Advances in Computational Intelligence (2022) 2:27
https://doi.org/10.1007/s43674-022-00039-7

ORIGINAL ARTICLE

Multi-agent-based dynamic railway scheduling and optimization: a coloured petri-net model

Poulami Dalapati[1] · Kaushik Paul[2]

Received: 21 February 2021 / Revised: 2 February 2022 / Accepted: 16 May 2022 / Published online: 16 June 2022

Abstract

This paper addresses the issues concerning the rescheduling of a static timetable in case of a disaster, encountered in a large and complex railway network system. The proposed approach tries to modify the existing schedule to minimise the overall delay of trains. This is achieved by representing the rescheduling problem in the form of a Petri-Net and the highly uncertain disaster recovery time in such a model is handled as Markov decision processes (MDP). For solving the rescheduling problem, a distributed constraint optimisation (DCOP)-based strategy involving the use of autonomous agents is used to generate the desired schedule. The proposed approach is evaluated on the real-time data set taken from the Eastern Railways, India by constructing various disaster scenarios using the Java Agent DEvelopment Framework (JADE). The proposed framework, when compared to the existing approaches, substantially reduces the delay of trains after rescheduling.

Keywords Scheduling · Railway network · Multi-agent system · Distributed constraint optimization · Markov decision process · Petri-Net

1 Introduction

The framework of railway system is geographically distributed within the country and is considered as a vital mode of transportation. The development of an efficient and optimized train schedules in such a framework (Li et al. 2014; Krasemann 2010; Feynman and Vernon Jr. 2013; Kersbergen et al. 2013; Dalapati et al. 2014) is a tedious task due to the involvement of situational network complexities and numerous system constraints. Arrival and departure of trains are highly dependent on the constraints such as availability of tracks between stations and platforms at the stations. The occurrence of any disturbance or disruption in the railway framework such as natural disasters, vandalism, blockage in platform and any mishap on tracks and platform make the offline schedule sub-optimal for use. Moreover, the uncertainties in recovery time from these situations influence the availability of the transportation resources and the schedule, making the global optima less-effective. These situations require dynamic rescheduling of the trains to minimize the consequences of such scenarios, which leads to the continual alteration in the objective function along with the changes in the constraints over time (Nguyen and Yao 2009; Leurent 2011). With these background, this entire scheduling problem is modelled as a multi-agent-based distributed constraint optimization problem (DCOP) (Atlas and Decker 2007; Nguyen et al. 2012), where In this scenario each and every agent collaborate among themselves based on jointly agreed protocols and constraints. The notion of Markov decision process (MDP) is used to mathematically represent the uncertainty and probabilistic nature involved in recovery time (Vidal 2010; Bernstein et al. 2002, 2009). Each node in MDP is considered as a possible state of the disastrous situation in the railway framework. The mapping of the state transition functions is done in relation to the constraint of DCOP, where each agent chooses its action to minimize the expected delay based on the given policy. In an MDP setting, the integral function of all the agents is to perceive the optimal solution considering all the necessary constraints, which ensures the minimization of the entire delay of the railway network.

In this paper, the railway framework consists of several trains, stations and tracks, where certain number of trains are on tracks and others are at stations. During the occur-

✉ Poulami Dalapati
poulami.dalapati@lnmiit.ac.in

[1] Department of Computer Science and Engineering, The LNM Institute of Information Technology, Jaipur, Rajasthan, India

[2] Department of Electrical Engineering, BIT Sindi, Dhanbad, India

rence of a disaster, in or within the vicinity of a station, the nearby stations, the arriving and departing trains are notified about the accidental situation by the station authorities. The train and the station agents thereafter check for the disaster recovery time and the availability of the resources to arrive at destination. It is to be noted that if the disaster recovery time does not impact the scheduled arrival or departure time of the trains then there is no alteration in the original schedule.

Otherwise, the proposed rescheduling approach, formulated in Sect. 5, is implemented to generate a new dynamic schedule providing the necessary information to the trains.

The rest of this paper is organized as follows: in Sect. 2 some previous works in the related domain are discussed followed by a brief discussion of scope of the current work. Section 3 describes railway network system and its architectural model with Petri-Net, respectively. The notion of DCOP and MDP is formulated in Sect. 4. A proposed solution approach is illustrated in Sect. 5 with some examples which elucidates detailed analysis of the approach with a case study. Simulation results are evaluated in Sect. 6. Finally Sect. 7 concludes the proposed work.

2 Background and motivation

In real world, a few sorts of research have been done on train rescheduling in light of line topology (Meng and Zhou 2011) and network topology (Acuna-Agost et al. 2011b, a) under disturbances (Meng and Zhou 2011; Acuna-Agost et al. 2011b, a; Chalumuri and Yasuo 2014). The disturbance scenarios can also be modelled as certain and uncertain (Yang et al. 2011) based on its recovery time. Depending on the circumstances, the decision scenarios of rescheduling also vary like retiming, rerouting or reordering (Takouna and Rojas-Cessa 2008; Meng and Zhou 2014) with respect to delay management of passenger railway services (Acuna-Agost et al. 2011b; Alwadood et al. 2012; Abid and Khan 2015).

In this section, we mainly discuss some recent state-of-the-art which include classical methods such as first come first serve (FCFS) (Gafarov et al. 2015), tabular method (Mukherjee and Basu 2011) etc, as well as stochastic methods such as bionomic algorithm (Beg et al. 2011), hybrid meta-heuristic algorithm (Salehipour et al. 2013; Hancerliogullari et al. 2013), particle swarm optimization (PSO) (Yaman et al. 2018; Talal 2014), chicken swarm optimization (He et al. 2019), ant colony optimization (ACO) (Eaton et al. 2017, 2016) etc. The main disadvantages of classical methods are that, their outputs are restricted by the initial solutions, which causes to fall into local optima. Whereas, stochastic methods are found to overcome such loopholes and eventually give better feasible solutions. Again, such solutions seem to be better as these are less expensive computationally.

To start with, we discuss the research paper by Li et al. (2014) and Kersbergen et al. (2013), as these work are the main motivation of our further research. In this paper a track-backup approach (Li et al. 2014) is proposed to minimise the negative effects arising from the disturbances while rescheduling. This approach optimally assigns a backup track to each affected train, based on the original timetable, estimated recovery time, and track changing cost in line-topology. Authors in Kersbergen et al. (2013) introduced an MILP (mixed integer linear programming)-based model to reduce the computation time of rescheduling.

Here, they have taken a threshold delay and assumed that all the delays are below it. Though this approach is fast and beneficial for small network, it takes hours for real-world network which is not desirable.

Gaied et al. (2018) in their work implemented P-time Petri nets to examine the traffic condition in the railway network of Sahel Tunisia and assessed the performance of the same. The proposed approach led to the efficient evaluation of the collision and enhanced the operational performance of the lines. In addition to the proposed work a control scheme was developed to amend the sojourn time of the locations to mitigate the influence of disturbance on the traffic. The adopted control strategy minimizes the delay satisfying the customer satisfaction. In another research Gaied et al. (2019) modelled the railway transportation system as discrete event framework, based on time constraint. The Petri nets are used to evaluate discrete framework model of the railway transportation system to develop an efficient schedule to avoid collision and delay. Some other Petri-Net-based railway modelling can also be found in Giglio and Sacco (2016), Yianni et al. (2017), Wang et al. (2019), Li and Yan (2019), and Utomo and Arfi (2019). Giglio and Sacco (2016) model a railway network through Petri-Net for physical and logical part of the same network to analyze, optimize, and control it. A time Petri Net (TPN)- based mental-workload evaluation model is described in Wang et al. (2019) which handles multitasking scheduling of departure phase of trains, though optimized strategy given by the authors varies for different person. An improved and efficient locomotive scheduling is modeled in Li and Yan (2019) using timed Petri-Net.

Hassan and Reynolds (2018)) deployed genetic algorithm (GA) for supply chain scheduling of iron ore railway system. A comparative analysis was established with the non-linear programming and it was seen that the GA preformed efficiently for the scheduling phenomenon. Abourraja et al. (2017) formulated a multi agent-based approach for crane scheduling. The proposed approach was based on two planer agents and time horizon planning have been assigned to each of the agents. The obtained results exhibit that the adopted approach efficiently minimized the waiting time and outperformed the other existing framework.

In contrast with these discussed methodologies, our approach considers *distributed network topology* in the simulation. Again, in most cases, our proposed approach outperforms the state-of-the-art work in terms of computation time, i.e., we get more optimized results in the application domain. In the paper Li et al. (2014), it is assumed that all the stations must have siding track, but we consider more realistic situation, where there may be only one track in the station. This makes the scenario more realistic and complex as well, while decision regarding dynamic rescheduling is to be made.

In light of the discussion above, the scope and the main contributions of this research work include:

- Modelling of real-time railway system as a Petri-Net along with mathematical representation of the scenario with DCOP and MDP to enable formal analysis.
- An autonomous agent-based disaster handling and rescheduling approach is also proposed considering network topology, which is capable of providing sufficiently good solution.
- Situational complexity of scalability issues in terms of number of decision variables and constraints are also taken into consideration while optimising total journey time delay of trains.

3 Case study: railway network and scheduling topology

A railway network is geographically distributed and consists of stations (S), trains (T) and tracks (Γ) as shown in Fig. 1. At any time instant, multiple trains are either at stations or running on tracks. In case of any inconvenience, railway scheduling and rescheduling is performed in a dynamic way to ensure that, the highest priority train is rescheduled first. These priorities may change dynamically to reduce total journey time delay depending on the current scenario, which is discussed further later in this section.

3.1 Assumptions

To model and implement this work some assumptions have been taken, such as:

- There can be multiple tracks between two stations.
- Each station may consist of one or more than one platforms.
- Stations can communicate directly with trains and other neighbouring stations.
- Trains can communicate with its nearby or monitoring stations only.
- Station conveys the recovery status of the blocked tracks or platform to trains and its neighbouring stations.
- All the trains begin and end their journey at stations.

3.2 Classification of railway parameters

- Stations (S)

- Stations where train T is scheduled to stop.
- Stations where train T does not stop, but station is a junction.
- Stations where train T neither stops nor the station is a junction, but in case of inconvenience, train may stop, so that other trains can pass.

For generalisation, dwell-time is taken as a parameter. It specifies the scheduled amount of time a train stops at a station for passenger conveniences. If dwell-time of train T at any stations S is greater than zero, then S is a stopping station for that train T.

- Tracks (Γ)

- Double-line tracks between two stations, *UP* and *DOWN*.
- Only a single line between two stations, used as either *UP* or *DOWN* as per schedule.
- Three or more tracks between two stations, *UP*, *DOWN* and general tracks, used as either *UP* or *DOWN* as per need.

For simplicity, we consider different directions of same track as two or more different tracks, i.e. *UP* as one resource and *DOWN* as another, and rescheduling of either UP or DOWN line trains at a time.

- Trains (T) In a railway network, depending upon various criteria, such as speed, facilities, distance covered, public demand, frequencies etc., train T can be classified as long-distance train (T^L) and short-distance train (T^S). (T^L) can again be categorised as premium train (T^{Pr}), Mail Trains (T^M), and freight train (T^F), whereas (T^S) can be categorised as passenger train (T^P) and local train (T^{Lo}). i.e., $T = T^L \cup T^S$, $T^L = T^{Pr} \cup T^M \cup T^F$, $T^S = T^P \cup T^{Lo}$.

Except T^{Pr}, during busy hours (when rate of passenger flow is higher), T^S get higher priority than any T^L. Otherwise, during normal hours, T^L get higher priority. Again, as railway system faces delay due to many reasons, these priorities change dynamically over time. As an example, if any T^L is delayed by more than the permissible threshold delay, then other trains get higher priority which are on time. Priorities are assigned such as $y_1 = Prio\,(T^{Pr})$, $y_2 = Prio\,(T^M)$, $y_3 = Prio\,(T^F)$, $y_4 = Prio\,(T^P)$, $y_5 = Prio\,(T^{Lo})$. This priority allocation policy is defined by the existing railway system of the region, considered in the experiments as described in Sect. 6.1. In general, $y_1 > y_2 > y_4 > y_5 > y_3$.

Table 1 Notations

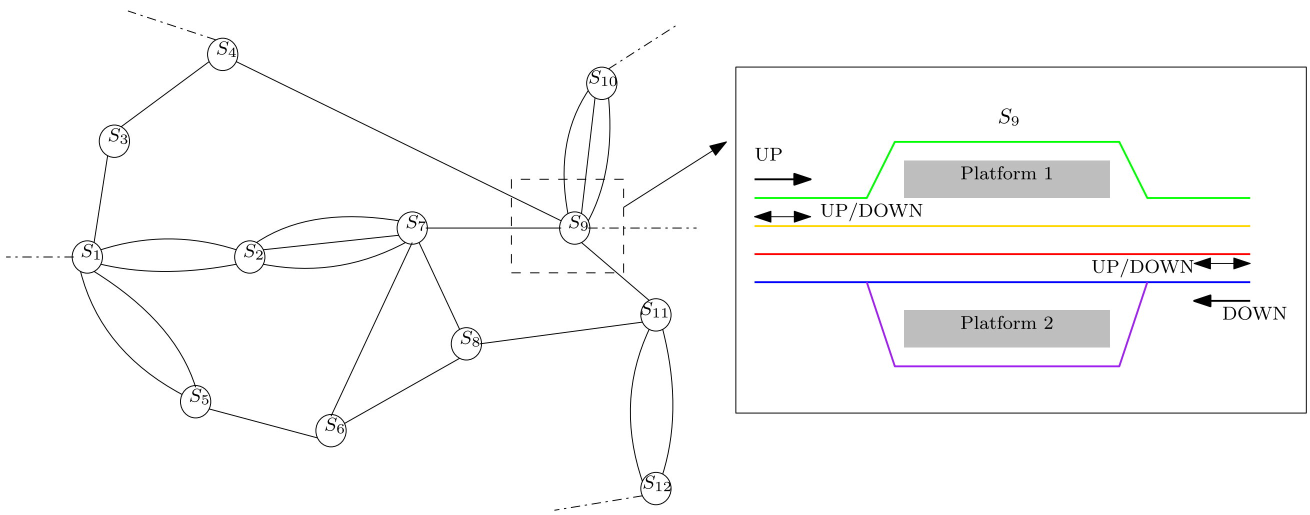
Indices and parameters			
S	Stations	o_{ji}^{J}	Journey time of train j in original timetable
T	Trains	o_{ji}^{d}	Original dwell time of train j at station S_i
Γ	Tracks	x_{jl}^{J}	Journey time of train j on track l
i	Station index	a	Agent index
j	Train index	q	Number of agents, where $q = m + n$
l	Track index	t	Time instant
k	Platform index	t_D	Time of disaster
n	Number of stations	t_R	Time of recovery
m	Number of trains	t_{Busy}	Busy time period of the day
p	Maximum number of platforms at each station	$i' \in [1, n] \backslash i$	Index of station other than the ith station
o_{ji}^{AT}	Arrival time of train j at station i in original timetable	$j' \in [1, m] \backslash j$	Index of train other than the jth train
o_{ji}^{DT}	Departure time of train j from station i in original timetable	$i'' \in [1, n] \backslash i, i'$	Index of station other than the ith and i'th station
δ_{ji}	Delay of train j at station i	$j'' \in [1, m] \backslash j, j'$	Index of train other than the jth and j'th train
δ_{jl}	Delay of train j on track l	τ^B	Buffer Time, where $t_D + \tau_1 \leq \tau^B \leq t_D + \tau_2$
δ_{Th}	Threshold value for delay of all trains		
Decision variables			
x_{ji}^{AT}	Arrival time of train j at station i due to disaster	x_{ji}^{DT}	Departure time of train j from station i due to disaster
P_{jik}	Platform indicator, $P_{jik} = 1$ if train j occupies kth platform of station i, otherwise 0	L_{jil}	Track indicator, $L_{jil} = 1$ if train j occupies lth track connecting to station i, otherwise 0 and when $L_{jil} = 1$, $L_{j'il} = 0$
$Prio(T_j)$	Priority of train T_j	x_{ji}^{d}	Actual operation time of train j at station S_i
τ_1	Minimum time required to recover from the disaster	τ_2	Maximum time required to recover from the disaster
τ_R	Time to recover with the density function $\phi(x)$, where $x \in [\tau_1, \tau_2]$		

However, when $t = t_{Busy}$ (from 9 : 00 am to 11 : 00 am and 5 : 00 pm to 7 : 00 pm), $y_1 > y_4 > y_5 > y_2 > y_3$. Again, in case of delayed trains, priority changes dynamically such as $y_4 \geq y_1 \geq y_5 \geq y_2 > y_3$.

There are two kinds of inputs in our system, static input, which is pre-planned as per the schedule and the other is dynamic input, which is triggered by the changes due to disruption. At any time instant t, each station has a fixed number of incoming and outgoing tracks. Station database is updated with the information about incoming and outgoing trains in terms of their arrival and departure time. When a disaster occurs, one or more tracks between stations get deleted from the databases and platform counts decreases from station databases if any disruption occurs at the platform. It is assumed that one platform is directly connected to at most two tracks. As railway network is represented as a connected multigraph, there may be other possible paths to reach to the destination. After disaster has occurred, system checks for the trains which may reach a particular station within the calculated buffer time τ^B.

3.3 Railway architecture model

Modelling of a transportation system is considered to be a complex jon which needs development of proper mathematical and infrastructural model. This model should ensure smooth activity of all railway entities and safety of the system. Given this background, a *multi-agent system (MAS)*



(a) Railway Network as Multigraph **(b)** Station consisting of multiple tracks and platforms

Fig. 1 Railway architecture

(Vidal 2010; Wooldridge 1966) is a natural choice for modelling such distributed system. Here, we represent the railway network (RN) as a pair of multigraph (G) and an agency (Ag). i.e., $RN = < G, Ag >$. Again, $G = < V, E >$, where V is set of vertices and E is set of edges.

From notations in Table 1, $V = \{v_i | \ i \in [1, n]\}$ and $v_i = S_i$ means vertex is a station, E represents tracks between stations, $T = \{T_j | \ j \in [1, m]\}$, indicates trains (see Fig. 1, and $Ag = \{Ag_a | \ a \in [1, q]\}$, denotes agency. Each station and train is associated with an agent. SA and TA denote the set of station agents and set of train agents, respectively, where $S_i \in S$ with $Sa_a \in SA$ and $T_j \in T$ with $Ta_a \in TA$. Some parameters, indices, and decision variables are defined in Table 1. The goal of such modelling is to give a better overview of how a railway system operates.

3.4 Formal description of Petri-Net model for railway system

Till date different modelling techniques and tools are proposed, which can depict a railway transport system (Dong et al. 2012; Gaied et al. 2018; Cavone et al. 2017). Among those, Petri-Nets seems to be most suitable one to describe the distributed nature of such system. Sect. 3.4 introduces the general concepts of Petri-Net (Yen 2006; Murata 1989) for better understanding of the proposed approach. Major use of various kinds of Petri-Net (Chaki and Bhattacharya 2006; Jensen and Kristensen 2009; Wang et al. 2016) is modelling of static and dynamic properties of complex systems, where concurrent occurrences of events are possible, but there are constraints on the occurrences, precedence or frequency of these occurrences. Graphically a Petri-Net is a directed bipartite graph, where nodes represent places, transition and directed arcs which link places to transitions or transitions to places. The state of a Petri-Net is given by the marking, describing the distribution of tokens in the places. Our proposed Petri-Net model deals with dynamism, uncertainty, and conflict situations in decision making choices upon different conditions. To overcome such conflicts the idea of colour token is introduced that enables a particular condition. Agents are considered as a token which can move from one environmental state to other. In real time system agents perform some action if it sense a particular environment; in contradiction some states are just used as an intermediate one. The Petri-Net model for railway network is proposed with six tuples, which is as follows:

$\{P, Tr, F, Tok, f_C, M_0\}$

$P : \{P_1, P_2, \ldots, P_b\}$, where $b > 0$ is a finite set of *Places*.

$P = P_N \cup P_{f_c}$, where P_N is the set of places where no explicit function is executed on arrival of resource token and P_{f_c} is the set of places which executes a function or checks condition on arrival of resource token.

$Tr : \{Tr_1, Tr_2, \ldots, Tr_z\}$, where $z > 0$ is a finite set of *Transitions*.

$Tr = Tr_I \cup Tr_c$, where Tr_I is the set of immediate transition which is fired as soon as the required tokens are available at input place and an action is performed. Tr_c is the set of colour transition which is fired when the colour token is available in the input place.

$F : (P \times Tr) \cup (Tr \times P)$ is the set of *flow function*.

$F = F_+ \cup F_-$, where F_+ refers finite set of input flow and F_- refers finite set of output flow.

Tok : Set of *Token*.

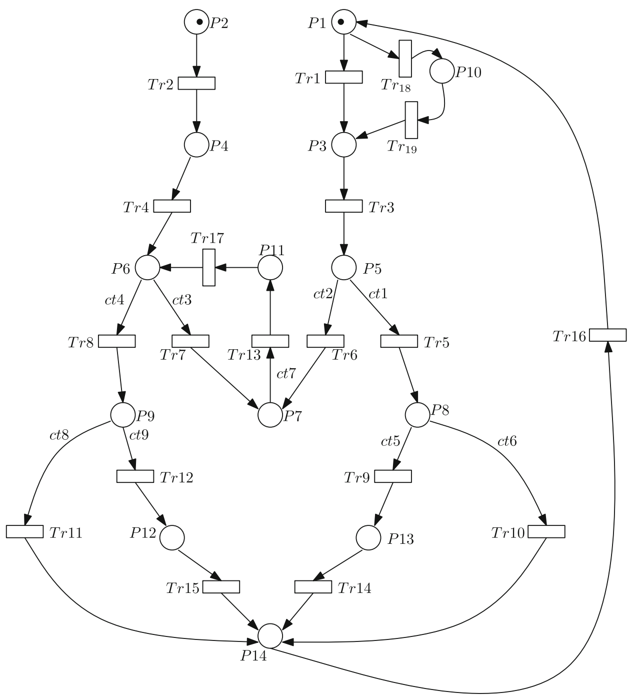


Fig. 2 Petri-Net model $PN1$ of railway network

$Tok = Tok^c \cup Tok^{Ag}$ and $Tok^c \cap Tok^{Ag} = \phi$, where Tok^c is the set of color token, c represents colour and Tok^{Ag} is the resource token (Agent token).

$f_c : \{f_c1, f_c2, \ldots, f_cu\}$, where $u \geq 0$ is the set of *Functions* that execute in P_{f_c} when a resource token arrived at the place. Function can generate color token or perform some operations.

M_0 : *Initial marking* of Petri-Net.

$\beta : F_+(Tok^c \times Tok^{Ag}) \rightarrow F_-(Tok^{Ag})$ or $(Tok^c \times Tok^{Ag}) \rightarrow Tok^{Ag}$

β says that color token is only used for taking a decision to resolve conflict. It won't propagate to next state.

The framework of our proposed work is based on Eastern Railway, India. This part of the network consists of four divisional headquarter stations: Howrah, Sealdah, Asansol, and Maldah; and many intermediate stations. A train, when is at station, counts waiting time before it departs and occupies a track. our propsed Petri-Net model considers bidirectional tracks between stations. To ensure safety, in terms of avoiding collision between two trains, the concept of color token which is generated by places. These color tokens decides firing of a particular transition. Figure 2 represents the Petri-Net model of our network, where place is represented by a circle, transition by a rectangle, and input and output flow by arrow and the corresponding description of individual places, transitions, and tokens are given in Tables 2, 3, and 4, respectively.

4 DCOP and MDP representation of the proposed rescheduling approach

4.1 Representation as DCOP

The aforementioned problem of railway rescheduling can be modelled mathematically as a distributed constraint optimization problem. The objective is for multiple agents (stations, trains etc) to optimize the local constraint with local information to reach a global common results

which is here defined in terms of time delay of the trains. Here the problem of train rescheduling is represented as DCOP with four tuples, $\langle Ag, X, D, C \rangle$, where Ag : Set of agents,

X : Set of variables, $X = \{x_{ji}^{AT}, x_{ji}^{DT}, P_{jik}, L_{jil}\}$,

D : Set of domains, $D = t_D, \ldots, t_D + \tau_R$,

C : Set of constraints.

Constraints (C)

• *Continuity constraint:*

$$x_{ji}^{AT} \geq x_{ji'}^{DT} + x_{jl}^{J}, \tag{1}$$

i.e. the arrival time of any train at any station depends on its departure time from the previous station and the total journey time between these stations.

• *Time delay constraint:*

$$\left.\begin{array}{l} x_{ji}^{AT} \geq o_{ji}^{AT} \\ x_{ji}^{AT} - o_{ji}^{AT} = \delta_j, \end{array}\right\} \tag{2}$$

i.e. the actual arrival time of T_j at S_i must be greater or equal to its original arrival time at that station. If both are equal then the train is on time, otherwise there is some delay δ_j. Delay is calculated by the difference between the original journey time and the actual journey time.

• Index of platforms that trains occupy can not be greater than p:

$$\forall j \in [1, m]\ \exists\ P_{jik} \in [0, 1],\ where\ 1 \leq k \leq p,\ \forall\ S_i \tag{3}$$

and

$$\sum_k P_{jik} \leq p,\ where\ 1 \leq k \leq p \tag{4}$$

• If train T_j occupies lth track, connecting to station S_i, then train $T_{j'}$ can not occupy the same track at the same time:

$$if\ L_{jil} = 1,\ then\ L_{j'il} = 0. \tag{5}$$

Table 2 Description of places of $PN1$

Places (P)	Description
P1	Trains T_j is at station S_i and count waiting time to leave
P2	Trains $T_{j'}$ is at station $S_{i'}$ and count waiting time to leave
P3	T_j starts running
P4	$T_{j'}$ starts running
P5	T_j is on track
P6	$T_{j'}$ is on track
P7	T_j and $T_{j'}$ both sense junction station and checks whether it has free platform
P8	T_j senses simple station and checks whether it has free platform
P9	$T_{j'}$ senses simple station and checks whether it has free platform
P10	T_j has completed its journey and ready for the next journey
P11	T_j entering into station
P13	No free platform for T_j
P14	T_j andor $T_{j'}$ reached to the station

Table 3 Description of transitions of $PN1$

Transitions (Tr)	Description
Tr1	It will fire when T_j finishes its waiting time at S_i
Tr2	It will fire when $T_{j'}$ finishes its waiting time at S_i
Tr3	It will fire when T_j is on track
Tr4	It will fire when $T_{j'}$ is on track
Tr5	It will fire if T_j senses a simple station
Tr6	It will fire if T_j senses a junction station
Tr7	It will fire if $T_{j'}$ senses a junction station
Tr8	It will fire if $T_{j'}$ senses a simple station
Tr9	It will fire if no platform is free for T_j
Tr10	It will fire when at least one platform is free for T_j
Tr11	It will fire when at least one platform is free for $T_{j'}$
Tr12	It will fire if no platform is free for $T_{j'}$
Tr13	It will fire when at least one platform is free for highest priority train
Tr14	It will fire if a platform gets free for T_j
Tr15	It will fire if a platform gets free for $T_{j'}$
Tr16	It will fire when train entering into the station
Tr17	It will fire when train leaving a junction station
Tr18	It will fire when T_j reaches its destination
Tr19	It will fire when T_j is ready to leave for a new journey

Table 4 Description of color tokens of $PN1$

Colour token (ct)	Description
ct1	P5 generates it if T_j senses a simple station in front of it and enables transition Tr5
ct2	P5 generates it if T_j senses a junction station in front of it and enables transition Tr6
ct3	P6 generates it if $T_{j'}$ senses a junction station in front of it and enables transition Tr7
ct4	P6 generates it if $T_{j'}$ senses a simple station in front of it and enables transition Tr8
ct5	P8 generates it if no free platform is available and enables transition Tr9
ct6	P8 generates it if at least one platform is available and Tr10 is enabled and enables transition Tr10
ct7	P7 generates it if at least one platform is available and enables transition Tr13
ct8	P9 generates it if at least one platform is available and Tr11 is enabled and enables transition Tr11
ct9	P9 generates it if no free platform is available and enables transition Tr12

- Required resources of T_j at time t, $Re(T_j)|^t$, is either a platform at a station or a track between two stations:

$$Re(T_j)|^t = L_{jil} \text{ } or \text{ } Re(T_j)|^t = P_{jik}. \tag{6}$$

- Route of the train T_j, $Rou(T_j)$, is a series of P and L:

$$Rou(T_j) = \left(\bigwedge_{i=1}^{n-1} P_{jik} L_{jil} \right) . \cup P_{jnk}, \text{ } where \text{ } j \in [1, m] \tag{7}$$

4.2 Representation as MDP

In real world, agents inhabit an environment whose state changes either because of agent's action or due to some external event. Agents sense the state of world and the choice of new state depends only on agent's current state and agent's action. As railway system is very prone to disasters, the uncertainty of the situation makes the environment more vulnerable. In this paper, this scenario os modelled as Markov decision process (MDP) with its states (W) and transitions (Ψ). The detailed description of the same with respect to railway system is as follows:

– *Set of world state* (W)
  W represents the set of agent's state(s) in railway network under disturbance. Train agents can sense three kind of states; if S_i is assumed to be a station, where disaster happens, then T_j is either on track connecting S_i or at a platform of S_i or at a platform of station$S_{i'}$, connected to S_i. i.e. $W = \{(L_{jil} = 1), (P_{jik} = 1), (P_{j'i'k} = 1)\}$
– Transition function (Ψ) The state transition function is denoted as $\Psi(\omega, C, \omega')$. In our proposed approach, each action C maps to constraint(s) of DCOP to satisfy to reach from state ω to the next state ω', where $\omega, \ \omega' \in W$.

  If the train T_j is on the lth track, connecting to station S_i, where disaster happened, i.e. $L_{jil} = 1$, but no platform is available, i.e. $P_{jik} = 1$, then T_j must wait on the current track. Therefore, there is no state change from the current state $L_{jil} = 1$. Now, if platform is free, i.e. $P_{jik} = 0$, then T_j can reach to the next station. Therefore, state transition from current state $L_{jil} = 1$ is possible.
  If T_j is on pth platform of S_i and the lth track is free but the platform at the next station $S_{i'}$ is not free, i.e. $P_{ji'k'} = 1$, no state change is possible from current state $P_{jik} = 1$. Similarly, even if there is free platform at $S_{i'}$, if the lth track is not free or both the track and platform is not available at time t, no state change is possible.
  In Fig. 3, every node represents a state of railway environment and each arc represents an action which is indeed a constraint. The transition from one state to another state happens if and only if the corresponding agent satisfies the specific constraint(s).

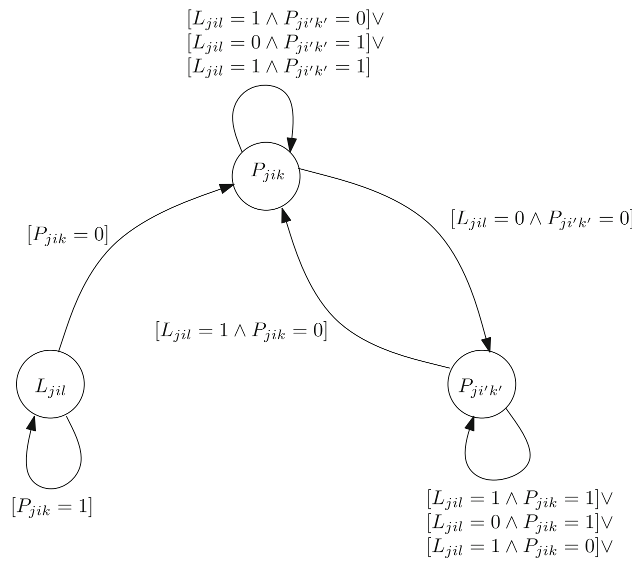


Fig. 3 MDP representation with set of states and transition functions.

5 Disaster handling and rescheduling approach

According to real-time scenario, in case of platform blockage or track blockage, disaster handling and rescheduling model of railway system refers three situations:

– *Delay or stop* at the station or on track (*Retiming*).
– *Change in departure sequence* of trains at the station depending on priority of trains (*Reordering*).
– *Reschedule* to alternative path (*Rerouting*).

5.1 Case 1: partial node deletion from the graph *G*

A station S_i faces problem due to disaster and the train T_j is on track l, approaching to the station S_i, that is

$$L_{jil} = 1 \tag{8}$$

Case 1.1

If S_i has a free platform at the time, when T_j reaches to S_i, then the system can allow T_j to reach S_i, if and only if the priority of the incoming train T_j has the highest priority amongst all trains T and the resource (Re) required for any

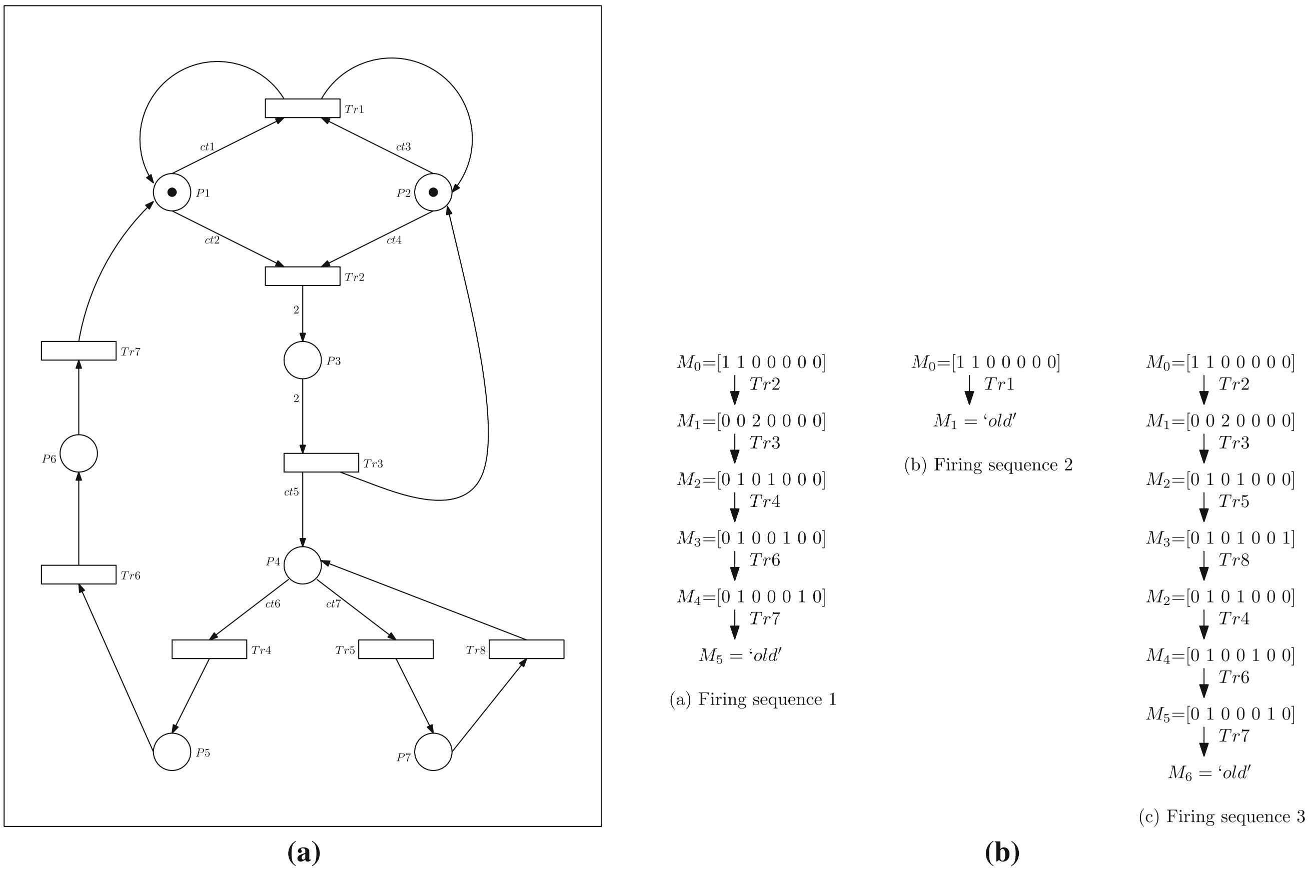


Fig. 4 Petri-Net $PN2$. **a** Petri-Net model $PN2$ of Case 1. **b** Reachability tree of $PN2$ for different firing sequences

other high priority train $T_{j''}$ does not hamper the resource requirement of T_j:

$$P_{jik}|_{x_{ji}^{AT}} = 1,\ if\ and\ only\ if\ [Prio(T_j) > Prio(T_{j'})_{j\neq j',\ j\in[1,\ m]}] \wedge [Re(T_{j''})|^{\tau^B}_{Prio(T_{j''})>Prio(T_j)} \neq Re(T_j)|^{\tau^B}] \quad (9)$$

• *Case 1.1.1* After satisfying the condition described in case 1.1, if all the necessary resources are available throughout its journey, selecting any alternative path, then reroute train T_j. Route of T_j after departure from S_i at time x_{ji}^{DT} is $[Rou(T_j)|_{t\leq x_{ji}^{DT}} + Rou(T_j)|_{x_{ji}^{AT}>x_{ji}^{DT}}]$,
that is

$$if\ Rou(T_j)|_{x_{ji}^{AT}>x_{ji}^{DT}} = \bigwedge_{i'>i}^{n-1}(P_{ji'k}L_{ji'l}) \cup P_{jnk} = 0 \quad (10)$$

then reschedule T_j.

• *Case 1.1.2* If case 1.1.1 is invalid, then stop T_j at S_i until recovery is done or any other alternative path becomes free. Therefore, T_j occupies one of the platforms at S_i,
that is

$$P_{jik} = 1 \quad (11)$$

and

$$x_{ji}^{DT} = o_{ji}^{DT} + \delta_j \quad (12)$$

where

$$\delta_j = t_R - o_{ji}^{DT}\ and\ t_R = t_D + \tau_R$$

Case 1.2

If the scenario does not conform with case 1.1, then stop train T_j on the current track. Therefore, now T_j occupies lth track, that is

$$L_{jil} = 1 \quad (13)$$

and

$$x_{ji}^{AT} = o_{ji}^{AT} + \delta_j \quad (14)$$

The above described scenario depicted in Eqs. (8)–(14) is now represented in Petri-Net model in Fig. 4 and the corresponding description of respective places, transitions and tokens are described in Tables 5, 6, and 7.

Table 5 Description of places of $PN2$

Places (P)	Description
P1	T_j is on track l and approaching to disastrous station S_i
P2	$T_{j'}$ is on track l' and approaching to disastrous station S_i
P3	T_j and $T_{j'}$ both approaching to same station S_i
P4	T_j reaches to S_i and is at S_i
P5	T_j reaches its destination
P6	T_j completed its previous journey and started the next
P7	T_j is waiting for availability of resources

Table 6 Description of transitions of $PN2$

Transitions (Tr)	Description
Tr1	It will fire if platform is not available at S_i the time of arrival
Tr2	It will fire if platform is available at S_i the time of arrival
Tr3	It will fire if T_j gets highest priority
Tr4	It will fire if all resources are available to continue the journey
Tr5	It will fire if all resources are not available to continue the journey
Tr6	It will fire when T_j reaches its destination
Tr7	It will fire when T_j is ready for its new journey
Tr8	It will fire when T_j is not allowed to start its journey

Analysis of PN2:

Tables 5 and 6 presents the description of the places $P = \{P1, P2, P3, P4, P5, P6, P7\}$ and transitions $Tr = \{Tr1, Tr2, Tr3, Tr4, Tr5, Tr6, Tr7, Tr8\}$ and the initial marking is $M_0 = [1, 1, 0, 0, 0, 0, 0]$.

- Reachability graph analysis:
  Reachability graph analysis is the simplest method to analyse the behaviour of a Petri-Net. It decides whether the system is bounded and live or not. From our resultant tree in Fig. 4 it can be proved that: (a) the reachability set $R(M_0)$ is finite, (b) maximum number of tokens that a place can have is 2, so our $PN2$ is 2-bounded, (c) all transitions can be fired, so there are no dead transitions.
- State equation:
  The structural behaviour of the Petri-Net can be measured using the algebraic analysis of the incidence matrix. The incidence matrix contains the information about the structure of the Petri-Net. The information, such as connections between places and transitions, the number of total transitions can be represented through the incidence matrix. If marking M is reachable from initial marking M_0 through the transition sequence σ, then the following state equation holds: $\boldsymbol{M}_0 + [\boldsymbol{A}] \times X_\sigma = \boldsymbol{M}$.
  Incidence matrix is defined as $\boldsymbol{A} = [e_{uv}]$, it is a $r_A \times c_A$ matrix, where $(1 \le u \le r_A)$, $(1 \le v \le c_A)$. The order of the places in the matrix is $P = \{P1, P2, P3, P4, P5, P6, P7\}$, denoted by rows and the order of the transitions is $Tr = \{Tr1, Tr2, Tr3, Tr4, Tr5, Tr6, Tr7, Tr8\}$, denoted by columns.
  X_σ is an m-dimensional vector with its jth entry denoting the number of times transition t_j occurs in σ:

$$\boldsymbol{A} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}$$

Thus, if we view a marking $\boldsymbol{M}_0$ as a k-dimensional column vector in which the ith component is $M_0(p_i)$, each column of $[\boldsymbol{A}]$ is then a k-dimensional vector such that $\boldsymbol{M}_0 \xrightarrow{\sigma} \boldsymbol{M}$.

In our system, marking $\boldsymbol{M} = [1, 1, 0, 0, 0, 0, 0]$ is reachable from initial marking $\boldsymbol{M}_0 = [1, 1, 0, 0, 0, 0, 0]$ through the firing sequence $\sigma_1 = Tr2, Tr3, Tr4, Tr6, Tr7$.
$M_0 \xrightarrow{Tr2} M_1 \xrightarrow{Tr3} M_2 \xrightarrow{Tr4} M_3 \xrightarrow{Tr6} M_4 \xrightarrow{Tr7} M_5(= \text{‘}old'\text{)}$:

$$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Similarly, marking $\boldsymbol{M} = [1, 1, 0, 0, 0, 0, 0]$ is reachable from initial marking $\boldsymbol{M}_0 = [1, 1, 0, 0, 0, 0, 0]$ through the firing sequence $\sigma_2 = Tr1$ and $\sigma_3 = Tr2, Tr3, Tr5, Tr8, Tr4, Tr6, Tr7$.
$M_0 \xrightarrow{Tr2} M_1(= \text{‘}old'\text{)}$.
$M_0 \xrightarrow{Tr2} M_1 \xrightarrow{Tr3} M_2 \xrightarrow{Tr5} M_3 \xrightarrow{Tr8} M_2 \xrightarrow{Tr4} M_4 \xrightarrow{Tr6} M_5 \xrightarrow{Tr7} M_6(= \text{‘}old'\text{)}$:

$$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Table 7 Description of color tokens of $PN2$

Colour token (ct)	Description
ct1	P1 generates it when Eq. (8) is satisfied but T_j senses no free platform is available and enables transition Tr1
ct2	P1 generates it when Eq. (8) is satisfied but T_j senses free platform is available and enables transition Tr2
ct3	P2 generates it when Eq. (8) is satisfied but $T_{j'}$ senses no free platform is available and enables transition Tr1
ct4	P2 generates it when Eq. (8) is satisfied but $T_{j'}$ senses free platform is available and enables transition Tr2
ct5	P3 generates it to denote T_j has highest priority and enables transition Tr3
ct6	P4 generates it T_j senses all resources are available and enables transition Tr4
ct7	P4 generates it T_j senses that the resources are not available and enables transition Tr5

$$\begin{bmatrix}1\\1\\0\\0\\0\\0\\0\end{bmatrix}+\begin{bmatrix}0&-1&0&0&0&0&1&0\\0&-1&1&0&0&0&0&0\\0&1&-1&0&0&0&0&0\\0&0&1&-1&-1&0&0&1\\0&0&0&1&0&-1&0&0\\0&0&0&0&0&1&-1&0\\0&0&0&0&1&0&0&-1\end{bmatrix}\times\begin{bmatrix}0\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}=\begin{bmatrix}1\\1\\0\\0\\0\\0\\0\end{bmatrix}$$

5.2 Case 2: partial node deletion from the graph *G* with deletion of an edge

Currently multiple trains are on various platforms which follow a sequence according to their departure time within the buffer time τ^B. Here, τ^B is related to the disaster recovery time t_R of that station. The track is free but more than one trains are yet to come,

that is

$$L_{jil}=0 \tag{15}$$

and

$$\sum_{j=1}^{m} P_{jik}\le(p-1) \tag{16}$$

As the station S_i faces disaster, a particular kth platform can not be used until the recovery time has elapsed. If there is any incoming train T_j within buffer period τ^B, the system allows T_j to reach S_i if a platform is available, i.e. $P_{jik}=0$. The system also checks for the priority of T_j to reorder the departure schedule of all trains from S_i introducing delay δ_{ji} to T_j, if needed.

i.e. $\forall j$, if $Prio(T_{j'})>Prio(T_j)$

$$x_{ji}^{DT}=o_{ji}^{DT}+\delta_{ji}. \tag{17}$$

Otherwise, if all the resources are available for T_j and it has the highest priority among all the trains currently waiting at S_i, the scheduled departure of T_j is the original departure time as per the original railway timetable,
that is

$$x_{ji}^{DT}=o_{ji}^{DT} \tag{18}$$

if and only if $Prio(T_j)>Prio(T_{j'})_{j\neq j',\ j\in[1,\ m]}$

The scenario of Case 2 with Eqs. (15)–(18) is represented in Petri-Net model in Fig. 5 and the corresponding description of respective places, transitions and tokens are described in Tables 8, 9, and 10.

Analysis of PN3:

Tables 8 and 9 presents the description of the places $P=\{P1, P2, P3, P4, P5, P6\}$ and transitions $Tr=\{Tr1, Tr2, Tr3, Tr4, Tr5\}$ and the initial marking is $\boldsymbol{M}_0=[2, 3, 0, 0, 0, 0]$.

- Reachability graph analysis: Similarly, as discussed in Sect. 5.1, here initial marking M_0 is the root node as shown in Fig. 5. From our resultant tree it can be proved that : (a) the reachability set is $R(M_0)$ finite, (b) maximum number of tokens that a place can have is 3, so our $PN3$ is 3-bounded, (c) all transitions can be fired, so there are no dead transitions.
- State equation:
  The order of the places in the incidence matrix $\boldsymbol{A}$ is $P=\{P1, P2, P3, P4, P5, P6\}$, denoted by rows and the order of the transitions is $Tr=\{Tr1, Tr2, Tr3, Tr4, Tr5\}$, denoted by columns:

$$\boldsymbol{A}=\begin{bmatrix}-1&1&-1&0&0\\-1&0&0&0&0\\1&-1&0&0&0\\0&1&-1&0&0\\0&0&1&0&0\\0&0&0&1&1\end{bmatrix}$$

Here, marking $\boldsymbol{M}=[1, 2, 0, 1, 1, 0]$ is reachable from initial marking $\boldsymbol{M}_0=[2, 3, 0, 0, 0, 0]$ through the firing sequence $\sigma_1=Tr1, Tr2, Tr3, Tr4$. $M_0\xrightarrow{Tr1}M_1\xrightarrow{Tr2}M_2\xrightarrow{Tr3}M_3\xrightarrow{Tr4}M_4$:

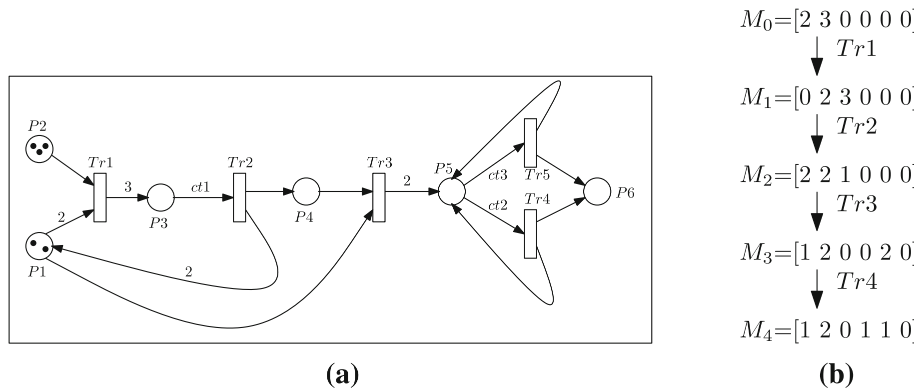


Fig. 5 Petri-Net $PN3$. **a** Petri-Net model $PN3$ of Case 2. **b** Reachability tree of $PN3$ for different firing sequences

Table 8 Description of places of $PN3$

Places (P)	Description
P1	Trains are at S_i, where disaster happens and only one platform is free
P2	Trains are at station connecting to S_i
P3	More than one trains are requesting for a single platform
P4	Highest priority train reaches to S_i
P5	All the trains are at S_i waiting to depart
P6	Highest priority train departs from S_i

$$\begin{bmatrix}2\\3\\0\\0\\0\\0\end{bmatrix} + \begin{bmatrix}-1 & 1 & -1 & 0 & 0\\ -1 & 0 & 0 & 0 & 0\\ 1 & -1 & 0 & 0 & 0\\ 0 & 1 & -1 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 1\end{bmatrix} \times \begin{bmatrix}1\\1\\1\\1\\0\end{bmatrix} = \begin{bmatrix}1\\2\\0\\1\\1\\0\end{bmatrix}$$

5.3 Case 3: extended impact of edge and node deletion

Train T_j is neither waiting at the station S_i, where disaster happened, i.e. $P_{jik} = 0$ nor on the connecting track, i.e. $L_{jil} = 0$. However, T_j reaches the station S_i within τ^B.

Case 3.1

Train T_j is at station $S_{i''}$, where $S_{i''}$ is in neighbourhood of S_i,

that is

$$P_{ji''k} = 1,\ S_{i''} \in S \setminus S_i \text{ and } i \in [1, n] \tag{19}$$

If any platform is available at the next station and the connecting track is also free, the system checks for the priority of the train T_j.

T_j maintains its original schedule if and only if it has the highest priority while reaching S_i, i.e., if

$$(P_{jik'} = 0) \wedge (L_{jil} = 0) \wedge (Prio(T_j)|_{t=x_{ji}^{AT}} > Prio(T_{j'})|_{j \neq j',\ j \in [1,\ m]}) \tag{20}$$

then

$$x_{ji}^{AT} = o_{ji}^{AT} \tag{21}$$

$$L_{jil} = 1 \text{ and } P_{jik'}|_{t=x_{ji}^{AT}} = 0 \tag{22}$$

$$P_{jik'}|_{t=x_{ji}^{AT}} = 1 \text{ and } L_{jil} = 0 \tag{23}$$

Here, damaged platform is k. $k' = \{1, 2, \ldots, p\} \setminus \{k\}$

Case 3.2

Train T_j is at $S_{i'}$, i.e. $P_{ji'k} = 1$, where $i, i' \in [1, n]$ and $i \neq i'$.

There are multiple tracks between two stations S_i and $S_{i'}$. i.e. $1 < l \leq 4$. If the track l breaks down due to disaster, it is assumed that track l is not free, that is

$$L_{ji'l} = 1,\ 1 \leq l < 4 \tag{24}$$

Table 9 Description of transitions of $PN3$

Transitions (Tr)	Description
Tr1	Only one platform is available at S_i and more than one train are approaching to S_i
Tr2	One platform is available and T_j has highest priority
Tr3	All the trains of are at S_i and requesting for same track to depart
Tr4	Reorder the scheduled trains as per priority
Tr5	Original departure schedule maintained as highest priority train is departing first as per original schedule

Table 10 Description of color tokens of $PN3$

Colour token (ct)	Description
ct1	P3 generates it to indicate that train T_j has the highest priority and enables transition Tr2
ct2	P5 generates it when reordering in train departure is decided and enables transition Tr4
ct3	P5 generates it if original ordering in departure schedule of trains are maintained and enables transition Tr5

Then, the trains which are scheduled to use that track face problem. In that case, first $S_{i'}$ checks for other available tracks, one of which can be allotted to T_j, provided T_j has the highest priority satisfying all the constraints and there is no resource conflict within τ^B.

$$[L_{ji'l'} = 0] \wedge [Prio(T_j)|_{t=x_{ji'}^{DT}} > Prio(T_{j'})|_{j \neq j',\ j \in [1,\ m]}] \wedge [Re(T_j)|_{x_{ji}^{AT}} \neq Re(T_{j'})|_{x_{ji}^{AT}}] \tag{25}$$

Figure 6 represents Petri-Net model for the scenario of Case 3, described in Eqs. (19)–(25) and the corresponding description of respective places, transitions and tokens are described in Tables 11, 12, and 13.

Analysis of PN4:

Tables 11 and 12 presents the description of the places $P = \{P1, P2, P3, P4, P5, P6\}$ and transitions $Tr = \{Tr1, Tr2, Tr3\}$ and the initial marking is $\boldsymbol{M}_0 = [2, 1, 0, 0, 1, 0]$.

– Reachability graph analysis:

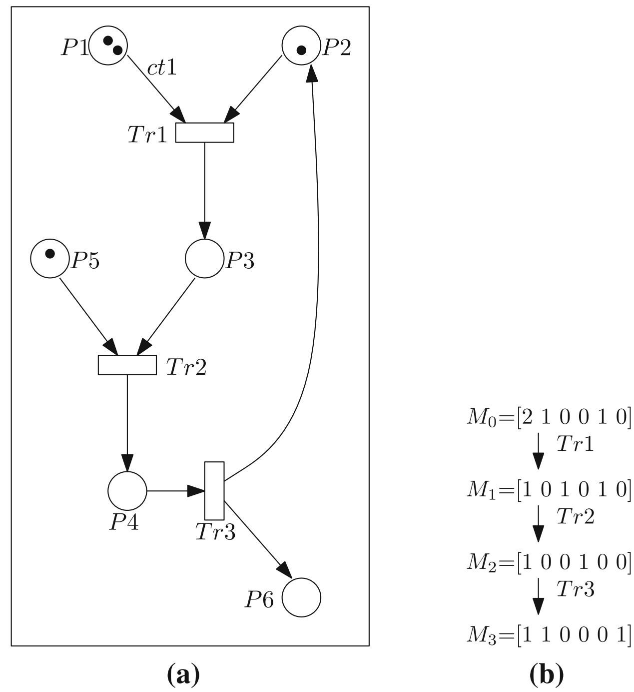


Fig. 6 Petri-Net $PN4$. **a** Petri-Net model $PN4$ of Case 3. **b** Reachability tree of $PN4$ for different firing sequences

Table 11 Description of places of $PN4$

Places (P)	Description
P1	T_j and $T_{j'}$ are at station $S_{i'}$, connecting to station S_i, where disaster happens
P2	Any one of the connecting track is free
P3	Trains are ready to leave
P4	Highest priority train T_j is running on the track
P5	Platform is free at S_i
P6	Highest priority train T_j reaches station S_i

Table 12 Description of transitions of $PN4$

Transitions (Tr)	Description
Tr1	It will fire if P1 has more than one tokens and generates ct1 and there is also a token available in P2
Tr2	It will fire if T_j has finished its waiting time at station $S_{i'}$ and a token is available at P5
Tr3	It will fire if a token is available at P4 indicating T_j is moving forward to S_i

Table 13 Description of color tokens of $PN4$

Colour token (ct)	Description
ct1	P1 generates it to indicate T_j has the highest priority while at $S_{i'}$ and enables transition Tr1

Table 14 Summary of disaster handling cases described in Sect. 5

Case no.	Description	Decision variable(s)	Decision taken
1	Station S_i faces problem and train T_j is on track l	L_{jil}	Reroute or retime
1.1	Station S_i has free platforms when T_j reaches	$P_{jik}, Prio(T_j), Re(T_j)$	Reroute or retime from station
1.1.1	$Re(T_j)$ is available after S_i	$P_{jik}, L_{jil}, x_{ji}^{DT}, x_{ji}^{AT}$	Reroute from S_i
1.1.2	No alternative route found for T_j from S_i	x_{ji}^{DT}	Delay at S_i
1.2	Station S_i has no free platforms when T_j reaches	P_{jik}, x_{ji}^{AT}	Stop on track l, retime
2	Number of trains are about to depart from S_i within buffer time τ^B	$L_{jil}, P_{jik}, Prio(T_j), x_{ji}^{DT}$	Reorder
3	Train T_j neither waits at affected station S_i nor on track l connected to S_i, but reaches to S_i within τ^B	$L_{jil}, P_{jik}, Prio(T_j), x_{ji}^{AT}$	Retime

As discussed in Sect. 5.1, initial marking M_0 is the root node as shown in Fig. 6. Again, (a) the reachability set is $R(M_0)$ finite, (b) maximum number of tokens that a place can have is 2, so our $PN4$ is 2-bounded, (c) all transitions can be fired, so there are no dead transitions.

– State equation:
Here, in the incidence matrix $\boldsymbol{A}$, $P = \{P1, P2, P3, P4, P5, P6\}$, denoted by rows. and the order of the transitions is $Tr = \{Tr1, Tr2, Tr3\}$, denoted by columns:

$$\boldsymbol{A} = \begin{bmatrix} -1 & 0 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In our system, marking $\boldsymbol{M} = [1, 1, 0, 0, 0, 1]$ is reachable from initial marking $\boldsymbol{M}_0 = [2, 1, 0, 0, 1, 0]$ through the firing sequence $\sigma_1 = Tr1, Tr2, Tr3$.
$M_0 \xrightarrow{Tr1} M_1 \xrightarrow{Tr2} M_2 \xrightarrow{Tr3} M_3$. :

$$\begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

5.4 Delay handling

Delay minimisation can be formulated as:

- Delay minimisation at station S_i (δ_{ji}^{min}).
- Delay minimisation on the track l (δ_{jl}^{min}).

$\boldsymbol{\delta_{ji}^{min}}$: This aims to minimise the delay in such a way that even if the train T_j comes late, it should try to minimise the deviation from scheduled departure time, that is

$$x_{ji}^{AT} \geq o_{ji}^{AT} \tag{26}$$

$$x_{ji}^{DT} = o_{ji}^{DT} \tag{27}$$

Therefore, it compromises dwell time of T_j at S_i, that is

$$x_{ji}^{d} < o_{ji}^{d} \tag{28}$$

$\boldsymbol{\delta_{jl}^{min}}$: This aims to minimise the delay considering the journey time (from source to destination), that is

$$x_{jl}^{J} = o_{jl}^{J} \tag{29}$$

In light of the aforementioned situations, the proposed rescheduling approach aims to minimise the total delay of trains in case of any disaster while rescheduling (Table 14). The objective function can be formulated as

$$\begin{aligned} min[\sum_j \delta_j] &= min\left[\sum_{Rou(T_j)} (\delta_{ji}^{min} + \delta_{jl}^{min})\right] \\ &= min\left[\sum_{P_{jik}} \delta_{ji}^{min}\right] + min\left[\sum_{L_{jil}} \delta_{jl}^{min}\right] \end{aligned} \tag{30}$$

6 Results and discussion

This section shows the advantages of our proposed model for dynamic train rescheduling. To evaluate the performance of the proposed approach, experiments are conducted on the same railway network in various scenarios with different combination of tracks, trains and stations. Both the delay on track and at station are considered in the experiments. For the experimental purpose real-time data set from Indian Railway (Eastern Zone) has been taken. The primary goal of this proposed approach is to optimize the journey time delay of trains while rescheduling them in a disastrous situation.

6.1 Experimental setup

To evaluate the performance of the proposed approach, the simulation is coded in Java in JADE[1] in UNIX platform of personal computer with 2.90 GHz processor speed and 4 GB memory. The results and computations are evaluated under same running environment.

Initially the railway database is populated with pre-defined details of all the station of the railway network. In this proposed work, it is assumed that the connectivity of any station with its neighbouring stations is established with a maximum number of 4 tracks. However, for some part of the network station are connected with 1 track. Each station is having maximum of $p(=6)$ number of platforms. Whenever a disaster happens, either a track or a platform, or both of them get affected. in this situation, the disruption effect is simulated by deleting it from the database, which decreases the number of links or platform index from the resource set. The permissible threshold delay is taken as 30 min and is varied within the range of 30–90 min. This is done to analyse the total delay variation with the change in threshold delay values. It is also assumed that all the stations are equidistant and all the trains are running at an average constant speed throughout its journey. In this paper, each of the stations and trains are considered as an autonomous agent (station agent and train agent, respectively). These agents are considered homogeneous due to no significant differences in their capabilities. Our proposed approach also handles the consequences of any disturbance occurred in railway network. This facilitates the models to simulate the effectiveness of the various disturbances in the system.

6.2 Illustration

The proposed approach is illustrated taking 10 different scenarios (see Table 15) at different clock time throughout the day. As a disaster can happen at any point of time, impacts are different on the delay of trains. For example, if a disaster

Table 15 Scenarios taken for experiments

Scenario	Time of disaster	Affected trains	Total delay Δ (in h)				
			$\delta_{Th} = 30$ min	$\delta_{Th} = 40$ min	$\delta_{Th} = 45$ min	$\delta_{Th} = 60$ min	$\delta_{Th} = 90$ min
1	8:45	T_1, T_2, T_4	2.375	2.041	1.75	1.416	1.166
2	9:10	T_2, T_3	3.00	2.875	2.417	2.083	1.75
3	10:30	$T_4, T_5, T_6, T_7, T_{10}$	22.667	21.833	21.083	20.25	18.541
4	12:00	T_7, T_{20}	4.25	4.333	3.875	3.375	2.875
5	16:30	T_8, T_9, T_{15}	2.375	2.166	1.875	1.625	1.333
6	17:30	$T_1, T_3, T_6, T_{12}, T_{15}, T_{19}$	29.458	28.583	24.083	21.708	20.50
7	18:20	T_8, T_9, T_{10}, T_{11}	3.875	3.375	2.75	2.166	2.041
8	18:50	T_3, T_{18}	0.667	0.667	0.667	0.417	0.417
9	19:40	$T_1, T_3, T_4, T_6, T_{15}, T_{19}$	39.916	37.041	34.33	32.458	30.166
10	00:15	T_1, T_6, T_{11}, T_{16}	9.75	8.375	7.708	6.916	5.083

[1] http://jade.tilab.com/.

happens at busy time, the chances of total system delay are higher as more number of trains gets affected. Similarly, the railway system faces less impact if disaster happens at normal time, when there are less number of trains in circulation.

Table 15 describes such scenarios, where the affected trains are listed and their total delays are also tabulated. Among the ten scenarios, scenario 2, 3 6, 7, and 8 are considered as 'busy- time' scenarios as discussed in Sect. 3 and rest are considered to be 'normal-time' scenarios. The trains, affected by the disaster, are enlisted in column 3. While calculating the total delay of the these trains, the delay throughout its journey is considered. each value under the delay column in the corresponding table denotes the total delay of all the trains, which are rescheduled. For example, the Δ value 2.375 when δ_{Th} is 30 min, is the total delay of all the three trains T_1, T_2, T_4 those get affected when a disaster happens at 08:45 am and likewise. It is clear from the table that, by varying threshold delay (δ_{Th}), keeping the scenario and all other constraints intact, the total journey time delay can be minimized while rescheduling the trains.

The simulation has been performed for a time period of 24 h and the achieved results are represented in Figs. 7 and 8. The threshold delay related to each of the disaster cases can be modulated to visualize the impact on the affected trains. It is also noticed from the simulated results that the trains can be influenced based on time (busy time and/or normal time) which can be considered as the total delays.

The results achieved from the proposed method have been compared with the existing centralised approach used in Indian Railway to highlight the advantages of this work. In currently used method, the central authority of the Indian Railway is responsible for taking all scheduling and rescheduling decisions, even at the dynamic scenarios. Each time all the feasible solutions are checked hierarchically to reach to the best possible solution. Then the decision is passed to the stations whenever needed by the technique of message passing. This is a troublesome job and time-consuming and, moreover, takes huge number of message passing to be occurred for a single event. Again, this method complies the disadvantages of centralized network management approach like single-point failure which can make the system vulnerable and the huge time requirement to achieve the results raises safety issues in the existing system in critical scenarios. On contrary, the proposed approach in this research paper uses an distributed approach which faces less message passing and the autonomous agents can take decisions by themselves as and when needed. This reduces the cost of message passing and also the time delay of overall system making such safety-critical system reliable. Hence the lesser delay is observed by the proposed approach as illustrated in Fig. 9.

In Table 16 the comparative study of total average delay Δ and average execution time $\mathcal{T}_e$ between proposed method and some existing methods are enlisted. The analysis shows

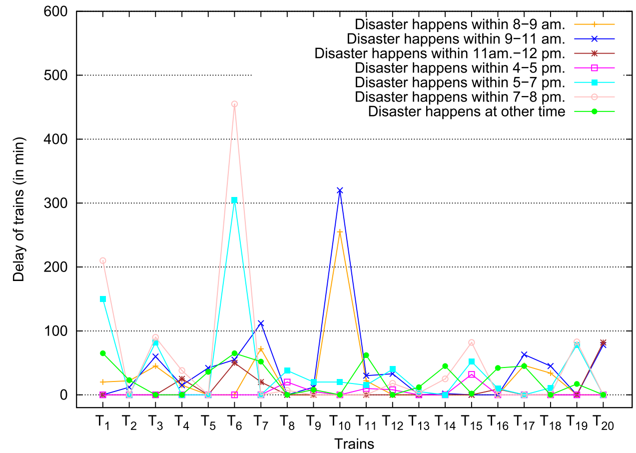


Fig. 7 Delay of trains when disaster happens in busy times and normal times.

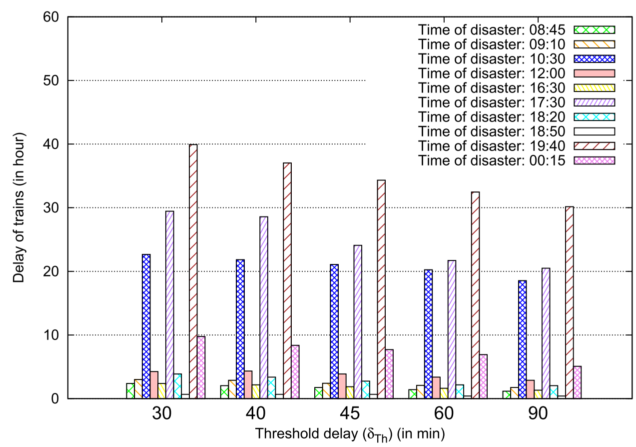


Fig. 8 Delay of affected trains when threshold delay varies.

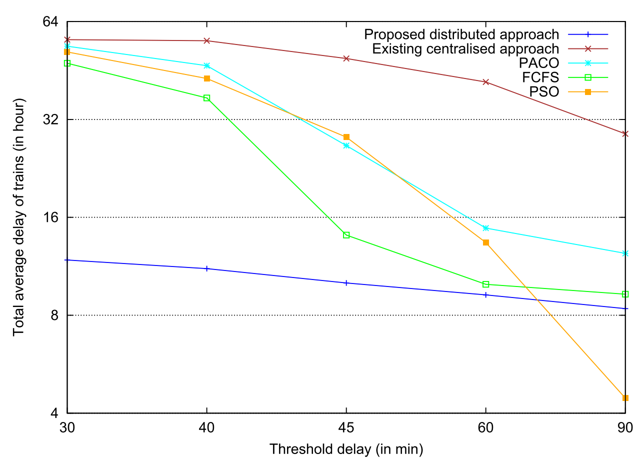


Fig. 9 Comparison of average total delay between proposed distributed and existing approaches

Table 16 Comparison of total delay Δ and computation time $\mathcal{T}_e$ for various threshold delay δ_{Th}

δ_{Th} (in min)	Proposed		Existing centralised		PACO		FCFS		PSO	
	Δ	$\mathcal{T}_e$	Δ	$\mathcal{T}_e$	Δ	$\mathcal{T}_e$	Δ	$\mathcal{T}_e$	Δ	$\mathcal{T}_e$
30	11.83	3.24	56.29	48.81	53.73	34.61	47.65	35.88	51.61	33.26
40	11.12	3.20	55.87	48.23	46.82	33.76	37.27	32.29	42.77	32.57
45	10.05	3.01	49.28	46.95	26.60	18.84	14.11	17.54	28.33	21.42
60	9.24	2.98	41.72	42.22	14.83	17.92	9.96	4.98	13.39	2.27
90	8.38	2.77	28.91	40.76	12.40	14.11	9.29	4.83	4.45	2.14

that our proposed method performs far better than the existing centralised method used in current railway managements system. Moreover, the proposed approach shows better results in terms of average delay than the other algorithmic approaches, such as PACO, FCFS, and PSO. Though in some cases, where threshold delay amount is larger (90 min), PSO showed better execution time, but on average the proposed distributed method outperforms in all scenarios.

Figure 10 depicts that our proposed method is faster than the other state-of-the-art methods, such as PACO (Eaton et al. 2017), FCFS (Gafarov et al. 2015), PSO (Yaman et al. 2018). This study is performed with Petri-Net for 20 trains for a simulation over 24 h time period.

The entire experimental study has been performed based on the data sheet taken from Indian Railway (eastern Zone) for a time frame of 45 days for the railway network considered in this research work.

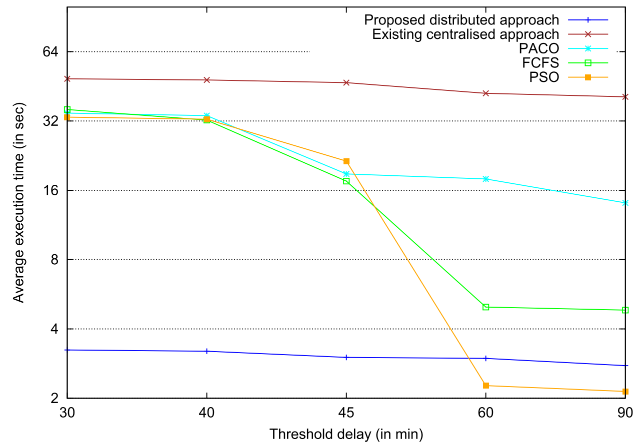


Fig. 10 Comparison of average execution time between proposed distributed and existing approaches

7 Conclusions

This research work presents a color Petri-Net- based distributed constraint optimization approach for railway timetable rescheduling when any disaster occurs. Authors considered multiple tracks and platforms for train scheduling problem. The main aim of this paper is to minimize the total journey time delay of the trains which are in circulation in railway network. The deviation between original and actual arrival and/or departure time is considered as delay of individual train. In a disastrous situation, the uncertainties in such system is also taken care of with the notion of Markov decision process (MDP) which handles the dynamic changes of the system constraints. For solving larger instances with large amount of threshold delay (upto 90 min) and 20 trains are taken in the proposed approach. A color Petri-Net here to resolve the conflict situation in scheduling problem. It is observed from the rigorous experiments on the real data set, taken from Indian Railway, that our proposed approach outperforms the existing approaches in terms of efficiency and execution time.

The future direction of this research work focuses on rescheduling of trains including freight trains (non-passenger commercial services) and signalling constraints. In addition, the proposed can be applied to other transportation domain as well for rescheduling entities with necessary and minor changes in constraints as per the domain infrastructure.

Acknowledgements This research work is a part of the project [Project Id No.: 1-5728195821] named"Multi-Agent Based Modelling for Collision Handling and Delay Optimization in Indian Railway System"and funded by MHRD and World-bank under TEQIP Collaborative Research Scheme, Govt. of India.

Data availability statement All the data sets, used for the experiments are embedded in the manuscript. No third party data has been used in this work.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Abid MM, Khan MB (2015) Sensitivity analysis of train schedule of a railway track network using an optimization modeling technique. In: European transport research review, vol 7. Springer, pp 2–7

Abourraja MN, Oudani M, Samiri MY, Boudebous D, El Fazziki A, Najib M, Bouain A, Rouky N (2017) A multi-agent based simulation model for rail-rail transshipment: an engineering approach for gantry crane scheduling. IEEE Access 5:13142–13156

Acuna-Agost R, Feillet D, Gueye S, Michelon P (2011a) A mip-based local search method for the railway rescheduling problem. In: Networks, vol 57. Wiley Periodicals, Inc, pp 69–86

Acuna-Agost R, Michelon P, Gueye S (2011) Sapi: statistical analysis of propagation of incidents. A new approach for rescheduling trains after disruptions. Eur J Oper Res 215:227–243

Alwadood Z, Shuib A, Hamid NA (2012) A review on quantitative models in railway rescheduling. Int J Sci Eng Res 3:1–7

Atlas J, Decker K (2007) A complete distributed constraint optimization method for non-traditional pseudotree arrangements. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems (AAMAS), p 111

Beg S, Khan A, Nauman U, Mohsin S (2011) Performance evaluation of bionomic algorithm (ba) in comparison with genetic algorithm (ga) for shortest path finding problem. Int J Comput Sci Issues (IJCSI) 8(6):238

Bernstein DS, Givan R, Immerman N, Zilberstein S (2002) The complexity of decentralized control of Markov decision processes. Math Oper Res INFORMS 27(4):819–840

Bernstein DS, Amato C, Hansen EA, Zilberstein S (2009) Policy iteration for decentralized control of Markov decision processes. J Artif Intell Res SCImago 34:89–132

Cavone G, Dotoli M, Epicoco N, Seatzu C (2017) Intermodal terminal planning by petri nets and data envelopment analysis. Control Eng Pract 69:9–22

Chaki N, Bhattacharya S (2006) Performance analysis of multistage interconnection networks with a new high-level net model. J Syst Archit ELSEVIER 52:57–70

Chalumuri RS, Yasuo A (2014) Modelling travel time distribution under various uncertainties on Hanshin expressway of Japan. Eur Transp Res Rev 6:85–92

Dalapati P, Singh JA, Dutta A, Bhattacharya S (2014) Multiagent based railway scheduling and optimization. IEEE TENCON, Bangkok

Dong H, Ning B, Chen Y, Sun X, Wen D, Hu Y, Ouyang R (2012) Emergency management of urban rail transportation based on parallel systems. IEEE Trans Intell Transp Syst 14(2):627–636

Eaton J, Yang S, Mavrovouniotis M (2016) Ant colony optimization with immigrants schemes for the dynamic railway junction rescheduling problem with multiple delays. Soft Comput 20(8):2951–2966

Eaton J, Yang S, Gongora M (2017) Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling. IEEE Trans Intell Transp Syst 18(11):2980–2992

Feynman R, Vernon Jr F (2013) A mip-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers. J Rail Transp Plan Manag 38–53

Gafarov ER, Dolgui A, Lazarev AA (2015) Two-station single-track railway scheduling problem with trains of equal speed. Comput Ind Eng 85:260–267

Gaied M, Lefebvre D, M'halla A, Othmen KB (2018) Modelling and performance evaluation of railway transport systems using p-timed petri nets. In: 2018 5th international conference on control. decision and information technologies (CoDIT). IEEE, pp 841–846

Gaied M, M'halla A, Lefebvre D, Ben Othmen K (2019) Robust control for railway transport networks based on stochastic p-timed petri net models. Proc Inst Mech Eng Part I J Syst Control Eng 233(7):830–846

Giglio D, Sacco N (2016) A petri net model for analysis, optimisation, and control of railway networks and train schedules. In: 2016 IEEE 19th international conference on intelligent transportation systems (ITSC). IEEE, pp 2442–2449

Hancerliogullari G, Rabadi G, Al-Salem AH, Kharbeche M (2013) Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem. J Air Transp Manag 32:39–48

Hassan GM, Reynolds M (2018) Genetic algorithms for scheduling and optimization of ore train networks. In: GCAI, pp 81–92

He D, Lu G, Yang Y (2019) Research on optimization of train energy-saving based on improved chicken swarm optimization. IEEE Access 7:121675–121684

Jensen K, Kristensen LM (2009) Coloured petri nets, modelling and validation of concurrent systems. Springer. https://doi.org/10.1007/b95112

Kersbergen B, van den Boom T, Schutter BD (2013) Reducing the time needed to solve the global rescheduling problem for railway networks. In: Proceedings of the 16th international IEEE annual conference on intelligent transportation systems (ITSC 2013), pp 791–796

Krasemann J (2010) Greedy algorithm for railway traffic re-scheduling during disturbances: a Swedish case. IET Intell Transport Syst 4:375–386

Leurent F (2011) Transport capacity constraints on the mass transit system: a systemic analysis. Eur Transp Res Rev 3:11–21

Li X, Shou B, Ralescu D (2014) Train rescheduling with stochastic recovery time: a new track-backup approach. IEEE Trans Syst Man Cybern Syst 44(9):1216–1233

Li W, Yan S (2019) Research on mine locomotive scheduling model and deadlock based on petri net. In: 2019 international conference on modeling, analysis, simulation technologies and applications (MASTA 2019). Atlantis Press

Meng L, Zhou XS (2011) Robust single-track train dispatching model under a dynamic and stochastic environment: a scenario-based rolling horizon solution approach. Transport Res Part B Methodol 45:1080–1102

Meng L, Zhou X (2014) Simultaneous train rerouting and rescheduling on an n-track network: a model reformulation with network-based cumulative flow variables. Transport Res Part B 67:208–234

Mukherjee S, Basu K (2011) Solution of interval pert/cpm network problems by a simplified tabular method. Opsearch 48(4):355–370

Murata T (1989) Petri nets: properties, analysis and applications. Proc IEEE 77:541–580. https://doi.org/10.1109/5.24143

Nguyen DT, Yeoh W, Lau HC (2012) Stochastic dominance in stochastic dcops for risk-sensitive applications. In: Proceedings of the 11th international conference on autonomous agents and multiagent systems, vol 1, pp 257–264

Nguyen TT, Yao X (2009) Benchmarking and solving dynamic constrained problems. In: IEEE congress on evolutionary computation, 2009. CEC'09. IEEE, pp 690–697

Salehipour A, Modarres M, Naeni LM (2013) An efficient hybrid meta-heuristic for aircraft landing problem. Comput Oper Res 40(1):207–213

Takouna I, Rojas-Cessa R (2008) Routing schemes for network recovery under link and node failures. IEEE, pp 69–73

Talal R (2014) Comparative study between the (ba) algorithm and (pso) algorithm to train (rbf) network at data classification. Int J Comput Appl 92(5):16–22

Utomo T, Arfi E (2019) The delay simulation on hierarchical structure for semi-double track of railway line using max-plus algebra. J Sci Appl Technol 2(1):96–99

Vidal JM (2010) Fundamentals of multiagent systems

Wang P, Ma L, Goverde RMP, Wang Q (2016) Rescheduling trains using petri nets and heuristic search. IEEE Trans Intell Transp Syst 17:726–735

Wang P, Fang W, Guo B (2019) Mental workload evaluation and its application in train driving multitasking scheduling: a timed petri net-based model. Cogn Technol Work 1–15

Wooldridge M (1966) An introduction to multiagent systems

Yaman O, Karakose E, Karakose M (2018) Pso based traffic optimization approach for railway networks. In: 2018 international conference on artificial intelligence and data processing (IDAP). IEEE, pp 1–4

Yang L, Gao Z, Li K (2011) Railway freight transportation planning with mixed uncertainty of randomness and fuzziness. Appl Soft Comput 11(1):778–792

Yen HC (2006) Introduction to petri net theory. Studies in computational intelligence. Springer, p 25

Yianni PC, Rama D, Neves LC, Andrews JD, Castlo D (2017) A petri-net-based modelling approach to railway bridge asset management. Struct Infrastruct Eng 13(2):287–297