

# Solving train formation plan problem in freight transportation corridors

Xinhong Zhang

<sup>1</sup>Graduate department  
China Academy of Railway Sciences  
Beijing, China

<sup>2</sup> Division of price management  
Nanchang Railway Administration  
Nanchang, China  
happyxh@sina.com

Liujiang Kang, Wei Liu, Yu Yao

School of Traffic and Transportation  
Beijing Jiaotong University  
Beijing, China  
ljkang@bjtu.edu.cn  
13114210@bjtu.edu.cn  
402983962@qq.com

**Abstract**—The train formation plan develops the train services and assigns the transportation demands whose optimal solutions are normally difficult to find. In this paper, a new linear programming model for the train formation plan problem is proposed by the use of mixed integer programming methods. In addition, considering the long computing time by the mixed integer programming solver, an effective heuristic algorithm is designed to explore the solution space, where a neighborhood based search method evaluates, selects, and implements the moves. To verify the proposed mixed integer programming model and the solution method, numerical problems have been solved, and results are compared with those obtained by the ILOG CPLEX 12.6 version.

**Keywords**—freight transportation; train formation plan; mixed integer programming model; CPLEX

## I. INTRODUCTION

Since the 1970s, with the increasingly heavy problem of energy crisis, urbanization issued resources shortage, traffic congestion, traffic accidents and environment, railway transportation has attracted much attention in both developed and developing countries. Due to the rapid expansion of railway networks, the level of transport supply has been improved; the transportation capacity for long-distance or large-scale freight transportation has been increased; and the social benefit as well as the city-to-city connections has also been increased.

Railway freight transportation is a classical application area of operations research; a recent overview is given by Nemani and Ahuja [1]. Fig. 1 shows the hierarchical planning for freight trains in demand-driven railways, which is divided into three levels, i.e., strategic, tactical and operational levels. The train formation is classified into a tactical level problem that consists of station selection, accumulation and reformation tasks. The train formation plan (TFP) is one of the most important and difficult studies in railway transportation planning. The TFP models determine trains that need to be assembled, including the definition of assignments, frequencies, sizes, and profiles of shipments, and traction according to the proposed objective, physical, and operational

constraints of the railway network in order to meet the demands in the specific demands [2-3].

The TFP problem can be extended to a special case of network routing problem, whose goal is to make services and operations to satisfy demands and ensure the profitability of the companies [4]. Generally it is difficult to find the optimal solutions for the large-scale TFP problem, limited to the memory and CPU time. Therefore, most studies use the heuristic and hybrid algorithms, such as [5] and [6]. Study [6] provided a slope scaling heuristic that combines a Lagrangian perturbation scheme with intensification and diversification mechanisms based on long-term strategy; and study [7] presented a heuristic local branching for the capacitated fixed-charge network routing problem.

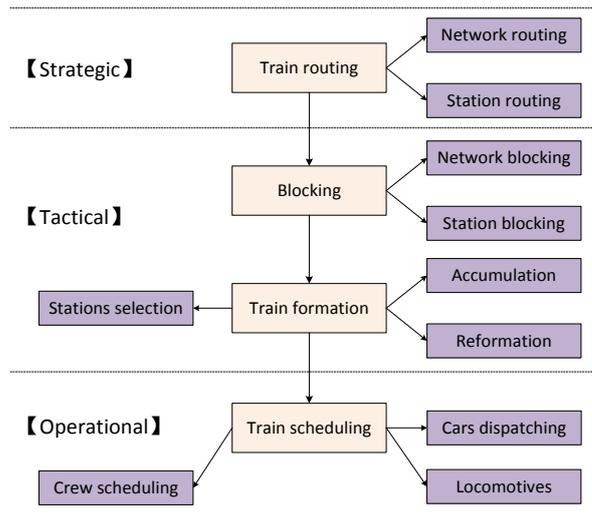


Fig. 1. Planning levels of demand-driven railways

## II. LINEAR PROGRAMMING MODEL

### A. Model objective function

Define  $W = (S, T)$  as a direct railway network with a set of stations  $S$  and a set of trains  $T$ . The objective function (1) minimizes the total cost of train accumulation time and reformation time in the railway network. As shown in objective function (1), train accumulation cost equals the sum of the unit time of accumulating one vehicle on one train at station  $s$  ( $t_{acc}^s$ ) multiplying the number of vehicles accumulated on train  $t$  that ships from station  $s$  to station  $s'$  ( $c_{ss'}^t$ ). Train reformation cost equals the sum of the unit time of reformulating one vehicle on one train at station  $s$  ( $t_{ref}^s$ ) multiplying the number of vehicles reformulated from train  $t$  to train  $t'$  at station  $s$  ( $r_{tt'}^s$ ).

$$\text{Min } f(c, r) = \sum_{s \in S} \sum_{s' \in S} \sum_{t \in T} t_{acc}^s \cdot (c_{ss'}^t - \delta) + \sum_{s \in S} \sum_{t \in T} \sum_{t' \in T} t_{ref}^s \cdot r_{tt'}^s \quad (1)$$

### B. Model constraints

Constraints (2) can directly eliminate the accumulation cost of indirect trains  $t$  running from station  $s$  to station  $s'$ . Constraints (3) state that the total number of accumulated vehicles in the network should be equal to the planned number of vehicles  $C$ . Constraints (4) set the limitation of the number of vehicles accumulated on a train. Constraints (5) ensure that the number of vehicles, which can be assembled in each station, does not exceed the station capacity  $\phi_s$ . Constraints (6) and (7) set limitations for vehicle reformulations. Equation (8) shows that  $\beta_{ss'}^t$  is a binary variable that equals one if train  $t$  can be run as a direct train (origin  $s$  and destination  $s'$ ). With  $\beta_{ss'}^t$  so defined, the values of  $\beta_{ss'}^t$  can be calculated in Equation (9), where  $\delta$  represents the required minimum number of vehicles on a direct train. When  $c_{ss'}^t$  is smaller than  $\delta$ , Equation (9) sets  $\beta_{ss'}^t = 0$ , which means train  $t$  is an indirect train. When  $c_{ss'}^t$  is larger than  $\delta$ ,  $\beta_{ss'}^t = 0$  or  $\beta_{ss'}^t = 1$ , which means train  $t$  can be assembled as a direct train.

$$\phi_{ss'}^t = c_{ss'}^t - \delta + 1 \geq 0. \quad (2)$$

$$\sum_{s \in S} \sum_{s' \in S} \sum_{t \in T} c_{ss'}^t = C. \quad (3)$$

$$\sum_{s' \in S} c_{ss'}^t \leq \chi_s. \quad (4)$$

$$\sum_{t \in T} \sum_{s' \in S} c_{ss'}^t \leq \phi_s. \quad (5)$$

$$\sum_{t \in T} r_{tt'}^s \leq \gamma_t^s. \quad (6)$$

$$\sum_{s \in S} \sum_{t \in T} r_{tt'}^s \leq \pi_s. \quad (7)$$

$$\beta_{ss'}^t = \begin{cases} 1, & \text{if train } t \text{ can be a direct train.} \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

$$c_{ss'}^t - \delta - M \cdot (\beta_{ss'}^t - 1) \geq 0 \quad (9)$$

## III. SOLUTION METHOD

In this section, we design a memory-less heuristic algorithm to solve the proposed model. Although the heuristic algorithm does not guarantee to converge on global optima, it can achieve a high quality solution in a reasonable CPU time.

### A. Pseudo-code for the heuristic algorithm

The main structure of the heuristic algorithm pseudo-code is illustrated in Fig. 2. As can be observed, *solution* (*i*) and *initial\_solution* represent the current and the initial solutions, respectively, and *objective* (*i*) and *best\_objective* represent the current and the best objective values. In addition, *iteration* (*i*) records the number of iterations. In the main loop, a neighborhood search method as well as a mutation operator is executed, and the generated solutions will be accepted if they meet the constraints. Then the best solution in this iteration will be selected out. After that the number of iterations will be updated. At the end of the algorithm, the best found solution as well as the objective value is returned.

---

```

Let solution(i) = Generate initial_solution;
Let objective(i) = f(solution(i));
Let best_objective = objective(i);
Let iteration(i) = iteration(0);
// main loop
While (iteration(i) <= iteration(maximum))
{
    Execute Neighborhood Search Method;
    Execute Mutation Operator;
    Select the best solution among generated solutions by the above methods;
    Update iteration(i) = iteration(i+1);
}
// end of main loop
Output solution(iteration(maximum));
Output best_objective;
End.
```

---

Fig. 2. Pseudo-code for the proposed heuristic algorithm

## IV. NUMERICAL EXAMPLE PROBLEM

In this section, an example test is designed to illustrate the effectiveness and efficiency of the proposed model. Fig. 3 is the map of the sample network, which consists of six stations and two transportation routes (route A-B-C-D-E and route A-B-C-D-F). Each station contains four tracks. The transportation demands of origin-destination (O-D) pairs, e.g., A-B, A-C, A-D, etc. are given in the map. The model parameters are set as follows.

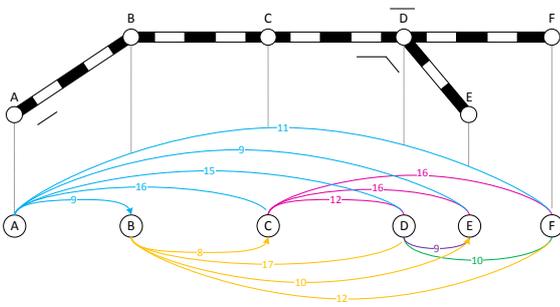


Fig. 3. Map of the example and transportation demands of O-D pairs

**A. Experimental results**

The optimum TFP is illustrated in Fig. 4. Direct train A-C carries the freight cars A-to-C, A-to-D, A-to-E, and A-to-F (the number of cars are shown above the arrows). Similarly, the cars carried by direct train B-D, train C-E and train C-F are given in the figure. In addition to these direct trains, the remainder cars are carried by the indirect trains operated in each segment.

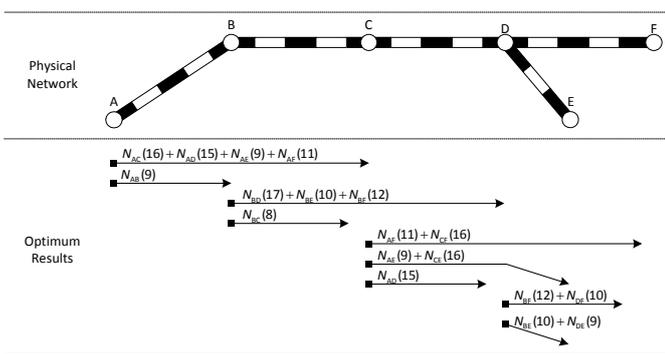


Fig. 4. The optimum train formation plan for the sample network

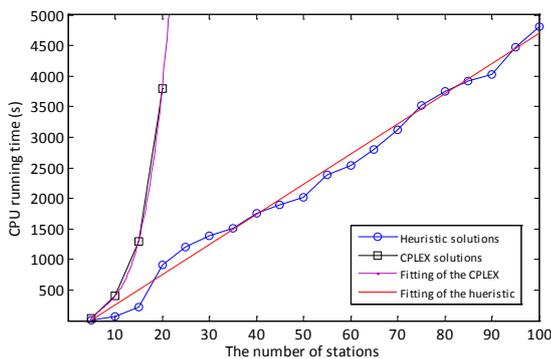


Fig. 5. Computing complexity of the heuristic and the CPLEX

The required CPU time for each problem size is depicted in Fig. 5, where the x-axis represents the number of stations in the corridor, and the y-axis shows the corresponding CPU times. For the solutions obtained by the CPLEX, there are only four problem sizes tested successfully within the fixed time, i.e., the maximum number of stations is twenty, and the

fitting curve is an exponential type. Moreover, we fit the values obtained by the heuristic algorithm, which returned 20 pairs of results, and the curve is a linear one. According to Fig. 5, we observe that the CPLEX is difficult to solve the large-scale TFP problem because of the computing time complexity (then leading to memory problem). However, the heuristic algorithm is capable of solving large-scale real-world problems.

**V. CONCLUSIONS**

In this paper, a nonlinear programming model for the TFP problem is proposed, and then the model is linearized by the use of MIP methods. In addition, a heuristic algorithm is designed to explore the solution space, where a neighborhood search method evaluates, selects, and implements the moves. The validity and the efficiency of the algorithm are tested over a series of computational experiments on a sample case.

In addition, this paper compares the heuristic and the CPLEX that capture good TFP results. However, retrieving a feasible solution in a shorter time is another challenge to algorithms. Improvements to the heuristic, creating hybrid algorithms or developing an adaptive approach can be considered in future research.

Finally, modeling the TFP together with train timetabling as well as train movements in to and out from stations in more detail will drive the TFP problem to be more practical. These extensions need to be studied in greater depth.

**Acknowledgment**

Many people were involved in the work leading to this paper, and the authors would in particular like to thank the transportation department at Beijing railway bureau for providing the data and details on Beijing-shanghai railway transportation corridor.

**References**

- [1] A.K. Nemani, R.K. Ahuja, "OR models in freight railroad industry," Cochran JJ, Cox LA, Keskinocak P, Kharoufeh JP, Smith JC, eds. Wiley Encyclopedia of Operations Research and Management Science (John Wiley & Sons, Inc., New York). 2011.
- [2] M. Yaghini, M. Momeni, and M. Sarmadi, "Solving train formation problem using simulated annealing algorithm in a simplex framework," Journal of Advanced Transportation, vol. 48, pp. 402-416, 2014.
- [3] M. Bohlin, S. Gestrelus, F. Dahms, M. Mihalák, and H. Flier, "Optimization methods for multistage freight formation," Transportation Science, Article in advance, pp. 1-18, 2015.
- [4] T.G. Crainic, "Service network design in freight transportation," European Journal of Operational Research, vol. 122, pp. 272-288, 2000.
- [5] I. Ghamlouche, T.G. Crainic, M. Gendreau, "Cycle-based neighborhoods for fixed-charge capacitated multicommodity network design," Operations Research, vol. 51, pp. 655-667, 2003.
- [6] T.G. Crainic, M. Gendreau, G. Henu, "A slope scaling/ Lagrangian perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design," Journal of Heuristics, vol. 10, pp. 525-545, 2004.
- [7] I. Rodríguez-Martín, J.J. Salazar-González, "A local branching heuristic for the capacitated fixed-charge network design problem," Computers & Operations Research, vol. 37, pp.575-581, 2010.