

The problem of scheduling for the linear section of a single-track railway

Cite as: AIP Conference Proceedings **1738**, 110005 (2016); <https://doi.org/10.1063/1.4951874>
Published Online: 23 June 2016

Elena N. Akimova, Damir N. Gainanov, Oleg A. Golubev, Ilya D. Kolmogortsev, and Anton V. Konygin



View Online



Export Citation

ARTICLES YOU MAY BE INTERESTED IN

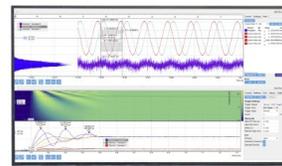
[A parallel matrix sweep algorithm for solving linear systems with block-fivediagonal matrices](#)
AIP Conference Proceedings **1648**, 850028 (2015); <https://doi.org/10.1063/1.4913083>

[On finding a density in a curvilinear layer by biconjugate gradient type methods](#)
AIP Conference Proceedings **1863**, 050009 (2017); <https://doi.org/10.1063/1.4992206>

[Railway obstacle detection algorithm using neural network](#)
AIP Conference Proceedings **1967**, 040017 (2018); <https://doi.org/10.1063/1.5039091>

Challenge us.

What are your needs for
periodic signal detection?



Zurich
Instruments

The Problem of Scheduling for the Linear Section of a Single-track Railway

Elena N. Akimova^{a,b}, Damir N. Gainanov^b, Oleg A. Golubev^b,
Ilya D. Kolmogortsev^b and Anton V. Konygin^{a,b}

^a *Krasovskii Institute of Mathematics and Mechanics, Ural Branch of RAS, Yekaterinburg, Russian Federation*

^b *Yeltsin Ural Federal University, Yekaterinburg, Russian Federation*

Abstract. The paper is devoted to the problem of scheduling for the linear section of a single-track railway: how to organize the flow in both directions in the most efficient way. In this paper, the authors propose an algorithm for scheduling, examine the properties of this algorithm and perform the computational experiments.

Keywords: scheduling, combinatorial graph algorithms, optimization.

PACS: 02.60.-x, 02.60.Pn, 07.05.Tp, 89.20.Ff.

INTRODUCTION

In this paper, we consider the problem of scheduling for the linear section of a single-track railway. The schedule optimization allows to increase a track capacity of the section using the same physical resources and to simulate further modifications of the section.

This problem with different modifications is well-known (see [1–3]). For example, in [1] the simulation of trains on the railway based on the moving-block system and fixed-block system was presented. We consider a modeling algorithm based on the idea of consistent change of trains direction.

The algorithm is implemented in C++ using the MPI + OpenMP hybrid technology.

Let the linear section of a single-track railway be given. Thus, we have a graph with the set of vertices

$$V = \{v_i \mid 1 \leq i \leq n\} \quad (1)$$

and the set of edges E (here we use standard notations from graph theory, see [4]). Furthermore, we assume that the vertices are indexed in such a way that the edges look like $\{v_i, v_{i+1}\}$.

For each station v_i , we have a positive integer $m(v_i)$, being the number of auxiliary railway tracks at this station. These tracks are the only places where a train can stop. Each auxiliary track can hold only one train. Therefore, if all auxiliary tracks of the station are occupied then a train cannot stop at this station.

By $l(e)$ we denote the length of the track corresponding to edge e from E . Suppose all trains have equal speeds. It should be a nonnegative time interval ε between two trains moving in same direction. Also, there is an isolated station $w \in V$ such that every train passing this station is obliged to stay at it for a nonnegative time δ due to technical reasons (the change of locomotive crew, train inspection, etc.).

Stations v_1 and v_n are sources and receivers of trains. Thus, there are two directions of movement: from v_1 to v_n and from v_n to v_1 (these directions are denoted by $v_1 \rightarrow v_n$ and $v_n \rightarrow v_1$, respectively). We can assume that each train does not change the direction and does not visit any vertex twice. For passing each other when moving in different directions, one of the trains waits another one on an auxiliary track.

Problem Statement

The problem is to construct and implement a scheduling algorithm to send as much as possible trains for period of time $[0; T)$ in both directions (more specifically, we are interested in the schedule at which the minimum number of trains in both directions over a specified period of time is maximal).

Let A denote the class of algorithms, that create such a schedule that, at any moment, there are no trains moving in opposite directions. Thus, whole period of time $[0; T)$ can be divided into disjoint time intervals $J: \bigcup_{I \in J} I = [0, T)$, and for each interval $I \in J$, trains move in fixed direction or stop on auxiliary tracks.

Now, we define a subclass A' of class A . For algorithms from A' , there exist functions $m_1: V \rightarrow \{0, 1, \dots\}$ and $m_2: V \rightarrow \{0, 1, \dots\}$ such that $m_1(v_i) + m_2(v_i) = m(v_i)$ and, for each station v_i , there are $m_1(v_i)$ auxiliary tracks for the direction $v_1 \rightarrow v_n$ and $m_2(v_i)$ auxiliary tracks for the direction $v_n \rightarrow v_1$. Thus, we divide all the auxiliary tracks into two sets for both directions. Now, the trains can use auxiliary tracks corresponding to their directions and the whole task can be divided into two independent subtasks for each direction.

ASSESSMENT OF TRACK CAPACITY

Examining the class of algorithms A' , we estimate the maximum track capacity when every time interval $I \in J$ has the same length equal Δt . Since, adding new stations without auxiliary tracks, we get the same problem, we can suppose that, without loss of generality, all edges have the same length 1:

$$l(\{v_i, v_{i+1}\}) = 1. \quad (2)$$

Without loss of generality, we assume that trains pass one edge per the unit of time. We construct a schedule with the maximum track capacity.

Theorem 1.

For the linear section of single-track railway with $\varepsilon = 0$ and without any isolated station w , the track capacity upper bound P is given by

$$P \leq f_0 \cdot p, \quad (3)$$

$$f_0 = \min_i \sum_{i \leq j \leq i + \Delta t} m(v_j),$$

where p is the number of time intervals, f_0 is the number of trains coming at the station v_n for Δt .

Corollary.

Without loss of generality to achieve the maximal track capacity, we can assume

$$\begin{aligned} m_1(v_i) = m_2(v_i) &= \frac{m(v_i)}{2}, \text{ for even } m(v_i), \\ |m_1(v_i) - m_2(v_i)| &\leq 1, \text{ for odd } m(v_i). \end{aligned} \quad (4)$$

Lemma 1.

Let $\varepsilon \geq 0$ and v_1, v_2 be adjacent vertices. Then, the track capacity upper bound P_1 is given by

$$P_1 = p \cdot \frac{\Delta t}{\varepsilon}, \quad (5)$$

where p is the number of time intervals.

Lemma 2.

Suppose that $\varepsilon \geq 0$ and there is no isolated station w . Then, the track capacity upper bound P_2 is given by

$$P_2 = p \cdot \min_i \sum_{i \leq j \leq i + \Delta t} m(v_j). \quad (6)$$

Lemma 3.

Let $p_{w,\delta}$ is maximal number of trains that can pass station w with $\delta \geq 0$ for a day, then $p_{w,\delta}$ is given by

$$p_{w,\delta} = p \cdot \frac{\Delta t \cdot m(w)}{\delta}. \quad (7)$$

Theorem 2.

For the linear section of a single-track railway with $\varepsilon > 0, \delta > 0$, the track capacity upper bound P is given by

$$P \leq \min\{P_1, P_2, p_{w,\delta}\}. \quad (8)$$

NUMERICAL EXPERIMENTS

Consider a mathematical model of linear section of a single-track railway with 65 stations and two of them are isolated with $\delta_1 = 80$ and $\delta_2 = 135$ (as isolated station w with delta described in introduction).

Using corollary of the Theorem 1 for constructing a scheduling algorithms from class A' which send as much as possible trains in both directions (see Problem statement).

Algorithm

For the time intervals Δt from 60 to 720 minutes with 60 minutes let's find the maximal track capacity P and the best time interval. The intervals lower then 60 minutes don't make sense, since the trains won't be able to proceed entire way for the given period of time. Below we give verbal description and pseudocode of the algorithm (see Table 1).

The algorithm was implemented in C++ (we used data standard structures, see [5, 6]) using MPI + OpenMP hybrid technology [7]. The data was distributed between nodes of supercomputers. Each node processes its own predefined set of time intervals. The communication between nodes is minimal, therefore we have obtained almost linear speedup.

For the experiment, the period of time is equal to 518400 minutes, which is exactly 360 days, was selected. Execution time of the program for different configurations can be seen from Table 2.

TABLE 1. Algorithm

Verbal description	Pseudocode
Stepping from 0 to T.	$V_{i,t} \leftarrow []$ for i in $(1..n)$, t in $(0..\Delta t)$
1. If there is no train at the first station (v_1), we create it there.	for t in $(0..\Delta t - 1)$:
2. Iterate stations from last to first. For each station check whether we can send a train from it. If we can, then we send a train with a minimal number.	for i in $(n..2)$:
Conditions to send a train is verified as follows:	while $m(V_{i+1,t+1}) > 0$:
a. there are free auxiliary tracks at the moment of train's arrival	$train \leftarrow \min(V_{i,t})$
b. train has time to come to the station before derrection change of movement	remove $train$ from $V_{i,t}$
	add $train$ to $V_{i+1,t+1}$
	end while
3. As we reach the first station (v_1), we create and send new trains as much as possible, according to conditions 2.a and 2.b.	end for
4. Increase the time counter and go back to 2.	while $m(V_{2,t+1}) > 0$:
	create $newtrain$
	add $newtrain$ to $V_{2,t+1}$
	end while
	end for

TABLE 2. Execution time of the program

Number of nodes and threads	Time (minutes)
1 node 1 thread per node	58
1 node 4 thread per node	16
2 node 4 thread per node	9

CONCLUSION

An algorithm from the class A^I was designed. Estimations of track capacity upper bound with $\varepsilon \geq 0$ and $\delta \geq 0$ were obtained. The software, which implements the algorithm using MPI + OpenMP hybrid technology, was created. The numerical experiments were performed on the Uran supercomputer with Intel Xeon E5-2650 processors.

ACKNOWLEDGMENTS

This work was supported by the Center of Excellence "Quantum and Video Information Technology" of Ural Federal University Development Program.

REFERENCES

1. KePing Li, ZiYou Gao, Bin Ning, "Cellular automaton model for railway traffic", in *Journal of Computational Physics*, vol. **209** (1), 179–192 (2005).
2. S. Harrod, "Optimal Scheduling of Mixed Speed Trains on a Single Track Line", in <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.85.5597>.
3. A. Higgins, E. Kozan, L. Ferreira, "Optimal Scheduling of Trains on a Single Line Track", in *Transpn. Res.-B.*, vol. **30** (2), 147–161 (1996).
4. Reinhard Diestel, "Graph theory", Springer-Verlag, New York 1997, 2000.
5. Th. Cormen, Ch. Leiserson, R. Rivest, Cl. Stein, "Introduction to Algorithms", third edition, MIT Press, 2009.
6. R. Sedgewick, K. Wayne, "Algorithms", fourth edition, Addison-Wesley, 2011.
7. W. Gropp, E. Lusk, R. Thakur, "Using MPI-2 Advanced Features of the Message-Passing Interface", MIT Press, 1999.