Simulation Modelling Practice and Theory 20 (2012) 124–141

Contents lists available at SciVerse ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

# A feasible timetable generator simulation modelling framework for train scheduling problem

Özgür Yalçınkaya *, G. Mirac Bayhan

*Dokuz Eylül University, Department of Industrial Engineering, 35160 Buca-İzmir, Turkey*

ARTICLE INFO

*Article history:*
Received 12 November 2010
Received in revised form 20 August 2011
Accepted 24 September 2011
Available online 22 October 2011

*Keywords:*
Train
Scheduling
Timetabling
Simulation
Deadlock

ABSTRACT

An important problem in management of railway systems is the train scheduling/timetabling problem. This is the problem of determining a timetable for a set of trains that do not violate track capacities and satisfy some operational constraints. In this study, a feasible timetable generator framework for stochastic simulation modelling is developed. The objective is to obtain a feasible train timetable for all trains in the system. The feasible train timetable includes train arrival and departure times at all visited stations and calculated average train travel time. Although this study focuses on train scheduling/timetabling problem, the developed simulation framework can also be used for train rescheduling/dispatching problem if this framework can be fed by real time data. The developed simulation model includes stochastic events, and can easily cope with the disturbances that occur in the railway system.

© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction

Management of railway systems is increasingly becoming an important issue of transport systems. One of the important problems in management of railway systems is the train scheduling/timetabling problem (*TrnSchPrb*). This is the problem of determining a timetable for a set of trains that do not violate track capacities and satisfy some operational constraints. Several variations of the problem can be considered, mainly depending on the objective function to be optimised, decision variables, constraints and complexity of the relevant railway network.

A general most common *TrnSchPrb* in the literature considers a single track linking two major stations with a number of intermediate stations in between [5]. It is assumed that $S = \{1,\ldots,s\}$ represents the set of stations, numbered according to the order in which they appear along the rail line. In particular, 1 and $s$ denote the initial and final stations, respectively. Analogously, it is assumed that $T = \{1,\ldots,t\}$ denotes the set of trains which are candidate to be run in a given time horizon. For each train $j \in T$, a starting station $f_j$ and an ending station $l_j$ ($l_j > f_j$) are given. Let $S^j = \{f_j,\ldots,l_j\} \subseteq S$ be the ordered set of stations visited by train $j$. A timetable defines, for each train $j \in T$, the arrival and departure times for the stations $f_j, f_j + 1,\ldots,l_j - 1, l_j$. The running time of train $j$ in the timetable is the time elapsed between origin station and destination station of the train [5]. This general *TrnSchPrb* can be more sophisticated by adding some real life behaviour of rail systems or relaxing some assumptions made related with the railway system under consideration.

The *TrnSchPrb* has been studied by researchers and so far many efforts have been spent to solve the problem. In early years, due to the limitations of computers' abilities and the complexity of the problem, the problem was relaxed by unrealistic assumptions and generally deterministic models were studied. Depending on the increasing computer capabilities more realistic models were developed. Although simulation for modelling has been used in some articles, none of them includes a

* Corresponding author. Tel.: +90 2324127627; fax: +90 2324127608.
*E-mail addresses:* ozgur.yalcinkaya@deu.edu.tr (Ö. Yalçınkaya), mirac.bayhan@deu.edu.tr (G. Mirac Bayhan).

1569-190X/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2011.09.005

comprehensive framework. This has been the main motivation for us to develop a feasible timetable generator simulation modelling framework.

In this study, a feasible timetable generator framework for stochastic simulation modelling is developed for obtaining a feasible train timetable for all trains in a railway system. This framework includes train arrival and departure times for all stations visited by each train and calculated average train travel time. A general stochastic simulation modelling framework is developed and depicted step by step in order to guide to researchers who aim to develop a simulation model of railway transportation systems. By using this framework all the railway transportation systems can be modelled with only problem/infrastructure specific modifications and feasible solutions are easily obtained. In order to avoid a deadlock, a general *blockage preventive algorithm* is also developed and embedded into the simulation model.

In Section 2 a literature review on the *TrnSchPrb* is given. A feasible timetable generator simulation modelling framework is demonstrated in detail in Section 3. A hypothetic problem and the proposed simulation model are also introduced in this section. In Section 4 the obtained results are discussed. Concluding remarks and future work directions are exhibited in the last section.

## 2. Literature review on train scheduling/timetabling problem

The existing studies about the *TrnSchPrb* aim at achieving a train timetable with arrival and departure times of all trains at the visited stations in the system. These studies generally begin with a planned *infeasible* initial (draft) timetable with many conflicts. After these conflicts were solved a *feasible* train timetable is composed, and the train operating authority runs the trains according to the timetable. In review papers; Assad [1], Cordeau et al. [9], Newman et al. [19] and Caprara et al. [6] some railway optimisation problems are considered, the *TrnSchPrb* is regarded in only one section, and none of these review papers concentrates solely on the *TrnSchPrb*. In papers that focus on the *TrnSchPrb* mathematical models are; Frank [10], Szpigel [23], Mees [18], Jovanovic and Harker [13] and Odijk [20]. On the other hand Higgins et al. [12], Brännlund et al. [2], Tormos et al. [24] and Liu and Kozan [17] are used heuristics and meta-heuristics as solution approaches.

Two study series draw our attention. The first serial includes four articles; Caprara et al. [4,5,7] and Cacchiani et al. [3]. The second one consists of Zhou and Zhong [27] and Castillo et al. [8]. Caprara et al. [4,5] concentrate on train timetabling problem relevant to a single, one way track linking two major stations with a number of intermediate stations between them. A graph theoretic formulation is proposed for the problem using a directed multigraph in which nodes correspond to departures or arrivals at a certain station at a given time instant. The objective is to maximise sum of the profits of the scheduled trains. Caprara et al. [7] extend the problem considered by Caprara et al. [5], by taking into account additional real world constraints. On the other hand, Cacchiani et al. [3] propose heuristic and exact algorithms for the periodic and non-periodic train timetabling problem on a corridor to maximise the sum of the profits of the scheduled trains. The heuristic and the exact algorithms are based on the solution of the relaxation of an integer linear programming formulation in which each variable corresponds to a full timetable for a train. This approach is in contrast with previous approaches proposed by Caprara et al. [4,5,7] so that these authors considered the same problem and used integer linear programming formulations in which each variable was associated with a departure and/or arrival of a train at a specific station in a specific time instant. Zhou and Zhong [27] focus on single track and propose a generalised resource constrained project scheduling formulation for train timetabling problem. The developed algorithm chronologically adds precedence relation constraints between conflicting trains to eliminate conflicts, and the resulting sub-problems are solved by the longest path algorithm to determine the earliest start times for each train in different segments. Castillo et al. [8] use an optimisation method to solve train timetabling problem for a single tracked bidirectional line, similar to the one presented by Zhou and Zhong [27] but more complex, and discuss the problem of sensitivity analysis. A three stage method is proposed to deal with the problem and a sequential combination of objective functions is used for solution.

In recent years, the authors, Zhou and Zhong [26], Liebchen [16], and Lee and Chen [15] have spent many efforts to optimise multi objective train scheduling problems.

In only a few papers a simulation model was developed for the *TrnSchPrb*. To our knowledge, Wong and Rosser [25] are the first authors who developed a simulation model for train scheduling problem. The output of the simulation model comprises a pictorial representation of the pattern of train movements as well as detailed statistics for each train. The problem is to determine where a crossing or overtaking should be allowed to occur, and the objective is to minimise the sum of weighted costs of delaying trains at passing loops where the weights chosen reflect the importance of each type of train. To improve the system performance, train starting times are varied, and one train at a time heuristic iterative procedure is used for improvements. Petersen and Taylor [22] presented a state space description for the problem of moving trains over a line, and an algebraic description of the relationships that must hold for feasibility and safety considerations was given. The line blockage problem at high traffic intensities was discussed under conditions that ensure the blockage not to occur. The objective of the study is to minimise the terminating times of the trains. Geske [11] focused on the railway scheduling problem and developed a constraint based deterministic simulation model with the objective of reducing the lateness of trains. Selecting alternative paths in stations was an optimisation task to reduce lateness and to find a conflict free solution. The results of the proposed sequentially train scheduling heuristic was compared with those of a genetic algorithm.

Above, a brief review of existing research related to the *TrnSchPrb* has been presented. While analytical results were obtained by exact algorithms, simulation models and meta-heuristics with approximate outcomes were also employed. The meta-heuristics were employed by the researchers in the relevant area after 1990s, multi objectives were optimised after 2000s. On the other hand, in a few papers simulation models were constructed for the scheduling problem. Simulation models are flexible and solve real problems without making too many restricting assumptions as in most analytical models. Although simulation for modelling has been used in some articles, none of them includes a comprehensive framework. This has been the main motivation for us to develop a feasible timetable generator simulation modelling framework.

In the current study, a general stochastic simulation modelling framework is developed and depicted step by step in order to guide to researchers who aim to develop a simulation model of railway transportation systems. By using this framework all the railway transportation systems can be modelled with only problem/infrastructure specific modifications and feasible solutions are easily obtained.

## 3. A feasible timetable generator simulation modelling framework

In this section, a feasible timetable generator simulation modelling framework for the *TrnSchPrb* is given. The objective is to obtain a feasible train timetable for all trains in the system. The feasible train timetable includes train arrival and departure times at all visited stations and additionally calculated average train travel time. This section involves three subsections. In the first subsection, a hypothetic *TrnSchPrb* is introduced. In the next subsection, the simulation modelling framework is developed and applied on the hypothetic *TrnSchPrb*. In the last subsection, the results are discussed.

### 3.1. A hypothetic train scheduling problem

The proposed simulation modelling framework is implemented on a hypothetic *TrnSchPrb* that is similar with the common problem studied in the literature. The infrastructure in the problem has a line structure inspired by a real railway line system, and has a planned initial timetable with arrival and departure times of trains only at two end stations of the infrastructure.

#### 3.1.1. Railway line description

The railway line, which is inspired by a real line, is a single track corridor as analogous to many studied lines in the literature and in real railway systems. The line–station diagram of the single track corridor and the infrastructure of stations are shown in Fig. 1. There are 10 real stations on the single track corridor that are labelled as $S_i$ ($i = 1, 2, \ldots, 10$) from the east to the west. The single track corridor has two terminuses, $TS_1$ and $TS_{10}$, which indicate the beginning and the finishing points of the single track corridor.

As it is seen in Table 1, the total track length from the $TS_1$ to the $TS_{10}$ is 286,270 m. Since all the real stations have 200 m platform, the whole length of the corridor is 288,270 m.

#### 3.1.2. Planned initial train timetable

The train arrival and departure times at two end stations are given in Table 2, where $WB_i$ ($i = 1, 2, \ldots, 10$) indicates a *westbound* train that begins its trip from the first real station on the *east* of the corridor and plans to finish at the first real station on the *west* of the corridor $EB_i$ ($i = 1, 2, \ldots, 10$) indicates an *eastbound* train which has an opposite direction to WB trains.

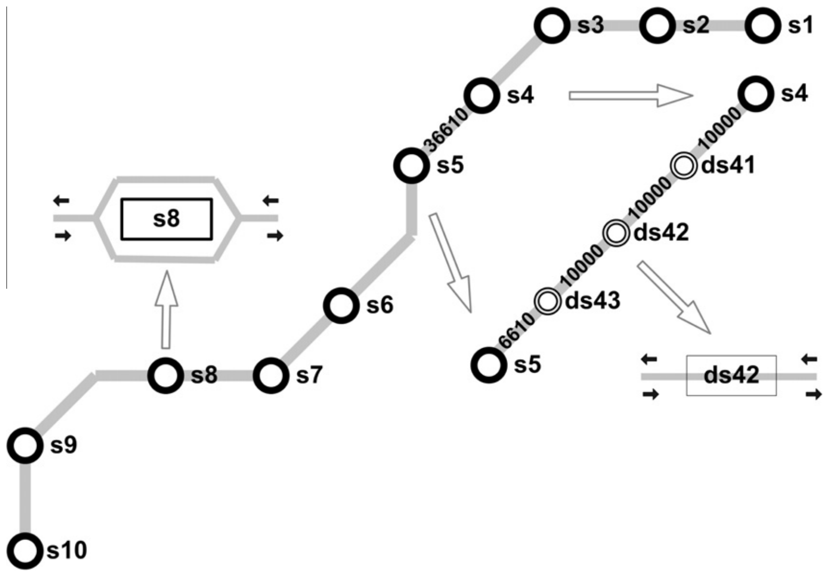


**Fig. 1.** Line–station diagram of the single track corridor.

**Table 1**
Track lengths between the real stations.

| From \ To | $TS_1$ (*East*) | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ | $S_{10}$ | $TS_{10}$ (*West*) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $TS_1$ (*East*) | 0 | 500 | 28,070 | 60,170 | 88,400 | 125,210 | 170,060 | 197,060 | 214,460 | 243,560 | 285,770 | 286,270 |
| $S_1$ | 500 | 0 | 27,570 | 59,670 | 87,900 | 124,710 | 169,560 | 196,560 | 213,960 | 243,060 | 285,270 | 285,770 |
| $S_2$ | 28,070 | 27,570 | 0 | 32,100 | 60,330 | 97,140 | 141,990 | 168,990 | 186,390 | 215,490 | 257,700 | 258,200 |
| $S_3$ | 60,170 | 59,670 | 32,100 | 0 | 28,230 | 65,040 | 109,890 | 136,890 | 154,290 | 183,390 | 225,600 | 226,100 |
| $S_4$ | 88,400 | 87,900 | 60,330 | 28,230 | 0 | 36,810 | 81,660 | 108,660 | 126,060 | 155,160 | 197,370 | 197,870 |
| $S_5$ | 125,210 | 124,710 | 97,140 | 65,040 | 36,810 | 0 | 44,850 | 71,850 | 89,250 | 118,350 | 160,560 | 161,060 |
| $S_6$ | 170,060 | 169,560 | 141,990 | 109,890 | 81,660 | 44,850 | 0 | 27,000 | 44,400 | 73,500 | 115,710 | 116,210 |
| $S_7$ | 197,060 | 196,560 | 168,990 | 136,890 | 108,660 | 71,850 | 27,000 | 0 | 17,400 | 46,500 | 88,710 | 89,210 |
| $S_8$ | 214,460 | 213,960 | 186,390 | 154,290 | 126,060 | 89,250 | 44,400 | 17,400 | 0 | 29,100 | 71,310 | 71,810 |
| $S_9$ | 243,560 | 243,060 | 215,490 | 183,390 | 155,160 | 118,350 | 73,500 | 46,500 | 29,100 | 0 | 42,210 | 42,710 |
| $S_{10}$ | 285,770 | 285,270 | 257,700 | 225,600 | 197,370 | 160,560 | 115,710 | 88,710 | 71,310 | 42,210 | 0 | 500 |
| $TS_{10}$ (*West*) | 286,270 | 285,770 | 258,200 | 226,100 | 197,870 | 161,060 | 116,210 | 89,210 | 71,810 | 42,710 | 500 | 0 |

**Table 2**
Planned initial train timetable.

| Station | Train | Arrival time | Departure time |
|---|---|---|---|
| $S_1$ | $WB_1$ | 00:00 | 00:10 |
| | $WB_2$ | 02:00 | 02:10 |
| | $WB_3$ | 04:00 | 04:10 |
| | $WB_4$ | 06:00 | 06:10 |
| | $WB_5$ | 08:00 | 08:10 |
| | $WB_6$ | 10:00 | 10:10 |
| | $WB_7$ | 12:00 | 12:10 |
| | $WB_8$ | 14:00 | 14:10 |
| | $WB_9$ | 16:00 | 16:10 |
| | $WB_{10}$ | 18:00 | 18:10 |
| $S_{10}$ | $EB_1$ | 00:00 | 00:10 |
| | $EB_2$ | 02:00 | 02:10 |
| | $EB_3$ | 04:00 | 04:10 |
| | $EB_4$ | 06:00 | 06:10 |
| | $EB_5$ | 08:00 | 08:10 |
| | $EB_6$ | 10:00 | 10:10 |
| | $EB_7$ | 12:00 | 12:10 |
| | $EB_8$ | 14:00 | 14:10 |
| | $EB_9$ | 16:00 | 16:10 |
| | $EB_{10}$ | 18:00 | 18:10 |

### *3.2. A feasible timetable generator simulation model*

A feasible timetable generator simulation model is developed by using Arena discrete event simulation software [14] in a modular manner. In Arena, the user builds an experiment model by placing boxes of different shapes that represent processes or logic. Connector lines are used to join these boxes together and specify the flow of entities. While boxes have specific actions relative to entities, flow, and timing, the precise representation of each boxes and entity relative to real-life objects is subject to the modeller. Statistical data can be recorded and outputted as reports.

First, the railway corridor with links, intersections and the stations is modelled, and track failures and repairs are included. Then, train movement logic on the corridor is modelled. We use a rule for track allocation to candidate trains that are the trains waiting at neighbour stations of the track to use it. Fixed train speeds are relaxed, and additional unplanned delays at the stations are inserted. The number of trains in the system is increased and randomness is added to the planned initial train timetable. As a last step, animation of the system is developed. Some assumptions are made during the modelling phase of the simulation model. It must be noted that many of our assumptions are the fundamental assumptions made by the existing studies in the literature. Some of the assumptions made for our simulation model are:

(1) The unit for length and time is metre and second, respectively.
(2) It takes 32 s for trains to reach the real stations $S_1$ and $S_{10}$ from park area, then the trains wait 568 s at these stations, i.e., they spend totally 600 s (10 min) as a dwell time. First trips are planned to begin at 00:10:00 o'clock. But due to additional unplanned delays at the stations lateness may occur.
(3) Time spent for reaching to a terminus ($TS_1$ or $TS_{10}$) from the park area is negligible.

(4) The WB trains' departure station is the $S_1$ and destination is the $S_{10}$, and the EB trains' departure station is the $S_{10}$ and destination is the $S_1$.
(5) There will be 20 trains running in a day, 10 of them are the WB and the other 10 are the EB trains.
(6) All the trains are the same type.
(7) Passengers are ignored at this level of the model.
(8) There is time headway (40 s) between two consecutive trains at a station, which have the same trip direction, in order to have a safe trip.
(9) More than one train that have the same direction can use the same track with distance headway (1000 m) between them.
(10) The train lengths are 50 m.
(11) Earliness and lateness time in the planned initial train timetable, due to some uncontrollable events that occur out side of the corridor, is uniformly distributed between −900 and +900 s.

*3.2.1. Railway corridor modelling*

The railway corridor is a union of intersections and links, and modelled via the *Networks Element* of the Arena. The links are the track parts on which train traverses during its trip from a station to another neighbour station. The links are modelled via the *Links Element* of the Arena. The intersections are connection points of the links. There are three kinds of intersections in the model. The first includes the intersections related to the real stations, and have lengths in metres. The second includes the intersections that are only used for connecting the links and dummy stations that are located on the tracks between the real stations to keep a train wait during the repairing of a track failure, and have no length. The third is related to the park areas where the empty trains can park. The intersections are modelled via the *Intersections Element* of the Arena.

The stations are locations where a train can stop for boarding and alighting events, for parking or for waiting until a failure is accomplished. The real stations, $S_i$ ($i = 1, 2, \ldots, 10$), are interrelated with two intersections. The dummy stations, $dS_{ij}$ ($j = 1$ for $i = 7$; $j = 1, 2$ for $i = 1, 2, 3, 6, 8$; $j = 1, 2, 3$ for $i = 4, 5, 9$), are located on the tracks between the real stations to keep a train wait during the repairing of a failure, if the failure occurs while a train is traversing between the real stations. The stations are modelled via the *Stations Element* of the Arena.

In the simulation model the links, the intersections and the track failures are controlled via variables. The *Variables Element* of the Arena is used for defining the variables.

Assumptions related to the railway corridor part of the simulation model are:

(1) The railway system is a single track line, a corridor.
(2) The traffic on tracks is bidirectional, two way.
(3) All the real stations have 200 m platforms for boarding and alighting events.
(4) There are 10 real stations and 20 dummy stations on the corridor, that is, the corridor is 288,270 m long.
(5) The terminuses ($TS_1$ and $TS_{10}$) have infinite train capacity.
(6) The $TS_1$ is located on the east point, and the terminus $TS_{10}$ is located on the west point of the corridor.
(7) Every middle real station has capacity of two trains, that is, there will be at most two trains at a real station at the same time.
(8) Every dummy station has capacity of one train, that is, there will be at most one train at a dummy station at a specific time.
(9) There are 100 m length park areas near the terminuses. These park areas are used by a train, which finished its trip, while leaving the corridor, or waiting to enter the corridor.
(10) Distance between these park areas and terminuses are 100 m.

*3.2.2. Track failure modelling*

Track failure is an event that prevents a train to occupy the impaired track for a trip. The train can use the track after it is repaired. Distributions for failure times and repair times for the failed tracks are depicted in Table 3. In the first two columns, location and length of the tracks are given. These tracks are ranked according to their lengths. The shortest one has rank 1 and lies between the $S_7$ and the $S_8$, and is selected to be the *base track*. The longest one has rank 9 and lies between the $S_5$ and the $S_6$. The ratios are obtained by dividing the lengths of the tracks to the length of the *base track*.

We divided the long tracks into smaller parts (short tracks) and located the dummy stations between these short tracks. After obtaining failures for tracks according to the distributions shown in Table 3, these failures are transferred to the short tracks with the probabilities exhibited in Table 4. In the simulation model, the track failures are controlled via variables. If a failure occurs in a track part, trains are prevented to use this part until it is repaired. The event diagrams for *track failure event* and *track repair event* are exhibited in Figs. 2 and 3 respectively.

Assumptions related to the track failure part of the simulation model are:

(1) There will be track failures that will stop traffic on the related track. The failure and repair times distributions are given in Table 3.
(2) The failure time of the *base track* is distributed exponentially with a mean of 86,400 s (24 h), that is, it is expected to observe one failure for the *base track* in a day.

**Table 3**
Failure times and repair times distributions.

| Location | Length (m) | Rank | Ratio | 1/ratio | Failure time distribution | Repair time distribution |
|---|---|---|---|---|---|---|
| $S_1$–$S_2$ | 27,370 | 3 | 1.591 | 0.628 | Expo (54,296) | Expo (2864) |
| $S_2$–$S_3$ | 31,900 | 6 | 1.855 | 0.539 | Expo (46,586) | Expo (3338) |
| $S_3$–$S_4$ | 28,030 | 4 | 1.630 | 0.614 | Expo (53,017) | Expo (2933) |
| $S_4$–$S_5$ | 36,610 | 7 | 2.128 | 0.470 | Expo (40,592) | Expo (3831) |
| $S_5$–$S_6$ | 44,650 | 9 | 2.596 | 0.385 | Expo (33,283) | Expo (4673) |
| $S_6$–$S_7$ | 26,800 | 2 | 1.558 | 0.642 | Expo (55,451) | Expo (2805) |
| $S_7$–$S_8$ | 17,200 | 1 | 1.000 | 1.000 | Expo (86,400) | Expo (1800) |
| $S_8$–$S_9$ | 28,900 | 5 | 1.680 | 0.595 | Expo (51,421) | Expo (3024) |
| $S_9$–$S_{10}$ | 42,010 | 8 | 2.442 | 0.409 | Expo (35,374) | Expo (4396) |

**Table 4**
Failure probabilities of the short tracks.

| Location of long track | Length of long track (m) | Location of short track | Length of short track (m) | Probability (%) |
|---|---|---|---|---|
| $S_1$–$S_2$ | 27,370 | $S_1$–$dS_{11}$ | 10,000 | 33 |
| | | $dS_{11}$–$dS_{12}$ | 10,000 | 33 |
| | | $dS_{12}$–$S_2$ | 7370 | 34 |
| $S_2$–$S_3$ | 31,900 | $S_2$–$dS_{21}$ | 10,000 | 33 |
| | | $dS_{21}$–$dS_{22}$ | 10,000 | 33 |
| | | $dS_{22}$–$S_3$ | 11,900 | 34 |
| $S_3$–$S_4$ | 28,030 | $S_3$–$dS_{31}$ | 10,000 | 33 |
| | | $dS_{31}$–$dS_{32}$ | 10,000 | 33 |
| | | $dS_{32}$–$S_4$ | 8030 | 34 |
| $S_4$–$S_5$ | 36,610 | $S_4$–$dS_{41}$ | 10,000 | 25 |
| | | $dS_{41}$–$dS_{42}$ | 10,000 | 25 |
| | | $dS_{42}$–$dS_{43}$ | 10,000 | 25 |
| | | $dS_{43}$–$S_5$ | 6610 | 25 |
| $S_5$–$S_6$ | 44,650 | $S_5$–$dS_{51}$ | 10,000 | 25 |
| | | $dS_{51}$–$dS_{52}$ | 10,000 | 25 |
| | | $dS_{52}$–$dS_{53}$ | 10,000 | 25 |
| | | $dS_{53}$–$S_6$ | 14,650 | 25 |
| $S_6$–$S_7$ | 26,800 | $S_6$–$dS_{61}$ | 10,000 | 33 |
| | | $dS_{61}$–$dS_{62}$ | 10,000 | 33 |
| | | $dS_{62}$–$S_7$ | 6800 | 34 |
| $S_7$–$S_8$ | 17,200 | $S_7$–$dS_{71}$ | 10,000 | 50 |
| | | $dS_{71}$–$S_8$ | 7200 | 50 |
| $S_8$–$S_9$ | 28,900 | $S_8$–$dS_{81}$ | 10,000 | 33 |
| | | $dS_{81}$–$dS_{82}$ | 10,000 | 33 |
| | | m–$S_9$ | 8900 | 34 |
| $S_9$–$S_{10}$ | 42,010 | $S_9$–$dS_{91}$ | 10,000 | 25 |
| | | m–$dS_{92}$ | 10,000 | 25 |
| | | $dS_{92}$–$dS_{93}$ | 10,000 | 25 |
| | | $dS_{93}$–$S_{10}$ | 12,010 | 25 |

(3) Failure times for other tracks are also exponentially distributed with a mean of 86,400 s (24 h) times 1/ratio value. That is, failure times of other tracks have expected values inverse ratio to their lengths, namely the longer the track part the more frequently the track failure occurrence. For instance, it is assumed that failure time of the longest track is distributed exponentially with a mean of 33,283 s (9.25 h), that is, it is expected to observe more than two (24/9.25 = 2.6) failures in a day for the longest track.
(4) The repair time distribution is exponential with 1800 s (0.5 h) mean for the *base track*.
(5) Repair times for other tracks are also exponentially distributed with a mean of 1800 s (0.5 h) times related ratio value.
(6) After failures for the tracks were created according to the probability distributions shown in Table 3, these failures are transferred to the links due to the probabilities given in Table 4.
(7) If a track failure happens while a train is traversing on this track and if the next station is a dummy one, train goes to next dummy station and a check is made if the failure is on that train's destination direction or not. If the failure is on its destination side, the train waits until failure is repaired, else the train goes on its trip.

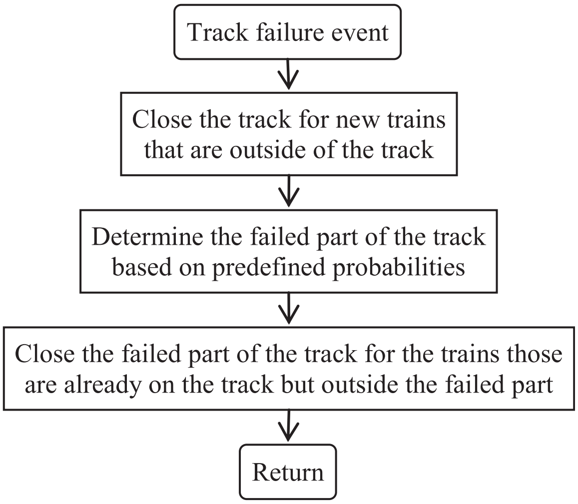


**Fig. 2.** Flowchart for track failure event.

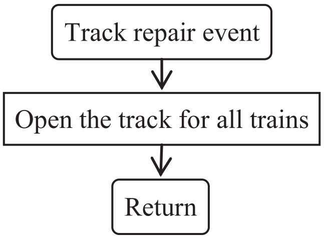


**Fig. 3.** Flowchart for track repair event.

*3.2.3. Train movement modelling*

Train movement logic from park area to a real station via a terminus, at a real station and at a dummy station are expressed by event diagrams denoted in Figs. 4–9. The event diagrams for *train movement from the park area event* and *train arrival to the terminus event* are given in Figs. 4 and 5 respectively. The event diagrams for *train arrival to the real station event* and *train departure from the real station event* are exhibited in Figs. 6 and 7 respectively. The event diagrams for *train arrival to the dummy station event* and *train departure from the dummy station event* are exhibited in Figs. 8 and 9 respectively.

Assumptions related to the train movement part of the simulation model are as follows.

(1) Each train stops at real stations except terminuses.
(2) A train stops at a dummy station if there is a failure in a track placed in front of that train.
(3) Trains' speeds are uniformly distributed over an interval (90 km/h, 110 km/h).
(4) Trains that have reverse directions can cross each other only at the real stations.
(5) Dwell times for each station are 600 s (10 min). That is each train will stop at least 600 s at the all stations for boarding and alighting events.
(6) To represent unplanned delays at a station, delay time is defined. It is assumed that delay time is exponentially distributed with a mean of 90 s. Delay time is added to the dwell times. Due to this unplanned delay, overtaking is possible.
(7) Track occupying decision is taken at the real stations based on the answers given the following questions; Are the links and intersections suitable? Does a track failure exist? Does this decision cause a deadlock?
(8) First come first served (FCFS) dispatching rule is used to select one train among the candidate trains, which are the trains waiting at neighbour stations of the track that want to use the same track and has finished waiting for dwell time and additional unplanned delay time. Namely the candidate trains are the trains that deserved to begin checking the conditions. If the all conditions to move are suitable for a candidate train, which arrived first to one of the neighbour station of the track it will begin to trip, else the same check is made for another train arrived second. Checking goes on until a suitable train is found.

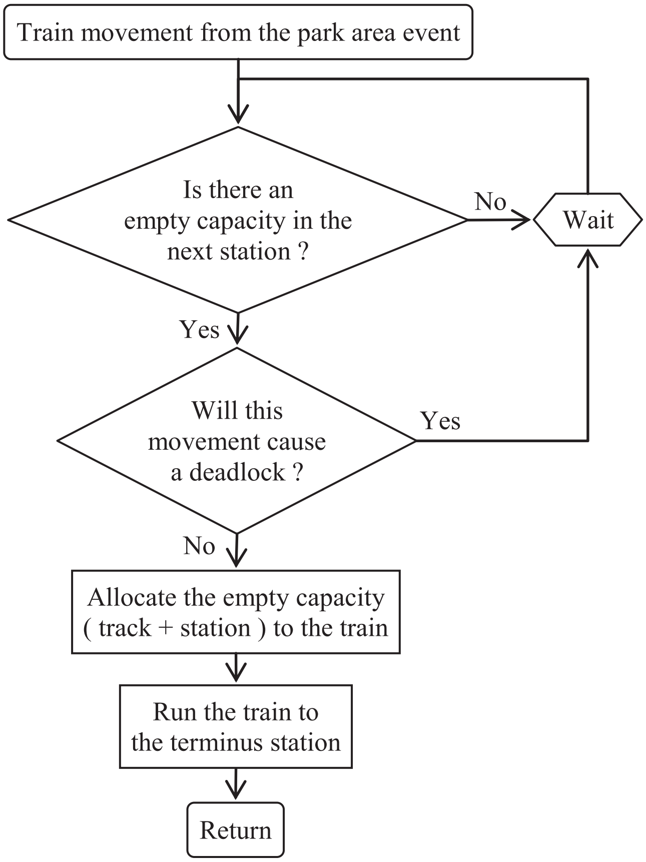


**Fig. 4.** Flowchart for train movement from the park area event.

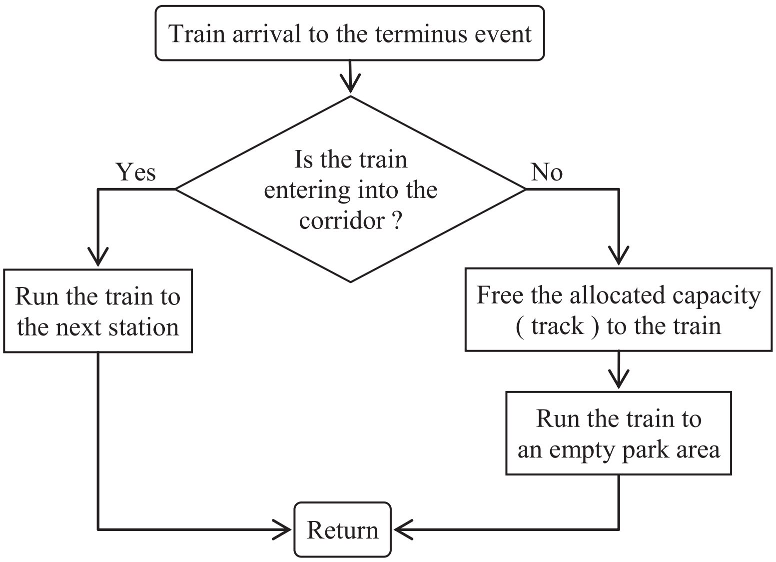


**Fig. 5.** Flowchart for train arrival to the terminus event.

### *3.2.4. Blockage preventive algorithm*

A common potential deadlock is exhibited in Fig. 10 where there are four trains; two trains are the WB trains and located at S($i$) and the other trains are the EB trains and located at S($i+1$). As can be seen in this figure, the system has a deadlock. Deadlock situation goes on until one of those trains reverses its direction.

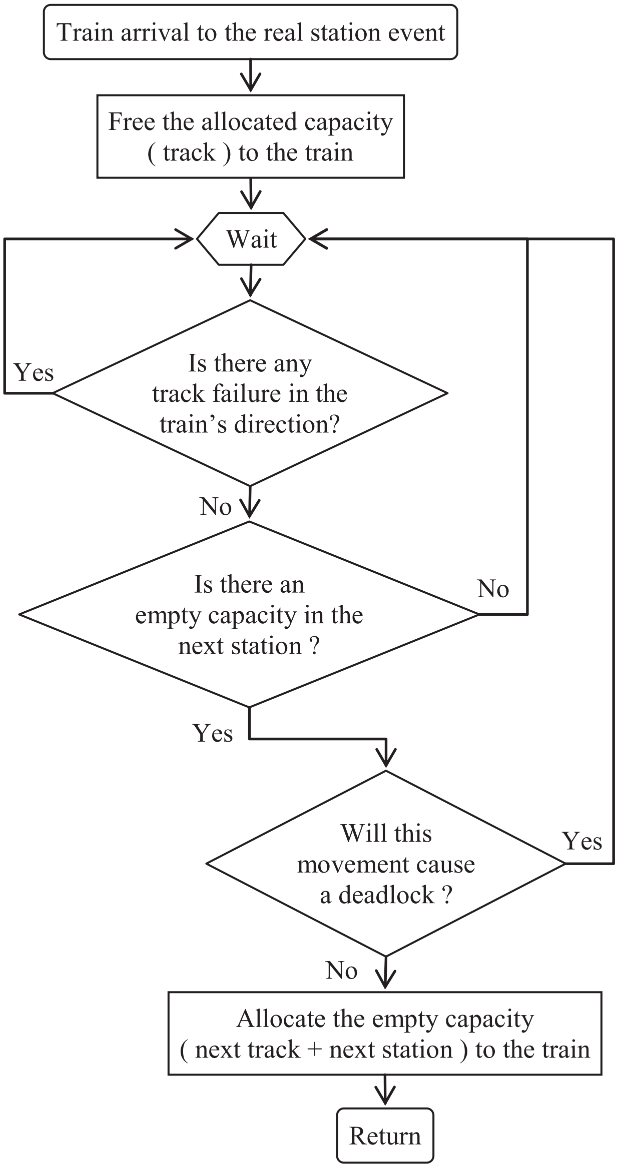


**Fig. 6.** Flowchart for train arrival to the real station event.

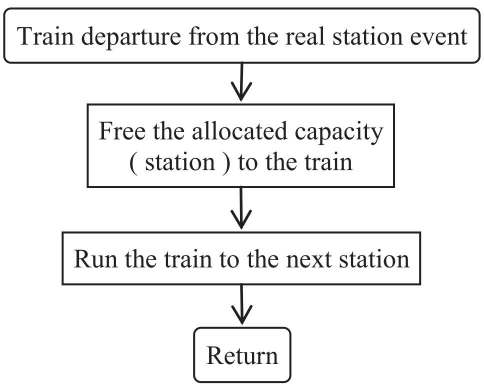


**Fig. 7.** Flowchart for train departure from a real station event.

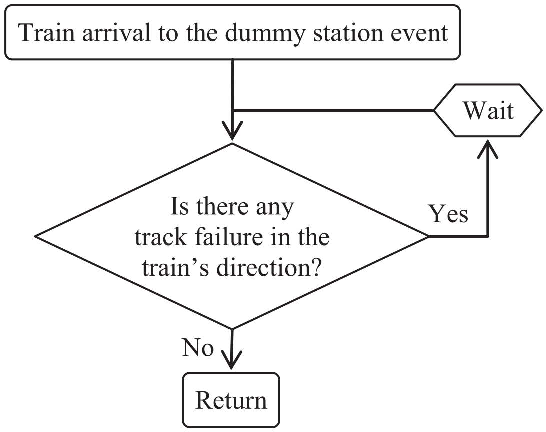


**Fig. 8.** Flowchart for train arrival to the dummy station event.

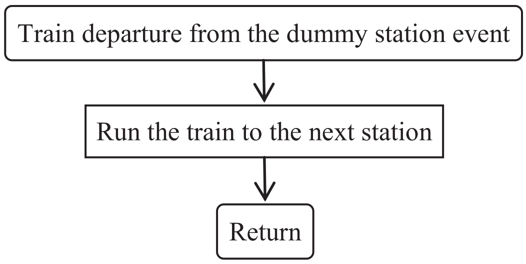


**Fig. 9.** Flowchart for train departure from the dummy station event.

Another example of a deadlock is shown in Fig. 11. There are two WB trains at S($i$), two EB trains at S($i+2$) and two trains (not important to be a WB or an EB train) at S($i+1$). Since we avoid reversing the direction of trains the deadlock problem cannot be solved. In order to obtain a feasible train timetable we must take course of action to prevent a deadlock.

In a recent study, Pachl [21] reviewed the existing strategies proposed to prevent the deadlock problem in railway systems. The author concluded that the proposed models are based on the game board philosophy and are not suitable to be used in control logic of real simulation systems. Pachl [21] proposed a rule-based deadlock avoidance method for simulation systems which follows the idea that a specified number of track sections ahead of a train must be reserved before this train is allowed to enter a track section. The number of track sections to be reserved depends on a set of logical rules. Distinctively, in our study long track sections are controlled in order to prevent a conflict which occurs if the long track section is used at the same time by the two trains in opposite directions. On the other hand, in order to avoid a deadlock that prevents the movement of the trains, we developed the *blockage preventive algorithm* that does not control the track sections but controls the real stations. Thanks to the algorithm given in Table 5, which follows the idea that the whole real stations in the direction of the train are checked before permitting the train to departure from its current station.

Our algorithm first checks the next station in front of the train while the train is in a station. For this train, if there is an empty capacity in the next station, the algorithm checks whether it is the last station or not. If the next station is the last

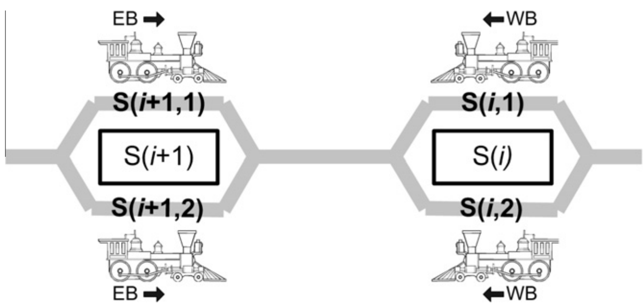


**Fig. 10.** An example of a deadlock.

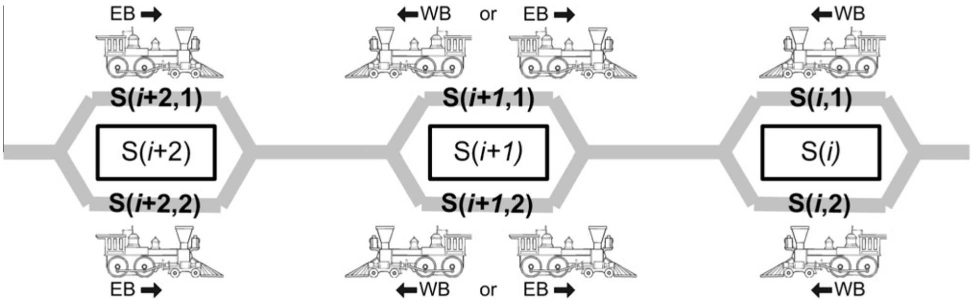


**Fig. 11.** Deadlock between six trains.

station, the checking finishes and the train goes on its trip, otherwise the checking goes on for the station behind the next station and checking lasts until the last station is checked. The empty capacities of the checked stations and also the directions of the trains that occupy the checked stations are important. The algorithm checks the whole real stations in front of the train, it never permits a deadlock and guaranties to obtain a feasible train schedule. The aim is to prevent any train departure that will cause a deadlock in the future.

The simulation model is verified by developing the model in a modular manner, using interactive debuggers, substituting constants for random variables, manually checking the results and animating the system. In order to develop the simulation model in a modular manner as explained in detail in Section 3.2 a step by step approach that gives ability to systematically model a complex system was used. The animation part of the simulation model was built by using the *Animate* tool of the Arena, to see if model is working as intended, and to understand the system clearer.

## 4. Discussion

In this section we discuss on some results of the simulation model. As a first step, we begin with an initial train timetable we obtained. This timetable is infeasible since it includes conflicts. Secondly, we focus on a feasible, conflict free initial train timetable we obtained by the deterministic simulation model. Lastly, a feasible train timetable we obtained by the stochastic simulation model is given by detail.

### 4.1. Infeasible planned initial train timetable

The trains are scheduled on a single track corridor regarding the initial train timetable, where all the inputs are deterministic, given in Table 2. The corridor has 10 real and 20 dummy, totally 30 stations. There is no randomness in the planned train arrival and departure times, failures and repairs are excluded, train speeds are fixed at 100 km/h, and no additional delay is added to the dwell times.

We begin our discussion on an empty corridor, on which only the $WB_1$ train is running. The simulation model is run for this scenario and the train–station diagram exhibited in Fig. 12 is obtained. The train travel time of the $WB_1$ is calculated as 16,350 s.

The next scenario is related to the $EB_1$ train that is running on the empty corridor. The train–station diagram exhibited in Fig. 13 is obtained by the deterministic simulation model. The calculated train travel time is the same as the previous one, 16,350 s.

After that, the train–station diagram for the planned initial train timetable given in Table 2 is manually obtained. This diagram is based on the $WB_1$ and the $EB_1$ train timetables, and depicted in Fig. 14. The timetable in Fig. 14 is not conflict free, it is infeasible. The conflict locations are indicated by dotted line circles, and it is calculated that there are 44 conflicts to be solved in such a deterministic system.

To display the problem clearer the conflicts between the $WB_3$ and some EB trains are displayed in Fig. 15. As it is seen in this figure, the $WB_3$ will have the first conflict with the $EB_1$ train between the $S_1$ and the $S_2$. The other conflicts will be with the $EB_2$, the $EB_3$, the $EB_4$ and the $EB_5$ trains between the $S_3$–$S_4$, between the $S_5$–$S_6$, between the $S_7$–$S_8$, and between the $S_9$–$S_{10}$, respectively.

### 4.2. Feasible planned initial train timetable

The developed simulation model framework, which is explained in detail in Section 3.2, has the ability to solve the conflicts and to create a conflict free feasible train timetable. The link control via the variables in the framework prevents the usage of the same long track part by the trains that have an opposite directions (the cause of the conflicts). In this scenario all the inputs are taken to be deterministic. That is, it is not allowed randomness in the planned train arrival and departure

**Table 5**
Blockage preventive algorithm.

Assume that a WB train is at the S($i$, $j$) station and aimed to travel to the S($i+1$, 1) station
where $i = 1,\ldots,9$ denotes the station number and $j = 1, 2$ denotes the platform number

**Step = 1**
**S($i$ + 1, 1) empty?**
*Yes*: Go next
*No*: Prohibit the train to travel, STOP the algorithm for that train
***i* + 1 = 10?**
*Yes*: Permit the train to travel, STOP checking blockage
*No*: Go next

**Step = 2**
**S($i$ + 1, 2) empty *or* allocated to an EB train?**
***or* S($i$ + 2, $j$) empty *or* allocated to a WB train?**
*Yes*: Go next
*No*: Prohibit the train to travel, STOP the algorithm for that train
***i* + 2 = 10?**
*Yes*: Permit the train to travel, STOP checking blockage
*No*: Go next

**Step = 3**
**S($i$ + 1, 2) empty *or* allocated to an EB train?**
***or* {S($i$ + 2, $j$) empty?}**
***or* S($i$ + 3, $j$) empty *or* allocated to a WB train?**
*Yes*: Go next
*No*: Prohibit the train to travel, STOP the algorithm for that train
***i* + 3 = 10?**
*Yes*: Permit the train to travel, STOP checking blockage
*No*: Go next

**From Step = 4 to Step = 8**
**S($i$ + 1, 2) empty *or* allocated to an EB train ?**
***or* {S($i$ + $k$, $j$) empty?; $k$ = 2,..., (Step − 1)}**
***or* S($i$ + Step, $j$) empty *or* allocated to a WB train ?**
*Yes*: Go next
*No*: Prohibit the train to travel, STOP the algorithm for that train
***i* + Step = 10?**
*Yes*: Permit the train to travel, STOP checking blockage
*No*: Go next

**Step = 9**
**S($i$ + 1, 2) empty *or* allocated to an EB train?**
***or* {S($i$ + $k$, $j$) empty?; $k$ = 2,..., (Step − 1)}**
***or* S($i$ + Step, $j$) empty *or* allocated to a WB train?**
*Yes*: Permit the train to travel, STOP the algorithm for that train
*No*: Prohibit the train to travel, STOP the algorithm for that train

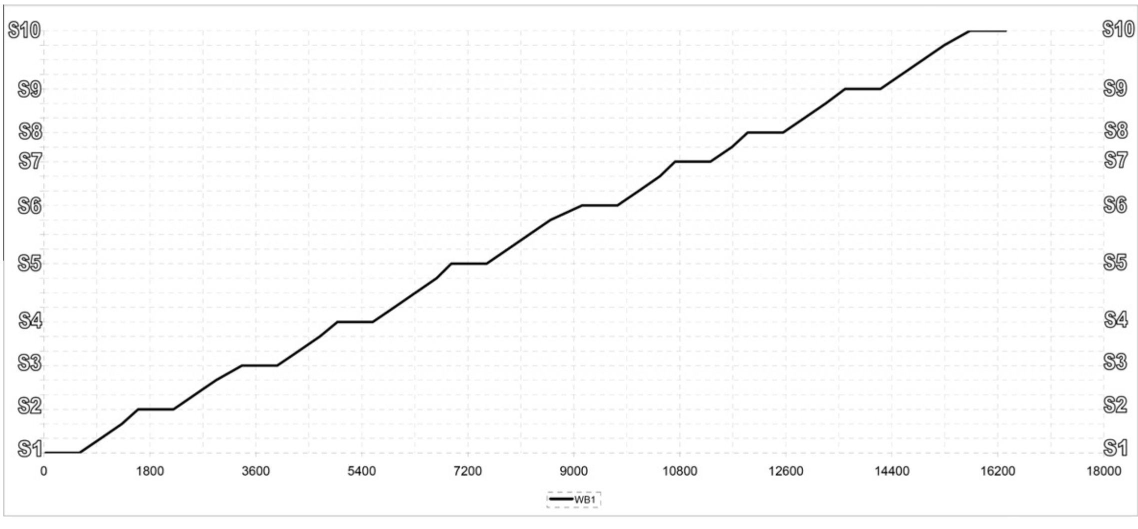


**Fig. 12.** Train–station diagram for the $WB_1$.

times. Train speeds are fixed at 100 km/h. The simulation model is run for these deterministic input values. The feasible train–station diagram obtained by our deterministic simulation model is given in Fig. 16.

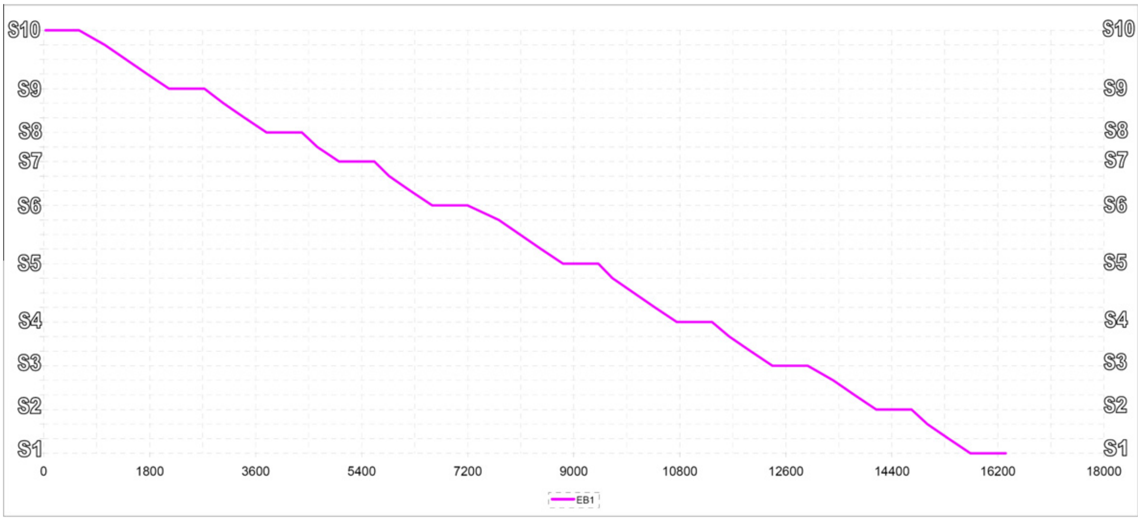


**Fig. 13.** Train–station diagram for the $EB_1$.

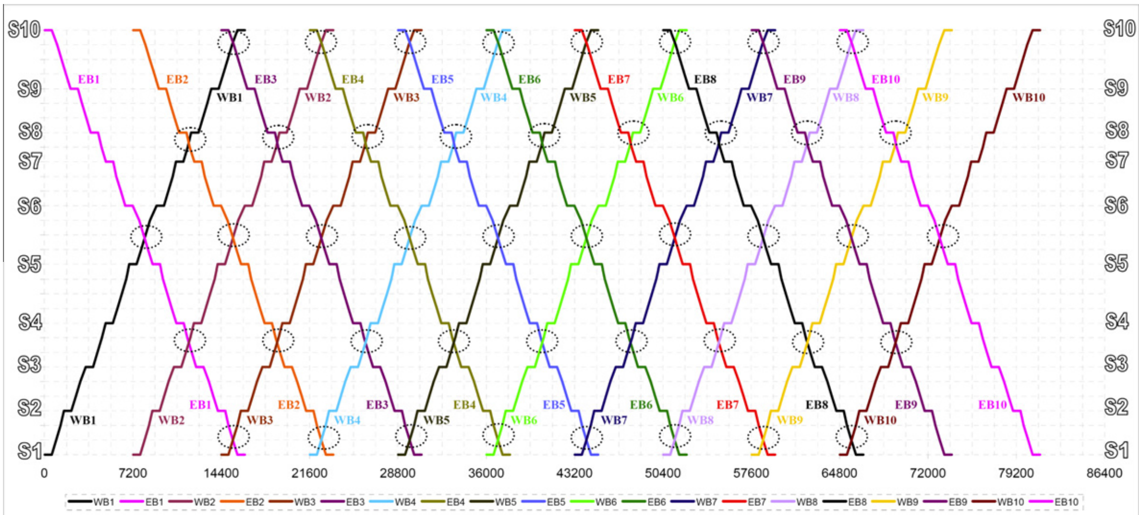


**Fig. 14.** Infeasible train–station diagram for the planned initial train timetable.

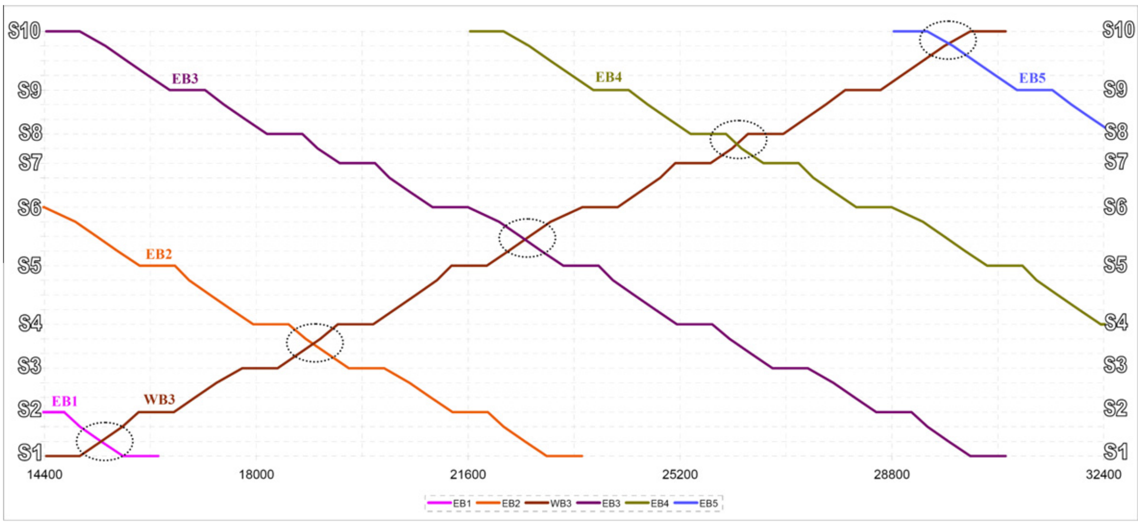


**Fig. 15.** Infeasible train–station diagram for the $WB_3$.

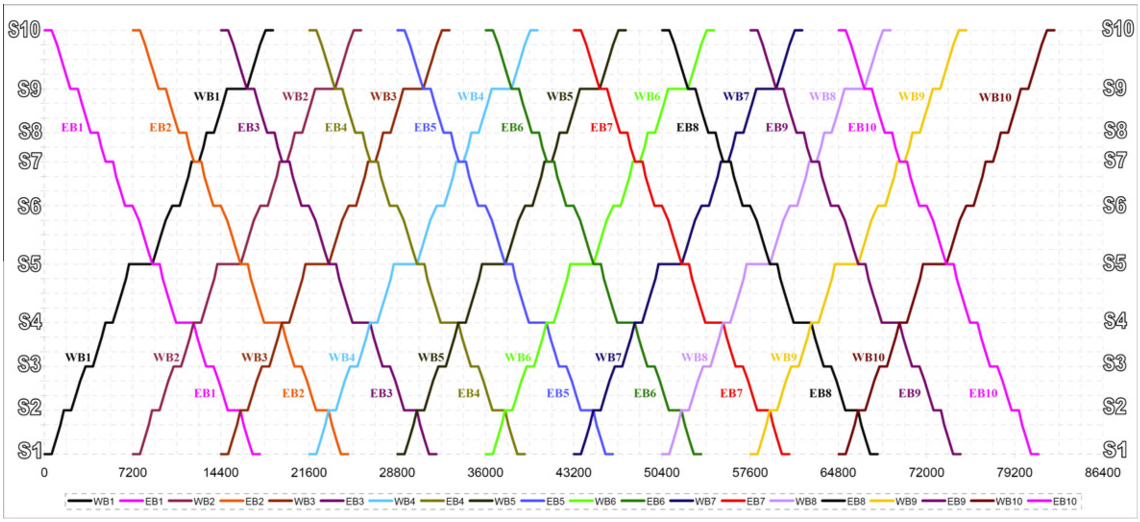


**Fig. 16.** Feasible train–station diagram for the planned initial train timetable.

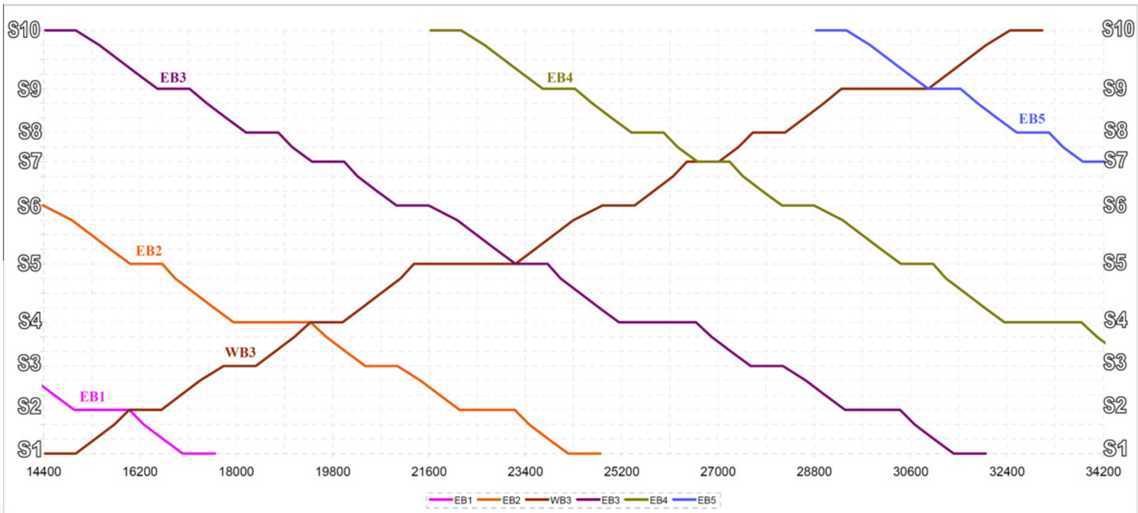


**Fig. 17.** Feasible train–station diagram for the $WB_3$.

Fig. 17 shows the conflict free feasible train station diagram for the $WB_3$. We see that the $EB_1$ train waits at the $S_2$ and permits the $WB_3$ to travel from the $S_1$ to the $S_2$, and the $EB_2$ waits the $WB_3$ at the $S_4$ for empty the track part between the $S_4$ and the $S_3$. On the other hand the $WB_3$ waits at $S_5$ to permit $EB_3$ to travel from the $S_6$ to the $S_5$. Since the $WB_3$ spends additional time while waiting the $EB_3$, the $WB_3$ meets the $EB_4$ at the $S_7$ and passes without spending additional time. The $WB_3$ waits the $EB_5$ at the $S_9$ and then finishes its trip. The calculated average train travel time for this scenario is 17,956 s.

In order to observe the change in the computer running time versus the number of trains in the deterministic simulation model, the model is run for different number of trains, the result are shown in Fig. 18. As it is seen in Fig. 18a the computer running time is increasing nonlinearly when we increase the number of trains in the system. On the other hand, Fig. 18b shows the increments in computer running time versus the added two trains. It is seen that the amount of increment caused by addition of two trains into the system depends on the number of current trains in the system. To make it clearer, we assume that there are four trains in the system. In this case, the computer running time is 9.0 s. When two new trains are added in the system (i.e., there will be six trains in the system), the computer running time will be 15.0 s (i.e., the increase in computer running time will be 6.0 s). On the other hand the increment will be 17.4 s if two trains are also added into the system that has 18 trains.

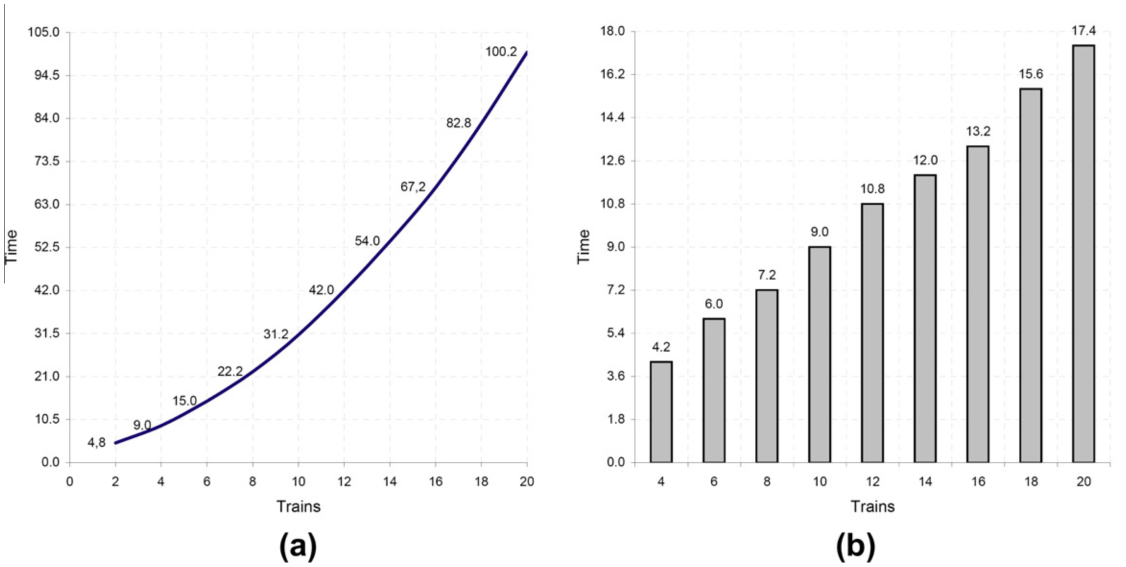


**Fig. 18.** Computer running time for 20 replications versus the number of trains in the system.

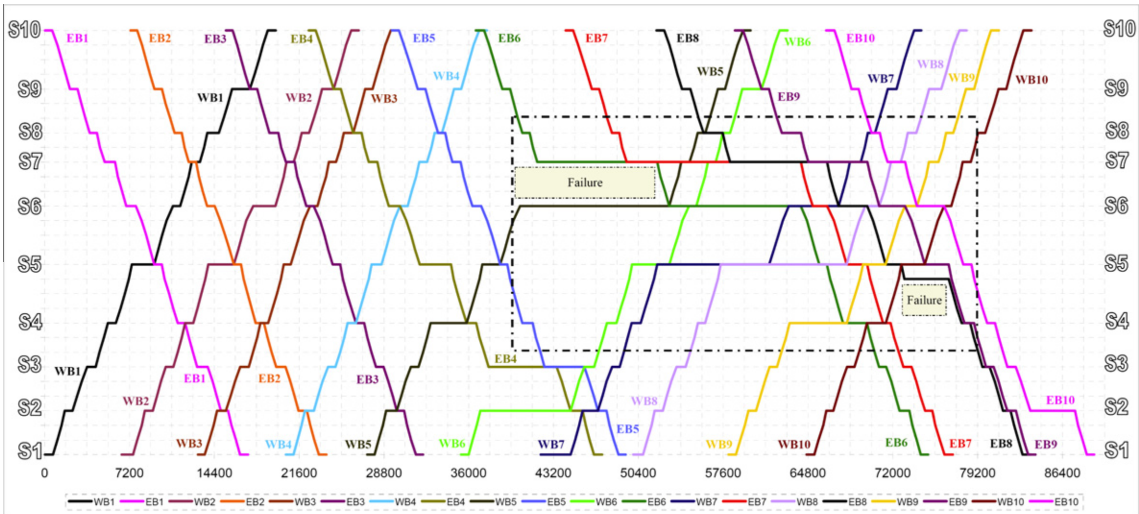


**Fig. 19.** Feasible train–station diagram.

### 4.3. Feasible train timetable

In this section we discuss on a feasible train timetable obtained by our stochastic simulation model. The train–station diagram of the feasible solution of which the calculated average train timetable is 24,218 s is given in Fig. 19.

In order to see what happens in the system after a failure has occurred, Fig. 20 should be examined. In this figure, while the simulation model is running through 39,600–79,200 s a part of the system between the stations $dS_{31}$ and the $dS_{81}$ is displayed. As can be seen in the dotted line circle denoted by 1, a failure occurred after the $EB_8$ train begins its trip from the $S_5$ to the $S_4$. Therefore, the $EB_8$ waits at the $dS_{43}$ during the repair, and then the $EB_8$ and the $EB_9$ trains traverse on the track part between the $S_5$–$S_4$.

While the simulation model is running through 39,600–70,200 s, a part of the system between $dS_{43}$ and the $dS_{71}$ is displayed in Fig. 21, which is the dotted line rectangle denoted by 2 in Fig. 20. In this figure, if we look at the dotted line rectangle, there is a track failure between the $S_6$ and $S_7$. After the track is repaired the trains can travel. But at that time there are more than one candidate trains waiting for using the repaired track part.

To make it clearer we verbally explained the important events on the track part between the $S_5$–$S_7$, while the simulation model is running through 39,600–70,200 s, depicted in Fig. 21. At 39,600 s, all the three stations are empty, the $WB_5$ is travelling between the $S_5$–$S_6$, and there is a failure event between the $S_6$–$S_7$. After that, the $WB_5$ reaches $S_6$, the failure is still

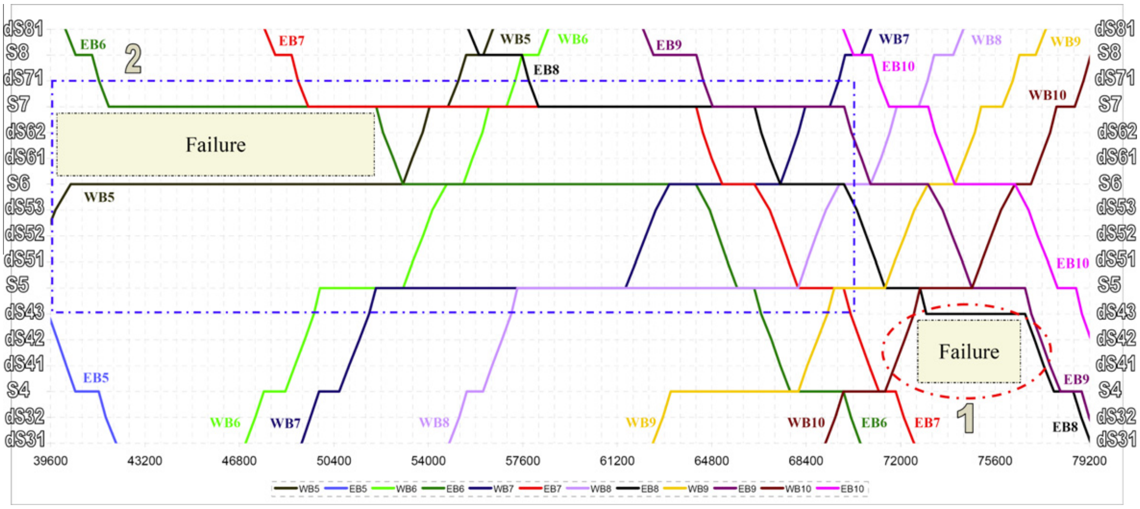


**Fig. 20.** Feasible train–station diagram for the $dS_{31}$–$dS_{81}$ part from 39,600 to 79,200 s.

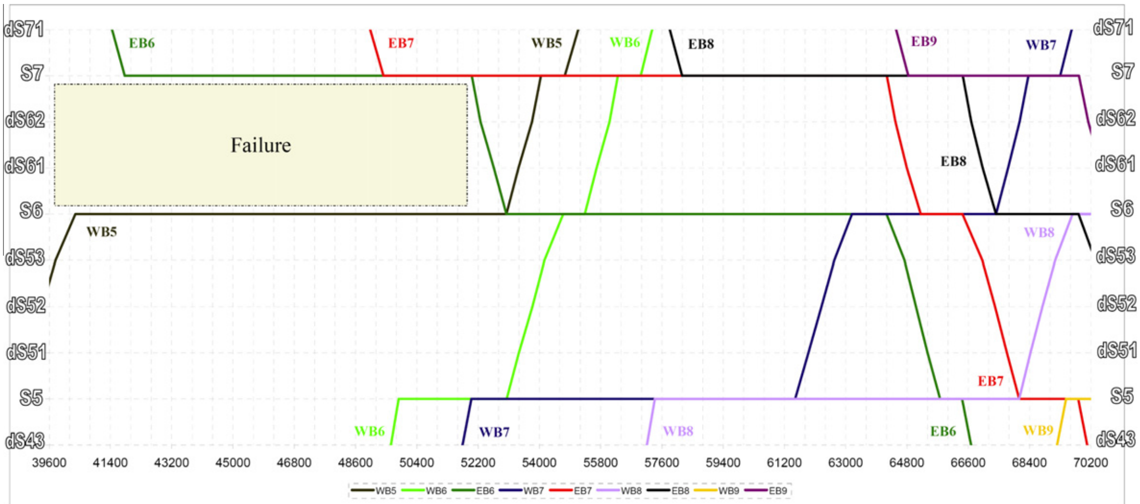


**Fig. 21.** Feasible train–station diagram for the $dS_{43}$–$dS_{71}$ part from 39,600 to 70,200 s.

going on. Next, the $EB_6$ reached to the $S_7$, and then the other part of the $S_7$ is occupied by the $EB_7$, the trains are waiting because the failure is still going on. Later, the $WB_6$ occupies the $S_5$, and then the other part of the $S_5$ is occupied by the $WB_7$. Although there is an empty capacity at the $S_6$, both the $WB_6$ and the $WB_7$ stop at the $S_5$, since a movement from the $S_5$ to the $S_6$ will cause a blockage. The failure is still going on, and the *five* trains are waiting for the track repair. After that, it is seen that the failure track has been repaired and has opened for the candidate trains. Although the first train in the queue is the $WB_5$, its move will cause a blockage. Thus, the $EB_6$ moves and uses the repaired track part. Next the $WB_5$ left and now there are four trains at this part (the track part between the $S_5$–$S_7$) of the system. Then, the $WB_6$ left, now there are three trains. After that, a new train, the $WB_8$, entered from $S_5$, now there are four trains. Next, a new train, the $EB_8$, entered from $S_7$, now there are five trains. Afterward, a new train, the $EB_9$, entered from $S_7$, now there are six trains. Then, the $EB_6$ left from $S_5$, now there are five trains. Next, the $WB_7$ left from $S_7$, now there are four trains. After that, a new train, the $WB_9$, entered from $S_5$, now there are five trains. Then, the $EB_7$ that is the last one entered this track part before its repair left from $S_5$, now there are four trains. Lastly, the simulation time is 70,200 s and there are now four trains that have entered this part after the repair at this part.

## 5. Conclusions

The *TrnSchPrb* is the problem of determining a timetable for a set of trains which satisfies some operational constraints without violating track capacities. The objective in the train scheduling/timetabling is to prepare a feasible train timetable that includes arrival and departure times of all trains at the visited stations.

Simulation gives a chance to researchers to model complex problems that have stochastic nature. Although simulation modelling has been used in a few articles those focused on the scheduling/timetabling, none of them includes a comprehensive framework. In this study, we developed a comprehensive feasible timetable generator simulation modelling framework for the train scheduling/timetabling problem. The simulation model was developed to cope with the disturbances, therefore stochastic events were allowed in the simulation model. To cope with disturbances is also the interest of rescheduling/dispatching. Therefore, the simulation framework can also be used for the train rescheduling/dispatching problem if it can be feed by the real time data.

By using the presented approaches, all the railway transportation systems can be modelled with only problem/infrastructure specific modifications and feasible solutions can be easily attained. In order to avoid a deadlock, a general *blockage preventive algorithm* is developed. This algorithm can be embedded in the simulation model and can be easily adapted to problem/infrastructure specific modifications.

Future work directions can be as follows:

(1) We dealt with the problem from the service provider (train operating authority) point of view, but there are also the service users (passengers or freight transporting companies) in the system. The simulation modelling framework can be extended by including the service users.
(2) The infrastructure considered in the hypothetic problem is a single track corridor. It can be extended to have double track parts that can also be modelled as one way. The corridor can be extended to a network by adding other capillary lines which may be interurban or urban. Some of these capillary lines may cut the corridor, some may use only a part of tracks and stations of the corridor, and also any station of the corridor may be a last station of a capillary line. Our developed framework can be also used to model any network.
(3) The simulation modelling framework can be extended by including different train types, which have different priorities and different speed limits and visit different stations.
(4) If very long multi platforms where more than one train can accommodate are regarded, the problem of sequencing of trains on the multi platforms arises and new decision variables are needed to solve the problem.

## Acknowledgments

We want to thank TÜBİTAK for the support in the scope of "TÜBİTAK-BİDEB 2211-National Ph.D. Scholarship Programme", and the authors would gratefully like to extend their appreciation to the anonymous referees whose valuable suggestions lead to an improved organisation of this paper.

## References

[1] A.A. Assad, Models for rail transportation, Transportation Research Part A 14A (1980) 205–220.
[2] U. Brännlund, P.O. Lindberg, A. Nou, J.-E. Nilsson, Railway timetabling using lagrangian relaxation, Transportation Science 32 (4) (1998) 358–369.
[3] V. Cacchiani, A. Caprara, P. Toth, A column generation approach to train timetabling on a corridor, 4OR: A Quarterly Journal of Operations Research 6 (2) (2008) 125–142.
[4] A. Caprara, M. Fischetti, P.L. Guida, M. Monaci, G. Sacco, P. Toth, Solution of real-word train timetabling problems, in: 34th Hawaii International Conference on System Sciences, 2001, pp. 1–10.
[5] A. Caprara, M. Fischetti, P. Toth, Modeling and solving the train timetabling problem, Operations Research 50 (2002) 851–861.
[6] A. Caprara, L. Kroon, M. Monaci, M. Peeters, P. Toth, Passenger railway optimization, Handbooks in Operations Research and Management Science 14 (2007) 129–187.
[7] A. Caprara, M. Monaci, P. Toth, P.L. Guida, A lagrangian heuristic algorithm for a real-world train timetabling problem, Discrete Applied Mathematics 154 (2006) 738–753.
[8] E. Castillo, I. Gallego, J.M. Ureña, J.M. Coronado, Timetabling optimization of a single railway track line with sensitivity analysis, Top 17 (2) (2009) 256–287.
[9] J.-F. Cordeau, P. Toth, D. Vigo, A survey of optimization models for train routing and scheduling, Transportation Science 32 (4) (1998) 380–404.
[10] O. Frank, Two-way traffic on a single line of railway, Operations Research 14 (5) (1966) 801–811.
[11] U. Geske, Railway scheduling with declarative constraint programming, Lecture Notes in Artificial Intelligence 4369 (2006) 117–134.
[12] A. Higgins, E. Kozan, L. Ferreira, Heuristic techniques for single line train scheduling, Journal of Heuristics 3 (1997) 43–62.
[13] D. Jovanovic, P.T. Harker, Tactical scheduling of rail operations: the SCAN I system, Transportation Science 25 (1991) 46–64.
[14] W. Kelton, R. Sadowski, N. Swets, Simulation with Arena, McGraw-Hill Higher Education, 2009.
[15] Y. Lee, C.-Y. Chen, A heuristic for the train pathing and timetabling problem, Transportation Research Part B 43 (2009) 837–851.
[16] C. Liebchen, The first optimized railway timetable in practice, Transportation Science 42 (4) (2008) 420–435.
[17] S.Q. Liu, E. Kozan, Scheduling trains as a blocking parallel-machine job shop scheduling problem, Computers and Operations Research 36 (2009) 2840–2852.
[18] A.I. Mees, Railway scheduling by network optimization, Mathematical Computer Modelling 15 (1991) 33–42.
[19] A.M. Newman, L.K. Nozick, C.A. Yano, Optimization in the rail industry, Handbook of Applied Optimization (2002) 704–719.
[20] M.A. Odijk, A constraint generation algorithm for construction of periodic railway timetables, Transportation Research Part B 30 (6) (1996) 455–464.
[21] J. Pachl, Deadlock avoidance in railroad operations simulations, Transportation Research Board 90th Annual Meeting, Washington DC, USA, Paper No. 11-0175, 2011.
[22] E.R. Petersen, A.J. Taylor, A structured model for rail line simulation and optimization, Transportation Science 16 (1982) 192–206.
[23] B. Szpigel, Optimal train scheduling on a single track railway, Operational Research 72 (1973) 343–352.
[24] P. Tormos, A. Lova, F. Barber, L. Ingolotti, M. Abril, M.A. Salido, A genetic algorithm for railway scheduling problems, Studies in Computational Intelligence 128 (2008) 255–276.
[25] O. Wong, M. Rosser, Improving system performance for a single line railway with passing loops, New Zealand Operational Research 6 (1978) 137–155.

[26] X. Zhou, M. Zhong, Bicriteria train scheduling for high-speed passenger railroad planning applications, European Journal of Operational Research 167 (3) (2005) 752–771.
[27] X. Zhou, M. Zhong, Single-track train timetabling with guaranteed optimality: branch-and-bound algorithms with enhanced lower bounds, Transportation Research Part B 43 (2007) 320–341.