OPTIMIZATION OF TRAIN DISPATCHING SUPPORT SYSTEM: A CASE STUDY

[1]Nurye Hassen, [2]Deenadayalan, [3]Rajagopal,

[1, 2, 3] Department of Electrical and Computer Engineering, Dire Dawa Institute of Technology, Dire Dawa University, Ethiopia

Abstract: One of the significant challenges in the daily operation of train dispatching is making the right decision upon unplanned conflict occurrence. The resolution process by itself will introduce an additional delay on the railway network unless well resolved. So, it is a decisive and challenging issue for train dispatchers and railway operation planners to decide which of the trains to stop or to pass from the trains involved in the conflict to bring minimum propagated delay. Such an operation with effective conflict resolution requires an intelligent decision support system that considers minimization of future dwelling time. This paper developed a decision support system that provides an intelligent decision to the train dispatcher by detecting a conflict on a rail network with an optimal resolution of the conflict and cost. The aim of this case study is addressed the minimization of overall delay due to the conflict resolution in addition to detection and resolution. A mixed integer linear programming approach has been implemented to find optimal combinations of arrival and departure events that bring minimum propagated delay. The model has been applied to Ethio-Djibouti Railway enterprise railroad, which consists of eight stations. The program was able to reach an optimal solution with minimum cost when compared to the manual (heuristic) approach especially for an increased number of trains and stations. The model was also tested based on various hypothetical assumptions and showed that it is a powerful tool to be used for train dispatchers for ensuring operational optimality and safety of the railway line.

***IndexTerms* - Conflict Detection and Resolution, Decision Support System, Mixed integer linear programming, Re-scheduling, Train Dispatching.**

I. INTRODUCTION

Due to pre-planning of its operations, railroad systems are considered to be more reliable, cost-effective, and safer than other means of transportation [1]. Conversely, in real time rail operations, the pre-planned schedule (timetable) for trains usually exhibits variations compared to the actual records on the route[2]. The cause of this deviation is due to technical failures like a signal problem, catenary problem, engine breakdown or other problems like staff shortage and weather condition[3]. Such unplanned circumstances result in increased dwelling times, running times, arrival and departing events [4].Because of train interactions, these variations will also be propagated to other trains and brings interruption of the whole rail network specifically to the scheduled timetable[5]. This interruption inevitably causes infeasibility of the previous schedule. Consequently, it needs to be modified with the correct timetable comprising of new arriving and departing events from/to all stations with all conflicts resolved and normal traffic movement restored. This process demands effective solutions within a short duration of time and is called train dispatching or conflict resolution[1].A feasible railroad timetable should specify the departure and arrival time of all trains to each location of its route so that the line capacity, safety, and other operational constraints are satisfied[6][7]. In the last few decades, the competition among transport service providers has created awareness of the need for quality service. Under such pressure for improvement, computer tools have been developed to help Dispatchers and planners to do their work more efficiently and quickly [5]. Those tools have also been able to assist train dispatchers in decision making process. However, search for a better and optimal solution is still drawing the attention of researchers [2][3][6].

In Ethiopia, the former railroad network (Ethio Djibouti railway Enterprise) which still runs between Dire Dawa and Djibouti (Daouenle), implements a manual approach to generate train timetable by drawing trains on the time-distance diagram, whereby the train diagram is manually adjusted so that all associated conflicts resolved. Such a process can be time-consuming furthermore it doesn't ensure operational optimality [4][7]. In this case study, developed a real-time and intelligent dispatching support system that detects a conflict on single track railroad network and provides optimal resolution. The foremost objective of this model was to minimize the overall delay arisen from the stopping of any train while solving those conflicts. The user-friendly characteristic of the GUI makes it easy to be used by any level of skill.

Railways are built with a different number of tracks, depending on the traffic characteristics of the rail network. The most common railway tracks are single track and double track. Single line tracks can serve only one train at a time while double line tracks can serve up to pairs of a train at a time travelling in an opposite or same direction[8]. Specifically, on single line rail track trains are allowed to overtake(pass) and cross(meet) each other only at particular locations such as sidings and meet points sometimes referred as stations that are pre-established along the track line[9]. From an operation point of view, double line tracks operate very much simpler than that of single line tracks. Trains moving in a same direction use one single track and trains in opposing direction use the other. i.e., opposing trains do not exist if every train keeps its route as per the dispatching plan. To conclude, in single line track operation, movement of opposing trains and fast train following slow train demands adequate dispatching tool to avoid colliding of a train with others within the network[1].

II. SATEMENT OF THE PROBLEM

On single track railroads, where trains moving in both directions share the same track, overtake and cross operation has to be done only at stations or sidings. Meeting of trains outside of this region will eventually lead to rout conflict and cause train collision. Additionally, for technical or external problems trains continue to exhibit deviation from the pre-developed schedule. The disrupted schedule cause unstable railway operation and an increasing propagating delay. Thus, it is crucial to detect and resolve such abnormal conditions precisely to limit the damage likely to happen in the traffic system. A train timetable is well planned to strictly meet rail traffic constraints such as track capacity and operational constraints. Train dispatcher uses this detailed timetable for dispatching trains within the system. The train dispatcher may encounter problems due to the existence of unforeseen circumstances in the journey of the train. A train breakdown, for example, will cause unavailability of the occupied track for other scheduled trains and therefore, cause conflicts within the network which needs resolution on time and the timetable should be updated. At this stage, the train dispatcher must reschedule involved trains completely to eliminate emerged conflicts and restore normal traffic movement.

Such a process requires adequate and intelligent dispatching support system that can generate a conflict-free schedule in a reasonable amount of time by considering both operational and safety constraints. The system also aims to minimize the time deviation of trains from the previously planned schedule. Driven by such a complex problem, in this case study real-time and intelligent dispatching support system is developed that detects a conflict on a rail network and provides optimal resolution. Beside detection and resolution of trains, the target of this model further extends to minimize the overall delay resulted from stopping of trains while resolving those conflicts. The general objective of this case study is to optimize performance of train dispatching support system for a single-track system

III. METHODOLOGY

The methods followed in this case study to achieve the objectives of the research include the following core steps.

- Literature review, a study on optimization problems and their respective appropriate methods as well as depth understanding of the research problem.
- Necessary data and information required as input for the research collected.
- The problem formulated as an optimization problem with proper objective function and constraints.
- Appropriate algorithm and method selected for solving the problem, which is mixed integer linear programming with branch and bound technique.
- Selection of suitable software environment, MATLAB R2015a, was chosen for its advantage of containing built-in methods for solving optimization problems, for writing the program and implementation of the model.
- Performance evaluation carried out, mainly testing the developed application and performing result analysis. The flow chart depicting the methodology is shown below in figure 1.

IV. MODEL DEVELOPMENT

An optimal solution approach toward decision-making problems involves three core steps. Constructing an appropriate mathematical model, selecting an efficient algorithm for solving that model, and finally implementing the solution obtained by the algorithm. These core steps are discussed under one by one. The overall system architect is depicted in figure 2 below, showing the type of information exchanged with the dispatcher and the incorporation of the decision support system with the whole dispatching process.

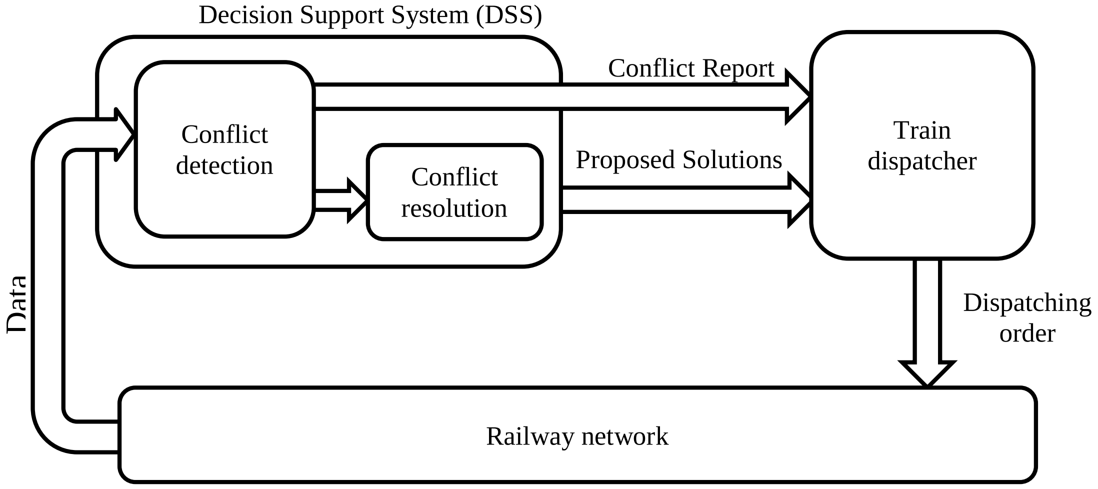

Figure 2: System Architecture

This model is a very useful tool for dispatchers to enable them to analyze the consequence of their decision before implementing and provides all other feasible solutions. Thus, train dispatchers will have more option to consider and to select the one which best fits with the rules and regulation predefined by the railway authority.

4.1Model Description

Before modeling the system, it is necessary to define and discuss the problem environment and assumptions. This case study considers a single-track railway that serves trains traveling in both directions on a single rail line. In normal operations, trains are allowed to meet or pass each other only at stations or sidings otherwise conflict is said to have occurred. Thus, a meet point is not just station but also sidings or any other point on the track where two trains meet or pass each other. This assumption makes the traffic control and management, complex, realistic, and a very crucial issue for safety reasons. Directions are defined as westbound for trains going from right to left and eastbound for the reverse movement. Numbering for meet point and track segment is made in eastbound directions. As shown figure 3, the first and last meet points are either interface to a double track or to a denser rail network. Either way, safety time interval between arrival and departure at these stations remains unchecked. The safety technology considered is fixed block i.e. trains can follow each other on track segment keeping minimum safety headway. Three types of train priorities are chosen for this model namely, fast passengers train, slow passengers train, and a freight train.

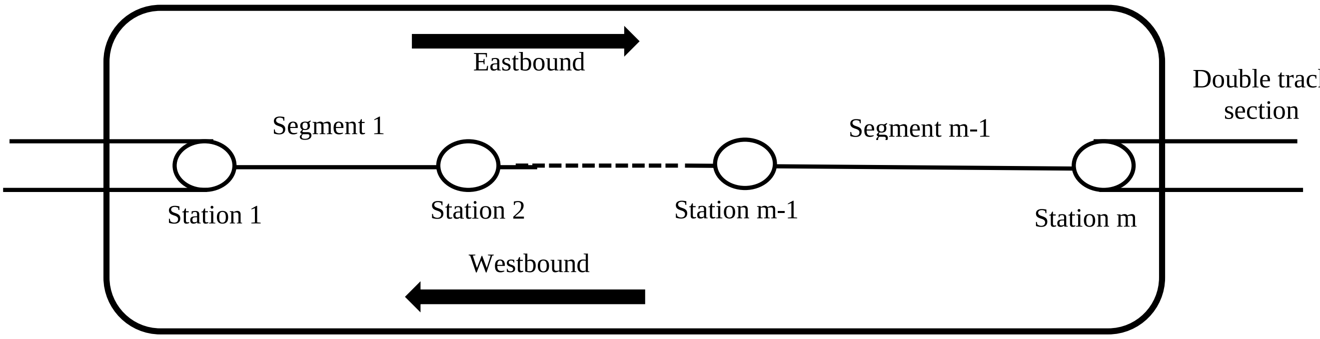

Figure 3: Problem Description

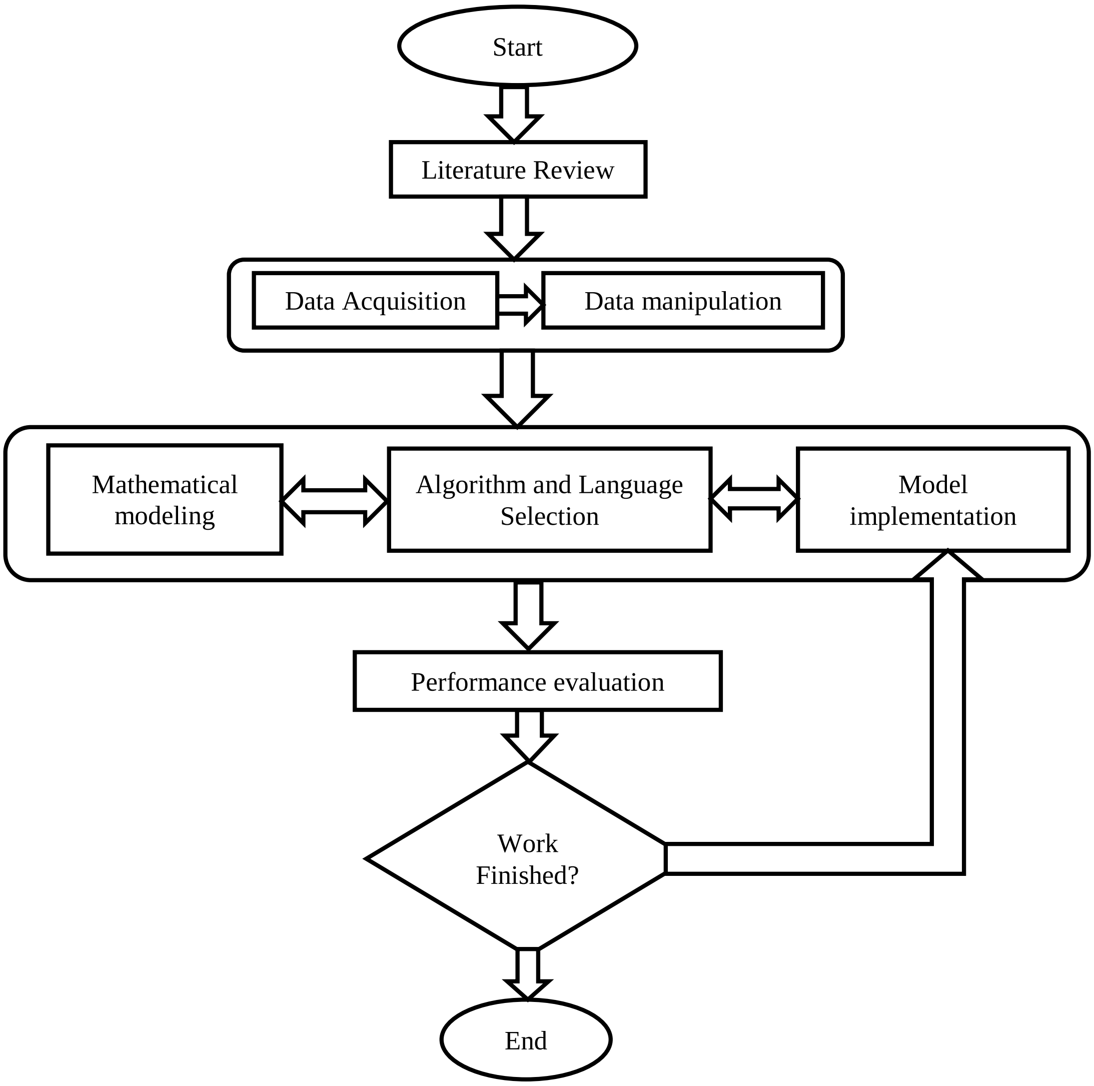

Figure 1: workflow chart

4.2 Selection of Appropriate Algorithm

Optimization techniques are used to find a set of design parameters, $x = \{x_1, x_2 \ldots xn\}$, that can in some way be defined as optimal. In the context of this thesis, the design parameters are set of all events that give an optimal schedule. It consists of all arrivals, departures, dwell times, running times and delay times. The objective is to minimize sum of delays that arise from all trains measured on their destination subject to constraints listed earlier. These constraints can be categorized under equality constraint, inequality constraints, and parameter bounds, $xl,\ xu$(lower and upper bounds). An efficient and accurate solution to this problem depends not only on the size of the problem in terms of the number of constraints and design variables but also on characteristics of the objective function and constraints. When both the objective function and the constraints are linear functions of the design variable, the problem is known as a Linear Programming (LP) problem. If some of the design parameters are needed to be integers, the problem will be a Mixed integer linear programming (MILP). This is exactly the characteristics of our problem, i.e. the objective function as well as all constraints is linear and for practical reasons, all elements in the train schedule should be integers. Apparently, the mathematical formulation presented earlier is called a mixed integer linear program (MILP**)**. MATLAB has introduced the solver called "*intlinprog*" for solving mixed integer linear programs in its optimization toolbox since 2014. The MILP has become a much-known mathematical language in the literature since the 1960th. With this in mind, there are also many other optimization software's for solving MILP problems besides, the capability of the intlinprog solver to deal with a large number of variables and availability of easily customizable and vast options suits to be a choice for this problem.

V. ALGORITHM IMPLEMENTATION

In this study, MATLAB is used to solve the Optimization problem and to develop the overall interactive train dispatching support system. Hereunder some of the reasons are discussed. MATLAB is software and programming language specially oriented for matrix manipulations thus it performs faster numerical calculations faster than other programming languages. In the rescheduling process, the most tasks are to manipulate timetables (other forms of matrices). This present one significant advantage of using this software. MATLAB optimization toolbox contains easily customizable options for users and built-in heuristic procedures that may simplify the solution procedure and decrease solution time in some cases. Its user friendless and capability to deal with a large number of variables made it the choice for this problem. Being train diagram creation and graphical user interface development important feature of this work, the GUI inside MATLAB and its libraries will ease and make the application more interactive to train dispatchers. The overall dispatching support system contains two major sections, conflict detection, and conflict resolution. The conflict detection section of the program first takes all the necessary inputs mainly the initial train schedule. Using the detection algorithm, it will search whether there is any potential conflict or not.

5.1. Conflict Detection

The detection mechanisms for any conflict or in another term detection of any violations of the traffic rules. To recall four conflicts were considered in this thesis, crossing (trains meeting) conflict, overtake (trains passing) conflict, capacity conflict and safety interval at station conflict. To detect crossing and overtake take conflicts, there are two critical actions taken for reducing the complexity of the problem for saving time and computer memory. i.e., a blind scan was not performed to search a conflict upon all trains in the schedule. Instead, adequate and intelligent search for potential candidates of conflict has carried as stated below. The first approach toward the complexity reduction is to make the algorithm to check whether the potential trains are in the same direction or not. If they are toward the same direction only potential overtake conflict will be carried. Otherwise, for trains moving in different directions a potential crossing conflict alone will be examined. The second approach: it is unnecessary to check each constraint to detect conflicts between all trains in the schedule. Preferably, it is enough to sort trains in entering orders into segments and checking the constraints only between consecutive trains[10].Using this approach, the original problem which was a combinatorial problem (i.e., for n trains there were C_2^n comparisons) it can be reduced into n-1 comparisons only. The procedure to detect either of the two mentioned conflicts is first to sort the timetable according to entering times in ascendant order into segments (track section between two meet points) and secondly to check the cross and overtake constraints between consecutive trains figure4, pictures the segments, meet points, start and finish times.

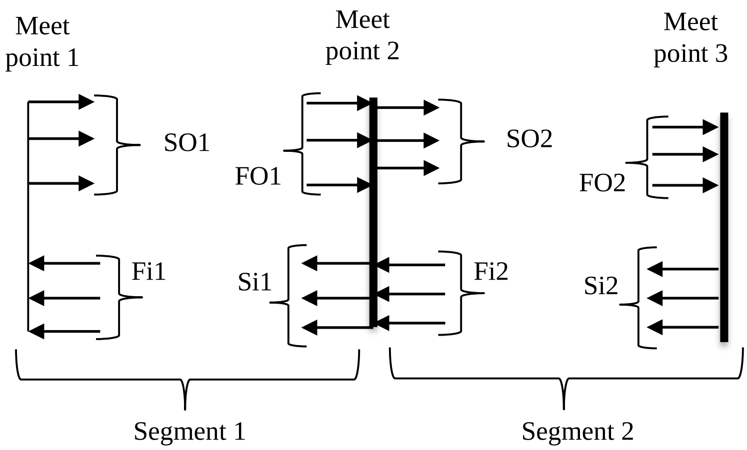

Figure 4: Start and Finish point of trains to meet points

SO1, FO1	= Start and finish time of eastbound trains from/at segment one respectively
Si1, Fi1	= Start and finish time of westbound trains from/at segment one respectively
SO2, FO2	= Start and finish time of eastbound trains from/at segment two respectively
Si2, Fi2	= Start and finish time of westbound trains from/at segment two respectively

There are three meet points. Thus, there exist two track segments because the segment track is always one less than the total meet points as illustrated in figure 4. The example was written in the form of arrival and departure time from meeting points. The detection algorithm proceeds by writing the table in a track segment arrangements are shown in table 1 and sort according to entering orders to the segments as displayed in table 2

Table1: Segment start and finish time

	Train No.	Segment 1		Segment 2	
		Start	**Finish**	**Start**	**Finish**
Eastbound trains					
Train 1	424	**10**	**15**	**15**	**22**
Train2	10	**15**	**30**	**50**	**65**
Train3	12	**30**	**36**	**42**	**48**
Westbound trains					
Train4	17	**37**	**42**	**32**	**36**
Train5	105	**58**	**65**	**46**	**52**
Train6	15	**50**	**70**	**5**	**10**

Table 2: Train sorted in order of entrance time to segments

Segment one			**Segment two**		
	Train no.	Entering Time		Train no.	Entering Time
Train 1	424	10	Train 2	10	50
Train 2	10	15	Train 1	424	54
Train 3	12	30	Train 3	12	62
Train 4	17	149	Train 5	105	66
Train 6	15	190	Train 6	15	105
Train 5	105	300	Train 4	17	136

A track segment is part of the railway line that is bounded by two meet points or stations. Therefore, both eastbound and westbound trains start their journey from either side of the segment. For single track railways entering times to segment matters a lot. Opposing trains are not authorized to enter the same track segment at a time. Additionally, trains running in the same direction are not entitled to enter the same track segment without minimum safety headway. Track segment one is bounded by meet point one from the left and meet point two from the right. Therefore, only eastbound trains (trains 1-3) from the left and westbound trains (trains 4-6) from the right start their journey at this specific track segment. For track segment two, it is bounded by meet point two from the left and meet point three from the right. Here under table 2 shows trains sorted according to entering time to each track segment.

VI. RESULTS AND DISCUSSION

To evaluate the efficiency of the model, in optimally solving conflicts, sample problems were tested on sample single-track rail network and optimal solutions were found. The tool developed under this thesis is not supposed to replace the train dispatcher but help them in making an optimal decision. As shown in figure 5, the proposed solution is generated and given to the dispatcher then the dispatchers have all the necessary information including the optimal solution, rules and regulation and other situation on the ground to be able to solve real-time dispatching problems. The sample problem contains a different number of stations and trains. For movement of trains from source to destination, the sample railway network considers only a single track which is more complicated and difficult for solving dispatching problems as well as a widely available type of track formation in regional networks than a double track .During the resolution, the headway time considered between trains is 20 minutes, while the interval between departure and arrivals 5 minutes. Sample problems are given in figure 9 and their respective optimal solution is discussed here under.

6.1. Problem 1: Input Schedule Sheet

Table 3: Input Schedule for problem 1

Trains No.	**Dire Dawa**		**Chenele**		**Harraouah**	
	Arrival	**Departure**	**Arrival**	**Departure**	**Arrival**	**Departure**
Eastbound trains						
10	**0**	**10**	**15**	**15**	**22**	**1000**
11	**0**	**15**	**30**	**50**	**65**	**1000**
12	**0**	**30**	**36**	**42**	**48**	**1000**
Westbound trains						
406	**42**	**1000**	**36**	**37**	**0**	**32**
408	**65**	**1000**	**52**	**58**	**0**	**46**
422	**70**	**1000**	**10**	**50**	**0**	**5**

The developed interactive model has an option to generate a train diagram depicted in readily distinct form by dispatchers more specifically those of cross and overtake conflicts as shown in figure 5. Other conflict types are not easily traceable by the dispatcher from the train diagram. Thus, the dispatcher needs to get them from the conflict report section of the dispatching support system. After properly loading the input schedule, the preview option of the graphical user interface (GUI) generates the following interactive window shown in figure 5.

The GUI is comprised of four sub-sections:

1. This area is called the input section. The user is allowed to choose the input file when pressing the button and is a place where the dispatcher specifies the weight of each train type.

2. This section is a place where the dispatcher can preview the input or look at the optimal resolution by selecting the desired radio button and pressing start. There are also few other options here like reset and drawing the diagram for a specific train.
3. Here a preview of the input or the plan of the optimal resolution can be seen. Result panel, the dispatcher gets the conflict report and the timetable in a separate window

Initial Train diagram (Schedule):

The dispatcher has an option to see the input data from excel both on the GUI and out of the GUI window. This enables train dispatchers to verify if the input data is correct .figure 6 shows the train diagram separate from the GUI

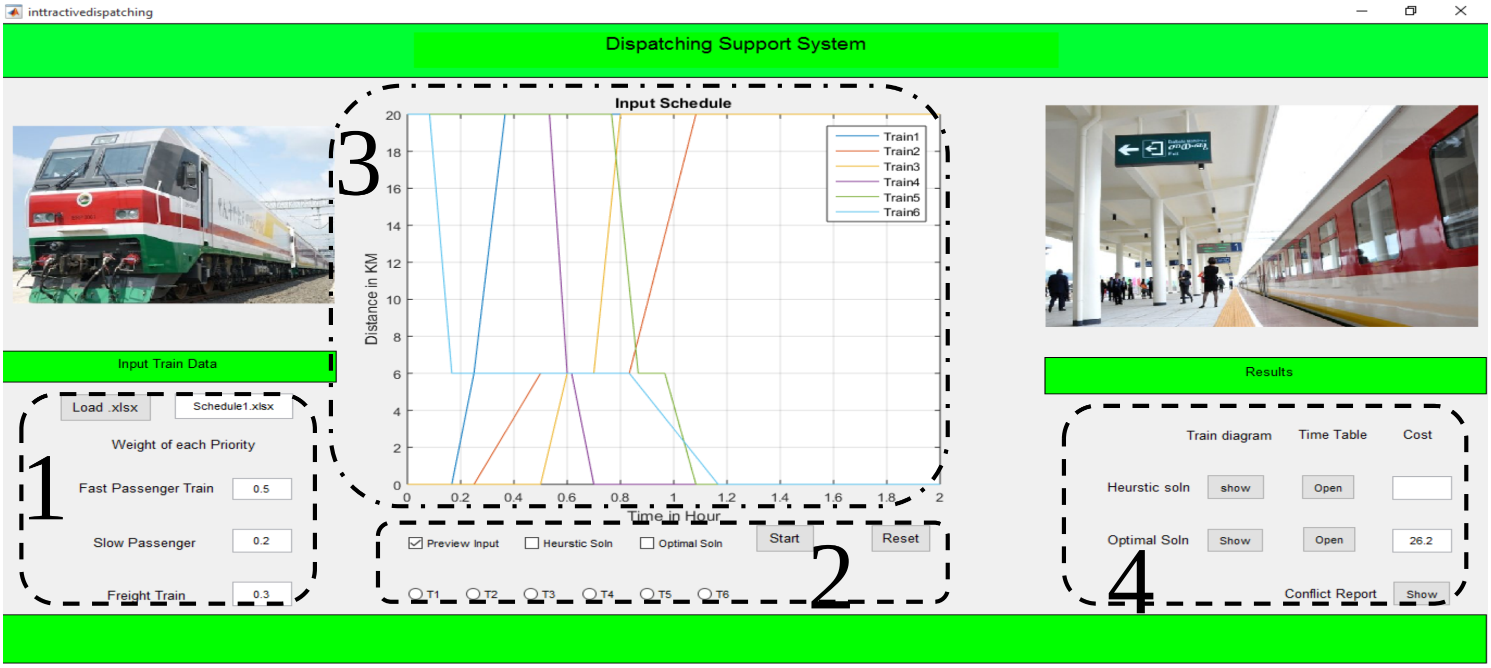

Figure5: Dispatching support system preview

For this train diagram, there are three stations (meet points) located at 0, 6, and 20 kilometers. If two trains meet other than this station, a conflict has had occurred. As clearly seen from the train diagram plot figure 6, for the initial schedule, there are two meet conflicts and one pass conflict between trains mentioned in the conflict report below. The conflict report contains other forms of conflict as well. The conflict report as presented below displays both initial conflicts and those conflicts raised during the resolution process by itself. Finally, the program automatically saves this report into the folder provided as a text file

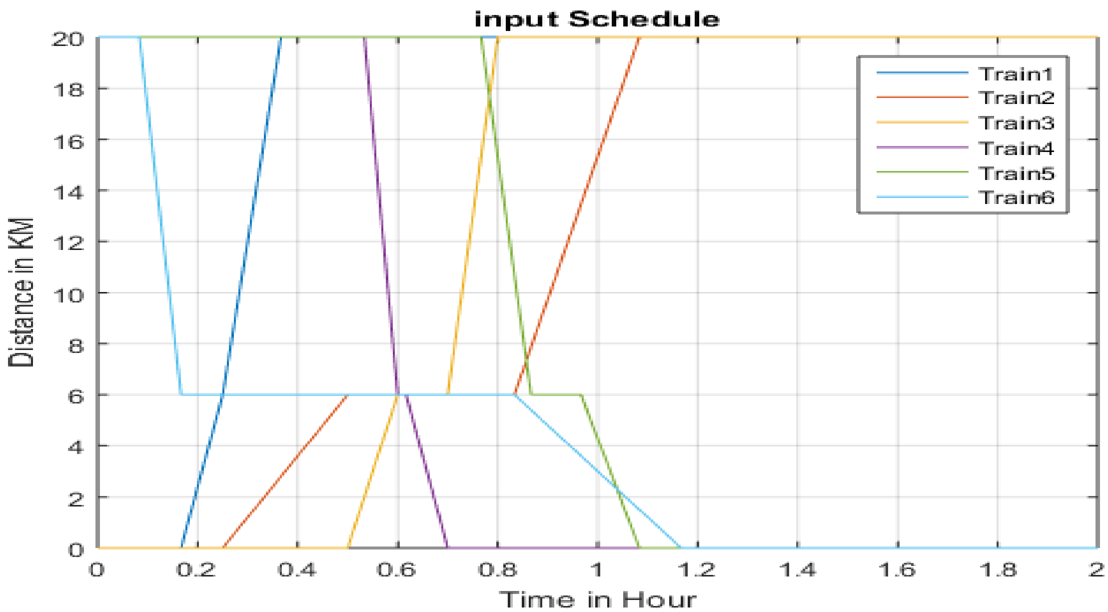

Figure 6: Problem 1 initial train diagram

Problem 1 Optimal solution:

Selecting the radio button for the option of optimal resolution generates the following train diagram in the interactive window with its corresponding cost. In figure7, there are six trains running through three stations. Three of the trains are traveling eastbound while the others are traveling in westbound. The train diagram plot for the optimal solution as clearly seen from figure7, trains are only meeting on stations (0, 6, and 9 kilometers). Thus, there are no cross, overtake or other forms of conflicts. This shows that the dispatching support system has solved the conflict neatly and optimally.

From the generated input train diagram some of the conflicts that can be easily detected by looking at the picture are circled for an explanation. However, station conflicts and other conflicts due to safety constraints cannot be seen directly from the diagram. The conflict report demonstrates all conflicts in detail. The train diagram shows that there is one cross(meeting) conflict on track segment two between trains two and eight, on track segment four cross conflict between trains three and ten, two cross conflicts on track segment five between trains one and eight and between trains three and eleven. It should be noted that these are not the only conflicts in the timetable.

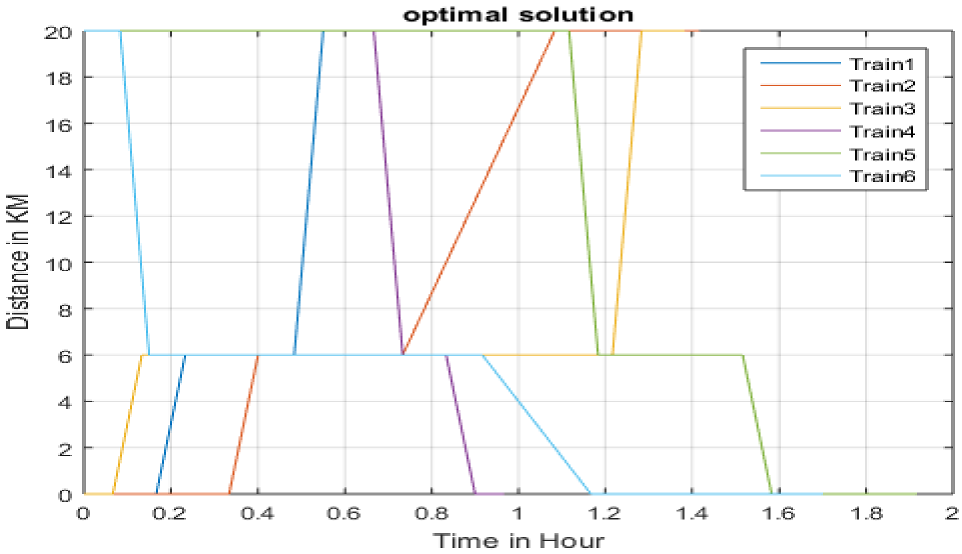

Figure 7: Problem 1 optimal solution

Problem 2: Initial Schedule Sheet

Table 4: Input Schedule Sheet for problem 2

M1		M2		M3		M4		M5		M6	
ARRIV	DEP	ARRIV	DEP	ARRIV	DEP	ARRIV	DEP	ARRIV	DEP	ARRIV	DEP
0	5	25	30	61	65	80	82	130	170	195	215
240	245	265	270	301	305	320	322	345	355	410	420
410	415	435	440	471	475	490	492	515	525	580	590
640	645	665	670	701	705	720	722	745	755	810	820
810	815	835	840	871	875	890	892	915	925	980	990
960	965	985	990	1021	1025	1040	1042	1065	1075	1130	1140
160	500	140	147	105	110	90	90	60	65	8	8
320	660	300	307	265	270	250	250	221	225	168	168
435	775	415	430	385	390	365	370	336	340	283	283
575	915	555	570	525	535	505	510	476	480	423	423
665	1005	645	660	615	625	595	600	566	570	513	513
785	1125	765	780	740	740	715	720	686	690	633	633

Problem 2 Optimal solution

In figure 9 below, a train diagram for the optimal solution is sketched. The resolution of the conflicts depicted by circles on the input train diagram and hidden conflicts was successful. To achieve this the developed solution algorithm and all the constraints presented in this thesis have been strictly followed. For this particular problem, there are six meet points and twelve trains. Six of the trains are traveling in the eastbound direction, and the others are traveling in the opposite direction in a single-track railway. The stations are located at 0, 25, 85,134,160, and 230. On the input train diagram plot on figure8 there were several conflicts between these stations, but here in figure 9, there are no conflicting movements in between before mentioned locations. Other station conflicts are also resolved and can be seen in the text conflict report

A schedule with the form of a real-world rail network was used to test and evaluate the developed application. The sample rail network problem considered both complex and manageable schedule to see the effect of variation in the number of train and station on the result of the developed model. The following test is carried out to see the effect of the weight of the priority of trains on the optimal solution according to the following assumptions. Assumptions 1: $priority\ 1\ >\ priority\ 2\ > priority\ 3$ Assumption 2: $\%\,(priority\ 1) +\ \%\,(priority\ 2) + \%\,(priority\ 3) = 100\%$. In table 5 assigned each train type three types of trains are considered fast passenger, slow passenger, and freight train) with respective priorities as given in the table 5.

As shown in table6 for each priority separate weight have been given to test the impact of priority on the optimal solution. The dispatcher has an option to set the weigh at the GUI. In table 7 hereunder shows the test result from the developed model based on the assumption presented above. Based on these results it can be said that, for most of the cases, the more evenly distributed set of priorities the less cost the optimal solution will be. However, this is not the case sometimes as for set 2 and set 3 in schedule 2 showed different results from this conclusion. Therefore, it will be up to the train dispatcher to decide which set of weights best fit the rules and regulation of the railway company as well as the reality on the ground. In addition, as expected

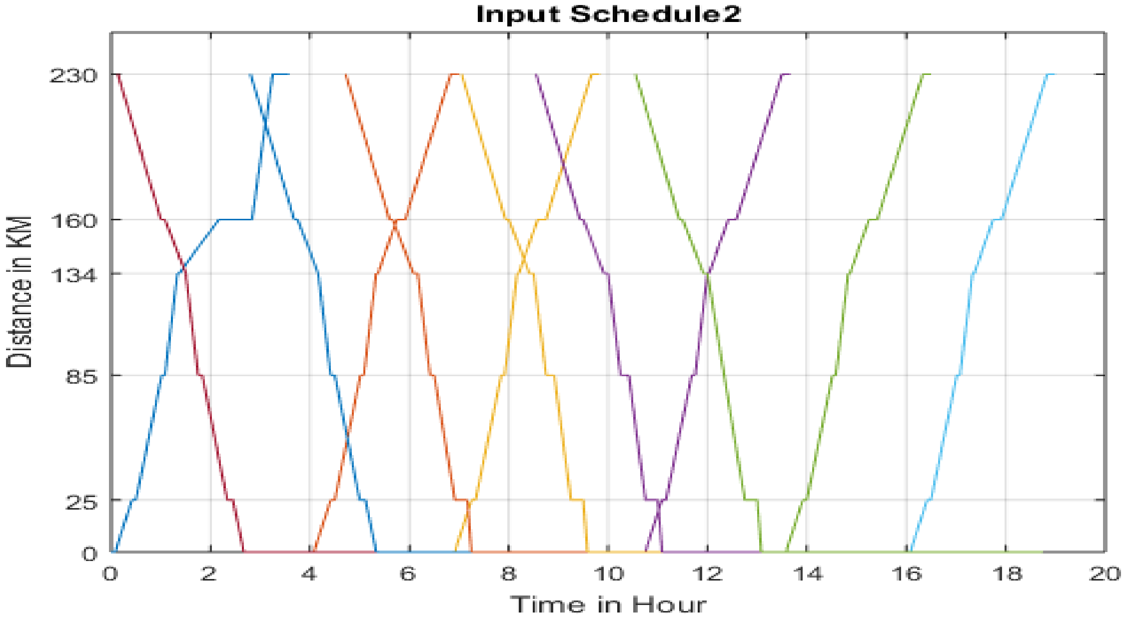

Figure 8: Problem 2 initial schedule

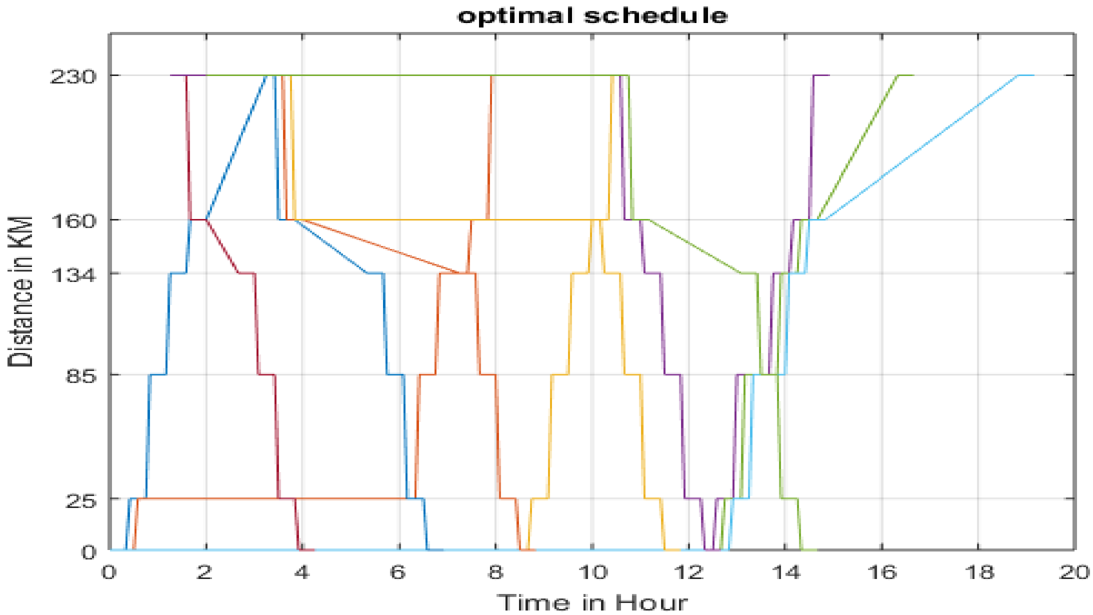

Figure 9: Problem 2 optimal solution

Table5: Priority of each train type

Train type	Weight
fast passenger train	Priority 1
slow passenger train	priority 2
freight train	Priority 3

Table6: Set of weights considered

	Priority 1	Priority 2	Priority 3
Set 1	0.5	0.3	0.2
Set 2	0.6	0.3	0.1
Set 3	0.7	0.2	0.1

Table 7: Variation in optimal solution cost

Input schedule	Set of weights for each priority	Optimal solution cost
1	Set 1	41
	Set 2	49.2
	Set 3	57.4
2	Set 1	85.5
	Set 2	96
	Set 3	95.5

CONCLUSION

A depth study on train dispatching problems has been done by formulating the train dispatching problem as a mixed integer linear program. The corresponding optimization criteria (objective function) take priority of train and total tardiness into consideration. The formulation consists of a carefully chosen practical rail network constraints. The method called “intlinprog” was used to solve the problem and the overall interactive dispatching support system was developed on MATLAB R2015b.The algorithm produced an optimal solution for all sample data from the tests carried out on the different sample data. The result was finally presented using an interactive GUI that accepts all necessary inputs from the dispatcher and process using the developed algorithm and then returns the outputs as train diagram, timetable, and conflict report. The main objectives of this case study , which was to obtain an optimal resolution for train dispatching problem and to develop intelligent dispatching, was achieved as it helps the dispatchers in making decisions in solving dispatching problems they face by finding optimal solution. Finally, independent of the complexity of the problem the algorithm was able to reach an optimal solution.

REFERENCES

[1] S. A. Afeez, "Train Dispatching: Heuristic Optimization," Dalarna University, Sweden, 2006

[2] P. Pellegrini, G. Marlière, and J. Rodriguez, "A detailed analysis of the actual impact of real-time railway traffic management optimization," *J. Rail Transp. Plan. Manag.*, vol. 6, no. 1, pp. 13–31, 2016

[3] B. Adenso-Díaz, M. Oliva González, and P. González-Torre, "On-line timetable re-scheduling in regional train services," *Transp. Res. Part B Methodol.*, vol. 33, no. 6, pp. 387–398, 1999.

[4] S. Ismail, "Railway traffic control and train scheduling based on inter-train conflict management," *Transp. Res. Part B*, vol. 33, pp. 511–534, 1999.

[5] A. D. Ariano, D. Pacciarelli, and M. Pranzo, "Discrete Optimization A branch and bound algorithm for scheduling trains in a railway network," vol. 183, pp. 643–657, 2007.

[6] G. Caimi, L. Kroon, and C. Liebchen, "Models for railway timetable optimization: Applicability and applications in practice," *J. Rail Transp. Plan. Manag.*, vol. 6, no. 4, pp. 285–312, 2017.

[7] "Single Line Operation | The Railway Technical Website | PRC Rail Consulting Ltd." [Online]. Available: http://www.railway-technical.com/signalling/single-line-operation.html. [Accessed: 23-Oct-2018].

[8] J. Harbering and A. Ranade, "Single Track Train Scheduling," *Plan. a Public Transp. Syst. with a View Towar. Passengers' Conv.*, p. 151, 2015

[9] X. Zhou, "Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds," *Transp. Res. Part B*, vol. 41, pp. 320–341, 2007.

[10] V. Cacchiani *et al.*, "An overview of recovery models and algorithms for real-time railway rescheduling," *Transp. Res. Part B Methodol.*, vol. 63, pp. 15–37, 2014.

[11]M. Dotoli, N. Epicoco, M. Falagario, A. Piconese, F. Sciancalepore, and B. Turchiano, "A real time traffic management model for regional railway networks under disturbances," in *IEEE International Conference on Automation Science and Engineering*, 2013.

[12] A. D. Ariano, M. Pranzo, and U. Siena, "Conflict resolution and train speed co-ordination for solving timetable perturbations," *Energy*, vol. 2005, no. June, pp. 1–21, 2005.