A Framework for Improvement of Railway Timetabling

R V Satwik
IIT Bombay
Prof. Madhu N. Belur

November 30, 2021

Outline

Section

1. Introduction
2. Reinforcement Learning
3. Results

GQD Network and its importance

- Golden quadrilateral and diagonals (GQD)
- Not the complete Indian Railways (IR) network
- 15% of network route share; **60% of total traffic**
- Economically connects the four major metros via high-speed services
- Consists of trains of mainly two categories, **coaching trains**, and **freight trains**.
- Coaching trains have a predefined **order of priority** based on which an overtake of one coaching train over another is decided
- Freight trains have the least priority in comparison to trains of other categories (including coaching trains)
- A timetabling is **first completed** on the GQD network

GQD network

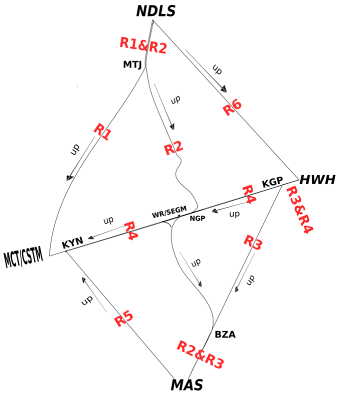

Figure: GQD network as graph, with junction and end stations labelled. (source: [1])

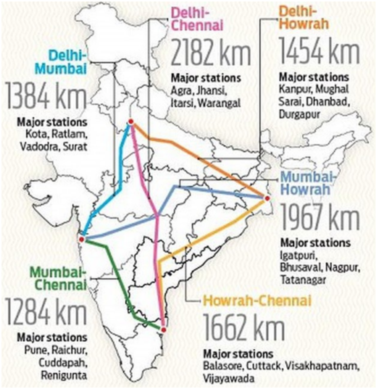

Figure: GQD network marked on Indian map. (source: [2])

Significance of Project

- Only a fraction of GQD has sections with permissible speed $\geqslant$ 130 kmph
- High **operating ratio** of Indian Railways
- Ease of **scheduling new trains**

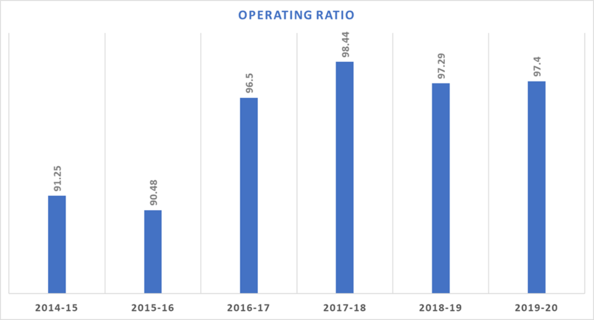

Figure: Indian Railways operating ratio between 2014 to 2020 as per data from [3]

Significance of Project

- Increase **average speed** of trains
- Improve the total **utilization** of the network

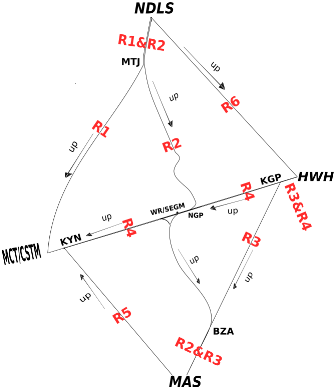

Figure: GQD network as graph, with junction and end stations labelled. (source: [1])

Utilization of Stations

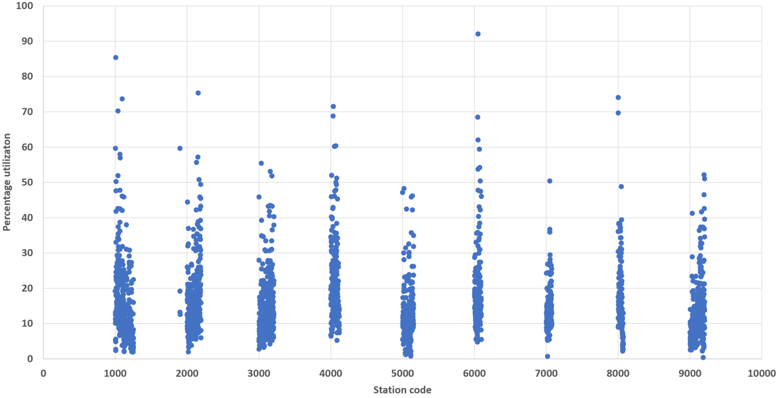

Figure: Percentage utilization at various stations in ZBTT proposed timetable *(November,2020)*. The stations are encoded with station codes that range from 1000 to 9000

Requirement of Freight Corridor

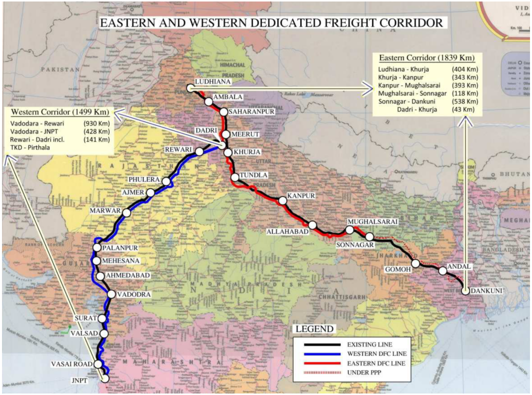

Figure: Eastern and Western Dedicated Freight Corridors Source: [2]

Performance metrics

- Delay of a train
- Priority weighted delay

$$\text{Priority weighted delay of } i^{th} \text{ train} = \frac{\text{Delay of } i^{th} \text{ train}}{\text{priority of } i^{th} \text{ train}}$$

- Utilization of network

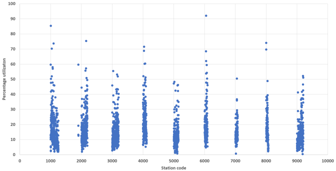

Figure: Percentage utilization at various stations in ZBTT proposed timetable *(November,2020)*. The stations are encoded with station codes that range from 1000 to 9000

Existing solution

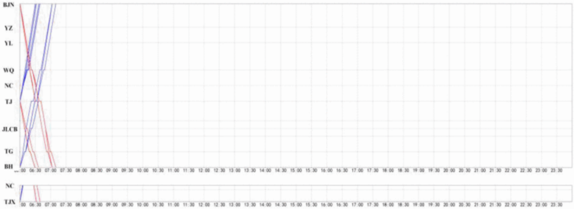

Figure: Initial schedule in [4]

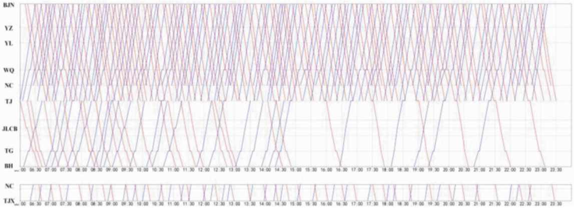

Figure: Final schedule in [4]

Existing solution

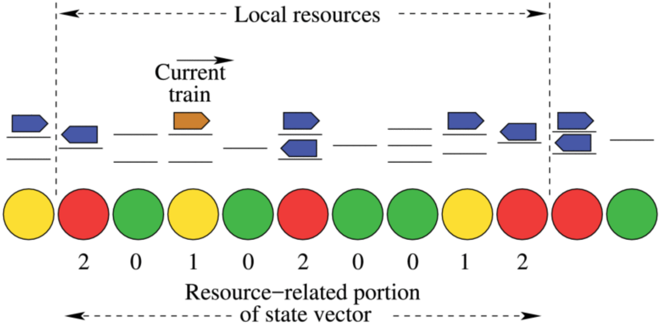

Figure: Building the state vector for trains [5]

Section

Reinforcement Learning example

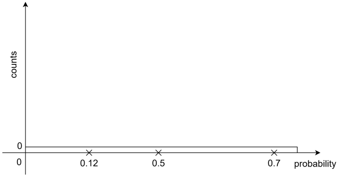

Figure: No estimation performed

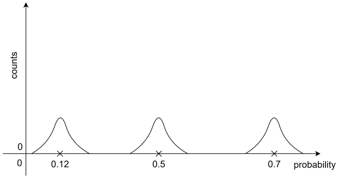

Figure: Estimation without reward maximization

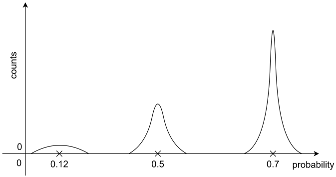

Figure: Estimation with reward maximization

Framework

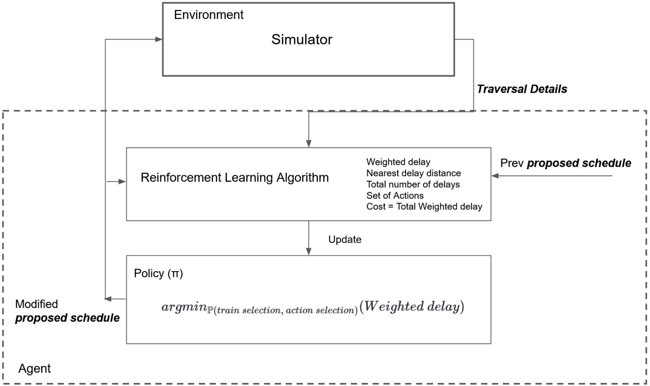

Figure: A block diagram showing a learning agent in feedback to the simulator

State of an Output Timetable

	minimum_tt(mins)	simulation_tt(mins)	traversal_time_delay(mins)	weighted_delay(mins)	_delay_distance(perc	est_delay_time(	nber_of_delays
10002	942.55	942.55	0.00	0.00			0.00
10003	979.55	988.85	9.29	9.29	0.00	7.00	8.00
10005	937.55	995.95	58.39	58.39	0.00	13.00	34.00
10007	945.55	1024.19	78.64	78.64	0.00	20.00	50.00
10009	942.55	1028.46	85.91	85.91	0.00	27.00	49.00
10010	942.55	1033.06	90.51	90.51	0.00	34.00	59.00
10017	1052.85	1061.39	8.54	8.54	0.00	40.00	7.00
10018	1052.85	1075.09	22.24	22.24	0.00	47.00	19.00
10001	982.06	1089.44	107.38	53.69	0.00	54.00	53.00
10004	982.06	1097.27	115.21	57.60	0.00	61.00	58.00
10006	982.06	1104.14	122.09	61.04	0.00	67.00	71.00
10008	982.06	1110.00	127.95	63.97	0.00	74.00	74.00
10011	982.06	1115.78	133.72	66.86	0.00	81.00	75.00
10012	914.86	914.86	0.00	0.00			0.00
10013	648.16	648.16	0.00	0.00			0.00
10019	648.16	661.75	13.59	6.79	49.93	6.00	10.00
10020	982.06	1121.55	139.50	69.75	0.00	88.00	76.00
10014	290.22	457.78	167.56	55.85	75.02	76.00	3.00
10015	331.49	331.49	0.00	0.00			0.00
10016	1134.55	1337.25	202.70	67.57	39.72	25.00	4.00
12001	1502.20	1502.97	0.76	0.19	0.00	94.00	1.00
12002	1502.20	1516.83	14.63	3.66	0.00	101.00	13.00
12003	1502.20	1528.21	26.00	6.50	0.00	108.00	21.00
12004	1502.20	1541.13	38.93	9.73	0.00	115.00	27.00
12005	1502.20	1551.09	48.89	12.22	0.00	122.00	34.00
12006	1502.20	1561.87	59.67	14.92	0.00	129.00	37.00
12007	1502.20	1572.16	69.95	17.49	0.00	136.00	41.00
12008	1502.20	1582.76	80.55	20.14	0.00	143.00	44.00
12009	1502.20	1592.05	89.84	22.46	0.00	150.00	49.00

Figure: A state of the system containing 29 trains

State of an Output Timetable

- Minimum Traversal Time
- Simulated Traversal Time

$$\text{Simulated Traversal Time} \geqslant \text{Minimum Traversal Time}$$

$$\text{Traversal Time Delay} = \text{Simulated Traversal Time} - \text{Minimum Traversal Time}$$

$$\text{Weighted Delay} = \frac{\text{Traversal Time Delay}}{\text{Priority of train}}$$

- Nearest Delay Distance
- Nearest Delay Time
- Total Number of Delays

Determining Minimum Traversal Time

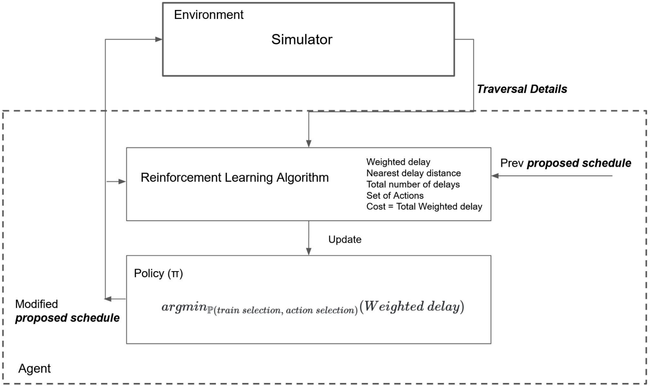

Figure: A block diagram showing a learning agent in feedback to the simulator

State of an Output Timetable

	minimum_tt(mins)	simulation_tt(mins)	traversal_time_delay(mins)	weighted_delay(mins)	_delay_distance(perc	est_delay_time(n	ber_of_delays
10002	942.55	942.55	0.00	0.00			0.00
10003	979.55	988.85	9.29	9.29	0.00	7.00	8.00
10005	937.55	995.95	58.39	58.39	0.00	13.00	34.00
10007	945.55	1024.19	78.64	78.64	0.00	20.00	50.00
10009	942.55	1028.46	85.91	85.91	0.00	27.00	49.00
10010	942.55	1033.06	90.51	90.51	0.00	34.00	59.00
10017	1052.85	1061.39	8.54	8.54	0.00	40.00	7.00
10018	1052.85	1075.09	22.24	22.24	0.00	47.00	19.00
10001	982.06	1089.44	107.38	53.69	0.00	54.00	53.00
10004	982.06	1097.27	115.21	57.60	0.00	61.00	58.00
10006	982.06	1104.14	122.09	61.04	0.00	67.00	71.00
10008	982.06	1110.00	127.95	63.97	0.00	74.00	74.00
10011	982.06	1115.78	133.72	66.86	0.00	81.00	75.00
10012	914.86	914.86	0.00	0.00			0.00
10013	648.16	648.16	0.00	0.00			0.00
10019	648.16	661.75	13.59	6.79	49.93	6.00	10.00
10020	982.06	1121.55	139.50	69.75	0.00	88.00	76.00
10014	290.22	457.78	167.56	55.85	75.02	76.00	3.00
10015	331.49	331.49	0.00	0.00			0.00
10016	1134.55	1337.25	202.70	67.57	39.72	25.00	4.00
12001	1502.20	1502.97	0.76	0.19	0.00	94.00	1.00
12002	1502.20	1516.83	14.63	3.66	0.00	101.00	13.00
12003	1502.20	1528.21	26.00	6.50	0.00	108.00	21.00
12004	1502.20	1541.13	38.93	9.73	0.00	115.00	27.00
12005	1502.20	1551.09	48.89	12.22	0.00	122.00	34.00
12006	1502.20	1561.87	59.67	14.92	0.00	129.00	37.00
12007	1502.20	1572.16	69.95	17.49	0.00	136.00	41.00
12008	1502.20	1582.76	80.55	20.14	0.00	143.00	44.00
12009	1502.20	1592.05	89.84	22.46	0.00	150.00	49.00

Figure: A state of the system containing 29 trains

Actions

$$\text{Proposed starting time} = \text{Previous starting time} + \text{Traversal Time Delay}$$

$$\text{Proposed starting time} = \text{Previous starting time} + \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} \sum_{0\%}^{40\%} \text{Delay} \\ \sum_{40\%}^{80\%} \text{Delay} \\ \sum_{80\%}^{100\%} \text{Delay} \end{bmatrix}$$

$$\text{Proposed starting time} = \text{Previous starting time} - \text{Traversal Time Delay}$$

$$\text{Proposed starting time} = \text{Previous starting time} - \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} \sum_{0\%}^{40\%} \text{Delay} \\ \sum_{40\%}^{80\%} \text{Delay} \\ \sum_{80\%}^{100\%} \text{Delay} \end{bmatrix}$$

$$\text{chosen train} = \arg \max_{train} \Big(\sum_{stations} \text{traversal time delay} \Big)$$

$$\text{chosen train} = \arg \max_{train} \Big(\sum_{stations} \text{weighted delay} \Big)$$

$$\text{chosen train} = \arg\max_{train} (n(\text{delay}))$$

State of a Learning Agent

Train_Action	0	1	2	4997	4998	4999
(0, 0)	1	1	2	504	504	504
(0, 1)	1	1	1	330	330	330
(0, 2)	1	1	1	125	125	125
(0, 3)	1	1	1	18	18	18
(1, 0)	1	1	1	29	29	29
(1, 1)	1	1	1	336	336	336
(1, 2)	1	1	1	36	36	36
(1, 3)	2	2	2	207	207	207
(2, 0)	1	1	1	731	731	732
(2, 1)	1	1	1	67	67	67
(2, 2)	1	1	1	137	137	137
(2, 3)	1	1	1	19	19	19

Figure: An example of the agent learning the best actions after 5000 iterations among the available 12 actions using PIA algorithm

$$T(s, a, s' \,\Big|\, \text{exploitation}) = \mathcal{P}(\text{action}_1, \text{action}_2, \cdots, \text{action}_{12})$$

Markov Decision Process

- A state S, is sufficient to define the environment state and there is nothing else affecting how environment behaves
- Once an proposed schedule data is considered and an action is chosen, the next state s_n does not require the information regarding any of the previous states and actions $((s_0, a_0), (s_1, a_1), (s_2, a_2), (s_3, a_3), \cdots, (s_{n-2}, a_{n-2}))$ and hence the simulation-feedback process can be considered as a Markov decision process

$$\mathcal{P}\left(\text{next timetable}(s_n)\middle|(s_0, a_0), (s_1, a_1), (s_2, a_2), \cdots\right) = \mathcal{P}\left((s_n)\middle|(s_{n-1}, a_{n-1})\right)$$

$$T(s') \leftarrow T(s), a, \text{cost}$$

Data-flow diagram

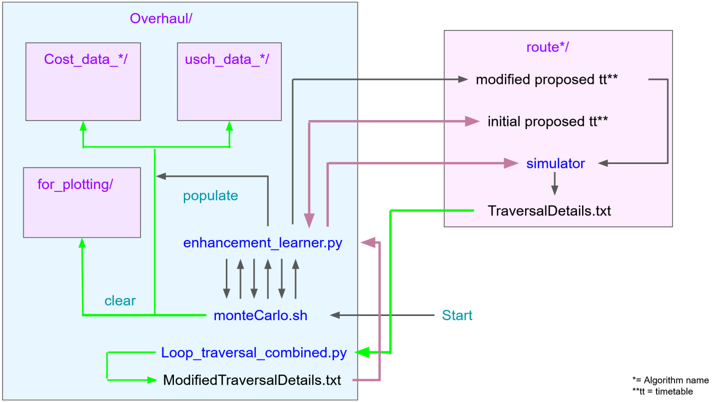

Figure: Framework used for testing various algorithms

Parallel execution of Code

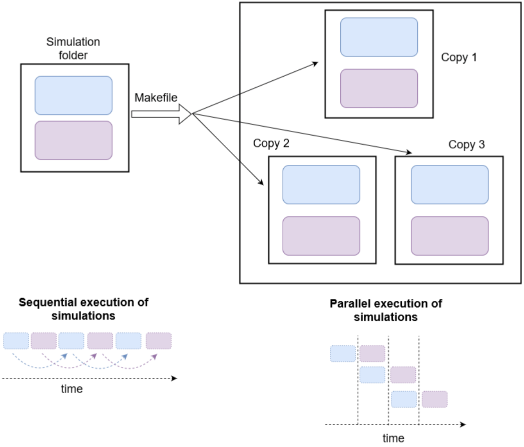

Figure: Parallel execution of simulations

Section

PI Algorithm

Initialization
$V(s) \leftarrow U[0,1]$
Policy evaluation
while $N \neq 0$ **do**
  if $U(0,1) < \alpha$ **then**
    $a \leftarrow Rand\pi(s)$;
  else
    $a \leftarrow max(\mathcal{P}(\pi(s)))$;
  end
  $N \leftarrow N - 1$;
  if $\Delta C < 0$ **then**
    $V(s) \leftarrow a + 1$;
  end
end

PIA results

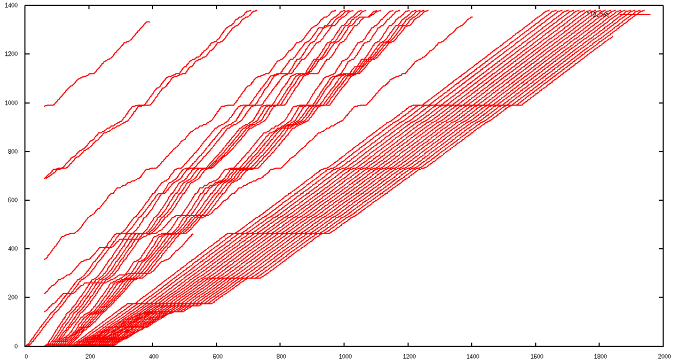

Figure: Initial proposed schedule

Priority Weighted Delay (Cost) = 1352 minutes

PIA results

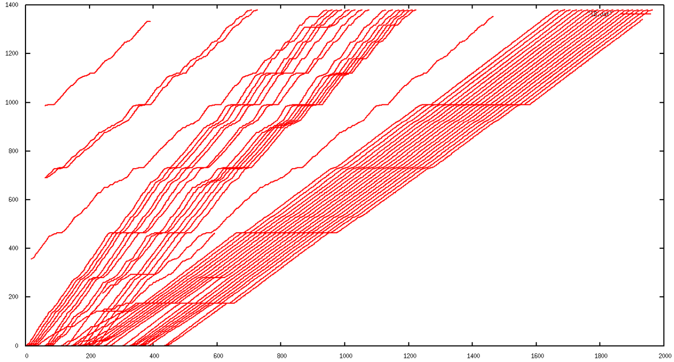

Figure: Distance vs time graph after 30 iterations of PIA

Priority Weighted Delay (Cost) = 832 minutes

PIA results comparison

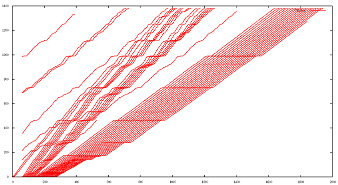

Figure: Initial proposed schedule

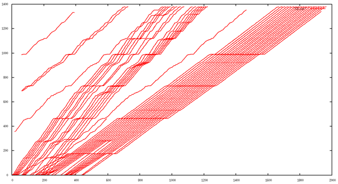

Figure: Distance vs time graph after 30 iterations of PIA

Priority Weighted Delay (Cost) = 1352 minutes

Priority Weighted Delay (Cost) = 832 minutes

PIA results

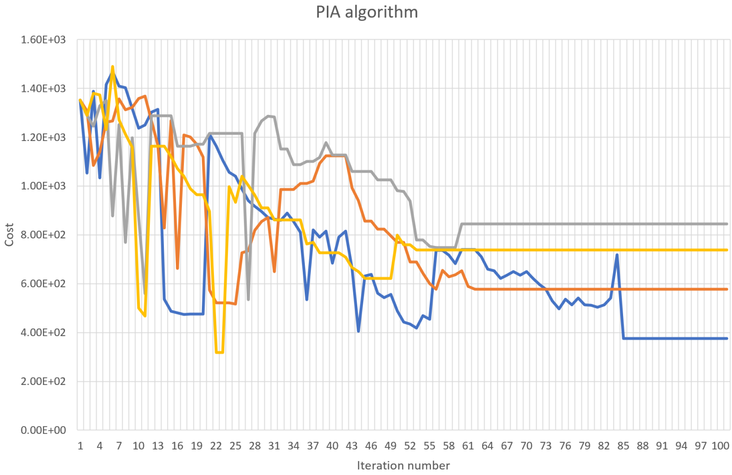

Figure: Comparison of 4 different PIA simulations

SA Algorithm

Initialization

$V(s) \leftarrow U[0,1] \quad T \leftarrow T_0$

Policy evaluation

while $N \neq 0$ **do**

   $T \leftarrow T/N$

   if $U(0,1) < \alpha$ **then**

      $a \leftarrow Rand\pi(s)$;

   else

      $a \leftarrow max(\mathcal{P}(\pi(s)))$;

   end

   $N \leftarrow N - 1$;

   $D \leftarrow C - C_{previous}$;

   $M \leftarrow e^{(\frac{-D}{CT})}$;

   if $\Delta C < 0$ or $Rand < M$ **then**

      $V(s) \leftarrow a + 1$;

   end

end

SA results

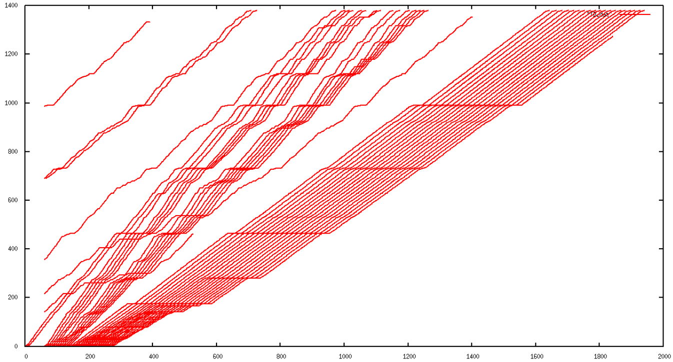

Figure: Initial proposed schedule

Priority Weighted Delay (Cost) = 1352 minutes

SA results

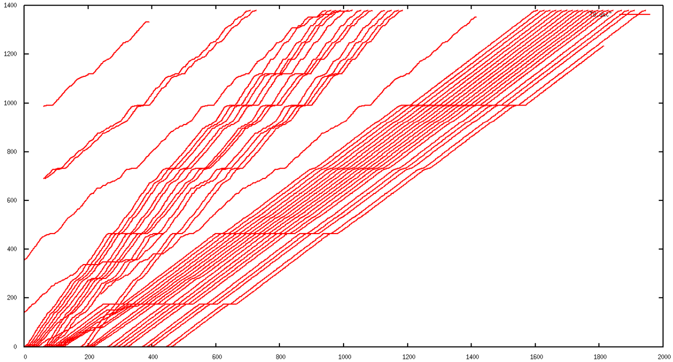

Figure: Distance vs Time graph after 30 iterations of SA

Priority Weighted Delay (Cost) = 572 minutes

SA results

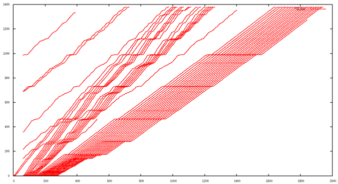

Figure: Initial proposed schedule

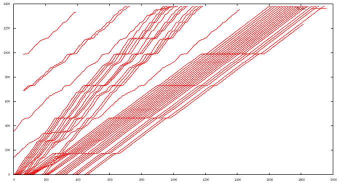

Figure: Distance vs Time graph after 30 iterations of SA

Priority Weighted Delay (Cost) = 1352 minutes

Priority Weighted Delay (Cost) = 572 minutes

SA results

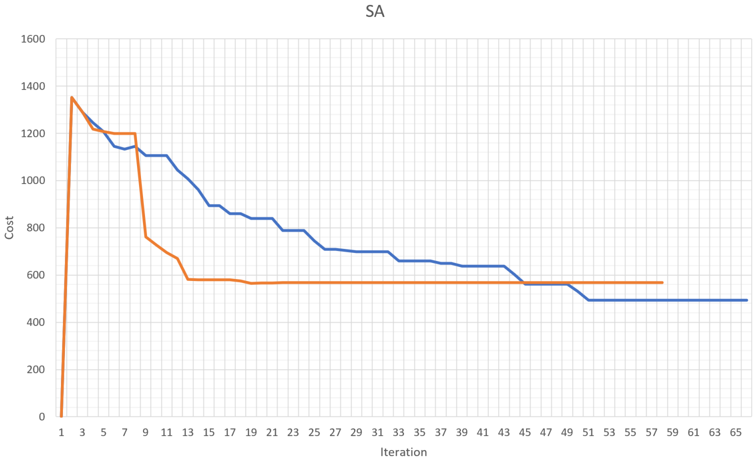

Figure: Comparison of 2 different SA simulations

Results

TrainNo.	Initial	Final		TrainNo.	Initial	Final
11012001	60	1335		11010001	60	879
11012002	60	966		11010002	60	20
11012003	60	724		11010003	60	8
11012004	60	982		11010004	60	371
11012005	60	882		11010005	60	229
11012006	60	1258		11010006	60	787
11012007	60	1241		11010007	60	0
11012008	60	1682		11010008	60	456
11012009	60	852		11010009	60	455
11012010	60	1386		11010010	60	43
11012011	60	821		11010011	60	289
11012012	60	1122		11010012	60	1102
11012013	60	1299		11010013	60	60
11012014	60	1188		11010014	60	556
11012015	60	1144		11010015	60	60
11012016	60	1003		11010017	60	60
11012017	60	1308		11010018	60	96
11012018	60	909		11010019	60	74
11012019	60	1516		11010020	60	268
11012020	60	1495				

Figure: Initial and Final starting times of proposed schedule for 5000 iterations of PIA

TrainNo	Initial	Final		TrainNo	Initial	Final
11012001	60	155		11010001	60	276
11012002	60	410		11010002	60	60
11012003	60	408		11010003	60	60
11012004	60	447		11010004	60	316
11012005	60	512		11010005	60	2
11012006	60	186		11010006	60	230
11012007	60	466		11010007	60	275
11012008	60	475		11010009	60	226
11012009	60	528		11010010	60	0
11012010	60	519		11010011	60	315
11012011	60	462		11010012	60	0
11012012	60	600		11010013	60	60
11012013	60	588		11010015	60	60
11012014	60	309		11010016	60	472
11012016	60	339		11010017	60	60
11012017	60	275		11010018	60	60
11012018	60	302		11010019	60	60
11012019	60	384		11010020	60	388
11012020	60	362				

Figure: Initial and Final starting times of proposed schedule for 5000 iterations of SA

Summary

In this thesis,

- A framework was generated to modify the existing timetable through multiple stochastic paths to generate multiple timetables using an existing railway network simulator
- A metric that can be used for comparing two versions of timetables has been used to execute two learning algorithms (namely PIA and SA)

Future work

- Degree of **centrality** associated with nodes of a graph $\implies$ The bottleneck nodes/sections will be given more weightage in the penalty/cost function
- **Modifying** the starting times of **multiple** of trains in one epoch
- Unrealistic starting times $\implies$ regularization in the cost function
- Actor-critic based neural network $\implies$ estimate the non-linear relation between features of train (priority, net delay, nearest delay etc.) and the amount of delay/pre-pone time
- Careful characterization rewards for freight windows as it complements the existing reward procedure
- Implement SQL database for exploratory analysis
- The arrival/departure of coaching trains must not be between 12 am and 5 am
- Consider maintenance block timings for feedback

Thank You!

References I

M. Belur, R. Gopalakrishnan, S. Kumar, S. Pulapadi, N. Rangaraj, S. Ramisetty, S. Singh, M. Salekin, and M. Tyagi, "Network-wide mixed-rail traffic scheduler: challenges and implementation aspects," submitted to the journal INOC 2022.

A. Punit, "Mechanised maintenance philosophy of railway assets." [Online]. Available: https://uic.org/IMG/pdf/d2_s4_mechanised_maintenance_philosophy_of_railway_assets_punit_agrawal_india_part2.pdf

P. I. Bureau, "Performance of indian railways in the last five years." [Online]. Available: https://pib.gov.in/PressReleaseIframePage.aspx?PRID=1602424

Y. Guo, "A reinforcement learning approach to train timetabling for inter-city high speed railway lines," in *2020 IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 2020, pp. 493–497.

References II

H. Khadilkar, "A scalable reinforcement learning algorithm for scheduling railway lines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 727–736, 2019.