Multi Agent Based Railway Scheduling and Optimization

Poulami Dalapati
Dept. of IT
NIT Durgapur
India
dalapati89@gmail.com

Arambam James Singh
Dept. of IT
NIT Durgapur
India
jamesastrick@gmail.com

Animesh Dutta
Dept. of IT
NIT Durgapur
India
animesh.dutta@it.nitdgp.ac.in

Swapan Bhattacharya
Dept. of CSE
NIT Surathkal
India
bswapan2000@yahoo.co.in

***Abstract*—This paper proposes a multi agent based timetable scheduling algorithm for railway system which handles the in−between time delay of the newly introduced train. The delay management indeed optimizes the total journey time, hence increases the total utility of the whole railway system as well. Here we show that schedule generated by our proposed algorithm is the most optimized schedule. It is done by using the notion of DCOP(Distributed Constraint Optimization Problem), where we define some metric to analyze the system to achieve our goals. We use JADE(Java Agent DEvelopment Framework) platforms to simulate our work and test it using a small network. We also take a small case study to compare our proposed work with the existing one and the results are therefore presented.**

***Keywords*-Multi agent system, Railway Timetable, Railway scheduling, Distributed Constraint Optimization.**

I. Introduction

Managing interactions among autonomous entities with ever increasing inter− dependencies have been one of the biggest challenges for distributed artificial intelligence and for multi agent systems. These aim at developing and analyzing models derived from social interactions in human societies, and applying them to computational systems in order to resolve conflicts in organizational structures of various types. The increasing demand for mobility in the 21st century poses a challenge to researchers [21] which includes necessity of efficient tools and techniques in order to deal with transportation problems as control, optimization, efficient assignment of the demand, and so on. During the last years, agent based approaches to traffic simulation have shown that they are able to capture necessary details at entity level and is able to generate relevant realistic phenomena. They are explicitly presented as active, heterogeneous entities in the environment where their behavior can be visualized, monitored, and validated at individual level, leading to new possibilities for analyzing, debugging, and illustrating traffic phenomena. So there are number of challenges present which should be handled efficiently and as well as automatically as much as possible to reach the level of perfection. The main challenges are like, *Timetable scheduling, Dynamic rescheduling, Signaling, Collision detection and resolution* etc. Among all these scenario, timetable or railway scheduling is the most important task in public transport. Not surprisingly, the construction of timetables in a sufficiently fast and optimized way is a troublesome job as the situational complexity of the physical network and the several constraints should be taken into account. A feasible train timetable should specify the departure and arrival time of all trains in such a way so that all the trains get required resources without any conflict, i.e. they satisfy all the constraints and as well as their total journey time gets minimized. In practical scenario the railway network of any country is very large. So, the difficulty increases in such complex and high loaded networks. The whole network is basically distributed all over the country which cover a very large region. So, when a train is to be scheduled it will go through number of tracks and stations throughout its journey. In some developed country like European Railway System, Swiss Railways, the whole railway system is managed by various companies. Their architecture and business strategies are also different. Practically, all the time there are large number of trains in circulation. So, introducing a new train in any domain, may it be the train whose journey covers number of zones or not, is not an easy job. Many conflicting issues appear when a new train comes into picture, mainly in inter−zonal journey. So, practically the authority has to consider all these constraints. Now this kind of real world problem is dynamic in nature, where not only the objective function changes over time, the constraints are also changing [1], [2]. The main challenges in such Distributed Constrained Optimization Problem(DCOP) is: the dynamic constraints might make tracking the previous global optima less effective.

The structure of the paper is organized as follows: In section 2 we discuss about some previous works in related domain, Section 3 present scope of our work. Section 4 & 5, model our proposed system where we define some metrics. Section 6 highlights some problems as well as its solution following the algorithms of these solutions. In section 7 we discuss about our experiments and results and finally section 8 concludes.

II. Related Work

Man made systems very often have increasingly higher complexity levels and usually it is possible to decompose them into complex units with certain well defined functionality, which originate as a result of close collaboration of lower level unit groups. Various traffic environments, their control

978-1-4799-4075-2/14/$31.00

and scheduling have been focused by number of researchers [4], [5], [7]. Some of them have followed traffic regulation in centralized manner using traffic signs and control elements. The problem here with such centralized system is that the control is on only one system(central server), which is very vulnerable in practice. David Handford and Alex Rogers [3] have used utility based path trees to represent the factors which affect agent decisions. A multi agent approach especially a holonic approach [6] is very useful to describe such units. The main goal of creating a holonic agent system was first proposed by Koestler [22]. Holons may be treated in two ways: as elements making a bigger whole, as well as elements which are build from smaller components. Federio et. al. [8] in their work have presented a friendly, flexible, automated Decision Support System(DSS): MOM, whose main goal is to make a feasible and optimized timetable in efficient manner. This system generates the timetable automatically with modifications in parameters like departure interval, delay, number of tracks etc. according to the requirements and railway traffic condition. They have basically designed an architecture combining an interface and an optimizer to search for optimized, feasible schedule. They claimed that this DSS can optimize existing timetable schedule and automatically optimize timetable for new train and make them compatible with the existing one. L. Ingotolli et. al. [9] have proposed a solution for timetabling by assigning some priorities for each track to resolve conflicts between trains. But in case of delay in high priority trains, there is always a chance that the lower priority trains get affected all the time and hence the total delay of the whole system goes on increasing. Some researchers [10], [11] have proposed meta−tree based structure to solve the railway scheduling problem as a Constraint Satisfaction Problem(CSP). Where they partitioned the whole network to construct the whole problem as number of connected sub problems. In this way M.A. Salido et. al.[11] assumed that two connected stations have only one line connecting them. They have used a software METIS to partition the networks. But this did not take into account the additional information like railway infrastructure or the type or priorities of trains to guide the partition process. Hence the ultimate cluster may not ensure accurate one. Most of the previous researches [12] have been dedicated to compute optimum railway timetables where delay optimization are not addressed properly. Some work have been done on agent based dynamic decision making or scheduling [13], [14] where researchers presented self interested agents, which can automatically cooperate with other agents to solve delay problem. Ying Yu et. al [14] gave an idea of Resource Constraint Project Scheduling (RCPSP) to achieve a valid scheduling solution to minimize the whole duration or cost of the system. Jhiag Zhibin and Xie Chao have discussed about delay simulation and management [15] on the basis of multiagent system where there are several number of agents like train agent, strategy agent. All these agents gather respective knowledge from the environment and accordingly make strategies to handle delay. Besides such conventional approaches very few researches are there where techniques like Logic Programming [16] and SSA(Singular Spectrum Analysis) [17] are used in train scheduling. Alessio Cuppari et. al. have adopted a logic programming based approach: CaseLP [19] for scheduling and management of freight train traffic along the railway line. They designed two kind of agents: Logical−Reasoning and Interface. Logical agents have complex reasoning capabilities and suitable for performing high level control and coordination tasks between MAS components. Whereas interface agents provides the link between external modules and the agents in MAS. CaseLP allows to realize MAS that which agent of different kinds and different architectures may coexist, cooperate and exchange information. They have used ontological analysis in this regard.

III. Scope of the Work

With this background there are number of works in train scheduling, very few significant work has come into notice which handle delay optimization in an effective manner in accordance with DCOP [1], [2]. They represent the timetable scheduling, but delay optimization which considers all practical constraints altogether are not there [11]. Hence, there is a huge scope of work in this regard and with this notion, we work on delay optimization in distributed railway network while introducing a new train in railway timetable taking into account all other trains which are already in circulation. To introduce a new train, the system first decides the source and destination station and the in−between path according to the demands, i.e. the intermediate stations the train will visit throughout its journey (Refer Fig. 1.). The system also takes into account various constraints like free time−slot, track connecting two stations, platforms etc. and accordingly checks for an optimized solution which satisfies all these constraints.

IV. System model

Now, as per as the Railway System is concerned, the problem of scheduling a new train in an existing timetable can suitably modeled through discrete mathematics where we can represent the Railway System as a multi graph(i.e. more than one arcs can be there between a source and destination node).

We put Railway Network(RN) as a pair of a graph(G) and an agency(A).

$RN =< G, A >$

G again consists vertices V and edges E where V is such that it can be modeled as station S. In general, a station can have more than one platforms and trains can stop here. So,

$V = \{v_i | i \in [1, n]\}$ and $v_i = s_i$ means vertex is a station.

$E = \{e_{ij} | i, j \in [1, n]\}$, where i is the index of source vertex and j is the index of destination vertex. e_{ij} exists iff v_i is directly connected to v_j .

There exist number of trains (T) which are already in circulation.

$T = \{T_i | i \in [1, k]\}$.

The agency is composed of number of agents as,

$A = \{A_i | i \in [1, m]\}$ Each station and train is associated with

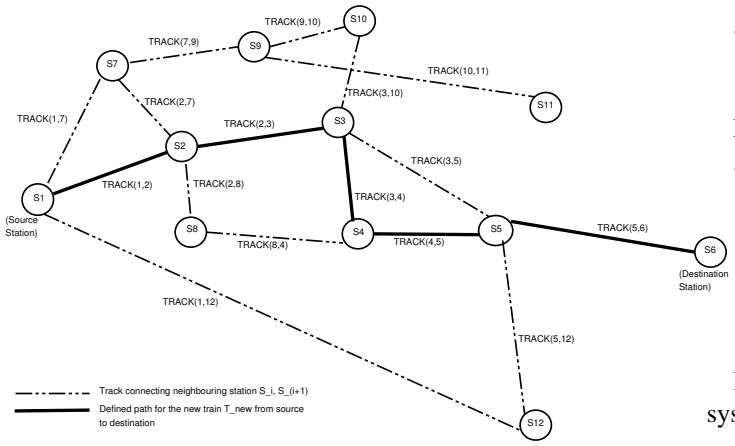

Fig. 1: Railway Network

an agent. SA and TA denote the station agent and train agent respectively, where $s_i \in S$ with $sa_i \in SA$ and $T_i \in T$ with $Ta_i \in TA$

A. Properties of the System

Properties of the Railway System can be expressed by fluents(functions whose values change over time) and by persistent functions(whose values do not change over time).

Persistent Function of the RN Physical Network

- $max\_capacity : S \rightarrow N$
  $max\_capacity(s_i) = n$ iff station s_i can host at most n trains. This information is only available to station agent sa_i.

Fluents Representing RN's Features

- $current\_capacity : N \times S \rightarrow N$
  $current\_capacity_t(s_i) = n$ iff station s_i has room for n more trains at time t. This information is only available to station agent sa_i .
- $running\_on : N \times E \rightarrow T$
  $running\_on_T(e(i,j,k)) = T_n...T_1$ iff $T_n, ..., T_1$ are the trains currently running on edge $e(i,j)$, where,T_1 is the first train that left the station and T_n is the last who is following previous trains maintaining a critical distance. This information is only available to the station agent sai in charge of the station from which the edge exits.

Persistent Functions of Train Schedule

- $route : T \times N \rightarrow S$
  $route(t_i, i_r) = S_i$ iff the i_r th station in $t's$ scheduled path is $S_i(i_r$ ranges between 1 and the maximum number of stations that t_i is expected to traverse)
- $scheduled\_arrival : T \times S \rightarrow N$
  scheduled arrival $(T_i, S_j) = n$ iff n is the time instant when T_i should arrive in S_j according to the planned schedule.
- $max\_speed : T \rightarrow R$
  $max\_speed(T_i) = r$ iff the maximum allowed speed for T_i is r, expressed in some suitable speed unit measure.
- $stop\_value : T \times S \rightarrow Bool$
  $stop\_value(T_i, S_i)$ is true if T_i will stop in station S_i and false otherwise.

Fluents of Train's Features

- $current\_position : N \rightarrow (V \cup E) \times R$
  current position $(T_i) = (v/e, d)$ iff T_i is currently either on vertex v, in which case r is 0, or on edge e, in which case d is the distance from the edge origin expressed in a suitable distance measure unit.

V. Metric Definition

Here we proposes some of the metrics to analyze the whole system of task allocation to agents.

- *Utility of resource availability*$(t_i \otimes t_j)$: The utility of the resources $(track\ (Track(i,j)), platform(P))$ is a binary element which takes either 0, if the track(path) connecting two neighboring stations(S_i, S_j) in the contiguous time are free and $P(S_j)$ is available when T_{new} arrives at S_j. The value is 1 otherwise. By including this utility term in the utility function of our system, we can check whether the schedule is possible or not as if it is 1 then the track resource is not free and as we deducting this term in the utility function, hence the total utility will be much less. So we can conclude that schedule is not possible.
- *Utility for single path delay*(U_δ): In practical scenario almost every train faces some delay in their journey due to various constraints. For our system, we have taken some predefined utility value for delay for every single track connecting two neighboring stations, where utility is inversely proportionate with the amount of delay, i.e. for the best case scenario the delay $\delta = 0$, so utility value of that particular amount of delay $(U_\delta) = 1$. When δ increases, the utility value decreases uniformly. This takes care of the feasibility of the new schedule.
- *Threshold delay*(δT_h): The theshold delay is the maximum permissible delay beyond which the new train cannot be scheduled as we have to optimize the total journey time of the train. So to get an optimized schedule $\delta \leq \delta T_h$.
- *Total system utility*(U_{Total}): Using above mentioned metric we have formulated the total utility $U(S_i, S_{(i+1)}\ )$ as $U(S_i, S_{(i+1)}\ ) = (\gamma(t_i) - (t_i \otimes t_j) + U_\delta - U_{Th})$ for every connecting path of the neighboring stations $S_i, S_{(i+1)}$. $\gamma(t_i)$ has taken a fixed positive value for all free time slots of every station. Hence the total system utility can be calculated as $(U_{Total}) = \sum U(S_i, S_{(i+1)}\ )$, where $i = 1, 2, ..., n$.

VI. Problem formulation

We introduce a new train in the existing timetable with some delay optimization so that the train would face minimum journey time. And as the delay gets minimized so the total utility gets maximized leading minimization in cost as well. There are n number of stations, i.e. $S = \{S_i | i \in [1, n]\}$ and m number of trains where, $T = \{T_j | j \in [1, m]\}$ which

are already in circulation. We are assuming that every station S_i has a particular number of platforms $P, (0 < P < n)$ and there is only one track connecting S_i to $S_{(i+1)}$. First every station S_i will take their individual free time slot t_i at which they have available resources, i.e $(P \cup track > 0)$ and station agent SA_i will put a value $\gamma(t_i) = 0.5$ for every free slot, and -0.5 otherwise. And if the resources are available then $(t_i \otimes t_j) = 0$, where $(t_j) = t_i + Journey\_time_{Tnew}$ $(S_i, S_{(i+1)}\ )$ and 1 otherwise. We will set a threshold level for delay δ for each station within which the train can be introduced, denoted by δ_{Th} . We also fix some predefined utility value U_δ for each value of δ, like when $\delta = 0, U_\delta = 1$ the maximum value as system is facing no delay at all and will decrease as the delay increase i.e. for $\delta = 1, U_\delta = 0.8$ and so on. Now, the source station Si for its first available free slot will send a message $t_i + Journey\_time_{Tnew}$ $(S_i, S_{(i+1)}\ )$ to its immediate neighbor station $S_{(i+1)}$ to check $S_{(i+1)}'$s availability. Accordingly $S_{(i+1)}$ will check whether $t_i + Journey\_time_{Tnew}$ is matching any of its free time slots or not from its own database. So, there will arise three cases: *Case 1*: If the slot is available i.e. there are free resources at the expected time then it will simply forward a message to its next neighbor in terms of time t and send back acknowledgement ACK to its previous station. *Case 2*: Else if there is any other free time slot within the threshold delay limit i.e. starting time of free time slot at $S_{(i+1)}$, i.e. $(t_{S(i+1)}$ $-t_i + Journey\_time_{Tnew}\ (S_i, S_{(i+1)}\ ) < \delta_{Th})$, then it send a request to accelerated the train so that when it reaches $S_{(i+1)}$, the arrival time coenside with its free slots to provide required resources. *Case 3*: Else $S_{(i+1)}$ will send $NACK$ to its previous station S_i and no message will be forwarded to its next neighbor. For every path the utility will be calculated. As there can be many feasible timetable path for introducing a new train (without considering any optimization), we have introduced the modification of Max−sum to find out the optimized path to minimize the total journey time. So for each track the utility will be calculated as:
$U(S_i, S_{(i+1)}\ ) = (\gamma(t_i) - (t_i \otimes t_j) + U_\delta - U_{Th})$ and,
$U_{Total} = \sum U(S_i, S_{(i+1)})$, where $i = 1, 2, ..., n$.
$\delta_{Total} = \sum \delta(S_i, S_{(i+1)})$, where $i = 1, 2, ..., n$.
Algorithm 1. and **Algorithm 2.** therefore represent this phenomena to generate the optimized railway timetable schedule.

A. *Proof of Optimality*

lemma: *If there is a feasible schedule which is obtained from Algorithm 1. and Algorithm 2. in the above discussed way, then the schedule is optimized railway schedule for the train Tnew which faces minimum journey time.*

Proof: As previously mentioned, our railway system consists of n stations and m trains, where $m, n \geq 2$. m includes the trains in circulation as well as the new train that will be introduced. Journey time of T_j , where $j \in [1, m]$, from station S_i to S_{i+1} , where $i \in [1, n]$, is denoted by $JT(S_i, S_{i+1})$. A time schedule for a particular track is a sequence of tuples $(T_j, ST_{T_j}\ , FT_{T_j})$, where,
$T_j : T_{new}$, i.e. the train to be introduced,

Algorithm 1 : Scheduling T_{new} and Delay Handling

for $\forall T_{new} \in T$ **do**
  for $\forall S_i \in S$ **do**
    if $(TRACK(S_i, S_{(i+1)t_i}\ ) \bigwedge P(S_{i+1}) = 0$ **then**
      $Free\_Slot(S_i) = t_i$
    end if
    for $\forall t_i$ **do**
      $\gamma(t_i) = 0.5$
    end for
    $JT(S_i, S_{i+1}) = FT(S_{i+1}) - ST(S_i)$
    $\delta = AJT(S_i, S_{i+1}) - JT(S_i, S_{i+1})$
  end for
  $\delta_{Th} = 30$
  $U_\delta = 1$
  for $\forall \delta$ **do**
    $U_\delta = U_\delta - 0.2$
  end for
  if $\delta \leq \delta_{Th}$ **then**
    $U_{Th} = 0$
  end if
  if $\delta > \delta_{Th}$ **then**
    $U_{Th} = 1$
  end if
  $CALL(Algorithm2.)$
end for

Algorithm 2 : Computing Total Utility

$U_{Total} = 0$
for $\forall t_i$ **do**
  for $\forall S_i \in S$ **do**
    $U(TRACK(S_i, S_{(i+1)}\ ) = (\gamma(t_i) + U_\delta - U_{Th})$
    $U_{Total} = U_{Total} + U(TRACK(S_i, S_{(i+1)}\ )$
  end for
end for

ST_{T_j}: departure or start time at station S_i, and
FT_{T_j} : arrival or finish time at station S_{i+1}).
$ST_{T_j}\ (S_i) < FT_{T_j}\ (S_{i+1}) \leq ST_{T_j}\ (S_{i+1})$, i.e. the train can only start from (S_{i+1}) after it finishes its journey from S_i at (S_{i+1}) and finish time will always be greater than the start time from its previous station.
and
$JT(S_i, S_{i+1}) = FT_{T_j}\ (S_{i+1}) - ST_{T_j}\ (S_i)$
For the best schedule it is required that the train T_j will spend exactly $JT(S_i, S_{i+1})$ amount of time on a track connecting S_i and (S_{i+1}). And also the time schedule will be such that, there will be no overlapping time sequence, i.e. the train can spend time only on a single track at a time.
Now, when $n = 2$ and $m = 1+1$, (one train in circulation and one is to be introduced), the optimal schedule can be trivially obtained. The train schedule is such that there is no delay for T_j inbetween except at the end of its journey time at (S_{i+1}) . Whereas there is also no delay for T_j except at the beginning at S_i . So, combining these two scenarios it is proved that the

delay is optimized in this case. But in some practical cases the required resources are not always free. So some delay comes into the picture.
$AFT_{T_j}\ (S_{i+1}) = FT_{T_j}\ (S_{i+1}) + \delta$, where,
$AFT_{T_j}\ (S_{i+1})$: actual finish time at station (S_{i+1}).
δ : delay in its journey.
and
$AJT_{T_j}\ (S_{i+1}) = JT(S_i, S_{i+1}) + \delta$,
where,
$AJT_{T_j}\ (S_{i+1})$: actual journey time of T_j.
For our proposed system, as δ is the minimum delay that should be given to T_j for getting the resources available and if $\delta < \delta_{Th}$, then we can consider it as no delay at all.
So, $JT(S_i, S_{i+1}) + \delta \approx JT(S_i, S_{i+1})$.
This also supports the best case scenario. Now in such way we can check for the ST_{T_j} where $i = 0$, i.e. source station, for which all the $FT_{T_j}\ (S_{i+1})$ as well as $ST_{T_j}\ (S_{i+1})$ will be a contiguous time sequence with that small permitted δ. Using δ_{min} where $\delta_{min} = \delta(S_i, S_{i+1}), i \in [1, n]$, we can obtain the optimal solution which supports our claim as well. In this way we can present the whole path from source to final destination as the summation of all the individual paths connecting a station to its immediate neighbor. For every single track or path, where $1 \leq track \leq n-2$, the lemma is true. So, it is true for the whole path as well. Now for the last station it is obvious that the train T_j will not face any delay at the end station according to our assumption. Hence the schedule we get is the optimal one.

B. Indian Railway System : A Case Study

In Indian Railway System, which is indeed a very very large network is broadly distributed all over the country (divided into 17 main zones in total, which are again divided into number of divisions and sub−divisions). Therefore throughout the day a large number of trains are in circulation. So, introducing a new train in any domain, may it be the train whose journey covers number of zones or not, is not an easy job. Many conflicting issues appear when a new train comes into picture, mainly in inter zonal journey. So, practically the authority has to consider all these constraints. Moreover traditionally, in Indian Railway System the making of timetable is done manually. As per the discussion with railway personnel of Eastern Indian Railway, every zone has their own database regarding the resources like station, track and record of trains in circulation with time basis. Whenever any new train is going to be introduced, at first the source and destination and also the route in which the train will be scheduled and at what time, is decided according to the demand. Then all the zones which are involved in this, individually checks for the availability of resources depending on time and forwards to the central authority. This authority then maps all the zones free resources for the particular train to check whether the new train can be scheduled on that route at that time or not and accordingly gives the permission to the zones. But this is not at all an easy process and takes huge amount of time to compose the whole issue. According to the information gathered from the railway personnel it takes at least 20 days to schedule a new train in internal routes. And as a train goes through several zones, so handling the inter zonal dependencies and conflicts in an optimized way(in terms of both time and cost) is really a challenge in present scenario. We hereafter represented current railway timetable scheduling as an algorithmic way in **Algorithm 3.**.

Algorithm 3 : Existing Timetable Scheduling

for $\forall T_{new} \in T$ **do**
  for $\forall S_i \in S$ **do**
    $Arr\_time(S_{i+1}) = Dep\_time(S_i) + JT(S_i, S_{i+1})$
    for ∀Arr_time(S_{i+1}) **do**
      if $(TRACK(S_i, S_{(i+1)t_i}) \bigwedge P(S_{i+1}) = 0$ **then**
        $Free\_Slot(S_i) = TRUE$
        $CENTRAL\_SERVER(S_i) = 1$
      end if
    end for
    $Result = (CENTRAL\_SERVER(S_i)) \bigwedge$ (CENTRAL_SERVER(S_{i+1}))
  end for
  if $Result = 1$ **then**
    $SCHEDULED$!!!
  end if
end for

VII. Experiments and Results

In order to simulate our proposed methodology i.e. to schedule a new train in an existing timetable with delay optimization, we use JADE(Java Agent DEvelopment Framework)[23], which is a software Framework implemented in Java. We consider six stations with defined paths along with six trains(which are already in circulation) and a new train T_{new} which is to be introduced. Stations can have more than one platforms and there is only one track connecting two neighboring stations. These two parameters(tracks and platforms) are considered as resources of the railway system. We run our algorithm for different trains at different time slots as the scenario is dynamic in nature and time dependent. Fig. 2 represents the feasible optimized schedule for the new train. We also consider different amount of threshold delay as mentioned before in Section 5 to check the changes in optimal solutions.
Fig. 3 shows the comparison between existing solution and our proposed solution. It is clearly seen that our scheduling algorithm generates much less delay. And also we can see that in existing algorithm no schedule can be produced for time slot 2 from the source station, which is possible with our algorithm.

VIII. Conclusion

An algorithm and its simulated result is presented in this paper based on the optimization and scheduling approach. The features of the system along with the working principle

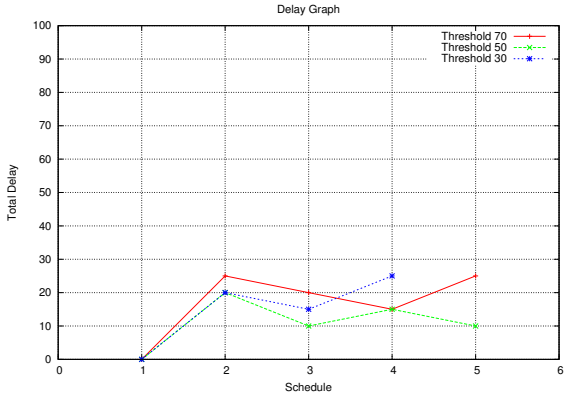

Fig. 2: Optimized Schedule

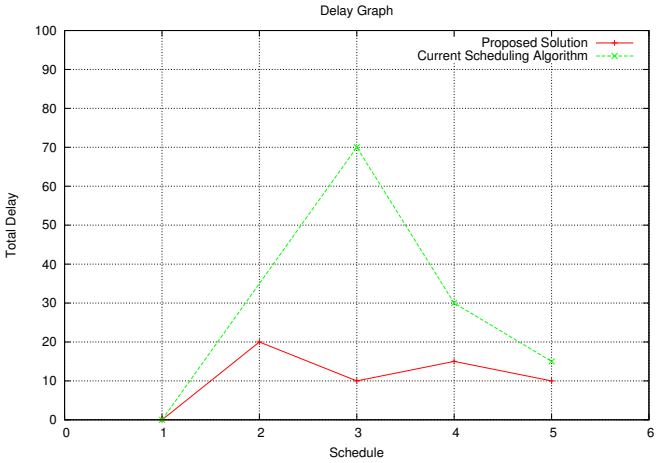

Fig. 3: Comparison between existing schedule and our proposed optimized schedule

is described. We perform a case study to prove our claim. Future work focuses on the dynamic rescheduling extending this approach.

REFERENCES

[1] Trung Thanh Nguyen and Xin Yao, Benchmarking and Solving Dynamic Constrained Problems, IEEE Congress on Evolutionary Computation, pp. 690-697 (2009).

[2] Jian Caoa and Sijie Caib and Yanhai Zhaoc, A Distributed Asynchronous Constraint Optimization Algorithm based on Dynamic Reduction of Constraint Graph, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 365-369 (2010).

[3] David Handford and Alex Rogers, Modelling Driver Interdependent Behaviour in Agent-Based Traffic Simulations for Disaster Management, Advances on Practical Applications of Agents and Multiagent Systems, AISC 88, Springer- Verlag Berlin Heidelberg, pp. 163-172 (2011).

[4] Maksims Fiosins and Jelena Fiosina and Jorg P. Muller and Jana Gormer, Agent-Based Integrated Decision Making for Autonomous Vehicles in Urban Traffic, Advances on Practical Applications of Agents and Multiagent Systems, AISC 88, Springer-Verlag Berlin Heidelberg, pp. 173-178 (2011).

[5] Neila Bhouri and Flavien Balbo and Suzanne Pinson, Towards Urban Traffic Regulation Using a Multi-Agent System, Advances on Practical Applications of Agents and Multiagent Systems, AISC 88, Springer-Verlag Berlin Heidelberg, pp. 179-188 (2011).

[6] Jarosaw Kozlak and Sebastian Pisarski and Magorzata Zabinska, Applica- tion of Holonic Approach for Transportation Modelling and Optimising, Ad- vances on Practical Applications of Agents and Multiagent Systems, AISC 88, Springer-Verlag Berlin Heidelberg, pp. 189-194 (2011).

[7] Jarosaw Kozlak and Sebastian Pisarski and Magorzata Zabinska, Multi-agent Models for Transportation Problems with Different Strategies of Environment Information Propagation, PAAMS 2013, LNAI 7879, Springer-Verlag Berlin Heidelberg, pp. 145-156 (2013).

[8] Federico Barber and Pilar Tormos and Antonio Lova and Laura Ingolotti and Miguel-Angel Salido and Montserrat Abril, A Decision Support System for Railway Timetabling (MOM): The Spanish Case, pp. 235-244.

[9] L. Ingolotti, F. Barber, P. Tormos, A. Lova, M. A. Salido, and M. Abril, A Scheduling Order-Based Method to Solve the Train Timetabling Problem, AR- RIVAL (2006).

[10] Montserrat Abril, Miguel A. Salido, and Federico Barber, Application of Meta- Tree-Based Distributed Search to the Railway Scheduling Problem, Interna- tional Joint CP/ICAPS Workshop on Constraint Satisfaction Techniques For Planning and Scheduling Problems, pp. 6-14 (2007).

[11] M.A. Salido, M. Abril, F. Barber, L. Ingolotti, P. Tormos, and A. Lova, Domain-dependent distributed models for railway scheduling , ELSEVIER, Knowledge-Based Systems 20, pp. 186194 (2007).

[12] Christian Liebchen, Michael Schachtebeck, Anita Schobel, Sebastian Stiller, and Andre Prigge, Computing Delay Resistant Railway Timetables, Future and Emerging Technologies Unit of EC (IST priority - 6th FP), under contract no. FP6-021235-2 (project ARRIVAL), pp. 1-42 (2007).

[13] Shih-Chang Wang and Samuel Y. Ruan, Agent-Based Collaborative Decision- Making Processes for Dynamic Scheduling, IEEE, pp. 273-277 (2011).

[14] Ying Yu, Chengwei Wang, and Ji Zhu, An Agent-Based Dynamic Scheduling Solution for Resource-Constrained Project Scheduling, Fourth International Conference on Computational Intelligence, Modelling and Simulation, IEEE, pp. 120-123 (2012).

[15] Jiang Zhibin and Xie Chao, Multi-agent Delay Simulation Model in Mass Rail Transit System, International Conference on Measuring Technology and Mechatronics Automation, IEEE, pp. 717-720 (2009).

[16] Alessio Cuppari, Pier Luigi Guida, Maurizio Martelli, Viviana Mascardi, and Floriano Zini, Prototyping Freight Trains Traffic Management Using Multi- Agent Systems.

[17] Jafar Razmi, Nima Mirabedini, and Hassan Mina, A Novel Integrated Two- Stage Approach to Programming Train Scheduling with SSA Technique, Jour- nal of Management Science and Practice, vol. 1, Iss. 2, pp. 52-62 (2013).

[18] Zhang Shuxin and Jia Hongfei, Research on agent-based approaches to freight transport scheduling, International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE) December 16-18, Changchun, China, pp. 1840-1843 (2011).

[19] Alessio Cuppari, Pier Luigi Guida, Maurizio Martelli, Viviana Mascardi and Floriano Zini, An Agent-Based Prototype for Freight Trains Traffic Management.

[20] Mohamed Haitam Laarabi, Claudio Roncoli, Roberto Sacile, Azedine Boulmak- oul and Emmanuel Garbolino, An Overview of a Multiagent-Based Simulation System for Dynamic Management of Risk Related to Dangerous Goods Trans- port, IEEE (2013).

[21] Ana L. C. Bazzan and Franziska Klugl, A review on agent-based technology for traffic and transportation, The Knowledge Engineering Review, Cambridge University Press, pp. 1-29 (2013).

[22] Koestler, A., The Ghost in the Machine, Arkana Books (1989).

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