JOURNAL OF TRANSPORTATION
SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY
Volume 9, Issue 6, December 2009
Online English edition of the Chinese language journal

Available online at www.sciencedirect.com
ScienceDirect

Cite this article as: J Transpn Sys Eng & IT, 2009, 9(6), 136−141.

RESEARCH PAPER

Bottlenecks Detection of Track Allocation Schemes at Rail Stations by Petri Nets

JIA Wenzheng[1,2], MAO Baohua[1,*], HO Tinkin[2], LIU Haidong[1], YANG Bo[3]

1 MOE Key Laboratory for Urban Transportation Complex System Theory & Technology, Beijing Jiaotong University, Beijing 100044, China
2 Department of Electrical Engineering, Hong Kong Polytechnic University, Hong Kong, China
3 Nancang Station, Beijing Railway Bureau, Tianjin 300402, China

Abstract: Robustness of the track allocation problem is rarely addressed in literatures and the obtained track allocation schemes (TAS) embody some bottlenecks. Therefore, an approach to detect bottlenecks is needed to support local optimization. First a TAS is transformed to an executable model by Petri nets. Then distrubances analysis is performed using the model and the indicators of the total trains' departure delays are collected to detect bottlenecks when each train suffers a disturbance. Finally, the results of the tests based on a rail hub linking six lines and a TAS about thirty minutes show that the minimum buffer time is 21 seconds and there are two bottlenecks where the buffer times are 57 and 44 seconds respectively, and it indicates that the bottlenecks do not certainly locate at the area where there is minimum buffer time. The proposed approach can further support selection of multi schemes and robustness optimization.

Key Words: rail stations; track allocation schemes; bottlenecks detection; Petri nets

1 Introduction

Routing trains at railway stations is a common problem in railway scheduling and operation[1,2]. The objective is to allocate conflict-free inbound & outbound routes and platforms to trains while ensuring the operations safety and achieving reasonable infrastructure utilization. Solutions to this track allocation problem (TAP) are referred to as track allocation schemes (TAS) in the paper. The objectives in the literatures are normally the maximum platforms preference, workload balance of the platform tracks and the minimum of shunting trains. Buffer times imply the delay-tolerance of a TAS; however, they are rarely addressed in previous studies. Disturbances are inevitable in real-life operations and the TAS delay-tolerance is good if the sensitivity to disturbances is low, and the performance is referred to as robustness[3] which is an emerging issue in railway and air transport timetabling. The train scheduling in China mainland railways is reviewed in the literature[4] where an integrated approach is proposed and the TAP development is also discussed. In addition, the TAS quality has influence on the rescheduling in real-life operations[5]. Therefore, the bottlenecks in a TAS should be detected and eliminated to improve the robustness of a TAS.

The approach to detecting bottlenecks in a TAS is proposed in the paper. A TAS is modeled by timed colored Petri nets and delay propagation under disturbances is collected, and then bottlenecks can be identified using departure delays. Finally, case study based on a real station layout in China mainland railways and a thirty minutes timetable show that the approach enables identification of the bottlenecks in a TAS. The results would provide supports to TAS evaluation and robust TAP.

2 Track allocation schemes

Conflict free inbound & outbound routes to trains are formulated in a TAS. The set of trains is denoted by T, and the set of inbound and outbound routes are R^I and R^O, respectively. The arrival and departure time for the train $t \in T$ is denoted by a_t and d_t respectively, and the inbound and outbound routes for the train t are i_t and o_t, and the job for the train t can be described by the five-tuple $\langle t, a_t, d_t, i_t, o_t \rangle$. Thus, the TAS including all the trains can by modeled by the following formula.

Received date: Jun 3, 2009; Revised date: Sep 27, 2009; Accepted date: Oct 11, 2009
*Corresponding author. E-mail: bhmao@china.com
Copyright © 2009, China Association for Science and Technology. Electronic version published by Elsevier Limited. All rights reserved.
DOI: 10.1016/S1570-6672(08)60094-X

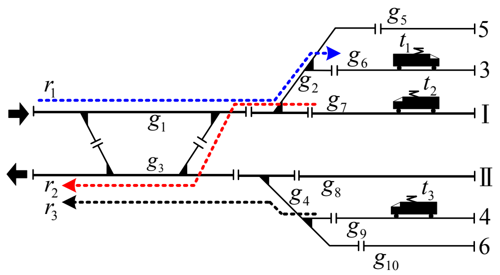

Fig. 1 A simple station layout and three trains

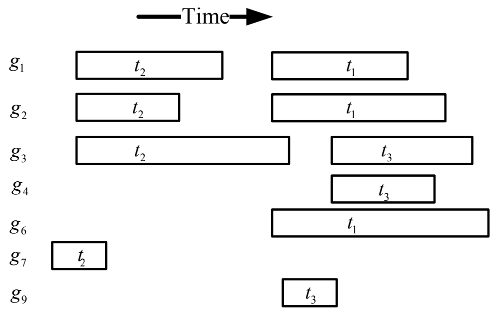

Fig. 2 Time windows for the activities of three trains

$$TAS = \bigcup_{t\in T} \left\{ \left\langle t, a_t, d_t, i_t, o_t \right\rangle \right\} \tag{1}$$

Although a feasible TAS ensures that there are no conflicts between any two jobs, the impact of a single job upon the TAS is neglected. In order to analyze the jobs in a TAS, its dynamic behaviors should be modeled first.

2.1 Concurrence and resources conflicts

The job $\left\langle t, a_t, d_t, i_t, o_t \right\rangle$ is composed of the arrival and departure events j_t^I and j_t^O for the train t, and the set of arrival and departure events of all the trains is denoted by J^I and J^o, respectively. A route is built up by linked track circuit sections (TC), and a job is composed of successive activities if occupancy to a TC for a train is considered an activity and the TCs in a route are released one by one. The set of TCs is G and $G(r)$ is the queue of the TCs in the route r with $r \in R = R^I \cup R^O$ and $G(r) \subset G$. The train for the job j is denoted by $t_j \in T$ for $\forall j \in J = J^I \cup J^O$ and $r_j \in R$ is the route for the job j, and $a(j, g)$ is the activity for the job j at the TC $g \in G(r_j)$ and it can be denoted by the following formula.

$$a(j,g) = \left\langle t_j, r_j, g, g^-_{t_j,r_j}, g^+_{t_j,r_j} \right\rangle \tag{2}$$

The start and end times for the activity $a(j, g)$ are $g^-_{t_j,r_j}$ and $g^+_{t_j,r_j}$ respectively, and it means that the TC g is occupied by activity j during the time window $\left[g^-_{t_j,r_j}, g^+_{t_j,r_j} \right]$. The time windows for the activities of the jobs in a TAS can be obtained using the simulator in the literature[6].

The occurrences of some activities are parallel if their time windows overlap. Fig. 1 shows a simple station layout with six tracks, eight switches and ten TCs. Supposed that there is a TAS including three jobs $\left\{ j^I_{t_1}, j^O_{t_2}, j^O_{t_3} \right\}$ and their routes are $i_{t_1} = \left\langle g_1, g_2, g_6 \right\rangle$, $o_{t_2} = \left\langle g_2, g_1, g_3 \right\rangle$ and $o_{t_3} = \left\langle g_4, g_3 \right\rangle$. The activities of the jobs are listed in Fig. 2 with rectangles which represent the TCs and time windows in the vertical and horizontal directions respectively. There are ten activities in Fig. 2 and the ones $a\left(j^I_{t_1}, g_1\right)$ and $a\left(j^O_{t_3}, g_3\right)$ are parallel.

There is potential conflict between two jobs if they would visit the same TC. In other words, the following one might be delayed if the disturbance to the former one occurs and this situation is referred to as resource conflicts.

2.2 Buffer times

There should be a certain interval between the time windows if there are resource conflicts for two jobs, and the time interval is the buffer time which means that a delay would be propagated to the following job if the disturbance to the former one is larger than the buffer time. The two conditions shown in formulas (3)–(5) should be fulfilled for the existence of buffer time: the first is resource conflicts and the other expresses that the two jobs should occur successively at a TC.

$$G\left(r_j\right) \cap G\left(r_k\right) \neq \varnothing, \forall k \in J, k \neq j \tag{3}$$

$$g^-_{t_k,r_k} > g^+_{t_j,r_j}, \forall k \in J, k \neq j, g \in G\left(r_j\right) \cap G\left(\left(r_k\right)\right) \tag{4}$$

$$g^+_{t_k,r_k} < g^-_{t_m,r_m}, \forall m \in J, m \neq k \tag{5}$$

Take jobs j and k for example: the resource conflicts is shown in formula (3), and formulas (4) and (5) reflect that activity $a(j, g)$ is followed by $a(k, g)$. Thus, the buffer time between jobs j and k at TC g is $g^-_{t_k,r_k} - g^+_{t_j,r_j}$. Therefore, the buffer time between j and k is the minimum of those at all the TCs and it is denoted by $b_{j,k}$:

$$b_{j,k} = \min\left\{ g^-_{t_k,r_k} - g^+_{t_j,r_j} \right\}, \forall g \in G\left(r_j\right) \cap G\left(r_k\right) \tag{6}$$

In Fig. 2, the set of conflict resources for jobs $j^I_{t_1}$ and $j^O_{t_2}$ is $\{g_1, g_2\}$ and the buffer time between the two jobs is located between the two activities at g_1, and the buffer time for jobs $j^O_{t_2}$ and $j^O_{t_3}$ is at g_3.

2.3 Bottlenecks detection

Delay propagation will occur if the disturbance is larger than the minimum buffer time in a TAS. When a job is disturbed, the delay propagation area is however determined not only by the disturbance and buffer times but by its neighbors. Thus, the bottlenecks are the jobs which cause the most serious delay propagation when each job is similarly disturbed. When a job j suffers the disturbance d, the train t is delayed for the time $l_t(j, d)$ and the sum of delays of all the trains is denoted by

$$Delays(j,d) = \sum_{t\in T} l_t(j,d)$$

Therefore, the job j satisfying formula (7) would be a bottleneck.

$$j : \max_{j\in J} Delays(j,d) \tag{7}$$

It means that the delay caused by disturbance (j, d) is more serious than that by (j', d) with $\forall j' \in J \wedge j' \neq j$. Thus, Perturbation analysis is a means of bottlenecks detection, so it is necessary to establish the dynamic model of a TAS.

Download English Version:

https://daneshyari.com/en/article/108632

Download Persian Version:

https://daneshyari.com/article/108632

Daneshyari.com