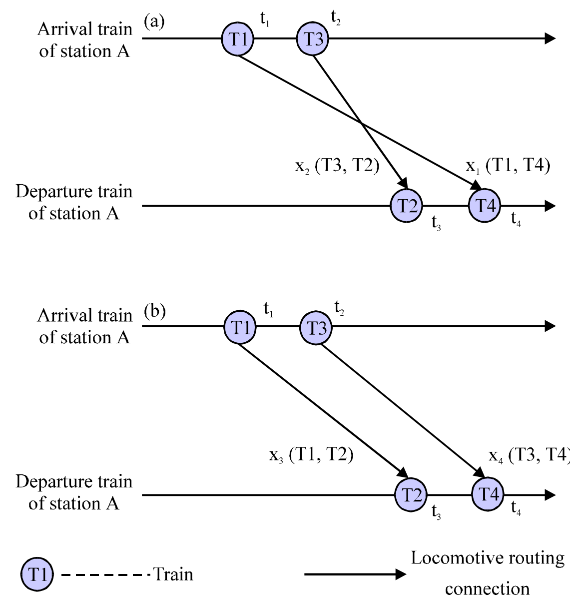

Fig. 2(a-b) : Locomotive routing connection

As the locomotive routing connection shown in Fig. 2a, where the locomotive routing connection x_1 (T1, T4) and x_2 (T3, T2), which can be transformed into x_3 (T1, T2) and x_4 (T3, T4) shown in Fig. 2b. In the Fig. 2, the total time of locomotives stay at the station in Fig. 2a is the same as Fig. 2b and the total locomotive turnaround time still is $t = t_4+t_3-t_2-t_1$ but the service time is $x_1>x_3>x_4>x_2$. Obviously, the locomotives service time at the station is balanced in the Fig. 2b, which will be beneficial to the locomotive dispatch. On the contrary, the service time of x_2 is the shortest in the Fig. 2. Once the train delayed, it is difficult to continue the locomotive routing connection, thus, the Fig. 2b is much better than Fig. 2a.

From above discussion, a conclusion is drawn as follows: Once the locomotive routing lines intersect, it can be transformed into the form of disjoint in a bipartite graph model. At the turnaround station, let T_i (i = 1..., m) represent the inbound trains in the order of the arrival time and T_j (j = 1..., m) represent the outbound trains in the order of the arrival time. Then, when a locomotive routing connection is represent as (T_i, T_j), any feasible solution of the locomotive routing connection of the station can be expressed as the form such as ···, (T_{i-1}, T_{j-1}), (T_i, T_j),

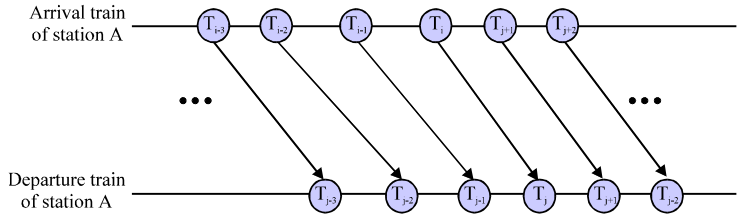

Fig. 3: Sequential locomotive routing connections

$(T_{i+1}, T_{j+1})\cdots$, that is, the arrived locomotives were allocated to the outbound trains sequentially. As is shown in the Fig. 3.

Algorithm flow: The objective function is minimizing total locomotive turnaround time, which is composed of the pulling locomotive running time:

$$\sum_{i=1}^{M} t_i n_i$$

the deadheading locomotive running time:

$$\sum_{i=1}^{M} t_i f_i$$

and the locomotive service time:

$$\sum_{i=1}^{M}\sum_{j=1}^{M} c_{ij} t_{ij}$$

For a timetable, the pulling locomotive running time is a constant. Therefore, the locomotive schedule can be transformed into two sub-problems: the locomotive routing connection problem and the locomotive deadheading problem. Thus, the algorithm can be divided into two stages. The task of the first stage is to formulate the locomotive routing connection scheme, aiming to minimize the total locomotive dwell time except necessary service time at each turnaround station. The task of the second stage is to work out the locomotive deadheading scheme with comprehensive optimization. The whole process of the algorithm is shown in Fig. 4.

Algorithm of locomotive routing connection for the first stage: Based on the discussion above, in the bipartite graph model, the locomotive routing connection lines must not intersect each other, just like parallel lines. Thus, once one locomotive routing connection is determined, all the others can be deduced sequentially according to the locomotive arrival and departure sequence. If one of the routing connections can't satisfy the constraints, the solution is invalid. The best solution can make sure that the total locomotive dwell time is the least.

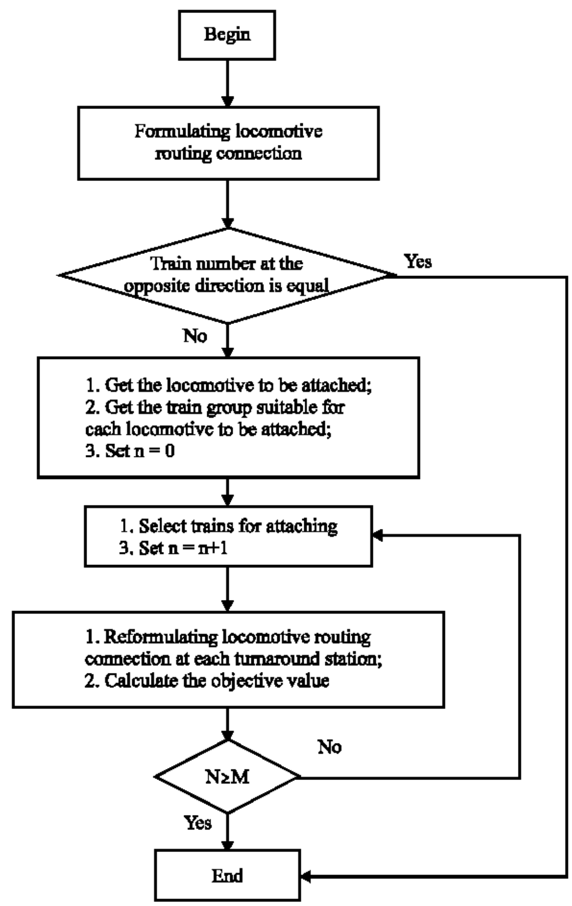

Fig. 4: Algorithm flow chart

How to determine the first locomotive routing connection is a key problem. It is the locomotive dense arrival or departure point-in-time that is selected as the starting point for the first locomotive routing connection. The method will be introduced in detail as follows:

At the turnaround station, the inbound trains and outbound trains are sorted according to the order of arrival/departure time and mark each train with the value. As is shown in Fig. 5, from the first train on the left, mark

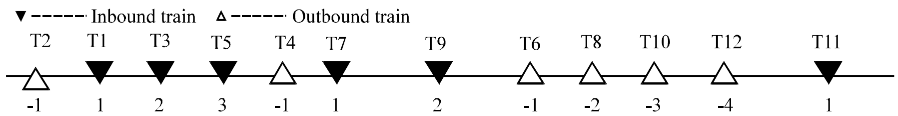

Fig. 5: Locomotive dense arrival and departure point

every train with integer value which is the positive value for inbound trains and negative value for outbound trains. The first train number is 1. If a train and its previous train is the same train type (arrival type or departure type), the train's value is the previous value plus 1, or-1. Repeat this process until all the trains is marked.

Then the maximum absolute value is chosen and it was the locomotive densely arrival and departure point. As is shown above in Fig. 5, the train with the maximum absolute value is T12 and the nearest inbound train before T12 is T9. Thus, the first locomotive routing connection is x_{ij} (T12,T9).

The algorithm of locomotive routing connection is as follows:

Step 1: Sort the inbound trains at station k by the arriving time, then all the trains are included in the collection U_d, the size of which is denoted as P. Sort the outbound train at station k by the departing time, then are included in the collection U_a, the size of which is denoted as Q, set t = 0

Step 2: Get the locomotive dense arrival or departure point-in-time as mentioned above, then get the first locomotive routing connection; sort the trains in set U_d and set U_a started from the locomotive dense arrival or departure point-in-time

Step 3: Set F = P-Q, f = 0, R = max {P, Q}, $r_a = 1$, $r_b = 1$, initialize the locomotive routing connection matrix X, $\forall x_{ij} \in X$, set $x_{ij} = 0$

Step 4: Set p = 1, q = r_b, initialize the locomotive routing connection matrix X'_k, $\forall x'_{ij} \in X'_k$ and set $x'_{ij} = 0$

Step 5: Calculate the locomotive staying time t_{ij} of the locomotive routing connection x'_{pq}

Step 6: If $t_{ij} < T_k$, go to step 7; otherwise go to step 8

Step 7: If r_a = p, set $r_b = r_b+1$, go to step 4; otherwise set r_a = p, q = q+1, go to step 5

Step 8: If p = r_a, set x'_{pq}, go to step 14

Step 9: If f = F, set r_a = p, q = q+1, go to step 5

Step 10: Calculate the locomotive staying time $t_{i+1,j+1}$ of $x_{p+1,q+1}$

Step 11: If $t_{i+1,j+1} < T_k$, Set r_a = p, q = q+1, go to step 5

Step 12: Set x'_{pq}

Step 13: Set p = p+1, if p>P, go to step 15; otherwise go to step 5

Step 14: For $\forall x'_{ij} \in X'_k$, find the correspond element x_{ij} in the matrix C, set $x_{ij} = x'_{ij}$

Step 15: The matrix C is the solution, end

Locomotive deadheading scheme for the second stage: If the number of locomotives between up and down direction is unpaired, locomotive deadhead or light travel is required. But light travel would cost train lines and then influence the railway line capacity. Thus, locomotive deadheading is brought in to reposition locomotives from excess locations to deficit locations. The problem of selecting trains for deadheading, which is a combinatorial problem, the number of solutions would be very large. Here, a priory select groups is first constructed and then a heuristic algorithm is designed to get the optimized solution.

After finishing the locomotive routing connection of a station according to algorithm of the first stage, then it can get the excessive locomotives needs to be attached and the non-assigned outbound trains needs allocating locomotive, which is shown as follows:

On the locomotive diagram, make the arrival time of the excessive move the service time T_k forward and make the departure time of deficit move the service time T_k backward, then make a line connect the two time-point. After that, the trains which intersected with the connecting line could make up a train group V_r. For example, as is shown in Fig. 6, at station A, T2 is the excessive train and in station B, T12 is the not-assigned train, then according by the intersecting lines, the train group that can deadhead locomotives is V_r{T3,T5}. The trains in group V will be better for deadheading because the excessive locomotive from one station can less waiting for outbound train after arriving than deadheaded by others.

For the unpaired locomotive diagram, each unpaired train will generate a priority deadheading trains group. Among each groups a train is chosen, then a feasible solution of the locomotive deadheading solution formed. Based on the locomotive deadheading solution, the locomotive diagram have changed into a train paired diagram and then locomotive scheduling can be worked out by formulating the locomotive routing connection with algorithm of the first stage. Calculating and comparing the objective function value by iteration, finally the optimum solution is gotten. If the number of

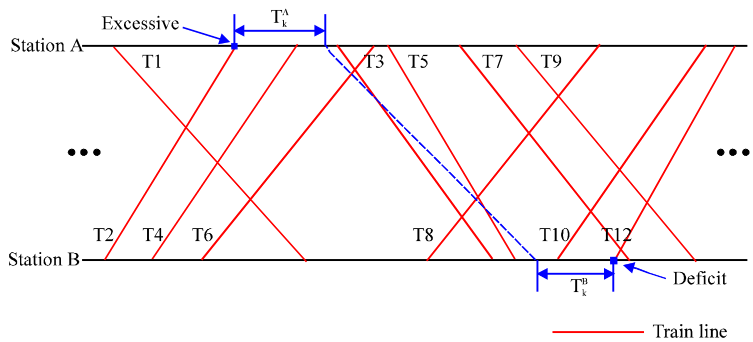

Fig. 6: Trains selection for locomotive deadheading

unpaired trains is large, the quantity of groups will be great, then the solution space become very large. As a result, the number of iterations can be set as termination condition.

At station k, the number of inbound trains is P_k, the number is outbound trains is Q_k. Defining cycle-index coefficient H, the algorithm of locomotive deadheading scheme is expressed as follows:

Step 1: $\forall k \in K$, find the excessive trains or deficit trains at station k in the order of time with the algorithm mentioned in 4.2, include them into collection U_p and U_q, set n = 1, m = (P_k-Q_k) *H, t = 8

Step 2: Assemble the elements of U_p and U_q to make up train groups collection V

Step 3: For each train group element in collection V, select a train to construct a deadheading solution $\{y_1, y_2, y_3,...\}$

Step 4: $\forall k \in K$, formulating the locomotive routing connection at station k with the algorithm in the first stage, then get the process solution

Step 5: Calculate the total locomotive turn around time t_n of

Step 6: If $t_n<t$; set $t = t_n$

Step 7: Set n = n+1, if n<m, go to step 3

Step 8: end

NUMERICAL EXPERIMENT

The data of North JiaoLiu railway line in China is adopted to work out the locomotive schedule. The locomotive type is SS6B and 6K. The train number in the up direction is 86 and the train number in the down direction is 74. The locomotive routing mode is half-cycle. The locomotive service time standard is 95 min at Luoyangbei station, 100 min at Xiangyangbei station, 115 min at Pingdingshan station and 40 min at other

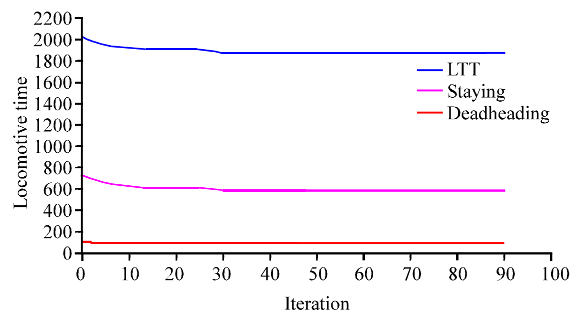

Fig. 7: Optimal solution by iteration

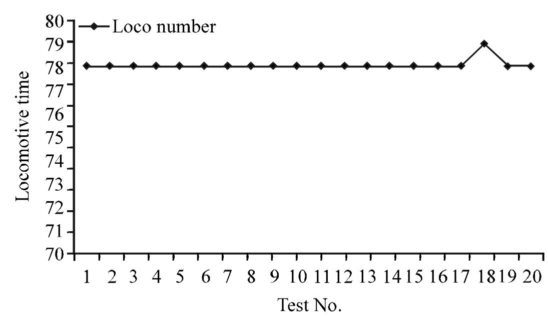

Fig. 8: Summary of the results of some efforts

turnaround station. By programming with VC++, the process and the results are shown as follows.

Table 1 shows the solving process of the optimal algorithm at each station for the first stage. Fig.7 shows us that the optimal solution can be obtained very quickly. The better solution has occurred when the iteration time is 30.

Formulating the locomotive scheduling many times, some of the test results is shown in Fig. 8. Among them, most get the optimum result. Table 2 shows the comparison between the results of solving by the algorithm of this study and the actual locomotive

Table 1: Result of first stage

Depot station	Service time (min)	Staying (h)	Avg. staying (min)	CPU (ms)
Luoyangbei	95	268.1	201.0	50
Pingdingshan	115	252.8	213.6	20
Xiangyangbei	100	44.5	296.7	82

Table 2: Result comparison with actual data

Comp.	LTT (h)	Loco. No.	Avg. staying (min)	Deadhead (h)
This study	1872	78.0	212.0	90.6
Actual	1992	83.0	255.8	93.7
Percent (%)	94.0	94.0	82.9	96.7

scheduling. It can be seen from the comparison that the total locomotive turnaround time would be save accounts for 6% and 5 locomotives saved, which can save about $6 million, which show that the algorithm is better than the actual locomotive scheduling.

CONCLUSION

This study studied the locomotive scheduling problem in China. It was proved that the solution of locomotive routing connection at a station can be transformed into the form of sequential type but no change for the total locomotive turnaround time. Accordingly, a two-stage heuristic algorithm is designed for locomotive scheduling. Finally, the operation practical data of China Railway is used in numeric experiment and it shows that the algorithm can work out the optimized solution quickly, which was better than the actual locomotive scheduling. The LSP under condition of unpaired train numbers is very difficult, especially when the locomotive maintenance business is considered, which is under research now.

ACKNOWLEDGMENT

This study was supported by Natural Science Foundation of China (No.60870004).

REFERENCES

Ahuja, R.K., J. Liu, J.B. Orlin, D. Sharma and L.A. Shughart, 2002. Solving real-life locomotive scheduling problems. Working Paper 4389-02, MIT Operations Research Center, Massachusetts Institute of Technology, USA.

Booler, J.M.P., 1980. The solution of a railway locomotive scheduling problem. J. Oper. Res. Soc., 31: 943-948.

Cordeau, J.F., G. Desaulniers, N. Lingaya, F. Soumis and J. Desrosiers, 2001. Simultaneous locomotive and car assignment at VIA Rail Canada. Transp. Res. Part B: Methodol., 35: 767-787.

Lu, H., M. Yang and J. Mao, 1998. Working out of non pair locomotive working diagrams with a computer. J. Southwest Jiaotong Univ., 33: 20-24.

Shi, F. and A. Hu, 1996. A linear algorithm to deploy a locomotive circulating diagram. J. China Railway Soc., 8: 18-24.

Vaidyanathan, B., R.K. Ahuja, J. Liu and L.A. Shughart, 2008. Real-life locomotive planning: New formulations and computational results. Transp. Res. Part B: Methodol., 42: 147-168.

Wright, M.B., 1989. Applying stochastic algorithms to a locomotive scheduling problem. J. Oper. Res. Soc., 40: 187-192.

Xiao, L., 1999. Computerized planning of the optimal locomotive working diagram. J. Changsha Railway Univ., 17: 52-57.

Xie, J., 2010. Optimal algorithm of unpaired locomotive working diagram with compact return locomotives at return station. J. China Railway Soc., 18: 9-13.

Yan, H. and Y. Cui, 2006. Optimization model of locomotive working diagram. China Railway Sci., 27: 123-128.

Yang, X., 1990. A simple algorithm of optimizing locomotive working diagram. Railway Transp. Econ., 12: 31-34.

Ziarati, K., F. Soumis, J. Desrosiers and M.M. Solomon, 1999. A branch-first, cut-second approach for locomotive assignment. Manage. Sci., 45: 1156-1168.

Ziarati, K., F. Soumis, J. Desrosiers, S. Gelinas and A. Saintonge, 1997. Locomotive assignment with heterogeneous consists at CN North America. Eur. J. Oper. Res., 97: 281-292.