Scientia Iranica A (2018) 25(2), 646–662

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions A: Civil Engineering
http://scientiairanica.sharif.edu

A simulation-based optimization approach to rescheduling train traffic in uncertain conditions during disruptions

M. Shakibayifar[a,*], A. Sheikholeslami[a], and F. Corman[b]

a. *Department of Transportation Engineering and Planning, School of Civil Engineering, Iran University of Science & Technology, Narmak, Tehran, P.O. Box 1684613114, Iran.*
b. *Department of Maritime and Transport Technology, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands.*

Received 5 May 2016; received in revised form 10 August 2016; accepted 12 November 2016

KEYWORDS
Train rescheduling;
Simulation-based optimization;
Train delays;
Dynamic priority;
Blockage.

Abstract. Delays and disruptions reduce the reliability and stability of the rail operations. Railway traffic rescheduling includes ways to manage the operations during and after the occurrence of such disturbances. In this study, we consider the simultaneous presence of large disruptions (temporary full or partial blockage of tracks) as well as stochastic variation of operations as a source of disturbance. The occurrence time of blockage and its recovery time are given. We designed a simulation-based optimization model that incorporates dynamic dispatch priority rules with the objective of minimizing the total delay time of trains. We, moreover, designed a variable neighborhood search meta-heuristic scheme for handling traffic under the limited capacity close to the blockage. The new plan includes a set of new departure times, dwell times, and train running times. We evaluated the proposed model on a set of disruption scenarios covering a large part of the Iranian rail network. The result indicates that the developed simulation-based optimization approach has substantial advantages in producing practical solution quickly, when compared to commercial optimization software. In addition, the solutions have a lower average and smaller standard deviation than the currently accepted solutions, determined by human dispatcher or by standard software packages.

1. Introduction

Railway systems are frequently characterized by high flow density and mixed traffic, which makes them sensitive to various types of disturbances [1]. Railway rescheduling deals with disturbances that create delays of some trains in the rail network. Rail transit systems seek to schedule trains in order to avoid passenger dissatisfaction and improve service reliability [2]. The impact of larger disturbances (termed disruptions) is more pervasive and can propagate easily in time and space. In this situation, there is a need to update and re-schedule train services in a short period. Train rescheduling problem is a dynamic decision-making process that involves dispatching decisions. The simplest decisions are based on the planned timetable order, or static priorities (differentiating between classes of train services); however, in general, better decisions are made based on actual data of the trains, real-time

*. *Corresponding author.*
E-mail addresses: m_shakibayi@civileng.iust.ac.ir (M. Shakibayifar); sheikh@iust.ac.ir (A. Sheikholeslami); f.corman@tudelft.nl (F. Corman)

doi: 10.24200/sci.2017.4186

information of the disturbance, as well as operational constraints. Furthermore, the exact time and location of the disturbances may not be known in advance [3]. These facts bring many difficulties in designing train dispatching actions or policies.

The present research is motivated by the situation where the recovery of the train services is of our concern. In Iranian railway network, the important reasons for train delays are infrastructure failure, accident, engine breakdown, and unpredicted weather condition. Because of the complexity and dynamic behavior of the train traffic rescheduling, simulation modelling has become an effective method to assess the effectiveness of train rescheduling strategies. Simulation models are powerful tools to support resolving path conflicts in train rescheduling problem [4]. The integrated simulation models and optimization methods are able to address the complexity of real-time train rescheduling problems [5]. We argue that the stochastic factors, pertaining to small variations, as much as large disruptions, should be instead studied in more detail. To this end, we propose a simulation modeling in combination with an optimization procedure, which solves the train-rescheduling problem under uncertain operating conditions and in case of facing large disruptions.

Simulation modeling approaches have been used extensively in transportation applications as a flexible and powerful method to evaluate the robustness and reliability of the system (see [2,6-14]). However, despite the fact that the train-rescheduling problem has been analyzed extensively, a limited research has been directed to the combination of simulation platforms with advanced search techniques to solve train-rescheduling problems under uncertainty. By using advanced and flexible simulation systems to control trains, improved management of the rail transportation will be easy. Optimization models are also trying to minimize the cost of delay, finding solutions to repair and restore the disrupted scenarios and improve traffic flow on congested bottlenecks in the rail networks. A solution has to respect railway operational rules and capacity constraints, partial or full blockage during the disruption, and minimum headway constraints. The objective is to minimize the total average delay time of trains at rail stations. The main contributions of this study lie, therefore, in the subject of microscopic disruption management under uncertain conditions. First, we develop a flexible stochastic simulation model, which we use for generating disposition schedules following principles acceptable to the local dispatchers (priorities) in a very short time. Secondly, a dynamic priority rule is proposed to accelerate the performance in terms of speed and convergence of the search algorithm. Third, a two-stage optimization method is proposed based on meta-heuristic search, which further minimizes the delays, with particular focus on the disrupted areas. We also show that the combination of the dynamic priority rule with the meta-heuristic gives particularly good results.

The remainder of the paper is organized as follows. Section 2 presents a review of models and approaches to railway traffic rescheduling. In Section 3, the problem is described in detail. Afterward, the details of the methodology are presented in Section 4. The framework of the simulation method is discussed in Sections 5. We describe a real case in Section 6 to set up a comprehensive experimental study in Section 7. Conclusive remarks will close the paper in Section 8.

2. Literature review

The train rescheduling problem is known to be strongly NP-hard [15]. The management of train timetable is a complex procedure subject to the capacity and resource constraints [16]. This problem belongs to a wide-range class of combinatorial optimization models and methods being called railway disruption management. Railway disruption management mainly refers to the models and approaches used in the railway real-time traffic management [17]. A variety of approaches have been proposed, ranging from mathematical optimization (mixed-integer linear programs) to simulation techniques, heuristic and meta-heuristic methods. All those methods have shown their value in practice to evaluate the stability of the disturbance recovery strategies, or to generate near-to-optimal solutions in a reasonable computation time. Coverage surveys of railway disturbance management practice and theory can be found in [18-21]. In what follows, we discuss the most related contributions in this area of the research, according to the general structure proposed in this latter survey paper.

Cheng [22] proposed a new integrated approach of a knowledge-based system with an operation research technique to solve train rescheduling problems. The critical path method was used to find near-to-optimum solutions. In order to reach a global optimum, a feedback control function was designed to manage the delay and resolve the resource conflicts. The problem of controlling and coordinating rail traffic in a whole railway network is too hard to tackle in a reasonable time. Higgins et al. [23] applied a local search heuristic with an improved neighborhood structure, genetic algorithms, tabu search and two hybrid algorithms for train-scheduling problem. The computational result indicates that both hybrid algorithms provide better results compared with the other heuristics. A decision support system called ROMA was designed and implemented by D'Ariano [24] based on Alternative Graph (AG) techniques to cope with real-time train rescheduling problem with multiple delays more efficiently. The aim was to improve punctuality through better utilization of the railway infrastructure. The applicability of the ROMA was verified through extensive computational tests on instances of the Dutch railways. ROMA system was first implemented to optimize railway traffic within a single dispatching area. The system was extended by Corman et al. [25] to present an innovative distributed approach to manage train movements more effectively in a multi-area dispatching setting. The performance of the distributed approach was compared with the existing models in terms of computation time and reduction of total delay.

Corman et al. [26] proposed a novel approach to deal with multiple train classes in train rescheduling problem. An efficient scheduling procedure was adopted in order to generate feasible train timetables according to a set of priority classes. In each step, an advanced branch and bound algorithm was used to solve the sub-problems optimality. Dündar and Şahin [27] designed a decision support system using Genetic Algorithms (GAs) and Artificial Neural Networks (ANNs) for real-time conflict resolution problem. The methodology was tested with actual data extracted from train operations in Turkish State Railways. Hassannayebi and Kiaynfar [28] proposed three meta-heuristic algorithms based on Greedy Randomized Adaptive Search Procedure (GRASP) for finding a near-optimal train timetable in double-track railway lines. The output results show the effectiveness of the proposed meta-heuristic algorithm in solving large-sized instances of the train-timetabling problem. Dollevoet et al. [29] proposed an optimization method that solves a macroscopic delay management problem as well as a microscopic train scheduling model. The headway constraints were captured in the model with full details of the railway infrastructure, especially within the stations. The resulting disposition timetable was evaluated thoroughly for a bottleneck segment of the rail network. Some studies integrate the concept of priorities, easily understood and accepted by practitioners. Hassannayebi and Zegordi [30] proposed variable and adaptive local search algorithms to minimize the total and maximum waiting time of the passengers for urban rail transit systems. Narayanaswami and Rangaraj [3] designed a multi-agent system model with a learning mechanism for real-time train rescheduling in a bi-directional railway traffic on a single-track route. The developed framework employs a dynamic scheme of priority assignment procedure that allows for dynamically dispatching the disturbed trains in real-time and constructs a deadlock-free disposition schedule. Hassannayebi and Zegordi [31] proposed linear and nonlinear mathematical models for train scheduling problem. In order to tackle large-sized instances of the problem, variable neighborhood search approaches were designed. The efficiency of the meta-heuristic algorithms was verified through its application to the Tehran metropolitan network.

The topic of railway rescheduling has attracted attention mostly concerning small delays, while the study of large disruptions and inclusion of many stochastic factors have been limited so far. The dynamic changes over time, in those situations, are quite strong; there needs to be an inherently dynamic environment, proposing adjustments such as rescheduling and partial reordering during operations, which is currently to be found in simulation environments. Therefore, despite the interesting scientific results reached by optimization models, there is a need to develop flexible simulation systems able to evaluate different partial reordering possibilities. Furthermore, the trade-off between delivered schedule quality and the rescheduling process time is of critical importance in the practical implementation of a train-rescheduling tool. On the other hand, even though several simulation models have been developed for rail operation management formerly, an acceptable solution with regard to inclusion of optimized asynchronous choices has not been attained in this aspect. To the best of our knowledge, a direct application of the flexible simulation-based optimization approaches to train rescheduling problem has not been found in the literature; if so, it has not been addressed to the same extent that accounted in the present study.

With particular regards to uncertain effects of small delays and large disruptions, we merge the descriptive power of stochastic simulations with the easily-accepted priority-based scheduling for disruption management. We present an advanced discrete-event object-oriented simulation model, implemented on a commercial event-driven simulation package. In order to optimize the performance measures, a variable neighborhood search technique is proposed to improve solutions under the strong capacity limitations due to the disruption. The developed simulation-based optimization approach has the flexibility of adjusting train operations under time and resource constraints in an efficient way.

3. Problem statement and formulation

This section provides the problem statement and the notations used for the train-rescheduling model. The main assumptions and characteristics of the problem are given in Table 1. It is assumed that an initial timetable for trains on the network or by the simulation model presented in this study is given. At the starting moment of the disruption, the trains are in a position, considered known, and set as data inputs for the simulation model. The considered rail infrastructure is illustrated in Figure 1. It includes a

Table 1. Assumptions and characteristics of the problem.

Assumptions	Description
Rail infrastructure	A corridor with single or double-track segments
Disruption	Temporary partial or full blockage
Disturbance management strategies	The sequence of trains at the disruption site and elsewhere is changed Overtaking is permitted at stations with available capacity
Train types	Passenger railway
Priority of trains	Different and variable over time
Travel time function per train	Probabilistic
Rail capacity at stations	Limited (taking into account the number of tracks and platforms)
Signaling systems	Absolute fixed block signaling system between stations

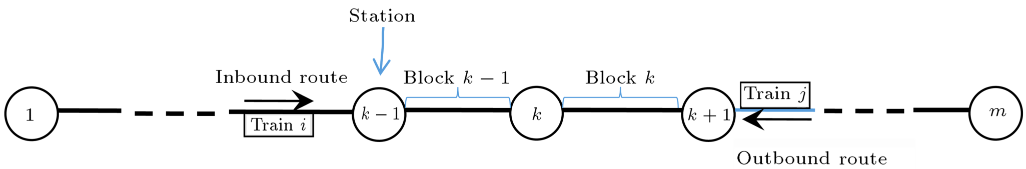

Figure 1. The considered rail infrastructure.

set of stations ($k = 1, 2, \cdots, m$) and a set of operating trains ($i = 1, 2, \cdots, n$). The segments between each pair of stations are single/double-track block sections. We consider absolute fixed block operations between stations by which block sections begin and end only at stations. Only one train is allowed on a track between two stations. An overtaking operation is allowed only at stations.

During the normal operations, a train can move from the current station if the successor block section is available and there is at least one free track segment in the next station (absolute fixed block operations between stations). When a disruption occurs in a block section, the train traffic is heavily perturbed. At that moment, the main goal is to provide a new disposition schedule for all operating trains at the end of the scheduling horizon, so that the total delay cost is minimized. The model proposed in this study produces new disposition schedule from a combination of the following actions (in the vicinity of the disruption, or elsewhere): reordering (changing the sequence of trains on the block sections), adjusting the departure times, and changing the stop times at stations.

A main constraint is that at stations, a train is not permitted to depart, in any case, before its scheduled departure time. A conflict happens when at least two trains request to use the same block section at the same time. In this case, a conflict resolution procedure is required to decide on the ordering of the trains. This procedure aims locally (or globally) to decrease the total delay of the trains. The total delay is defined as the difference between the actual train arrival time and the scheduled time at a set of predefined stations in the network. Total delay of a train consists of two parts termed as initial delay and secondary delay. The initial delay is triggered by disruptions and longer travelling time and cannot be recovered by rescheduling model. The secondary delay is the extra delay needed to resolve the potential conflicts during a planning time horizon. In this study, the train-rescheduling model considers dynamic priority of trains during disturbed operations in order to minimize the total delay of the train services. The next section provides the assumptions made on the train operations during both normal and degraded modes. Before the disturbance occurs (normal condition), the railway capacity utilization is at the regular level. In the first state (normal-to-disrupted situation), the utilization necessities to be reduced to achieve a utilization level that can be reserved during the disturbance. During the second transition state (disrupted situation), the disruption recovery actions should start. In this state, the new timetable is functioned and the utilization level is steady. The third state (disrupted-to-normal) changes the utilization level to the normal condition. In our research, the focus is on the second and third transition states.

3.1. Train operation modeling during normal operation

This section provides the assumptions made on the train operation during the simulation experiments.

The stochastic parameters here are train-running times on block segments. We consider a stochastic simulation model to account for the inherent and relevant probability distribution of the running time. In this regard, a stochastic distribution of train running times is estimated and used instead of the commonly considered deterministic running time. The normal distribution, thus, proved to be a good model for the large array of phenomena, which can be found in real-life operations [32]. The probabilistic train-running time distributions are fitted at a given level of significance (95%). From the statistical analysis, we conclude a good fit of the experimental data with the normal distribution. All the data sets show that the running times between consecutive stations fit the normal distribution at the level of significance of 0.05 according to Kolmogorov-Smirnov (KS) statistical tests. Thus, the hypothesis of normal distribution is not rejected at the desired level of significance. However, in order to make the running time distribution more practical, we truncated the travel time function using the maximum train speed.

In order to formulate the running time function, we consider distances between any consecutive stations k and $k+1(L_k)$. Let μ and σ be the mean and standard deviation of the running time distribution. We assume that the running time of train i between consecutive stations k and $k+1(t_{ik})$ follows a normal distribution with average $\mu = \frac{L_k}{V_i^{\text{ave}}}$ and variance σ^2 (minutes^2), where V_i^{ave} and $V_i^{\max}$ are defined as average and maximum speeds of train i, respectively. The variance can be determined through sampling methods. It should be noted that the running times between stations cannot be less than the minimum technically feasible. Thus, to ensure that all trains cannot exceed their maximum technically speed, the minimum running time ($\frac{L_k}{V_i^{\max}}$) is defined in Eq. (1). A similar running time function was proposed by Nie and Hansen [33]:

$$t_{ik} = \max\left\{\text{Normal}\left(\frac{L_k}{V_i^{\text{ave}}}, \sigma^2\right), \frac{L_k}{V_i^{\max}}\right\} \qquad \forall\ i, k. \tag{1}$$

3.2. Train operations during disruptions

An infrastructure failure occurring in the route is termed disruption, or degraded mode. Without loss of generality, consider a single-track segment of a railroad line between two major intersections as shown in Figure 2. The dispatching rules on this single-track segment manage the movement of trains for both directions. Different dispatching policies will cause different amounts of delays for trains. In this research, the length of the train is not considered. The effect of train length on train delay would be insignificant if the distance of the track segment is much lengthier than the length of the train. It is assumed that the disruption occurs at one block section and degrades the traffic in partial or in full. According to railway safety rules, no more than one train at a time is permitted to dwell in any block section (referring to the conflict-free situation). In this article, we focus on two frequent degraded modes in railway systems. In the first degraded mode (full blockage), the normal operation of a single-track block section is disrupted due to an incident between two neighboring stations as illustrated in Figure 2. In this figure, the traffic flow under normal condition and the location of service disturbance are depicted. As can be seen, there is no possibility of passing during the disturbance. However, during the normal condition, the traffic between two consecutive stations is bi-directional. After recovery of the disturbance, trains start normal operation on the single-track segment. According to the accepted operational rules in Iran, the allowed control actions in this case consist of retiming or re-routing the incoming trains toward the disturbance location, while there is no possibility for cancelling or short-turning trains. Thus, in the first degraded mode, the main decision variables are which train should be reordered, or delayed at what locations.

The second degraded mode is a blockage of one track out of a double-track block segment (Figure 3). In this situation, trains moving toward the disrupted area can bypass the blockage and after traversing a number of switch points, they go back to the original route. The reordering policy (as illustrated in Figure 3) enables the waiting train to switch to the bypass direction track. Crossover tracks allow trains to be transferred from one track to another, enabling trains to bypass the incident location. During the disruption, the system effectively becomes a single-track between two consecutive switch points, and the trains requesting to pass through this part of route wait at stations until the single line is suitable for their trips. After the blocked track is repaired (which might take a long time), the system again becomes a two-parallel-track line, and traffic flow returns to normal. The conflict resolution of train on the single-track segment involves sequencing the

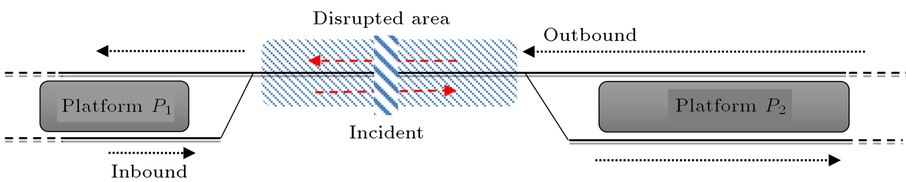

Figure 2. The line blockage in the degraded mode #1 .

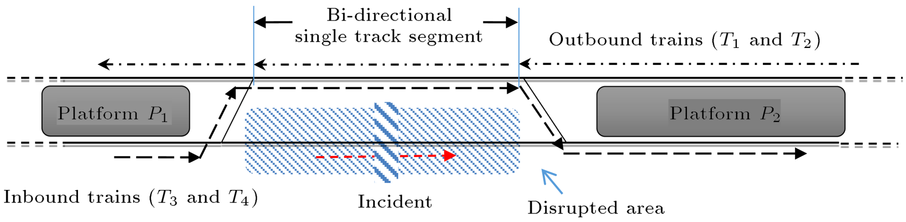

Figure 3. Train services on the single-track segment during degraded mode #2.

inbound and outbound trains. Thus, the optimization model aims to find the best crossing order of the waiting trains and the related times of operations. We focus particularly on finding the order of crossing the disrupted area, as we experimentally found out that it has a major influence. To this end, we define meta-heuristic procedures, explained in the remainder of the paper.

4. An object-oriented event-driven simulation framework for train rescheduling

Discrete event simulation systems are extensively used for modeling the behavior of a complex dynamic system within a discrete time framework based on an event list. An event indicates the occurrence of a change in the status of the rail system at a specific time. Different modeling approaches, e.g. process-oriented [9,34] and object-oriented [35,36], can be used for railway systems. The object-oriented simulation model provides a flexible build-in framework that supports the design process of railway network layout. In the present study, we use Enterprise Dynamics (ED) simulation software as a simulation platform due to its capability of designing customized rail objects and ability to implement optimization algorithms. Enterprise Dynamics is a leading object-oriented simulation platform to design and implement simulation models [37]. It has also a built-in programming language called 4DScript, which can be used for advanced modeling purposes. Another application of the 4DScript is the capability of programming, which allows us to include the meta-heuristic optimization approaches right into the simulation system [38].

The main procedure of the simulation optimization of the train dispatching is presented in Table 2. When a train enters a waiting queue, a dispatching algorithm is applied to check the operational and safety

Table 2. The main procedure of the simulation-optimization.

Step 1: Initialize necessary simulation parameters
(Confidence level $1-\alpha$, number of replications)
Set simulation clock ($t = 0$)
Initialize system state
Prepare event list (ascending order of time)
Step 2: Perform several simulation runs using dynamic dispatching rules
Step 3: Use a look-ahead procedure to find out a new event (either stop at the current station or move to the next station)
Step 4: Update the priority of the train according to the accumulated delay when it arrives at a station
Step 5: Aggregate simulation results and store them in the database
while stopping criteria is not met **do**
    Load aggregated parameters into simulation system
    Solve the optimization model using VNS
    Write new decision variables corresponded to the re-ordering and adjusted priority to the database
    Load new decision rules into simulation model
    Perform several simulation runs to evaluate the solution
    Aggregate simulation results and store them in the database
end-while
    Load the best found solutions
    Generate output reports
    - New rescheduled train graphs (time-station diagrams)
    - Estimate the expected value and the variance of the train delays at destinations.

constraints. If all conditions are satisfied, then the train is allowed to move to the successor block section according to its route. Else, an event is created to execute dispatching algorithm and the involved train waits in the current position until all conditions are met. The conditions are the available free track on the next segment and a free platform for the train that has a stopping plan at the next station.

5. Simulation-based optimization approach

The proposed two-stage simulation-based optimization framework for train rescheduling problem is illustrated in Figure 4. As can be seen, it follows a kind of black-box approach to tackle large and complex simulation-based optimization problems. We consider solutions based on either the static timetable order or priority based. Furthermore, the priorities can be static or dynamic. The latter is generally more flexible and can deliver better solutions with regard to delays. Concerning dynamic priorities, we restrict ourselves to dispatching algorithms that calculate and assign priorities based on time-based parameters. Our procedure works in two stages. In the first stage, the initial random solutions are generated based on heuristic dispatching rules based on dynamic priority. At this point, the goal is to reach a relatively good new disposition schedule in order to handle the disrupted traffic conditions on the route in a short period of time. In the second stage ,the initial generated schedules are evaluated by their total average delays (considered a reliability index) via simulations' experiments. The best solution from this stage is regarded as a starting point for further optimization. A meta-heuristic algorithm (variable neighborhood search) is used in order to improve the solution to the train rescheduling problem algorithm in terms of mean and variance of delay times. During stage 1, due to time constraints, the number of simulation replication must be determined carefully for solving train-rescheduling problem. The updated values of the decision variables refer to control actions such as retiming and reordering of train at rail segments. At each iterative step of the meta-heuristic optimization algorithm, a set of new departure times is decided, and a new tentative schedule is tested by the simulation model. Given a train scheduling solution, the simulation model obtains statistical bounds on the objective value, and the optimization model iteratively improves the expected value until the time limit is reached. At the end of each simulation run, the current solution is evaluated based on the quality and reliability criteria. The process continues to achieve a desired response plan with respect to time constraint. The objective function is defined as the total average delay of train services at all visiting stations.

5.1. *Dynamic priority rule-based heuristic*

This sub-section gives the explanation regarding how to resolve the conflicts and generate good-quality initial rescheduled plans rapidly by means of dynamic priority scheduling. For this purpose, heuristic dispatching rules are proposed that change the priorities of the operating trains dynamically in order to reduce the secondary delays. As we observe, train dispatchers in different railway companies mostly perform a preference-based process of conflict detection and resolution. The traditional dispatching rules do not take into account the updated information of the trains and may fail to find appropriate solutions. Consequently, it is essential to recognize the decision-making process in order to develop innovative conflict resolution models. The application of the dynamic priority rule-based model presented in this study seems to be an effective method to meet this objective. In what follows, we explain our method of train conflict resolution. As mentioned earlier, an initial static priority class is assigned to each train before the implementation of the rescheduling procedure. In the simulation experiments, the static priority class is then updated in order to further adjust the importance and urgency of the train service to-

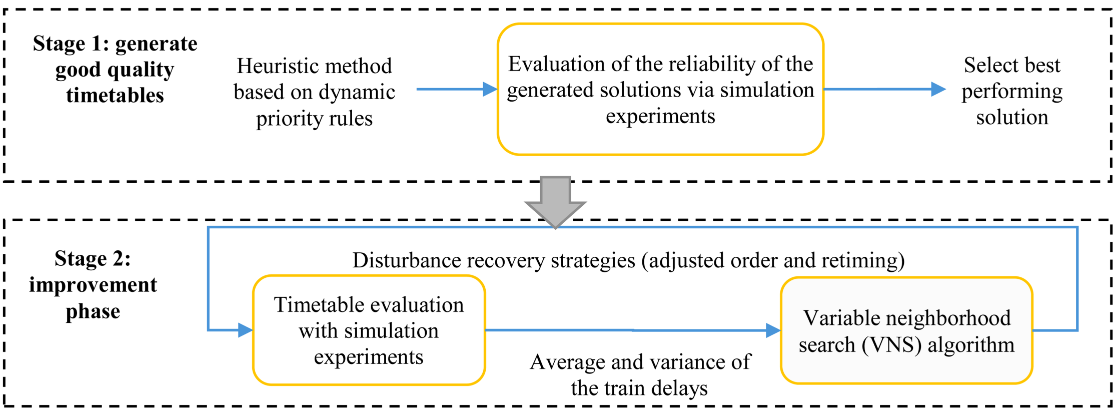

Figure 4. The proposed two-stage simulation-based optimization framework for train rescheduling problem.

Table 3. Notations used in dynamic priority rule-based heuristic method.

Symbol	Definition
p_i	The initial priority of train i
f_i	The accumulated delay of train i
a_i	The allowed (maximum) delay of train i
Δ_i	The surplus delay regarding the maximum allowed delay
U	The utility function
a_{0i}, a_{1i}	The start and end break points of the utility function
L	The total number of train conflicts in the schedule

wards the objective value. Our dynamic priority-based train rescheduling is a type of scheduling algorithm in which the train priorities are computed during the execution of the simulation model. The main goal of dynamic scheduling is to adjust to dynamically dispatching order and design a good quality solution in an adaptive approach. The proposed framework is an iterative probabilistic procedure for determining the dispatching priorities. The dynamic priority of a specific train is calculated as a function of the initial priority class, the actual (accumulated) delay, and the allowed (maximum) delay time. Notations of dynamic priority rule-based heuristic are summarized in Table 3. Every time a conflict happens, one train should be delayed to resolve the conflict. The resolving of the conflicts is mainly based on train priorities. The proposed dynamic priority rule preserves the overall delay of the trains and resolves the train conflicts in a short period.

The initial values of the train priorities are based on the train classes. The adjusted priorities impose additional challenges on the optimization problem so that the priority of a particular train may change several times during a single simulation run. In the first step of the heuristic method, a new conflict-free disposition schedule is constructed through a conflict resolution model by the adjusted priorities. An initial train schedule is represented by a set of potential conflicts (C). C_j represents the jth conflict as it occurs in time. Train conflicts are resolved according to the relative priority ratio chronologically. The adjusted priorities are calculated according to the calculated values of accumulative delay of trains. A piecewise linear utility function (U) is used to determine the adjusted train priority in terms of the deviation from the allowed delay. The train priorities update whenever they depart from or arrive at stations according to Eq. (3). $U(x)$ consists of K linear pieces joined together at breakpoints $0 \leq d_i \leq \infty$. The priority of the trains increases when the accumulated delay exceeds the allowed delay. Using the initial set of train weights or initial priority, it is possible to calculate dynamic priority, p_i, of any train, i; a conflict is resolved according to their value with a higher value resulting in a higher priority to reserve the path. In the second step, the objective function of the generated solution is measured by the reliability criteria. Once the limit is reached, the heuristic algorithm terminates and provides the best solution found by the heuristic method. We consider a time limit of 10 minutes to execute the algorithms:

$$C = \{C_1, C_2, \cdots, C_L\}, \tag{2}$$

$$p_i \leftarrow p_i + U(\max\{f_i - a_i, 0\}) = p_i + U(\max\{\Delta_i, 0\}), \tag{3}$$

$$U(x) = a_{0k} + a_{1k}x, \tag{4}$$

$$d_{k-1} \leq x \leq d_k, \qquad k = 1, 2, \cdots, K. \tag{5}$$

5.2. Variable neighborhood search algorithm

In this section, we explain the proposed variable neighborhood search algorithm. This is specifically introduced to deal with the strong shortage of capacity near the blockage, which asks for more sophisticated optimization approaches. The Variable Neighborhood Search implemented here fixes the full order for trains going on the disrupted area, possibly overriding (dynamic) priorities. In the improvement stage of the proposed two-stage simulation-based optimization approach, a local search algorithm is required to perform a sequence of local moves in neighborhood $N(x)$ of initial solution x to improve the performance value until a local optimum solution (x') is obtained. The basic function of the local search algorithm can be improved in order to avoid trapping in local optima. One of the most practical extensions of the local search is Variable Neighborhood Search (VNS). In this method, the systematic changes in neighborhood structure are performed to escape from the local optima. VNS method was originally introduced by Mladenović and Hansen [39]; after that, it received increasing attention both in theoretic extension and large-scale optimization problems [40]. VNS has been applied to a wide-range of combinatorial optimization problems including capacitated vehicle routing problems [41].

The notations are given in Table 4 to better explain the method. Moreover, the pseudo code of the proposed variable neighborhood (a Variable

Table 4. Notations of the proposed VND method.

Symbol	Definition
k	The counter of the neighborhood structures ($k = 1, 2, \cdots, k_{\max}$)
$k_{\max}$	The total number of neighborhood structures
x	The candidate solution for local search
x_o	The initial solution
x_{best}	The best incumbent solution
$N_k(x)$	The set of solutions in the kth neighborhood structures
$F(x)$	The fitness function
R_p	The number of simulation replications to evaluate the objective function
i	The index of algorithm iteration
$\text{Iter}_{\max}$	The maximum algorithm iterations

Table 5. Pseudo code of the proposed variable neighborhood search method for train rescheduling.

1	*Input* $(k_{\max}, x_o, \text{Iter}_{\max}, R_p)$
2	$x := x_o$;
3	$i := 1$;
4	$k := 1$;
5	**While** $i \leq \text{Iter}_{\max}$ **do**
6	**While** $k <= k_{\max}$ **do**
7	$x' :=$ **BestImprovement** (x, k);
8	$F(x') :=$ **simulation** (R_p, x');
9	$i := i + 1$;
10	**If** $F(x') < F(x)$ **then**
11	$x_{\text{best}} := x'$;
12	$k := 1$;
13	**Else**
14	$k := k + 1$;
15	**EndWhile**
16	$k := 1$;
17	**EndWhile**
18	**Return** (x_{best});
19	End.

Neighborhood Descent (VND)) method, in the terminology of Mladenović and Hansen [39], is provided in Table 5. In our implementation, the search method changes the neighborhoods in a deterministic way. The VNS algorithm starts with the best-found solutions from the first stage optimization. VNS algorithms start with iteratively changing the properties of an incumbent solution. A neighborhood search heuristic is performed by picking initial solution, xo, determining a search direction of descent from this solution within a neighborhood, $N(x)$, and continuing to the minimum of $F(x)$ within neighborhood space, $N(x)$. In Step 7, neighborhood $N(x)$ of x is explored completely. The highest direction of descent is related to be the best

Table 6. Pseudo code of best improvement (highest descent) heuristic.

Function BestImprovement ($\boldsymbol{x}$)
1 **repeat**
2    $x' \leftarrow x$
3    $x \leftarrow$ **argmin**$\{F(y)\}, y \in N(x)$
4 **until** $(F(x) \geq F(x'))$
5 **return** $\boldsymbol{x}$

improvement that is summarized in Table 6. The process of a move from a basic solution to a possibly better one is guided by the evaluation of the fitness function value. Since the problem considered in this study has a stochastic nature, each potential solution of the problem is evaluated using the discrete-event simulation model. In Step 8, the simulation function is employed to evaluate each solution like x. The result is the average fitness value of the solution that is stored in $F(x)$.

In what follows, the neighborhood structures proposed in this paper are described in detail. A move alters the current solution to the neighboring one by shifting the relative order of some trains. We propose a combined remove-insertion with a variable step-size mechanism to alter the order of trains. In order to explain better the way that the search method performs, we provide an illustrative example. In this example, six trains approach the disrupted location. We report them along a time axis (horizontal) and space axis (vertical), with two stations; trains can overtake each other only at stations. Thus, each train $(a - f)$ is a line crossing the diagram from top to bottom, or vice versa. The blockage has occurred on a single-track segment between stations k and $k + 1$ as illustrated with a shaded rectangular block in time and space. Let $\{a, b, c, d, e, f\}$ be the initial order of train incoming to the disturbance location. Each of them has an origin and an initial priority (P_0). If there is no change in the train orders, then the resulting time-

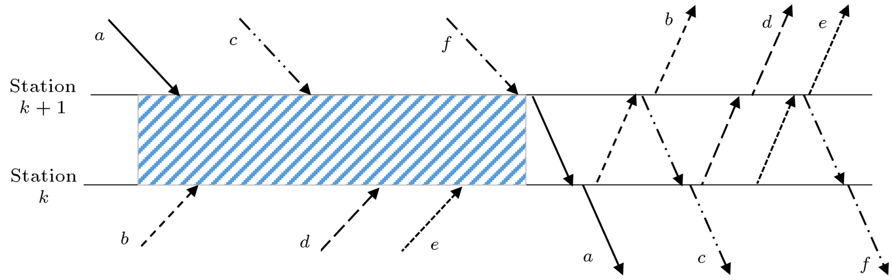

Figure 5. The disposition schedule based on the static train order.

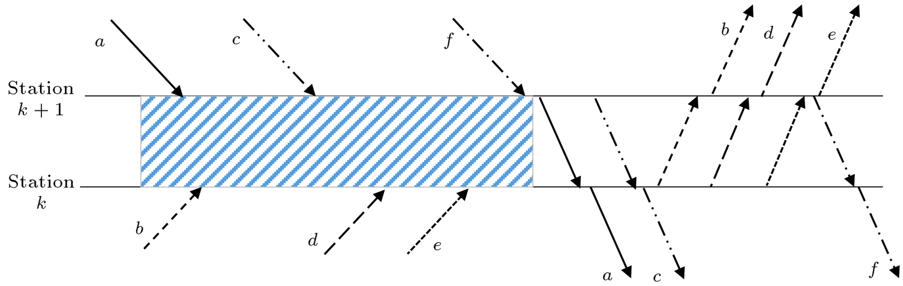

Figure 6. The disposition schedule associated to one neighbor with $k = 1$ (train b).

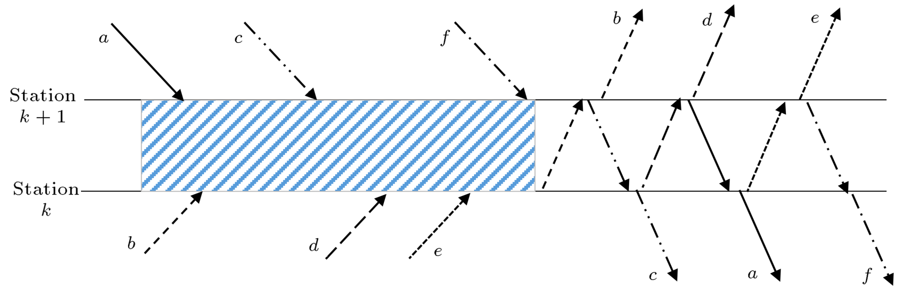

Figure 7. The disposition schedule associated to one neighbor with $k = 3$ (train a).

station graph is illustrated in Figure 5. In this graph, the train departs with the same order and follows the First-Come, First-Serve (FCFS) dispatching rule.

Now, consider the case that train order changes by insertion operators. The kth neighborhood structure is defined as the neighbors described by k shifts performed in the train orders. For example, the first neighborhood structure only performs one shift in the sequence. In this case, a single train is selected randomly, removed from the sequence, and inserted either immediately before or immediately after its original position. For example, an adjusted sequence using one-step move operator is $\{a, c, b, d, e, f\}$ where train b is selected and inserted after train c (Figure 6). Another example is to perform a three-step ($k = 3$) move. Assume that train a is removed from the sequence and inserted after train d. The adjusted train order is $\{b, c, d, a, e, f\}$ as illustrated in Figure 7. As can be seen, train a faces with the highest delay after the order adjustment.

In the cases presented in Figures 5 to 7, each station (between segments k and $k+1$) hosts three trains. In longer blockage periods, station capacities may be insufficient to host all the visiting trains. We explain how simulation model deals with these occasions. In case of higher traffic volume, the trains must wait at the former stations to be ready to dispatch when the next station has a free track (or a free platform in case of passenger load/unloading). The main objective of applying the above-mentioned moves is the attempt to recover the train schedule in terms of reducing total average delay time. In this regard, the validity of the proposed solution method is demonstrated in the next section.

6. Test case description: Tehran-Razi corridor

6.1. Infrastructure

Tehran-Razi corridor is considered as a case study. This corridor is one of the most congested ones in Iran. The infrastructure considered is a main part of the railway network in the west of the Iran. As presented in Figure 8, the network is composed of three major stations with dense traffic: Tehran, Tabriz, and Razi.

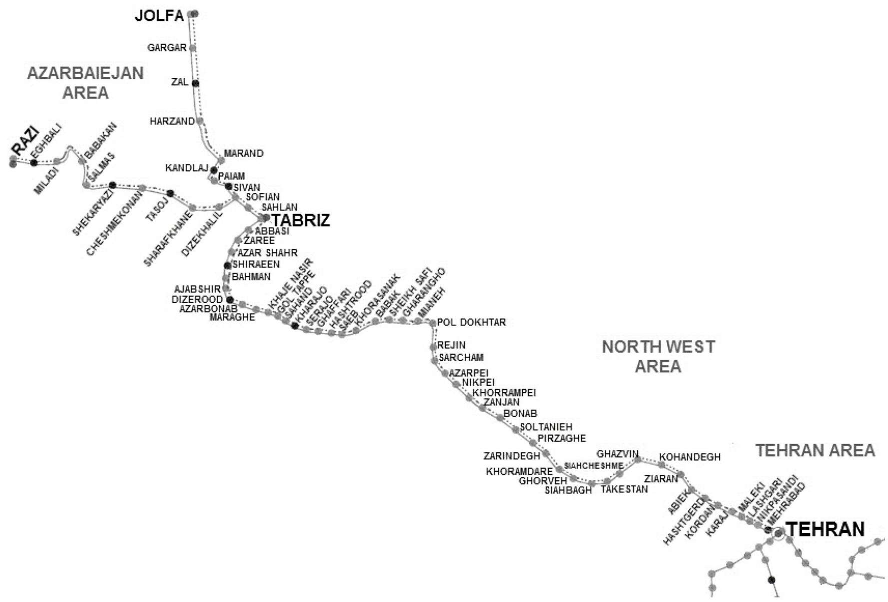

Figure 8. The considered Iranian railway network (Tehran-Tabriz-Razi corridor).

Table 7. The priority classes of static priority set S.

Priority class (w_i)	**Train type description**	**Number of trains**
1	Local	12
2	Intercity	16
3	Express	18

Other intermediate stations in the network are also considered in our simulation model. The line between Tehran and Razi (eastbound route) serves the two main traffic directions to Tehran (westbound route). The network comprises a combination of single and double tracks of different length, with a maximum distance between two end stations of about 930 km. Tehran-Razi corridor consists of 62 stations, 57 single-track blocks, and 4 double-track block sections. The total number of daily operating trains is 46. Overall, there are more than 202 track segment and 90 platforms (i.e., track segments and platforms are actually used at stations). The networks operate based on absolute block operations, i.e. only one train is allowed at all times in the segment between two stations for each track. The predefined classes of the static priority set are given in Table 7.

6.2. Disruptions scenarios

Due to the unknown nature of the disruptions, different possibilities for the start time and location of the disturbance are probable. We consider the most effective disruption scenarios, which start before the most congested period of the day, in the bottlenecks with the highest traffic. Formally, we identified those locations and times by computing a Congestion Factor (CF) as the number of train conflicts in a schedule during a specified interval. Because of the high density of traffic through bottleneck segments, the buffer times added in the initial timetables are deficient to absorb train delays caused by unpredicted disruptions. According to the above explanation, the identified test cases are summarized in Table 8. The disruption scenarios are characterized by the location, type of the degraded mode, the expected duration, and hours of the blockage.

7. Computational results

7.1. Performance

This section provides the computational results on the simulation model and the meta-heuristic technique proposed in this study. The simulation runs are executed via Enterprise Dynamics 8.2.5. All the experiments are performed on an Intel(R) Core2 Due personal computer with 3.3 GHz and 4 GB of RAM.

We report the performance of the proposed VNS method compared with those obtained from the simulation-optimization methods embedded in OptQuest package. OptQuest is a well-known registered optimization solution of OptTek Systems, Inc. (available in www.opttek.com). OptQuest works iteratively using a black-box approach as a general-purpose optimizer that performs a series of simulation experiments to find optimal or near-optimal solutions. OptQuest utilizes a mix of meta-heuristics algorithms, including Scatter Search (SS), Genetic Algorithm (GA), Tabu Search (TS), and neural network learning algorithms, to find the global optimum [42]. In the present study, OptQuest is employed which takes advantage of the decision-support features of the Enterprise Dynamics

Table 8. The disruption scenarios and the associated specifications.

Disruption scenario #	Location	Degraded mode	Expected duration (hour)	Blockage interval
1	Qazvin-Kohandezh	Full blockage	1	7:00-8:00
2	Qazvin-Kohandezh	Full blockage	2	7:00-9:00
3	Qazvin-Kohandezh	Full blockage	3	7:00-10:00
4	Zanjan-Khorram pey	Full blockage	1	2:00-3:00
5	Zanjan-Khorram pey	Full blockage	2	2:00-4:00
6	Zanjan-Khorram pey	Full blockage	3	2:00-5:00
7	Karaj-Rabet	One out of two tracks	2	7:00-9:00
8	Karaj-Rabet-Aprin	One out of two tracks	3	6:00-9:00
9	Abyek-Ziaran	Full blockage	1	20:00-21:00
10	Abyek-Ziaran	Full blockage	2	20:00-22:00

Table 9. Result of the proposed two-stage simulation-based optimization method versus FCFS and OptQuest solver.

Disruption scenario	Total average delay (hours)			Total traveling time (hours)		
	FCFS*	VNS with dynamic priority	OptQuest	FCFS	VNS with dynamic priority	OptQuest
1	43.01	27.16	32.72	391.43	388.71	393.91
2	54.32	38.83	42.69	398.19	392.95	397.23
3	65.83	42.34	47.51	410.58	394.70	398.02
4	36.38	29.86	33.62	370.42	365.52	370.46
5	54.73	45.02	47.09	353.62	348.04	358.90
6	76.17	62.30	71.77	382.66	380.17	383.47
7	37.05	24.33	29.02	394.98	385.46	391.06
8	44.00	23.75	32.55	396.45	389.13	386.24
9	39.23	28.31	31.76	391.61	390.99	394.12
10	66.54	44.41	49.82	381.83	394.06	399.77
Average	**51.73**	**36.63**	**41.86**	**387.18**	**382.97**	**387.32**

simulation software with the use of global optimization algorithms. The experimental results of the proposed two-stage simulation-based optimization method and OptQuest solver are given in Table 9. In this table, two performance metrics are calculated:

1. Total average travelling time;
2. Total average delay.

The cumulative delay time is defined as the sum of total delays at all relevant locations. It measures for all train the positive arrival time with the due date employed to measure the train delay with respect to the time at which the operation is planned, i.e. the arrival of a train at a planned stop or its planned exit from a relevant point. The best-found solutions with 10 minutes of execution are reported in this table. As can be seen in this graph, disruption scenario #6 has the most disruptive impact on the performance measure compared with the other disruption scenarios. To compare the computation efficiencies of the VNS and OptQuest, we recorded the computation time of the two methods when solving the train-rescheduling problem. Both the solution quality and computational performance are compared to the OptQuest. According to the data in Table 9, the total average delay time of all trains at all stops is reduced by almost 12.48% and 29.18% compared with the FCFS and the OptQuest, respectively. Furthermore, the total average travelling time of all trains is decreased by nearly 1.12% and 1.09% compared with FCFS and OptQuest, correspondingly. It should be noted that the total travelling time averaged over all test instances is nearly similar for VNS and OptQuest as well as FCFS.

As mentioned earlier, a two-stage simulation-based optimization method was proposed that incor-

Table 10. Computational results of two-stage VNS versus single-stage and static priority-driven scheduling methods.

Disruption Scenario	Single-stage: VNS+planned timetable order	Two-stage: VNS+orders updated on static priority	Two-stage: VNS with orders updated by dynamic priority
1	29.71	30.98	27.53
2	44.90	43.82	39.32
3	45.12	48.66	42.72
4	32.84	33.29	30.32
5	47.31	45.48	45.20
6	67.38	69.01	63.05
7	26.35	25.58	24.40
8	28.98	26.39	24.17
9	29.34	31.18	28.73
10	47.75	46.36	44.85

porates a dynamic priority rule-based algorithm to find the initial solutions. In what follows, we analyze the performance of the VNS without heuristic method, i.e. considering as initial solution the train order specified offline in the original timetable. This approach is being called single-stage VNS throughout the article. Furthermore, it is worth showing the significance of our contribution by testing the performance of two-stage VNS, i.e. with a starting solution optimized. In this case, we refer to a static priority-driven scheduling method against the proposed dynamic priority scheduling approach. The computational results of two-stage VNS versus single-stage VNS and a static priority-driven scheduling method are given in Table 10. This table provides the objective value (the expected value of the total average delay in hours) of the best-found solutions obtained for each algorithm within 10 minutes. The outcomes indicate that the average improvement gained by the dynamic priority method is about 7.6% compared with the single-stage VNS and the two-stage VNS with static priority. As can be seen, the two-stage VNS outperforms the single-stage VNS and static priority-driven scheduling methods in all disruption scenarios. This finding verifies the applicability and improvement of the dynamic scheduling method over the single-stage VNS and static priority-driven train rescheduling.

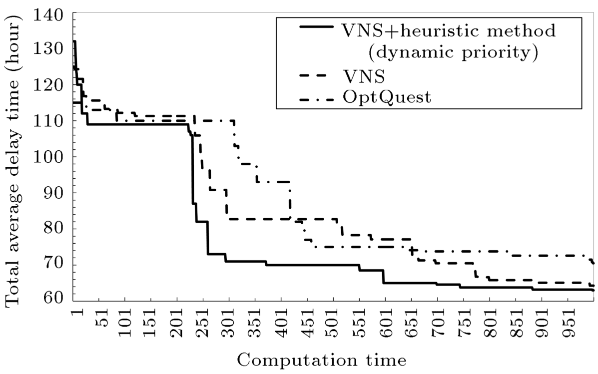

Figure 9. The search profiles of the two-stage VNS, single-stage VNS, and OptQuest (scenario #6).

7.2. Convergence analysis

In case of an unexpected disruption, it is vital that dispatchers speedily provide a good solution in order to reduce the annoyance for the travelers. Thus, a faster convergence rate results in a better and more realistic solution to the real-time train-rescheduling problem. We provide the plot of solution quality against time, which accounts for the convergence analysis of the proposed as well as benchmark algorithms. The search profiles of the VNS and OptQuest, which are both iterative procedures, are illustrated in Figure 9. The graph shows the best objective value over search time for scenario No. 6. We remark that the computation time is analogous to considering the iterations, as each simulation replication takes a nearly constant time. The computation result of the proposed VNS plus heuristic method (dynamic priority) illustrates that it could find better solutions with improved computational efficiency compared with OptQuest. The VNS plus heuristic method (dynamic priority) converges faster than the pure VNS as well as the algorithms of OptQuest to find the solution of train rescheduling problem. It can be seen that the proposed two-stage simulation-based optimization method is superior in terms of solution quality and convergence performance.

7.3. Statistical analysis

This section provides a reliability-oriented evaluation of the generated solutions. As mentioned earlier, the designed system can stochastically evaluate train schedules by means of simulation. It is important to draw attention to the fact that while train-rescheduling strategies will aim at optimizing efficiency, the impact of stochastic variables during a rescheduling procedure

Table 11. Statistical analysis of the best-found solution by VNS under different disruption scenarios.

Disruption scenario #	Total average delay time (hour)	Standard deviation (hour)	LB[a] (95%)	UB[b] (95%)	Minimum value (hour)	Maximum value (hour)
1	27.53	1.14	26.53	28.53	24.09	30.21
2	39.32	1.81	37.73	40.91	35.93	44.51
3	42.72	0.57	42.22	43.22	37.83	45.31
4	30.32	1.41	29.08	31.56	25.51	33.08
5	45.2	0.69	44.60	45.80	44.55	50.12
6	63.05	0.59	62.53	63.57	58.89	67.66
7	24.4	2.83	21.91	26.89	18.35	29.34
8	24.17	0.14	24.05	24.29	20.87	26.91
9	28.73	1.18	27.69	29.77	23.79	33.99
10	44.85	1.77	43.29	46.41	41.07	49.56

[a]LB: Lower bound, [b]UB: Upper bound.

is mostly neglected. In the present study, by applying discrete-event simulations, the solutions are analyzed statistically in a test environment. The result of statistical analysis of the best-found solution under different disruption scenarios is summarized in Table 11. The reported results include the expected value, standard deviation, minimum and maximum values, and the value of the (lower and upper) 95% confidence intervals (reported as LB and UB, respectively). The reliability of the result is represented by confidence interval that indicates the probability (e.g., 95%) that the response variable is within the range specified. For every performance measure (PFM), observation w_i is collected after each observation period i. Each statistic is estimated based on raw data $w_1, w_2, \cdots, w_n$, where n is the number of replications [43]. The lower and upper bounds of the Confidence Interval (CI) are obtained by Eq. (6). Values $t_{n-1,1-1/2\alpha}$ and $\mu_{1-1/2\alpha}$ are obtained from a table of t-values where $\alpha = 1 -$ Reliability:

$$\mathrm{CI} = \begin{cases} \bar{w} \pm t_{n-1,1-1/2\alpha}\cdot\frac{s}{\sqrt{n}} & n \leq 30 \\ \\ \bar{w} \pm \mu_{1-1/2\alpha}\cdot\frac{s}{\sqrt{n}} & n > 30 \end{cases} \tag{6}$$

For example, in the first scenario, the standard deviation of delay time is reduced by 60% compared to the OptQuest best-found solution. For the second scenario, the blockage takes 2 hours, and the standard deviation of delay time decreases by 47% compared to the OptQuest best-found solution. In the third case, which imposes longer duration of disruption, the standard deviation of the delay time decreases by more than 40% compared to the OptQuest best-found solution. The result of the simulation model indicates that the average and standard deviations of the delay times are affected by the duration of the blockage. Compared to the results in Table 12, the standard deviation of the delay times calculated with the proposed VNS method is on average 1.95 hours less than that with the OptQuest package. It can be concluded that the approach proposed in this paper has more dominant optimizing capability compared with the OptQuest. From the computational results, we also conclude that the performance of the proposed simulation-based optimization method is robust because of the less variance of delays. We also remark that the performance of FCFS is much less attractive compared to the other two; therefore, we skip reporting it in full.

8. Conclusion

Railway systems operate growingly at maximum capacity, timetables are becoming further at risk of instabilities, and delays propagate and reduce the service level perceived by the passengers. This makes real-time train traffic planning become more and more challenging as a result motivating the developing railway decision support systems. The procedure of disturbance management in rail transportation systems faces different challenges, e.g. the irregular occurrence time, the strong limitation of capacity for long period, and the presence of many other stochastic phenomena of smaller magnitude occurring in the network. This study developed an object-oriented discrete-event simulation model, which is able to model heavy disruption and small stochastic variations due to smaller delays, which optimizes traffic by means of dynamic priorities rules and further employs a variable neighborhood search algorithm as a global conflict resolution method in order to decrease the total delays after line blockage disruptions. The computational experiments along with a discussion about practical strengths and limitations of the proposed simulation-based optimization

Table 12. Statistical analysis of the best-found solution by OptQuest under different disruption scenarios.

Disruption scenario #	Total average delay time (hour)	Standard deviation (hour)	LB (95%)	UB (95%)	Minimum value (hour)	Maximum value (hour)
1	31.31	2.89	28.74	33.88	26.61	38.88
2	40.88	3.45	37.81	43.95	35.86	44.64
3	45.57	0.95	44.72	46.42	39.15	49.96
4	32.71	3.33	29.74	35.68	27.69	40.20
5	46.97	4.94	42.58	51.36	38.86	53.53
6	70.33	2.26	68.32	72.34	66.94	77.42
7	27.92	4.89	23.57	32.27	22.23	37.83
8	30.82	2.64	28.47	33.17	25.25	34.54
9	30.42	2.79	27.93	32.91	28.59	32.41
10	49.59	3.46	46.51	52.67	43.44	57.43

approach were conducted on real-world test cases of the Iranian railway network. The outcomes indicate that the proposed variable neighborhood search meta-heuristic outperforms the commercial OptQuest optimization toolbox in both solution quality (delays and their deviation) and computational time. Computational results of the developed model on an important part of the Iranian railway network illustrate that the simulation-based optimization approach is capable of finding near-optimal solutions in a reasonable computation time.

As accounted for the future research, many of the modeling characteristics can be adapted to more realistic situation. One important extension of the current study is to consider the network case. The determination of train priority can be handled through a comprehensive predictive model, or a more abstract optimization approach could be implemented, based, for instance, on robust or stochastic programming. In addition, it is worth mentioning that the simulation model can be extended to analysis of other performance measures such as punctuality and robustness. Finally, a still open challenge for the railway community is the development of exact algorithms for scheduling traffic under stochastic factors.

References

1. Törnquist, J. and Persson, J.A. "N-tracked railway traffic re-scheduling during disturbances", *Transportation Research Part B: Methodological*, **41**(3), pp. 342-362 (2007).
2. Hassannayebi, E., Zegordi, S.H., Amin-Naseri, M.R., and Yaghini, M. "Train timetabling at rapid rail transit lines: a robust multi-objective stochastic programming approach", *Operational Research*, pp. 1-43 (2016).
3. Narayanaswami, S. and Rangaraj, N. "A MAS architecture for dynamic, realtime rescheduling and learning applied to railway transportation", *Expert Systems with Applications*, **42**(5), pp. 2638-2656 (2015).
4. Hansen, I.A. and Pachl, J., *Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation* (2014).
5. Quaglietta, E., Corman, F., and Goverde, R.M. "Stability analysis of railway dispatching plans in a stochastic and dynamic environment", *Journal of Rail Transport Planning & Management*, **3**(4), pp. 137-149 (2013).
6. Suhl, L., Mellouli, T., Biederbick, C., and Goecke, J. "Managing and preventing delays in railway traffic by simulation and optimization", In *Mathematical Methods on Optimization in Transportation Systems*, Springer, pp. 3-16 (2001).
7. Takeuchi, Y., Tomii, N., and Hirai, C. "Evaluation method of robustness for train schedules", *Quarterly Report of RTRI*, **48**(4), pp. 197-201 (2007).
8. Sajedinejad, A., Mardani, S., Hasannayebi, E., and Kabirian, A. "SIMARAIL: simulation based optimization software for scheduling railway network", In *Simulation Conference (WSC)*, Proceedings of the 2011 Winter, IEEE (2011).
9. Motraghi, A. and Marinov, M.V. "Analysis of urban freight by rail using event based simulation", *Simulation Modelling Practice and Theory*, **25**, pp. 73-89 (2012).
10. Hasannayebi, E., Sajedinejad, A., Mardani, S., and Mohammadi, K.A.R.M. "An integrated simulation model and evolutionary algorithm for train timetabling problem with considering train stops for praying", In *Simulation Conference (WSC), Proceedings of the 2012 Winter.* IEEE (2012).
11. Büker, T. and Seybold, B. "Stochastic modelling of delay propagation in large networks", *Journal of Rail Transport Planning & Management*, **2**(1), pp. 34-50 (2012).

12. Eskandari, H., Rahaee, M.A., Memarpour, M., Nayebi, E.H., and Malek, S.A. "Evaluation of different berthing scenarios in Shahid Rajaee container terminal using discrete-event simulation", In *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World. 2013*, IEEE Press (2013).

13. Hassannayebi, E., Sajedinejad, A., and Mardani, S. "Urban rail transit planning using a two-stage simulation-based optimization approach", *Simulation Modelling Practice and Theory*, **49**, pp. 151-166 (2014).

14. Abbott, D. and Marinov, M.V. "An event based simulation model to evaluate the design of a rail interchange yard, which provides service to high speed and conventional railways", *Simulation Modelling Practice and Theory*, **52**, pp. 15-39 (2015).

15. Acuna-Agost, R., Michelon, P., Feillet, D., and Gueye, S. "A MIP-based local search method for the railway rescheduling problem", *Networks*, **57**(1), pp. 69-86 (2011).

16. Hassannayebi, E., Zegordi, S.H., and Yaghini, M. "Train timetabling in urban rail transit line using a Lagrangian relaxation approach", *Applied Mathematical Modelling*, **40**(23-24), pp. 9892-9913 (2016).

17. Corman, F., D'Ariano, A., Pacciarelli, D., and Pranzo, M. "Dispatching and coordination in multi-area railway traffic management", *Computers & Operations Research*, **44**, pp. 146-160 (2014).

18. Qu, W., Corman, F., and Lodewijks, G. "A review of real time railway traffic management during disturbances", *in Computational Logistics*, Springer, pp. 658-672 (2015).

19. Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., and Wagenaar, J. "An overview of recovery models and algorithms for real-time railway rescheduling", *Transportation Research Part B: Methodological*, **63**, pp. 15-37 (2014).

20. Corman, F. and Meng, L. "A review of online dynamic models and algorithms for railway traffic management", *Intelligent Transportation Systems, IEEE Transactions on*, **16**(3), pp. 1274-1284 (2015).

21. Fang, W., Yang, S., and Yao, X. "A survey on problem models and solution approaches to rescheduling in railway networks", *IEEE Transactions on Intelligent Transportation Systems*, **16**(6), pp. 2997-3016 (2015).

22. Cheng, Y. "Optimal train traffic rescheduling simulation by a knowledge-based system combined with critical path method", *Simulation Practice and Theory*, **4**(6), pp. 399-413 (1996).

23. Higgins, A., Kozan, E., and Ferreira, L. "Heuristic techniques for single line train scheduling", *Journal of Heuristics*, **3**(1), pp. 43-62 (1997).

24. D'Ariano, A. "Innovative decision support system for railway traffic control", *IEEE Intelligent Transportation Systems Magazine*, **1**(4), pp. 8-16 (2009).

25. Corman, F., D'Ariano, A., Pacciarelli, D., and Pranzo, M. "Centralized versus distributed systems to reschedule trains in two dispatching areas", *Public Transport*, **2**(3), pp. 219-247 (2010).

26. Corman, F., D'Ariano, A., Hansen, I.A., and Pacciarelli, D. "Optimal multi-class rescheduling of railway traffic", *Journal of Rail Transport Planning & Management*, **1**(1), pp. 14-24 (2011).

27. Dündar, S. and Şahin, İ "Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways", *Transportation Research Part C: Emerging Technologies*, **27**, pp. 1-15 (2013).

28. Hassannayebi, E. and Kiaynfar, F. "A greedy randomized adaptive search procedure to solve the train sequencing and stop scheduling problem in double track railway lines", *Journal of Transportation Research*, **9**(3), pp. 235-257 (2012).

29. Dollevoet, T., Corman, F., D'Ariano, A., and Huisman, D. "An iterative optimization framework for delay management and train scheduling", *Flexible Services and Manufacturing Journal*, **26**(4), pp. 490-515 (2014).

30. Hassannayebi, E. and Zegordi, S.H. "Variable and adaptive neighbourhood search algorithms for rail rapid transit timetabling problem", *Computers & Operations Research*, **78**, pp. 439-453 (2017).

31. Hassannayebi, E. and Zegordi, S.H. "Variable and adaptive neighbourhood search algorithms for rail rapid transit timetabling problem", *Computers & Operations Research* (2016).

32. Grinstead, C.M. and Snell, J.L., *Introduction to Probability*, American Mathematical Soc. (2012).

33. Nie, L. and Hansen, I.A. "System analysis of train operations and track occupancy at railway stations", *European Journal of Transport and Infrastructure Research*, **5**(1), pp. 31-54 (2005).

34. Borndörfer, R., Schlechte, T., and Swarat, E. "Railway track allocation-simulation, aggregation, and optimization", In *Proceedings of the 1st International Workshop on High-Speed and Intercity Railways*, Springer (2012).

35. Paolucci, M. and Pesenti, R. "An object-oriented approach to discrete-event simulation applied to underground railway systems", *Simulation*, **72**(6), pp. 372-383 (1999).

36. Venkatesan, K., Gowri, A., and Sivanandan, R. "Development of microscopic simulation model for heterogeneous traffic using object oriented approach", *Transportmetrica*, **4**(3), pp. 227-247 (2008).

37. Hullinger, D.R. "Taylor enterprise dynamics", In *Simulation Conference Proceedings*, Winter, IEEE (1999).

38. Hassannayebi, E., Sajedinejad, A., and Mardani, S. "Disruption management in urban rail transit system: A simulation based optimization approach", In

Handbook of Research on Emerging Innovations in Rail Transportation Engineering, IgI-Global. pp. 420-450 (2016).

39. Mladenović, N. and Hansen, P. "Variable neighborhood search", *Computers & Operations Research*, **24**(11), pp. 1097-1100 (1997).
40. Hansen, P., Mladenović, N., and Pérez, J.A.M. "Variable neighbourhood search: methods and applications", *Annals of Operations Research*, **175**(1), pp. 367-407 (2010).
41. Xiao, Y., Zhao, Q., Kaku, I., and Mladenovic, N. "Variable neighbourhood simulated annealing algorithm for capacitated vehicle routing problems", *Engineering Optimization*, **46**(4), pp. 562-579 (2014).
42. Kleijnen, J.P. and Wan, J. "Optimization of simulated systems: OptQuest and alternatives", *Simulation Modelling Practice and Theory*, **15**(3), pp. 354-362 (2007).
43. Kelton, W.D., Sadowski, R.P., and Sadowski, D.A., *Simulation with ARENA*, McGraw-Hill, Inc (2002).

Biographies

Masoud Shakibayifar is a PhD candidate at the Department of Transportation Engineering and Planning, School of Civil Engineering, Iran University of Science & Technology (IUST).

He is doing his PhD thesis entitled: "Train Re-Scheduling model for Reducing Delay under Disturbances". This research focuses on designing simulation-optimization methods for train traffic rescheduling in railways systems.

His research interests are railway electrification, railway scheduling and rescheduling, and optimization of transportation systems under uncertainty. He obtained BSc (1997-2001) and MSc (2004-2006) degrees in Civil Engineering at Sharif University of Technology, Tehran, Iran. He has managed more than 15 transportation-planning projects in Iranian Railway Company and the Ministry of Roads and Urban Development. Recently, he has managed a research project related to the design of railway traffic simulation software for Iranian Railway Company.

Abdorreza Sheikholeslami is an Associate Professor at the Department of Transportation Engineering and Planning, School of Civil Engineering, Iran University of Science & Technology (IUST). He received his BSc and MSc degrees from IUST. He received his PhD degree in Transportation Engineering and Planning from Iran University of Science and Technology in 2006.

His research interests are transportation planning, transportation economics, railway scheduling, disturbance management, mathematical models and optimization techniques for designing transportation systems.

He was elected as the top student in technical and engineering group in Iran in 2001. He has teaching experience of more than 20 years at IUST. He has published more than 80 scientific papers in ISI journals and international conferences.

Francesco Corman is an Assistant Professor in Transport Engineering and Logistics, TUDelft, and a Guest Professor at KULeuven.

His research interest relates to optimization of transport networks under uncertainty. This substantiates optimization and automation in supply chain networks and logistic systems, especially with interconnected systems and modes, mathematical models and optimization techniques for traffic control in railways systems, optimal coordination strategies, equity issues in logistics; robustness, reliability, and resilience of transport networks under stochastic phenomena; analytics, information, and uncertainty dynamics; a-priori and online data collection and assessment of quality of information.

He has more than 25 high-impact articles in journals and book chapters, and more than 100 scientific papers in journals or peer-reviewed international conferences; he received numerous awards and nominations.

He has been involved in organizing and chairing of the International Conference on Computational Logistics 2015, among others. He has been the General Chair of Problem Solving Competition, INFORM Railway Application Section 2015 and 2016. He is a guest editor in major transportation journals (TRC, TRE, Public Transport), and participates actively in TRB and Informs community. He is the initiator of many international research exchange and coordination networks.