



A simulation-based optimization approach for passenger train timetabling with periodic track maintenance and stops for praying

Zahra Bahramian · Morteza Bagheri

Received: 8 July 2014 / Revised: 17 May 2015 / Accepted: 19 May 2015 / Published online: 19 June 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract This paper presents two optimization methods for solving the passenger train timetabling problem to minimize the total delay time in the single track railway networks. The goal of the train timetable problem is to determine departure and arrival times to or from each station in order to prevent collisions between trains and effective utilization of resources. The two proposed methods are based on integration of a simulation and an optimization method to simulate train traffic flow and generate near optimal train timetable under realistic constraints including stops for track maintenance and praying. The first proposed method integrates a cellular automata (CA) simulation model with genetic algorithm optimization method. In the second proposed approach, a CA simulation model combines with dynamically dimensioned search optimization method. The proposed models are applied to hypothetical case study to demonstrate the merit of them. The Islamic Republic of Iran Railways (IRIR) data and regulations have been used to optimize train timetable. The results show the first method is more efficient than the second method to obtain near optimal train timetabling.

Keywords Train timetabling problem (TTP) · Cellular automata (CA) · Genetic algorithm (GA) · Dynamically dimensioned search (DDS) algorithm · Station · Track maintenance · Praying

1 Introduction

The train timetabling is one of the most challenging and difficult problems in railway transportation planning. The train timetable qualification greatly impacts the service quality and operating cost. The goal of train timetabling models is to determine train arrival and departure times to and from each station with the minimum total delay time while satisfying a set of operational constraints.

Preliminary review on analytical railway optimization models was undertaken by Assad [1]. Similarly, Caprara et al. [2] reported an excellent state-of-the-art review of railway optimization problems. Finding the optimum train timetable while satisfying all constraints increases the complexity of this problem. Therefore, train timetabling problem belongs to the class of computationally difficult NP-hard problems. This implies that it is not expected to find exact algorithm for solving any instance to optimality in polynomial time [3, 4]. Therefore, meta-heuristic algorithms have been used to determine optimum solution. Optimization of train timetabling problem has been investigated in different studies [5–8].

Several attempts have been made to use complex search procedures including look-ahead search [9], backtracking search and meta-heuristics algorithms [10, 11], mixed-integer linear programming, branch and bound, tabu search (TS) [12], an enhanced local search heuristic (LSH), GA, TS, and two hybrid algorithms [13] and GA [14–17]. However, these studies have not optimized train traffic flow considering stops for track maintenance and praying.

Electronic supplementary material The online version of this article (doi:10.1007/s40534-015-0077-z) contains supplementary material, which is available to authorized users.

Z. Bahramian
GIS Division, Department of Surveying and Geomatic
Engineering, College of Engineering, University of Tehran,
1439957131 Tehran, Iran
e-mail: zbahramian@ut.ac.ir

M. Bagheri (✉)
School of Railway Engineering, Iran University of Science and
Technology, 1684613114 Tehran, Iran
e-mail: morteza.bagheri@iust.ac.ir

A number of studies have examined CA methodology to simulate train traffic flow [18–31]. However, these studies have not considered stops for track maintenance and praying. Also they have only focused on simulation and not dealt with optimization of train traffic flow.

To the best our knowledge, Ref. [32] is the only refereed publication dealing with train timetabling problem with considering train stops for praying using an integrated simulation model and evolutionary algorithm. However, the authors applied enterprise dynamics simulation with evolutionary path re-linking algorithm to solve the train scheduling problem and only focused on stops for praying and did not consider other operational constraints. The authors suggested further investigations using different simulation models and optimizations algorithms.

In summary, the research to date did not provide adequate scientific evidence to optimize train timetabling problem considering realistic constraints including stops for track maintenance and praying. In the absence of such evidence, current practice has been guided to optimize passenger train timetabling problem on a single-line railway using GA and DDS and evaluate the fitness of each chromosome using CA model under operational constraints.

The remainder of the paper was as follows. In Sects. 2 and 3, the proposed methods and the optimization results are presented and compared. Finally, Sect. 4 states the conclusion and the future works.

2 Proposed methodology

The proposed methods integrate a CA simulation and optimization algorithm to generate near optimal train timetable. The GA and DSS methods have been used to optimize the train timetable problem. The fitness (i.e., total delay of the trains) of each solution has been evaluated by a simulation of train traffic flow known to be CA method. Figure 1 shows the architecture of the proposed methods.

This paper integrates a CA simulation and optimization algorithm to generate near optimal train timetable. The assumptions underlying this study included the following statements. (1) The trains move on single line track. (2) Similar train type moves on with the same speed limits. (3) A train service begins from the first station and ends at the last station. (4) The minimum instantaneous distance between two successive trains is constant. (5) The train speed limit at stations is constant. (6) The distance that a train can move after decelerating at a station is constant. (7) The time and position of blocked segments due to maintenance are fixed. (8) The praying times at each station have been given.

The route characteristics (e.g., track and structure), rolling stock attributes (e.g., train), maintenance/inspection schedule, and praying time windows are given as follows:

- The infrastructure characteristics:
 - The position of stations
 - The minimum stop time at stations
 - The maximum train speed at stations

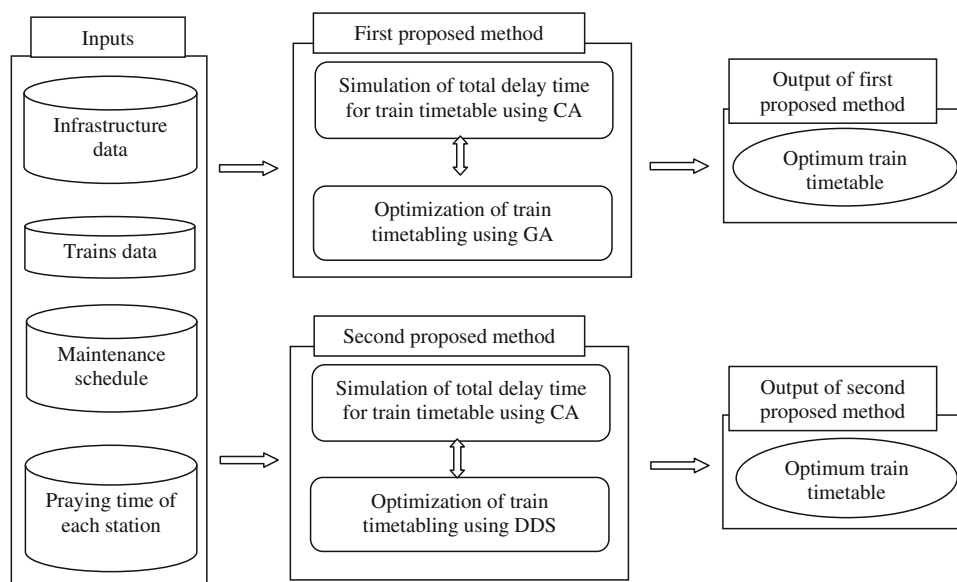


Fig. 1 The architecture of the proposed method

- The train characteristics:
 - The initial departure interval with tolerance
 - Safety rules
 - The maximum speed limitation
 - The minimum instantaneous distance between two successive trains
 - The distance that a train can move after decelerating at a station
- The maintenance schedule characteristics:
 - Track maintenance periods (section unavailability times)
 - The sections/segments that were blocked due maintenance operations
- The praying time characteristics:
 - The praying times of each station
 - The required time for praying at each station

2.1 Optimization model

In the train timetabling problem, the optimization objective is minimization of total delay time which includes the predefined time the train must stop in the stations, the time to stop the trains due to maintenance operations, and the time to stop the trains due to pray (Eqs. 1–7).

Minimize DelayTime

$$= \sum_{i=1}^{n_{\text{train}}} (\text{DelayTime}_{\text{PredefinedDwellTime}}^i + \text{DelayTime}_{\text{Maintenance}}^i + \text{DelayTime}_{\text{Praying}}^i), \tag{1}$$

$$\text{DelayTime}_{\text{PredefinedDwellTime}}^i = \sum_{j=1}^{n_{\text{station}}} (X_{\text{PredefinedDwellTime}}^{ij} \times 2 \text{ min}), \tag{2}$$

$$X_{\text{PredefinedDwellTime}}^{ij} = \begin{cases} 1, & \text{if } i\text{th train travels on } j\text{th station} \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

$$\text{DelayTime}_{\text{Maintenance}}^i = \sum_{k=1}^{n_{\text{Maintenance}}} (Y_{\text{Maintenance}}^{ik} \times t_{\text{Maintenance}}^{ik}), \tag{4}$$

$$Y_{\text{Maintenance}}^{ik} = \begin{cases} 1, & \text{if } i\text{th train stops for } k\text{th maintenance} \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

$$\text{DelayTime}_{\text{Praying}}^i = \sum_{l=1}^3 (Z_{\text{Praying}}^{il} \times 20 \text{ min}), \tag{6}$$

$$Z_{\text{Praying}}^{ik} = \begin{cases} 1, & \text{if } i\text{th train stops for } l\text{th maintenance} \\ 0, & \text{otherwise} \end{cases}, \tag{7}$$

where DelayTime is the total delay time, $\text{DelayTime}_{\text{PredefinedDwellTime}}^i$ is the predefined time the i th train must stop in the stations, $\text{DelayTime}_{\text{Maintenance}}^i$ is the time the i th train stop due to maintenance operations, and $\text{DelayTime}_{\text{Praying}}^i$ is the time the i th train stop due to pray. $X_{\text{PredefinedDwellTime}}^{ij}$, $Y_{\text{Maintenance}}^{ik}$, and Z_{Praying}^{il} are binary indicator variables. $t_{\text{Maintenance}}^{ik}$ is the required time that i th train stops to carry out the k th maintenance operation. This paper has used GA and DSS to optimize the train timetable problem.

2.1.1 Genetic algorithm

As train timetabling problem is known to be NP-hard [5, 7, 15, 33], a meta-heuristic algorithms have been applied to solve it. It has been shown that GA has high potential in finding the global optimum in a large, poorly defined search space even in the presence of difficulties such as high dimensionality, multi-modality, discontinuity, and noise [15]. GA has been successfully applied to combinatorial problems and is able to handle huge search spaces as those arising in scheduling problems [33]. Therefore, in this paper, GA has been used to optimize train timetable.

GA implemented optimization strategies based on simulation of the natural evolution of a species. Through repetitive application of selection, crossover, and mutation, a population of candidate solutions is evolved to generate better solution. More details on the GA process can be found in [34]. In this paper, the genetic parameters are listed in Table 1.

In this paper, the population size is 20. Each chromosome included 18 genes, which equals to the total number of trains. It consists of train departure times from origin station, during a 24 h period. Each train has a departure time interval which could be set by user. The default length of initial interval time is 2 h. It means that if the user set the assumed time of departure to 7, system can select any time between 5 and 9 for departure.

The first population was created randomly within maximum half an hour difference from the predefined timetable for each train. After generating the first population, evaluation of each chromosome was carried out using simulation. For calculating the fitness of each chromosome, a simulation model was used, and total delay of the trains was recorded as the value of fitness. Following the evaluation, the process of creating new generation is started.

Table 1 GA parameters

Metaphor	Optimization	Railway model
Evolution	Problem solving	Finding the optimum timetable
Individual	Solution	Train timetable
Fitness	Objective function	Total delay time
Environment	Optimization problem	Train timetabling problem
Locus	Elements of the solution	Each train departure time
Allele	Value of the element (locus)	Value of each train departure time

In this paper, after calculating fitness of all chromosomes (i.e., total delay time), the first and second best timetables (elites) are copied over to the next GA generation. GA functions are designed on the basis of the rank selection method. GA sorts the chromosomes in ascending order (better chromosome placed first). Selection opportunity of each chromosome is increased by its rank through GA operations.

In each generation, with a probability of 0.9 (crossover rate = 0.9), a pair of parent chromosome was selected for breeding and two children chromosomes were produced using one-point crossover operator. This meant that a single crossover point on both parents' chromosome was selected and all train departure time beyond that point in either parent was swapped between the two parents. Then, there were some chromosomes with some of their genes changed within maximum 1-h period by mutation. The mutation rate is set to 0.05. This process has been repeated until reaching a termination condition (predefined number of iteration). Experiments showed that around 400 generations had to be passed to achieve the result.

2.1.2 Dynamically dimensioned search algorithm

The train timetabling problem is an optimization problem. It has been shown that DSS has high potential in finding the global optimum without any algorithm parameter tuning in less iteration. DSS has been successfully applied to automatic calibration of watershed simulation models [35–38]. Therefore, in this paper, DSS has been used to optimize train timetable.

The DDS algorithm is a simple, stochastic, single-solution-based, heuristic, global search algorithm that was developed for the purpose of finding a good globally optimal solution within a specified maximum number of function evaluation limit [35–38]. The algorithm initially searches globally and becomes increasingly local when the number of iterations approaches the maximum number of iterations. The transition from global to local search

involves dynamically and probabilistically reducing the number of dimensions in the neighborhood. Candidate solutions are created by perturbing the current solution values in the randomly selected dimensions only. The DDS algorithm pseudo code for minimization is as follows (Eqs. 8–17).

- Step 1. Define DDS inputs:
 Neighborhood perturbation size parameter, r (0.2 is default)
 Maximum number of iterations, m
 Vector of lower, x^{\min} , and upper, x^{\max} , bounds for all decision variables (departure time of each train)
 Initial solution (predefined train timetable), $x^1 = [x_1, \dots, x_D]$
- Step 2. Set initial solution as the best solution and initialize i with 1:
 $F^{\text{best}} = F(x^i)$, and $x^{\text{best}} = x^i$. (8)
- Step 3. Select a subset of decision variables for perturbation:

Initialize delta= \emptyset (9)

for ith iteration
 $P(i)=1-\ln(i)/\ln(m)$ (10)

end
 for each decision variable d
 Get a uniform random number, rand
 if rand< $P(i)$
 delta(d)=1 (11)

end
 if delta= \emptyset
 rand'=ceiling(rand. D)
 delta(rand')=1 (12)

end

- Step 4. Generate a candidate solution:

Get a standard normal random number, randn
 for each decision variable d
 $x_d^{\text{new}} = x_d^{\text{best}} + (x_d^{\text{max}} - x_d^{\text{min}}) * \text{delta}(d) * r$ (13)

end
 if $x_d^{\text{new}} < x_d^{\text{min}}$
 $x_d^{\text{new}} = x_d^{\text{min}} + (x_d^{\text{min}} - x_d^{\text{new}})$
 if $x_d^{\text{new}} > x_d^{\text{max}}$
 $x_d^{\text{new}} = x_d^{\text{max}}$ (14)

end
 if $x_d^{\text{new}} > x_d^{\text{max}}$
 $x_d^{\text{new}} = x_d^{\text{max}} - (x_d^{\text{new}} - x_d^{\text{max}})$
 if $x_d^{\text{new}} < x_d^{\text{min}}$,
 $x_d^{\text{new}} = x_d^{\text{min}}$ (15)

end

- Step 5. Evaluate the candidate solution x^{new} , and update current best solution if necessary:

```

if  $F(x^{\text{new}}) \geq F^{\text{best}}$ 
     $F^{\text{best}} = F(x^{\text{new}})$ 
     $x^{\text{best}} = x^{\text{new}}$ 
end

```

16

- Step 6. Check stopping criterion:

```

if  $i < m$ 
    Go to Step 3
     $i = i + 1$ 
else
    Stop
end

```

(17)

The only algorithm parameter to set in the DDS algorithm is the scalar neighborhood size perturbation parameter (r) that defines the random perturbation size standard deviation as a fraction of the decision variable range. In this paper, the value of the r parameter is set to 0.2.

2.2 Simulation model

In this paper, the fitness (total delay of the trains) evaluation of each chromosome is carried out using a simulation method. The simulation methods can be categorized as synchronous simulation and asynchronous simulation [15]. Synchronous simulation models processed the movements of all trains at the same time, but asynchronous modeled rank trains according to their priorities and inserted trains into simulation procedure sequentially based on the ranks. In this paper, the synchronous simulation approach was applied for the scheduling trains.

Train traffic flow models can be divided into two major categories: macroscopic and microscopic [39]. In macroscopic models, the aggregate behavior of traffic was described and the characteristics of traffic flow like average velocity, density, flow, and mean speed of a traffic stream were studied. A main limitation of macroscopic models was their aggregate nature [39]. A primary advantage of microscopic models was the ability to study individual train motion position and velocity of the trains.

The CA model is one of the most popular microscopic modeling approaches. CA has been used to study complicated non-linear systems such as train traffic flow. The CA method proposed by Cremer and Ludwing [40] can be used to study complicated non-linear systems such as traffic flow simulation in railway transportation. Preliminary work on CA application in transportation traffic flow was undertaken in Ref. [41]. NaSch model is a one-dimensional probabilistic cellular automata model which models traffic flow on a single-lane. Space, time, and the state of NaSch model are discrete. It is a minimal model that reproduces

the basic features of real traffic. In comparison with other continuous models, it has very fast calculations. In NaSch model, the railway is divided into L cells ($i = 1, 2, \dots, L$), and the time is discrete. Each site can be either empty or occupied by a train with an integer speed $v_i = 0, 1, \dots, v_{\text{max}}$, where v_{max} is the maximum speed. According to four successive steps, all sites are simultaneously updated (Eqs. 18–22):

- Step 1. Acceleration

$$v_i(t+1) = \min(v_i(t) + 1, v_{\text{max}}). \quad (18)$$

- Step 2. Slowing down

$$v_i(t+1) = \min(v_i(t), \text{gap}_i(t)). \quad (19)$$

$$\text{gap}_i(t) = x_{i+1}(t) - x_i(t) - 1, \quad (20)$$

where $\text{gap}_i(t)$ expresses the gap between i th train and $(i+1)$ th train at time t and $x_i(t)$ is the position where i th train is at time t .

- Step 3. Randomization

$$v_i(t+1) = \max(v_i(t) - 1, 0). \quad (21)$$

In this step, $v_i(t)$ is decreased by 1 with randomization probability q if $v_i(t) > 0$.

- Step 4. Movement

$$x_i(t+1) = x_i(t) + v_i(t+1). \quad (22)$$

CA has been used to study train traffic flow. In this method, the dynamic system was divided to discrete cells in space and time, each in one of a finite number of states (empty or occupied by a train). For each cell, a set of rules is defined as its neighborhood. In initial state, a state is assigned to each cell. Using the current state of the cell, the states of the cells in its neighborhood, and a transition rule, the new state of each cell was determined. Using very simple rules, this model can reproduce the basic phenomena encountered in real traffic. By modification of the simple rules, more complex situations (such as the multi-lane traffic, bidirectional traffic, and the traffic with different types of trains) can be described. In this paper, a CA method has been used to simulate train traffic flow and calculate the total delay time of a train timetable. In addition, according to the IRIR needs, two constraints (stops for track maintenance and praying) were considered. The proposed model simulated the complex railway traffic using some simple transition rules as follows:

2.2.1 Case 1

When the $(n-1)$ th train was in front of the n th train at the t th time step, to avoid the collision between two successive trains, the distance between these trains must be larger than or equal to the minimum instantaneous distance. The

velocity of the n th train varied by comparing the headway distance (Δx_n) and the minimum instantaneous distance (d_n) to determine whether accelerating or decelerating (Eqs. 23–24).

• Step 1: Acceleration

$$\begin{aligned} &\text{if } \Delta x_n > d_n, v_n = \min(v_n + a, v_{\max}); \\ &\text{else if } \Delta x_n < d_n, v_n = \max(v_n - b, 0); \\ &\text{else } v_n = v_n \\ &\text{end.} \end{aligned} \tag{23}$$

• Step 2: Movement

$$x_n = x_n + v_n. \tag{24}$$

2.2.2 Case 2

When the n th train was near the station:

- If the station has been occupied by the train that was in front of the n th train, the safety distance must be maintained between the n th train and its leading train. The update rules were as same as the update rules of the case 1.
- If the station was empty, the n th train was allowed to travel into the station and stop at the station. Then it needed to wait for a time, and then left the station (Eqs. 25–30).

– Step 1: Acceleration

$$\begin{aligned} &\text{if } dx_n > d_b, v_n = \min(v_n + a, v_{\max}) \\ &\text{else if } dx_n < d_b, v_n = \max(v_n - b, 0) \\ &\text{else } v_n = v_n \\ &\text{end} \end{aligned} \tag{25}$$

where dx_n is the distance from the n th train to the station in front of the n th train, and d_b is the distance that the train n can arrive at the station by decelerating.

– Step 2: Slowing down

$$v_n = \min(v_n, \Delta x_n). \tag{26}$$

– Step 3: Movement

$$x_n = x_n + v_n. \tag{27}$$

– Step 4: Acceleration

If the actual dwell time of the n th train (t_{dwell}) was more than the planned dwell time (T_d)

$$v_n = \min(v_n + a, v_{\max}), t_{\text{dwell}} = 0. \tag{28}$$

If the actual dwell time of the n th train was less than the planned dwell time

$$v_n = 0. \tag{29}$$

– Step 5: Movement

$$x_n = x_n + v_n. \tag{30}$$

2.2.3 Case 3

In order to improve the safety of the railway, inspection and routine maintenance of track and structures are necessary. Based on predefined schedules of maintenance works, the (starting and ending) locations that carried out the maintenance works, the starting maintenance time and its duration (3 h) were determined. It is required to block the line so the crew can get to the site.

2.2.4 Case 4

Based on religious obligations, during predefined time periods (local praying time window), train must stop to perform praying services. Therefore, after starting praying time, each train stopped 20 min in the first next station in praying time windows. Passengers got off the train and prayed in the station mosque. It should be considered that each station had a different local praying time windows (starting and ending praying times).

In the proposed CA method, the parameters included the minimum instantaneous distance (d_n), the distance that the train can arrive at the station by decelerating (d_b), the planned dwell time (T_d), and predefined schedules of maintenance works. The IRIR data have been used to set them. The results of the proposed CA method were analyzed and compare with other methods in [31].

3 Case study

In this section, the proposed methodology is applied to a rail corridor that connects Tehran to Mashhad with 926 km of track and 50 stations while nine of them (Tehran, Varamin, Garmsar, Semnan, Damghan, Shahrood, Neghab, Neyshabur, and Mashhad) are considered as the main stations where train stops for getting on/off. The merits of a new simulation-based optimization model lies in its ability to achieve minimum total delay time with considering stops for routine track maintenance and praying. In this application, 10 trains traverse on a single-line rail corridor. It is assumed that the trains should dispatch within 2 h before or after their predetermined departure time. According to the predefined schedules of maintenance works, Garmsar–Semnan, Damghan–Shahrood, and Neyshabur–Mashhad section were blocked in 19–22, 13–16, and 4–7, respectively.

In the first proposed approach, an integration of a CA simulation and GA optimization algorithm was applied.

Table 2 The range and the best value of investigated genetic algorithm parameters

Parameter	Range	Best value
Population size	5,10,15	10
Number of best timetables that are copied over to the next generation	0.05,0.1,0.2	0.1
Crossover rate	0.9,0.65,0.4	0.9
Mutation rate	0.01,0.05,0.2	0.05

The best objective value in different iterations for GA has been achieved after convergence for solving the problem. In this paper, four genetic algorithm parameters are investigated, namely, population size, number of best timetables that are copied over to the next generation, crossover rate, and mutation rate with rang values shown in Table 2. The range values of investigated genetic algorithms parameters are examined, and the best value of each parameter is determined based on their fitness values (Table 2).

Using the best value of the parameters, the fitness at each GA iteration is shown in Fig. 2. Considering departure time constraint, GA found best fitness value (447 min) in the 241 iteration and stayed in this value up to the 400th

iteration, while the fitness value of the initial predefined schedules of train was 496 min. Therefore, the result revealed a potential for 9 % total delay time reduction over the currently used timetable. If there was no departure time constraint, the best fitness value is 383 min (in the 218 iteration) and total delay time reduced 23 % over the currently used timetable.

In the second proposed approach, an integration of a CA simulation and DSS optimization algorithm was applied. The scalar neighborhood size perturbation parameter (r) is set to 0.2. It considers departure time constraint. The total delay time at each DDS iteration is shown in Fig. 2. The DDS found best fitness value (456 min) in the 211 iteration and stayed in this value up to the 400th iteration. Therefore, the result revealed a potential for 8 % total delay time reduction over the currently used timetable.

Table 3 shows the predefined train schedule, the result of the GA optimization phase without or with departure time constraint (the trains dispatch within 2 h before or after their predetermined departure time) and the result of the DDS optimization phase with departure time constraint.

Fig. 3 shows the trajectory of the trains in optimum train schedule by the space–time diagram for the single-line railway using GA without or with considering departure

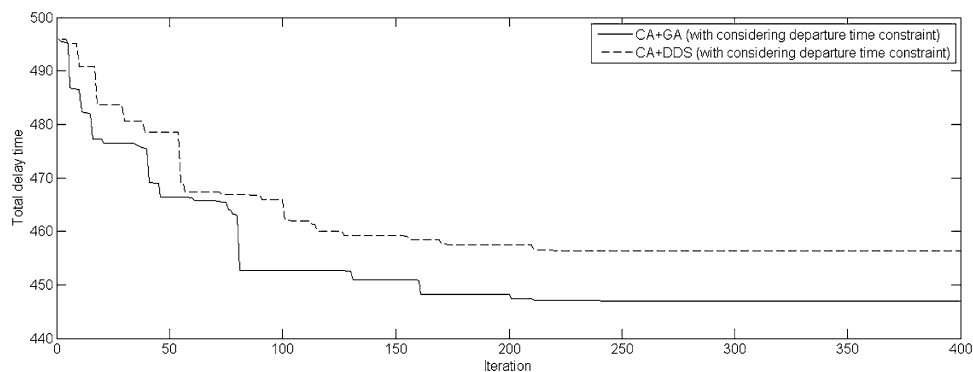
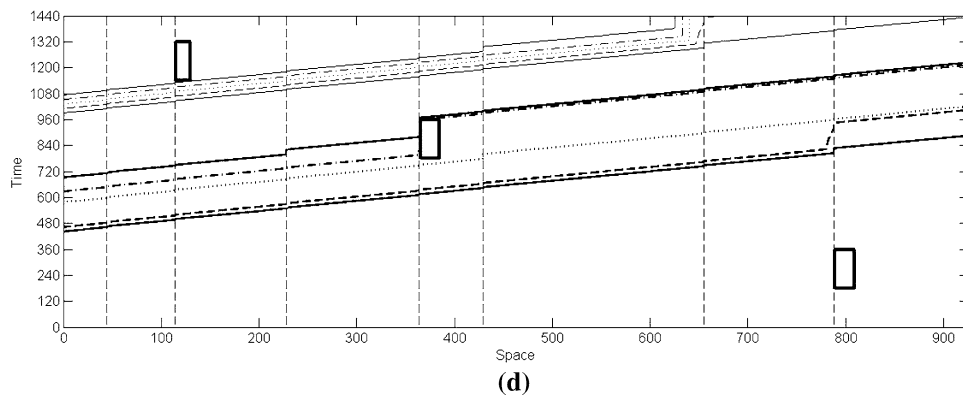
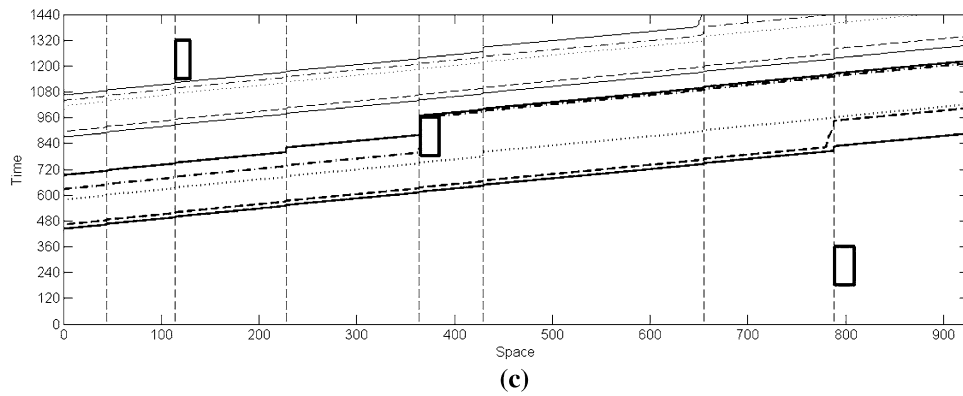
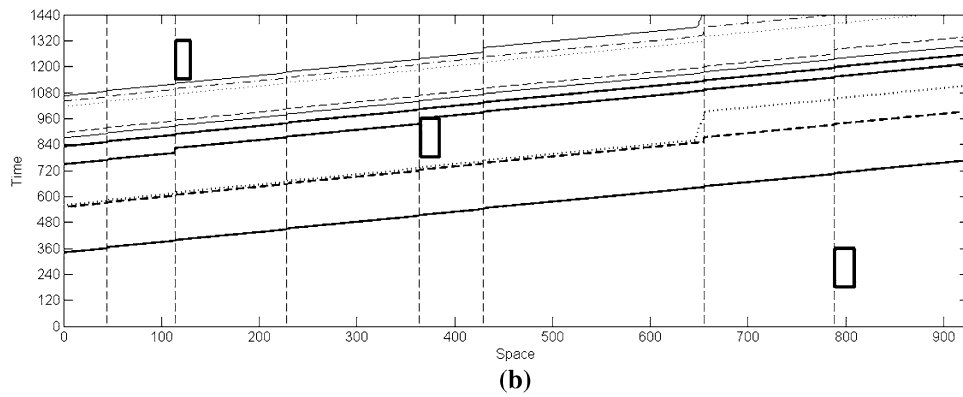
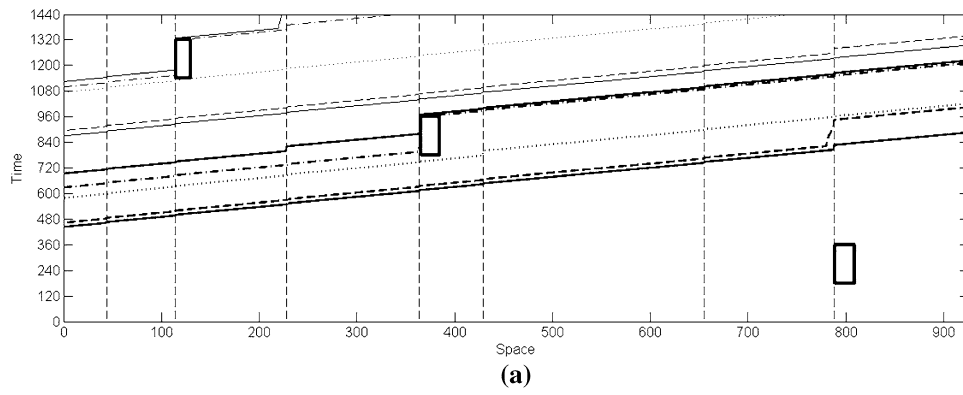


Fig. 2 The fitness of two proposed methods at each iteration

Table 3 The predefined and optimum train schedule (with or without departure time constraint)

Train number	1	2	3	4	5
Predefined departure time	7:20	7:40	9:35	10:25	11:30
Optimum departure time (GA without departure time constraint)	5:20	9:06	7:35	12:25	13:30
Optimum departure time (GA with departure time constraint)	6:57	7:12	8:49	10:36	11:54
Optimum departure time (DDS with departure time constraint)	6:15	7:19	9:40	10:31	11:28
Train number	6	7	8	9	10
Predefined departure time	14:10	14:35	16:35	17:00	17:25
Optimum departure time (GA without departure time constraint)	14:02	14:36	16:08	17:04	17:21
Optimum departure time (GA with departure time constraint)	14:13	14:38	16:22	16:54	17:34
Optimum departure time (DDS with departure time constraint)	13:04	14:58	16:10	16:45	15:25



◀ **Fig. 3** Space–time diagram of the (a) predefined train schedule (b) optimum train schedule using GA optimization algorithm without considering any constraint on the departure time of the train (c) optimum train schedule using GA optimization algorithm with considering departure time of the trains is within 2 h before or after their predetermined departure time (d) optimum train schedule using DDS optimization algorithm with considering departure time of the trains is within 2 h before or after their predetermined departure time

time constraint and DDS with considering departure time constraint. As can be seen, the railway was blocked in the time of track maintenance and each train stop at a station three times a day for praying. The proposed method minimizes the total delay time due to predefined train stop in the station, train stop due to maintenance operations and praying. In this example, the GA tried to reduce total delay time due to maintenance operations by having the trains departed later or sooner from its origin. If there was no constraint on departure time, the trains departed later or sooner to have no delay time due to maintenance operations. If there was departure time constraint, the delay time due to maintenance is as little as possible. Reducing travel delay due maintenance operations led to reduce the travel time and the need to stop for praying.

Train timetable deals with uncertainties and disturbances in their operations. Small-scale disturbances, often caused by minor delays, do not require a significant change of the schedules. Robust train timetable is necessary to absorb the small-scale disturbances. In this paper, to measure the resilience of a timetable to minor disruptions in dwell time and evaluate the robustness of the proposed approaches' results, uncertainty in dwell time is considered. An 8 min disturbance in one station, dwell time is simulated and its effect on the timetables of two constraint methods (integration of CA and GA, and integration of CA and DDS) are evaluated. The results show 0 and 9 extra total delay time in their train timetables, respectively. Therefore, integration of CA and GA is more robust rather than integration of CA and DDS.

4 Conclusion and future works

This paper integrates simulation and optimization algorithm to generate near optimal train timetable. The GA (in the first proposed method) and DDS (in the second proposed method) have been used to optimize train timetabling problem, while a CA simulation approach has been used to model the train traffic flow and evaluate the fitness of each solution (i.e., the total delay time generated during the optimization). In the proposed update rules, four cases were considered: a train was in front of another train, the train was near the station, the railway was blocked for

maintenance, and the train must stop in a station three times a day for praying. The results reveal GA is more efficient than DDS.

Specific recommendations for future research directions include developing the proposed method in more complex situations such as the multi-lane and bidirectional railway traffic, and considering different types of trains. The next step in the process of train timetabling optimization is to obtain a train timetable that minimizes the total delay time and maximizes the capacity of rail network simultaneously. Other meta-heuristic methods can be used to optimize train timetable. Also, optimization of maximum speed, dwell time, maintenance time, and rail station for praying can be considered.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Assad A (1980) Models for rail transportation. *Transp Res Part A* 14(3):205–220
2. Caprara A, Kroon L, Monaci M, Peeters M, Toth P (2006) Passenger railway optimization. In: Henderson SG, Nelson BL (eds) *Handbooks in operations research and management science*. Elsevier, Amsterdam, pp 129–187
3. Cai X, Goh CJ, Mees AI (1998) Greedy heuristics for rapid scheduling of trains on a single track. *IIE Trans* 30(5):481–493
4. Caprara AM, Fischetti M, Toth P (2002) Modeling and solving the train timetabling problem. *Oper Res* 50(5):851–861
5. Higgins A, Ferreira L, Kozan E (1996) Optimisation of train schedules to achieve minimum transit times and maximum reliability. In: *Proceedings of the 13th international symposium on transportation and traffic theory*, 589–614
6. Lindner TH (2000) Train schedule optimization in public rail transport. PhD. thesis, Fachbereich für Mathematik und Informatik der Technischen Universität
7. Törnquist J (2005) Computer-based decision support for railway traffic scheduling and dispatching: a review of models and algorithms. In: *Proceedings on ATMOS*
8. Kroon L, Huisman D, Maroti G (2007) Railway timetabling from an operations research perspective, Technical report EI2007-22, Economic Institute, Erasmus Universiteit Rotterdam, p 25
9. Sahin I (1999) Railway traffic control and train scheduling based on inter-train conflict management. *Transp Res Part B* 33(7):511–534
10. Higgins A A, Kozan E, Ferreira L (1996) Optimal scheduling of trains on a single line track. *Transp Res Part B* 30(2):147–161
11. Liu S, Kozan E (2009) Scheduling trains as a blocking parallel-machine job shop scheduling problem. *Comput Oper Res* 36:2840–2852
12. D'Ariano A, D'Ariano P, Samá M, Pacciarelli D. (2013) Real-time train scheduling: from theory to practice, Technical report RT-DIA-207-2013, pp 1–13, Oct 2013, Dipartimento di Ingegneria, Sez. Informatica e Automazione, RomaTre

13. Higgins AE, Kozan Ferreira L (1997) Heuristic techniques for single line train scheduling. *J Heuristics* 3:43–62
14. Babar Khan MD, Zhang, Jiang Z (2006) An intelligent search technique to train scheduling problem based on genetic algorithm. In: 2nd international IEEE conference on emerging technologies, Peshawar, Pakistan, pp 593–598, 13–14 Nov 2006
15. Sajedinejad A, Hasannayebi E, Mardani S, Mir Mohammadi KSAR, Kabirian A (2011) SIMARAIL: simulation based optimization software for scheduling railway network. In: Winter simulation conference 2011, pp 3735–3746
16. Nirmala G, Ramprasad D (2014) A genetic algorithm based railway scheduling model. *Int J Sci Res (IJSR)* 3(1):11–14
17. Deotale N, Katkar M (2014) Automated train using genetic algorithm for railway scheduling with GPS. *Int J Curr Eng Technol* 4(3):1553–1555
18. Li KP, Gao ZY, Ning B (2005) Cellular automaton model for railway traffic. *J Comput Phys* 209(1):179–192
19. Li KP, Gao ZY, Ning B (2005) Modeling the railway traffic using cellular automata model. *Int J Mod Phys C* 16:921–932
20. Ning B, Li KP, Gao ZY (2005) Modeling fixed-block railway signaling system using cellular automata model. *Int J Mod Phys C* 16(11):1793
21. Zhou HL, Gao ZY, Li KP (2006) Cellular automaton model for moving-like block system and study of train's delay propagation. *Acta Phys Sin* 55(4):1706–1710
22. Li KP, Gao ZY, Yang LX (2007) Modeling and simulation for train control system using cellular automata. *Sci China Ser E* 50(6):765–773
23. Tang T, Li KP (2007) Traffic modeling for moving-block train control system. *Commun Theor Phys* 47(4):601–606
24. Zhou H (2008) Cellular automaton model for railway transportation safety system. In: IEEE International Symposium on Knowledge Acquisition and Modeling (KAM), pp 790–793
25. Fu YP, Gao ZY, Li KP (2008) Modeling study for tracking operation of subway trains based on cellular automata. *J Transp Syst Eng Inf Technol* 8(4):89–95
26. Xun J, Ning B, Li KP (2009) Station model for rail transit system using cellular automata. *Commun Theor Phys* 51(4):595–599
27. Zhou H, Jia L, Qin Y (2009) Integrating GIS with cellular automaton model for railway transportation safety system. *WRI World Congr Comput Sci Inf Eng* 5:503–507
28. Zhang ST, Chen YCH (2011) Simulation for influence of train failure on railway traffic flow and research on train operation adjusting strategies using cellular automata. *J Phys A* 39(21):3710–3718
29. Xu Y, Cao CHX, Li MH, Luo JL (2012) Modeling and simulation for urban rail traffic problem based on cellular automata. *Commun Theor Phys* 58(6):847
30. Feng FL, Lan D (2012) Heavy-haul train's operating ratio, speed and intensity relationship for Daqin railway based on cellular automata model. *Inf Technol J* 11(1):126–133
31. Bahramian Z, Bagheri M (2013) Modeling train traffic flow using cellular automata. In: ASCE Conference Proceedings of the 2nd International Conference on Transportation Information and Safety (ICTIS), American Society of Civil Engineering (ASCE), Wuhan, China, pp 1926–1931. doi:[10.1061/9780784413036.257](https://doi.org/10.1061/9780784413036.257)
32. Hasannayebi E, Sajedinejad A, Mardani S, Mir Mohammadi AR (2012) An integrated simulation model and evolutionary algorithm for train timetabling problem with considering train stops for praying. In: Winter simulation conference, pp 1–13
33. Tormos P, Lova A, Barber F, Ingolotti L, Abril M, Salido MA (2008) A genetic algorithm for railway scheduling problems. *Stud Comput Intell (SCI)* 128:255–276
34. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, New York
35. Tolson BA, Shoemaker CA (2007) Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resour Res* 43:W01413. doi:[10.1029/2005WR004723](https://doi.org/10.1029/2005WR004723)
36. Tolson BA, Asadzadeh M, Maier HR, Zecchin A (2009) Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization. *Water Resour Res* 45:W12416. doi:[10.1029/2008WR007673](https://doi.org/10.1029/2008WR007673)
37. Behrangi A, Khakbaz B, Vrugt JA, Duan Q (2008) Comment on dynamically dimensioned search algorithm for computationally efficient watershed model calibration by Bryan A. Tolson and Christine A. Shoemaker. *Water Resour Res* 44:W12603. doi:[10.1029/2007WR006429](https://doi.org/10.1029/2007WR006429)
38. Huang X, Liao W, Lei X, Jia Y, Wang Y, Wang X, Jiang Y, Wang H (2014) Parameter optimization of distributed hydrological model with a modified dynamically dimensioned search algorithm. *Environ Modell Softw* 52:98–110
39. Benjaafar S, Dooley K, Setyawan W (1997) Cellular automata for traffic flow modeling. University of Minnesota, Minneapolis
40. Cremer M, Ludwig J (1986) A fast simulation model for traffic flow on the basis of Boolean operations. *J Math Comp Simul* 28(4):297–303
41. Nagel K, Schreckenberg MJ (1992) A cellular automaton model for freeway traffic. *Phys I* 2(12):2221–2229