Network-centric approach to adaptive real-time train scheduling

Alexander A. Belousov*, Petr O. Skobelev*, Maxim E. Stepanov*

* Samara State Aerospace University (National Research University)
34, Moskovskoye shosse, Samara, 443086, Russia

# SEC “Smart Solutions”
443013, 17 Moscovskoe Shosse, office 1201, Samara, Russia
belousov@smartsolutions-123.ru

ABSTRACT

The paper proposes a network-centric approach to creating intelligent adaptive system of real-time train scheduling on the basis of multi-agent technologies. The architecture of the network-centric multi-agent system consisting of base planning subsystems is described. Subsystem interaction protocols and protocols of agent interaction within each subsystem are presented. The example of schedule planning in various situations is presented. Productive characteristics of the developed system are presented. Good quality of schedule planning and system performance is shown.

Keywords: railway dispatcher systems, train scheduling, multi-agent systems, network-centric approach, intelligent systems, method of conjugate interactions, adaptive planning.

1 INTRODUCTION

When we solve complex automation tasks of railway real-time train scheduling, we constantly have to deal with a lot of disruptive events, which differ in intensity and source and can lead to conflicts in schedules of different kinds of trains. This makes matters even worse in large-scale railway systems with a high level of connectivity. Today solving conflict situations in trains deviating from master schedule completely depends on the experience of the train dispatcher, which often leads to irrational decisions, especially in stressful situations. The constantly growing intensity and speed of the passenger traffic and hence results in increasing complexity of disruptive situations. This also provokes questions as how to reduce dependency on the human factor by automating the decision-making process, how to build intelligent system, enabling fast and effective adjustments in case of a disruptive event.

Train re-scheduling during railway traffic disturbances is an important problem. A Mixed-Integer Linear Program (MILP) model is proposed for train re-scheduling of N-tracked railway traffic during disturbances in [1]. In [2], the same model as proposed in [1] is used along with two solution methods: (i) right-shift re-scheduling to produce the initial feasible solution and (ii) local search to limit the search.

The train scheduling problem can be formulated as a job shop scheduling problem, as in [3, 4], where train trips are jobs which are scheduled on tracks that are considered as resources. Two studies [3, 5] addressed the problem from the perspectives of capacity, robustness, and dependencies. The heuristics and integer solution methods along with analysis are given in [3]. A variable speed dispatching system is proposed in [4] to control railway traffic by considering acceleration and deceleration time in the model. Furthermore, the work in [4] extends [6] with detailed microscopic and comprehensive models to fulfill additional requirements.

Studies of the computational complexity of disturbance handling in large railway networks are done in [7] and [8]. An experimental study of optimization (i.e. Minimize delay cost) and myopic based policies (i.e. First Come First Served etc.) concludes that both complement each other [7]. The performance of each one is dependent on the region and disturbance type. A performance evaluation of centralized and distributed strategies for dispatching trains is given in [8]. A greedy depth-first search branch-and-bound algorithm is proposed in [9] to handle the re-scheduling problem. It generates a feasible solution within 30 sec in most cases.
This paper describes the network-centric approach to solving the complex task of real-time train scheduling in a large-scale system. Modern large-scale railway train scheduling systems are characterized not only by a large number of participants, but also by high intensity and speed of trains, which is increasing year by year. Due to the high traffic intensity, trains are highly interconnected: changes in one train’s schedule, or a conflict with this train, will most certainly affect the next train, and with high probability have an impact on the whole train network. In this case re-scheduling of all trains in the network might be required, which should be done quickly, on the fly in real-time, which is a very difficult task given the whole variety of planning conditions, preferences and constraints. Constraints must be applied individually and can be regulated in the work process.

The presented network-centric approach was used to develop real-time train scheduling system. This system was adapted for production use in Russian Railways on a section of the high-speed rail Saint-Petersburg - Moscow and between Saint-Petersburg and Buslovskaya.

2 NETWORK-CENTRIC APPROACH TO ADAPTIVE TRAIN SCHEDULING

2.1 Problem statement

The suggested network-centric approach to solving the complex task of train scheduling is aimed at creating optimal schedules for trains, which are delayed due to unforeseen events in real time.

The developed system builds the initial train schedule, and then performs schedule corrections according to different events (infrastructure repair requests, actual train placements, infrastructure elements occupations and so on).

The input data consists of: railway infrastructure (stations, railway switches and block sections of railway infrastructure), requirements for train schedule (master-plan), maintenance requirements, updates on the current situation in the operating domain about trains and states of infrastructure block sections (signals of busy condition, information on unavailability). It is worth mentioning that the scale of the task is enormous which makes it a large-scale task.

The main limitations of the system are traffic security requirements, normative route-building requirements, train priorities, dispatcher rules etc.

Besides the limitations listed above there are ones which are hard to formalize (no thickening of schedule lines of train routes, no unjustified changing of tracks, no traffic jams between stations, no unjustified train stops on the main tracks, correct routing of arrivals etc.), which should be taken into consideration while planning. At the same time, the implementation of any requirements depends on the current situation.

For instance, a train may not choose opposite tracks for movement, but if there is a busy infrastructure block section on its path, then it can do so, in order to bypass the obstacle and stay on. On the other hand, it may stay on the same tracks, but only wait a little, if this delay is not long and the train can catch up its schedule. But waiting at a station means stop for a little while, and a train may only choose block sections of certain length for stay and so on. Hence, a simple decision whether the route should be changed or not, is connected to many conditions, which require finding a balance and consensus. For well-balanced decisions in such situations every train agent in the system is supposed to have collectable virtual “currency” (similar to potential energy), which is used for rewarding successful decisions and, on the other hand, can be spent on fines in case of bad decisions, compensating for the change of the route and the train schedule. This way, all the requirements and limitations which are hard to formalize, can be reduced to a universal measure and be considered in the scheduling of train routes.

The system for solving the task of adaptive real-time train scheduling runs on Vektor-M program platform [10, 11], which allows for keeping the dynamic infrastructure model of the operating domain, get signals from block sections, appointed maintenance windows, satellite and other information.
2.2 Network-centric system architecture

The architecture of the developed system is built on the network-centric principles, where every subsystem has its own individual task and the final solution is reached through negotiating between individual decisions [12].

The primary plan building takes place in 2 subsystems. Each subsystem builds a train route schedule on its own level of understanding of the scene in such way that the initial rough decision is transformed into more precise one. A decision made in every subsystem is conflict-free for its level of understanding (no converging train routes, the security requirements are intact). This layer-based train scheduling eliminates the combinatorial explosion of possibilities, make the scheduling process more stable to disruptions due to reducing the scale of the task on higher levels and step-by-step considering all possible limitations according to the level of importance and impact on other layers.

All events arriving in the system can be divided into two main types: new request or update on the current situation. Requests in their turn can be of the two basic types: request to let the train pass on schedule or request to conduct maintenance works. Update on the current situations can be either a train moving along block sections or a state of an infrastructure (damage or busy condition).

The first level of planning is represented by a trajectory scheduler, the second one by a time scheduler. A general decision-making method and the role of schedulers in it are shown in Figure 1.

The diagram describes interaction between systems on the high level of abstraction. The diagram blocks in their turn can be subsystems with complex inner structure.

Work in the system starts with a message “Start”, which is sent to the agent “Path scheduler agent”. In response the function “Prepare initial schedule” appears, which prepare the planning scene and after which the event “Received conflict” appears in the time scheduler. In the result, the time scheduler detects current schedule conflicts and determines further direction. If no conflicts are found, “Path scheduler agent” is sent a message “All conflicts resolved”, and by processing the message the function “Prepare final schedule” forms the overall schedule. If conflicts are found, there two variants of events: to send the message “Rebuild problem trajectories” to the trajectory scheduler or to solve conflict on its level, by changing time of resources occupation and sending “Resolve conflict” to station and station limits agents.

In the first case, the trajectory scheduler will change tracks and planned block sections of trains’ stops in the schedule problem zone and will send a message “Problem trajectories are re-built” to “Time scheduler agent”; then the agent “Update scene” is called, which start link updating process of requirement-possibilities network between requirements of train traffic and infrastructure elements, after which the event “Received conflict” emerges again in the time scheduler.

In the second case, in the process of resolving conflicts in their own schedule, station and station limits agents create conflicts in train schedule, so called “Gap”, which appear due to the difference in arrival and departure time on the adjacent elements of infrastructure. Solving “Gap” conflicts takes place in the method “Close up gap”, implemented by train agents, resulting in new conflicts of resources occupancy, which is checked when event “Received conflict” appears. Reaching a compromise between train agents and agents of infrastructure elements (stations and station limits) is the final objective of the planning system.
If a compromise is reached and all conflicts are resolved, but decision does not satisfy the requirements, a message “Check scene” initialize the process of proactivity, in result of which certain agents try to improve their own schedule and cause a new field of resolving conflicts.

Time scheduler represents an operating domain as a set of station limits and stops, which builds a train schedule in less detail, conflicts are resolved by queueing the trains, speeding them up and slowing them down.

Time planning is similar to a visual schedule analysing. The main task is to build a new possible train schedule considering the normative schedule limitations and train priorities. The solution is based on the method of conjugate interactions for managing resource allocation in real time [4]. On this level a train agent creates subtasks (operations) for passing a station limit or stay at a station for a certain amount of time [13]. An agent of every subtask of this kind looks for a placement for itself in the respective resource, trying to find the most profitable position by negotiating with other subtask agents. High-priority trains are more active in finding a placement (have more energy for pushing other requests for resources).

The main decision-making condition here is accomplishing the task with minimal divergence (1).

\[
DEV_T = \sum \left( |TD_S - TP_S| + |TD_F - TP_F| \right) \rightarrow \min ,
\]  

\(1\)
where TDs – scheduled starting time, TDF – scheduled finishing time, TPs – actual starting time, TPf – actual finishing time, N – number of resources (station limits, station platforms), where operations of passing and stopping can be implemented. Additional conditions implemented while making decisions are listed below. The result of the time scheduler system’s work is a schedule for stay and passing stations and station limits in an operating domain, which is sent to the trajectory component in order to build train routes according to infrastructure block sections, provided it is possible.

While scheduling on time scheduler there are two types of interaction:

1) Interaction between interval agents of different trains on one block section: this algorithm enables to provide for an order change of moving trains on block sections;

2) Interaction between interval agents of one train on different block sections. This algorithm enables to keep the trajectory entirety of each train (if a train is late at some block section, it requires correcting the schedule of this train on all its block sections).

Each type of interaction is used for the decision-making of respective type of conflict, and one type of conflict might result in appearance of conflicts of other type.

In the trajectory subsystem a trajectory on block sections is built according to the calculated graphic in less detail, conflicts are resolved due to bypassing and route changing.

The primary task of scheduling paths and stays is to allocate routes for trains to take, and choose the block section for their stays considering overlapping routes of arriving and leaving for the parking. In this subsystem a train agent creates new subtask agents, which look for routes for passing the station and station limit according to the condition of minimal route costs. “Cost” is the cumulative key performance indicator of the route, which includes different normative requirements for train routes (correct or not, length, number of connections). After scheduling with minimal KPIs, station route agents enter the active phase of life cycle, where the main condition for decision-making is no overlapping of train routes in a block section. When such an overlap is found, a station route agent will try to transmit one of the conflicting subtasks to other route agents. Route agents communicate via the task exchange protocol [14, 15].

Building and negotiating the final train schedule takes place in a close interaction between scheduler levels. In every planning subsystem there is a swarm of agents, representing the level, between the subsystems there are back links which come into play when a conflict cannot be resolved locally in the current subsystem. The primary allocation of tasks to resources is done based on the best decision possible independently and in concurrent threads, which allows for reduction of computing time by excluding the rest of possibilities. Such “greedy” allocation results in conflicts that are resolved by agents grouping together into structures within a swarm – domains. In each domain searching for a compromise takes place between agents in order to resolve the conflict.

2.3 Train movement modelling

System builds train schedule and dispatchers’ corrections based on train movement by infrastructure block sections modelling.

Train rout consists of ordered list of infrastructure block sections. These block sections must have connections between them. Besides this spatial continuity, train rout must have time continuity. I. e. infrastructure block section occupation time must be equal to release time of previous block section in a rout. Train rout is built by scheduling system.

Train movement model takes into account train characteristics of acceleration and braking on infrastructure block sections. Let us consider that there is only one value of acceleration or braking on one infrastructure block section during train movement. Train speed on an infrastructure block section will be in the form (2).

\[ v^i(t) = v_0^i + a^i t, \]  
(2)
where $v_0^i$ — train speed at the entering of block section number $i$, $a^i$ — train acceleration or braking at the infrastructure block section number $i$. If $a^i < 0$ we consider that train brakes, else train accelerate.

Consider that coordinate of the train “head” marks his placement at infrastructure. Then dependence of the train placement at the infrastructure block section from the time variable will be as follows (3).

$$x^i(t) = x_0^i + v_0^i t + \frac{a^i t^2}{2},$$

where $x_0^i$ — the start of infrastructure block section $i$, it must be equal to the end of previous block section. Then $t^i = \frac{v^i - v_0^i}{a^i}$ — movement time by the infrastructure block section $i$. If acceleration at infrastructure block section $a^i = 0$ (uniform motion case), then $t^i = \frac{x^i - x_0^i}{v_0^i}$.

Using these equations, modeling subsystem uniquely determines train placement at the infrastructure while evaluating the effect of different options and decisions in the scheduling scene.

### 3 SOFTWARE IMPLEMENTATION

**3.1 Realization features**

The following time indicators for performance analysis of the multi-agent system for adaptive real-time train route management were used: infrastructure load time, train load time in an operating domain, re-scheduling time for a new arrival, re-scheduling time depending on duration of maintenance window, re-scheduling time for added maintenance windows, re-scheduling time depending on the number of tracks with maintenance windows, re-scheduling time depending on speed limits on a block section, re-scheduling time depending on the number of limitations, speed on a block section, re-scheduling time depending on the number of tracks occupied due to a speed limit at a station.

A train in an operating domain has around 45 operations (operations of passing the operating domain, stop at a station or passing a station), every train operation has its own agent, and there are around 800 trains in total. Overall are around 36000 agents. Apart from that, there are around 800 train agents, 49 station agents, 500 station route agents, 3700 block section agents and around 100-200 maintenance request and availability agents. Dividing this many agents into levels and grouping them into isolated swarms of agents, which are active at certain points of time, allows to increase the system performance [12].

In order to get implementation features for each index, an average value was defined, based on the results of 10 experiments for two operating domains: Saint-Petersburg – Buslovskaya and Saint-Petersburg – Moscow. Cumulative decisions instead of ones made by separate schedulers have been taken into consideration.

Figures relevant for planning the features of the operating domains are represented in Table 1. The Moscow – Saint-Petersburg operating domain has 2.8 times more infrastructure objects than the Saint-Petersburg – Buslovskaya operating domain.

<table>
<thead>
<tr>
<th>Operating domain</th>
<th>Number of stations</th>
<th>Number of infrastructure objects</th>
<th>Number of turnouts</th>
<th>Average preparation time of infrastructure (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saint-Petersburg – Buslovskaya</td>
<td>17</td>
<td>1293</td>
<td>133</td>
<td>1452</td>
</tr>
</tbody>
</table>
Saint-Petersburg – Moscow

It should be noted that the load time of the Moscow – Saint-Petersburg operating domain infrastructure is about 4.9 times higher than on the Saint-Petersburg – Buslovskaya operating domain. It is caused by building of the infrastructure model within the system and special infrastructure station agents, station limits, block sections.

Table 2 represents the change of scheduling characteristics depending on the number of scheduled trains in the operating domains. According to the Table, the difference between the scheduling times on the operating domains with around the same number of trains is 2.6 times, which is close to the difference between the amounts of infrastructure elements. Thus, dependency can be observed, which is close to linear, between the number of infrastructure elements and the scheduling time of trains on an operating domain.

Rescheduling time is 15-40% less in comparison with initial scheduling time. It is caused by adaptive rescheduling based on incoming events instead of rescheduling everything from scratch. Increasing the number of trains increases the train scheduling time in direct proportion.

<table>
<thead>
<tr>
<th>Operating domain</th>
<th>Number of trains</th>
<th>Average scheduling time (ms)</th>
<th>Average rescheduling time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saint-Petersburg – Buslovskaya</td>
<td>12</td>
<td>1511</td>
<td>1185</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>2180</td>
<td>1429</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>2257</td>
<td>1799</td>
</tr>
<tr>
<td></td>
<td>49</td>
<td>2300</td>
<td>2092</td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>2746</td>
<td>2356</td>
</tr>
<tr>
<td>Saint-Petersburg – Moscow</td>
<td>71</td>
<td>7202</td>
<td>5063</td>
</tr>
<tr>
<td></td>
<td>86</td>
<td>8786</td>
<td>6164</td>
</tr>
<tr>
<td></td>
<td>152</td>
<td>12691</td>
<td>11420</td>
</tr>
<tr>
<td></td>
<td>241</td>
<td>22453</td>
<td>22899</td>
</tr>
<tr>
<td></td>
<td>311</td>
<td>36143</td>
<td>37351</td>
</tr>
</tbody>
</table>

Table 3 represents time scheduling characteristics depending on the density of disruptions. Increasing density increases the scheduling time.

<table>
<thead>
<tr>
<th>Operating domain</th>
<th>Density of disruptions</th>
<th>Average scheduling time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saint-Petersburg – Buslovskaya</td>
<td>No disruptions</td>
<td>2356</td>
</tr>
<tr>
<td></td>
<td>Very few disruptions</td>
<td>2476</td>
</tr>
<tr>
<td></td>
<td>Few disruptions of the same type</td>
<td>2505</td>
</tr>
<tr>
<td></td>
<td>Many disruptions of different types</td>
<td>2785</td>
</tr>
<tr>
<td>Saint-Petersburg – Moscow</td>
<td>No disruptions</td>
<td>37351</td>
</tr>
<tr>
<td></td>
<td>Very few disruptions</td>
<td>39631</td>
</tr>
</tbody>
</table>
According to the table, the scheduling time mostly depends on the number of infrastructure elements and the number of scheduled trains. Other factors such as disruptions, their number, duration and density have less influence on the total scheduling time. This can be explained by partial (adaptive) scheduling according to incoming disruptive events instead of complete rescheduling. Thus, one can speak about guaranteed time for decision-making in given infrastructure and given number of trains.

The following qualitative characteristics can be noted: no thickening of lines on graphics of train routes, no unjustified changing of tracks, no traffic jams between stations, keeping security intervals, almost no delays among intercity and high-speed trains in conflict situations, average train delays less than 9% (20 trains engaged in one conflict).

This outcome has been achieved on such large-scale planning tasks for the first time.

### 3.2 Examples of resolved conflict situations

Let us consider a situation with a high number of disruptions as shown in Figure 2, with 6 maintenance windows, two of which completely block the traffic between Roshino and Zelenogorsk for an hour.

![Figure 2. Bypassing 6 maintenance windows, resuming after the disruptions have been eliminated.](image)

Resolving this situation required involving back links between different planning levels. Due to a high number of thickened graphic lines after the window the trajectory subsystem was unable to create the final schedule, since the traffic security requirements didn’t allow the trains to stop and switch on to the alternative route. The path scheduler registers this mismatch as a conflict and sends a message to the time scheduler. In order to resolve the situation, the time subsystem must delay a few trains from previous stations (for example, 2048, 6155, 6163), taking the overload of the station limits into account. It sends the newly made decision to the path scheduler. It builds the route and checks the decision to satisfy traffic security requirements. As a result, the schedule has turned out to be more balanced and stable to possible further disruptions, the effect of the maintenance window has been localized, after which the schedules tend to be exemplary again.
An overload between two stations cannot be resolved without the time scheduler, because changing the route with switching on to the opposite tracks will cost much more than changing the train schedule.

4 CONCLUSIONS

The suggested network-centric system of adaptive train scheduling based on multi-agent technologies have been developed within the project of the unified intelligent train scheduling system for the Russian railways and now is in production usage [16, 17].

The expected results of the developed adaptive train scheduling system include: reduced reaction time, increased flexibility and quickness of decision-making in response to disruptive events, increased effectiveness of railway resource management in real time and securing on-time performance of trains, reduced man-hours for rescheduling trains, a completely new intelligent software system for traffic management in real time. The mentioned solutions will help increase the quality of decision-making and performance level of end-users.

The project has been funded by the Russian Foundation for Basic Research, the Ministry of Education and Science of Russia under the SSAU “Increase of competitiveness among the worldwide leading research and education centers” program for 2013-2020.

REFERENCES


