Automatic Construction of Train Arrival and 
Departure Schedules at Terminal Stations

Ken WAKISAKA** and Shigeru MASUYAMA***

** Department of Electronic and Information Engineering, Toyohashi University of Technology
1–1 Hibarigaoka Tenpaku-cho Toyohashi, Aichi, 441-8580, Japan
E-mail: wakisaka@ia.cs.tut.ac.jp

*** Department of Computer Science and Engineering, Toyohashi University of Technology
1–1 Hibarigaoka Tenpaku-cho Toyohashi, Aichi, 441-8580, Japan
E-mail: masuyama@tut.jp

Abstract
Recently, attention to the railway transportation has been revived from the viewpoint of
alleviating environmental problems such as reduction of CO2 emission amount. Moreover, commuter train services in urban areas and long distance train services such as Shinkansen are indispensable in Japan.

In this paper, we formulate the train arrival and departure optimization problem at a terminal station as a 0-1 integer programming problem, and we obtained solutions by using a solver for the case of Tokyo Station of Tohoku Shinkansen. Moreover, we transform this problem in polynomial time to the Maximum matching problem in a bipartite graph when some realistic conditions are assumed.

Key words: Scheduling, 0-1 Integer Programming, Railway Transportation System

1. Introduction
Recently, attention to the railway transportation has been revived from the viewpoint of alleviating environmental problems such as the reduction of CO2 emission amount. Moreover, commuter train services in urban areas and long distance train services such as Shinkansen are indispensable in Japan.

The opening of Kanazawa Station of Hokuriku Shinkansen, and the connection to Shin-Aomori Station of Tohoku Shinkansen, will or has result in further increase in demand for Shinkansen transportation. For example, trains of Tohoku Shinkansen in the vicinity of Tokyo Station are operated with high density, i.e. 14 trains per an hour. Therefore, the number of trains that can be operated at this terminal station will reach its limit in the near future. However, improvement of equipments of terminal stations to cope with this situation is expensive and needs much time. Therefore, we try to increase the number of trains by improving the train operation schedule at terminal stations.

Note that, this problem is not limited to the rapid-transit railway. Congestion of commuter train services in urban areas is a serious problem. This problem is especially remarkable in subways. However, construction of new route and/or extension of the station to solve the problem requires a large construction cost.

Therefore, it is desired to increase the number of trains operated by efficiently using existing facilities. For the above-mentioned reasons we studied a maximization problem of the number of trains that departs and arrives per an hour at a terminal station. (2)

For measuring the efficiency of train operations at a station “Track capacity” is often used. (7) However, various technologies enables trains to be operated more than this capacity safely in Japan.

The scheduling problems of the railway are surveyed in detail in (1). A number of problems in this field are complex and very large in scale, and requires heavy calculations to solve
them.

Carey(3) studied the allocation of trains at a busy station to a platform. However, they adopted satisfaction rate to the demand of each company as a criterion considering circumstances of railways in Europe, where the government own the railway track. Billionnet(4) studied allocation of the platform to trains that arrive at the station. However, this research assumes a situation from which the arrival time of a train is given, and their purpose is different from our research. Chakroborty(5) studied re-assignment of platforms to trains where a large delay is frequently caused. The objective functions for these researches are different from our research.

2. The Reason Why We Pay Attention to Terminal Stations

In this paper, a terminal station is defined as follows: The railway track connected from the station to the outside is composed of the arrival track and the departure track, and the train can arrive from and depart for only one direction.

A terminal station becomes a bottleneck in many cases as much work (e.g. the cleaning of the car, the reversal of seats, etc.) is required before its departure when a train arrived at a terminal station. This is especially the case for Shinkansen and express trains. Consequently, longer stoppage time is needed at terminal stations compared with midway stations. Moreover, no interaction among courses of trains is allowed for avoiding collision among trains. Thereby, in many cases, it is necessary to move a train exclusively. The above-mentioned exclusivity and required stoppage time makes the terminal station a bottleneck.

In this paper, we study an efficient (i.e. that can increase the number of trains on service e.g., scheduling of trains per an hour) while considering various problems concerning to a terminal station.

Recently, it has become possible to shorten the operation interval of trains due to the rapid advancement of various technologies; e.g. acceleration of trains, improvement of traffic signal control systems, etc. However, problems caused by the interaction among trains at a terminal station is still remained to be solved. Therefore, the scheduling at a terminal station is an important factor in increasing the number of trains, e.g., per an hour.

3. The Train Arrival and Departure Optimization Problem at a Terminal Station

In this paper, a terminal station where the traffic volume of the train is stringent is assumed. The objective function to be maximized is the number of trains at a station that trains arrive at and depart from in a unit time (e.g. an hour). Discrete time is assumed in this formulation.

In this paper, we impose the following assumptions on the train and the station:

- The station is connected to exactly one arrival track and one departure track, respectively, with outside of the station.
- Two or more trains never arrive at the station simultaneously. Similarly, two or more trains never depart from the terminal station simultaneously. Therefore, at most two trains can move simultaneously.

Fig. 1 Station track map of Tokyo Station at Tohoku-Shinkansen line
• The train type (e.g. express, rapid, local etc.) is not considered, because no passing train exists at a terminal station.
• At most one train can exist simultaneously in each platform.
• An arrival train comes from the arrival-track.
• A departure train goes to the departure-track.
• An arrival train moves from the arrival-track to the platform in exactly one unit-time.
• A departure train moves from the platform to the departure-track in exactly one unit-time.
• The train can arrive from the arrival-track to any platform, and it can depart from any platform to the departure-track.
• The arriving train and the departure train can move if there is no interaction among their courses. Only one train can move when an interaction of courses among trains happens.
• A train cannot depart before the fixed time has passed since it arrived at the platform.

4. Formulation as a 0-1 Integer Programming problem

We formulate the train arrival and departure optimization problem at a terminal station as a 0-1 integer programming problem.

4.1. Formulation

Notations
• \( L \): The set of platforms.
• \( t_{\text{max}} \): Length of time for considering for the scheduling (Note that discrete time is assumed)
• \( T = \{0, 1, \cdots, t_{\text{max}} - 1\} \): The set of units of time.
• \( s \): Stoppage time duration at the terminal station.
• \( c_{a,d} \); a, d \( \in L \). The upper bound of events (the event of the train arrival to platform \( a \) and the event of the train departure from platform \( d(a \neq d) \)) that can be done simultaneously. If they are able to execute simultaneously, it has value 2, otherwise, value 1.

<table>
<thead>
<tr>
<th>( c_{a,d} )</th>
<th>#20</th>
<th>#21</th>
<th>#22</th>
<th>#23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track #20</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Track #21</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Track #22</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Track #23</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Decision variable \( x_{t,l}^a \) is assumed to be 1 when the train arrives at platform \( l \) at a certain time \( t \), otherwise 0. Similarly, decision variable \( x_{t,l}^d \) is assumed to be 1 when the train departs from platform \( l \) at a certain time \( t \), otherwise 0.

The objective function and constraints are expressed as follows. We assumed that trains do not exist at each platform as an initial state at time 0.

Maximize \( \sum_{t \in T, l \in L} x_{t,l}^a \)  \( \text{(1)} \)

subject to

\[ \sum_{l \in L} x_{t,l}^a \leq 1, \; t \in T \] \( \text{(2)} \)

\[ \sum_{l \in L} x_{t,l}^d \leq 1, \; t \in T \] \( \text{(3)} \)

\[ \sum_{j=0}^{j} (x_{t,j}^a - x_{t,j}^d) \leq 1, \; j \in T, \; l \in L \] \( \text{(4)} \)
\[ x_{tj}^l = 0, \ 0 \leq t < s, \ l \in L \]  
(5)

\[ \sum_{t=0}^{j-s} x_{tj}^l - \sum_{t=0}^{j} x_{tj}^l \geq 0, \ s \leq j < t_{\text{max}}, \ l \in L \]  
(6)

\[ x_{tj}^l + x_{tj}^{l_2} \leq C_{l,l_2}, \ t \in T, \ l, l_2 \in L \]  
(7)

\[ x_{tj}^l \in \{0, 1\}, \ t \in T, \ l \in L \]  
(8)

\[ x_{tj}^l \in \{0, 1\}, \ t \in T, \ l \in L \]  
(9)

Expression (1) is the objective function that maximizes the number of operated trains. When considering a time zone long enough, the number of arrival and the number of starts becomes equal. Expression (2) means that two or more trains cannot arrive from an arrival-track to a platform simultaneously. Expression (3) means that two or more trains cannot depart from a platform to a departure-track simultaneously. Expression (4) means that two or more trains cannot exist at each platform. We assume that no train exists at platform \( l \) at the initial state. Therefore, if no train exists at platform \( l \), the number of arrival trains and that of departure trains at platform \( l \) from time 0 to time \( j \) must be equal. If the train stops at platform \( l \), the number of arrival trains is larger by one than that of departure trains at platform \( l \) from time 0 to time \( j \). Expressions (5) and (6) are constraints that ensure that the stoppage time of the train is at least \( s \) units of time. The next section 4.2 describes these expressions. Expression (7) is a constraint that train that might collide cannot move simultaneously. Expressions (8) and (9) are constraints that variables must be either 0 or 1.

### 4.2. Explanation of Expressions (5) and (6) that ensure stoppage time \( s \)

Expression (5) is the constraint that ensures no train departs at time \( 0 \leq t < s \) from platform \( l \). We assume no train exists at the initial state. Furthermore, we have to ensure the stoppage time of minimum \( s \) [unit time]. Therefore, even if a train arrives at time 0, it cannot leave by time \( s \).

Expression (6) is the constraint about the departure of a train in any platform \( l \) and any time \( j(s \leq j < t_{\text{max}}) \). The 1st term of the left-hand side of expression (6) means the total number of arrived trains from time 0 to time \( j - s \), and the 2nd term of the left-hand side means the total number of departing trains from time 0 to time \( j \). We consider three cases and explain platform \( l \) at time \( j(s \leq j < t_{\text{max}}) \).

**Case 1: When train \( t \) that ensures the minimum stoppage time constraints has stopped**

Train \( t \) must arrive at platform \( l \) by time \( j - s \) (Fig. 2). The 1st term is the total number of trains departed from platform \( l \) by time \( j \). This term considers train \( t \). Note that train \( t \) has not yet departed from platform \( l \), and train \( t \) is excluded when considering the 2nd term. In this case, the 1st term becomes only 1 large compared to the 2nd term. Note also that if train \( t \) just depart at time \( j \), and the 1st and the 2nd terms are equal. For this reason, expression (6) is always satisfied.

**Case 2: When train \( t \) that has stopped less than the minimum stoppage time \( s \)**

Train \( t \) does not arrive in time \( j - s \) (Fig. 3). Therefore, train \( t \) is not considered in the 1st term. Similarly, train \( t \) is not considered in the 2nd term because train \( t \) can not depart by time \( j \). Therefore, the 1st and the 2nd terms become equal. Expression (6) is satisfied if train \( t \) does not depart from platform \( l \) at time \( j \).

If train \( t \) depart at time \( j \), the value of the 2nd term will increase by one (equivalent to train \( t \)), and expression (6) will not be satisfied.

**Case 3: When no train has stopped at time \( j \)**

Let the train that departed from platform \( l \) at time \( j \) just before be train \( t \) (Fig. 4). Note that the 1st term is the total number of trains departed from platform \( l \) by time \( j \). It contains train \( t \). Note also that the 2nd term is the total number of trains arrived from platform \( l \) by time \( j - s \) that considers train \( t \), because train \( t \) must already arrive at time \( j - s \). (Train \( t \) stopped at
Fig. 2 When train $t$ that ensures the minimum stoppage time constraints has stopped

Fig. 3 When train $t$ that has stopped less than the minimum stoppage time $s$ has not arrived by time $j - s$.

Fig. 4 When no train has stopped at time $j$

As mentioned above, when trains respect constraints of stoppage time, the value of the left-hand side of expression (6) must be 0 or 1.

If the value of the left-hand side of expression (6) becomes two or more, the following two cases are considered:

Case A: When the train has stopped at platform $l$, another train arrives;  
Case B: The train that does not satisfy stoppage time constraints will depart from time $j - s$ to time $j$ from platform $l$.

In Case A, expression (4) is not satisfied at some time from time $j - s$ to time $j$. In Case B, expression (6) is not satisfied at some time from time $j - s$ to time $j$.

4.3. Computational Experiments

We solved the arrival and departure optimization problem for the terminal station at Tokyo Station of Tohoku Shinkansen as an example by using Solver GLPK4.39. Discrete time is assumed, where the unit time is defined as two minutes. This value is used based on actual values of train intervals and travel time from/to the platform of Tohoku Shinkansen at Tokyo Station. The minimum stoppage time is defined as 12 minutes, which is an actual value. We calculate the schedule for 30 units of time (During one hour in actual time).
Fig. 5 shows the obtained train arrival and departure pattern. We obtain an answer where 18 trains arrive in 30 units of time. The same arrival and departure pattern is repeated every 7 units of time. All trains arrive at and depart from in the shortest stoppage time, and each platform is used without idle time, and the best movement pattern is achieved.

When this pattern is repeated, four trains for every 14 minutes are able to be operated, specifically, 17.1 trains per an hour can be operated. 14 trains per an hour are operated now at Tokyo station of Tohoku Shinkansen, thus three more trains (about 4,800 people are possible to be on board) are able to be operated by the solution.

5. Polynomial Time Transformation from The Train Arrival and Departure Optimization Problem (at a Terminal Station) to Maximum Matching in the Bipartite Graph

An 0-1 integer programming problem is NP-hard, and believed to be a difficult problem. In this section, we transform from The Train Arrival and Departure Optimization Problem (at a terminal station) to the bipartite graph maximum matching problem, which can be solved in polynomial time. To do this, we introduce "Cycle", and add two conditions. "cycle" is defined as the train operation schedule on which a pair of an arrival event and a departure event are performed for a trains only one time, respectively, on each platform.

When the following two conditions are satisfied, the train arrival and departure optimization problem at a terminal station can be transformed to the maximum matching problem in the bipartite graph.

\textbf{condition 1} The constraint concerning the stoppage time is relaxed. The train can leave at the next time of its arrival at the platform.

\textbf{condition 2} During one cycle a train arrives at and departs from each platform exactly once.

In this section, we relax the constraint of the stoppage time by assuming these conditions. However, if the stoppage time is not secured, trains cannot be safely operated. This problem can be solved by appropriately inserting the gap time after the algorithm described later is applied.

The platform is restrictively used by some types of trains (express, limited express, local, etc.) at some station. However, all platforms are often evenly used in the station where it is used by a large number of trains.

A node at one side in the bipartite graph corresponds to a departure event of the train from each platform while a node at the another side corresponds to an arrival event. Then the combination of a node corresponding to an arrival event and a node corresponding to a departure event that are able to execute simultaneously is connected by an arc. The maximum matching is obtained on the bipartite graph. The arrival and departure events where corresponding nodes are mutually connected by an edge is done simultaneously, and events not corresponding to connected nodes cannot be done simultaneously, thus, we can obtain the train arrival and departure pattern in one cycle.

Thus, we have the following algorithm that works in polynomial time:

\textbf{Train operation pattern generation algorithm}

\textbf{Step 1} Make the "arrival and departure competition table". The arrival events to each platform
corresponds to each low of the table. The departure events from each platform correspond to each column of the table. If the pair of an arrival event and a departure event are able to execute simultaneously, it has entry "P"(Possible), otherwise, entry "I"(Impossible).

**Step 2** Generate bipartite graph \( G = (V_a, V_d, A) \) as follows:
- \( V_a \) corresponds to the low of the table.
- \( V_d \) corresponds to the column of the table.
- Connect node \( v_a \in V_a \) and \( v_d \in V_d \) when value of \( (v_a, v_d) \) is "P".

**Step 3** Solve the bipartite graph maximum matching problem on \( G = (V_a, V_d, A) \).

**Step 4** According to the following rules, we determine whether we perform an arrival and departure events simultaneously or perform an event independently.
- Connected events are performed simultaneously.
- Isolated events are performed independently.

**Step 5** Sort events so that the existence time of the train at platform may become the maximum. (The next train should arrive at each platform immediately after the departure of a train.)

For the station of a layout similar to that illustrated in Fig. 6, the "Arrival and departure competition table" is shown in Table 1. In this table, "P"(possible) denotes the combination of an arrival event and a departure event that can be executed simultaneously and "I"(Impossible) denotes combination of the arrival and departure events that cannot be executed simultaneously.

<table>
<thead>
<tr>
<th>arrive \ depart</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track #1</td>
<td>I</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Track #2</td>
<td>I</td>
<td>I</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Track #3</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>P</td>
</tr>
<tr>
<td>Track #4</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
</tbody>
</table>

![Fig. 6 An example of the track layout of a terminal station](image-url)

For instance, the arrival at track #2 and the departure from track #3 are possible to be executed simultaneously. (fig. 7)

![Fig. 7 Situation where train can move simultaneously (arrival at Track #2 and departure from Track #3)](image-url)

However, the arrival at track #3 and the departure from track #2 are not possible be to executed simultaneously. Because the course of trains collide where the railway tracks intersects. (fig. 8)
The bipartite graph $G = (V_a, V_d, A)$ generated from Table 2 is shown in Fig. 9. The edge shown by a heavy line is one selected by the maximum matching for this bipartite graph.

Next, the arrival and departure events performed simultaneously from the obtained matching is determined. Each fragment of Fig. 10 (a)-(e) was generated based on the bipartite graph maximum matching created at Step 3. Moreover, these fragments are equivalent to 1 unit time of a Gantt chart generated afterwards. The events connected with a matching are assigned on the same fragment. (e.g. the pair of "Track 1: Arrival" and "Track 2: Departure" corresponds to (a) of Fig. 10). Independent events (not connected with a branch) are assigned independently. (e.g. "Track 4: Arrival" corresponds with (e) of Fig. 10)

Next, these fragments are put in the order so that the stoppage time of each train may become as long as possible. Fragments are put so that the following train may arrive at the same platform, immediately after a train leaves from each platform. (e.g. fragment (b) is put immediately after fragment (a). Fragment (a) is put immediately after Fragment (d). Thus, the Gantt chart corresponding to one cycle is created.

Consequently, the pattern of one cycle illustrated in Fig. 11 can be obtained. This one
cycle can be repeatedly applied. A very tight schedule of trains, where the interval from the
departure of a train to arrival of the next train at a platform is as short as possible, can be
obtained.

![Table showing train tracks and operations](image)

**Fig. 11** an example of the timetable of generated “One cycle”

The maximum matching in the bipartite graph can be obtained in $O(mn)$ time where $m$
is the number of nodes, $n$ is the number of edges, and other steps can be done in polynomial
time. Therefore, it is possible to make a train operation pattern in polynomial time, when
conditions 1 and 2 are satisfied.

However, it is difficult to operate trains safely if the stoppage time does not have sufficient
length. Therefore, the constraints of the stoppage time cannot be actually disregarded.

If the length of stoppage time is insufficient, the pair of an arrival event and a departure
event that can be done simultaneously are executed not simultaneously. When all arrival or
departure events are never simultaneously performed with other events, the gap where any
train does not arrive and depart is inserted. Then, the stoppage time of all trains in one cycle
increases by one unit time.

We experimented on many existing terminal station in Japan (including Tokyo Station at
Tohoku-Shinkansen line). The number of trains that can be operated is greatly influenced by
the size of the station (e.g., the number of platforms) and the stoppage time.

Thus, the discussion only about the number of the trains is insufficient. Therefore, we
verified "Whether the stoppage time of the train was minimum or not?" and "Whether extra
time exists or not?". In any case we were able to obtain a very tight schedule.

However, we assume that the stoppage time of all trains is the same that in this paper.
The stoppage time of the train that becomes a forwarding train after it arrives at the station
might be short. Moreover, there is a possibility that an efficient schedule can be obtained by
choosing the strategy that allows some platforms not to be used. It is desirable to analyze the
computational complexity by considering these factors.

In this paper, we only consider double-tracked terminal stations. Thereby, the arrival-
and-departure competition table has been expressed by two dimensions (e.g. from fig.6 to
Table 2), and the problem was able to be expressed and solved on the bipartite graph maximum
matching problem. When considering the station of a four-track line, or a station on the way,
arrival-and-departure competition cannot be expressed by two dimensions.

It will become multi dimensions and two or more competition tables are needed. If this
is the case, it is difficult to express by a bipartite graph and some approach different from that
proposed in this paper is probably needed.

6. Conclusion

In this paper, we formulated the train arrival and departure optimization problem at ter-
mental stations as the 0-1 integer programming problem, and we succeeded in solving it by
using a solver for the case of Tokyo Station of Tohoku Shinkansen. Moreover, we showed
polynomial time transformation from this problem to the maximum matching problem in a
bipartite graph when some realistic conditions were assumed.
Acknowledgement

This work was supported in part by Global COE Program “Frontiers of Intelligent Sensing” from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References