ALGORITHM AND SOFTWARE DEVELOPMENT TO ALLOCATE LOCOMOTIVES FOR TRANSPORTATION OF FREIGHT TRAINS

V.M. Azanov, Moscow Aviation Institute, Moscow, Russian Federation, azanov59@gmail.com,
M.V. Buyanov, Moscow Aviation Institute, Moscow, Russian Federation, byiandj@gmail.com,
D.N. Gaynanov, Ural Federal University, Ekaterinburg, Russian Federation, damir.gainanov@gmail.com,
S.V. Ivanov, Moscow Aviation Institute, Moscow, Russian Federation, sergeyivanov89@mail.ru

We suggest a mathematical model to allocate locomotives for transportation of freight trains. The aim of the optimization in this model is to minimize the number of locomotives used for the transportation of the trains by choosing routes of the trains and locomotives. It is supposed that the trains can be transported only at defined time intervals (so-called train paths); every locomotive has possible routes called railway hauls. We take into account the necessity of periodic maintenance. We use graph theory and integer optimization to formulate the problem. We suggest mathematical definitions of a railway haul, a train path, a train route, and a locomotive route. An heuristic search algorithm to find an approximate solution of the problem is suggested. The main idea of the algorithm is maximal usage of locomotives that started earlier than other ones. The algorithm contains three stages. A solution of the previous stage is improved at each following stage. We use transfers of the locomotives to improve the current solution. We describe software development to optimize the model. We solve the problem using the historical data of Moscow railway.

Keywords: graph theory; integer optimization; allocation of locomotives.

Introduction

The problem of transportation by railway has been researched in many works, e.g. [1–6]. In these works, the structure of freight trains transportation is described and different mathematical models of freight transportation are suggested. A literature review on this topic may be found in [4].

It should be noticed that currently transportation by railway is not effective in Russia. It is related with delays of the freight trains formation, delays of prepared freight trains due to the lack of locomotives, delays of freight trains on the way, etc. Only 30% of planned traffic is performed on time. Moreover, only 30% of locomotives from depots are exploited effectively. 70% of locomotives are not exploited at each moment. So, it is necessary to perform a set of works to improve the quality of traffic planning. This problem contains the formation of freight trains, formation of so-called train paths (i.e., time intervals when trains may depart), timetable of trains departure and allocation of the locomotives, transfers of the locomotives in case of their deficit at a departure station. One

of the important parts of this problem is the selection of an effective strategy to allocate the locomotives. For solution of this problem it is needed to develop a mathematical model of the locomotives allocation for the freight trains transportation, to perform a software implementation of algorithms and to explore a computer program on real data.

In this paper the further development of mathematical model [7] is described. Unlike [7], the necessity of periodic maintenance and train weights are taken into account. The description of the developed algorithm and the computer program of an approximate problem solution search is presented.

1. Main Definitions

Let a directed graph $G = (V, A)$ correspond to a part of a railway, where V is a set of vertices, A is a set of edges. The vertices of the graph G are the main stations (stations where freight trains are made up and locomotives are changed). The edges of the graph G are the ways connecting the main stations. Some main stations are also depots. Let us denote the corresponding subset of V by D.

Locomotives may move only by defined routes (so-called railway hauls), so we introduce the following definition.

Definition 1. *A sequence of vertices $v_1, \ldots, v_{I_P}$ of the graph G is a railway haul P if the following conditions hold:*
1. Each vertex v_i is connected with the previous vertex: $(v_{i-1}, v_i) \in A$, $i = \overline{2, I_P}$.
2. All edges connecting adjacent vertices are different.
3. The first vertex coincides with the last one, it is a depot, and differs from other vertices in the sequence: $v_1 = v_P \in D$, $v_i \neq v_1$ for $i = \overline{2, I_P - 1}$.

Also, we will consider subhauls and simple subhauls.

Definition 2. *Any subsequence of adjacent vertices $v_i, v_{i+1}, \ldots, v_j$ $(1 \leq i < j \leq I_P)$ belonging to the haul P is a subhaul of the haul. Any couple of vertices (v_{i-1}, v_i) belonging to the haul P is called a simple subhaul.*

Let L be the set of all locomotives from the considered depots. For any locomotive $l \in L$, a set of feasible railway hauls is determined.

Let us define the set $\mathcal{P}_l$ for any set $\overline{\mathcal{P}}_l$, $l \in L$. The set $\mathcal{P}_l$ contains all simple subhauls of hauls of the set $\overline{\mathcal{P}}_l$. Suppose that a weight function $w_l(\cdot) \colon \mathcal{P}_l \to \mathbb{R}$ takes each simple subhaul of the haul $\overline{\mathcal{P}}_l$ for any locomotive $l \in L$ to the maximum weight of a train, which can be carried by this locomotive. Consider the following equivalence relation on the set L of locomotives:

$$l_1 \sim l_2 \Leftrightarrow \mathcal{P}_{l_1} = \mathcal{P}_{l_2}, \ \forall p \in \mathcal{P}_{l_1} \ w_{l_1}(p) = w_{l_2}(p). \tag{1}$$

The corresponding set of equivalence classes will be denoted by $\mathfrak{L}$.

Definition 3. *The set $\mathfrak{L}$ of equivalence classes generated by (1) is called the set of applicable types of locomotives.*

Let S be a set of carried freight trains. Each train is described by its weight w^s, dispatch station v_0^s, receiving station v_f^s, readiness time t_0^s, time τ_f^s, until which the train has to arrive at the receiving station. It means that the tuple $(w^s, v_0^s, t_0^s, v_f^s, \tau_f^s)$ corresponds

to the train s. Notice that this tuple defines the plan of transportation.

Locomotives and trains can move by preassigned routes only in certain periods of time called train paths.

Definition 4. *A sequence of tuples* (v_1, t_1, v_2, τ_2), (v_2, t_2, v_3, τ_3), ..., $(v_{I_N-1}, t_{I_N-1}, v_{I_N}, \tau_{I_N})$, *is said to be a path N if the following conditions hold:*
1) $v_i \in V$, $i = \overline{1, I_N}$, $t_i \in \mathbb{R}$, $i = \overline{1, I_N - 1}$, $\tau_i \in \mathbb{R}$, $i = \overline{2, I_N}$;
2) $(v_i, v_{i+1}) \in A$, $i = \overline{1, I_N - 1}$;
3) $t_i < \tau_{i+1}$, $i = \overline{1, I_N - 1}$;
4) $\tau_i \leq t_i$, $i = \overline{2, I_N}$.

In this definition, t_i corresponds to departure time from the station v_i, τ_{i+1} corresponds to arrival time to the station v_{i+1}. The introduced conditions show restrictions related with the movement of locomotives. Conditions 1 and 2 require that the locomotives can move only by edges, condition 3 means that the departure time from the current station is earlier than the arrival time to the next station, condition 4 means that the arrival time to a station is earlier than the departure time from this station.

By analogy with subhauls and simple subhauls, let us introduce subpaths and simple subpaths.

Definition 5. *Any subsequence of adjacent tuples belonging the train path N is called a subpath. Any tuple $(v_i, t_i, v_{i+1}, \tau_{i+1})$, $i = \overline{1, N - 1}$, belonging to the train path N is called a simple subpath.*

Let a set $\overline{\mathcal{N}}$ of train paths be given. Let the unordered set of all simple subpaths belonging to each path $N \in \overline{\mathcal{N}}$ denote by $\mathcal{F}(N)$. We denote the set of all simple subpath belonging to the set $\overline{\mathcal{N}}$ by $\mathcal{N}$, i.e.

$$\mathcal{N} = \bigcup_{N \in \overline{\mathcal{N}}} \mathcal{F}(N).$$

Define a function $W(\cdot)$ with domain $2^L \times \mathcal{N}$, which is the Cartesian product of the set of all locomotive combinations and the set of all simple subpaths. The function value defines the maximal weight, which can be carried by the corresponding locomotives combination. It is obvious that $W(\{l\}, (v, t, v', \tau)) = w_l((v, v'))$ if $l \in L$, $(v, v') \in \mathcal{P}_l$. Also, the values $w_l((v, v'))$ coincide for all $l \in \mathcal{L}$, where $\mathcal{L} \in \mathfrak{L}$ is an applicable type of locomotives. The combination of locomotives is called a complex locomotive. The complex locomotives are used to carry heavy trains.

Since the locomotives move only by paths and hauls, we introduce the definition of a feasible route with respect to the set of hauls. In this definition we take into account that the locomotives have to undergo maintenance not later than in time T (48 hours) after the previous maintenance. The duration of maintenance is denoted by t_{TO} (8 hours). It is assumed that at the initial time each locomotive $l \in L$ is described by time interval τ_{TO}^l, which has passed since the previous maintenance. If a locomotive undergoes maintenance at the initial time then the value τ_{TO}^l is negative and is equal in magnitude to the time remaining until the finish of the maintenance.

Definition 6. *A subsequence of simple subpaths (v_1, t_1, v_2, τ_2), (v_2, t_2, v_3, τ_3), ..., $(v_{I_l-1}, t_{I_l-1}, v_{I_l}, \tau_{I_l})$ is called a feasible route M_l of the locomotive l with respect to the*

set of hauls $\overline{\mathcal{P}_l}$ if the following conditions hold:
  1) $\tau_i \leq t_i$, $i = \overline{2, I_l - 1}$;
  2) $(v_i, v_{i+1}) \in \mathcal{P}_l$, $i = \overline{1, I_l - 1}$;
  3) there exists an increasing subsequence $i_1, \ldots, i_{f_l}$ of numbers from the set $\{2, 3, \ldots, I_l\}$ such that

$$\tau_{TO}^l + \tau_{i_1} \leq T, \tag{2}$$

$$t_{i_j} - \tau_{i_j} \geq t_{TO}, \quad j = \overline{1, f_l - 1}, \tag{3}$$

$$\tau_{i_j} - t_{i_{j-1}} \leq T, \quad j = \overline{2, f_l}, \tag{4}$$

$$\tau_{I_l} - t_{f_l} \leq T, \text{ if } f_l \neq I_l. \tag{5}$$

Condition 1 requires that the arrival time to a station is earlier than the departure time from this station. Condition 2 restricts a feasible transportation by the hauls. Condition 3 requires maintenances in certain time interval. The sequence $t_{i_1}, \ldots, t_{i_{f_l}}$ corresponds to the times of the maintenances start. According to (2) the time of the first maintenance start is not later than time interval T since the previous maintenance. According to (3) the maintenance cannot be less than t_{TO}. It follows from (4) that time interval between the departure after the maintenance and the next maintenance cannot be more than T. According to (5) the departure time after the last maintenance must be later than the time T before the arrival to the route termination.

Note that the route is a spatiotemporal concept. Let us denote the set of feasible routes of the locomotive l by $\mathcal{M}_l$. The initial and terminal stations of the route M_l are denoted by $v_0(M_l)$ and $v_f(M_l)$ respectively. The time of the first path beginning in the route is denoted by $t_0(M_l)$. The arrival time to the terminal station is denoted by $\tau_f(M_l)$.

Let us introduce the definition of a train feasible route. This concept is also spatiotemporal.

Definition 7. *A sequence of simple subpaths (v_1, t_1, v_2, τ_2), (v_2, t_2, v_3, τ_3), $\ldots$, $(v_{I_s-1}, t_{I_s-1}, v_{I_s}, \tau_{I_s})$ is called a feasible route of the train $s \in S$ and is denoted by R_s if the following conditions hold:*
  1) $v_1 = v_0^s$;
  2) $v_{I_s} = v_f^s$;
  3) $\tau_i \leq t_i$, $i = \overline{2, I_s - 1}$;
  4) $t_0^s \leq t_1$;
  5) $\tau_f^s \geq \tau_{I_s}$.

Conditions 1 and 2 define the initial and terminal stations of the route, condition 3 restricts departure and arrival time, conditions 4 and 5 require to carry out all needed transportation on time.

The set of feasible routes of the train s is denoted by $\mathcal{R}_s$.

Let us define the set $\mathcal{F}(M_l)$ of all simple subpaths belonging to the route M_l for all locomotives l, $l \in L$,. Also, we define the set $\mathcal{F}(R_s)$, $s \in S$, of all simple subpaths belonging to the route R_s of the train s.

For all simple subpaths $n \in \mathcal{N}$ and all sets $M = \{M_l\}_{l \in L}$ of routes let us define the set $\pi_n(M)$. This set contains all locomotives that move by the simple subpath n if the set M is fixed:

$$l \in \pi_n(M) \Leftrightarrow n \in \mathcal{F}(M_l). \tag{6}$$

76

Bulletin of the South Ural State University. Ser. Mathematical Modelling, Programming & Computer Software (Bulletin SUSU MMCS), 2016, vol. 9, no. 4, pp. 73–85

2. Statement

Let us consider the part of the railway, which corresponds to the graph $G = (V, A)$ defined above. Suppose that L is a set of locomotives, S is a set of trains, $\overline{\mathcal{N}}$ is a set of paths. Let $\mathcal{N}$ be the corresponding set of simple subpaths. Let a weight function $W(\cdot)$ of complex locomotives be given. For all locomotives $l \in L$ a set $\overline{\mathcal{P}}_l$ of hauls is defined. Let $\mathcal{P}_l$ be the set of simple subhauls.

At the initial time some locomotives can move, so it is supposed that a locomotive $l \in L$ can depart only from a fixed station v_0^l after the time t_0^l.

Let the time τ_{TO}^l since the previous maintenance be given for all locomotives $l \in L$. The set of feasible routes $\mathcal{M}_l$ is given according to definition 6 taking into account the time τ_{TO}^l for all locomotives $l \in L$. Similarly, for all trains $s \in S$, the set of feasible routes $\mathcal{R}_s$ is given. Let $|L|$ be the number of locomotives from the set L with non-empty route.

Let the set of paths $\overline{\mathcal{N}_s} \subset \overline{\mathcal{N}}$ that can be used to carry the train be given for all trains $s \in S$. The set of all corresponding simple subpaths is denoted by $\mathcal{N}_s$. These constraints are needed because some paths can be used only for certain transportation.

Let $M = \{M_l\}_{l \in L}$ be a selected set of routes for all locomotives, $R = \{R_s\}_{s \in S}$ be a selected set of routes for all trains, $\mathcal{M} = \{\mathcal{M}_l\}_{l \in L}$ be a set of feasible routes for all locomotives, $\mathcal{R} = \{\mathcal{R}_s\}_{s \in S}$ be a set of feasible routes for all trains.

It is required to select a set M of the locomotive routes and a set R of the train routes such that the number $|L|$ of the exploited locomotives is minimal and all train routes are covered by locomotive routes. It means that the following problem should be solved:

$$|L| \to \min_{M \in \mathcal{M}, R \in \mathcal{R}} \tag{7}$$

subject to

$$M_l \in \mathcal{M}_l, \ l \in L, \tag{8}$$

$$R_s \in \mathcal{R}_s, \ s \in S, \tag{9}$$

$$\bigcup_{s \in S} \mathcal{F}(R_s) \subset \bigcup_{l \in L} \mathcal{F}(M_l), \tag{10}$$

$$\mathcal{F}(R_s) \cap \mathcal{F}(R_{s'}) = \emptyset, \ s \neq s', \ s, s' \in S, \tag{11}$$

$$W(\pi_n(M), n) \geq w^s, \ n \in \mathcal{F}(R_s), \ s \in S, \tag{12}$$

$$\mathcal{F}(M_l) \subset \mathcal{N}, \ l \in L, \tag{13}$$

$$\mathcal{F}(R_s) \subset \mathcal{N}_s, \ s \in S, \tag{14}$$

$$v_0(M_l) = v_0^l, \tag{15}$$

$$t_0(M_l) \geq t_0^l. \tag{16}$$

Conditions (8) and (9) mean that we consider only the feasible locomotive routes and train routes. In particular, it means that feasible hauls exist for the feasible routes. Also, notice that the feasible route requires to carry out the plan on time.

Condition (13) requires that the train routes contain only simple subpaths, because the set $\bigcup_{l \in L} \mathcal{F}(M_l) \subset \mathcal{N}$ contains simple subpaths that belong to some route of locomotive. Condition (14) defines the similar requirement for train routes, moreover, it restricts the selection of feasible paths to carry a train. Condition (10) means that all simple subpaths

belonging to the route of a train are used for a locomotive, i.e., all trains are carried by the locomotives. Also, it follows from this condition that the locomotives can move by simple subpaths that are not used by trains. Therefore, each used path corresponds to a train with one or several locomotives or to a locomotive without train.

Condition (11) means that the routes of trains cannot intersect, i.e., one simple subpaths cannot be used for two trains. Since the locomotives can move together or with a train (so-called auxiliary run), the similar condition is not used for the locomotives.

Condition (12) sets weight norms of complex locomotives, i.e., the complex locomotive $\pi_n(M)$ exploited at a path $n \in \mathcal{F}(R_s)$ and carrying the train s has to be able to carry a train with weight $W(\pi_n(M), n)$ not less than w^s.

Conditions (15) and (16) define an initial state of the locomotives.

Notice that the set S of trains and the set $\overline{\mathcal{N}}$ of paths are defined by the daily transportation plan and by the number of days of the general plan.

In the formulated problem, routes of the locomotives and trains should be selected. Although practically the paths are given for the certain trains. So below we suppose that the set of feasible routes $\mathcal{R}_s$ of the train $s \in S$ contains only one route. Therefore, the problem is reduced to a problem of locomotive allocation to carry trains with certain routes.

3. Algorithm of Locomotives Allocation

Formulated problem (7) is a combinatorial problem of high dimension. It is difficult to find the exact problem solution. Therefore, we propose an heuristic algorithm for an approximate solution. We assume that the routes of all trains $s \in S$ are defined, i.e., a path to move is defined for each train. The algorithm consists of three stages: the construction of basic solution and two stages for the advancement of the solution obtained in the preceding stage. The algorithm is based on the idea of maximal usage of the locomotive with the minimum initial time of movement. That enables uniform utilization of the locomotives taking into account the maintenance. The locomotives move to the maintenance according to the using queue. The basic solution involves an unlimited number of locomotives at each station at the initial time. As a result the implementation of the transport plan gives great value $|L|$. To improve the basic solution at the second stage of the algorithm we use the transfers of locomotives finished their movement. As shown by the results of calculations, this transfer can improve the solution significantly. If the results are unsatisfactory, then we can use the third stage of the algorithm, which runs more complex transfers described below. For simplicity, in the description of the algorithms we do not consider limitations on the weight for the transported trains. In case of these restrictions we need to search complex locomotives.

3.1. Construction of a Basic Solution

Let a non-empty set $S = \{s_i \mid i = \overline{1, |S|}\}$ of trains with non-empty routes be given. Let $v_f(l)$, $\tau(l)$ be the terminal station of the route M_l (or initial station in case of an empty route) and the arrival time of the locomotive $l \in L$ at the station. The $\oplus$ denotes the concatenation operation of two routes of locomotives. The result of the concatenation of the two routes is a new route, obtained by attaching simple subpaths included in the second route, to the first one. The operation of concatenation is well-defined only for the

case when the second route starts at the same station as the first route ends.

Algorithm 1 (Construction of the basic solution)

1. Let $i := 1$, $j := 1$, $M_l := \emptyset, l \in L$.

2. If $j \leq I_s$, then fix the subpath j $n_j = (v_0(n_j), t(n_j), v_f(n_j), \tau(n_j))$ of the route of the train s_i. Otherwise $j := 1$, $i := i + 1$; if $i \leq |S|$, repeat step 2, otherwise finish the algorithm.

3. Find the locomotive $l \in L$ such that $v_0(n_j) = v_f(l)$, $\tau(l) \leq t(n)$, $M_l \oplus n_j \in \mathcal{M}_l$. If the locomotive is found, add simple subpath n_j of the route M_l: $M_l := M_l \oplus n_j$, $j := j + 1$ and go to step 2. If the locomotive is not found, then go to step 4.

4. Take the new locomotive $l \in L$ from the depot at the station v_0^s, $j := j + 1$, $M_l := \{n_j\}$, then go to step 2.

In this and all subsequent algorithms the priority is given to the locomotive with the earliest initial time of the movement in the case when several locomotives can be chosen.

3.2. Searching of the Complex Path

For a locomotive transfer we must find the path that connects station at which the locomotive has stopped and the station from which the train is departed. Let $t(n)$ be the initial time of the movement of simple subpath n, $\tau(n)$ be the terminal time of the simple subpath n. $\mathcal{N}_a$ denotes the set of simple subpaths corresponding to the edge $a \in A$. To each edge of the graph G the weight characteristics equal to the average time for its movement is assigned:

$$w_a = \frac{1}{|\mathcal{N}_a|} \sum_{n \in \mathcal{N}_a} (\tau(n) - t(n)), \tag{17}$$

where $|\mathcal{N}_a|$ denotes the cardinality of the set $\mathcal{N}_a$. Let $n^*(v_0, t, v_f, \tau)$ be the complex path, passing along the shortest path, joining station v_0 and v_f, with initial time later than t and terminal time not later than τ. To find the shortest paths between vertices of the weighted directed graph we can use, for example, the Floyd – Warshall algorithm [8].

3.3. Transfer 1

Suppose that as a result of the basic solution searching we have found a set $\tilde{L} = \{l_i \mid i = \overline{1, \tilde{l}}\}$ of locomotives with non-empty routes. Let $\tau(M_l)$ be the arrival time of the locomotive l, $v_f(M_l)$ be the terminal station of the route M_l.

Algorithm 2 (transfer 1)

1. Let $i := 1$.

2. If $i \geq \tilde{l}$, then finish the algorithm. Otherwise, $j := i + 1$ and go to step 3.

3. If $j > \tilde{l}$, then $i := i + 1$ and go to step 2. Otherwise, if $l_i \sim l_j$ and $\tau(M_{l_i}) < t_0(M_{l_j})$, then go to step 4; otherwise, $j := j + 1$ and repeat step 3.

4. Search a path $\theta = n^*(v_f(M_{l_i}), \tau(M_{l_i}), v_0(M_{l_j}), t_0(M_{l_j}))$, satisfying the condition $M_{l_i} \oplus \theta \oplus M_{l_j} \in \mathcal{M}_{l_i}$. If such path is found, go to step 5; otherwise $j := j + 1$ and go to step 3.

5. Assign $M_{l_i} := M_{l_i} \oplus \theta \oplus M_{l_j}$, $\tilde{L} := \tilde{L} \setminus \{l_j\}$, $J := j + 1$ and go to step 3.

The purpose of this algorithm is to find pairs of locomotives l_i and l_j such that all the locomotive l_j movements can be given to the locomotive l_i. Herewith, the trip of the locomotive with more ordinal number is given to the locomotive with the lower number.

3.4. Transfer 2

During application of the algorithm 2 locomotives are replaced by each other only if the start of the second locomotive movement is later than the end of the first locomotive movement. If the locomotive has a big break in the movement, then Algorithm 2 does not provide the transfer of locomotive trips. To give these trips to other locomotives, we offer the following algorithm. Suppose that we received a set $\bar{L} = \{l_i\}_{i=1}^{\bar{l}}$ of locomotives with non-empty routes as a result of applying algorithm 2.

Algorithm 3 (transfer 2)

1. Let $j := 2$.

2. If $j \leq \bar{l}$, let $i := 1$, $N^* := \emptyset$, $L^* := \emptyset$ and proceed to step 3. Otherwise, complete the algorithm.

3. If $\mathcal{F}(M_{l_j}) \subset N^*$, proceed to step 5. If $i \geq j$, let $j := j + 1$ and go to step 2, otherwise proceed to step 4.

4. If $l_i \sim l_j$ and there is a couple of simple subpaths $n_q^1 \in \mathcal{F}(M_{l_i})$, $N_r^2 \in \mathcal{F}(M_{l_j}) \setminus N^*$, satisfying the following conditions:

$$\tau(n_q^1) < t(n_r^2), \qquad t(n_r^2) < t(n_{q+1}^1); \tag{18}$$

$$\exists\, \theta_1 = n^*(v_f(n_q^1), \tau(n_q^1), v_0(n_r^2), t(n_r^2)); \tag{19}$$

$$\exists\, \theta_2 = n^*(v_f(n_r^2), \tau(n_r^2), v_0(n_{q+1}^1), t(n_{q+1}^1)); \tag{20}$$

$$(N_1^1, \ldots, n_q^1, \theta_1, n_r^2, \theta_2, n_{q+1}^1, \ldots, n_{I_{l_i}}^1) \in \mathcal{M}_{l_i}, \tag{21}$$

then $L^* := L^* \cup \{l_i\}$, $N^* := N^* \cup \{\theta_1\} \cup \{n_r^2\} \cup \{\theta_2\}$ and repeat step 4. Otherwise, $i := i + 1$ and go to step 3.

5. Allocate locomotives from the set L^* to the corresponding simple subpath from the set N^*, $\bar{L} := \bar{L} \setminus \{l_j\}$, $j := j + 1$ and go to step 2.

Conditions (18)–(21) mean that the locomotive l_i can make one trip by a path n_r^2 of l_j locomotive.

4. Software Implementation

C++ programming language was chosen to develop the software system to allocate locomotives for transportation of freight trains. Microsoft Visual C++ 12.0 was chosen as a development environment. This environment supports C++ 11 standard, which provides many powerful features actively used for this software. Initial data for the program is provided in CSV and XLSX formats. COM technology is used to read and write data with the help of Excel 2010 program from the Microsoft Office 2010. Component Object Model (COM) is a binary-interface standard for software components introduced by Microsoft. It is used to enable inter-process communication and dynamic object creation. Microsoft Office provides usage of COM technology, so it allows us to use all the features of this package to develop other programs. The program system uses COM to interact with Excel to operate with CSV and XLSX files.

The program system consists of 5 classes.

1. "Depot" class stores information about depot stations and adjacent hauls. Class methods provide correct locomotives movements.

2. "Station" class stores information about all main stations of the railway net. It is used to form the graph, presented in the form of an adjacency list. Map structure is used

to present this list. The first element of the map is a station, the second element is an adjacent stations vector.

3. "Locomotive" class stores information about locomotives (order number, tonnage, time of maintenance, route, etc.). This class is used to allocate locomotives to trains and for locomotive movements. Class methods form movement statuses on simple subpaths, namely point if the locomotive is performing with a train on the selected simple subpath or without train, or in a complex locomotive. The selected statuses with locomotives numbers are recorded in the appropriate "Path" class properties. There are also methods, which provide movement specifications calculations such as with train, without train, in complex locomotive.

4. "Train" class stores information about trains (train order number, freight weight, estimated freight delivery time). It is used to form the train route, so that to find a path, on which that is carried.

5. "Path" stores information about all simple subpaths (time, departure station, destination station, route matching appropriate subroute, information about locomotives movement by this subpath).

The software system consists of four parts:

1) initial data reading;

2) initial data processing;

3) algorithm execution;

4) result output.

Initial data is presented in the form of tables except for the railway graph which is presented as an adjacency list.

1. File containing information about stations which take part in the traffic, such as station type and number, depot presence and type.

2. File containing information about web topology presented by a graph in the form of adjacency list.

3. File containing information about paths.

4. File containing daily movements plan.

The initial data processing consists of:

1) graph creation;

2) simple subpaths set formation from the initial paths set by the help of division;

3) simple subpaths cauterization on graph edges and weight assignment;

4) search the shortest paths through all graph vertices using the Floyd – Warshall algorithm;

5) trains allocation to paths using algorithms 1–3.

There are many output formats for results visualization. All of them are tables stored in XLSX format. The main output format is a list of all train routes that is the sequence of all simple subpaths to be used by every locomotive. The visualization is performed on the last step of the program execution.

"Locomotives usage" table is formed for the quality assessment. This table provides efficiency evaluation of locomotive usage. This table is a lower triangular matrix. Let us describe the table structure. Table rows and columns corresponds to calendar days. The value of each cell with index (i, j) is formed from locomotives quantity used during the i-th day and departed from a depot during the j-th day.

The software system is designed to perform a full cycle of calculations for each set

of initial data. This means that every change in initial data requires a new launch of the system.

5. Numerical Experiment Results

Numerical experiment was executed with the help of data provided by Moscow Railway. The data was gathered from 20 Sep. 2015 until 1 Oct. 2015. Initial data characteristics are presented in Table 1.

Table 1

Initial data

The number of stations	40
The number of depot-stations	16
The number of marshalling yards	16
The number of trains in daily movement plan	592
The number of paths per day	1254
Plan period (days)	10

All calculations were made considering both restrictions on maintenance and without them. In case of considering restrictions it is assumed that locomotives should undergo 8-hour maintenance at least once every 48 hours. Software system execution time depends on the initial data. For example, the decrease of plan period will increase the execution time significantly. Any change in the trains quantity will not affect the execution time. For initial data presented above the execution time is about 10 minutes. The results are provided in Table 2.

Table 2

The solution of the problem

Algorithm	Number of locomotives (with maintenance)	Number of locomotives (without maintenance)
Algorithm 1	2700	2989
Algorithm 2	500	581
Algorithm 3	297	369

Table 3 "Locomotives usage" was made for the considering maintenance restrictions case. There are 11 rows in Table 3 because the first and the last days are not complete (12 hour period was considered). It is not difficult to see that the matrix track is the value of the objective function. Let us mention that new locomotives are not being put into operation at the end of the period. This indicates the stabilization process which means that the value of the objective function is close to the overall locomotive fleet count. Let us compare the solution with the historical data of locomotives usage by Moscow Railway. Overall locomotive Moscow Railway fleet is about 900 locomotives. Every day 700 of them

Table 3

Locomotives usage (with maintenance)

Number of trains	Number of locomotives exploited during the i-th day and taken in the j-th day										Total
299	201	0	0	0	0	0	0	0	0	0	201
592	198	83	0	0	0	0	0	0	0	0	281
592	201	78	48	0	0	0	0	0	0	0	327
592	197	83	34	9	0	0	0	0	0	0	323
592	193	69	40	7	14	0	0	0	0	0	323
592	199	81	34	9	10	7	0	0	0	0	340
592	181	69	33	9	14	5	1	0	0	0	312
592	195	79	36	9	11	5	1	2	0	0	338
592	183	71	34	7	14	4	1	0	0	0	314
592	187	75	33	6	13	5	1	0	0	4	324
293	139	47	17	4	9	3	0	0	0	1	220

are used. Thus we can make a conclusion that our solution is two times better than the current one.

Conclusion

In this work the mathematical model to allocate locomotives for transportation of freight trains is suggested, the algorithm to solve the problem and its software implementation are developed. The result of numerical experiments has shown that the suggested method allows us to reduce the number of exploited locomotives about twice. However, we should notice that in practice this result is difficult to obtain, because in the considered example a lot of constrains are not taken into account, particularly, we consider only one type of maintenance, we do not consider possible random deviations from the timetable. But the obtained result allow us to hope that the found solution will provide effective locomotives usage.

References

1. Belyy O.V., Kokurin I.M. [Organization of Rail Freight: Ways Optimization]. *Transport of the Russian Federation*, 2011, no. 4 (35), pp. 28–30. (in Russian)

2. Kibzun A.I., Naumov A.V., Ivanov S.V. [Bilevel Optimization Problem for Railway Transport Hub Planning]. *Large-Scale Systems Control*, 2012, no. 38, pp. 140–160. (in Russian)

3. Lazarev A.A., Musatova E.G. The Problem of Trains Formation and Scheduling: Integer Statements. *Automation and Remote Control*, 2013, vol. 74, issue 12, pp. 2064–2068.

4. Lazarev A.A., Musatova E.G., Gafarov E.R., Kvaratskheliya A.G. *Teoriya raspisaniy. Zadachi zheleznodorozhnogo planirovaniya* [Sheduling. Problems of Rail Planning]. Moscow, Institute of Control Science of Russian Academy of Science, 2012. 91 p. (in Russian)

5. Tereshina N.P., Galaburda V.G., Trikhunkov et al. *Ekonomika zheleznodorozhnogo transporta* [Economics of Railway Transport]. Moscow, UMTs ZhDT, 2006. 801 p. (in Russian)

6. Shapkin I.N. *Organizatsiya zheleznodorozhnykh perevozok na osnove informatsionnykh tekhnologiy* [Organization of Rail Transport on the Basis of Information Technologies]. Moscow, UMTs ZhDT, 2011. 320 p. (in Russian)

7. Gainanov D.N., Ivanov S.V., Kibzun A.I., Osokin A.V. Model of Optimal Allocation Locomotives to Rail Freight. *Intellektual'nyye sistemy upravleniya na zheleznodorozhnov transporte–2015* [Intelligent control systems in railway transport–2015]. Moscow, NIIAS, 2015, pp. 45–47. (in Russian)

8. Floyd R.W. Algorithm 97 – Shortes Path. *Communications of the ACM*, 1962, vol. 5, no. 6, p. 345.

АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ НАЗНАЧЕНИЯ ЛОКОМОТИВОВ С ЦЕЛЬЮ ПЕРЕВОЗКИ ГРУЗОВЫХ СОСТАВОВ

В.М. Азанов, М.В. Буянов, Д.Н. Гайнанов, С.В. Иванов

Предлагается математическая модель назначения локомотивов для перевозки грузовых составов. Целью оптимизации в модели является минимизация числа задействованных для перевозки составов локомотивов за счёт выбора маршрутов составов и локомотивов. Предполагается, что составы могут перевозиться в определённые промежутки времени, которые называются нитками, для каждого локомотива установлены допустимые варианты передвижения, называемые плечами. Учитывается необходимость периодического проведения технического осмотра локомотивов. Для формулировки задачи используется аппарат теории графов и целочисленной оптимизации. Формализуются железнодорожные понятия плеча, нитки, рейса состава и маршрута оборота локомотива. Предлагается эвристический алгоритм поиска приближённого решения задачи. Основной идеей предлагаемого алгоритма является максимальное использование локомотивов с ранним началом движения. Алгоритм состоит из трех этапов, на каждом из которых улучшается предыдущее найденное решение. Улучшение решения осуществляется за счет перегонок локомотивов. Приводится описание разработанных программных средств для оптимизации данной модели. Проведены численные эксперименты на примере исторических данных Московской железной дороги.

Ключевые слова: теория графов; целочисленная оптимизация; назначение локомотивов.

Литература

1. Белый, О.В. Организация грузовых железнодорожных перевозок: пути оптимизации / О.В. Белый, И.М. Кокурин // Транспорт Российской Федерации. – 2011. – № 4 (35). – С. 28–30.

2. Кибзун, А.И. Двухуровневая задача оптимизации деятельности железнодорожного транспортного узла / А.И. Кибзун, А.В. Наумов, С.В. Иванов // Управление большими системами. – 2012. – № 38. – С. 140–160.

3. Лазарев, А.А. Целочисленные постановки задачи формирования железнодорожных составов и расписания их движения / А.А. Лазарев, Е.Г. Мусатова // Управление большими системами. – 2012. – № 38. – С. 161–169.

4. Теория расписаний. Задачи железнодорожного планирования / А.А. Лазарев, Е.Г. Мусатова, Е.Р. Гафаров, А.Г. Кварацхелия. – М.: ИПУ РАН, 2012.

5. Экономика железнодорожного транспорта / Н.П. Терешина, В.Г. Галабурда, М.Ф. Трихунков и др. – М.: УМЦ ЖДТ, 2006.

6. Шапкин, И.Н. Организация железнодорожных перевозок на основе информационных технологий / И.Н. Шапкин. – М.: Учебно-методический центр по образованию на железнодорожном транспорте, 2011.

7. Модель оптимального назначения локомотивов при формировании грузовых составов / Д.Н. Гайнанов, С.В. Иванов, А.И. Кибзун, А.В. Осокин // Труды четвертой научно-технической конференции «Интеллектуальные системы управления на железнодорожном транспорте». – М.: НИИАС, 2015. – С. 45–47.

8. Floyd, R.W. Algorithm 97 – Shortes Path / R.W. Floyd // Communications of the ACM. – 1962. – V. 5, № 6. – P. 345.

Валентин Михайлович Азанов, студент, кафедра «Теория вероятностей», Московский авиационный институт (национальный исследовательский университет) (г. Москва, Российская Федерация), azanov59@gmail.com.

Михаил Владимирович Буянов, студент, кафедра «Теория вероятностей», Московский авиационный институт (национальный исследовательский университет) (г. Москва, Российская Федерация), byiandj@gmail.com.

Дамир Насибуллович Гайнанов, заведующий кафедрой «Аналитика больших данных и методов видеоанализа», Уральский федеральный университет имени первого Президента России Б.Н. Ельцина (г. Екатеринбург, Российская Федерация), damir.gainanov@gmail.com.

Сергей Валерьевич Иванов, кандидат физико-математических наук, доцент, кафедра «Теория вероятностей», Московский авиационный институт (национальный исследовательский университет) (г. Москва, Российская Федерация), sergeyivanov89@mail.ru.