A Problem of Scheduling Operations at a Locomotive Maintenance Depot

A. A. Lazarev[1,2,3,4], E. G. Musatova[1], E. M. Grishin[1,2(✉)], G. V. Tarasov[1,2], S. A. Galakhov[1,2], and N. A. Pravdivets[1]

[1] V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Profsoyuznaya str. 65, Moscow, Russia
jobmath@mail.ru, grishin.em16@physics.msu.ru, pravdivets@ipu.ru
[2] Lomonosov Moscow State University, Leninskie Gory str. 1/2, Moscow, Russia
[3] National Research University Higher School of Economics, Myasnitskaya str. 20, Moscow, Russia
[4] Moscow Institute of Physics and Technology (State University), Institute lane 9, Dolgoprudny, Moscow Region, Russia
http://www.orsot.ru/index.php/en/

Abstract. In this article, we consider the problem of planning maintenance operations at a locomotive maintenance depot. There are three types of tracks at the depot: buffer tracks, access tracks and service tracks. A depot consists of up to one buffer track and a number of access tracks, each of them ending with one service track. Each of these tracks has a limited capacity measured in locomotive sections. We present a constraint programming model and a greedy algorithm for solving the problem of planning maintenance operations. Using lifelike data based on the operation of several locomotive maintenance depots in Eastern polygon of Russian Railways, we carry out numerical experiments to compare the presented approaches.

Keywords: Maintenance · Scheduling · Dynamic programming · Constraint programming · Heuristic

1 Introduction

Railway scheduling is an entangled process of managing a large number of objects, including railway infrastructure, rolling stock, etc. This process requires to take into account a lot of conditions and restrictions. Foreground goals of the planning are safety and security of the whole system and minimizing transporting delays. According to Russian Railways safety requirements, every locomotive should undergo regular maintenance. In particular, there is a kind of maintenance that is carried out every several days and includes inspection of chassis, brake system, traction motors, auxiliary equipment, transformers and

This work was supported by Russian Railways and the Russian Foundation for Basic Research, project No. 17-20-01107.

M. Jaćimović et al. (Eds.): OPTIMA 2019, CCIS 1145, pp. 311–325, 2020.
https://doi.org/10.1007/978-3-030-38603-0_23

electric systems. The maintenance is carried out at special facilities, namely, Locomotive Maintenance Depots, or LMDs. Up to 100 locomotives are serviced at each LMD per day. The limitations on resources such as availability of maintenance crews, space on tracks, available special equipment, etc. result in growing downtimes. In order to reduce downtimes and thus increase efficiency in utilizing the locomotive fleet, scheduling process must be implemented.

In this paper we are going to find a solution for the described problem of scheduling locomotives maintenance. The paper is organized as follows. Section 2 includes a verbal statement of the problem, basic terms and notation, and a review of the literature on the subject. Section 3 is devoted to a constraint programming model. Section 4 proposes a heuristic algorithm. The results of numerical experiments on real data are presented and analyzed in Sect. 5.

2 Problem Statement

We consider a real-world problem of planning operation of locomotive maintenance depots of Russian Railways. There are three types of tracks at LMDs: buffer tracks, access tracks and service tracks. A typical LMD consists of up to one buffer track and a number of access tracks, each of them ending with one service track (see Fig. 1). Each of these tracks has a limited capacity measured in locomotive sections.

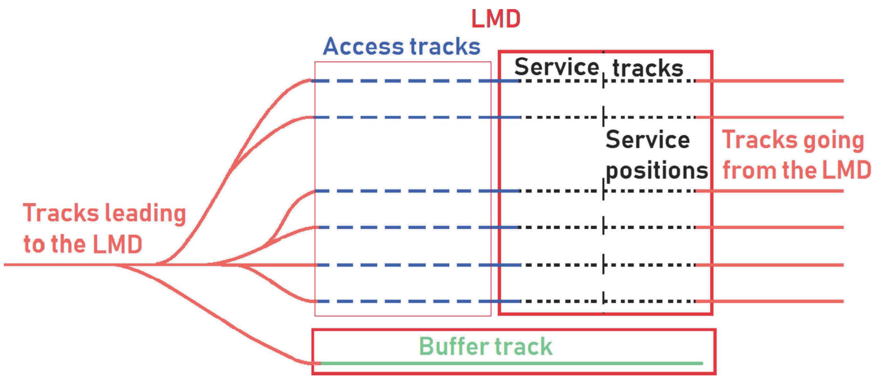

Fig. 1. Typical scheme of an LMD

Let $T = [T_0, T_1]$ be the planning horizon – the time interval during which all locomotives arrive to the LMD. We are provided a twenty-four hours plan of locomotives arriving to the LMD. All the time moments are considered to be integer. Let L be the set of all locomotives and $n = |L|$ be their number. The locomotives differ in models, number of sections and service time.

For each locomotive $l \in L$, let us define the following parameters.

- u_l is the number of sections of which l consists, $u_l \leq 4$.
- r_l is the moment of arrival of l to the LMD, $r_l \neq r_{l-1}, \forall l \in L$.
- τ_l is service time, or maintenance duration of l. Typically, the service time is from 57 to 120 min. For any locomotive, its service time is the same for any service track.
- v_l is the model number of l. Each service track can only maintain a certain set of locomotive models.

It is necessary to take into account all the limitations associated with the technical features of the locomotives and an LMD to build a feasible schedule.

Upon arriving at an LMD, each locomotive can be placed onto a buffer track. Let b be the buffer capacity, namely the number of locomotive sections that the buffer can contain at the same time. If an LMD does not include a buffer, the buffer capacity is considered to be zero. The buffer track operates according to the LIFO scheme: before a locomotive can leave the buffer, all the locomotives that came after it should leave first.

Upon leaving the buffer, each locomotive should be placed onto one of the access tracks, that are accessible from the buffer. If locomotive l was not placed into the buffer, it should be placed on an access track right at moment r_l of its arrival at the LMD. Let A be a set of all access tracks of the regarded LMD. For each access track $a \in A$, while let c_a denote the maximal number of locomotive sections that access track a can simultaneously accommodate.

Each of the access tracks ends with a single service track, and each service track can only be accessed from a single access track. The access and service tracks operate according to FIFO scheme: before a locomotive can leave, all the locomotives that came before it should leave first.

Each of the service tracks is divided into several service positions. We consider the case of 2 service positions on each service track. Each service position has its own capacity (up to 3 locomotive sections) and can only accept one locomotive at a time. A four-section locomotive takes both service positions of a service track. When several locomotives should be serviced on different positions of the same service track simultaneously, they should arrive at the track simultaneously, and leave simultaneously, too. Service sessions cannot be interrupted, and there should be a break between each pair of subsequent sessions on the same service track – typically, of at least 15 min. For simplicity, we include this time in service times of locomotives.

Upon leaving an access track, a locomotive is transported to its designated service position on the corresponding service track. Actually, any locomotive should be serviced only once and its maintenance can begin only after its arriving to the LMD.

Let P be the set of all service positions.

Let us define parameters of service position $p \in P$.

- d_p is the number of locomotive sections that p can contain at a time, $d_p \leq 3$.
- $a(p)$ is the service track that p belongs to.

– $i(p)$ is the index number of the service position on the service track (counting from the exit of the service track, starting from 1).
– R_p is the moment of time when position p becomes available. $R_p > T_0$ when, for example, position p was occupied by some locomotive at the beginning of planning horizon T.

Let $AP_{ap}, a \in A, p \in P$ be a Boolean matrix and $AP_{ap} = 1$ if it is possible to transport a locomotive from access track a to service position p, and $AP_{ap} = 0$ otherwise.

Let PM_{pm} be a Boolean matrix and $PM_{pm} = 1$ if inspection position p can be used to service locomotives with model number m, and $PM_{pm} = 0$ otherwise.

The goal is to decide for each locomotive:

– whether it should be placed onto the buffer track,
– when it should leave the buffer and head to the access tracks,
– which of the access tracks it should be placed on,
– when it should leave the access track and move onto the service track,
– which of the service positions of that track it should be serviced at.

Depending on the needs of a specific LMD, there might be different objective functions that one would need to implement. We are considering a number of different objectives to minimize, namely total idle time, total waiting time, maximum waiting time and makespan, and carrying out computational experiments.

A problem of scheduling maintenance of a large fleet of rolling stock in a single depot is investigated in [2]. The goal of this work is to schedule maintenance activities for a large number of trains arriving at the depot taking into consideration resource constraints, such as availability of service platforms, engineers, equipment, etc., while maximizing the throughput. Integer linear programming model and a heuristic algorithm are presented. However, this problem differs from ours one. It does not consider any access tracks, and a buffer track there is replaced with an infinite shunting yard with arbitrary input and output of trains.

If we could ignore the track capacity constraints and limit ourselves to considering only one service position (and, of course, only a subset of locomotives that can be serviced at that particular position), we could obtain the optimal schedule by using Smith's theorem [8]. However, the track capacity constraints are far too significant to overlook for our problem.

In [1] fifth chapter deals with the problem of optimal distribution of jobs on identical parallel machines with the same processing time. Various objective functions are considered. A polynomial algorithms is presented. This algorithm, however, does not take into account the possibility of multiple service positions existence on the same service track (i.e., some of the parallel machines would not be independent), possibility of access tracks presence (i.e. having a limited queue size for each of the machines), and various processing times.

One of the tasks of locomotive fleet management is the distribution of locomotives by location, taking into account various constraints. One of the criteria of optimal control of a transport park is matching the number of locomotive crews and locomotives. In [7], a heuristic algorithm for optimal control of a transport park is presented. The algorithm provides a high-quality solution of assembling and routing of the transport park and outperforms the IBM ILOG CPLEX [11] in terms of calculation time. However, the article does not consider maintenance of the trains.

In [3], problem of scheduling maintenance of series of machines is presented. Each unit of a series of machines (for example, a locomotive fleet) must be maintained. When maintenance of machines involves outsourcing specialists, it takes some time for specialists to arrive to the maintenance facility. Usually, the owner of a fleet has to pay not only for the work of these specialists, but he also has to cover the expenses of transporting them to a designated maintenance facility. Thus, one needs to reduce the transporting times by grouping planned maintenance events together. At the same time, the owner seeks to reduce the downtimes of the machines, while meeting necessary maintenance frequency.

In [6], the problem of the distribution a locomotive fleet between several depots for the maintenance is investigated. An algorithm for the organization of the process of operation and maintenance of locomotive fleet is presented. The algorithm allows to find the optimal distribution of the volume of maintenance of locomotives between depots and obtain a graphical solution of the unit cost of repairs. However, authors consider only the problem of distributing locomotives between LMDs, leaving the problem of operating each particular LMD aside.

Now it can be seen that in all the papers above there are significant differences from our problem, and these results cannot be directly applied to solving the problem considered in our article.

3 Constraint Programming Model

The first stage of the analysis of the problem is an attempt to find an exact solution using Constraint Programming (CP) approach. The important principle of constraint programming consists of distinguishing constraint propagation and decision-making search (see, for example, [10]). Constraint propagation is a deductive activity which consists in deducing new constraints from existing constraints. The large number of constraints in our problem contributes to high efficiency of CP methods. To use this approach, the problem needs to be formulated as a Constraint Satisfaction Problem. So, formalization of the problem in this article is made in terms of CP using the Optimization Programming Language (OPL) [12]. In the computational experiments for CP model the time moments were multiples of 15 min (arrival moments and service times were rounded off to the closest upper values).

3.1 Decision Variables, Functions and Other Denotations

To formulate in terms of constraint programming and solve the presented problem, we use the notion of interval variables. Each variable is a time interval in the horizon T. All of the intervals we use are optional, meaning that any of the processes associated with these intervals should be either carried out, according to the solution (in that case, the interval is called present), or skipped (in that case, the interval is called absent). Absent intervals are ignored by most of the constraints. Functions of interval variables can take an additional argument - a value to be returned when the interval argument of such function is absent.

We are using IBM Ilog CPLEX Optimizer 12.6.3 in Constraint Programming mode. The solver algorithm defines which intervals should be present in the solution, and what are start and end moments of those intervals that are present, given that all constraints are satisfied and the objective function is minimized (we provide the constraints and objective functions below).

Interval variables are the following.

– $intB_l$ is the interval during which locomotive l stays in the buffer $intB_l \in [r_l, T_1]$. If length of this interval is zero, or the interval is absent, the locomotive is treated as if it did not enter the buffer at all.
– $intA_{la}$ is the interval during which locomotive l occupies access track a, $intA_{la} \in [r_l, T_1]$.
– $intP_{lp}$ is the interval when locomotive l occupies service position p, $intP_{lp} \in \big[max(r_l, R_p), T_1\big]$. The length of this interval can't be less than τ_l.

Built-in OPL Functions. In order to work with the interval variables and model the constraints, we need the following built-in functions and constraints defined in the OPL framework. To make the text more laconic, we renamed built-in opl functions, see their original names in brackets.

– $s(int, dval)$ (in OPL: $startOf(int, dval)$) is a built-in function that evaluates to the start moment of interval int when it is present, and evaluates to $dval$ otherwise. If $dval = 0$, then it can be omitted.
– $e(int, dval)$ (in OPL: $endOf(int, dval)$) is a built-in function that evaluates to the end moment of interval int when it is present, and evaluates to $dval$ otherwise. If $dval = 0$, then it can be omitted.
– $p(int)$ (in OPL: $presenceOf(int)$) - a built-in function that evaluates to 1 if the interval variable int is present in the solution, and 0 if it is absent. If $dval = 0$, then it can be omitted.
– $pulse(int, h)$ is a built-in function that represents usage of a renewable resource in a process associated with interval int. h is the amount of resource used. At any time moment $s(int) \le t < e(int)$ usage of the resource by interval int is h, and at any other time moment the resource is not used by this interval. This function can be used in constraints like $\sum_{i \in I} pulse(i, h_i) \le a$, which means that at any time moment collective usage of some resource by intervals from set I cannot exceed a.

Built-in OPL Constraint 'startAtEnd'

– $startAtEnd(int_1, int_2)$ - built-in constraint that forces interval int_1 to start right when interval int_2 ends. If any of the intervals int_1, int_2 is absent, the constraint is automatically considered to be satisfied.

We also need to define a few cumulative functions that will represent the input of the locomotives in using different resources, such as buffer track, access tracks and service positions.

Cumulative Functions

- $BLoad = \sum_{l \in L} pulse(intB_l, u_l)$ is equal to the total number of locomotive sections in the buffer;
- $ALoad(a) = \sum_{l \in L} pulse(intA_{la}, u_l)$ is equal to the total number of sections of locomotives on access track s;
- $PLoad(p) = \sum_{l \in L} pulse(intP_{lp}, 1)$ is equal to the number of locomotives at service position p.

3.2 Objective Functions

In terms of the described decision variables, basic constraints and functions and supplementary functions, the four objective functions mentioned above take the following forms:

1. maximum waiting time $F_1(\pi) = \max_{l \in L, p \in P} s(intP_{lp})$;
2. total waiting time $F_2(\pi) = \sum_{l \in L, p \in P} s(intP_{lp})$;
3. total idle time $F_3(\pi) = \sum_{l \in L, p \in P} e(intP_{lp}) - \tau_l - r_l$;
4. makespan $F_4(\pi) = \max_{l \in L, p \in P} e(intP_{lp})$.

3.3 Constraints

The constraints of the problem will take the following form. To avoid ambiguity, here and below the symbol "$\Longrightarrow$" will denote logical implication ("$A \Longrightarrow B$" is equivalent to "if A, then B").

- Each locomotive, if it enters the buffer, does so right at the moment of arrival at LMD:
$$\forall l \in L : s(intB_l, r_l) = r_l.$$
- The buffer is used according to the LIFO scheme:
$$\forall l_1, l_2 \in L : p(intB_{l_1}) \wedge p(intB_{l_2}) \wedge \big(s(intB_{l_1}) \geq s(intB_{l_2})\big) \Longrightarrow \big(e(intB_{l_1}) \leq e(intB_{l_2})\big).$$
- Total count of sections of all the locomotives occupying the buffer at any time should not exceed the capacity of the buffer:
$$BLoad(b) \leq b.$$

- Each locomotive should be assigned to exactly one access track:

$$\forall l \in L : \sum_{a \in A} p(intA_{la}) = 1.$$

- Each locomotive enters its designated access track right when it leaves the buffer, or when it arrives at the LMD if it did not enter the buffer:

$$\forall l \in L, a \in A : (p(intA_{la}) \wedge p(intB_l)) \implies s(intA_{la}) == s(intB_l);$$

$$\forall l \in L, a \in A : (p(intA_{la}) \wedge !p(intB_l)) \implies s(intA_{la}) == r_l.$$

- Total count of sections of all locomotives that are occupying an access track at any given time should not exceed the capacity of the track:

$$\forall a \in A : ALoad(a) \leq c_a.$$

- The access tracks operate according to FIFO scheme:

$$\forall a \in A, l_1 \in L, l_2 \in L : \big(e(intB_{l_1}) \leq e(intB_{l_2})\big) \implies \big(s(intA_{l_1 a}) \leq s(intA_{l_2 a})\big).$$

- Upon leaving an access track, the locomotive should be transported to one of the service tracks:

$$\forall l \in L, a \in A, p \in P : startAtEnd(intA_{la}, intP_{lp}).$$

- A locomotive cannot be assigned to a service position that is inaccessible from the access track which that locomotive was assigned to:

$$\forall l \in L, a \in A, p \in P : p(intA_{la}) * p(intP_{lp}) \leq AP_{ap}.$$

- At any given time, any service position cannot be occupied by more than one locomotive:

$$\forall p \in P : PLoad(p) \leq 1.$$

- A locomotive cannot be assigned to a service position that does not accept this locomotive model:

$$\forall l \in L, p \in P : p(intP_{lp}) \leq PM_{p,v_l}.$$

- Each locomotive consisting of less than 4 sections should be assigned to exactly one service position:

$$\forall l \in L, u_l < 4 : \sum_{p \in P} p(intP_{lp}) = 1.$$

- Each locomotive consisting of 4 sections should be assigned to exactly two service positions:

$$\forall l \in L, u_l = 4 : \sum_{p \in P} p(intP_{lp}) = 2.$$

  These positions should be on the same service track:

$$\forall l \in L, p_1 \in P, p_2 \in P, a(p_1) \neq a(p_2) : p(intP_{lp_1}) * p(intP_{lp_2}) = 0.$$

– Total capacity of service positions occupied by a locomotive should be sufficient to accommodate the locomotive:

$$\forall l \in L : \sum_{p \in P} p(intP_{lp}) * d_p \geq u_l.$$

– When two locomotives are assigned to neighboring service positions and their service intervals begin simultaneously, these intervals should end simultaneously, too:

$$\begin{aligned}&\forall l_1, l_2 \in L, p_1, p_2 \in P, a(p_1) = a(p_2) : \\ &\qquad \big(p(intP_{l_1p_1}) * p(intP_{l_2p_2}) = 1\big) \wedge \big(s(intP_{l_1p_1}) = s(intP_{l_2p_2})\big) \\ &\qquad\qquad\qquad \implies \big(e(intP_{l_1p_1} = e(intP_{l_2p_2})\big).\end{aligned}$$

– The service positions operate according to FIFO scheme: when two locomotives are assigned to neighboring service positions and their inspection starts simultaneously, the locomotive that arrived at the contiguous access track earlier should be placed closer to the exit from the service track (i.e. it should be assigned to the service position with a smaller index number):

$$\begin{aligned}&\forall l_1, l_2 \in L, l_1 \neq l_2, p_1, p_2 \in P, a(p_1) = a(p_2) : \\ &\qquad \big(s(intA_{l_1a(p_1)}) < s(intA_{l_2a(p_2)})\big) \wedge \big(p(intP_{l_1p_1}) * p(intP_{l_2p_2}) = 1\big) \\ &\qquad\qquad \wedge \big(s(intP_{l_1p_1}) = s(intP_{l_2p_2})\big) \\ &\qquad\qquad\qquad \implies \big(i(p_1) < i(p_2)\big).\end{aligned}$$

As shown in Sect. 5, applying the exact method to solving the problem on real data takes a lot of time, which is unacceptable. Therefore, in the next section we propose a heuristic algorithm for solving the problem.

4 Greedy Algorithm

Let us number all locomotives $1, 2, \ldots, l, \ldots, n$ in the order of their arrival to the LMD. We will build a schedule step by step, where at each step l we will consider l first locomotives arrived to the LMD. Let us introduce a partial schedule $\psi(l)$, which is built for l first locomotives arrived to the LMD. Partial schedule $\psi(l)$ differs from the feasible schedule by the set of locomotives in it, and by the fact that not all locomotives must be serviced (arrived but not serviced locomotives should be buffered). At each time t for partial schedule $\psi(l)$, the following sets of locomotives can be distinguished:

– $I(\psi(l), t)$ is a set of locomotives that arrived earlier or at time t and sent to an access track immediately;
– $D(\psi(l), t)$ is a set of locomotives that arrived earlier or at time t and sent to the buffer, and then to an access track;
– $B(\psi(l), t)$ is a set of locomotives that are in the buffer at time t.

Partial schedule $\psi(l)$ provides the following information:

- a set of appropriate locomotives $\{1, 2, \ldots, l\}$;
- start and end service time of each locomotive from $I(\psi(l), t)$ and $D(\psi(l), t)$;
- service positions (and service tracks) on which each locomotive from sets $I(\psi(l), t)$ and $D(\psi(l), t)$ is serviced;
- a set of locomotives $B(\psi(l), t)$ that are buffered.

In final schedule $B(\pi, +\infty) = \emptyset$, because it is necessary to maintain all locomotives.

Let us introduce arriving time $\overline{r_e}$ of locomotive $e \in I(\psi(l), t) \cup D(\psi(l), t)$ onto an access track. For locomotives sent to an access track immediately after arriving $e \in I(\psi(l), t)$ the arriving time is equal to the time of its arrival to the LMD: $\overline{r_e} = r_e$.

Let us denote a set of all service tracks by Q. For each partial schedule $\psi(l)$ it is possible to allocate a subset $Q_1(\psi(l), t) \subset Q$ for which locomotives are standing on access tracks last and maintained on service tracks alone. We call the index of the last locomotive serviced on service track $q \in Q_1(\psi(l), t)$ under schedule $\psi(l)$ by $l_q(\psi(l), t)$.

Let us denote arriving time of locomotive e for maintenance onto a service track by $S^e(\psi(l), t), a \in I(\psi(l), t) \cup D(\psi(l), t)$; a maintenance completion time (time of exit from the LMD) by $C^e(\psi(l), t)$ in partial schedule $\psi(l)$. We introduce the concept of partial objective functions for the first l locomotives arrived to the LMD by time t:

1. maximum waiting time $F_1(\psi(l), t) = \max\limits_{i \in I(\psi(l),t) \cup D(\psi(l),t)} (S^i(\psi(l), t) - r_i)$;
2. total waiting time $F_2(\psi(l), t) = \sum\limits_{i \in I(\psi(l),t) \cup D(\psi(l),t)} S^i(\psi(l), t)$;
3. total idle time $F_3(\psi(l), t) = \sum\limits_{i \in I(\psi(l),t) \cup D(\psi(l),t)} (C^i(\psi(l), t) - \tau_i - r_i)$;
4. makespan $F_4(\psi(l), t) = \max\limits_{i \in I(\psi(l),t) \cup D(\psi(l),t)} C^i(\psi(l), t)$.

Henceforth, if it is obvious what partial schedule $\psi(l)$, what locomotive l and what time moment t we are talking about, we will omit the arguments. We associate each access track and a service track it leads to. Thus, when we point to service track $q \in Q$, we will also imply the access track and the group of service positions at the same track, taking into account all their characteristics (capacities of access tracks and service positions, and locomotive models they can service).

Let us describe procedures that are used in the heuristic algorithm. All the procedures can be performed either for locomotive e, which just arrived at the LMD, so the partial schedule will be $\psi = \psi(e - 1)$; or for last locomotive in the buffer ($e \in B(t)$) at time moment t of calling the procedure.

Choice_of_service_tracks(e, t, ψ)—a procedure for choosing a set of service tracks K^e for some locomotive e, on access track of which it can be located at time t. The output is a set of service tracks K^e (and corresponding access tracks), on which locomotive e can be serviced, in accordance with schedule ψ. The selected service tracks must correspond to the capacity of the access tracks and the type of locomotive e. If these conditions are not met at this step at time t, then locomotive e is sent to the buffer of a limited capacity (according to the LIFO scheme), and set K^e is empty.

Partial_objective_function$(e, q, \overline{r_e}(q), \psi)$—a procedure for calculating the value of a partial objective function $F(q)$ when setting locomotive e to the access track leading to service track $q \in K^e$. If the number of the locomotive sections $u_e < 4$ and $q \in Q_1(\psi, r_e)$ in schedule ψ, then the algorithm considers two options. Either locomotive e is sent to an access track and will be serviced alone, or locomotive e is sent to an access track and will be serviced along with locomotive l_q on service track q. Then for locomotive e the time of its entering to service position S^e and exit from it C^e are defined as follows.

– If locomotive e is maintained at service track q alone:

$$S^e = \max\{\overline{r_e}, C^{l_q}(\psi)\}, \tag{1}$$

$$C^e = S^e + \tau_e. \tag{2}$$

– If locomotive e is maintained at service track q along with another locomotive l_q:

$$S^e = max\{\overline{r_e}, S^{l_q}(\psi)\}, \tag{3}$$

$$C^e = S^e + max\{\tau_e, \tau_{l_q}\}. \tag{4}$$

If locomotive e is a four-section one, and the set of service tracks K^e is not empty, or the number of sections of locomotive e is less than four and the set of service tracks $K^e \cap Q_1$ is empty, then only one option is considered: locomotive e is sent to the first vacant service track alone. The time of entry and exit from the service position for locomotive e is determined similarly to a non-four-section locomotive when it is serviced alone, according to formulas (1), (2).

If set K^e of service tracks is empty, or in time S^e the number of serviced locomotives is greater than the number of repair crews, then locomotive e is buffered (see *Buffering*$(e, \overline{r_e}(q), B)$ below), if there is a free space. Otherwise, the algorithm stops working, as it is impossible to build the schedule.

Let f^1 be a value of the partial objective function, taking into account that locomotive l is serviced on service track q alone, and f^2 is a value of the partial objective function if the locomotive is serviced together with locomotive l_q. If f^2 exists, then $F(q) = min\{f^1, f^2\}$. Otherwise, $F(q) = f^1$.

Buffering(e, t, B)—a procedure for sending locomotive e to the buffer at time t, B is the set of all locomotives in the buffer. If locomotive e is buffered, then it becomes the last locomotive in the buffer. Moreover, sets I and D remain unchanged and $B = B \cup \{e\}$. In case when there is no vacant space on the access tracks and in the buffer, the locomotive cannot get into LMD, the schedule cannot be built using this algorithm and the algorithm stops working.

Buffer_check($r_{l-1}, r_l, \beta, \psi$)—a procedure for checking the buffer between time r_{l-1} of arriving locomotive $l-1$ and time r_l of locomotive l arriving. The procedure is performed for locomotive β, which is the last one in the buffer at the time of its call. The output information is set of service positions K^β, on which locomotive β can be maintained and possible time $\overline{r_\beta}(m)$ of sending the locomotive β on each applicable access track leading to service position $m \in K^\beta\}$. Let some locomotives have completed maintenance during the interval $(r_{l-1}, r_l]$. If a few locomotives have completed their maintenance on some service position, then we will consider only that one, which was released first. Let us denote the set of locomotives which ended their maintenance on each service position first during the interval $(r_{l-1}, r_l]$, by J, $|J| \leq |M|$. Obviously, $C^j(\psi) \in (r_{l-1}, r_l],\ j \in J$. Now call procedure *Choice_of_service_tracks(β, r_l, ψ)* for selecting a set of service tracks K^β for the locomotive β at time r_l with schedule ψ. The locomotive β can go to the access track to a service position from set K^β, as soon as maintenance on it is completed during interval $(r_{l-1}, r_l]$. For each service track $m \in K^\beta$ time moment $\overline{r_\beta}(m)$ of sending locomotive β to it will be defined as $\overline{r_\beta}(m) = C^j(\psi)$, where locomotive $j \in J$ was maintained on service track m.

In addition, if locomotive a is to be maintained on service track m along with locomotive l_m, then $S^{l_m}(\psi) = S^a(\psi), C^{l_m}(\psi) = C^a(\psi)$. For all other locomotives arrived to the LMD by time $\overline{r_a}$, the schedule remains the same.

The algorithm pseudo-code can found below, which uses the above procedures. The input information of the algorithm is: planning horizon $[T_0, T_1]$, schedule $\overline{\pi}$, built for the previous planned day and, accordingly, the set of locomotives $I(\overline{\pi}, T_0), D(\overline{\pi}, T_0), B(\overline{\pi}, T_0)$ at the start of the current planning period. The sets $I(\overline{\pi}, T_0), D(\overline{\pi}, T_0)$ determine time moments $R_p \geq T_0$ of the beginning of service positions availability. The locomotives that are buffered at the beginning of the current scheduling period (at time T_0) will be considered at the current scheduling period (for the locomotives from $I(\overline{\pi}, T_0), D(\overline{\pi}, T_0)$ the schedule remains constant). The output is schedule π of maintenance of all locomotives. If locomotive maintenance doesn't start before time T_1 in the schedule obtained using the algorithm, it will be considered in the next planning horizon $[T_1, T_2]$.

5 Results and Conclusions

The proposed approaches to solving the problem were tested on data provided by Russian Railways, which correspond to large enterprises of the Eastern Polygon. Figure 2 shows the main characteristics of the three test data sets for three LMDs. Each LMD has its own characteristics that must be considered. For example, in LMD 3 there is a shortage of repair crews, so not all service positions can work simultaneously. Four objective functions are considered: maximum waiting time (F_1), total waiting time (F_2), total idle time (F_3), makespan (F_4). Numerical experiments were carried out on a following personal computer: CPU Intel Core i7 7700 HQ 2800 MHz, 4 cores; 8 GB DDR4 RAM.

Figure 3 shows values of each objective function for each data set, obtained using the CP model and the heuristic algorithm. Figure 4 shows the average

Algorithm 1. Greedy algorithm

Input data: $T_0, T_1, I, D, B, \overline{\pi}$
$t' = T_0$
$\psi = \overline{\pi}$
for all $l = 1 \ldots n$ **do**
  if $B \neq \emptyset$ **then**
    for all $\beta \in B$ **do**
      $(K^{\beta}, [\overline{r_{\beta}}(m)]) \leftarrow \textsc{Buffer\_check}(t', r_l, \beta, \psi)$
      if $K^{\beta} \neq \emptyset$ **then**
        for all $m \in K^{\beta}$ **do**
          $F(m) \leftarrow \textsc{Partial\_objective\_function}(\beta, m, \overline{r_{\beta}}(m), \psi)$
        end for
        $m = \mathrm{argmin}_{i \in K^{\beta}} F(i)$
        $\psi \leftarrow \textsc{Schedule\_changes}(\beta, m, \overline{r_{\beta}}(m), \psi)$
      else
        Break this cycle
      end if
    end for
  end if
  $K^{\beta} \leftarrow \textsc{Choice\_of\_service\_tracks}(l, r_l, \psi)$
  if $K^{l} \neq \emptyset$ **then**
    for all $m \in K^{l}$ **do**
      $F(m) \leftarrow \textsc{Partial\_objective\_function}(l, m, r_l, \psi)$
    end for
    $m = \underset{i \in K^{l}}{\mathrm{argmin}}\, F(i)$
    $\psi \leftarrow \textsc{Schedule\_changes}(l, m, r_l, \psi)$
  else if $K^{l} = \emptyset \ \& \ |B| \leq b$ **then**
    $B \leftarrow \textsc{Buffering}(l, r_l, B(r_l))$
    $\psi \leftarrow \textsc{Schedule\_changes}(l, buffer, r_l, \psi)$
  end if
  $t' = r_l$
end for
if $B \neq \emptyset$ **then**
  for all $\beta \in B$ **do**
    $(K^{\beta}, [\overline{r_{\beta}}(m)]) \leftarrow \textsc{Buffer\_check}(r_n, +\infty, \beta, \psi)$
    for all $m \in K^{\beta}$ **do**
      $F(m) \leftarrow \textsc{Partial\_objective\_function}(\beta, m, \overline{r_{\beta}}(m), \psi)$
    end for
    $m = \underset{i \in K^{\beta}}{\mathrm{argmin}}\, F(i)$
    $\psi \leftarrow \textsc{Schedule\_changes}(\beta, m, \overline{r_{\beta}}(m), \psi)$
  end for
end if
$\pi = \psi$

	Number of locomotives	The presence of a buffer	Number of repair crews	Number of service positions
LMD 1	100	NO	28	12
LMD 2	45	YES	17	6
LMD 3	40	NO	4	8

Fig. 2. Characteristics of LMDs

deviations of the objective functions values obtained by the heuristic algorithm, relative to the values obtained by CP. The last table shows that the heuristic algorithm gives solution for the objective function F_4 (makespan) comparable to the optimizer solution. The value of the objective function obtained using heuristics is even less than that of the optimizer. This can be explained by the fact that in the CP model there is a time discretization by 15-min intervals. As you can see in Fig. 5, which represents a comparison of both methods to the current methodology of Russian Railways, a heuristic algorithm in most cases shows an advantage in comparison with the existing Russian Railways method.

	Constraint Programming				Greedy Algorithm			
Instance	F1	F2	F3	F4	F1	F2	F3	F4
LMD 1	105	1755	2415	1500	174	2051	2799	1493
LMD 2	90	375	525	1500	138	571	706	1471
LMD 3	105	165	180	1500	88	326	421	1435

Fig. 3. The values of the objective functions

F1	F2	F3	F4
33%	40%	26%	-3%

Fig. 4. The average error (CP vs GA)

In further research we plan to refine the proposed algorithms and to make new algorithms for the problem: dynamic programming and local search algorithms. A transition to a more complex problem statement is also planned. Knowing the characteristics of an LMD and the planned hourly arriving of the locomotives, it is necessary to estimate the maximum number of locomotives that can be serviced in the LMD. It is necessary to build such an autonomous model, which, having all possible combinations of locomotive arrivals, will produce a set of all possible outputs of locomotives from maintenance.

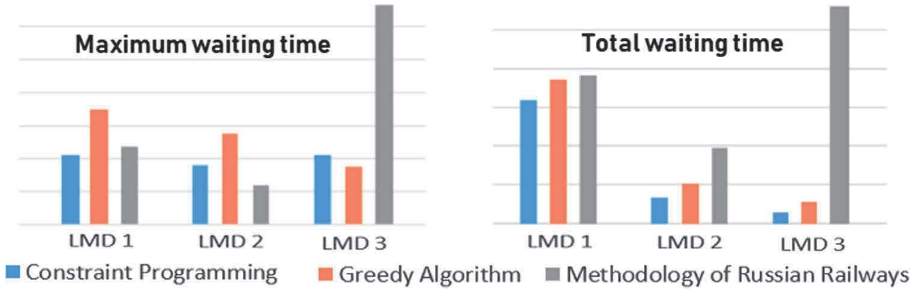

Fig. 5. Maximum Waiting Time (F_1) and Total Waiting Time (F_2)

References

1. Brucker, P.: Scheduling Algorithms. Springer, New York (2006). https://doi.org/10.1007/978-3-540-69516-5
2. Evers, R.P.: Algorithms for Scheduling of Train Maintenance, Netherlands (2010)
3. Huang, J.-Y.: New search algorithm for solving the maintenance scheduling problem for a family of machines. Optim. Methods Softw. **21**, 461–477 (2006)
4. Hansen, P., Mladenovic, N.: Variable neighborhood search: principles and application. Eur. J. Oper. Res. 335–350 (1999)
5. Hansen, P., Mladenovic, N.: Variable neighborhood decomposition search. J. Heuristics 449–467 (2001)
6. Kudayarov, M.: Selection of the optimal organization of repair of locomotives at the railway site. In: Proceedings of PTU (2012)
7. Mardanex, E., Lin, Q., Loxton, R.: A heuristic algorithm for optimal fleet composition with vehicle routing considerations. Optim. Methods Softw. **31**, 272–289 (2016)
8. Smith, W.E.: Various optimizers for single-stage production. Naval Res. Logist. Quart. **3**, 59–66 (1956)
9. Vaidyanathan, B., Ahuja, R.K., Orlin, J.B.: The locomotive routing problem. Transp. Sci. **42**, 492–507 (2008)
10. Dechter, R.: Constraint Processing. Morgan Kaufmann Publishers, San Francisco (2003)
11. IBM Ilog CPLEX. https://www.ibm.com/products/ilog-cplex-optimization-studio
12. IBM. https://www.ibm.com/analytics/data-science/prescriptive-analytics/optimization-modeling