Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-12 (2016) 226–230

On a generalized single machine scheduling problem with time-dependent processing times. ⋆

Alexander A. Lazarev * **Dmitry I. Arkhipov** ** **Frank Werner** ***

* *V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russian Federation; Lomonosov Moscow State University, Moscow, Russian Federation; Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation; International Laboratory of Decision Choice and Analysis, National Research University Higher School of Economics, Moscow, Russian Federation. (e-mail: jobmath@mail.ru)*

** *V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russian Federation. (e-mail: miptrafter@gmail.com)*

*** *Faculty of Mathematics, Institute of Mathematical Optimization, Otto-von-Guericke University Magdeburg, Germany (frank.werner@ovgu.de)*

Abstract: In this paper, a generalized formulation of a classical single machine scheduling problem is considered. A set of n jobs characterized by their release dates, deadlines and a start time-dependent processing time function $p(t)$ has to be processed on a single machine. The objective is to find a Pareto-optimal set of schedules with respect to the criteria $\varphi_{\max}$ and makespan, where $\varphi_{\max}$ is a non-decreasing function depending on the completion times of the jobs. We present an approach that allows to find an optimal schedule with respect to different scheduling criteria, such as the minimization of makespan, lateness or weighted lateness, tardiness and weighted tardiness etc. in time polynomially depending on the number of jobs. The complexity of the presented algorithm is $O(n^3 \max\{\log n, H, P\})$, where H and P are the complexity of calculating $\varphi_j(t)$ and $p(t)$, respectively.

Keywords: scheduling algorithms, single-machine scheduling, polynomial algorithms, makespan, Pareto set.

1. INTRODUCTION

We consider the problem of scheduling a set N of n jobs to be processed on a single machine. Each job is characterized by a *release time* r_j and a *deadline* D_j. The *processing times* of the jobs are defined by a start time-dependent function $p(t)$ such that the term $t + p(t)$ is non-decreasing and the value $p(t)$ can be calculated in P operations for any t. The machine can process only one job at a time. Preemption of processing is not allowed, i.e., the processing of any job started at time t will be completed at time $t + p(t)$. A *schedule* π assigns a *start time* $S_j(\pi)$ to each job j such that $S_j(\pi) \geq r_j$ and $S_j(\pi) \geq S_k(\pi) + p(S_k(\pi))$ for any job k with $S_k(\pi) < S_j(\pi)$. Let us define $C_j(\pi) = S_j(\pi) + p(S_j(\pi))$ as the *completion time* of the job j under the schedule π. We define the set of *feasible schedules* as $\Pi(N)$, where for each $j \in N$ the following inequality holds under the schedule $\pi \in \Pi(N)$:

$$C_j(\pi) < D_j.$$

We denote the *objective function* as $\max \varphi_j$, where for each job $j \in N$ function φ_j is non-decreasing in the completion time C_j and for each value y, it is possible to find the time t' with

$$t' = \min\{t|\varphi_j(t) \geq y\}$$

in H operations. The goal is to find a feasible schedule $\pi \in \Pi(N)$ satisfying

$$\min_{\pi\in\Pi(N)} \max_{j\in N} \varphi_j(\pi). \qquad (1)$$

According to the classical 3-field scheduling classification scheme proposed by Graham et al. (1979), this problem can be denoted as $1|r_j, p_j = p(t), D_j|\varphi_{\max}$. Such a problem formulation includes the following special cases:

- $1|r_j, p_j = p(t), D_j|C_{\max}$. $\varphi_j(t) = t$;
- $1|r_j, p_j = p(t)|L_{\max}$. $\varphi_j(t) = t - d_j$;
- $1|r_j, p_j = p(t)|T_{\max}$. $\varphi_j(t) = \max\{0, t - d_j\}$;
- $1|r_j, p_j = p(t), w_j \geq 0|wL_{\max}$. $\varphi_j(t) = w_j(t - d_j)$;

⋆ Supported by RFBR grants 13-01-12108, 15-07-07489, 15-07-03141 and DAAD grant A/1400328

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.603

- $1|r_j, p_j = p(t), w_j \geq 0|wT_{\max}$. $\varphi_j(t) = \max\{0, w_j(t - d_j)\}$,

where w_j is the weight for job j. It should be noted that in all cases the function $p(t)$ can set time intervals of the availability of the machine (time windows).

In this paper, we consider the problem of constructing a Pareto-set of schedules to solve the bi-criteria problem $1|r_j, p_j = p(t), D_j|\varphi_{\max}, C_{\max}$. To the best of our knowledge, there does not exist any solution algorithm for the problem under consideration in the literature. The existing literature about scheduling with time-dependent processing times has been reviewed in Alidaee et al. (1999) and Cheng et al. (2004). The detailed survey of single machine and parallel machine scheduling problems of jobs which have the deterioration and shorting rates presented in S. Gawiejnowicz (2008). A lot of different problems with time-dependent processing times was considered in Yin et al. (2015). The problem of minimizing the makespan for jobs with equal processing times $1|r_j, p_j = p, D_j|C_{\max}$ was considered by Simons (1978) and Garey et al. (1981). A lot of scheduling problems with equal processing times were considered in the survey by Kravchenko and Werner (2011). A polynomial LP algorithm for the problem $P|r_j, p_j = p, D_j|\max\varphi_j(C_j)$, where φ_j is a non-decreasing function for any j, was presented in Kravchenko and Werner (2007). The solution of the bi-criteria problem $1|r_j, p_j = p|L_{\max}, C_{\max}$ was presented in Lazarev et al. (2015).

The remainder of this paper is as follows. In Section 2, an auxiliary problem is formulated and an algorithm for its solution is presented. An algorithm to construct the Pareto set with respect to the criteria $\varphi_{\max}$ and makespan is presented in Section 3. The complexity of the presented algorithms is estimated in Section 4, and in Section 5 we give some concluding remarks.

2. AUXILIARY PROBLEM

Let $O_1(\pi), \ldots, O_n(\pi)$ be the sequence of jobs in which they are processed under the schedule π, and $O(j, \pi)$ be the ordinal number of job $j \in N$ under the schedule π, i.e.,

$$O(j, \pi) = i \quad \Leftrightarrow \quad O_i(\pi) = j.$$

We define a family of sets $F = \{N_0, N_1, \ldots, N_n\}$ such that for each $i = 0, 1, \ldots, n$, N_i is the set of jobs the ordinal number of which must be not larger than i, i.e., we have $N_0 \subseteq N_1 \subseteq \ldots \subseteq N_n$. For any $i = 0, \ldots, n$, we define $\overline{N_i} = N \setminus N_i$. We say that schedule π *satisfies* the family of sets $F = \{N_0, N_1, \ldots, N_n\}$ if for $i = 1, \ldots, n$ and for any job $j \in N_i$, the inequality

$$O(j, \pi) \leq i$$

holds. This implies that for each $i = 1, \ldots, n$, only jobs from the set $\overline{N_{i-1}}$ can be processed under the ordinal number i, i.e.,

$$O_i(\pi) \in \overline{N_{i-1}}. \tag{2}$$

It is obvious that $N_0 = \emptyset$, $N_n = N$ and for each $i = 0, \ldots, n$, the number of jobs, which belong to set N_i is not larger than i, i.e.,

$$|N_i| \leq i. \tag{3}$$

For each job $j \in N$, we define $N(j, F) = i$ if $j \in N_i$ and $j \notin N_{i-1}$, i.e.,

$$N(j, F) = \min_{i=0,\ldots,n} \{i | j \in N_i\}.$$

Let $\Phi(N, F, y) \subseteq \Pi(N)$ be the set of schedules such that $\pi \in \Phi(N, F, y)$ satisfies the given set F and for each job $j \in N$, the inequality

$$\varphi_j(\pi) < y$$

holds. Note that, if $F^0 = \{\emptyset, \ldots, \emptyset, N\}$, then

$$\Phi(N, F^0, +\infty) = \Pi(N).$$

Now let us formulate the auxiliary problem.

Auxiliary problem. *Find a schedule* $\pi(F, y) \in \Phi(N, F, y)$ *satisfying*

$$\min_{\pi \in \Pi(N)} \max_{j \in N} \{C_j(\pi) | \varphi_j(\pi) < y, C_j(\pi) < D_j\}. \tag{4}$$

To solve this problem, we need to prove the following lemma.

Lemma 1. Under the feasible schedule $\pi \in \Pi(N)$, the start time of the first job must satisfy the inequality

$$S_{O_1(\pi)}(\pi) \geq r_{O_1(\pi)} \tag{5}$$

and for each $i = 2, \ldots, n$, the start time of the job $O_i(\pi)$ must satisfy the inequality

$$S_{O_i(\pi)}(\pi) \geq \max\{r_{O_i(\pi)}, S_{O_{i-1}(\pi)}(\pi) + p(S_{O_{i-1}(\pi)}(\pi))\}. \tag{6}$$

Proof. Due to the definition of the schedule π, each job $j \in N$ must satisfy the inequality

$$S_j(\pi) \geq r_j$$

and for any job k with $S_k(\pi) \geq S_j(\pi)$, the inequality

$$S_j(\pi) \geq S_k(\pi) + p(S_k(\pi))$$

must hold. If $j = O_1(\pi)$, then statement (5) is true. If $k = O_{i-1}(\pi)$ and $j = O_i(\pi)$ for $i = 2, \ldots, n$, then inequality (6) is true.

Now let us present an algorithm to solve the auxiliary problem.

AUXILIARY ALGORITHM A:

0. Input data:
   $N, F, N(1, F), \ldots, N(n, F), y$.
1. For each $j = 1, \ldots, n$ and $i = 0, \ldots, n$, set:
   a) $m := 0$;
   b) $D_j(y) := \min\{\min t | \varphi_j(t) \geq y, D_j\}$;
   c) $N_i^m := N_i$.
2. Assign the ordinal numbers $i = n, \ldots, 1$ to the jobs according to the latest release date rule subject to the inclusion (2):
   $$O_i(\pi^m) := \arg \max_{j \in \overline{N_{i-1}^m}} r_j.$$
3. Set the earliest possible start times of the jobs according to Lemma 1:
   $$S_{O_1(\pi^m)}(\pi^m) := r_{O_1(\pi^m)};$$
   $$S_{O_i(\pi^m)}(\pi^m) := \max\{r_{O_i(\pi^m)}, S_{O_{i-1}(\pi^m)}(\pi^m) + \\ + p(S_{O_{i-1}(\pi^m)}(\pi^m))\}, i = 2, \ldots, n.$$

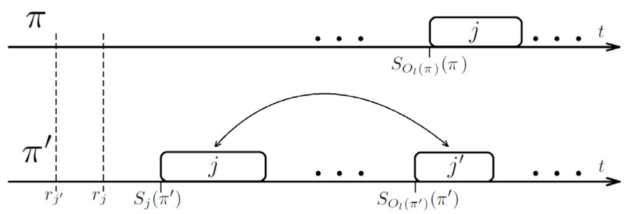

Fig. 1. Exchanging jobs under the schedule π'.

4. For the jobs $j = 1, \ldots, n$ and the ordinal numbers $i = N(j, F^m), \ldots, 0$, check the inequalities
$$C_{O_i(\pi^m)}(\pi^m) < D_j(y). \tag{7}$$
   a) If for any pair j, i, inequality (7) does not hold, include job j into the set N^m_{i-1}, change $N(j, F^m) := i - 1$, and continue with checking inequality (7).
   b) If inequality (7) is correct and $j \neq n$, then check the next job.
   c) If $j = n$ and the inequalities are correct for all pairs j, i with $i = O(j, \pi)$, then **return**$\{\pi^m\}$.
   d) If $j = n$ and for any pair j, i with $i = O(j, \pi)$ the inequality is incorrect, go to step 5.
5. For the sets $N^m_i, i = 0, \ldots, n$ check inequalities (3):
$$|N^m_i| \leq i.$$
   a) If for all sets condition (3) is correct, then set $m := m + 1$, for all $i = 0, 1, \ldots, n$, set $N^m_i := N^{m-1}_i$ and go to the next iteration (step 2).
   b) Otherwise, **return**$\{\emptyset\}$.

Lemma 2. Suppose that π^m is the schedule, constructed at iteration m of Algorithm A subject to F^m. Then for each schedule $\pi' \in \Phi(N, F^m, y)$, which satisfies the family of sets F^m, the following inequality holds for each $i = 1, \ldots, n$:
$$S_{O_i(\pi^m)}(\pi^m) \leq S_{O_i(\pi')}(\pi').$$

Proof. Let us compare the schedules π and π' successively looking for a difference between the jobs with the ordinal numbers $n, n-1, \ldots, 1$. Suppose that the first difference was found for a job with the ordinal number l, i.e.:
$$O_n(\pi) = O_n(\pi'), \ldots, O_{l+1}(\pi) = O_{l+1}(\pi'),$$
$$O_l(\pi) \neq O_l(\pi').$$
In this case, we get $O_l(\pi) = j$, $O_l(\pi') = j'$ and $O(j, \pi') < l$.

Both schedules π and π' satisfy the family of sets $F^m = \{N^m_0, \ldots, N^m_n\}$. Since the job at the step 2 of Algorithm A was selected according to the latest release date rule, we have
$$S_j(\pi') \geq r_j \geq r_{j'},$$
and hence,
$$S_{O_l(\pi')}(\pi') \geq r_j.$$
Thus, we can exchange the ordinal numbers of the jobs j and j' under the schedule π' (see Fig. 1). Note that the schedule obtained after the exchange satisfies the set F^m because of $j, j' \notin N^m_i$. Repeat such an operation until for each job j and the obtained set π'', the equality
$$O(j, \pi) = O(j, \pi'')$$
will hold and for any $i = 1, \ldots, n$, the equality
$$S_{O_i(\pi')}(\pi') = S_{O_i(\pi'')}(\pi'')$$
will hold. According to Lemma 1, we have
$$S_{O_1(\pi'')}(\pi'') \geq r_{O_1(\pi'')} = S_{O_1(\pi)}(\pi),$$
and for $i = 2, \ldots, n$, the inequalities
$$S_{O_1(\pi'')} \geq \max\{r_{O_i(\pi'')}, S_{O_{i-1}(\pi'')}(\pi'') + p(S_{O_{i-1}(\pi'')}(\pi''))\}$$
$$\geq \max\{r_{O_i(\pi)}, S_{O_{i-1}(\pi)}(\pi) + p(S_{O_{i-1}(\pi)}(\pi))\} = S_{(O_i(\pi))}(\pi)$$
hold. Therefore, for each $i = 1, \ldots, n$, the inequality
$$S_{O_i(\pi^m)}(\pi^m) \leq S_{O_i(\pi'')}(\pi'') = S_{O_i(\pi')}(\pi')$$
holds.

Lemma 3. Suppose that π^{m-1} is the schedule constructed at the iteration $m-1$ of Algorithm A subject to the set F^{m-1} and after the inclusions at step 4, the family of sets F^m was obtained. Then each schedule $\pi' \in \Phi(N, F^{m-1}, y)$, which satisfies F^{m-1}, satisfies F^m as well.

Proof. According to the inclusion rule of step 4a) of Algorithm A, for any job O_j and sets $N^{m-1}_i \in F^{m-1}$, $N^m_i \in F^m$ with $O_j \notin N^{m-1}_i$, $O_j \in N^m_i$, the inequality
$$C_{O_{i+1}(\pi^{m-1})}(\pi^{m-1}) \geq D_j(y)$$
must hold. The schedules π and π' satisfy the family of sets F^{m-1}. Hence, according to Lemma 2, for any $i = 0, \ldots, n-1$, the inequality
$$S_{O_{i+1}(\pi^{m-1})}(\pi^{m-1})) \leq S_{O_{i+1}(\pi')}(\pi')$$
holds. Since the expression $t + p(t)$ is non-decreasing in t, we get the inequality
$$C_{O_{i+1}(\pi^{m-1})}(\pi^{m-1})) \leq C_{O_{i+1}(\pi')}(\pi'),$$
and therefore,
$$C_{O_{i+1}(\pi')}(\pi') \geq D_j(y).$$
Hence, inequality $O(j, \pi') \leq i$ holds for any $j \in N^m_i$. Thus, the schedule π' satisfies the family of sets F^m.

Corollary 1. Suppose that Algorithm A constructed the schedule $\pi(F, y)$ successfully and at the last iteration, the family of sets F^m was obtained. Then each schedule $\pi' \in \Phi(N, F, y)$ which satisfies F, satisfies F^m as well, i.e., we have $\Phi(N, F, y) = \Phi(N, F^m, y)$.

Theorem 1. Suppose that Algorithm A returns the schedule $\pi(F, y) \neq \emptyset$. Then $\pi(F, y)$ has the minimal makespan value among all schedules of the set $\Phi(N, F, y)$. If $\pi(F, y) = \emptyset$, then $\Phi(N, F, y) = \emptyset$.

Proof. According to Corollary 1 of Lemma 3, any schedule $\pi' \in \Phi(N, F, y)$, satisfies the family of sets F^m obtained at the last iteration of the construction of the schedule $\pi(F, y)$ by Algorithm A, i.e., $\pi' \in \Phi(N, F^m, y)$. Hence, if Algorithm A failed at iteration m, then for some $i = 0, \ldots, n$, inequality (3) is violated and $\Phi(N, F^m, y) = \Phi(N, F, y) = \emptyset$. The second statement of the theorem is proved.

According to Lemma 1, for any $\pi' \in \Phi(N, F^m, y)$, the inequality
$$S_{O_i(\pi(F,y))}(\pi(F, y)) \leq S_{O_i(\pi')}(\pi')$$
holds for $i = 1, \ldots, n$. Since the expression $t + p(t)$ is non-decreasing in t and inequality
$$S_{O_n(\pi(F,y))}(\pi(F, y)) \leq S_{O_n(\pi')}(\pi')$$
holds, we have
$$C_{O_n(\pi(F,y))}(\pi(F, y)) \leq C_{O_n(\pi')}(\pi').$$

Thus, for any $\pi' \in \Phi(N, F, y)$, the following inequality holds:

$$C_{\max}(\pi(F, y)) \le C_{\max}(\pi').$$

Hence, Algorithm A constructs an optimal schedule with respect to the criterion (4).

3. SOLUTION OF THE MAIN PROBLEM

Now let us present an approach to construct the Pareto set $\Omega(N)$ with respect to the criteria $\varphi_{\max}$ and makespan.

MAIN ALGORITHM M:

0. Input data: $N, \Omega(N) = \emptyset, y = +\infty$.
1. Set $s := 0$, $N_n^s := N$ and for all $i = 0, \dots, n-1$: $N_i^s := \emptyset$. Moreover, for each job $j \in N$ set $N(j, F^s) = N$.
2. Construct the schedule $\pi_{s+1} = \pi(F^s, y)$ using Algorithm A. Let upon the completion of Algorithm A the family of sets $F^{s+1} = \{N_0^{s+1}, N_1^{s+1}, \dots, N_n^{s+1}\}$ be obtained.
3. If $\pi_{s+1} \neq \emptyset$:
   a) set $\pi^*(N) := \pi_{s+1}$, $y := \max\limits_{j \in N} \varphi_j(\pi_{s+1})$;
   b) if $C_{\max}(\pi_s) < C_{\max}(\pi_{s+1})$, include π_{s+1} into the $\Omega(N)$, set $s := s + 1$ and go to the next iteration (step 2);
   c) if $C_{\max}(\pi_s) = C_{\max}(\pi_{s+1})$, replace π_s on π_{s+1} in the set $\Omega(N)$, set $F^s := F^{s+1}$ and go to the next iteration (step 2).
4. If $\pi_{s+1} = \emptyset$, then **return** $\{\Omega(N), \pi^*(N)\}$.

Lemma 4. Suppose that $\pi_{s+1} = \pi(F^s, y^s)$ is a schedule constructed at the iteration $s + 1$ of Algorithm M. Then each schedule $\pi' \in \Pi(N)$ with $\varphi_{\max}(\pi') < y^s$ satisfies the set F^s, i.e., we have

$$\Phi(N, F^s, y^s) = \Phi(N, F^0, y^s),$$

where $F^0 = \{\emptyset, \dots, \emptyset, N\}$.

Proof. Note that $\Pi(N) = \Phi(N, F^0, +\infty)$. At the first iteration of Algorithm M, the schedule $\pi_1 = \pi(F^0, y^0), y^0 = +\infty$ was constructed by Algorithm A and a family F^1 was obtained. According to Corollary 1 of Lemma 3, we have

$$\Phi(N, F^1, y^0) = \Phi(N, F^0, y^0).$$

Since $y^1 = \varphi_{\max}(\pi_0) < y^0$, we have

$$\Phi(N, F^1, y^1) = \Phi(N, F^0, y^1),$$

and subject to Corollary 1 of Lemma 3

$$\Phi(N, F^2, y^1) = \Phi(N, F^1, y^1) = \Phi(N, F^0, y^1).$$

Hence

$$\Phi(N, F^2, y^2) = \Phi(N, F^0, y^2).$$

Apply successively a similar argument to iteration $i = 3, \dots, s$, we obtain

$$\Phi(N, F^i, y^i) = \Phi(N, F^0, y^i).$$

Theorem 2. The set of schedules $\Omega(N)$ returned by the Algorithm M is Pareto-optimal with respect to the criteria $\varphi_{\max}$ and $C_{\max}$. The schedule $\pi^*(N)$ returned by Algorithm M satisfies the optimality criterion (1).

If $\Omega(N) = \emptyset$, then there is no feasible schedule, i.e., $\Pi(N) = \emptyset$.

Proof. According to Theorem 1, the schedule $\pi(F^0, +\infty)$ obtained at the first iteration of Algorithm M has the minimal makespan value among all schedules of the set $\Phi(N, F^0, +\infty) = \Pi(N)$. If $\pi(F^0, +\infty) = \emptyset$, then $\Pi(N) = \emptyset$. The second statement is proved.

Suppose that $\pi_i = \pi(F^i, y^i))$ is a schedule constructed at the iteration i of Algorithm M. According to Theorem 1 and Lemma 4, the schedule π_i is optimal with respect to the makespan criterion among all schedules of the set $\Phi(N, F^i, y^i) = \Phi(N, F^0, y^i)$, i.e., for each π' with $\varphi_{\max}(\pi') < y^s$, the inequality

$$C_{\max}(\pi_i) \le C_{\max}(\pi')$$

holds. According to the Theorem 1 we have, that

$$C_{\max}(\pi_{i+1}) \ge C_{\max}(\pi_i)$$

for each two schedules $\pi_i, \pi_{i+1} \in \Omega(N)$. Subject to the steps 3b) and 3c) of the Algorithm M, for each two schedules $\pi_i, \pi_{i+1} \in \Omega(N)$ holds

$$C_{\max}(\pi_{i+1}) > C_{\max}(\pi_i)$$

. Since $y^i = \varphi_{\max}(\pi_{i-1})$ and $C_{\max}(\pi_{i+1}) > C_{\max}(\pi_i)$, we have

$$C_{\max}(\pi_1) < C_{\max}(\pi_2)) < \dots < C_{\max}(\pi_s),$$
$$\varphi_{\max}(\pi_1) > \varphi_{\max}(\pi_2) > \dots > \varphi_{\max}(\pi_s).$$

If the schedule $\pi(F^s, y^s)$ does not exist, then according to Theorem 1 and Lemma 4, we have $\Phi(N, F^0, y^s) = \Phi(N, F^s, y^s) = \emptyset$, i.e., there is no schedule with an objective function $\varphi_{\max}$ lower than y^s. Thus, the schedule $\pi^*(N) = \pi_s$ satisfies the optimality criterion (1):

$$\varphi_{\max}(\pi_s) = \min_{\pi \in \Pi(N)} \max_{j \in N} \{\varphi_j(\pi) | C_j(\pi) < D_j\},$$

and $\Omega(N)$ is a Pareto-optimal set with respect to the criteria $\varphi_{\max}$ and makespan.

4. ESTIMATION OF THE COMPLEXITY

Theorem 3. The overall complexity of Algorithm M is $O(n^3 \max\{\log n, H, P\})$, where n is the number of jobs, H is the complexity of finding time t' such that

$$t' = \min\{t | \varphi_j(t) \ge y\},$$

and P is the complexity of the calculation of $p(t)$.

Proof. First, let us estimate the complexity of Algorithm A step by step.

- We calculate the values $D_1(y), \dots, D_n(y)$ in $O(nH)$ operations and make some assignments in $O(n)$ operations at step 1.
- The assignments of the ordinal numbers to the jobs at step 2 takes $O(n \log n)$ operations.
- Setting the start times at step 3 takes $O(nP)$ operations, where P is the complexity of the calculation of $p(t)$.
- The complexity of the steps 4b) and 4d) is $O(n)$ and $O(1)$, respectively. The complexity of steps 4a) and 4c) will be estimated later.
- Checking $n + 1$ inequalities at step 5 takes $O(n)$ operations.

Hence, the steps 1, 2, 3, 4b), 4d), 5 of one iteration of Algorithm A take $O(n \max\{\log n, H, P\})$ operations.

At each iteration m of Algorithm A except the final one, there are some inclusions into the sets $N_0^m, N_1^m, \dots, N_{n-1}^m$. Suppose that the maximum value of the objective function under the schedule $\pi(F, y)$ was achieved for the job j, i.e., we have

$$\varphi_{\max}(\pi(F, y)) = \varphi_j(\pi(F, y)).$$

According to Lemma 1, Algorithm A constructs a schedule $\pi(F, y)$ with minimal possible starting times among all schedules from the set $\Phi(N, F, y)$. Since φ_j is a non-decreasing completion time-dependent function, according to Lemma 4, job j must be processed under any schedule with an objective function value lower than $\varphi_{\max}(\pi(F, y))$ with an ordinal number lower than $O(j, \pi(F, y))$. Hence, at each iteration of Algorithm M (steps 2-3), there are one or more inclusions into the sets of the family F^s. Thus, the total number of iterations of Algorithm A in the performance of Algorithm M can be estimated by the total number of possible inclusions into the sets $N_0, \dots, N_n$. According to (3), we obtain that the number of possible inclusions is not larger than

$$\sum_{i=0}^{n-1} |N_i| \le \frac{n^2 - n}{2} = O(n^2).$$

Thus, the overall complexity contribution of steps 1, 2, 3, 4b), 4d), and 5 of Algorithm A is $O(n^3 \max\{\log n, H, P\})$.

Let us estimate the overall complexity contribution of steps 4a) and 4c) of Algorithm A. Note that the number of steps 4a) in the performance of Algorithm M for any job $j \in N$ is equal to the number of inclusions of job j into the sets $N_0, \dots, N_{n-1}$. Thus, the total number of steps 4a) is not more than the total number of possible inclusions $\frac{n^2-n}{2} = O(n^2)$. Since the complexity of step 4a) is $O(1)$, the total contribution of the steps 4a) in the performance of Algorithm M is $O(n^2)$.

The total number of steps 4c) of Algorithm A in the performance of Algorithm M is equal to the number of schedules in the set $\Omega(N)$. This value is not more than the number of iterations of Algorithm A in the performance of Algorithm M, which is not more than $\frac{n^2-n}{2} = O(n^2)$ as it was noted earlier. The complexity of step 4c) is $O(1)$ and hence, the total complexity contribution of steps 4c) in the performance of Algorithm M is $O(n^2)$.

Thus, the overall complexity of Algorithm M is not more than $O(n^3 \max\{\log n, H, P\})$ operations.

Corollary 2. The number of schedules in the set $\Omega(N)$ is not larger than

$$\sum_{i=0}^{n-1} |N_i| \le \frac{n^2 - n}{2}.$$

5. CONCLUDING REMARKS

In this paper, an approach to solve the problem $1|r_j, p_j = p(t), D_j|\varphi_{\max}$ was presented. In addition, the Pareto set with respect to the criteria $\varphi_{\max}$ and $C_{\max}$ was constructed. The core idea of our approach was to construct a schedule with lower value $\varphi_{\max}$ than in the previous step using strict deadlines and the knowledge obtained in the previous steps for filling the family of sets F.

In future research we are going to focus on the problems with different time-dependent processing time functions $p_j(t)$ and on the development of some practical applications of the presented method.

REFERENCES

B Alidaee, N.K. Womer. Scheduling with time-dependent processing times: Review and Extensions. *Journal of the Operational Research Society*, Vol. 50, 711 - 720 pp., 1999.

T.C.E. Cheng, Q. Ding, B.M.T. Lin. A concise survey of scheduling with time-dependent processing times. *European Journal of Operational Research*, Vol. 152, 1 - 13 pp., 2004.

S. Gawiejnowicz. Time Dependent Scheduling. *Springer.* 377 p., 2008.

Y. Yin, T.C.E. Cheng, and C. Wu. Scheduling with Time-Dependent Processing Times. *Math. Probl. Eng.* 2015.

R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Discrete Math.* V.5. 287-326 pp., 1979.

B. Simons. A fast algorithm for single processor scheduling. *In 19th Annual Symposium on Foundations of Computer Science*, Ann Arbor, Mich., 246 - 252 pp., 1978.

M.R. Garey, D.S. Johnson, B.B. Simons and R.E. Tarjan. Scheduling unit-time tasks with arbitrary release times and deadlines. *SIAM Journal on Computing*, Vol. 10, No. 2, 256 - 269 pp., 1981.

S. Kravchenko and F. Werner. Parallel machine problems with equal processing times: a survey. *Journal of Scheduling*, Vol. 14, No. 5, 435 - 444 pp., 2011.

S. Kravchenko and F. Werner. On a parallel machine scheduling problem with equal processing times. *Otto-von-Guericke-Universität Magdeburg, FMA,* Preprint 26/07, 9 pp., 2007.

A. Lazarev, D. Arkhipov, F. Werner Single machine scheduling: Finding the Pareto setwith equal processing times with respect to criteria L_{max} and C_{max}. *Proceedings of the 7th MISTA Conference,* Prague / Czech Republic, 797 - 800 pp., 2015.