User Manual for Mixed-rail-traffic network-simulation tool

RB sponsored project at
IIT Bombay

User manual date: 4th Oct 2020

Contact persons

Prof. Narayan Rangaraj	Prof. Madhu N. Belur
Industrial Engineering and Operations Research	Department of Electrical Engg
Indian Institute of Technology Bombay	IIT Bombay
Mumbai 400 076	Mumbai 400 076
Ph: 022 2576 7882	Ph: 022 2576 7404
Email: narayan.rangaraj@iitb.ac.in	Email: belur@iitb.ac.in

Website for further details:
https://www.ee.iitb.ac.in/~belur/railways/mixed-traffic-network-scheduler/

Summary of the simulator

The simulator simulates trains running on a section and needs
- section infrastructure (station distances, block lengths, number of platforms, gradients, if any, and other details)
- train timetable (this includes train arrival/departure at various stations and a loop-number when there is a halt)

The simulator hence has four parts:
- (a) GUI (to enter infrastructure and timetable data, and add/edit existing data)
- (b) Simulate the train running (this includes scheduled and unscheduled freight trains), together with safety/headway constraints and speed, acceleration, deceleration characteristics of various types of trains).
- (c) Compare two timetable files to view train-wise or station-wise differences between the two timetable files, for example proposed and simulated timetables, or simulated timetables before and after an infrastructure change.
- (d) List of freight paths sorted based on path-quality (i.e., traversal time) so that one can pick the optimum freight paths based on acceptable traversal duration.

Point (d) above helps in optimizing the freight timings to get acceptable quality freight paths in a congested section. Point (c) helps in evaluating the benefit of adding a infrastructure or changing the passenger schedule by comparing the congestion/freight path quality before and after the change.

In addition to this detailed document, four spoken-tutorials (each is a 10-minute video showing how one
- starts adding infrastructure (stations, blocks, loops)
- adds trains on this infrastructure
- simulates trains (passenger and freight) on this infrastructure, obtain distance-time charts, saves the timetable as an xls file
- adds/edits infrastructure/trains using the GUI and/or spreadsheets.

Updated versions of the spoken tutorials are maintained at
http://www.ee.iitb.ac.in/%7ebelur/railways/RDSO/
together with a jar-file of the simulator. Detailed documentation is below.
For more information, please contact the persons on the first page.

1. Scope of Work

The areas identified under the theme of "Capacity Optimisation and Simulation Tool" of Railway planning and Operations are as follows:

1. Design of basic software tool for computation of line capacity of any Indian Railway section taking into consideration various parameters like type of block working, block length, signal spacing, speed restrictions, number of stations, number of loop lines at the stations, gradient, scheduled trains, unscheduled trains, length of trains and other parameters that would have an effect on line capacity.

2. The tool would be used for examining the effect on line capacity of the technological improvements like Intermediate Block Signaling, introduction of a new stations, doubling/tripling/quadrupling, etc. of the section, Automatic Signaling with 3 aspect/4 aspect, ETCS L2, Centralised Traffic control/Train Management System, other modern signaling systems being evolved or being used over Indian Railways /internationally, etc.

3. The tool would be used for examining the effect on line capacity of changes in operational patterns like using mix of goods and passenger trains, various loco load combinations, intelligent speed adaptation.

4. The tool would be able to assess the extent to which section line capacity could be increased and quantify the impact of each capacity enhancement. This could be applied to identify bottlenecks on the railway network.

5. The tool would do the assessment of effect of failure of specific equipment on the line capacity and plan the best deployment of the equipment.

6. The tool should be capable of placing the scheduled/unscheduled trains in the section in an optimum manner taking feed as the timetable of the section. Also the tool should be able to plan placement/graphing of the train when above mentioned improvements are simulated.

7. The simulation in this context would mean the numerical computation and graphing of the train running characteristics at the level of accuracy that is adequate for purpose of analysis.

1.1 Deliverables

1. Basic Simulator development for computing line capacity considering the various parameters as covered in point no 1 of the previous section.

2. Development of various applications covering the scope of the work as optimisation tool.

i) Quantify numerically/graphically, the effect on line capacity of the technological improvements including Intermediate Block Signalling, introduction of a new station, doubling / tripling/ quadrupling of a section, Automatic Signalling with 3 aspect/4 aspect. Technologies such as ETCS L2, Centralised Traffic control/Train Management System, other modern signalling systems being evolved or being used over Indian Railways and internationally would also be assessed.

ii) Quantify numerically/graphically, the effect on line capacity of changes in operational patterns including different mixes of goods and passenger trains and various loco load combinations.

iii) Application to assess the extent to which section line capacity could be increased and quantify the impact of each capacity enhancement. This could be applied to identify bottlenecks on the railway network.

iv) Application to assess the effect of failure of specific equipment on line capacity and plan the best deployment of the equipment.

v) Application for placing the scheduled/unscheduled trains in the section in an optimum manner taking feed as the timetable of the section and also to plan placement/graphing of the train when above mentioned improvements are simulated.

Note: Special equipment/software, etc. procured for the projectshall form deliverables to RDSO on completion of the project.

1.2 Responsibilities of IIT Bombay and RDSO regarding deliverables

1) The new technologies that need to be assessed would have to be described in technical detail by RDSO so that suitable simulation logic can be incorporated to the required level.

2) The simulator would be designed to allow for a number of different train operating characteristics and overall mix of traffic. Once a set of them is agreed upon (with consent of RDSO), the basic performance of the simulator would be demonstrated and validated with a representative set of train data (e.g. one set of passenger train characteristics and one set of freight train characteristics). Building a data base of all relevant loco load combinations that may be used on Indian Railways would not be the responsibility of IIT Bombay.

3) A nodal officer at RDSO has to be available for discussion regarding operational rules, and for the assessment of the progress from time to time and accepting the final deliverable of the project.

4) Data regarding test cases for validation should be made available to the IIT Bombay team. If such data is not available with RDSO, the responsibility of getting Indian Railways to provide this data to the RDSO/IIT Bombay team is with RDSO.

5) Equipment related data, including failure data, would have to be provided by RDSO. The failure

characteristics would be quantified by IIT Bombay and included in the simulator logic. This demonstration would be done for specific equipment identified mutually.

2. Functional Requirements Specifications

The simulator is designed to schedule trains in a feasible manner provided certain inputs. This chapter describes what the simulator takes as input, what the user can expect as output. Finally it also mentions the logic based on which the simulator actually works.

2.1 Input to the simulator

This section explains the inputs to the simulator along with the screen-shots of the input, additional information about the GUI is detailed in the section-GUI.

The first screen of the interface. In this screen the user is shown a new window with tabs 'Simulator' and 'Comparator'. For running a new or existing simulation the user needs to click on the 'Simulator' Tab.

Fig 1: The 1st screen of the simulator

2) After clicking on simulator, a drop-down menu will appear. From this menu the option 'Add/Edit Infrastructure/Train' should be selected which will open a new window as shown below.
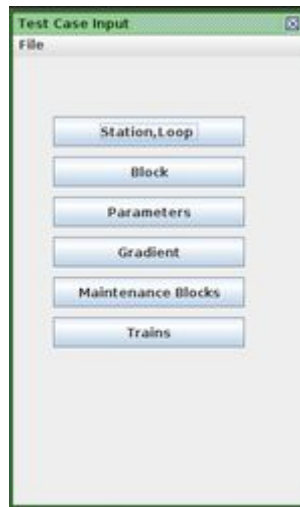
Fig 2: The window appearing on clicking on 'Add/Edit Infrastructure/Train'

3)Before creating new stations, blocks, loops the user needs to click on 'File'. From the drop-down menu 'New' needs to be selected. The new configuration file needs to saved in a desired location. Stations and loops needs to be entered after this. This can be done by clicking on the 'Stations,Loops' tab, which opens a new window as shown. (Fig. 3)

4) Once the station details have been entered, the user needs to click on the tab 'Add station', which again opens a new window for entering the loops, titled 'Loop Input for #Station Name#'. Upon entering a new loop 'Add new loop' should be clicked to actually add the loop. Once loops are entered according to requirement for a particular station, clicking on 'Done' will close this window. (Fig. 4)

Fig 3: Window where details of stations and loops needs to be entered.

5)The other stations and loops that are required can now be entered in the steps already described one-by-one. After all the data for all the stations and the loops have been entered the user can check the correctness of the data entered by clicking on 'View Station....' and 'View Loop Det...'. Two such windows are shown here for reference.

List of all loops

Id	Direction	Station Name	Type	Maximum Speed
10001	up	a	mainLine	100.0
11001	down	a	mainLine	100.0
10101	up	a	loop	15.0
11101	down	a	loop	15.0
20001	up	b	mainLine	100.0
21001	down	b	mainLine	100.0
20101	up	b	loop	15.0
21101	down	b	loop	15.0
30001	up	c	mainLine	100.0
31001	down	c	mainLine	100.0
30101	up	c	loop	15.0
31101	down	c	loop	15.0
40001	up	d	mainLine	100.0
41001	down	d	mainLine	100.0
40101	up	d	loop	15.0
41101	down	d	loop	15.0

List of stations

Station Id	Name	Start Km	End Km	Entry Speed
01	a	0.0	1.0	100.0
02	b	10.0	11.0	100.0
03	c	20.0	21.0	100.0
04	d	30.0	31.0	100.0

Fig 4: Window showing List of Loops and List of sections

6) If the entered data is correct, the user can proceed. If not, the user needs to 'edit' his previous entry. Once all the data regarding stations and loops are entered, 'Done' has to be clicked. Now block data needs to be entered. So, the user needs to click on 'Block' tab of the window of Fig 2. Once 'Block' is selected, a new window will open. In this window select a station from the drop-down menu that is present. (Even if the first station is already selected, select it again from the drop-down menu). The following window will appear.
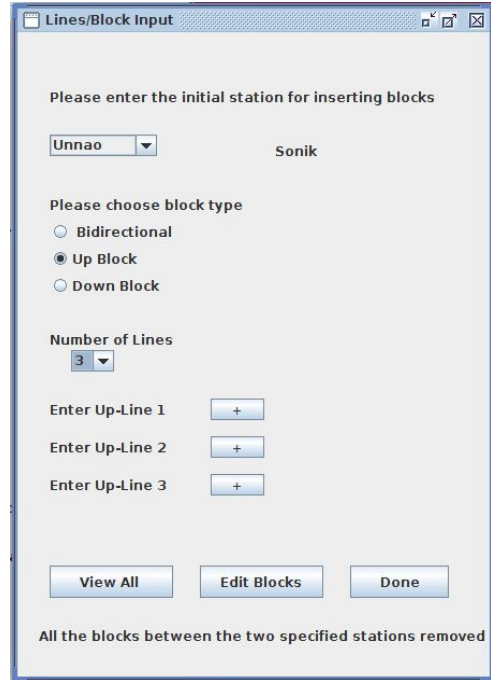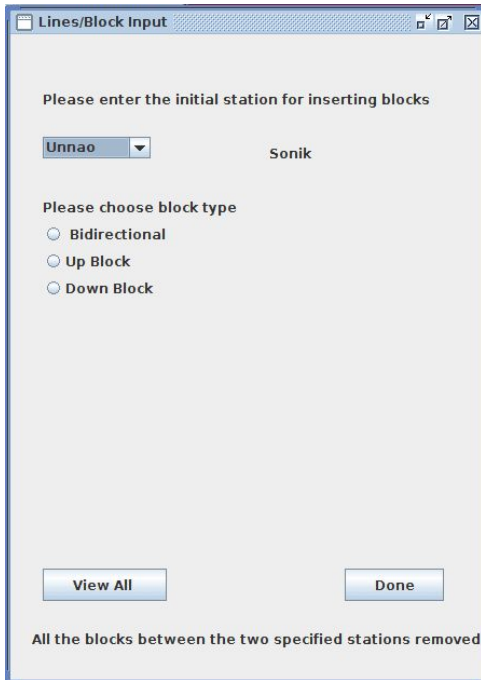
Fig 5.1: Window for entering block details after selecting the block option
FIg 5.2: Multi line GUI.
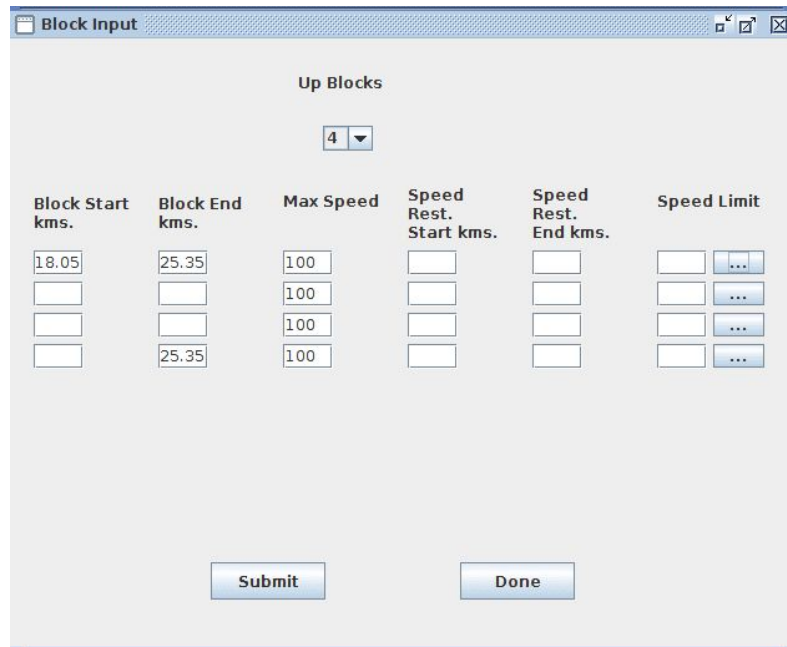
Fig 6: Window for entering block details of individual lines.

13

7)The user can select the radio-button 'Up Block', 'Down Block' or 'Bidirectional' and then from the list that appears select the number of lines (for multiple up/down lines) in either Up or Down directions (as selected). Now the details of the blocks can be entered via the "+" button for each individual line. The new window appears as shown in fig 6. The details can be added and for Multiple speed restriction, the "..." button can be used. fig 7 shows the GUI for muliple speed restriction.

Fig 7: Window for multiple speed restriction..

8) Once blocks are entered, the user should click on 'Submit' and then on 'Done'. Now suppose some block entries need to be modified. Again 'Block' needs to be clicked on and steps as before should be followed . As shown below the user should click on 'Yes' in the first window, and then 'Ok' in the second window as shown.
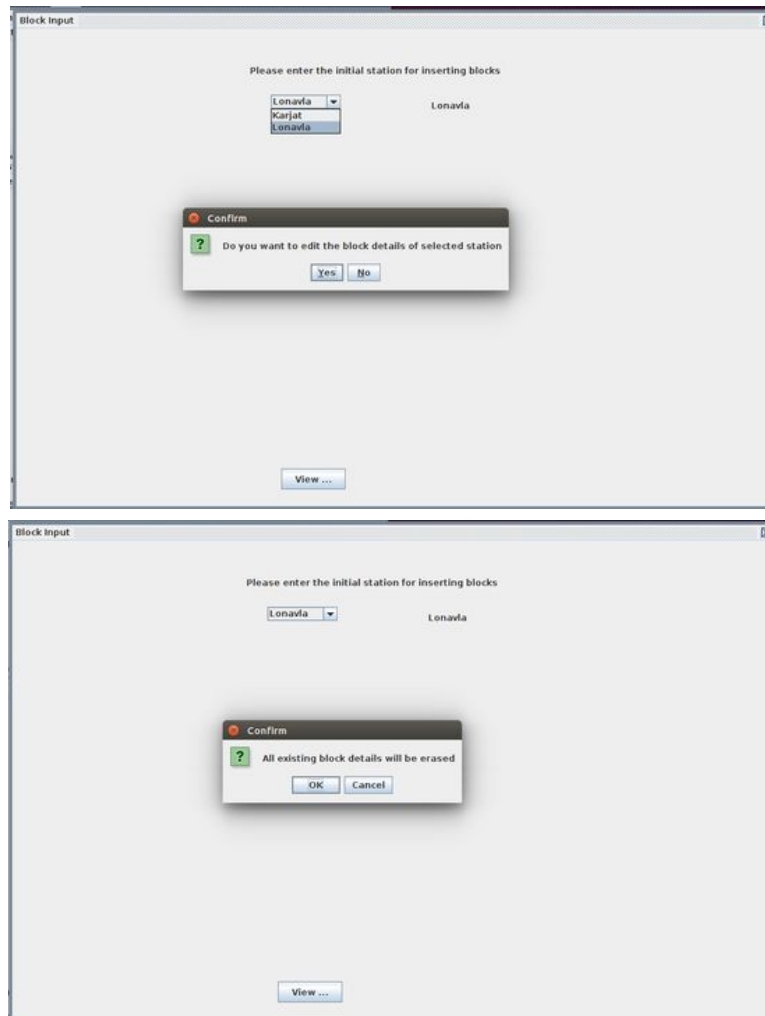
Fig 8: Screens appearing while editing blocks

9) 'Done' is to be clicked on after block entries are over. To enter parameters the user should follow 'Parameters'-----> 'Ok' ---------> 'Done'.
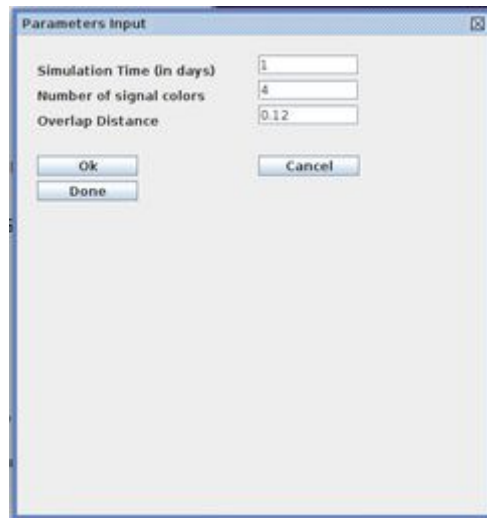
Fig 9: Window appearing while entering parameters

10) To enter gradients the user should follow 'Gradient'-----> 'Add' ---------> 'Done'.
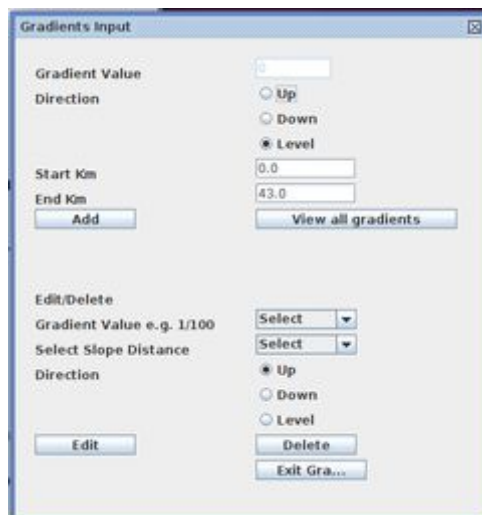
Fig 10: Window appearing while entering Gradient

16

11) Now trains need to be entered. After clicking on the tab 'Trains', in the new window that appears, trains that are required are to be added. User should select the loop number appropriately. After adding a particular train the user should click on 'Add train'. After repeating the procedure for all the trains that are required to be entered, the user should click on 'Done. The figure shows the window for entering the trains.
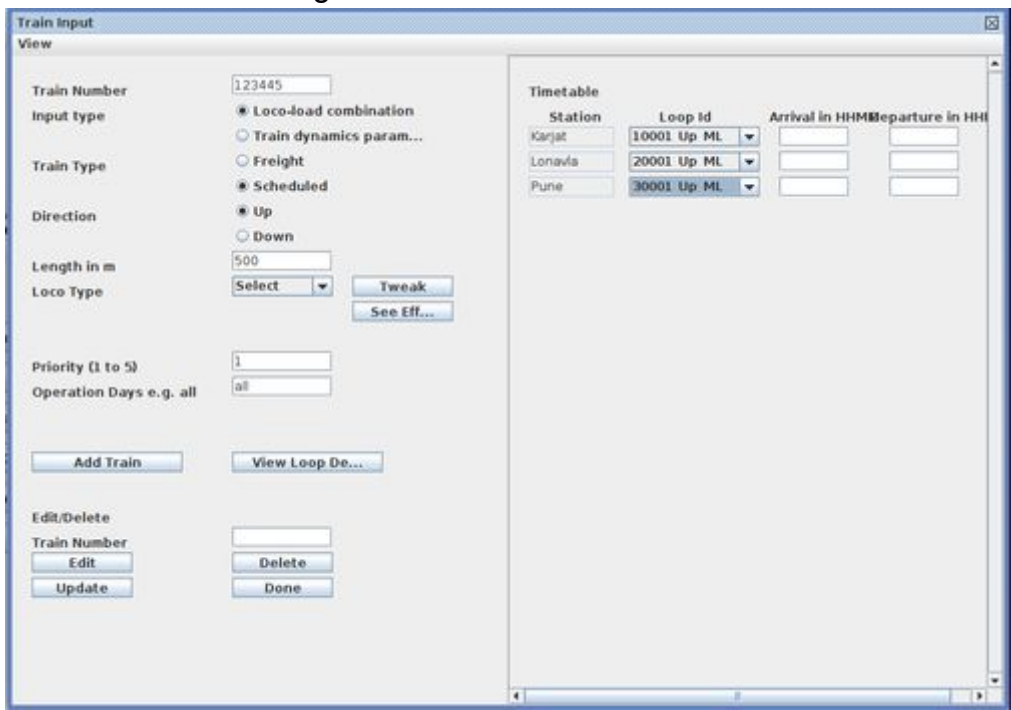
Fig 11: Window for entering trains

Once trains are entered, the input stage of the simulator is complete. The user can now save the configuration file by clicking on 'File'---->'Save'.

The inputs are saved as text files in the saved location, and the corresponding values can be manually changed in the file using default text editor.

2.2 Output of the simulator

Once the simulator inputs have been given, the simulator can actually be "run". For this the following steps are to be followed:

1) The user should click on 'Simulator' tab and then on 'Select Infrastructure/Train'. The following window should appear.
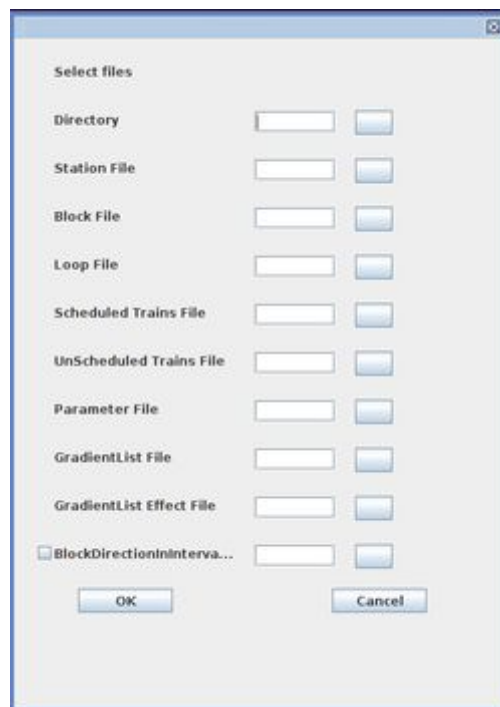
Fig 12: Window appearing when 'Select Infrastructure/Train' is clicked.

In this window, the user can select the directory where the configuration files are saved by clicking on the side button. All the files are automatically selected. Finally the user should click on 'Ok'. To add maintenance block details, the 'BlockDirectionInterval' checkbox must be checked

2) The user should now click again on 'Simulator' and then 'Simulate'. The simulator will now start executing and outputs will be generated.

3) A sample output screen is shown below:
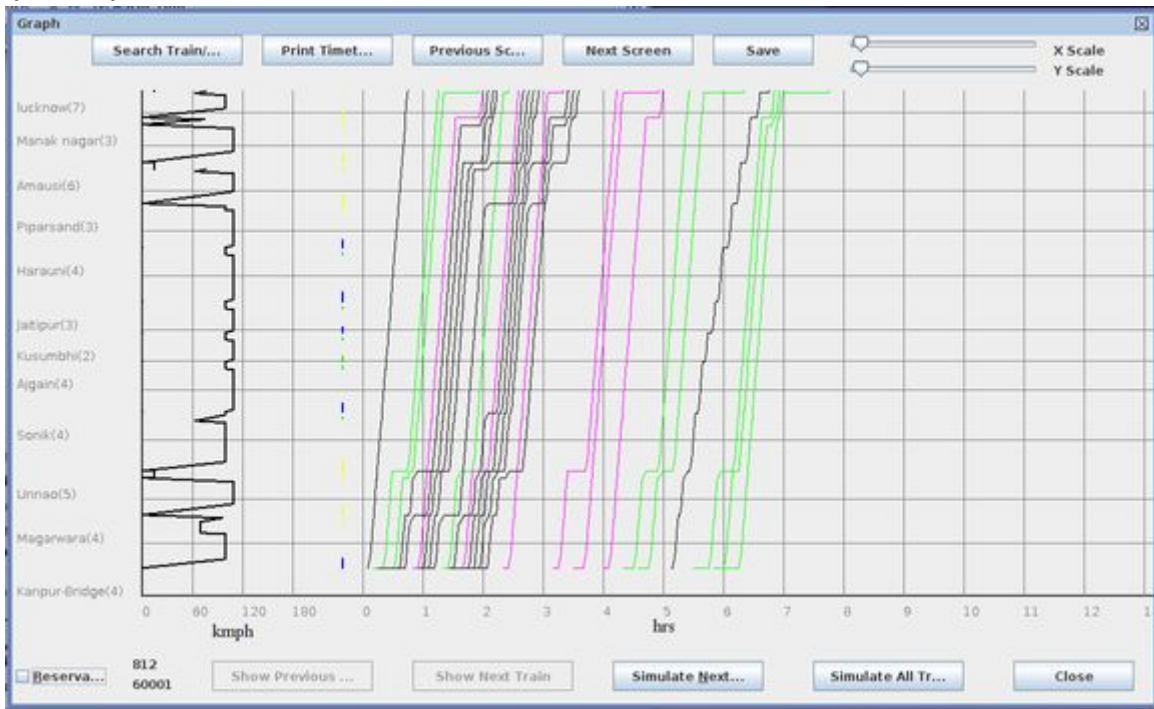
Fig 13: A sample output screen of the simulator

4) On clicking on 'Simulate All Trains' tab, the simulation will start and after the completion the user will be prompted to give a location for saving the excel file with the time table. On selecting this option, the timetable will be saved according to the user's choice. 'Simulate Next Train' tab can be used to simulate single trains one after the other so that the Distance-Time chart and velocity profile of the train can be analysed.

2.3 Logic of the simulator

 The logic of the simulator is discussed mainly under 3 heads:-

2.3.1 Blocks

They are sections between stations. Their specifications are as follows:-
1) They need to be specified by start and end mileposts and also their linkages after their use in each direction (ie. up linkages for an up block section, down linkages for a down block section and both for a common block section)
2) It is possible to enforce speed restrictions on any part of a section. Speed restrictions on a section within the block creates "tinyblocks" as far as the simulation software is concerned. The tinyblock captures the fact that before a train reaches a speed-restriction section, the deceleration has to begin: these distances are precomputed and stored using tinyblocks. This is also very relevant in the Gradients part: see next item.
3) Gradients - The logic underlying the implementation of gradients is complex: tinyblocks defined above is crucially used for capturing the effects of gradients in the tiny sections preceding and following a gradient section to capture the effect on a train's acceleration/deceleration depending on the train length and the gradient value. The value of the slope and the mileposts are recorded with respect to the infrastructure direction, and the acceleration or deceleration effect depends on the train's running direction: the acceleration and deceleration values are recorded in the gradients.txt and the GradientEffects.txt files. *Note that mileposts recorded in the files are* **relative** *to the block start/end points.*This detail is not relevant to a typical user.

2.3.2 Loops

They are sections at stations. Their specifications are as follows:-
1) Signalized track sections at stations are called 'loops'. They include 'main line loops at stations' where in principle trains can go through at normal speed without stopping or slowing down, directional loops (up, down and common) which require a change of track and which will have loop entry speed restrictions
2) In a double line section, at a station, there will be up and down mainline loops and in addition, there could be up, down or common loops.
3) Loops have lengths and can therefore accommodate trains of a certain length.

2.3.3 Trains

1) Trains are taken one by one in order of priority.

2) Paths are computed respecting occupancy reservations of already scheduled trains in each block section.

3) Speeds of trains are as per the occupancy of block sections ahead of the train

4) A lower priority train should not cause delay of a higher priority train - this is achieved by reserving the block occupancy of the higher priority train first

5) If a train cannot move ahead from a station, it will wait at an appropriate loop line - if no loop line is available, the algorithm backtracks to the previous station and if necessary to the beginning of the section, where it is assumed that there are a sufficiently large number of loop lines

6) All trains move as per their acceleration value (from rest or from any other speed) till they reach the maximum permissible speed and decelerate as per their deceleration value (till they reach zero or any other speed dictated by speed restrictions). This is a simplified model of train movement that is adequate for capacity calculations.

7) Scheduled trains move as per their velocity characteristics and also a timetable at every station in the section

8) Scheduled trains can arrive early at a station but cannot depart earlier than the scheduled departure time

9) Unscheduled trains have a firing time when they enter the section and try to find the earliest departure time that ensures a valid path through to the end of the section.

The simulator handles train scheduling on a linear section and generates a conflict free, feasible schedule which also includes complex scenarios like overtakes, originating & ending trains in between of the section, etc. The functionality of the algorithm is best described via the standard way of describing what an algorithm does:

1) Takes input from the user of the infrastructure details, signaling strategy, gradients, the trains schedule, maintenance block timings.

2) Using core logic it tries to schedule trains following trains schedule & maintenance timings. All Passenger trains are scheduled before Freight trains. Within passenger/freights itself, the train with higher priority is scheduled first. In the simulator, all trains have a priority. In the time horizon of simulation, trains with higher priority are scheduled first. Within trains of the same priority, trains that are scheduled to depart at the ends of the section earlier are scheduled first. For passenger (scheduled) train if the station in consideration is halt station (arrival & departure time have some gap in it) then both conditions: Train cannot leave before its departure time; Train must stop at least halting minutes (scheduled departure – scheduled departure) irrespective of its arrival time, must be satisfied. No two trains can occupy same block. Signaling aspect must be abided.

2.3.4 Backtracking for Freight trains

Freights should try to find feasible paths through the existing map of passenger trains versus time. If the part of the section where the freight has to start is vacant then a freight can start its movement. At a later stage there if there is a passenger train is there which is supposed to occupy / reserve the same block that freight will occupy / reserve at same time, then the freight will try to look for available loop lines at previous station to give way to passenger (scheduled) trains to overtake freight atthat station. If there isn't any then it will backtrack to another previous station.
3) Output: Conflict free, feasible schedules of as many input trains as possible in the given simulation time.

3. System Requirements Specifications

This chapter mentions the various software requirements of the simulator, along with the hardware requirements for running the required software. In other words they specify the software tools that are required for a person to run the simulator. The simulator is based on Java. The hardware specifications required for running the simulator is same as those required for running Java. They are listed in the link: http://java.com/en/download/help/sysreq.xml.
For Java versions 7.0 and 8.0 the required specifications are:

      Windows 10/8/7/Vista/Server 2008/Server 2012
      RAM-128 MB
      Disk Space-124 MB for JRE, 2 MB for Java Update
      Processor- Pentium 2 266 MHz or higher
      Browser- Internet Explorer 9 or above/ Mozilla Firefox/Google Chrome

Setup.exe
setup.exe file is programed to install all the dependencies automatically required by the tool simulator. it contains the java installer and the python installer along with its dependencies. Many anti-virus softwares do not allow setup.exe files to be downloaded/transferred from CD/pendrive, and hence this filename has also been made available as java-python-installer.exe.

This file will execute upon double click. If the program doesn't operate then the following steps for installation can be used for the same. Eclipse IDE need not be installed because the simulator can run independently using .jar file or the .exe file provided. Eclipse IDE is used to run the program directly from the source code. Steps to run .jar and .exe file is described in the following section.

The setup file is added in the site https://www.ee.iitb.ac.in/~belur/railways/RDSO

STEPS FOR INSTALLING JAVA AND ECLIPSE IDE

3.1 Steps for installing Java

1) Open Google Chrome/Internet Explorer/Mozilla Firefox.(This is explained for Google Chrome).
2) In the start page, locate the search bar.
3) In the search bar type in https://java.com/en/download/ as shown in the figure below.
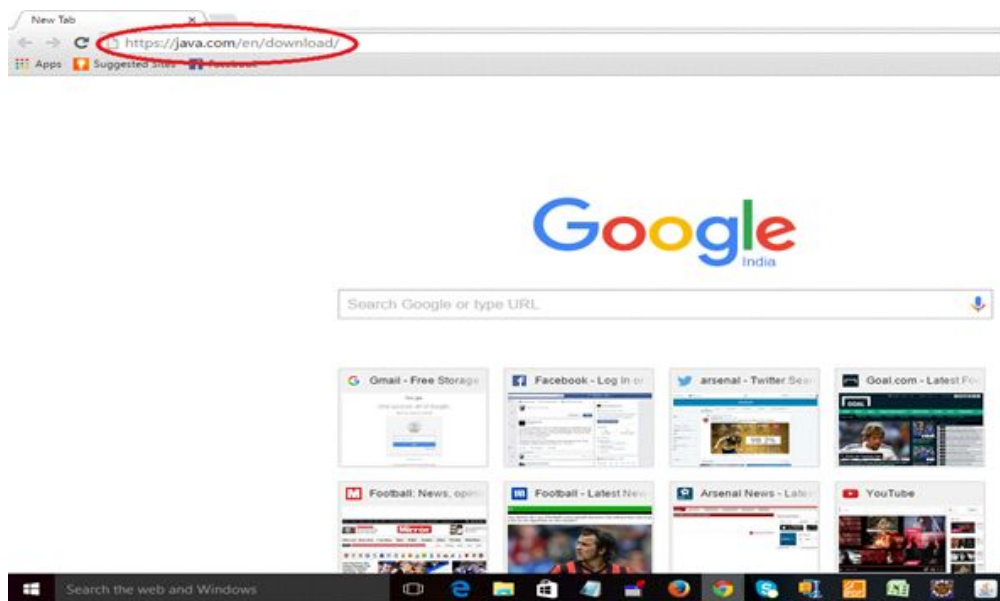
Fig 1: Searching for Java in Google Chrome

4) After entering the above link press Enter key. The following web-page should open.

Fig 2: Window for downloading Java

5) Now click on the "red" tab named "Free Java Download". On clicking this another window opens with another "red" tab named "Agree and Start free Download". Click this tab. This will result in JavaSetup8u60.exe being downloaded (in red circle).

Fig 3: Window showing download of Java setup

6) Once the download is complete click on the setup icon just downloaded. Java will start installing. Follow the instructions that appear on the screen and Java will be downloaded completely.

Thus after these steps Java will be installed in the machine.

24

3.2 Steps for installing Winrar

1)   In the search bar type in http://www.win-rar.com/postdownload.html?&L=0. Press enter and the download of wrar.exe starts.
2)   Once downloaded click on the downloaded setup and run it. The setup of Winrar will start.

Once it starts follow the instructions and winrar will be installed

3.3 Steps for installing Eclipse IDE

1) As before in the browser search tab type in https://www.eclipse.org/downloads/ and press Enter key. The following page should open.
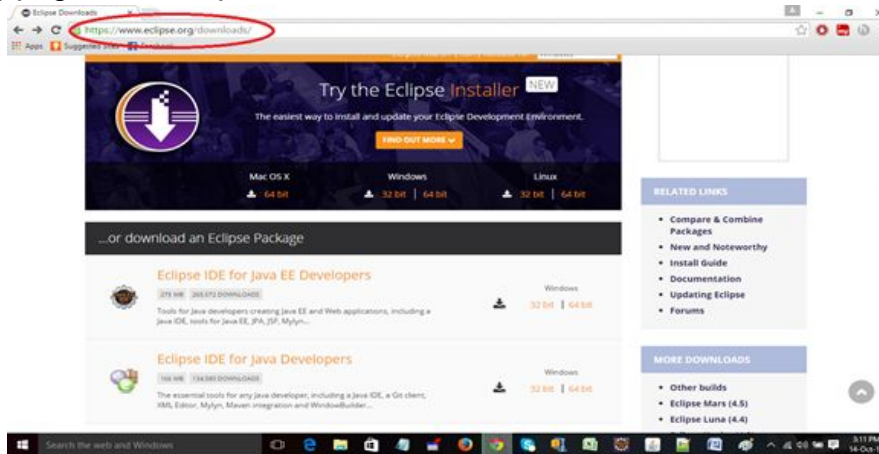
Fig 4: Window showing searching for eclipse

2) Depending on your machine being 32 bit/64 bit you should select either 32 bit/64 bit version of eclipse under the title Eclipse IDE for Java EE Developers. On selecting this option the following page should open. Select "Direct Link to File" on the top right pane of the window as shown.
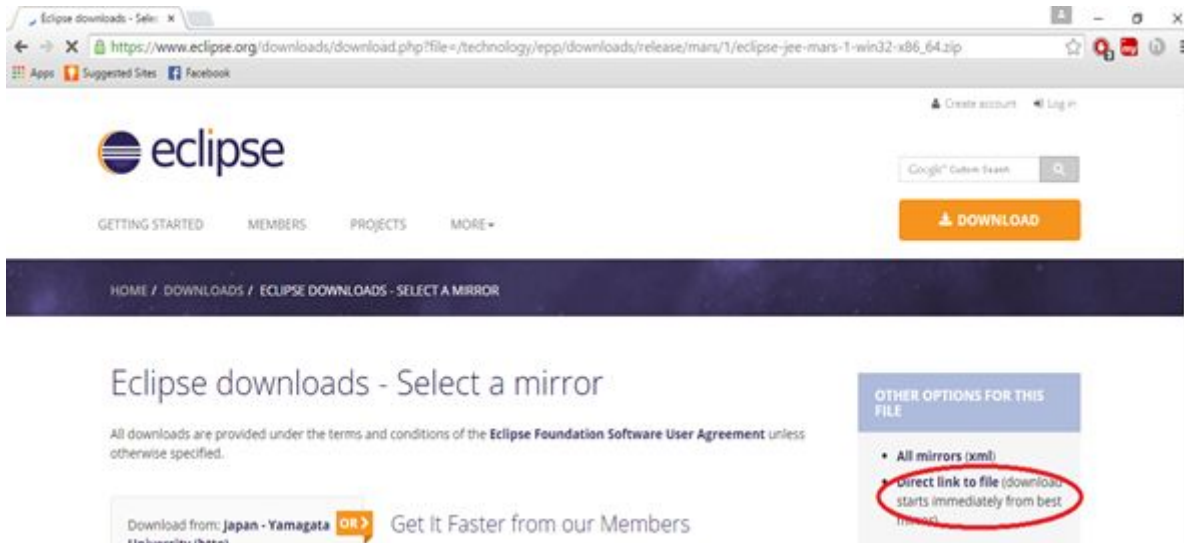
Fig 5: Window from where eclipse is downloaded

3) Now a setup of eclipse will be downloaded. It will be a .rar file. Right click on the file and select "Show in Folder". Once the folder opens right click on the file and select "Extract Files Here".
4) Now a folder "eclipse" will be created. Go into the folder and click on "eclipse".
5) On clicking on eclipse Eclipse IDE will opened.
Thus all the required software that is required for running the simulator is now installed.

3.4 Steps for running "simulator.exe" file

This file can be easily executed on a windows system. There are two versions available for the same simulator.
- simulator128m.exe
- simulator515m.exe

simulator515m.exe is prefered for extensive computation, while simulator128m.exe is light version for systems with less memory capability. Both the file will run directly on double click.

3.5 Steps for running "simulator.jar" file

1) Ensure that Java is installed in the system.
2) The simulator can be executed by simply double-clicking the "simulator.jar" file.

4. Raw and Mixed Capacity of a Section

Homogenous Capacity of a section is the maximum number of standard trains per day that can pass through that section. It is well-known that Scott's formula is a simple technique to obtain the homogenous capacity.

Scott's Formula for capacity of a section: capacity (number of trains per day)**:** $\frac{1440}{T+t}$ X $\frac{70}{100}$

> T- Running time of the slowest goods train over the critical block section (in minutes)
> t- Block working time (in minutes)
> 70% is the efficiency factor

In our simulator, the raw capacity computation is similar to the Scott's formula (without the efficiency factor).

The Scott's Formula is not really applicable in mixed traffic scenarios for a number of reasons. The main reason is that the achievable capacity of handling trains of different speeds and characteristics depends on the mix of trains and also on the sequence and schedule of trains in the mix.

The Scott's formula gives an insight on the capacity in simple scenarios. Using this formula across the board has led to the capacity of many sections to be reported above 100% which shows the incapability of this formula to capture the effects of mixed traffic. Hence there is a need for developing some other capacity calculation method.

4.1 UIC 406 Framework

As such there is no single definition of capacity that exists. Any given railway infrastructure capacity is dependent on the way it is used. The capacity not only depends on infrastructure characteristics such as the signaling system, speed restrictions, rolling stock characteristics, but also on the acceptable punctuality level and schedule of passenger trains vis-a-vis goods trains.

On a given infrastructure capacity is based on the number of trains, average speed, stability of the schedule and heterogeneity. The UIC 406 suggests an approach based on compressing the passenger train schedule based on the current occupancy pattern (to make use of all available spaces in the time space graph) and then inserting desired paths of new trains to get the achievable capacity. This ignores the impact of such a schedule on neighbouring sections, but is

justified, if done for the limiting or bottleneck section in a given network. The capacity can also be assessed using occupancies or headways in an equivalent manner.

4.2 Capacity Definition for Mixed Traffic for RDSO Simulator

This definition tries to answer the following question. Given a particular infrastructure on a section, and a passenger schedule, how many freight trains can we run with reasonable running times. First a freight train is run without any other conflicting trains, to get a free running time. Then with the infrastructure details and the schedule of passenger trains as input, a number of freight trains is computed so that the average traversal time is twice this free running time and this number is taken as the notional capacity of the section.

For a given set of freight trains fired at some good timings, the running times that result are tabulated in the ascending order of traversal times and the number where the average traversal times exceeds twice the free running time is taken as the capacity.

The major steps to implement this definition in simulator are:
- Running one simulation without scheduled trains to get an idea of the theoretical headway for the freight train
- Multiple simulations are possible to compute good timings for freight trains - this is a theoretically challenging problem and cannot be solved exactly in reasonable time, except for very small examples. In our simulator, we have currently fired freight trains in all reasonable gaps produced by the passenger train schedule and tabulated their running times.
- Freight traversal times versus firing time are stored in an array in ascending order of traversal time
- If the travel time of first freight train is K, then finding n where traversal time of n+1 freight would be more than 2K - generally at least some trains will find a good path, so K is taken to be close to the free running time.
- Apart from the capacity, this simulation also gives suggested freight paths for the capacity that is computed.

4.3 Use of tool for mixed capacity calculation

The recommended way using tool to find the mixed capacity, i.e. the number of freight paths, and the freight start timings, so that all the freight paths' traversal timings are not more than twice the best freight path, is to use the freight-details.xls file that is generated using the simulator. This is elaborated in the following section. If there are sufficiently many freight trains listed in the unscheduled.txt file, then the simulator simulates all of these and the freight-details.xls file lists all

the freight paths, sorted according to traversal timings. Thus, depending on the acceptable quality of freight paths, one can pick the freight paths (i.e. number of paths and the optimum freight start timings) so that there is a guarantee that the freight paths will be acceptable quality. While this can be iterated again and again, a single step is also sufficient for the purpose of finding optimum freight paths.

4.4 Freight-details.xls

Freight-details is an excel file generated after the simulation, which shows the time taken by each **freight** in **minutes**. The arrangement is in ascending order of time taken, so it can be used to identify the fastest freight path or set of paths among the given paths. Using the "View Trains" option inside "Trains" under "Add/Edit Infrastructure/Train" or by checking at "Unscheduled.txt" inside the infrastructure selected, the starting time of these fastest freight can be seen.

FreightNum	Direction	Speed(KMPH)	StartStation	StartLoop	EndStation	EndLoop	StartTime(HHMM)	Duration(Mins)
804	up	60	Gorakhpur-Gkp	10001	Barabanki-Bbk	340001	01:35	245.41
806	up	60	Gorakhpur-Gkp	10001	Barabanki-Bbk	340001	02:35	245.41
803	up	60	Gorakhpur-Gkp	10001	Barabanki-Bbk	340001	01:05	245.41
805	up	60	Gorakhpur-Gkp	10001	Barabanki-Bbk	340001	02:05	245.41
802	up	60	Gorakhpur-Gkp	10001	Barabanki-Bbk	340001	00:35	245.41
801	up	60	Gorakhpur-Gkp	10001	Barabanki-Bbk	340001	00:05	245.41
824	up	60	Gorakhpur-Gkp	10001	Barabanki-Bbk	340001	11:35	245.41
823	up	60	Gorakhpur-Gkp	10001	Barabanki-Bbk	340001	11:05	253.91
905	down	60	Barabanki-Bbk	341001	Gorakhpur-Gkp	11001	02:05	255.33
904	down	60	Barabanki-Bbk	341001	Gorakhpur-Gkp	11001	01:35	255.33
928	down	60	Barabanki-Bbk	341001	Gorakhpur-Gkp	11001	13:35	255.33
929	down	60	Barabanki-Bbk	341001	Gorakhpur-Gkp	11001	14:05	255.33
831	up	60	Gorakhpur-Gkp	10001	Barabanki-Bbk	340001	15:05	259.72
927	down	60	Barabanki-Bbk	341001	Gorakhpur-Gkp	11001	13:05	263.76
826	up	60	Gorakhpur-Gkp	10001	Barabanki-Bbk	340001	12:35	265.18

Fig 6: Screenshot of freight-details.xls

4.5 Comparator for studying effect on mixed capacity

The mixed capacity of the infrastructure can be studied using "freight-comparator" (section 8.1). The freight comparator can be used to compare two different timetables or can also be used to check a few stats about a timetable. It uses freight-details.xls as input and will show stats like:
- worst freight path: maximum time taken
- average of all freight paths: time taken
- average of best 10 freight paths
- average of best 20 freight paths
- average of best 30 freight paths

29

- average of best 40 freight paths
- average of best 50 freight paths

Using these stats, the mixed capacity of the infrastructure can be determined.

5. Modelling complex experiments: example cab signalling

This chapter describes how cab-signalling can be modelled using the simulator. The simulator is not primarily targeted towards this experiment, and hence this chapter can be skipped if cab-signalling is not the focus.

Cab means the crew compartment or the driver's compartment. In this method, the train has data about the location and movements of the train ahead of it in (near) continuous time. This signaling system gives the maximum capacity possible compared to all other signaling system (ceteris paribus) as every train has complete information about the location about the driver in front of it. In other systems with block working, the position of the train in front of it (position of a train refers to the position of the rear of the train) can only be known within a block. Thus, there are no arti cial blocks stopping trains for safety issues. The train will clear in continuous time, the part of the track which is some xed safety distance behind its rear. This distance between two trains is always maintained so that in no case will two consecutive trains come closer than this distance, thus ensuring safety. This is much like tra c on a road, where car drivers visually know the location of the car in front of them all the time and maintain a safety distance while driving their vehicles. Cab signaling is used in some parts of the world and news articles have mentioned the interest of railway authorities in using this technology for the Harbour line of Mumbai suburban railways.

Advantages: Maximum capacity can be achieved with this system while maintaining safety. No reliance on line side signals, either automated or manned.

Disadvantage: Unless all trains in a given section are equipped with the technology, cab signaling can't be used.

5.1 Modeling cab signaling in the simulator

The simulator implements block signaling for generating valid train schedules. The Railway simulator tool assumes that the track is divided into blocks and compulsorily requires that the whole track is divided into blocks and loops. Note that for absolute and intermediate block, it assumes that the response time for setup of next signal is zero even though there may be some delay due to manual skills employed. Now:

Cab signaling system does not have xed blocks per se. There are no line side signals involved with signaling which divide the track into smaller parts/blocks. Thus, the question arises as to how to model this signaling method with the simulator's logic.

Claim: Discretization: Divide the track into a large number of arbitrarily small blocks. Also ensure: Number of colors/aspects of simulation (col) = number of blocks (n) + 1.

The number of signal aspects, in automatic signaling with 4 aspects are: Green, double yellow, yellow and red, tell us how many blocks ahead of a given train is the train before it, as was seen in 2.1.3. One can theoretically have as many colors as one desires to get more information about the train in front of it in a similar way in the simulator's logic. Next we prove our claim.

Proof : The basic reason for adopting cab signaling is that it instantaneously clears the region which was previously occupied by the rear of any train for the train following it. In block system, the whole length of the blocks needs to be cleared before the next train can enter that patch of the track.

If block lengths are kept infinitesimally small, then the rear of the train will clear consecutive blocks very quickly, thus making it available for the following train in almost the same time as they are cleared, as is desired in cab signaling. Thus, it seems that a large number of small blocks should closely approximate cab signaling.

Coming to number of aspects: What cab signaling really means is that the train has real time data about the location of the train in front of it. Let's assume that there are very small blocks but only 2 signal aspects i.e. red and green. This does not capture the cab signaling behavior we want to model as although the trains clear consecutive blocks very quickly, a train can only know whether the one ahead of it has cleared at least the block right in front of it or not. It cannot determine the location of the train.

To mitigate this, as many colors as required should be used to accurately track the train ahead. One can easily see that we don't need more than n+1 colors as even if we did, the additional colors would never be used, as there are not enough blocks to see the remaining colors (red (1), Y (2), YY (3) up to n+1 colors may be observed, but none beyond that, as there are no blocks). Thus, we set col=n+1.

Essentially, with this setup, the next train can accurately locate the previous train's rear position within an error distance of L (+/- L/2 on either side of the middle of the block), where L is the size of the block the previous train's tail is in. As we use smaller and smaller blocks, this error tends to 0 and we closely approximate a feasible cab signaling solution.

5.2 Standard section exercise

This exercise was conducted to observe the effects of capacity on a gradient free, speed restriction free track, where the trains are uniform. The capacity definition used was to calculate the maximum number of trains that can be operated in a day, regardless of punctuality, average speed or heterogeneity or other factors, while assuming that new trains are always available for ring when there is a free train path. The train and section settings, different simulation settings, results and conclusions of this exercise are described next.

5.3 Train and track settings

Maximum velocity of train = 25 m/s = 90 km/hr

Acceleration = 0.2 m/s^2

Deceleration = 0.2 m/s^2

Size of train = 0.5 km (Note: The train can only use the specified acceleration and deceleration in the simulator and these are not indicative of a range. Also, the velocity profiles generated ensure that the train maintains the maximum velocity possible for as long as it can, accelerate as fast and early as possible and decelerate as late and quickly as possible.)

Maximum velocity on all blocks and loops, station entry velocity = 27.78 m/s = 100 km/h

Distance between the 2 stations = 10 km (This is the block length in absolute block signaling) Loops size (same as station size in the simulator) = 1 km (changed for cab signaling)

Settings for different simulations

1. Absolute block: block between station A and B. Block working time of 1 min considered.

2. Intermediate block signaling: 2 blocks each of size 5 km. Block working time of 1 min considered.

3. Automatic block signaling: 10 blocks each of size 1 km with 4 color aspects.

4. Cab to cab signaling: Blocks and loops of sizes 100 m each, thus total of 120 blocks and loops. Numbers of colors/aspects were kept at 121 using the criteria of c=b+1 as discussed previously. The safety distance for cab signaling is taken as 100 m in this analysis. Safety distance is the minimum distance to be maintained between a train's rear and the next train's front. This distance was implemented by increasing the length of the train to 600 m.

5. Train settings: 5 identical trains as defined by the parameters in the previous subsection scheduled in such a way that the next enters the loop at A and is ready to get on the track as soon as possible. This is achieved by keeping their scheduled departures from the station within a minute.
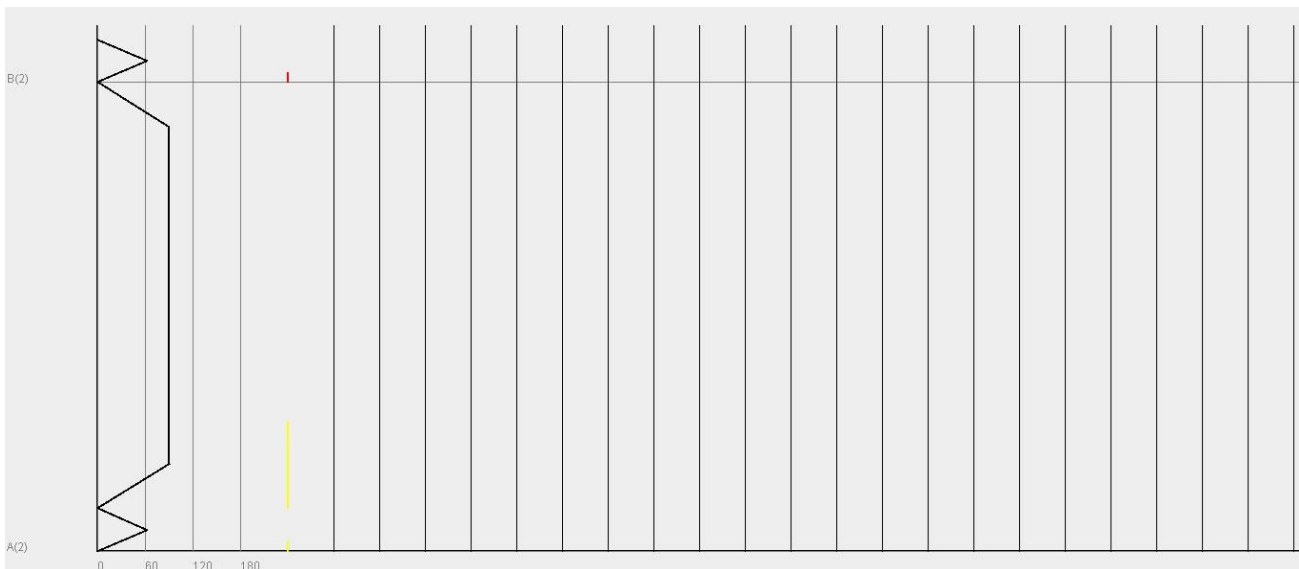
6. Simulation time = 1 day.

Fig 1: Absolute block signaling - standard section
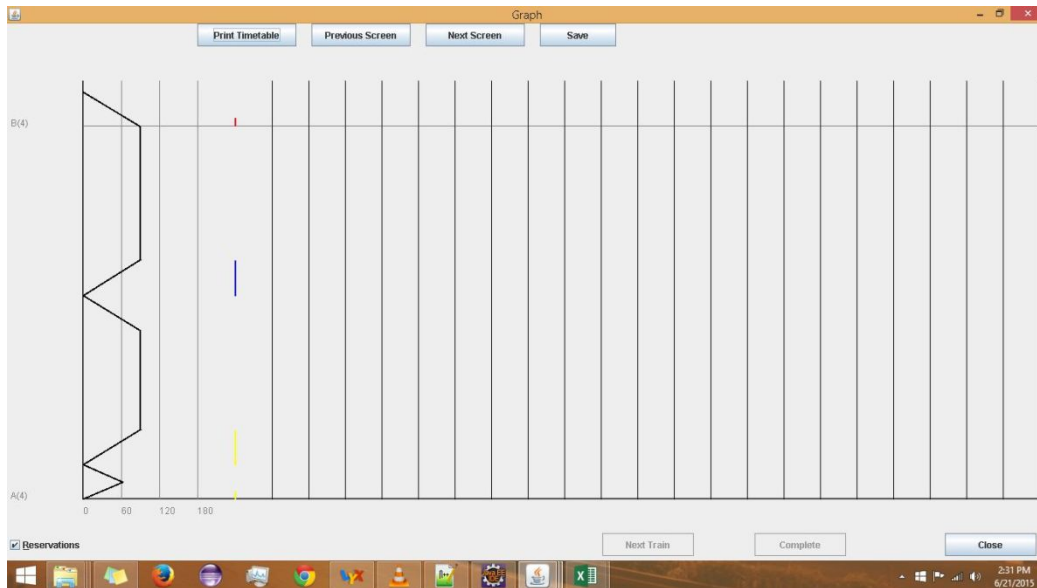
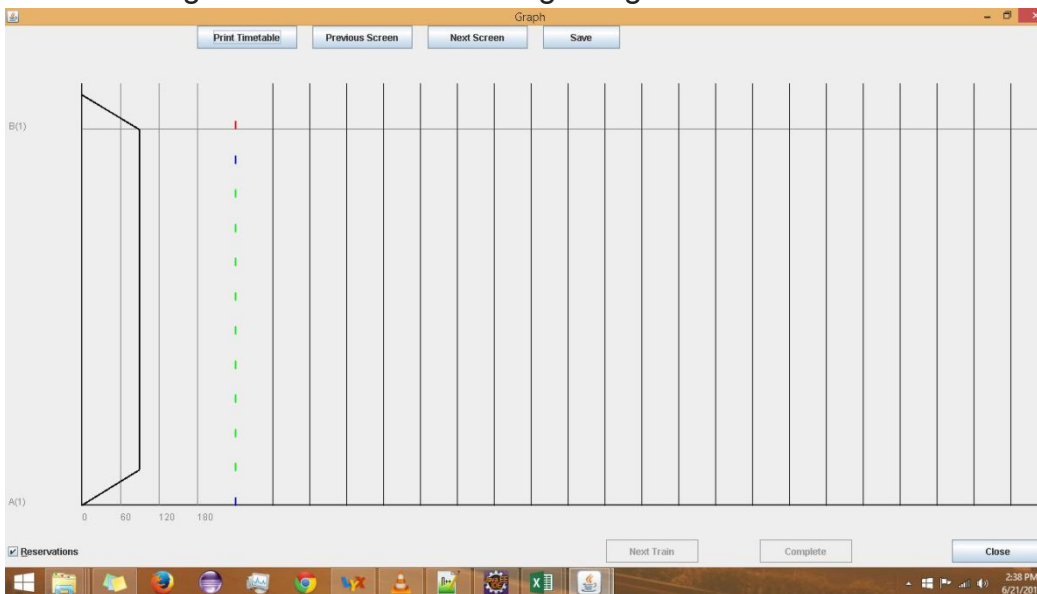Fig 2: Intermediate block signaling - standard section

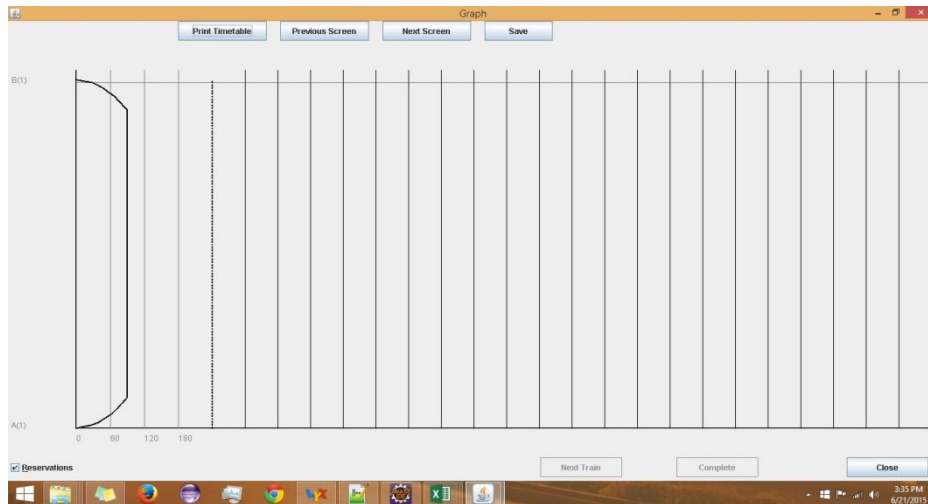Fig 3: Automatic signaling-Automatic section

Fig 4: Cab Signaling- Automatic Section

5.4 Results of exercise

Typical nature of velocity profiles of trains for different signaling systems experiment:

The following table summarizes the results obtained from the analysis of the schedules generated by the simulator:

Table 1: Standard section results

Signaling System	Latency (min) (departure at B- departure at A)**	Headway (min) (time difference to leave/arrive between 2 consecutive trains)	Line capacity(trains/day using Scott's formula(60% efficiency)*	Increase in capacity wrt absolute block	Increase wrt IBS	Increase wrt automatic block
Absolute Block Signaling	11.1	10.05	85	NA	NA	NA
Intermediate block signal	11.5	6.75	128	50%	NA	NA
Automatic Block Signaling	8.4	3.2	270	217%	111%	NA
Cab signaling	9.5	2.5	346	307%	170%	28%

*This is an empirical factor used in Indian Railway literature .
**Wait times at station B are 0: trains just pass through. We include the departure time within the latency.

5.5 Standard section conclusions

1. Waiting time, speed restrictions, loop entry velocity etc. aren't considered in this study. Also, it is assumed that trains are available at all times for leaving the station and they are scheduled the moment the block ahead becomes free. Thus, the capacity observed is much higher that what would be observed in practice for all forms of signaling.

2. The simulator schedules trains according to a FIFO basis and schedules trains immediately. This explains the output obtained in intermediate block signaling: trains will wait at the end of the rst block. If they had been red a certain time later, they would have found the second block also free. Thus, the capacity obtained is under the strategy described and don't correspond to the normal operating conditions of trains.

3. The capacity of cab to cab signaling is over 3 times that of absolute block signaling. This is a huge improvement and shows the potential it has if implemented on real sections currently using absolute blocks. Cab signaling gives a similar order of capacity as auto blocks of 1 km. A similar result is obtained in chapter 4 also. Thus, a cost benefit analysis should be done for the same (auto to cab) before implementing it on a real section.

5.6 Secunderabad Wadi section exercise

SC-W is a section in South Central Railway (SCR) zone of IR. This section has 21 stations with absolute blocks. Few station pairs have intermediate blocks also. Station jurisdictions, loop con gurations, gradients and signal information were provided within SC division rolling diagrams, route wise indexes and the working time table. We were provided with the gradients on the track, the loops at the stations. Details on other parameters were assumed as follows:

Length of train: 500 m
Acceleration: 0.5 km/min^2
Deceleration: 1 km/min^2
Maximum velocity: 90 km/h
Loop velocity: 20 km/h
Block velocity: 100 km/h

Capacity of section was defined again in a manner similar to the last section: What is the maximum number of trains of the type mentioned that can be pushed in a day without halting at intermediate stations and just going through from start to end of the section. Waiting time at stations wasn't considered here either. The current signaling system is intermediate block signaling and we want to measure the impact of cab signaling on the section. Now, the SC-W division is around 194 km long. To simulate cab signaling on this track would require roughly 1700 small blocks each of size 100m and correspondingly 1501 colors (assuming loops of 1 km each). As the version of simulator on which this exercise was conducted had manual data entry directly into text les involved with the back end of the code, this was infeasible. To resolve it: As seen on the standard section, even 1 km blocks with 4 signal aspects gave the same output as cab signaling. We used this result over here and approximated cab signaling with 1 km blocks and 4 color aspects on the section. 21 trains ready to leave as soon as a path is available were used again in this exercise. Typical velocity profile generated on the track and the results of the exercise

are:

Thus, with mentioned assumptions and definition of capacity, there is a 90% increase in capacity on a real section of IR by adopting automatic/cab signaling over intermediate blocks.
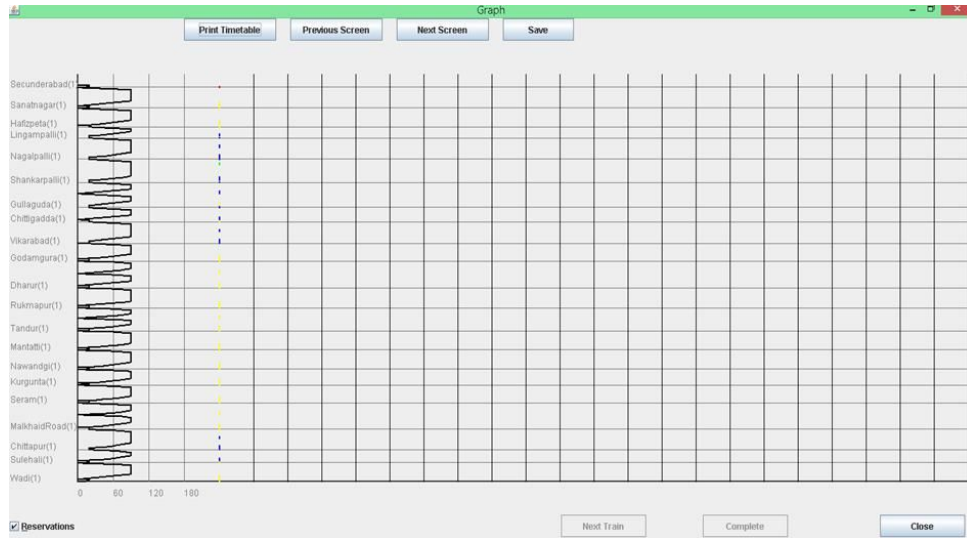

Fig 18: Velocity profile on SC - W division

Table 2:   SC - W intermediate and auto/cab signaling

Signaling Method	Average Latency (hr)	Capacity (trains/day)
Existing Intermediate Block	3.6	140
Cab Signaling/Automatic Block	3.2	267

6. Starting a new infrastructure

This chapter contains procedure to start/initiate infrastructure details inputting.

Run the simulator: **java -Xss515m -jar simulator.jar** from the DOS command prompt**,** or double-click on the **jar-file** or the appropriate **exe file.**

Depending on the local PC and the available/required RAM memory, there are multiple files kept available for download. The jar-file within the simulation exe files are the same as in the simulator-zip-file. The jar-file execution is with the PC's *default* memory size: often too low for executing when the number of trains is large and/or the infrastructure is complex.

- The jar-file is available within the zip file: simulator-zip-file, and execution procedure of the jar-file (after unzipping) is by either double-clicking the simulation-file.jar, or typing "java -Xss515m -jar simulator-file.jar" on the DOS command prompt.

Here "515m" stands for 515 Megabytes of memory allocation (often needed for more complex infrastructures). If the PC cannot allocate 515 Megabytes, then use "-Xss128m" instead: this is ample for simpler infrastructures

Alternatively, a self-extracting Windows simulator128exe-file (for PCs with moderate RAM memory availability) can be executed by double-clicking on this exe file. This exe file has the jar-file and will execute with 128MB of memory.

For PCs having more RAM memory: please use simulator515exe-file for execution. This exe file has the jar-file of option 1 and executes with 515MB of memory.

DOS command prompt
In MS-Windows, one can start the command prompt by typing the windows key (usually between the left Ctrl and Alt keys) and typing "cmd" in the "Search programs and files" space. Once the DOS-command-prompt is started, use "cd" to go to the correct directory and then type the above command.

Once the simulator has been successfully started, in the top-left corner, there are two menus, viz. Simulator and Comparator.

The detailed description of simulator menu:
1) Simulator menu contains the heart of the GUI. Almost all the functionalities lie in this menu. The sub-menus in this menu are:
    a. Add/Edit Infrastructure/Train
    b. Select Infrastructure/Train
    c. Simulate
    d. Calculate Mixed Capacity
    e. Calculate Raw Capacity
    f. Exit
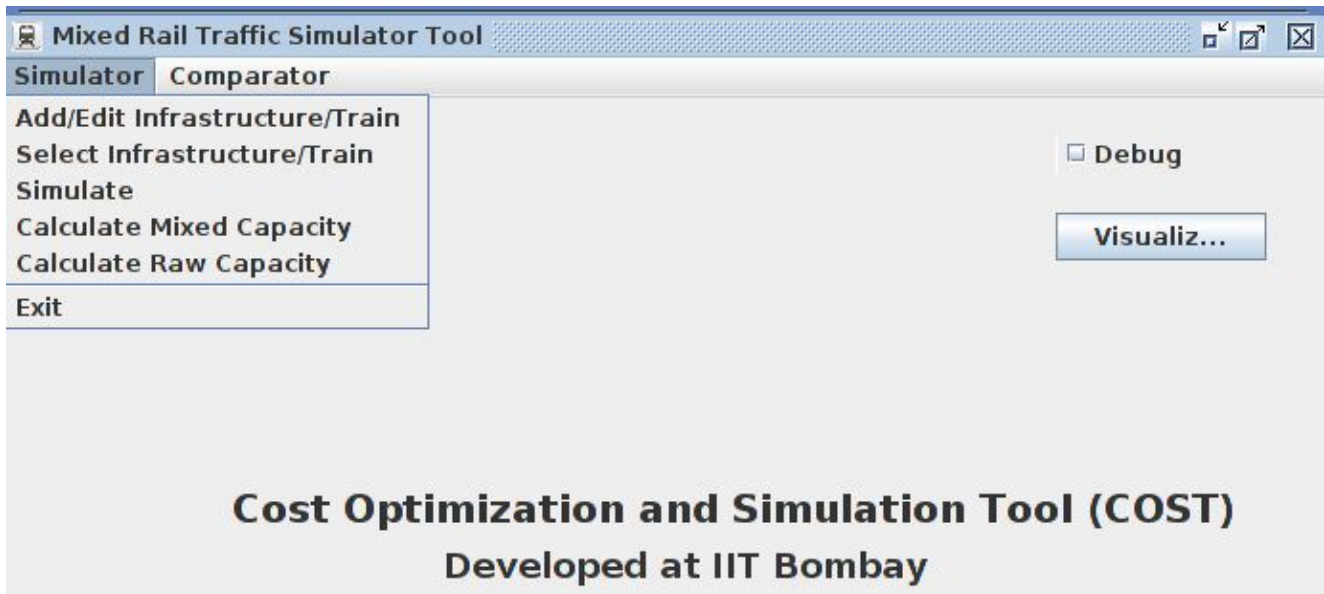
2) Comparator menu contains options:
      i) Freight Comparison
      ii) Advanced Comparison
details of both the features have been elaborated in the section *Comparator*
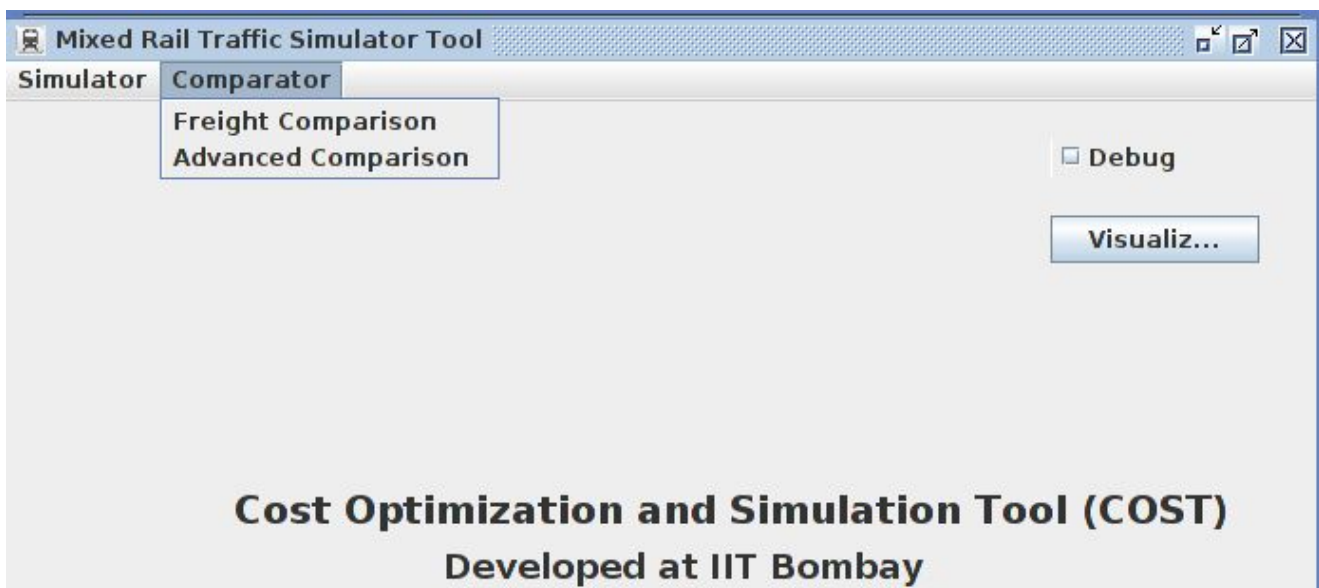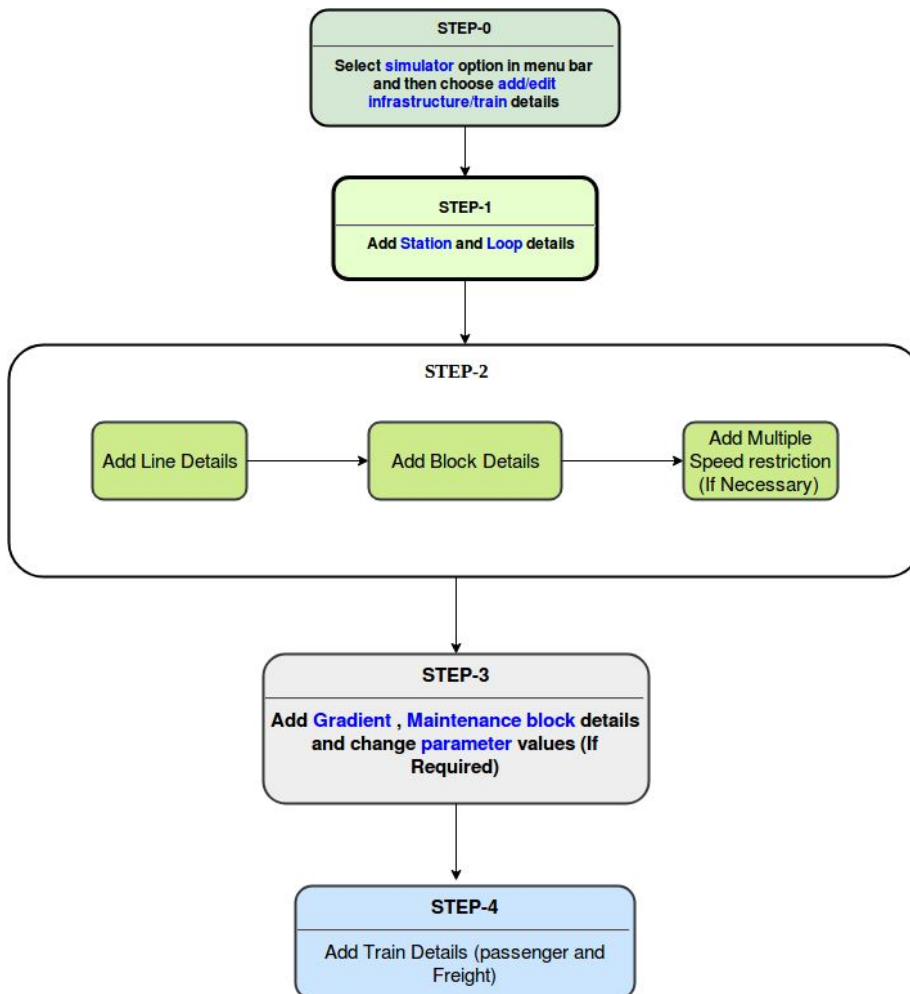
3) Visualization button in the main window displays the infrastructure of the selected section, details of the same is given in the section *Visualization*
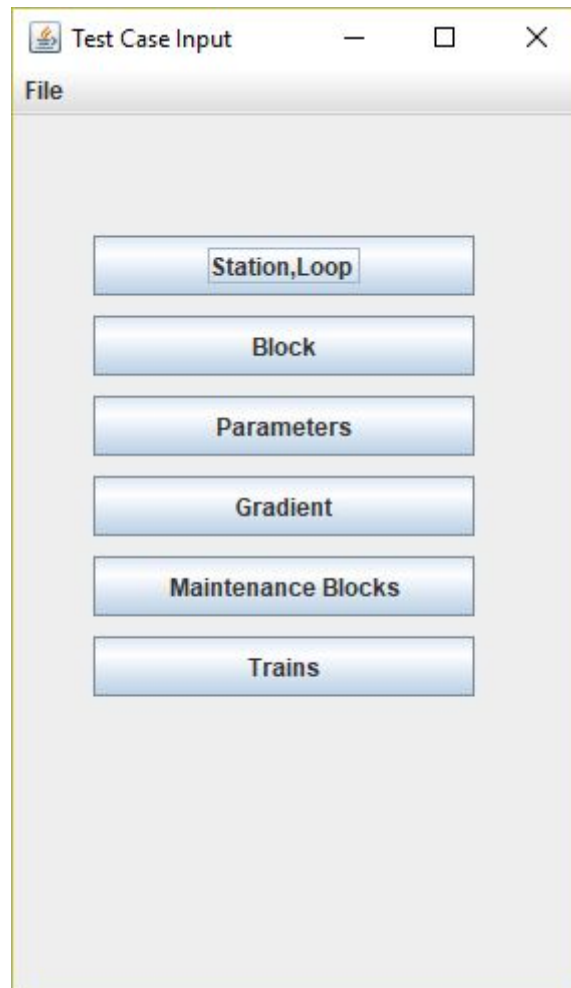
4) Debug check box on the simulator main window is for the Debug feature for simulation. This option is to print verbose text data about the runtime operation of simulation into the file *debug.txt*. It is recommended to keep this option unchecked, unless a review on the performance of the simulator is required.

6.1 Constructing input files

This section explains in detail on how to construct the prepare the input files using GUI.
A flow chart is given below and each steps are detailed in the further section.

```
                    ┌──────────────────────────┐
                    │         STEP-0           │
                    ├──────────────────────────┤
                    │ Select simulator option  │
                    │ in menu bar and then     │
                    │ choose add/edit          │
                    │ infrastructure/train     │
                    │ details                  │
                    └──────────────────────────┘
                               │
                               ▼
                    ┌──────────────────────────┐
                    │         STEP-1           │
                    ├──────────────────────────┤
                    │ Add Station and Loop     │
                    │ details                  │
                    └──────────────────────────┘
                               │
                               ▼
    ┌──────────────────────────────────────────────────────────┐
    │                        STEP-2                             │
    │  ┌──────────────┐   ┌──────────────┐   ┌──────────────┐  │
    │  │ Add Line     │──▶│ Add Block    │──▶│ Add Multiple │  │
    │  │ Details      │   │ Details      │   │ Speed        │  │
    │  │              │   │              │   │ restriction  │  │
    │  │              │   │              │   │ (If Necessary)│ │
    │  └──────────────┘   └──────────────┘   └──────────────┘  │
    └──────────────────────────────────────────────────────────┘
                               │
                               ▼
                    ┌──────────────────────────┐
                    │         STEP-3           │
                    ├──────────────────────────┤
                    │ Add Gradient ,           │
                    │ Maintenance block        │
                    │ details and change       │
                    │ parameter values (If     │
                    │ Required)                │
                    └──────────────────────────┘
                               │
                               ▼
                    ┌──────────────────────────┐
                    │         STEP-4           │
                    ├──────────────────────────┤
                    │ Add Train Details        │
                    │ (passenger and Freight)  │
                    └──────────────────────────┘
```

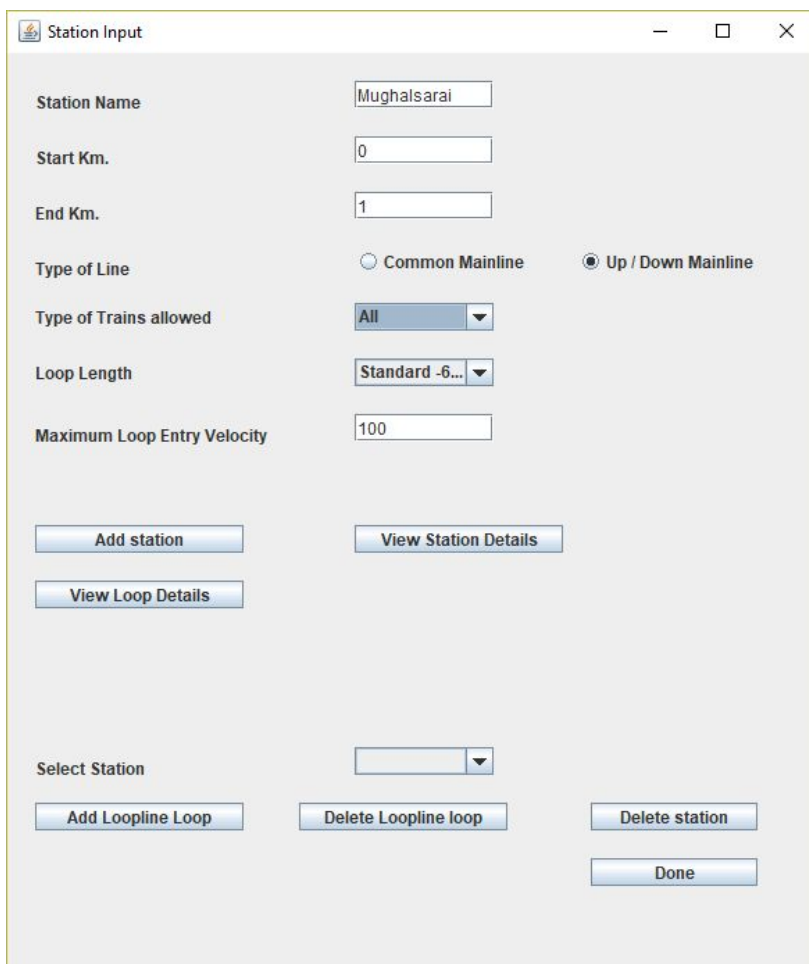## 6.2 Add/Edit Infrastructure/Train

Here you can create the .txt files as inputs to the simulator from scratch.

It must be noted here that the txt files are ideally entered and edited using the GUI. However, one can also use a spreadsheet tool (like MS Excel) to view/edit any detail. This should be done with more care since the format can get muddled. It is recommended to store a backup of relevant files (with a suggestive and time-stamped name) before opening in MS Excel. Procedure to open in MS Excel is elaborated in Appendix C at the end of the manual. The header for the relevant txt file is described within the sections below.

6.2.1 Creating "loop.txt" and "station.txt" files

The "Station, Loop" button: This consists of infrastructure within a station and the following subsections elaborate on this.

6.2.1.1 Adding a New Station/Loop

- Insert a station name in the "Station Name" field.
- Next enter the starting distance of the station from the reference point (which is usually the starting point of the first station) in the "Start Km." field.
- Now enter the ending distance of the station from the reference point in the "End Km." field.
- In the "Type of Line" checkbox, select the desired line type viz. Common Mainline or Up/Down Mainline.

    Note: Up/Down Mainline inserts 2 lines which are up line and down line respectively. Whereas Common Mainline inserts a single line which allows both Up and Down trains.

- Now select the type of train (All/Passenger/Freight) allowed on the loop from the "Type of Trains allowed" drop-down menu.
- Loop length can be selected from the next "Loop Length" drop-down menu.
      (By default it is set to "Standard -686".)
- Now enter the Maximum Loop Entry Velocity: By default set to 100 Kmph.
- Once the above steps are followed click on "Add Station".
- Enter at least 2 stations.
- To view all the stations added and their respective parameters, click on "View Station Details".
- Similarly loop details can be extracted from the "View Loop Details" option.

6.2.1.2 Editing the Existing Station/Loop

Once the infrastructure is generated, one might need to edit the station/loop details or delete them all together.

- Select the station to be edited from the "Select Station" drop-down menu.
- There are three options :

    I.   Add Loopline Loop

    II.   Delete Loopline loop

III.   Delete Station

A: Add Loopline Loop

- Select a direction (Up/Down/Common on up side/Common on down side/Common in middle) for the new loop to be added to the selected station.
- Now select the type of trains (All/Passenger/Freight) arriving at the loop.
- Now select the loop length from the following drop-down menu: by default it is "Standard -686".

48

- Specify the Maximum Loop Entry Velocity in the next field which is pre-assigned the value 15.
- All the loops at the specified stations can be viewed by clicking on "View All".
- Now click on "Add new loop" once desired parameters are selected to add that loop to the selected station.
- Repeat the above procedures to add as many loops as desired and then click on "Done".

B: Delete Loopline Loop
- Once a loopline is deleted, the selected station no longer has that loop anymore.

C: Delete Station:
- The selected station can be deleted using this option.
- Caution required as this process is irreversible.

Once the desired list of stations is created, click on "Done" to proceed.

```
/*Loop Number    Direction  Type  Station Code  Type-of-Trains-allowed   Lo

21001 down ml "BHK" all   100 "" "" "" "" "" "" "" "" "" "" ""
10001 up ml "Mughalsarai" all   100 "" "" "" "" "" "" "" "" "" "" ""
11001 down ml "Mughalsarai" all   100 "" "" "" "" "" "" "" "" "" "" ""
```

Fig. loop.txt

```
/*Name     startKM     endKM     stationEntrySpeed  */

"Mughalsarai" 0.0 1.0 100.0
"BHK" 5.0 6.0 100.0
```

Fig. block.txt

6.2.2 Creating Blocks: the "block.txt" file

Ensure at least 2 stations exist to define the section between them before selecting this function or you can't access this function.

Lines/Block Section Input

Please enter the initial station for inserting block section

Bhk ▼   Tundla ← Next Station

Please choose block section type

○ Bidirectional Section

◉ Up Block Section

○ Down Block Section

Number of Lines

2 ▼

Enter Up-Line 1   +

Enter Up-Line 2   +

| View All | Edit Blocks | Done |

● The window first displays a blank screen with a drop down selector, a station needs to be selected to access further functions.

- Select the starting station after which the blocks are to be added. This will initiate next level of functions.
- Select the block type from **Bidirectional**, **Up Block** and **Down Section** and also number of lines from the dropdown menu, maximum of three lines can be added for each direction.
- For each line, a '+' button provides addition of block details for each line on a new window.

- The drop down button on Top is for the **number of blocks** to be added for each line.
- **Block Start** km of the first block and **Block End kilometre** of the last block and maximum speed are already filled by default.
- **End kilometre** once entered for any block, the **Start kilometre** for the next block is filled automatically with the same value.
- The '...' button after each block generates another window for inputs on **multiple speed restrictions** on the same block.

- Enter **Speed Rest.**, **Start kms.**, **Speed Rest.**, **End kms.**, and **Speed Limit**.
- The '!' symbol shows error in each entry for the speed restriction. the details of start and end km of the speed restriction must lie inside the block length.
- Select '**Done**' after completion of each entry in the window.
- Caution must be maintained while entry of these details.
- In the main block window "Edit Block" button deletes all the previous entries of the line/section between the two stations and all the details can be entered again.
- At the bottom select **View All** in the main block window to view all the block data.

- Remember to click on **Submit** to save the block data and then **Done**.

```
/*Block Number  Direction  Starting Mile Post   Ending Mile Post      Block Velocity (kmph) Uplinks    Up Link Length  Priority  Cross-over Down link
22010 common 6.0 7.0  100 "22020" "0.0" "1" "#" "21001,21101,22201,22301,22101" "0.0,0.0,0.0,0.0,0.0" "1,2,5,3,4" "#,#,#,#,#" "80.0" "15.0" "50.0"
22020 common 7.0 8.0  100 "22030" "0.0" "1" "#" "22010" "0.0" "1" "#" "80.0" "0.0" "0.0"
22030 common 8.0 9.0  100 "22040" "0.0" "1" "#" "22020" "0.0" "1" "#" "80.0" "0.0" "0.0"
22040 common 9.0 11.0  100 "30001" "0.0" "1" "#" "22030" "0.0" "1" "#" "80.0" "0.0" "0.0"
```

Fig. block.txt

In case maintenance blocks or speed restrictions within a block need to be edited, then this editing can be done using a spreadsheet tool like MS Excel. Procedure to edit the input txt files (like block.txt) using MS Excel is elaborated in . In the case of block.txt, when opened in MS Excel, the speed restrictions can be edited in the relevant column by keeping note that multiple subsections within a block having possibly different allowed speeds are to be specified by

Table 5: Header of block.txt file (when opened in a spreadsheet tool)

/*Block-number	...other-columns...	Reduced-speed-in-kmph Speed1,Speed2,Speed3	Start-milepost Start1,Start2,Start3	End-milepost*/ End1,End2,End3
30010	...	35,56,90	0,0.43,3.5	0.43,1.2,3.7
31010	...	30,50	1,1.5	1.4,1.8

Note: The distances of the subsections' start and end milepost is relative to the block start point and not absolute distances from the section 0 km milepost.

6.2.3 Storing important parameters: the "param.dat" file

54
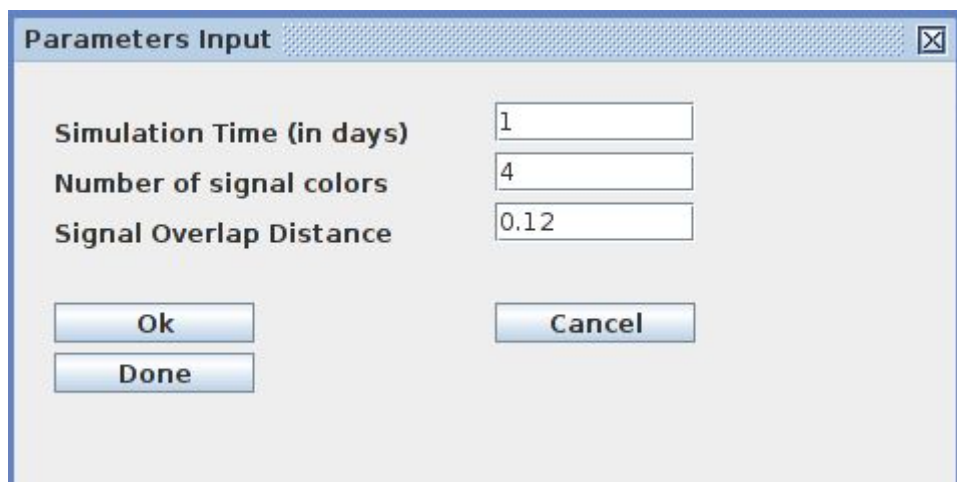
- This window contains 3 entries.
- **Simulation Time** is for number of days the simulation is to be run, by default it is 1 day
- **Signal overlap distance** entry takes value for distance of overlap in km.
- The fields are pre-filled. Changing the values will reflect in param.dat file.

```
/*SimulationTime    DelayFactor    blockWorkingTime    noOfSignalColors    redFailWaitTime    redFailWaitVelocity    warnerDistance    */
1 0 0 4 1 18 0.0
```

Fig. param.dat

6.2.4 Inputting Gradients

This section elaborates on the topic of gradients and also about the "gradientList.txt" and "gradientEffects.txt" files.

- **Gradient Value** field will be initialized to zero and **Level** check box will be selected by default.

- Change the inclination by selecting the appropriate value from the drop-down menu in **Up/Down** direction.
- A particular inclination can be **Edited** or **Deleted** by specifying correct **Gradient Value,Slope Distance** and **Direction.**

```
/* GradientValue    direction    startKM    endKM  */

"1/200" Up 0.0 6.0
```

Fig. gradientList.txt

```
/* GradientValue  Direction  accelerationChange  decelerationChange */

"1/200" ▯ -0.11 0.11
```

Fig. gradientEffects.txt

6.3 Creating "BlockDirectionInInterval" file

6.3.1 Maintenance Blocks

- All the loops and blocks are available for maintainance.
- Select the loop/block number from the drop-down box.
- Select the number of blocks and then enter the timing for the maintainance.
- Select start and end time for the block along with the direction.

```
/*Loop/Block Number  Start-time End-time  Direction  */

22010 "02:30:00,," "03:00:00,," "up,,"
```

Fig. BlockDirectionInInterval.txt

6.4 Creating "scheduled.txt" and "unscheduled.txt" files: trains

- Select the **Input Type** as **Loco-load** or **Train dynamics**
  - › Loco-load type will allow user to select the loco type and enter the loco details.
  - › Train Dynamics allows user to add kinematical details of the loco
- **Tweak** button can be used to change specific dynamics of the loco type.
- **See Effect** button shows the kinematic details of the selected loco type.
- Enter **Train Number** first and then select the **Input Type** and **Train Type** (Scheduled or Freight)
- Select the **Train Direction** viz. **Up** or **Down**.
- Trains can be edited or deleted by train number.

59

scheduled - Notepad

File   Edit   Format   View   Help

/*Train No	Direction		Length	Acceleration		Decelaration		Priority		Speed	Op.Days Number Of Halts Loop No.		Arr	Dep	Loop No.		
64583	up	0.5	0.3	0.8	1	100	all	25								
64157	up	0.5	0.33	0.8	1.05	100	all	11								
12301	up	0.5	0.13	0.8	1.1	130	all	19	10001	4700	5700					
12423	up	0.5	0.18	0.8	1.1	130	all	19	10101	5700	10700					
12313	up	0.5	0.18	0.8	1.1	130	all	19	12201	10700	11700					
22811	up	0.5	0.18	0.8	1.1	130	all	19	12202	11700	12700					
12877	up	0.5	0.18	0.8	1.1	130	all	19	10001	14700	15700					
12003	up	0.5	0.23	0.8	1.1	130	all	18								
19314	up	0.5	0.13	0.8	1.6	110	all	2								
12225	up	0.5	0.13	0.8	1.6	110	all	18								
19166	up	0.5	0.13	0.8	1.6	110	all	2								
13237	up	0.5	0.08	0.8	1.6	110	all	2								
12581	up	0.5	0.13	0.8	1.6	110	all	20								
14863	up	0.5	0.13	0.8	1.6	110	all	5								
64153	up	0.5	0.33	0.8	1.61	100	all	16								
12349	up	0.5	0.08	0.8	1.62	110	all	19	10101	14700	20700					
64165	up	0.5	0.3	0.8	1.63	100	all	11								
12817	up	0.5	0.13	0.8	1.64	105	all	21	12201	20500	21500					
11016	up	0.5	0.13	0.8	1.64	110	all	2								
13007	up	0.5	0.25	0.8	1.64	110	all	24	12202	22500	24000					
12150	up	0.5	0.13	0.8	1.64	110	all	3	10001	24200	25200					
12948	up	0.5	0.13	0.8	1.64	110	all	4	10101	30700	32200					
51804	up	0.5	0.13	0.8	1.64	110	all	2								
12303	up	0.5	0.13	0.8	1.64	130	all	20	10001	193700	194700					
22405	up	0.5	0.13	0.8	1.64	130	all	20	10101	225000	230000					
12561	up	0.5	0.13	0.8	1.7	110	all	19								
13201	up	0.5	0.15	0.8	1.7	110	all	10	12201	41200	42200			40001	44500	44700
63237	up	0.5	0.3	0.8	1.71	100	all	20	12202	43000	43000	30001	44700	44800	40001	45600	45700
12175	up	0.5	0.13	0.8	1.72	110	all	5	10001	45500	50500					
64591	up	0.5	0.35	0.8	1.73	100	all	31								
12323	up	0.5	0.13	0.8	1.75	110	all	19	10101	51000	52000					
12541	up	0.5	0.13	0.8	1.75	110	all	2								
15483	up	0.5	0.18	0.8	1.75	110	all	35	12201	53500	54500					
15004	up	0.5	0.23	0.8	1.75	110	all	7								
64168	up	0.5	0.33	0.8	1.76	96	all	15								
12311	up	0.5	0.23	0.8	1.77	110	all	26	12202	62500	64000					

Fig. scheduled.txt
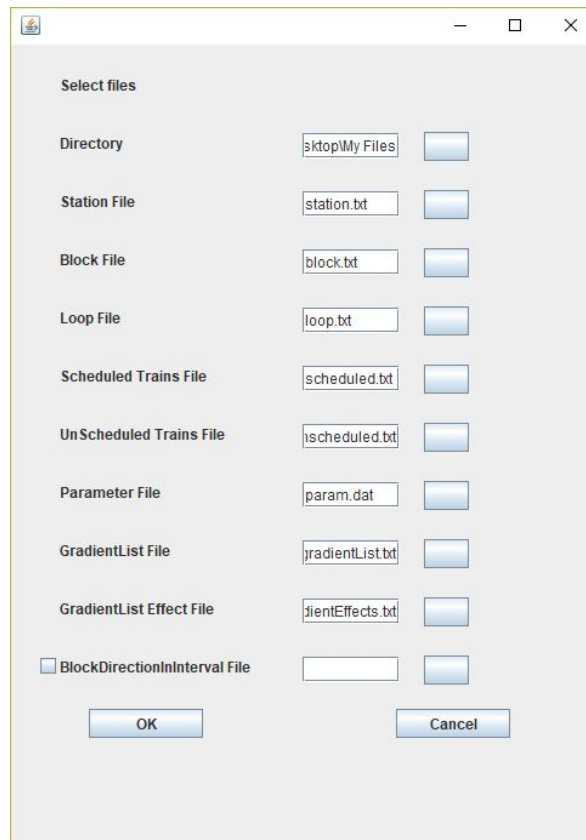
7. Running the simulator:
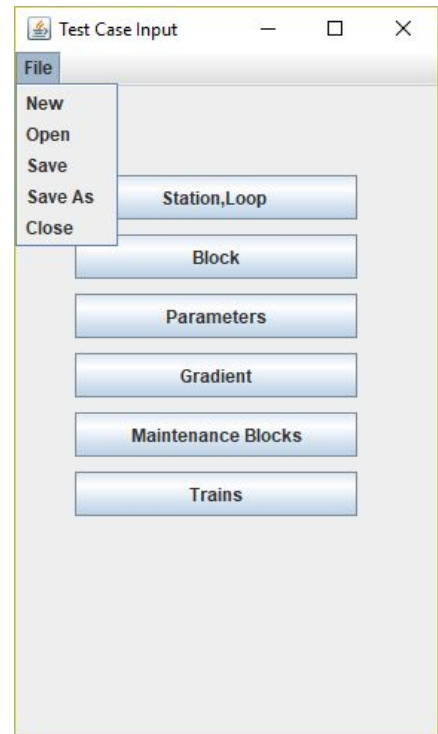
- Once the files are ready, click on **Save** from the **File** menu.
- Now click on X marked in the top right corner. **Close** option closes the entire GUI.
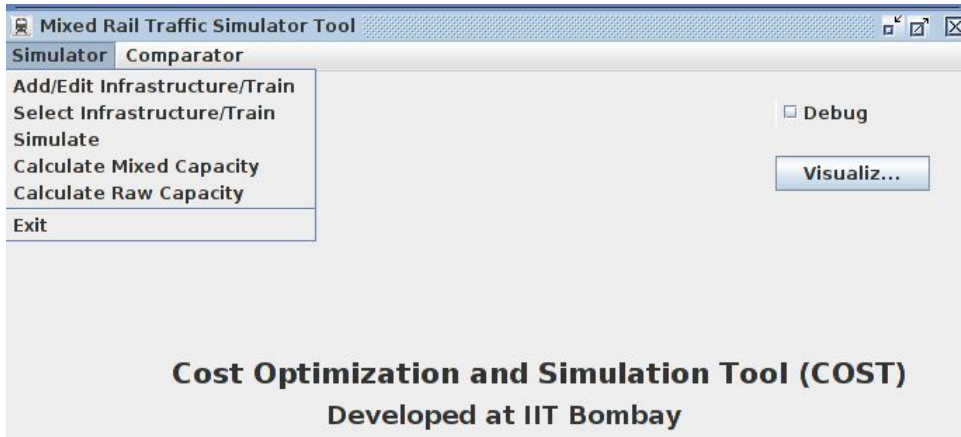
- Click on **Select Infrastructure/Train** to select the directory in which the above created files are present.

- The **Directory** if selected correctly will automatically fill all the remaining files except **BlockDirectionIntervalFile** which is the maintainance file.
- If needed manually check the box and select the maintainance file.
- Now click on **OK.** The dialog box will close and you return to the main GUI.
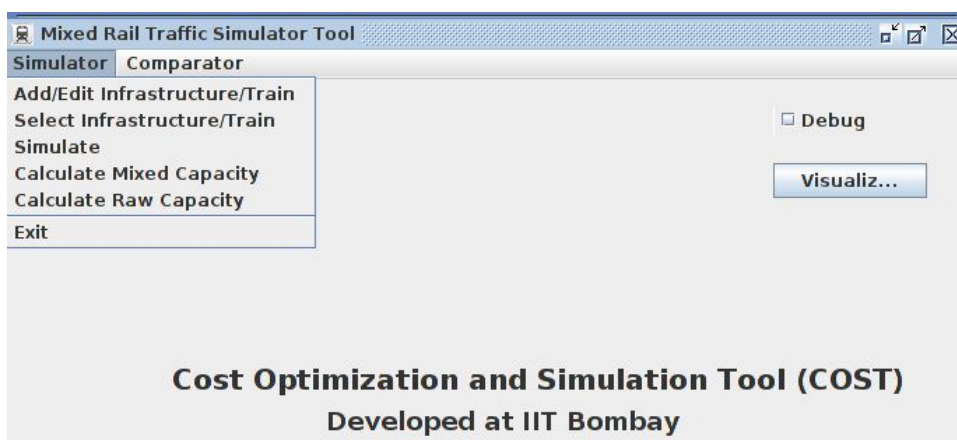
- Click on **Simulate** to run the simulator.
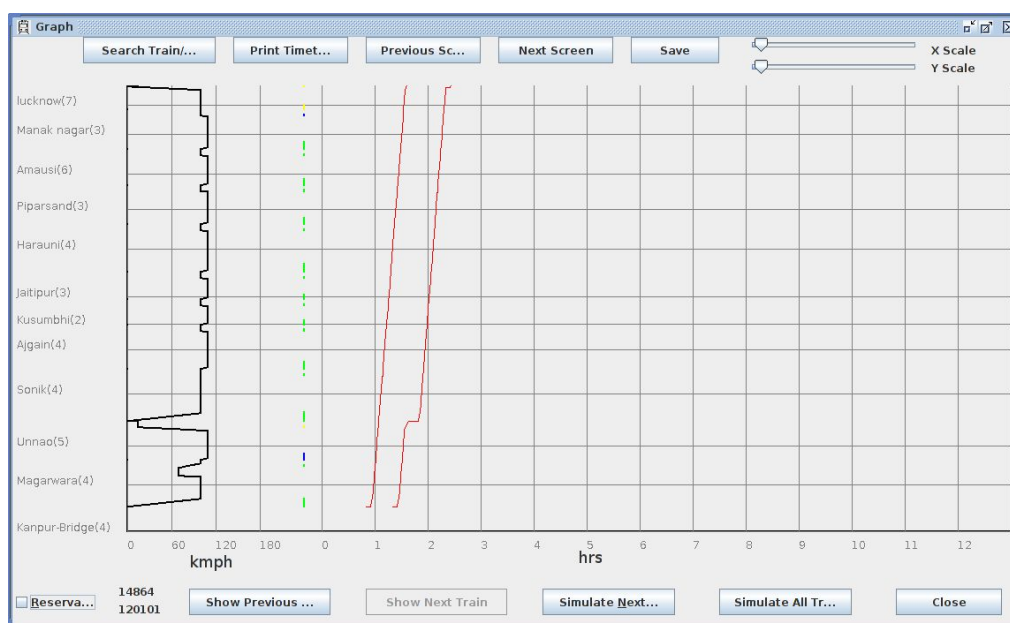
- The plot of every train on Time vs Distance with station name is shown on the simulator.
- **Reservations** check box shows the block occupation for the train. Uncheck it to see the clear traversal curve
- Click on **Simulate Next Train** to simulate next train or click on **Simulate All Trains** to simulate all the trains.
- **Show Next** and **Show Previous** buttons can be used to check the individual distance time curve of the trains.
- Progress bar on the top left corner shows the simulation progress in percentage.
- **X-Scale** and **Y-Scale** helps zoom into the plot.
- **Search Train** button shows the block occupancy details of the train.
- **Print TimeTable** button will save *traversaldetails.txt* file to the directory which records all the occupancy details of the trains and this file is required for the visualization feature which will be detailed in the section *Visualization.*
- **Save** button will save the distance vs time chart as a .png file to the directory.
- **Right click** on the plot to get block occupancy.
- After completion of the simulation a pop up will display average traversal time of freight trains and and ask to save the output timetable file.
- Now to save the generated Time-table, click on **OK**.
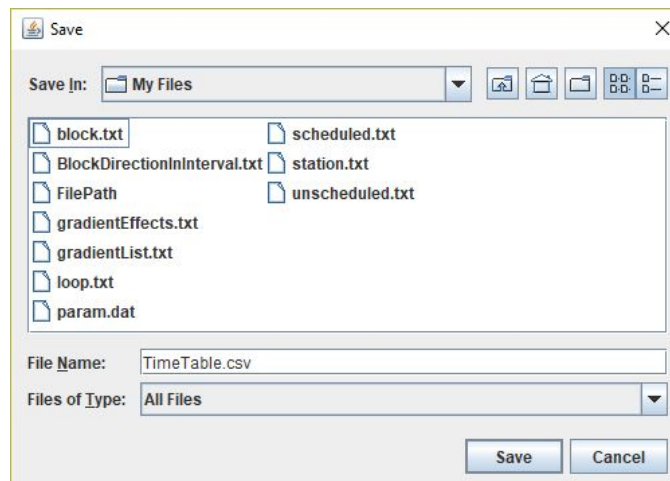
- Select the path and filename and click on **Save**.

8. Comparator

The tool has a comparator that compares between two timetable files. The comparison can be made between two timetable files at a time, for example,
- scheduled versus simulated,
- simulated with respect to one infrastructure versus simulated with respect to a changed infrastructure

The comparison can be analyzed with respect to:
- ❖ freight train statistics (for congestion analysis, since freight trains are run with low priority), or
- ❖ all trains statistics: this can be made train-wise or station-wise. This is elaborated below.

The comparator menu in the tool has two features
- ❖ "**Freight Comparison**" and
- ❖ "**Advanced Comparison**"

8.1 Freight Comparison

The effect of changing infrastructure or timetable can be studied by analyzing the statistics of the freight trains (which face the congestion effects most, since the freight trains are given paths, after all scheduled trains are given paths). Analysis of freight statistics (best/worst traversal times) can help in analyzing congestion.

Procedure to use **Freight Comparison :**

- Go to Comparator menu, Click on "Freight Comparison" option.
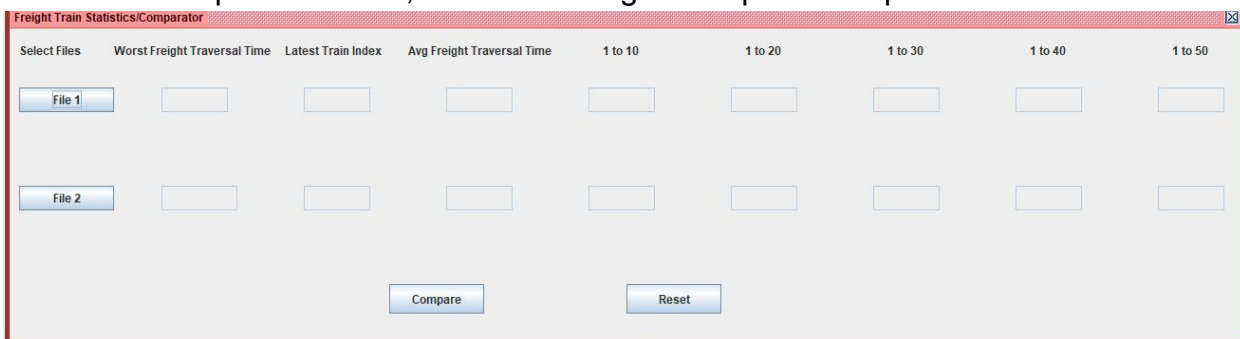
- After that you will see above window.

- Click on " File 1" button to choose first file, and same as for second file.
- After selecting both files have to click on "Compare" button for freight comparison.

8.2 Advanced Comparison

A more elaborate comparison: train-wise or station-wise can be done by reading-in two timetable files and listing all trains/stations where the difference between the two timetables is greater than or equal to a specified time-value. The comparison can be done with respect to either the arrival time or departure time.

"Advanced Comparison" window contains four tabs :
1. Single File Statistics.
2. Train-wise Comparison.
3. Station-wise Comparison.
4. Help.

1.Single File Statistics Tab :- Single File Statistics Tab is used to read schedule.txt and TimeTable.csv format files. This tab will read the file and list stations and trains information with respect to arrival time and departure time according by all search or single search with specified time value.

2.Train-wise Comparison Tab :-This tab will take two files as input and list result according to train that we selected from drop down list. This tab display comparison(train comparison based on train arrival and departure time on each station for that particular train) for train from both selected file with respect to arrival time and departure time.

3.Station-wise Comparison Tab :- This tab will take two files as input and show result according to station that we selected from drop down list.This tab display comparison(station comparison based on all trains arriving and departing at what time from that particular station) for station from both selected file with respect to arrival time and departure time.

4.Help Tab :- This tab has the above information for quicker access.

Procedure to use **Advanced Comparison** :-

Single File Statistics Tab

● Go to Comparator menu, Click on "Advanced Comparison" option
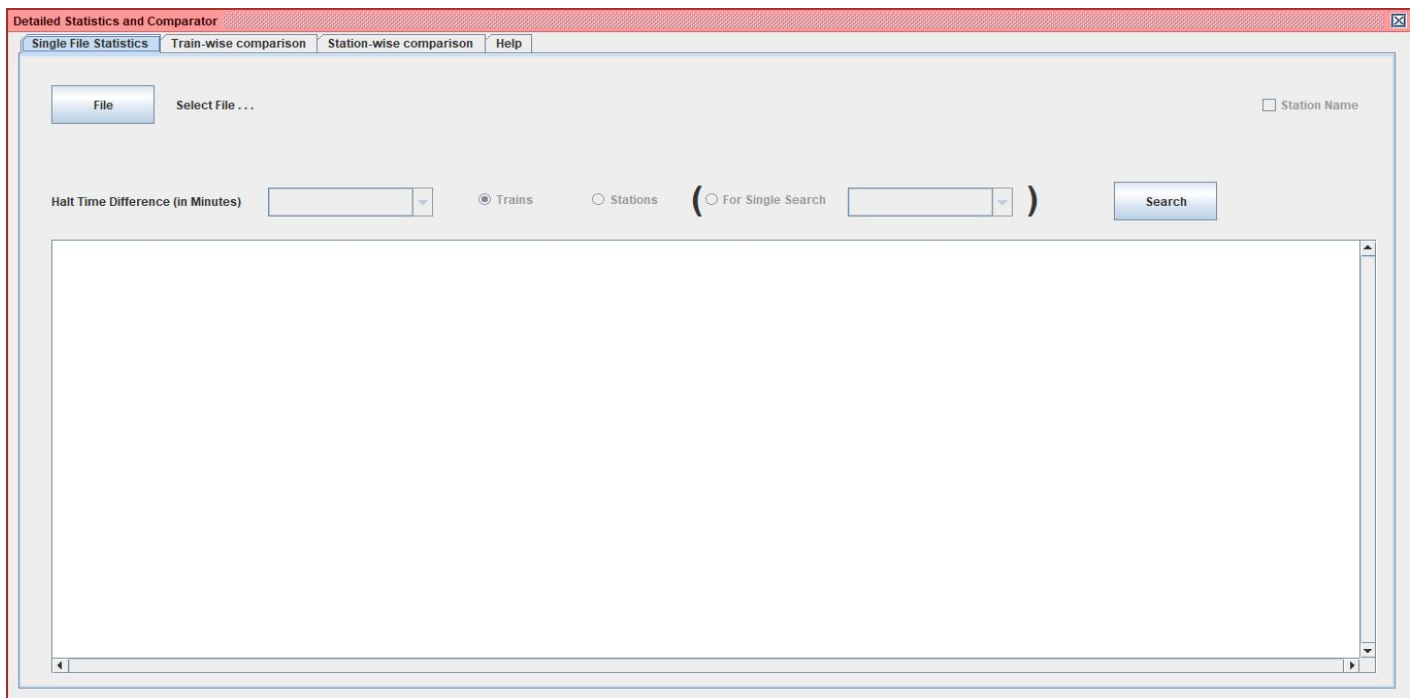
● After that you will see above window i.e. Single File Statistics.
● Click on **File** button to select file from the relevant directory on your computer.
● After selecting file. Other options will get activate.
● Select Halt Time Difference from drop down list which is in minutes (default time is set to 6 min).
● After that either you can select **Trains** option or **Stations** option. (for all search)
  ❖ If you select **Trains** option and click on **search** button, it will show all Trains whose halt time is equal or more than selected **Halt Time Difference** at respective stations**,** or

  ❖ If you select **Stations** option and click on **search** button. It will show all Stations at which respective trains list whose halt timing is equal or more

than Selected **Halt Time Difference** for each train.

- If you want to find result for particular train or station then click on   **(for single search)** option and click on **search** button.
- Top-right corner has a Checkbox (Station Name) option which helps to read station.txt file from same location at which File selected .
- To convert station numbers into station names click on **Station Name** option.

Train-wise Comparison Tab:-

- Go to Comparator menu, Click on "Advanced Comparison" option
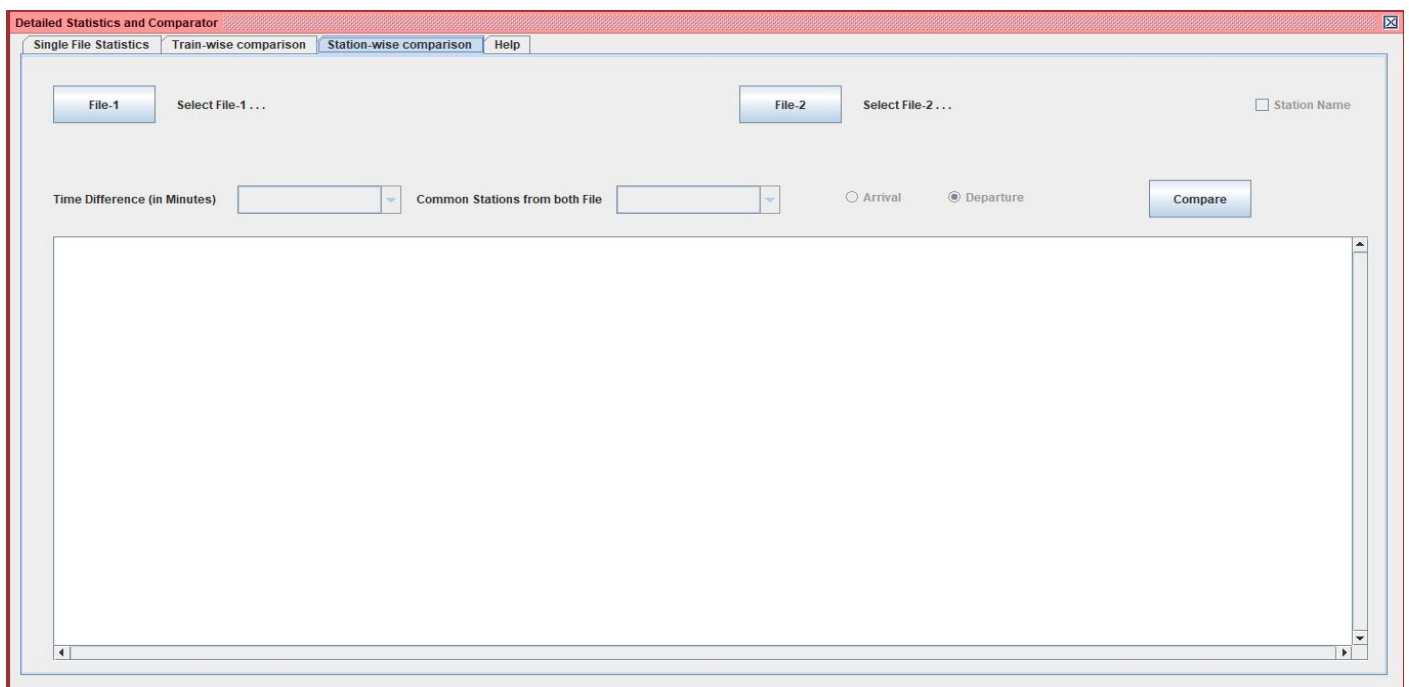- Click on Train-wise Comparison tab.
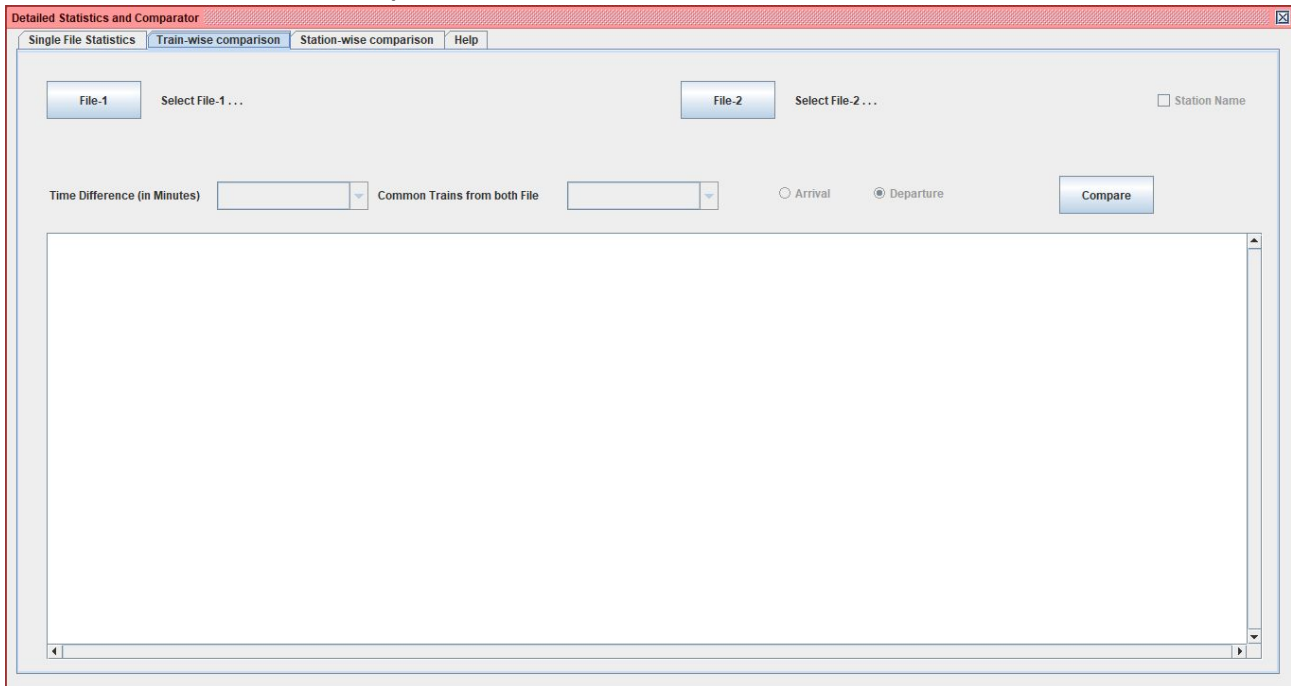
- After that you will see above window i.e Train-wise Comparison.
- This tab will show result after comparing from two files, So it takes two file as input.
- Click on **File-1** button to choose first file and click on **File-2** button to choose second file.
- Select Halt Time Difference from drop down list which is in minutes (default time is set to 6 min).

- Select Train from list to compare.
- Train comparison is possible on the basis of arrival and departure time of train on stations from both files.
- After selecting arrival and departure just click on Compare button. after that result will be shown in below white square box.
- Top-right corner have Checkbox(Station Name) option which helps to read station.txt file from same location at which File-1 selected .
- To convert station numbers into station names click on **Station Name** option.

<u>**Station-wise Comparison Tab:-**</u>

- Go to Comparator menu, Click on "Advanced Comparison" option
- Click on Station-wise Comparison tab.

- After that you will see above window i.e Station-wise Comparison.
- This tab will show result after comparing from two files, So it takes two file as input.
- click on **File-1** button to choose first file and click on **File-2** button to choose second file.
- Select Halt Time Difference from drop down list which is in minutes (default time is set to 6

min).

- Select Station from list to compare.
- Station comparison is possible on the basis of arrival and departure time of train from both files.
- After selecting arrival and departure just click on Compare button. after that result will be shown in below white square box.
- Top-right corner has Checkbox(Station Name) option which helps to read the station.txt file from the same directory from which File-1 was selected .
- To convert station numbers into station names click on **Station Name** option.
  Note- Click on **Station Name** option after selecting File-1 and before selecting File-2

9. Visualization

The Visualization feature allows the user to visualize the infrastructure in a line form. This program is sun using Python and the system needs to install the same with other dependencies like matplotlib. These dependencies will be installed using setup.exe.
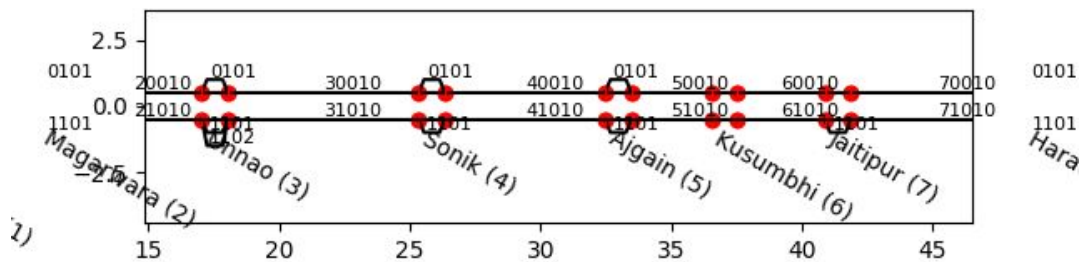
The program displays information such as:
- Station Name
- distance between stations
- Block Numbers
- Loops Numbers
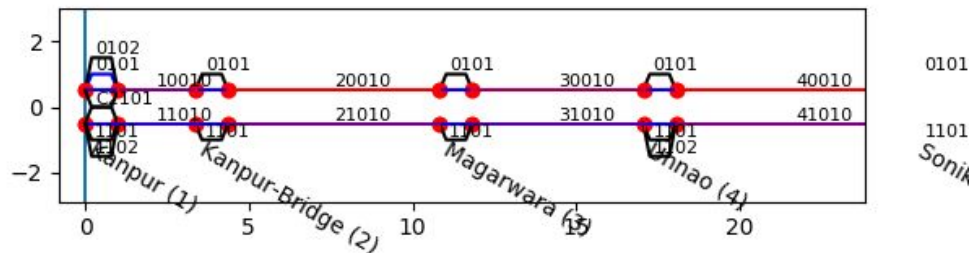
There's an additional feature which allows to display colour coded image of the same where the range of colors display the intensity of usage of the block or the loop. This feature requires a file *traversaldetails.txt,* which is installed in the infrastructure folder if "*Print Timetable*" option was selected while simulation. The program automatically searches for the file from the folder.

Following are the details of the colour codes
- Black denotes the loop/block is unutilized.
- Red denotes the loop/block is heavily utilized during simulation
- Blue denotes the loop/block is lightly utilized during the simulation.
- colours of combination of red and blue show the intermediate usage of various degree.

Appendix A: Loco-Load acceleration calibration

In practice, the tractive effort that a locomotive can provide decreases as speed of the locomotive increases. A typical curve is shown below. This causes the acceleration to decrease as the velocity increases and hence the actual velocity versus time curve is as shown by curve C1 in the figure. However, the java-based section simulator tool assumes a constant-acceleration during the acceleration phase.

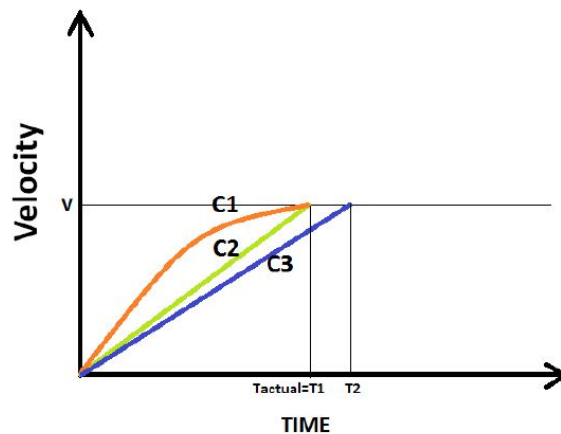Figure 1: Typical tractive effort vs speed (source: AREMA 2012)

Figure 2: Speed vs time curves: C1: actual speed, C2, C3: constant-acceleration based curves

Constant acceleration curves (C2 and C3) are characterized by constant-slope lines, i.e. velocity increases linearly with time. One can consider two values of constant acceleration:

Method 1 (time-matched): here the curve C2 is such that the maximum speed is achieved under constant acceleration, but in the same time as the curve C1.

Method 2 (distance-matched): here the curve C3 is such that the maximum speed is achieved under constant acceleration, but the same amount of distance is covered by then as covered by curve C1.

Of course, this is relevant only for the acceleration phase.

The program takes various loco/load values from the user and uses equivalent acceleration/deceleration values from a look-up table for the simulation. Some of these have been calculated by fitting a quadratic curve by regression and integrating to find the distance covered.

Table 1: Equivalent Acceleration

Engine	Load	Speed V_f (kmph)	Method 1 (**time** matched)			Method 2 (**distance** matched)		
			Eq. acc (m/s²)	**Time** (min, sec)	Distance (km)	Eq. acc (m/s²)	Time (min, s)	**Distance** (km)
WAP 7	18 coaches	110	0.1765	2m 53s	2.64	0.155	3m 17s	3.0
WAP 7	18 coaches	130	0.1574	3m 49s	4.14	0.133	4m 30s	4.88
WAG 9	5400T	60	0.047	5m 52s	2.93	0.044	6m 15s	3.12
WAG 9	4000T	75	0.063	5m 30s	3.46	0.057	6m 3s	3.78

Tractive effort, net wagon load, acceleration effects

Assuming that the tractive effort TE (in tonnes) depends only on the locomotive, and the load that the loco pulls purely scales the TE as: acceleration = g*TE/load. This assumption means that the resistive effort (curve shown above) does not change with load. Further, as explained above, the tool assumes a constant acceleration and hence an "equivalent" tractive effort is calculated for the acceleration phase (to match the time required to reach maximum speed, or the distance covered before reaching the maximum speed). The simulator GUI has the option of specifying the **equivalent** TE (in tonnes), the load (in tonnes) and the maximum speed: this is specified while entering train details and one can edit the file containing these parameters or edit using the "Tweak" option. The equivalent constant acceleration value is all that matters for the simulation of the running of trains and the loco-load parameters are merely to use from a pre-calibrated look-up table.

An xls file having the calculation formulas is kept available at
https://www.ee.iitb.ac.in/%7Ebelur/railways/mixed-traffic-network-scheduler/   together with a brief readme.

Appendix B: Validation for one/multi trains using example

In this appendix, we consider a single train and then two trains and validate the values obtained using the simulator using direct calculation.

B.1 Single train case

Consider a line from station A to station B. The network layout is shown below:-

Station A: start:
0km, end: 1km

Station B: start:
10km, end: 11km
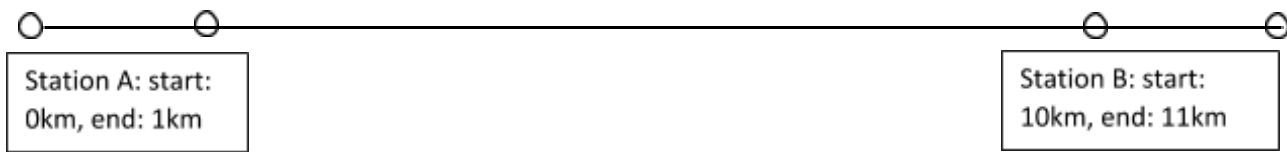
Let us consider a single train with the following parameters:

Acceleration and Deceleration: 0.1 m/s^2

Length: 500 m

Maximum Allowable Speed: 60 km/hr=16.67 m/s

Running the simulator gives the (average) traveling time to be 12.77 minutes. We verify the details below. Consider the traveling time to be the time taken by the train to start from station A and it finally coming to a stop in station B.

If the train starts with zero initial velocity, the time it takes to reach 16.67 m/s with the given acceleration is given by

$$v - u = at \quad v : final\ velocity,\ u : initial\ velocity,\ a : acceleration$$

$$t = \frac{16.67}{0.1} = 166.67\ s$$

So in this time the train must have gone some distance, which we calculate by

$$v^2 - u^2 = 2aS \text{ where } S \text{ is the required distance}$$

$$S = \frac{v^2}{2a} = \frac{16.67^2}{2*0.1} = 1389.44\ m$$

So, when the train reaches its full speed it has already covered a distance of (1389.44+500) m from the start of station A. It has to decelerate at an appropriate time for coming to stop at station B. It has to stop at 10,500m. However it has

75

a stopping distance of 1389.44 m. Hence it must start decelerating from (10500-1389.44) m. So we can say that it travels at a speed of 16.67 m/s from 1889.44 m to 9110.56 m. The time taken by the train to cover this distance is

$$\frac{9110.56-1889.44}{16.67} = 433.18 \ s$$

So, the total time that it takes traveling from A to B is

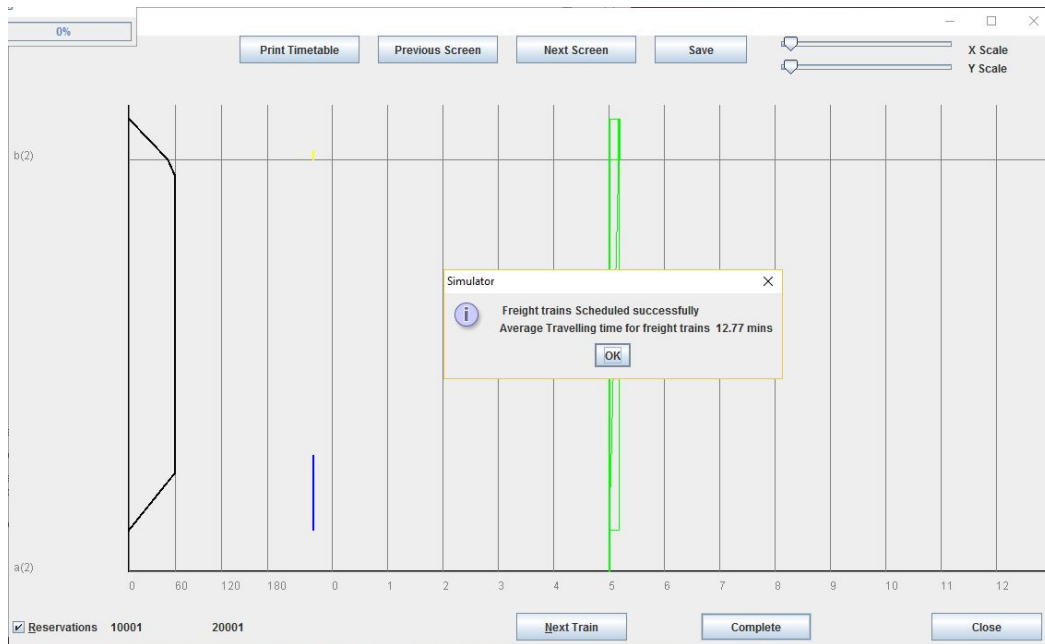$$\frac{433.18+166.67+166.67}{60} = 12.77 \ mins$$

Fig 1: Screen shot of the simulator after running the above test case

B.2 Two train case: multi-train example

Let us now consider a case with two trains-one scheduled and one unscheduled in our system. Consider that the scheduled train receives a higher priority. We consider three stations with each station length being 1km. Both the trains have a length of 500m. The scheduled train has the following specifications:-
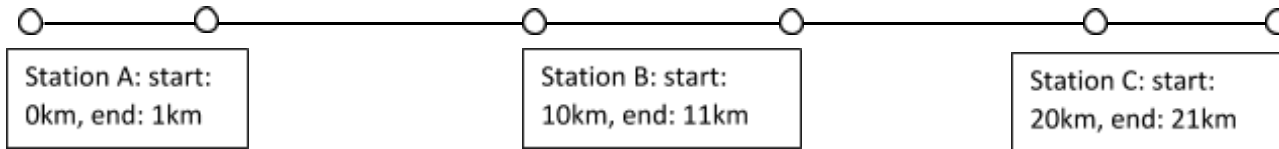
Acceleration and Deceleration: 0.5m/s²

Maximum Speed: 130 km/hr=81.25 m/s

The freight train has the following specifications:

Acceleration and Deceleration: 0.1m/s²

Maximum Speed: 60 km/hr=16.67 m/s

The timetable generated by the Simulator is shown in simplified terms:

TRAIN TYPE	STATION A :ARR	STATION A: DEP	STATION B: ARR	STATION B:DEP	STATION C:ARR	STATION C:DEP
Scheduled Train	05:13	05:14	05:27:14	05:28:09	05:37:49	05:44:17
Unscheduled Train	04:50	04:50	05:04:25	05:39:32	05:54:16	05:56:44

Now we analyze the generated time-table for any violation of safety norms.

Station B is a station where there are no stops for the scheduled train. For consistency in calculations both arrival and departure of trains have been considered with respect to the tail of the train.

The unscheduled train in this case goes from station A to station B thereby reserving the block between these two stations up to 05:04:25. However the scheduled train arrives at station A only at 05:13 and the unscheduled train is thus allowed to move to station B. Now the simulator calculates that the unscheduled train would make the scheduled train wait at station B if the unscheduled train is allowed to leave station B. This is because the block between Station B and C would then be reserved by the unscheduled train. So, it gives a higher priority to the scheduled train and makes the unscheduled train wait at station B, where it is overtaken by the scheduled train. Thus overtaking takes place at station B. This is shown circled in red in the screenshot of the simulator output that follows. Thus safety norms are adhered to.

All the traversal times can be checked to be in accordance with the laws of motion as in the previous example. Thus the simulator output for the multiple train case is verified.
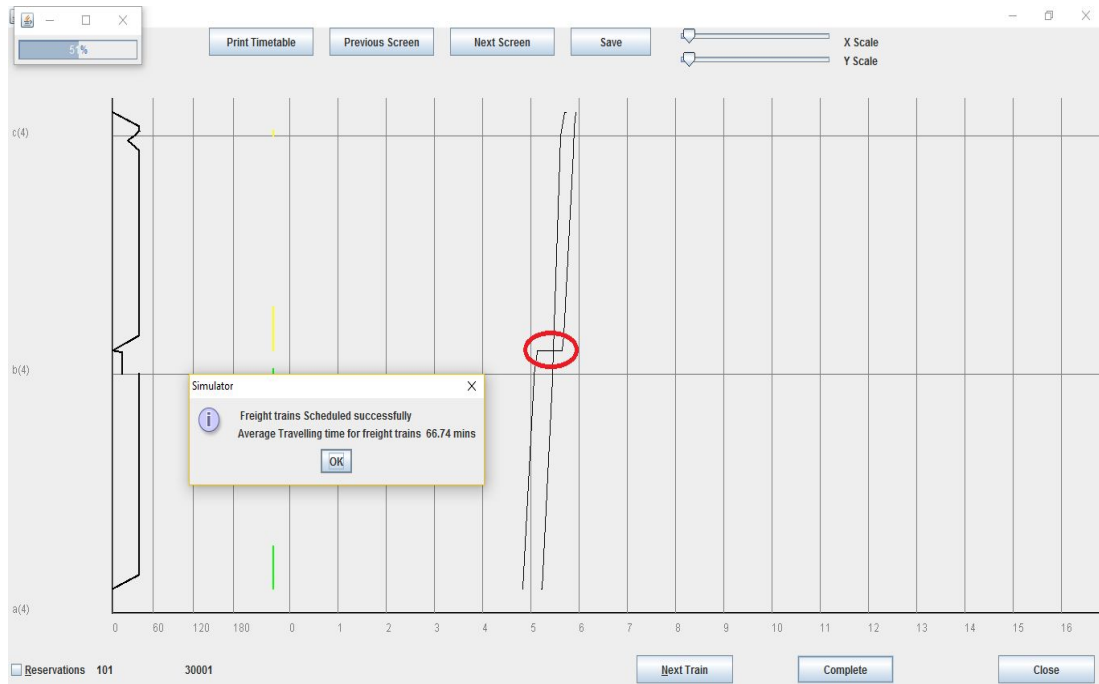
Fig 2: Screenshot of the simulator after running the second test case

Cases where there are more than two trains and they have different priorities can also be run using the simulator. The working of simulator can be verified in a similar fashion as has been shown in this document.

For updates about the Simulator and code documentation visit: https://www.ee.iitb.ac.in/%7Ebelur/railways/

Appendix C: Opening/editing txt files in Spreadsheet (like csv files)

While the various outputs are generated in an xls file format, the inputs that are needed by the simulator (and are hence created by the GUI for infrastructure/train adding/editing) are created in the txt file format. This section elaborates about how the txt files can be viewed and edited directly using a spreadsheet tool, for example, MS Excel.

Note: please backup the files with suggestive names and dates before opening/editing.

C.1 Brief background about csv files

The txt files are plain text files with the first line as the header and the following lines as the data. For any csv file, opening in a spreadsheet requires explicitly specifying the delimiter. Originally, the file extension "csv" stood for "comma separated values", but now this extension is used more widely for many other delimiters, for example: tab, space, semicolon, colon, comma. Amongst the input txt files used by tool, the current mixed-traffic simulator, space has been used as a delimiter.

In order to open the txt file using a spreadsheet tool like MS Excel, we suggest the following two methods. (If there are other easier methods, please write by email to the contact-persons listed in the beginning of this manual.)

The procedure below is applicable for all txt-files, and more details about the headers is explained separately. See Section 6.2.3 for example about block.txt and other txt files.

C.1.1 Opening txt after first starting MS-Excel

Start MS-Excel first, and then choose "Open file" and search for "All files" in the directory instead of just "All Excel files". Look for the relevant txt file and choose the delimiter as "Space". This would open the txt file with the relevant headers for each column.

C.1.2 Opening txt file using MS Excel by changing the file-type

MS-Excel would not open a txt file assuming different columns (using a suitable delimiter), and hence a change of file-type is needed. However, the simulator looks for the relevant txt file when running: like block.txt, loop.txt, etc. Hence if this method is to be used, one has to change the file-type into csv, edit this csv file using MS Excel using delimiter as "Space" and then change the file-type back to txt.

Contributors

The simulator is an outcome of many years of effort and various faculty, students, project employees at IIT Bombay have contributed at different periods of time (between roughly 2012 and 2018).

Contributors to the mixed rail traffic network simulator
Devendra Shelar, Peeyush, Soumya Dutta, P. Sudarshan, Sneha Mistry, Sushil Bobade, Abhishek Singh, Anuj Sahu and R. Vidyadhar played a key role in the development of the code. Recent additions of features were driven by Sidhartha Kumar, Akshay Bajpai, Kumar Appaiah, Samruddhi Pai, Soham Naha, Satwik R.V., Mitul Tyagi, Chandrashekhar Bobade. The simulator was used for various academic and Indian railway projects by many other students at IIT Bombay: this helped modify and improve various features. Aayush Aggarwal provided inputs in the loco-load acceleration calibration.

Help was also taken from ITSPE, Hyderabad.

Development of an earlier version of this tool was partially supported by RDSO, Lucknow.

Features of this tool, together with development advice, are thanks to Madhu Kumar Reddy, Mandeep Bhatia, R. Gopalakrishnan, Merajus Salekin, Amit Kumar Jain.

The initial development of the simulator was closely supervised by Abhiram Ranade and Narayan Rangaraj, and more recently by Shiva Gopalakrishnan, Kumar Appaiah, Madhu Belur and Narayan Rangaraj. Bug-reporting and queries about the use can be pursued with them. Details can be found at

http://www.ee.iitb.ac.in/%7Ebelur/railways