Real-time Rescheduling in Distributed Railway Network: An Agent-Based Approach

Poulami Dalapati¹,², Piyush Agarwal¹, Animesh Dutta¹, Swapan Bhattacharya²

¹Dept of Information Technology, National Institute of Technology Durgapur, India
²Dept of Computer Science and Engineering, Jadavpur University, Kolkata, India

E-mail: dalapati89@gmail.com

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 schedule so as 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 times 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 actual schedule of the Eastern Railways, India by constructing various disaster scenarios using the Java Agent DEvelopment Framework (JADE). When compared to the existing approaches, the proposed framework substantially reduces the delay of trains after rescheduling.

1 Introduction

The railway system is a major mode of transport which is geographically distributed throughout the country [1]. The construction of schedules for trains in such system [2–6] in an efficient and optimised manner is a challenging task with considerations like situational complexity of the network and the enormous constraints that have to be handled. Some of these constraints are availability of tracks between stations and availability of platforms on those stations, which influence the arrival and departure time of trains. Moreover, any disruption in railway network [1, 7] due to natural calamities, sabotage, temporary platform blockage and accident on track(s) or platform make the offline schedule sub-optimal for use. Affected trains need to be rescheduled dynamically to minimise the impact of such disruptions. Where not only the objective function changes over time but the constraints can also transform [8–9]. This uncertainty and dynamic constraints make the global optima less effective. Therefore, the entire scheduling problem is considered as an agent based Distributed constraint optimization problem (DCOP) [10–11], where all agents cooperate with each other using commonly agreed protocols and constraints. The uncertainty of recovery time and its probabilistic nature is represented mathematically in terms of Markov decision processes (MDP) [12–14]. Here, each node is considered as a possible state of disaster scenario in railway network. The state transition functions are mapped to the constraints of DCOP, where each agent chooses its action to minimise its expected delay based on its policy. The action of the set of all agents in an MDP setting is to find the optimal solution which minimises the total delay of the railway network with the constraints in place.
In this paper, a framework with multiple trains, stations and tracks is considered where some of the trains are on tracks and some others are at stations, as shown in Figure 1a. In case of a disaster, in and around a station, the station authorities inform the neighbouring stations, the incoming and the outgoing trains. According to the proposed approach, each train and station agent checks for disaster recovery time and resources available to reach the destination. If the disaster recovery time does not affect the scheduled arrival or departure time of trains, then original schedule is maintained. Otherwise, the proposed rescheduling method, as described in section 6, is used to generate a new dynamic schedule and trains are informed.

The rest of the paper is organised as follows: In section 2 some previous works in related domain are discussed. Section 3 is devoted to the description of railway network and scheduling topology. Section 4 models the system. DCOP and MDP representations of the system are depicted in section 5. Disaster handling and rescheduling approach is formulated in section 6. The simulation results are evaluated in section 7. Finally section 8 concludes the propose work with future scope.

2 Literature Reviews

The existing approaches mainly consider line topology and network topology while rescheduling trains under disturbance [15-18]. The disturbance scenarios are modelled as certain and uncertain [19] based on its recovery time. Further decision scenarios for rescheduling are classified as retiming, rerouting or reordering [20,21] with respect to delay management of passenger railway services [16,22,23].

A rescheduling system in tuberail trains dispatching problem is proposed in [24], which focuses on freight gross transport [19] in time, but not the delay time of passengers in railway and subway. Here, the operation is centralised and controlled by operation support system which communicates with all trains, turnouts and handling equipment, so that all trains are operated in global information condition. A track-backup rescheduling approach [2] is proposed to minimise the negative effects arising from the disturbances, which optimally assigns a backup track to each affected train, based on original timetable, estimated recovery time, and track changing cost in line topology. In [25], a heuristic-based mixed-integer linear programming model is proposed to tackle delay propagation in traffic disturbances. This model is robust to its configuration provided an appropriate selection of boosting method is performed. A rescheduling model for last trains with the consideration of train delays caused by incidents that occurred in train operations is also discussed in [26]. Here authors aim to minimise running-time, dwell-time (as defined in [27]), and differences between rescheduled and original timetable and maximise average transfer redundant time. Similarly in [20], a proactive rerouting mechanism is proposed to minimise computational overhead and congestion where links are affected by failures. An agent based game theoretic coalition formation model is proposed in [28] to re-optimise a railway timetable. Train rerouting on an N-track railway network [21] and robust railway station planning [29] as well as optimisation in multi-train operation in subway system [30] are also proposed to improve the robustness of rescheduling process in the complex scenario.

Although the train rescheduling problem has been widely studied, most of the previous work consider either a centralised approach or line topology. However, a real time railway network is distributed in nature with dynamic entities and disruption can happen anytime. Authorities have to take decisions on the basis of real world scenario and concurrent decisions have to be made for efficient handling of the problem. Moreover, recovery time of such disturbances is highly uncer-
tain [31]. The distributed nature of the system can be effectively represented as DCOPs [10, 11] and probabilistic decision making can be mapped with MDP [12], where the outcomes are partly random and partly under the control of decision maker. Again, the inherent dynamic nature and concurrent decision making can be suitably modelled using Petri-Net [32–36].

In light of the discussion above, our main contributions in this paper 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 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 Railway Network and Scheduling Topology

A railway network consists of Stations ($S$), Trains ($T$) and Tracks as shown in Figure 1a. Multiple trains are either at stations or running on tracks at time instant $t$. Railway scheduling and rescheduling is performed in such a way that, the highest priority train is rescheduled first. Depending on the scenario, the priority changes dynamically reducing total journey time delay.

3.1 Assumptions

- There can be multiple tracks between two stations.
- Each station may consist of multiple platforms.
- Stations can communicate with trains and neighbouring stations.
- Trains can communicate with 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

3.2.1 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.

So, for generalisation, dwell-time [27] is taken as a parameter. If dwell-time of train $T$ at any stations $S$ is greater than zero, then $S$ is a stopping station for $T$. 
3.2.2 Tracks \((T)\)
As depicted in Figure 1b,

- 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.

3.2.3 Trains
In a railway network \([1]\), 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 schedule (i.e. office hours), \(T^S\) 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 like, \(y_1 = Prio(T^{Pr}), y_2 = Prio(T^M), y_3 = Prio(T^F), y_4 = Prio(T^P), y_5 = Prio(T^{Lo})\). This priority allocation policy is defined by the existing railway system of the region, considered in the experiments as described in subsection 7.1. In general, \(y_1 > y_2 > y_4 > y_5 > y_3\). 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 like, \(y_4 \geq y_1 \geq y_5 \geq y_2 \geq y_3\).

There are two kinds of inputs in our system, static input, which is pre-planned as per the schedule and 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. 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 $\tau^B$.

4 Railway Architecture Model

Given this background, a Multi-agent System (MAS) \cite{12, 37} 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 Figure 1a) 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 $S\alpha_a \in SA$ and $T_j \in T$ with $T\alpha_a \in TA$.

Table 1 Notation

| Indices and Parameters | | |
|---|---|---|
| Stations | $\sigma_j^T$ | Journey time of train $j$ in original timetable |
| Trains | $\sigma_j^L$ | Original dwell time of train $j$ at station $S_i$ |
| $i$ | $x_{ij}^T$ | Journey time of train $j$ on track $l$ |
| Station index | | |
| $j$ | | |
| Train index | | |
| $l$ | $q$ | Number of agents, where $q = m + n$ |
| Track index | | |
| $k$ | $t$ | Time instant |
| Platform index | | |
| $n$ | $t_D$ | Time of disaster |
| Number of stations | | |
| $m$ | $t_R$ | Time of recovery |
| Number of trains | | |
| $p$ | $t_{busy}$ | Busy Time Period of the day |
| Maximum number of platforms at each station | | |
| $\phi(x^T_j)$ | $i' \in [1,n]\setminus i$ | Index of station other than the $j^{th}$ station. |
| Arrival time of train $j$ at station $i$ in original timetable | | |
| $\phi(x^T_j)$ | $j' \in [1,m]\setminus j$ | Index of train other than the $j^{th}$ train. |
| Departure time of train $j$ from station $i$ in original timetable | | |
| $\delta_{j_i}$ | $i'' \in [1,n]\setminus i, i'$ | Index of station other than the $i^{th}$ and $i''^{th}$ station. |
| Delay of train $j$ at station $i$ | | |
| $\delta_{j_l}$ | $j'' \in [1,m]\setminus j, j'$ | Index of train other than the $j''^{th}$ and $j^{th}$ train. |
| Delay of train $j$ on track $l$ | | |
| $\tau^B$ | $\tau^B$ | Buffer Time, where $t_D + \tau_1 \leq \tau^B \leq t_D + \tau_2$ |
| Threshold value for delay of all trains | | |

| Decision Variables | | |
|---|---|---|
| Arrival time of train $j$ at station $i$ due to disaster | $x_{ij}^L$ | Departure time of train $j$ from station $i$ due to disaster |
| $P_{ijk}$ | $L_{ijl}$ | Track indicator, $L_{ijl} = 1$ if train $j$ occupies $l^{th}$ track connecting to station $i$, otherwise 0 and when $L_{ijl} = 1$, $L_{ijl} = 0$ |
| Platform indicator, $P_{ijk} = 1$ if train $j$ occupies $k^{th}$ platform of station $i$, otherwise 0 | | |
| Priority of train $T_j$ | $x_{ij}^L$ | Actual operation time of train $j$ at station $S_i$ |
| $Prio(T_j)$ | $\tau_2$ | Maximum time required to recover from the disaster |
| $\tau_1$ | | |
| Minimum time required to recover from the disaster | | |
| | Time to recover with the density function $\phi(x)$, where $x \in [\tau_1, \tau_2]$ | |

5
4.1 Petri-Net Model of Railway System

Now we introduce the general concepts of Petri-Net [32, 33] describing a railway network. Major use of various kinds of Petri-Net [34–36] 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 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.

\(Tok = Tok^c \cup Tok^{Ag}\) and \(Tok^c \cap Tok^{Ag} = \phi\), where \(Tok^c\) is the set of colour 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 colour token or perform some operations.

\(M_0\) : Initial Marking of Petri-Net.

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

\(\beta\) says that colour token is only used for taking a decision to resolve conflict. It won’t propagate to next state.

In our Petri-Net model, shown in Figure 2, place is represented by a circle, transition by a rectangle, and input and output flow by arrow and the corresponding description is given in Table 2.
| 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. |
| P12       | No free platform for $T_j$. |
| P13       | No free platform for $T_j'$. |
| P14       | $T_j$ and/or $T_j'$ reached to the station. |

| 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. |

| 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. |
5 DCOP and MDP Representation of the Proposed Rescheduling Approach

5.1 Representation as DCOP

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, \( X = \{ x_{ji}^{AT} \cup x_{ji}^{DT} \cup P_{jik} \cup L_{jil} \} \)
- **D**: Set of domains, \( D = t_D + \tau_R \)
- **C**: Set of constraints

5.1.1 Constraints (C)

- **Continuity Constraint**:
  \[
  x_{ji}^{AT} \geq x_{ji}^{DT} + x_{jl}^J
  \]
  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**:
  \[
  \begin{align*}
  x_{ji}^{AT} &\geq o_{ji}^{AT} \\
  x_{ji}^{AT} - o_{ji}^{AT} &\geq \delta_j
  \end{align*}
  \]
  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 \( \delta_j \). 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 \( p \).

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

and

\[
\sum_k P_{jik} \leq p, \text{ where } 1 \leq k \leq p
\]
If train $T_j$ occupies $l^{th}$ track, connecting to station $S_i$, then train $T_{j'}$ can not occupy the same track at the same time.

$$\text{if } L_{jil} = 1, \text{ then } L_{j'il} = 0$$ (5)

• 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.

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

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

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

5.2 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.

• Set of World State ($W$)

Here, $W$ 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. $W = \{(L_{jil} = 1), (P_{jik} = 1), (P_{jik'} = 1)\}$

• Transition Function ($\Psi$)

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 $\omega$ to the next state $\omega'$, where $\omega, \omega' \in W$.

If the train $T_j$ is on the $l^{th}$ 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. So, 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. So, state transition from current state $L_{jil} = 1$ is possible.

If $T_j$ is on $p^{th}$ platform of $S_i$ and the $l^{th}$ track is free but the platform at the next station $S_{i'}$ is not free, i.e. $P_{jik'} = 1$, no state change is possible from current state $P_{jik} = 1$. Similarly, even if there is free platform at $S_{i'}$, if the $l^{th}$ track is not free or both the track and platform is not available at time $t$, no state change is possible.

In Figure 3, every node represents a state and each arc represents an action which is indeed a constraint. The transition from one state to another state happens iff the corresponding agent satisfies the specific constraint(s). 

9
Figure 3. MDP representation with set of states and transition functions.

6 Disaster Handling and Rescheduling Model

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**).

6.1 **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 \( S_i \),
i.e.
\[
L_{jit} = 1 \quad \text{(8)}
\]

6.1.1 **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 \( S_i \), iff the priority of the incoming train \( T_j \) has the highest priority amongst all trains \( T \) and the resource (Re) required for any other high priority train \( T_{j''} \) does not hamper the resource requirement of \( T_j \).

\[
P_{jik} | x_{AT}^{ji} = 1, \text{ if } \left[ \text{Prio}(T_j) > \text{Prio}(T_{j'}) \right]_{j \in [1, m]} \land \left[ \text{Re}(T_{j''}) \right]_{\text{Prio}(T_{j''}) > \text{Prio}(T_j)} \neq \text{Re}(T_j) \text{ } \right\}^T \quad \text{(9)}
\]

6.1.1.1 **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 \( \left[ \text{Rou}(T_j) \right]_{t \leq x_{ji}^{DT}} + \text{Rou}(T_j) | x_{ji}^{AT} > x_{ji}^{DT} \]

i.e.
\[
\text{if } \text{Rou}(T_j) | x_{ji}^{AT} > x_{ji}^{DT} = \bigwedge_{i' > i}^{n-1} \left( P_{ji'k} L_{ji'lt} \right) \cup P_{jnk} = 0 \quad \text{(10)}
\]

then reschedule \( T_j \).
6.1.1.2 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. So, $T_j$ occupies one of the platforms at $S_i$.

i.e.

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

and

$$x_j^{DT} = o_j^{DT} + \delta_j \quad (12)$$

where,

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

6.1.2 Case 1.2

If the scenario does not conform with case 1.1, then stop train $T_j$ on the current track. So, now $T_j$ occupies $l^{th}$ track.

i.e.

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

and

$$x_j^{AT} = o_j^{AT} + \delta_j \quad (14)$$

The above described scenario depicted in equations (11)-(14) is now represented in Petri-Net model in Figure 4a and the corresponding description of respective places, transitions and tokens are described in Table 3.

Figure 4. Petri-Net $PN2$

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

Analysis of $PN2$:

Table 3 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 $M_0 = [1, 1, 0, 0, 0, 0, 0]$. 
Table 3 Description of PN2

| Places(P) | Description | | Description of Places of PN2 |
|---|---|---|---|
| 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. |

| Transitions(Tr) | Description | | Description of Transitions of PN2 |
|---|---|---|---|
| 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. |

| Colour Token(ct) | Description | | Description of colour tokens of PN2 |
|---|---|---|---|
| ct1 | P1 generates it when equation (8) is satisfied but $T_j$ senses no free platform is available and enables transition Tr1. |
| ct2 | P1 generates it when equation (8) is satisfied but $T_j$ senses free platform is available and enables transition Tr2. |
| ct3 | P2 generates it when equation (8) is satisfied but $T_j'$ senses no free platform is available and enables transition Tr1. |
| ct4 | P2 generates it when equation (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 all resources are available and enables transition Tr5. |

- **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 4b 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 by using the algebraic analysis of the incidence matrix. If marking $M$ is reachable from initial marking $M_0$ through the transition sequence $\sigma$, then the following state equation holds: $M_0 + [A] \times X_\sigma = M$. Incidence matrix is defined as $A = [e_{uv}]$, it is a $r_A \times c_A$ matrix where $(1 \leq u \leq r_A), (1 \leq v \leq c_A)$. 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. $X_\sigma$ is an m-dimensional vector with its $j^{th}$ entry denoting the number of times transition $t_j$
occurs in $\sigma$.

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

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

In our system, marking $M = [1, 1, 0, 0, 0, 0, 0]$ is reachable from initial marking $M_0 = [1, 1, 0, 0, 0, 0, 0]$ through the firing sequence $\sigma_1 = Tr2, Tr3, Tr4, Tr6, Tr7$.

$M_0 \xrightarrow{Tr2} M_1 \xrightarrow{Tr3} M_2 \xrightarrow{Tr4} M_3 \xrightarrow{Tr6} M_4 \xrightarrow{Tr7} M_5 (= 'old')$.

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

Similarly, marking $M = [1, 1, 0, 0, 0, 0, 0]$ is reachable from initial marking $M_0 = [1, 1, 0, 0, 0, 0, 0]$ through the firing sequence $\sigma_2 = Tr1$ and $\sigma_3 = Tr2, Tr3, Tr5, Tr8, Tr4, Tr6, Tr7$.

$M_0 \xrightarrow{Tr2} M_1 (= 'old')$.

$M_0 \xrightarrow{Tr2} M_1 \xrightarrow{Tr3} M_2 \xrightarrow{Tr5} M_3 \xrightarrow{Tr8} M_4 \xrightarrow{Tr4} M_5 \xrightarrow{Tr6} M_6 (= 'old')$.

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

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

### 6.2 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 $\tau^B$. Here, $\tau^B$ 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. i.e.

$$L_{jil} = 0 \quad (15)$$

and

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

As the station $S_i$ faces disaster, a particular $k^{th}$ platform can not be used until the recovery time has elapsed. If there is any incoming train $T_j$ within buffer period $\tau^B$, the system allows $T_j$ to reach $S_i$. 

13
if a platform is available, i.e. $P_{jk} = 0$. The system also checks for the priority of $T_j$ to reorder the departure schedule of all trains from $S_i$ introducing delay $\delta_{ji}$ to $T_j$, if needed. i.e. $\forall j$, if $Prio(T_j') > Prio(T_j)$

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

(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.

i.e.

$$x_{ji}^{DT} = o_{ji}^{DT}$$

(18)

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

The scenario of Case 2 with equations (15)-(18) is represented in Petri-Net model in Figure 5a and the corresponding description of respective places, transitions and tokens are described in Table 4.

**Figure 5. Petri-Net PN3**

a Petri-Net model $PN3$ of Case 2.

b Reachability tree of $PN3$ for different firing sequences.
Table 4 Description of PN3

| 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. |

| 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 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. |

| Description of colour 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. |

Analysis of PN3:
Table 4 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 \( M_0 = [2, 3, 0, 0, 0, 0] \).

- Reachability graph analysis:
  Similarly, as discussed in subsection 6.1.2 here initial marking \( M_0 \) is the root node as shown in Figure 5b. 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 \( A \) 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.

\[
A = \begin{bmatrix}
-1 & 1 & -1 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\]

Here, marking \( M = [1, 2, 0, 1, 1, 0] \) is reachable from initial marking \( M_0 = [2, 3, 0, 0, 0, 0] \) through the firing sequence \( \sigma_1 = Tr1, Tr2, Tr3, Tr4 \).

\( M_0 \xrightarrow{Tr1} M_1 \xrightarrow{Tr2} M_2 \xrightarrow{Tr3} M_3 \xrightarrow{Tr4} M_4 \).
6.3 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. $L_{jil} = 0$. But $T_j$ reaches the station $S_i$ within $\tau^B$.

6.3.1 Case 3.1

Train $T_j$ is at station $S_i''$, where $S_i'' \in S \setminus S_i$ and $i \in [1, n]$.

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 iff it has the highest priority while reaching $S_i$.

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

Then,

$$x_{ji}^{AT} = o_{ji}^{AT}$$

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

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

Here, damaged platform is $k$.

$k' = \{1, 2, \ldots, p\} \setminus \{k\}$

6.3.2 Case 3.2

Train $T_j$ is at $S_i'$, i.e. $P_{jiv'} = 1$, 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. i.e.

$$L_{jil} = 1, 1 \leq l < 4$$

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 $\tau^B$.

$$[L_{jil} = 0] \land [\text{Prio}(T_j) |_{t-x_{ji}}^{DT} > \text{Prio}(T_{j'}) |_{j \neq j', j \in [1, m]}] \land [\text{Re}(T_j) |_{x_{ji}}^{AT} \neq \text{Re}(T_{j'}) |_{x_{ji}}^{AT}]$$

Figure 6a represents Petri-Net model for the scenario of Case 3, described in equations (19)-(25) and the corresponding description of respective places, transitions and tokens are described in Table 5.
Figure 6. Petri-Net PN4
a Petri-Net model PN4 of Case 3.
b Reachability tree of PN4 for different firing sequences.

Table 5 Description of PN4

| 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 $S_{i'}$. |

| 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 $S_{i'}$. |

| Description of Colour 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. |

Analysis of PN4:
Table 5 presents the description of the places $P = \{P_1, P_2, P_3, P_4, P_5, P_6\}$ and transitions $Tr = \{Tr_1, Tr_2, Tr_3\}$ and the initial marking is $M_0 = [2, 1, 0, 1, 0, 1, 0]$.

- Reachability graph analysis:
  As discussed in subsection 6.1.2 initial marking $M_0$ is the root node as shown in Figure 6b. 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 $A$, $P = \{P1, P2, P3, P4, P5, P6\}$, denoted by rows and the order of the transitions is $Tr = \{Tr1, Tr2, Tr3\}$, denoted by columns.
  \[
  A = \begin{bmatrix}
  -1 & 0 & 0 \\
  -1 & 0 & 1 \\
  1 & -1 & 0 \\
  0 & 1 & -1 \\
  0 & -1 & 0 \\
  0 & 0 & 1
  \end{bmatrix}
  \]

  In our system, marking $M = [1, 1, 0, 0, 0, 1]$ is reachable from initial marking $M_0 = [2, 1, 0, 0, 1, 0]$ through the firing sequence $\sigma_1 = Tr1, Tr2, Tr3$.

  \[
  M_0 \xrightarrow{Tr1} M_1 \xrightarrow{Tr2} M_2 \xrightarrow{Tr3} M_3.
  \]

  \[
  \begin{bmatrix}
  2 \\
  1 \\
  0 \\
  0 \\
  1 \\
  0
  \end{bmatrix}
  \begin{bmatrix}
  -1 & 0 & 0 \\
  -1 & 0 & 1 \\
  1 & -1 & 0 \\
  0 & 1 & -1 \\
  0 & -1 & 0 \\
  0 & 0 & 1
  \end{bmatrix}
  \begin{bmatrix}
  1 \\
  1 \\
  0 \\
  0 \\
  1 \\
  1
  \end{bmatrix}
  =
  \begin{bmatrix}
  1 \\
  1 \\
  0 \\
  0 \\
  0 \\
  1
  \end{bmatrix}
  \]

  \[
  \begin{array}{c|c|c}
  \text{Case No.} & \text{Description} & \text{Decision Variable(s)} & \text{Decision Taken} \\
  \hline
  1 & \text{Station } S_i \text{ faces problem and train } T_j \text{ is on track } l & L_{ijt} & \text{Reroute or Retime} \\
  1.1 & \text{Station } S_i \text{ has free platforms when } T_j \text{ reaches } & P_{ijk}, Pri\theta(T_j), Re(T_j) & \text{Reroute or Retime from station} \\
  1.1.1 & \text{Re(T}_j) \text{ is available after } S_i & P_{ijk}, L_{ijt}, x_{ji}^{AT}, x_{ji}^{MT} & \text{Reroute from } S_i \\
  1.1.2 & \text{No alternative route found for } T_j \text{ from } S_i & P_{ijk}, x_{ji}^{AT} & \text{Delay at } S_i \\
  1.2 & \text{Station } S_i \text{ has no free platforms when } T_j \text{ reaches } & P_{ijk}, x_{ji}^{MT} & \text{Stop on track } l, \text{Retime} \\
  2 & \text{Number of trains are about to depart from } S_i \text{ within buffer} & L_{ijt}, P_{ijk}, Pri\theta(T_j), x_{ji}^{MT} & \text{Reorder} \\
  3 & \text{Train } T_j \text{ neither waits at affected station } S_i \text{ nor on track } l \text{ connected to } S_i, \text{but reaches to } S_i \text{ within } \tau^{d_0} & L_{ijt}, P_{ijk}, Pri\theta(T_j), x_{ji}^{MT} & \text{Retime} \\
  \end{array}
  \]

  \[
  \text{Table 6 Summary of Disaster Handling Cases Described in Section 6}
  \]

### 6.4 Delay Handling

Delay Minimisation can be formulated as:

- **Delay minimisation at station $S_i$ ($\delta_{ji}^{\text{min}}$).**

- **Delay minimisation on the track $l$ ($\delta_{jl}^{\text{min}}$).**

  $\delta_{ji}^{\text{min}}$: 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, i.e.
  \[
  x_{ji}^{AT} \geq o_{ji}^{AT} \tag{26}
  \]
  \[
  x_{ji}^{DT} = o_{ji}^{DT} \tag{27}
  \]
  So, it compromises dwell time of $T_j$ at $S_i$, i.e.
  \[
  o_{ji}^{d} < o_{ji}^{d} \tag{28}
  \]
  $\delta_{jl}^{\text{min}}$: This aims to minimise the delay considering the journey time (from source to destination), i.e.
  \[
  x_{jl}^{J} = o_{jl}^{J} \tag{29}
  \]
6.4.1 **Evaluating Optimised Objective Function**

The proposed rescheduling approach aims to minimise the total delay of trains in case of any disaster while rescheduling. Objective Function:

\[
\min \left[ \sum_{j} \delta_{j} \right] = \min \left[ \sum_{Rou(T_j)} \left( \delta_{ji}^{min} + \delta_{jl}^{min} \right) \right] = \min \left[ \sum_{P_{jik}} \delta_{ji}^{min} \right] + \min \left[ \sum_{L_{jkl}} \delta_{jl}^{min} \right]
\]

(30)

7 **Simulation Results**

To evaluate the performance of the proposed approach, experiments are conducted in different scenarios with different combination of tracks, trains and stations. Both the delay on track and at station are considered in the experiments.

![Diagram of Howrah and Asansol Division, Eastern Railway, India.](image)

**Figure 7.** Major part of Howrah and Asansol Division, Eastern Railway, India.
Table 7 Parameter used in Experimental Studies with Station and Train details

| Parameter used in Experimental Studies with Station and Train details |
|--------------------------------------------------|
| Parameter | Values |
| Total number of stations | 28 |
| Max number of platforms at station | 6 |
| Min number of tracks at station | 1 |
| Max number of tracks at station | 4 |
| Total number of trains | 21 |
| Threshold delay | 30 (in minute) |

Station Details of Eastern Railway (Howrah and Asansol division)

| Station Code | Station Name   | Station Code | Station Name   |
|--------------|----------------|--------------|----------------|
| HWH          | Howrah         | BMGA         | Bhamgara       |
| BLY          | Bally          | PAN          | Panagarh       |
| SHE          | Sheoraphuli Junction | PAW         | Pandabeswar   |
| DKAES        | Dankuni        | DGR          | Durgapur       |
| BDC          | Bandel Junction | UDL          | Andal          |
| KQU          | Kamarkundu     | RNG          | Raniganj       |
| TAK          | Tarkeshwar     | ASN          | Asansol        |
| KWAE         | Katwa Junction | STN          | Simtara        |
| SKG          | Saktigarh      | CRJ          | Chittaranjan   |
| BWN          | Bardhaman      | JMT          | Jamtara        |
| KAN          | Khana Junction | MDP          | Madhupur       |
| SNT          | Sainthia       | STL          | Simultala      |
| RPH          | Rampurhat      | JAJ          | Jhajha         |
| DHN          | Dhanbad        | BRR          | Barakar        |

Train Details of Eastern Railway (Howrah and Asansol division)

| Train No. | Train Name                                      | Category | Priority |
|-----------|------------------------------------------------|----------|----------|
| 12313     | Sealdah-New Delhi Rajdhani Express             | $T_{1}^{Pr}$ | $y_1$    |
| 12301     | Howrah - New Delhi Rajdhani Express            | $T_{2}^{Pr}$ | $y_1$    |
| 12273     | Howrah - New Delhi Duronto Express             | $T_{3}^{Pr}$ | $y_1$    |
| 12303     | Poorva Express                                 | $T_{4}^{Pr}$ | $y_2$    |
| 12019     | Howrah-Ranchi Shatabdi Express                 | $T_{5}^{Pr}$ | $y_1$    |
| 22387     | Black Diamond Express                          | $T_{6}^{Pr}$ | $y_4$    |
| 13051     | Hool Express                                   | $T_{7}^{Pr}$ | $y_4$    |
| 12329     | West Bengal Sampark Kranti Express             | $T_{8}^{Pr}$ | $y_2$    |
| 12339     | Coalfield Express                              | $T_{9}^{Pr}$ | $y_4$    |
| 12341     | Agnibina Express                               | $T_{10}^{Pr}$ | $y_4$   |
| 13009     | Doon Express                                   | $T_{11}^{Pr}$ | $y_2$    |
| 37211     | Howrah-Bandel Jn Local                         | $T_{12}^{Pr}$ | $y_5$    |
| 13017     | Ganadevta Express                              | $T_{13}^{Pr}$ | $y_4$    |
| 37911     | Howrah-Katwa Jn Local                          | $T_{14}^{Pr}$ | $y_5$    |
| 63541     | Asansol-Gomoh MEMU                             | $T_{15}^{Pr}$ | $y_5$    |
| 53061     | Barddhaman Jn-Hatia Passenger                  | $T_{16}^{Pr}$ | $y_5$    |
| 15662     | Kamakhya - Ranchi Express                      | $T_{17}^{Pr}$ | $y_2$    |
| 63525     | Barddhaman Jn-Asansol Jn MEMU                  | $T_{18}^{Pr}$ | $y_5$    |
| 63523     | Barddhaman Jn-Asansol Jn MEMU                  | $T_{19}^{Pr}$ | $y_5$    |
| 53131     | Sealdah-Muzaffarpur Fast Passenger             | $T_{20}^{Pr}$ | $y_4$    |
| 12359     | Kolkata - Patna Garib Rath Express             | $T_{21}^{Pr}$ | $y_1$    |
7.1 Experimental Setup

The simulation is coded in Java in JADE [38] in UNIX platform of personal computer with 2.90 GHz processor speed and 4GB memory. The results and computations are evaluated under same running environment. A part of Eastern Railway, India [39], shown in Figure 7, is taken for experimental studies.

Table 7 describes the parameters with their values taken for the experiments, station details and the train categorisation as discussed in subsection 3.2. Total 28 major stations and 21 different types of trains are taken to generate the real-time scenarios.

First the database is set with all the station details and train details and the neighbourhood of the stations in railway network. For both the Asansol and Howrah division, there are maximum 4 tracks in between two stations and for some part, stations are connected with 1 single track. Each station is assumed to have max $p = 6$ number of platforms. The permissible threshold delay, as discussed before, is taken as 30 min. For simplicity we assume, all stations are equidistant and trains are running at a constant speed throughout its journey.

7.2 Illustration

To illustrate the proposed method 7 different scenarios are taken at different times of a day, based upon the cases discussed in Section 6. Some scenarios describe affected stations, whereas some describe the disruption on track as well as blocked stations. The first scenario $Sc^1$ describes the blocked station $BLY$, but no track is blocked. The disaster happened at 6:00 am. Based on the proposed approach, train no. 12019, 13051, 12303 are delayed and rescheduled. Total delay encountered here is 15 min. Similar incidents are described in $Sc^2$, $Sc^3$, and $Sc^5$ in different times, where only stations are blocked. Scenario $Sc^7$ highlights a disaster, happened at 21:00, where track no. 2, i.e., DOWN track between stations $UDL$ and $RNG$ is blocked. Train no. 12341, 13009, 12359 are rescheduled in this case with our proposed approach, facing a total delay of 24 min. Similar scenarios are mentioned in $Sc^4$, $Sc^6$. The details of blocked stations, blocked tracks, affected trains, and total delay observed in each scenario is described in Table 8.

| Scenario | Time of disaster | Blocked Station | Blocked Track | Affected Train No. | Total Delay (in min) |
|----------|------------------|-----------------|---------------|-------------------|---------------------|
| $Sc^1$   | 6:00             | BLY             | -             | 12019, 13051, 12303 | 15                  |
| $Sc^2$   | 7:00             | KAN             | -             | 12030, 12019, 13051, 53131 | 34                  |
| $Sc^3$   | 7:30             | UDL             | -             | 12019, 53061, 22387 | 29                  |
| $Sc^4$   | 13:00            | BWN, KAN        | (BWN $\leftrightarrow$ KAN)$^2$ | 12273 | 0                  |
| $Sc^5$   | 17:40            | KAN             | -             | 12339, 12313, 12301, 63523, 63525 | 15                  |
| $Sc^6$   | 20:00            | ASN, STN        | (ASN $\leftrightarrow$ STN)$^3$ | 12339, 12341, 12359, 63525 | 10                  |
| $Sc^7$   | 21:00            | UDL, RNG        | (UDL $\leftrightarrow$ RNG)$^2$ | 12341, 13009, 12359 | 24                  |

Depending upon the impact of the occurred disaster, the scenario is distinguished in four different categories, such as: (a) More number of trains in circulation and high impact of disruption, (b) More number of trains in circulation and low impact of disruption, (c) Less number of trains in circulation and low impact of disruption, (d) Less number of trains in circulation and high impact of disruption. In figure 8a, graphical representation of the expected delay of each train is shown depending upon this categorisation.

In Figure 8b, the changes in delay of each train is shown when same disaster happens at normal time and at busy time. Without any disaster, all trains maintain original schedule and no delay is observed. So, the graph maintains a straight line with zero delay for all trains.
Figure 8. Simulation Results
a Change of number of affected trains and their expected delay under disruption scenarios.
b Delay of trains with disaster at normal time and busy time.
c Delay Minimisation through our proposed method.
d Comparison between existing centralised approach and our proposed agent-based distributed approach.
Figure 8c represents the delay minimisation, achieved through proposed approach. Threshold delay is taken as 30 min. We vary the time of disaster in 24h time period to observe the total delay of trains through the proposed method.

To exhibit the advantage of proposed approach, results are compared with the existing centralised decision making approach of Indian Railway. In this method, all the rescheduling decisions are taken by the central authority. All the low level authorities pass the necessary messages to the next higher level authority in the railway hierarchy and so on. The higher authority checks for all feasible solutions and the best decision message for rescheduling is passed from central authority to the lower authorities for necessary changes. This procedure is time consuming and may face disadvantages of traditional centralised systems like, single-point failure, lesser autonomy, under utilisation etc. The comparison between existing centralised and the proposed agent-based distributed approach is shown in Figure 8d. In case of every disaster scenario, happened in different time of a day in the railway network, significant reduction in delay is observed through the proposed approach.

8 Conclusion

This paper proposes a new train rescheduling approach to handle delay optimisation in case of disruptions in a railway network. An agent based solution using the DCOP and MDP was designed to address the distributed nature of the scenario and the uncertainty of disaster recovery time. Experimental studies are conducted on Eastern Railway, India to evaluate the effectiveness of the approach. In disastrous situation with noticeably large recovery time, the proposed approach is shown to produce lower delay than existing approaches.

One of the future research directions will aim at extending this approach for rescheduling of trains which follow a meet-pass sequence [40] using headway time [41]. This will increase the number of constraints noticeably which need to be handled efficiently. Further, cross-over points between any two stations will also be considered which can help in handling various collision scenarios.

9 References

[1] Indian railways. https://en.wikipedia.org/wiki/Indian_Railways. Last modified on 10 May 2016.

[2] Xiang Li Biying Shou and Dan Ralescu. Train rescheduling with stochastic recovery time: A new track-backup approach. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2014.

[3] J.T. Krasemann. Greedy algorithm for railway traffic re-scheduling during disturbances: a swedish case. IET Intell. Transp. Syst., 4:375–386, 2010.

[4] R.P Feynman and F.L Vernon Jr. A mip-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers. Journal of Rail Transport Planning & Management, pages 38–53, 2013.

[5] Bart Kersbergen, Ton van den Boom, and Bart De Schutter. Reducing the time needed to solve the global rescheduling problem for railway networks. Proceedings of the 16th International
IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), pages 791–796, 2013.

[6] Poulami Dalapati, James Arambam Singh, Animesh Dutta, et al. Multiagent based railway scheduling and optimization. IEEE TENCON, Bangkok, Thailand, 2014.

[7] Indian rail incidents. https://en.wikipedia.org/wiki/List_of_Indian_rail_incidents Last modified on 6 May 2016.

[8] Trung Thanh Nguyen and Xin Yao. Benchmarking and solving dynamic constrained problems. IEEE Congress on Evolutionary Computation, pages 690–697, 2009.

[9] Fabien Leurent. Transport capacity constraints on the mass transit system: a systemic analysis. Eur. Transp. Res. Rev., 3:11–21, 2011.

[10] James Atlas and Keith Decker. A complete distributed constraint optimization method for non-traditional pseudotree arrangements. AAMAS, 2007.

[11] Duc Thien Nguyen, William Yeoh, and Hoong Chuin Lau. Stochastic dominance in stochastic dcops for risk-sensitive applications. AAMAS, 2012.

[12] Jose M Vidal. Fundamentals of multiagent systems. 2010.

[13] Daniel S. Bernstein, Robert Givan, Neil Immerman, et al. The complexity of decentralized control of markov decision processes. UNCERTAINTY IN ARTIFICIAL INTELLIGENCE PROCEEDINGS, 2000.

[14] Daniel S. Bernstein, Christopher Amato, Eric A. Hansen, et al. Policy iteration for decentralized control of markov decision processes. Journal of Artificial Intelligence Research, 34:89–132, 2009.

[15] L. Meng and X. S. Zhou. Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach. Transp. Res. B, 45:1080–1102, 2011.

[16] Rodrigo Acuna-Agost, Philippe Michelon, Dominique Feillet c, et al. Sapi: Statistical analysis of propagation of incidents. a new approach for rescheduling trains after disruptions. Eur. J. Oper. Res., 215:227–243, 2011.

[17] R. Acuna-Agost, D. Feillet, S. Gueye, et al. A mip-based local search method for the railway rescheduling problem. Networks, 57:69–86, 2011.

[18] Ravi Sekhar Chalumuri and Asakura Yasuo. Modelling travel time distribution under various uncertainties on hanshin expressway of japan. Eur. Transp. Res. Rev., 6:85–92, 2014.

[19] Lixing Yang, Ziyou Gao, and Keping Li. Railway freight transportation planning with mixed uncertainty of randomness and fuzziness. Applied Soft Computing, 2010.

[20] Ibrahim Takouna and Roberto Rojas-Cessa. Routing schemes for network recovery under link and node failures. IEEE, pages 69–73, 2008.

[21] Lingyun Meng and Xuesong Zhou. Simultaneous train rerouting and rescheduling on an n-track network: A model reformulation with network-based cumulative flow variables. ELSEVIER, Transportation Research Part B, 67:208–234, 2014.
[22] Zuraida Alwadood, Adibah Shuib, and Norlida Abd. Hamid. A review on quantitative models in railway rescheduling. *International Journal of Scientific & Engineering Research, 3*, 2012.

[23] Malik Muneeb Abid and Muhammad Babar Khan. Sensitivity analysis of train schedule of a railway track network using an optimization modeling technique. *Eur. Transp. Res. Rev.*, 7:2–7, 2015.

[24] Li Wang, Limin Jia, Yong Qin, Jie Xu, and Xuelei Meng. Method for tuberail train rescheduling system. *International Conference on Computer Application and System Modeling (IC-CASM)*, pages 563–567, 2010.

[25] Paola Pellegrini, GrAl'gory MarliÃtre, et al. Recife-milp: An effective milp-based heuristic for the real-time railway traffic management problem. *IEEE Transactions On Intelligent Transportation Systems*, 2015.

[26] Liujiang Kang, Jianjun Wu, Huijun Sun, et al. A practical model for last train rescheduling with train delay in urban railway transit networks. *Elsevier, Omega 50*, pages 29–42, 2015.

[27] Terminal dwell time. [https://en.wikipedia.org/wiki/Terminal_dwell_time](https://en.wikipedia.org/wiki/Terminal_dwell_time). Last modified on 4 February 2016.

[28] Vito Fragnelli and Simona Sanguineti. A game theoretic model for re-optimizing a railway timetable. *Eur. Transp. Res. Rev.*, 6:113–125, 2014.

[29] Thijs Dewilde, Peter Sels, Dirk Cattrysse, et al. Robust railway station planning: An interaction between routing, timetabling and platforming. *ELSEVIER, Journal of Rail Transport Planning & Management*, 3:68–77, 2013.

[30] Shuai Su, Tao Tang, Xiang Li, and Ziyou Gao. Optimization of multitrain operations in a subway system. *IEEE Transactions On Intelligent Transportation Systems*, 15, 2014.

[31] Mariagrazia Dotoli, Nicola Epicoco, Marco Falagario, et al. A real time traffic management model for regional railway networks under disturbances. *IEEE International Conference on Automation Science and Engineering (CASE)*, 2013.

[32] Hsu-Chun Yen. Introduction to petri net theory. *Studies in Computational Intelligence*, 25, 2006.

[33] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77:541–580, 1989.

[34] N. Chaki and S. Bhattacharya. Performance analysis of multistage interconnection networks with a new high-level net model. *Journal of Systems Architecture, ELSEVIER*, 52:57–70, 2006.

[35] Kurt Jensen and Lars M. Kristensen. Coloured petri nets, modelling and validation of concurrent systems. *Springer*, 2009.

[36] Pengling Wang, Lei Ma, Rob M. P. Goverde, et al. Rescheduling trains using petri nets and heuristic search. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 17:726–735, 2016.

[37] Michael Wooldridge. *An Introduction to Multiagent Systems*. 1966.
[38] Jade. [http://jade.tilab.com/](http://jade.tilab.com/).

[39] Eastern railway zone. [https://en.wikipedia.org/wiki/Eastern_Railway_zone](https://en.wikipedia.org/wiki/Eastern_Railway_zone) Last modified on 4 May 2016.

[40] Lingyun Meng and Xuesong Zhou. Robust single-track train dispatching model under a dynamic and stochastic environment: a scenario-based rolling horizon solution approach. *Publication in Transportation Research Part B*, pages 1–33, 2011.

[41] Headway. [https://en.wikipedia.org/wiki/Headway](https://en.wikipedia.org/wiki/Headway) Last modified on 29 February 2016.