Applied Mathematics & Information Sciences

Volume 10
Issue 5 *Sep. 2016*

Article 16

9-1-2016

Optimal Scheduling for the Linear Section of a Single-Track Railway with Independent Edges Orientations

Elena N. Akimova
Ural Federal University, Ekaterinburg, Russian Federation\\ Krasovskii Institute of Mathematics and Mechanics, Ural Branch of RAS, Ekaterinburg, Russian Federation, aen15@yandex.ru

Damir N. Gainanov
Ural Federal University, Ekaterinburg, Russian Federation, aen15@yandex.ru

Oleg A. Golubev
Ural Federal University, Ekaterinburg, Russian Federation, aen15@yandex.ru

Ilya D. Kolmogortsev
Ural Federal University, Ekaterinburg, Russian Federation, aen15@yandex.ru

Anton V. Konygin
Ural Federal University, Ekaterinburg, Russian Federation\\ Krasovskii Institute of Mathematics and Mechanics, Ural Branch of RAS, Ekaterinburg, Russian Federation, aen15@yandex.ru

Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100516

Optimal Scheduling for the Linear Section of a Single-Track Railway with Independent Edges Orientations

Elena N. Akimova[1,2,*], *Damir N. Gainanov*[1], *Oleg A. Golubev*[1], *Ilya D. Kolmogortsev*[1] *and Anton V. Konygin*[1,2]

[1] Ural Federal University, Ekaterinburg, Russian Federation
[2] Krasovskii Institute of Mathematics and Mechanics, Ural Branch of RAS, Ekaterinburg, Russian Federation

Received: 13 Apr. 2016, Revised: 12 Jun. 2016, Accepted: 13 Jun. 2016
Published online: 1 Sep. 2016

Abstract: The paper is devoted to the problem of organizing the flow in both directions, in the most efficient way, for the linear section of a single-track railway. The authors propose an algorithm for scheduling with independent orientations of edges, investigate the properties of this algorithm and perform computational experiments. The authors also present some estimates for the track capacity of the section.

Keywords: Train scheduling, Combinatorial graph algorithms, Optimization, Track capacity

1 Introduction

In this paper we consider the problem of scheduling for the linear section of a single-track railway (see also [1]).

This problem occurs in scheduling trains. There are countries (e.g. Australia) where a significant part of the railway network is single-track. And almost all countries have single-track sections on their networks. Frequently, such sections are bottleneck. The time-table optimization allows one to increase the track capacity of the section using the same physical resources and to simulate further modifications of the section.

There are different modifications of the problem (see, e. g., [2], [3], [4], [5], [6]).

In [2] the simulation of trains on the railway based on the moving-block system and fixed-block system was presented.

In [5] is considered a case with only two stations. This case appears, for example, in private railways when a company transports loads between two production centers. It represents also an elementary section of a larger railway network. There are segments on the track and only one train can travel on a segment at one time.

In [6] authors consider single-track train scheduling problem in a case, when the track is divided into several block sections, each block can be occupied by only one train at the same time. However, between the blocks there are stations which have unlimited capacity. Here trains can wait in order to let trains from the opposite direction pass.

In this work, we consider the case where the whole time interval is divided into time intervals, in such a way that in every interval the direction of motion on tracks is invariable. The capacity of the stations are limited. Numerical experiments are based on an algorithm which is implemented in C++ with the use of the MPI+OpenMP hybrid technology.

There is the linear section of a single-track railway, i.e. some its stations are connected with single-track spans. Stations have auxiliary tracks. Auxiliary tracks are used for letting trains pass, for example. We will consider the sections as a labeled graph: the stations are the vertices, the spans are the edges, and the graph vertices are labeled by the numbers of auxiliary tracks.

2 Problem statement

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

$$V = \{v_i \mid 1 \le i \le n\}$$

* Corresponding author e-mail: aen15@yandex.ru

and the set of edges E (we use the standard terminology of graph theory, see [7]). Furthermore, we assume that the vertices are indexed in such a way that the edges look like $\{v_i, v_{i+1}\}$ for $1 \le i \le n-1$. We associate with each station v_i a non-negative integer $m(v_i)$ that is the number of auxiliary railway tracks at this station.

We denote by $l(e)$ the length of the track corresponding to an edge e from E. We will assume that all trains have the same velocity. The stations v_1 and v_n are the source and receiver of trains, respectively. Thus, there are two directions of motion: from v_1 to v_n and from v_n to v_1 (we denote these directions by $v_1 \mapsto v_n$ and $v_n \mapsto v_1$, respectively). We can assume that each train does not reverse the direction of motion and it does not visit any vertex twice. For passing each other, when moving in different directions, one train waits for another one on an auxiliary track.

Let $\Gamma = (V, E)$, $m : V \to \mathbb{N}^0$, and let T be the whole time period of scheduling. By adding new stations without auxiliary tracks we get the same problem. Without loss of generality we may assume that all edges have the same unit length:

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

We can also assume that a train passes one edge per one unit of time.

We consider a single-track railway. Trains can move on every edge only in one direction at any specific moment of time.

Thus, for a railway time-table R on Γ, we can define a map $s_R : T \times E \to \{-1, 1\}$, where $s_R(t, e) = 1$, if there is a motion in the direction $v_1 \mapsto v_n$ in R on $e \in E$, and $s_R(t, e) = -1$, otherwise. Let $\mathscr{R}$ be the class of all railway time-tables such that there exists a partition of the whole time period T into disjoint half-intervals of the same length τ

$$T = \cup_{i=0}^{k} [t_i, t_i + \tau),$$

where for all $R \in \mathscr{R}$ and $e \in E$ we have $s_R(t_i, e) = s_R(t_i + \tau', e)$, for all $0 \le \tau' < \tau$. Thus, each time-table from $\mathscr{R}$ has the following property: on every edge, in every semi-interval of time $[t_i, t_i + \tau)$, all trains are either immovable or they move in the same direction.

We need the following classes of railway time-tables.

1. Let a class $\mathscr{A}$ consist of all time-tables R from $\mathscr{R}$, such that for all $t \in T$ we have $s_R(t, e) = s_R(t, e')$ for all $e, e' \in E$. Thus, for $R \in \mathscr{A}$ there are no trains moving simultaneously in opposite directions — in any specific moment of time trains move in a fixed direction, or they stop on auxiliary tracks.
2. Let a class $\mathscr{B}$ consist of all railway time-tables R from $\mathscr{R}$ with the property that there exist functions $m_1 : V \to \mathbb{N}^0$ and $m_2 : V \to \mathbb{N}^0$ such that $m_1(v) + m_2(v) = m(v)$ for all $v \in V$, and there are $m_1(v)$ auxiliary tracks on a station v for the direction $v_1 \mapsto v_n$ and $m_2(v)$ auxiliary tracks for the direction $v_n \mapsto v_1$. Thus, we divide all auxiliary tracks into two sets for both directions. The trains can only use auxiliary tracks corresponding to their directions, and the whole task can be decomposed into two independent subtasks for each direction.

The problem consists in the construction and implementation of a scheduling algorithm that sends as much as possible trains for a period of time T in both directions. More specifically, we are interested in a railway time-table at which the minimum number of trains in both directions over a specified period of time is maximal; we call this number per unit of time the *track capacity* of the section with the given railway time-table.

Some properties of time-tables from $\mathscr{A}$ and $\mathscr{A} \cap \mathscr{B}$ were studied in [1]. In the present paper, we continue our research. In addition, we perform numerical experiments that confirm our supposition about the class $\mathscr{B}$.

3 Assessment of the track capacity

We first consider the case where we have the one-way time-table R_0 on the graph Γ. This time-table can be obtained, for example, from the usual two-way time-table by removing time intervals that correspond to the direction $v_n \mapsto v_1$. Then there is motion only in one direction $v_1 \mapsto v_n$. Let us find the *maximal mean track capacity* for this case. It is clear that for an accumulation of trains on stations with their successive releasing, we can get a larger momentary track capacity. The quantity of the average track capacity is more useful for us.

In other words, we are interested in the largest track capacity which can be obtained during an arbitrary long time interval.

Let $f_{R_0}(t_j, v_i)$ be the number of trains that pass the station v_i in the semi-interval of time $[t_{j-\tau}, t_j)$, and

$$M(v_i) := \sum_{k=0}^{\tau} m(v_{i+k}).$$

Obviously,

$$f_{R_0}(t_j, v_i) \le M(v_{i-\tau}).$$

Hence, the number of trains that pass the station v_i in the semi-interval of time, of length t, does not exceed

$$\frac{t}{\tau} M(v_{i-\tau}).$$

If we suppose, that the section was empty at the beginning, then the maximal number of trains that pass the station v_n in the semi-interval of time, of length t, is not greater than the maximal number of trains that pass the station v_i in the semi-interval of time of the same length. Thus, the mean track capacity of the sections does not exceed

$$\frac{1}{\tau} M(v_{i-\tau})$$

and, hence,

$$\frac{1}{\tau} \min_i M(v_i).$$

Let us prove that the mean track capacity of the section equals $\frac{1}{\tau}\min_i M(v_i)$. In order to show that, let us construct a one-way time-table R_1, and find its track capacity. Define R_1 to be the time-table such that in each semi-interval of time $[t_{j-\tau}, t_j)$ the farthest trains are departed first. Thus defined time-tables are correct and unambiguous up to trains located on the same station, and they are indistinguishable for us.

Let $i' = \arg\min_i M(v_i)$, $M' = M(v_{i'})$ and

$$F(t_j, v_i) := \sum_{k=0}^{\tau} f(t_j, v_{i+k}).$$

Moving on by one station (beginning from $v_{i-\tau}$), it is easy to prove that $i-\tau \geq i'$ and $F(t_j, v_{i-\tau}) \geq M'$ imply $F(t_{j+1}, v_i) \geq M'$. In particular, the track capacity of the section for the time-table R_1 is greater or equal to M' on the time interval τ (beginning from some moment of time $t_{j'}$). Thus, we have the following lemma.

Lemma 1. The maximal mean track capacity equals $\frac{1}{\tau}\min_i M(v_i)$.

Let $R \in \mathscr{R}$. Then there are two one-way time-tables and two one-way motions. Let f_1 and f_2 be the maximal mean track capacities of these time-tables. Define

$$f_R := \min\{f_1, f_2\}$$

to be the maximal mean track capacity of the sections Γ for the time-table R.

Theorem 2. Let R be a time-table from $\mathscr{A} \cap \mathscr{B}$. Then the maximal mean track capacity f_R of the section for the time-table R equals

$$\frac{1}{2\tau} \min_{i \in \{1,2\}, j} \sum_{k=0}^{\tau} m_i(v_{j+k}).$$

Proof. Since $R \in \mathscr{A} \cap \mathscr{B}$, the time-table R consists of two independent one-way time-tables R_1 and R_2. Using lemma 1, we get

$$f_{R_i} = \frac{1}{2\tau} \min_j \sum_{k=0}^{\tau} m_i(v_{j+k}).$$

Hence,

$$f_R = \frac{1}{2\tau} \min_{i \in \{1,2\}, j} \sum_{k=0}^{\tau} m_i(v_{j+k}).$$

End of proof.

For $\mathscr{S} \subseteq \mathscr{R}$ define

$$f_{\mathscr{S}} := \sup_{R \in \mathscr{S}} f_R.$$

Thus, $f_{\mathscr{S}}$ is the limit maximal mean track capacity of the section for the time-tables from the set $\mathscr{S}$.

Corollary 3.

$$f_{\mathscr{A} \cap \mathscr{B}} = \frac{1}{4\tau} \min_j \sum_{k=0}^{\tau} m(v_{j+k}).$$

Proof. The assertion follows from the fact that if R is a time-table from $\mathscr{A} \cap \mathscr{B}$, then there exists a time-table R' from $\mathscr{A} \cap \mathscr{B}$ such that the track capacity of the section with the time-table R equals the track capacity of the section with the time-table R', and for R' we can assume

$$m_1(v_i) = m_2(v_i) = \frac{m(v_i)}{2}, \textit{ for even } m(v_i),$$

$$|m_1(v_i) - m_2(v_i)| \leq 1, \textit{ for odd } m(v_i).$$

End of proof.

Proposition 4. The maximal mean track capacity for the time-tables from $\mathscr{B}$ does not exceed the doubled maximal mean track capacity for the time-tables from $\mathscr{A} \cap \mathscr{B}$:

$$f_{\mathscr{B}} \leq \frac{1}{2\tau} \min_j \sum_{k=0}^{\tau} m(v_{j+k}) = 2 f_{\mathscr{A} \cap \mathscr{B}}.$$

Proof. The assertion follows from the estimates of the maximal mean track capacities for the section with one-way schedule.

Similarly,

Proposition 5.

$$f_{\mathscr{R}} \leq \frac{1}{\tau} \min_j \sum_{k=0}^{\tau} m(v_{j+k}) = 4 f_{\mathscr{A} \cap \mathscr{B}}.$$

We suppose that the following is correct:

Supposition 1. The track capacity for the section with a time-table from $\mathscr{B}$ does not exceed the track capacity of the section with some time-table from $\mathscr{A} \cap \mathscr{B}$:

$$f_{\mathscr{B}} \leq f_{\mathscr{A} \cap \mathscr{B}}.$$

Supposition 2. The track capacity for the section with a time-table from $\mathscr{R}$ does not exceed the track capacity of the section with some time-table from $\mathscr{A} \cap \mathscr{B}$:

$$f_{\mathscr{R}} \leq f_{\mathscr{A} \cap \mathscr{B}}.$$

4 Numerical experiments

Let us consider a mathematical model of the linear section of a single-track railway with 65 stations; two of them are isolated with $\delta_1 = 80$ and $\delta_2 = 135$ (an isolated station w with δ described in Introduction).

We estimate the value of $f_{\mathscr{A} \cap \mathscr{B}}$ for the above mentioned parameters of the model and for different τ.

Table 1: The dependence of $f_{\mathscr{A}\cap\mathscr{B}}$ value of τ

τ	$f_{\mathscr{A}\cap\mathscr{B}}$
1440	85
720	88
480	87
360	88
288	85
240	84
205	84
180	88
160	81
144	80
130	88
120	84
110	78
102	84
96	75
90	80
84	68
80	72
75	57
72	60
68	63
65	66
62	46

Algorithm

For τ from 60 to 720 minutes, and for arbitrary independent orientations of edges, let us find the maximal track capacity. We give below a verbal description and present a pseudocode of the algorithm (see algorithm 1).

For $t \in T$

1. if there is no train at the first station v_1 or at the last station v_n, we create it there;
2. iterate stations from last to first
   (a) for each station, do the following;
   (b) if $s_R(t, \{v_i, v_{i-1}\}) = -1$ and train will be able to move to the next station v_{i+1} then we send it there;
   (c) if there is a train on v_i then go to (a);
3. as we reach the first station v_1, we create and send new trains as much as possible;
4. iterate stations from first to last
   (a) for each station, do the following;
   (b) if $s_R(t, \{v_i, v_{i+1}\}) = 1$ and train will be able to move to the next station v_{i-1} then we send it there;
   (c) if there is a train on v_i then go to (a);
5. as we reach the last station v_n, we create and send new trains as much as possible;
6. increase the time counter and go to 2.

The algorithm was implemented in C++ (we used standard data structures, see [8], [9]) using Intel Xeon Phi with offload mode and MPI [10]). The data was distributed between nodes of supercomputers and calculated on Intel Xeon Phi. Each node processes its own predefined set of time intervals. The communication between nodes is minimal, therefore we have obtained almost linear speedup. When using this algorithm, the allocation of memory is 4 MB per stream, or 960 MB per 240 streams.

For our experiments we use $|T| = 14400$ (equal to 10 days), cluster with 6 nodes. A node configuration is shown in Table 2.

The algorthim uses class *Generator*, that defines the distribution of traffic accident trends for each plot in the interval τ. Further, for a given distribution, we build our schedule using the above described algorithm. After processing, we choose the best schedule among all possible ones.

Tasks for nodes are distributed with MPI in a way such that all cores of the processor are used. Thus, we create 24 threads with OpenMP for every node (2 processors with 12 cores per node).

We use Intel Xeon Phi with offload mode and create 240 threads with OpenMP.

Execution times, speedup and efficiency of the program for different configurations can be seen from

```
V_(i,t) ← []
V_(1,0) ← m_1(i,t)
V_(n,0) ← m_2(i,t)
for t in T do
    for i in (n..2) do
        if s_r(t, e_{i,i-1}) == -1 then
            while m_1(i,t) > 0 do
                train
                ← minNumberTrainOnStation(V_{i,t})
                remove train from V_{i,t}
                add train to V_{i-1,t+1}
            end
        end
    end
    while m_1(1,t+1) == 0 do
        create newTrain
        add newTrain to V_{1,t+1}
    end
    for i in (1..n-1) do
        if s_r(t, e_{i,i+1}) == 1 then
            while m_2(i,t) > 0 do
                train
                ← minNumberTrainOnStation(V_{i,t})
                remove train from V_{i,t}
                add train to V_{i+1,t+1}
            end
        end
    end
    while m_2(n,t+1) == 0 do
        create newTrain
        add newTrain to V_{n,t+1}
    end
end
```

Algorithm 1: Pseudocode of the algorithm

Table 2: Configuration for one node of the cluster

Processor	2 x Intel Xeon Processor E5-2620 6C 2.0GHz 15MB Cache 1333MHz 95W
RAM	4 x 8GB (PC3L-10600 CL9 ECC DDR3 1333MHz LP RDIMM)
Coprocessor	2 x Intel Xeon Phi 5110P
Hard disk	IBM 500GB 7.2K 6Gbps NL SATA 3.5” G2SS HDD
Network adapter	Emulex Dual Port 10GbE SFP+ Embedded VFA III for IBM System x
Network	10 Gb Ethernet
Commutator	IBM System Networking RackSwitch G8124E (Rear to Front)

Table 3 and Table 4. Table 3 describes speedup and efficiency of the program compared with 1 node with 12 threads. In this paper, the speedup is the ratio of the execution time for 1 node with 12 threads to the value of the Table 3. The efficiency is the ratio of speedup to number of devices (number of nodes or number of Intel Xeon Phi).

Table 3: Execution time of the program

Number of nodes and number of threads per node	**Times (minutes)**
1 node x 12 threads	296
2 node x 12 threads	168
6 node x 12 threads	61
1 node x 1 mic	249
1 node x 2 mic	133
2 node x 1 mic	149
2 node x 2 mic	71

Table 4: Speedup and efficiency of the program

Number of nodes and number of thread per node	**speedup**	**efficiency**
2 node x 12 threads	1.761	0.88
6 node x 12 threads	4.852	0.8
1 node x 2 mic	1.87	0.93
2 node x 1 mic	1.67	0.83
2 node x 2 mic	3.5	0.82

5 Conclusion

Estimates of the track capacity from $\mathscr{B}$ were obtained. The software, which implements the algorithm using MPI and Intel Xeon Phi coprocessor, was created. The numerical experiments were performed on the supercomputer with Intel Xeon Phi. Thus, we have obtained a numerical confirmation of Supposition 1. In future we are going to continue our research and check the correctness of Supposition 2.

References

[1] E. N. Akimova, D. N. Gainanov, O. A. Golubev, I. D. Kolmogortsev, A. V. Konygin, The Problem of Scheduling for the Linear Section of a Single-Track Railway, ICNAAM 2015. AIP Conference Proceedings, 1738.

[2] L. KePing, G. ZiYou , N. Bin, Cellular automaton model for railway traffic, Journal of Computational Physics, **209 (1)**, 179-19, 2005.

[3] S. Harrod, Optimal Scheduling of Mixed Speed Trains on a Single Track Line, http://citeseerx.ist.psu.edu/ viewdoc/summary?doi=10.1.1.85.5597

[4] A. Higgins, E. Kozan, L. Ferreira, Optimal Scheduling of Trains on a Single Line Track, Transpn. Res.-B., **30 (2)**, 147-161, 1996.

[5] E.R. Gafarov, A. Dolgui, A.A. Lazarev, Two-Station Single-Track Railway Scheduling Problem With Trains of Equal Speed, Computers & Industrial Engineering, **85**, 260-267, 2015.

[6] J. Harbering, A. Ranade, M.E. Schmidt, Single-Track Train Scheduling, In Proceedings of MISTA 2015.

[7] R. Diestel, Graph Theory, Springer-Verlag, New York, 2000.

[8] Th. Cormen, Ch. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2009.

[9] R. Sedgewick, K. Wayne, Algorithms, Addison-Wesley, 2011.

[10] W. Gropp, E. Lusk, R. Thakur, Using MPI-2 Advanced Features of the Message-Passing Interface, MIT Press, 1999.

Elena N. Akimova is Doctor of Physical and Mathematical Sciences, Professor of “Numerical Methods and Equations of Mathematical Physics” Department of Institute of Radioelectronics and Information Technologies of Ural Federal University named after first President of Russia B.N. Yeltsin. Leading research scientist at the Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences, Ekaterinburg, Russian Federation. Her research interests are in the areas of numerical methods, parallel algorithms, multiprocessor computing systems, ill-posed problems, inverse geophysical problems and applied mathematics.

Damir N. Gainanov Damir N. Gaynanov is the head of "Big Data and Data Analysis" Department of Institute of Radioelectronics and Information Technology of Ural Federal University named after first President of Russia B.N. Yeltsin. He has more than 10 year activities in Data Mining and Big Data concepts implementation in different industries. His current focus lies in Big Data and Data Analytics for metallurgy and transport: technological processes optimization in automated systems. Earlier he had worked in the Institute of Mathematics and Mechanics of the Ural Division of the Russian Academy of Sciences. His main scientific interests are concentrated in Data mining, optimization algorithms on graphs, image processing, and their industrial applications.

Oleg A. Golubev is master student at the Institute of Radioelectronics and Information Technologies of Ural Federal University named after first President of Russia B.N. Yeltsin. His research interests are: parallel algorithms, numerical methods, big data, data mining, machine learning, supercomputer technologies and applied mathematics. Also his research interests are high-perfomance computing, computer vision, GPUs, optimization problems, graph algorithms, multiprocessor computing systems, ill-posed problems, inverse geophysical problems.

Ilya D. Kolmogortsev is master student at the Institute of Radioelectronics and Information Technologies Ural Federal University named after first President of Russia B.N. Yeltsin. His main research interests are: numerical methods, big data, data mining, machine learning, parallel algorithms, supercomputer technologies and applied mathematics. Also his research interests are high-perfomance computing, computer vision, GPUs, optimization problems.

Anton V. Konygin is Research Associate at the Institute of Mathematics and Mechanics of the Ural Division of the Russian Academy of Sciences, and he is Associate Professor of "Big Data and Data Analysis" Department of Institute of Radioelectronics and Information Technology of Ural Federal University named after first President of Russia B.N. Yeltsin. His research interests are in the areas of graphs and groups, combinatorics, optimization, computer science and applied mathematics.