S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru

Том 16, стр. 406–426 (2019) УДК 519.854.2
DOI 10.33048/semi.2019.16.023 MSC 90B35

SOME POSITIVE NEWS ON THE PROPORTIONATE OPEN SHOP PROBLEM

SERGEY SEVASTYANOV

Abstract. The special case of the open shop problem in which every job has equal length operations on all machines is known as a *proportionate open shop problem*. The problem is NP-hard in the case of three machines, which makes topical such traditional research directions as designing efficient heuristics and searching for efficiently solvable cases. In this paper we found several new efficiently solvable cases (wider than known) and designed linear-time heuristics with good performance guarantees (better than those known from the literature). Besides, we computed the exact values of the power of preemption for the three-machine problem, being considered as a function of a parameter γ (the ratio of two standard lower bounds on the optimum: the machine load and the maximum job length). We also found out that the worst-case power of preemption for the m-machine problem asymptotically converges to 1, as m tends to infinity. Finally, we established the exact complexity status of the three-machine problem by presenting a pseudo-polynomial algorithm for its solution.

Keywords: open shop, proportionate, scheduling, makespan minimization, power of preemption, polynomial time heuristic, dynamic programming.

1. Introduction

We consider the classical m-machine *open shop* problem with the minimum makespan objective. Two versions of this problem: with or without operation interruptions allowed, are normally denoted as $\langle O \,|\, pmtn \,|\, C_{\max}\rangle$ and $\langle O \,||\, C_{\max}\rangle$, respectively. Since the first version is known to be polynomially solvable (as shown by

---

Sevastyanov, S.V., Some positive news on the proportionate open shop problem.

The work is supported by the Russian Foundation for Basic Research (grant 17-07-00513) and by the program of fundamental scientific researches of the SB RAS (project 0314-2019-0014).

Received December, 21, 2018, published March, 29, 2019.

Gonzalez and Sahni (1976)), we will entirely concentrate on the second version, although the question on the exact value of the worst case ratio of the optima of these two problems (so called *power of preemption*) will also be within the sphere of our interests.

Open Shop problem. We are given a set of jobs $\{J_1, \ldots, J_n\}$ that should be processed on a given set of dedicated machines $\{M_1, \ldots, M_m\}$. Each job J_j consists of m parts $\{O_j^i \mid i = 1, \ldots, m\}$, called *operations*. Operation O_j^i should be processed on machine M_i, which requires p_j^i time units taken as a solid piece of time. (This requirement distinguishes this version from that where interruptions of operations are allowed.) The processing of different operations on the same machine, as well as of any two operations of the same job should not overlap in time. Among the set of *feasible schedules*, which meet the above requirements, we wish to find a schedule that minimizes the maximum job completion time ($C_{\max}$).

This scheduling problem was first posed by Gonzalez and Sahni (1976), who proved that already the three-machine problem $\langle O3 \,||\, C_{\max}\rangle$ is ordinary NP-hard. However, the question on its strong NP-hardness remains open till today.

The NP-hardness of the $\langle O \,||\, C_{\max}\rangle$ problem (that will be further referred to as a *basic problem*) makes topical further investigation of its special cases. One of such special cases is known as a *proportionate open shop problem* (or a *POS-problem*, for short). The specificity of this problem lies in the fact that each job J_j consists of m operations of **the same length** p_j (thus, the operation processing times, $\{p_j^i\}$, are only job-dependent).

In scheduling literature (see, e.g., (Naderi et al., 2014) and (Matta and Elmaghraby, 2010)) this problem is sometimes represented in a dual form, when **each machine** has n equal-length operations, one operation per job, while all jobs are identical, and the operation processing times are only machine-dependent. Ywis, there is nothing unlawful in using such a dual form (in view of the duality of the basic notions, "machines" and "jobs", in the open shop problem). However, when an author uses for that dual form **the same notation** "prpt", this may cause a great jumble. For example, when he claims that "$\langle Om \,|\, prpt \,|\, C_{\max}\rangle$ for any fixed number of machines m can be solved in polynomial time", this sounds strange, because it is well known that already the three-machine problem $\langle O3|prpt|C_{\max}\rangle$ is NP-hard[1].

To avoid such a mishmash in further publications on this problem (yet allowing the author to use the form that he or she prefers), we suggest to distinguish between these two forms by using **different notations** for them: "j-prpt" for the version with job-dependent processing times, and "m-prpt" for the version with machine-dependent processing times. In the current paper, we will strictly stick to the $\langle O \,|\, j\text{-}prpt \,|\, C_{\max}\rangle$ form, shortly denoting it as Z (Zm, when we need to specify the number of machines, m).

We omit here extensive arguments and examples convincing the reader of the "wide practical applicability" of this special case of the Open Shop problem. — The canvass of this issue can be found in scheduling literature dedicated to this problem. Thus, we may entirely concentrate on its theoretical aspects. To our mind, the main reason for theoretical investigation of this special case is the need in a

[1]Ordinary NP-hardness of the 3-machine *POS*-problem can be easily shown by a straightforward reduction from PARTITION. A formal proof of this folklore statement was first presented by Liu and Bulfin (1987). Yet up today, it was not known, if this problem is strongly NP-hard or not.

further clarifying of how the complexity of the basic Open Shop problem depends on various constraints imposed on its parameters.

As an example, we consider the constraints imposed on operation processing times; precisely, — those ones that make the problem a "proportionate" one. It is shown in the paper that $Z3$-problem admits exact solution in a pseudo-polynomial time (and therefore, is not strongly NP-hard), which establishes the **exact complexity status** of this problem.

To continue investigation of the complexity, we introduce the parameter $\gamma^*(I)$ (defined for any instance I) which crucially impacts the problem complexity. Indeed, some **exact boundaries of NP-hardness** of Z-problem have been derived here in terms of values of this parameter.

Among other "positive results" presented in the paper there are both "purely analytic" (such as the investigation of the extremum values of the optimum), and constructive (namely, construction of exact algorithms for polynomially solvable classes of instances, as well as efficient heuristics with worst-case performance guarantees).

2. Basic notions and notation

Here we introduce several notions and notation used in further formulations of results.

$\mathcal{I}_m$ and $\mathcal{I}$ denote the *classes of instances* of the basic problem $\langle O\,||\,C_{\max}\rangle$ in the cases of a fixed and an arbitrary number of machines (m), respectively. Similarly, $\mathcal{I}_{Zm}$ and $\mathcal{I}_Z$ play the same role for problems Zm and Z, respectively. When the set of instances of Zm-problem is restricted to a subset X of admissible values of a fixed problem parameter $x(I)$, we will denote it as $\mathcal{I}_{Zm}\{x(I) \in X\}$ (possible variants are: $\mathcal{I}_{Zm}\{x(I) \le \bar{x}\}$, $\mathcal{I}_{Zm}\{x(I) = \tilde{x}\}$, *etc.*).

Let $N = \{1, \dots, n\}$ be the set of job indices. We denote the *maximum operation processing time* for a given instance I as $p_{\max}(I)$; $ML(I) \doteq \max_{i=1,\dots,m} \sum_{j\in N} p_j^i$ and $JL(I) \doteq \max_{j\in N} \sum_{i=1}^m p_j^i$ denote the *maximum machine load* and the *maximum job length*, respectively; both these amounts (as well as their maximum, $LB(I) \doteq \max\{ML(I), JL(I)\}$), are standard lower bounds on the optimum, $OPT(I)$. Clearly, for any instance I of Zm-problem, $ML(I) = \sum_{j\in N} p_j$, and $JL(I) = mp_{\max}(I)$.

Next, we define the *power of preemption* ($PoP(I)$) as the ratio of $OPT(I)$ to the preemptive optimum of instance I. Due to Gonzalez and Sahni (1976), it is well known that the latter coincides with the lower bound $LB(I)$ for every instance $I \in \mathcal{I}$. Thus, computing (or investigation) of the power of preemption for a given instance I is equivalent to investigation of its "abnormality" defined as the ratio of its non-preemptive optimum to the standard lower bound $LB(I)$. We thereby may further "forget" about the preemptive problem setting, and concentrate on the investigation of the abnormality of instance I, still denoting it as $PoP(I) \doteq OPT(I)/LB(I)$.

Evidently, $PoP(I)$ can never be less than 1. As we also repeatedly saw, property $OPT(I) = LB(I)$ holds not only for every instance I of the preemptive open shop problem, but is, in fact, widespread among instances of the non-preemptive version, as well. (For example, it holds for all instances of the two-machine $\langle O2\,||\,C_{\max}\rangle$ problem, as shown by Gonzalez and Sahni (1976).) Moreover, there is a real feeling that this property holds for the **majority of instances**, which gave ground to Kononov et al. (1999) to introduce the following definitions: an instance I is called *normal*, if $OPT(I) = LB(I)$ (and so, $PoP(I) = 1$), and is called *abnormal* otherwise; a subclass of instances $\mathcal{I}' \subseteq \mathcal{I}$ with the property $PoP(I) = 1$, $\forall\ I \in \mathcal{I}'$, is called *normal*, as well.

In scheduling literature, the notion of "abnormality" of instance I of the non-preemtive Open Shop problem was introduced by Chernykh and Sevastyanov (2017). For estimating the maximum abnormality (or the maximum power of preemption) over all instances, we define here two functions of the number of machines, m (for the basic Open Shop and for Z-problem, respectively):

$$\begin{aligned} PoP[m] &= \sup\{PoP(I) \mid I \in \mathcal{I}_m\}, \\ PoP_Z[m] &= \sup\{PoP(I) \mid I \in \mathcal{I}_{Zm}\}. \end{aligned}$$

We also introduce the parameter $\gamma^*(I) \doteq ML(I)/JL(I)$ being the ratio of the two lower bounds on the optimum and taking the values from the interval $[1/m, \infty)$. As will be seen below, this parameter plays a central role in nearly all our research presented here.

We define the function

$$PoP_{Zm}(\gamma) \doteq \sup_{I \in \mathcal{I}_{Zm}\{\gamma^*(I)=\gamma\}} PoP(I), \tag{1}$$

and denote by

$$\rho_H(I) \doteq \frac{C_{\max}(\sigma_H(I)) - OPT(I)}{OPT(I)} \tag{2}$$

the *relative error* of the solution $\sigma_H(I)$ found by a heuristic H for a given instance I. The *worst case relative error* of heuristic H on a class of instances $\mathcal{I}'$ is defined as

$$\rho_H(\mathcal{I}') \doteq \sup\{\rho_H(I) \mid I \in \mathcal{I}'\}$$

A few other conventional notation for Zm-problem:
— when the instance is fixed, then the argument "(I)" of all parameters showing their dependence on I will be normally omitted;
— the largest job will be often denoted as J_1, resulting in $p_1 = p_{\max} = 1$ (with $p_{\max}$ chosen for the time unit);
— $d(J_j) = mp_j \le m$ will stand for the length of job J_j;
— $P(N'') \doteq \sum_{j \in N''} p_j$ for a subset $N'' \subseteq N$ of job indices;
— $N' = N \setminus \{1\}$ is the set of job indices but job J_1, $\mathcal{J}' \doteq \{J_j \mid j \in N'\}$;
— $P' = P(N') = ML - 1$ will denote the machine load minus p_1.

3. Known and new results

As the reader will ascertain soon, all our results revolve around the problem parameters $LB(I)$ and $\gamma^*(I)$. Indeed, the **relative error** of our heuristics is determined via estimating the deviation of the solution obtained from the lower bound $LB(I)$. Next, the investigation of the most and least favorable cases of the POS-problem is based on the estimation of the relative deviation of the optimum from the lower bound $LB(I)$ (i.e., on the estimation of $PoP(I)$). The **most favorable cases** (at which $PoP(I)$ takes the least possible value, equal to 1) constitute *normal classes* of instances. At that, all the conditions sufficient for the normality of a class of instances (those derived in our paper) are formulated in terms of admissible values of the parameter $\gamma^*(I)$. (As is shown, the bounds on those values obtained are tight, which means that the conditions of normality are not only sufficient, but also necessary.) **Least favorable cases**, at which the relative deviation of the optimum from $LB(I)$ attains its maximum, yield exact values of the function $PoP_Z[m]$. In the case of $m = 3$, it appears that the value of $PoP_Z[3]$ is attained on the instances with values $\gamma^*(I) = 1$.

All normal classes of instances found in the paper appear to be polynomial-time (and quite efficiently) solvable. It should be noted that until today there were no known polynomial-time solvable classes of the POS-problem that were not normal. In this paper, we define first such **polynomial-time solvable classes** of instances of $Z3$-problem that **are not normal**. More precisely, class $\mathcal{I}_{Z3}[\varepsilon]$ is defined for any fixed $\varepsilon > 0$ as the set of all instances $I \in \mathcal{I}_{Z3}$ satisfying the inequality $|\gamma^*(I) - 1| \geq \varepsilon$. Thus, we defined an infinite series of such classes dependent on the parameter ε. These classes are continuously expanding along with decreasing ε, thereby infinitely closely approaching to a "nearly complete" set of instances of $Z3$-problem — "complete" with the exception of the class of instances at which $\gamma^*(I) = 1$. (Exactly on that class of instances the NP-hardness of $Z3$-problem was proved.) At that (as could be easily proposed) the running time of the exact solution of instances from class $\mathcal{I}_{Z3}[\varepsilon]$, while being polynomially bounded (for each fixed ε), exponentially depends on $1/\varepsilon$.

3.1. **Normal and polynomially solvable classes.** First we note that the lower bound $LB(I)$ appears to be quite sharp. To begin with, it is absolutely sharp for the problem version with operation preemption allowed. When preemption is not allowed, it was proved that every instance I of the basic open shop problem meets the strict inequality $PoP(I) < 2$. At that, no one scheduler managed to find an instance with $PoP(I) \geq 1.5$. And even more: as challengingly Sviridenko's conjecture sounds (that for any $I \in \mathcal{I}$, its optimum is never greater than $LB(I) + p_{\max}(I)$), but no counter-example to this conjecture has been found up today (even for arbitrarily large m). At the same time, for the majority of real instances the exact equality $PoP(I) = 1$ is a "normal" event.

It should be noted that the normality property of a class of instances of the Open Shop problem is tightly connected with its polynomial-time solvability. Indeed, all normal classes found up to now appear to be *ps-classes* (which means, solvable in polynomial time). And visa-versa: all known till today ps-classes of the POS-problem appeared to be normal (that is why it was reasonable to call such classes *efficiently normal*, as was done by Kononov et al. (1999)). The only known exception to this rule found by Drobouchevitch and Strusevich (1999) belongs to the two-stage three-machine Open Shop problem which, by definition, cannot be a "proportionate" one. Chronologically, the first normal classes of the Open Shop problem appeared as follows (not counting the two above mentioned classes discovered by Gonzalez and Sahni (1976)). The first such class was found by Fiala (1983). A "pass" for an instance I to enter that class was a condition of the form

$$ML(I) \geq \varphi_t(m) p_{\max}(I), \tag{3}$$

where $\varphi_t(m)$ is a function of the number of machines. In Fiala's version, it was defined as $\varphi_1(m) = 16m'k + 5m'$, where $k = \lceil \log_2 m \rceil$, $m' = 2^k$. Beyond the normality, Fiala's class was also "efficiently normal", admitting the exact solution of every instance in $O(n^2 m^3)$ time.

The parameters of that class (its boundaries and the measure of its efficiency) were improved soon by Bárány & Fiala (1982)[2]: function $\varphi_t(m)$ was diminished nearly twice (being replaced by $\varphi_2(m) = 8m'k + 5m'$) at a much better running time ($O(nm^3)$).

The third version of Fiala's algorithm was presented by Sevast'yanov (1992), who further relaxed on the condition of normality, by considering the "pass-function"

$$\varphi_3(m) = \frac{16}{3}mk + \frac{13}{9}m - \frac{4m}{9m'}(-1)^k,$$

and showing that it guarantees the exact problem solution in time $O(nm^2 \log m)$.

The fourth *ps*-class with a condition of type (3) was defined in (Sevast'janov, 1995) via implementing the compact vector summation technique. It was shown that, given an instance $I \in \mathcal{I}_m$, the condition

$$ML(I) \geq mR(I) + 2(m-1)p_{\max}(I) \tag{4}$$

is sufficient for its normality, where $R(I)$ is the radius of the ball within which $(m-1)$-dimensional vectors $\psi_j = (p_j^1 - p_j^m, \ldots, p_j^{m-1} - p_j^m)$ defined for each job J_j can be summed. By implementing the algorithm from (Sevastyanov, 1991) which guarantees summation of the defined above vectors within the ball of radius

$$R(I) \leq \left(m - 2 + \frac{1}{m-1}\right) p_{\max}(I)$$

in time $O(n^2m^2)$, we obtain one more "pass-function" sufficient for an instance I to be normal:

$$\varphi_4(m) = m^2 - 1 + \frac{1}{m-1}, \tag{5}$$

with the polynomial bound $O(n^2m^2)$ on the time of its exact solution.

Let us use $\langle O3 \,||\, C_{\max}\rangle$ for a test showing how the values of $\varphi_t(3)$ changed in the course of the competition described above:

$$\varphi_1(3) = 148;\ \varphi_2(3) = 84;\ \varphi_3(3) = 37\frac{1}{3};\ \varphi_4(3) = 8\frac{1}{2}.$$

The best known result of this type was obtained in (Sevastianov, 1998), where it was proved that the value $\varphi_5(3) = 7$ specifies an efficiently-normal class of instances. That was accompanied by an exact solution algorithm running in $O(n \log n)$ time (which was further replaced by a linear-time algorithm). However, this is not yet a "point" at the end of this "sentence", because nobody proved that $\varphi_t(3) = 7$ is the least possible value of $\varphi_t(3)$ that guarantees the normality of the corresponding class of instances.

Given $x \geq 0$, let us define the class $\mathcal{I}_m[x] \doteq \{I \in \mathcal{I}_m \mid ML(I) \geq xp_{\max}(I)\}$. Then we have to confess that the problems of finding the exact values of the functions

$$Nor(m) \doteq \inf\{x \mid \mathcal{I}_m[x] \text{ is a normal class}\},$$

$$PS(m) \doteq \inf\{x \mid \mathcal{I}_m[x] \text{ is a } ps\text{-class}\}$$

still remain open for the basic Open Shop problem.

As is quite clear, all the above mentioned results, as well as other results obtained for the basic problem, are as good applicable to all its special cases. It is also clear that general results obtained for the basic problem could be substantially

---

[2]While being written later than Fiala's paper, it however was published earlier, in a local journal (in Hungarian).

improved on, while being applied to special classes of instances. For example, while considering the proportionate case of the Open Shop problem, it can be observed that for any instance $I \in \mathcal{I}_{Zm}$, all vectors ψ_j (defined above) are zero-vectors, and thus, **any order** of their summation puts the summing trajectory into the ball of radius $R(I) = 0$. This means that applying the condition (4) to Zm-problem, we obtain[3] a normal class of type (3) with a function

$$\varphi_6(m) = 2(m-1). \tag{6}$$

The class also appears to be polynomially solvable in time $O(n)$, as will be shown in Section 4.4, where it is proved that function (6) yields the widest normal class $\mathcal{I}_{Zm}[x]$ among those defined by condition (3), thereby providing the exact value of the function $Nor_Z(m) = 2m - 2$. The exact value of $PS_Z(3) = 3$ will also be derived in Section 4.4.

Slightly different sufficient conditions of normality were found in (Koulamas and Kyparisis, 2015) for $Z3$-problem. They showed that either of the two conditions

$$ML(I) \le 2p_1 + p_2,$$
$$ML(I) \ge 3p_1 + p_3$$

is sufficient for the normality of instance $I \in \mathcal{I}_{Z3}$, where the jobs are assumed to be indexed in the non-increasing order of $\{p_j\}$ (and thus, $p_1 = p_{\max}$). They also claimed that two normal classes defined by the above conditions are polynomial-time solvable (although, without providing any bounds on the running time of the correspondent algorithms). As we will make certain later, in both cases optimal solutions can be found in time linear in n (by applying our procedure $\mathcal{P}_{seg}$ — in the second case; the first case is trivial).

In our paper, we found a few new **normal classes** of problem Z, which also appear to be **polynomial-time solvable**. The first such class is defined by the condition

$$ML(I) \le (m-1)p_{\max}(I), \tag{7}$$

which goes well with the sufficient condition

$$ML(I) \ge 2(m-1)p_{\max}(I)$$

mentioned above. Optimal schedules of length $LB(I)$ can be found in $O(n+m\log m)$ time — in the first case, and in $O(n)$ time — in the second case.

Condition (7) can be improved on for small values of m. In particular, it will be shown in Section 4.4 that in the case $m = 3$ condition (7) can be replaced by $ML(I) \le 2.5p_{\max}$, which is tight. And again, the optimal schedule can be found in linear time.

Although it is proved in our paper that the bounds of normal classes found for $Z3$-problem (namely, of classes $\mathcal{I}_{Z3}\{\gamma^*(I) \le \frac{5}{6}\}$ and $\mathcal{I}_{Z3}\{\gamma^*(I) \ge \frac{4}{3}\}$) are tight, they appear to be not tight for the property of polynomial solvability of a class. As we show, the above ps-classes can be further extended up to the classes $\mathcal{I}_{Z3}\{\gamma^*(I) \le 1-\varepsilon\}$ and $\mathcal{I}_{Z3}\{\gamma^*(I) \ge 1+\varepsilon\}$, respectively, for any fixed $\varepsilon > 0$. Let $\mathcal{I}_{Z3}[\varepsilon]$ denote the union of the above two classes. Then the running time of the algorithm solving the instances from $\mathcal{I}_{Z3}[\varepsilon]$ is bounded above by $O(n) + 2^{O(1/\varepsilon)}$. On the one hand,

[3]This fact was also observed by Kyparisis & Koulamas (1997), who applied the result from (Sevast'janov, 1995) to a more general special case of the Open Shop problem, defined by conditions $p_j^i \le p_j^m,\ \forall\, i,j$.

this bound exponentially increases as ε tends to zero. But on the other hand, it is linear in n for any fixed ε, which provides the polynomial-time solvability of the class. Thus, we have an infinite family of polynomial-time solvable classes defined for all positive values of the parameter ε which is set to be fixed for each particular class $\mathcal{I}_{Z3}[\varepsilon]$. This enables us to speak on a *polynomial-time exact solution scheme* (a *PTESS*, analogues to a *PTAS* defined for schemes of approximation algorithms).

3.2. **Optima localization intervals, power of preemption, and efficient heuristics. Efficient heuristics.** It should be noted that, despite the existence of the well known and quite efficient PTASes designed by Sevastianov and Woeginger (1998, 2001) for the basic open shop problem and for its extension to the case of several identical machines at each stage of job processing, such a traditional research direction as designing efficient heuristics (both for the basic problem, and for its special cases, such as Z-problem) still remains urgent, as evidenced by recent publications on the issue. For example, Naderi et al. (2014) presented an efficient heuristic with the worst-case performance guarantee of $2 - \frac{1}{m}$ for Zm-problem (in the form of $\langle Om \,|\, m\text{-}prpt \,|\, C_{\max}\rangle$), while Koulamas and Kyparisis (2015) designed an efficient heuristic for $Z3$-problem with running time $O(n \log n)$ and with the worst-case relative error 1/6. At that, Matta and Elmaghraby (2010) presented an efficient heuristic for the extension of the basic problem to the case of identical parallel machines (considered in the form of $\langle O(P) \,|\, m\text{-}prpt \,|\, C_{\max}\rangle$).

Optima localization intervals and the power of preemption. In (Sevastianov and Tchernykh, 1998), it was proved that for any instance $I \in \mathcal{I}_3$, its optimum belongs to the interval $[LB(I), \frac{4}{3}LB(I)]$, and the bounds of the interval are tight. The tightness of the right bound immediately yields the exact value of the power of preemption for that class: $PoP[3] = \frac{4}{3}$. (As a byproduct of the constructive proof of that theoretical result, a linear-time heuristic H_{ST} was designed which for any instance $I \in \mathcal{I}_3$ could find a schedule $\sigma_{H_{ST}}(I)$ with the absolute bound on its length: $C_{\max}(\sigma_{H_{ST}}(I)) \le \frac{4}{3}LB(I)$, or with the bound $\rho_{H_{ST}}(I) \le \frac{1}{3}$ on its relative error.)

New results. In the present paper, we are also presenting two efficient heuristics with characteristics considerably outperforming the known ones. In fact, all results mentioned in this subsection appear to be tightly connected with each other by the following scheme.

- While considering some case of problem Z (let us call it a Z'-problem), we start with designing an efficient heuristic H' with an absolute performance guarantee of the type:

$$C_{\max}(\sigma_{H'}(I)) \le LB(I) + \lambda_{H'} p_{\max}(I).$$

- Replacing $p_{\max}(I)$ by $JL(I)/m \le LB(I)/m$, we derive the bound on the relative worst-case error of heuristic H', valid for any instance $I \in \mathcal{I}_{Z'}$:

$$\rho_{H'}(I) = \frac{C_{\max}(\sigma_{H'}(I))}{OPT(I)} - 1 \le \frac{C_{\max}(\sigma_{H'}(I))}{LB(I)} - 1 \le \frac{\lambda_{H'} p_{\max}(I)}{LB(I)} \le \frac{\lambda_{H'}}{m}$$

and the optima localization interval:

$$OPT(I) \in [LB(I), (1 + \lambda_{H'}/m)LB(I)].$$

- By presenting an instance $I = I^*$ whose optimum attains the upper bound of the above interval, we thereby prove the tightness of the interval.

• This yields the exact value of the power of preemption for problem Z':

$$PoP_{Z'}[m] = 1 + \lambda_{H'}/m.$$

Let us now describe new results presented in the paper in these directions. First, we design an efficient, $O(n + m\log m)$-time heuristic $\mathcal{H}_{reg}$ that for any m and any instance $I \in \mathcal{I}_{Zm}$ composes a schedule $\sigma_{\mathcal{H}_{reg}}(I)$ of length

$$C_{\max}(\sigma_{\mathcal{H}_{reg}}(I)) < \left(1 + \frac{1}{m}\right) LB(I).$$

(Note that the bound **is not tight**.) This fact has several important consequences.

Firstly, it implies that the relative error, $\rho_{\mathcal{H}_{reg}}(I)$, is not greater than $\frac{1}{m}$, therefore, heuristic $\mathcal{H}_{reg}$ provides asymptotically optimal solutions for Zm-problem, as $m \to \infty$. The relative error also outperforms this characteristic of the algorithm presented in (Naderi et al., 2014). Secondly, it provides the upper bound on the power of preemption:

$$PoP_Z[m] \leq 1 + \frac{1}{m}, \tag{8}$$

which implies that the difference between the optima of preemptive and non-preemptive versions of the POS-problem vanishes as the number of machines infinitely increases.

Another heuristic ($\mathcal{H}_3$) with running time $O(n)$ and the absolute performance guarantee

$$C_{\max}(\sigma_{\mathcal{H}_3}(I)) \leq \frac{10}{9} LB(I)$$

is designed for $Z3$-problem. Accompanied by an instance I whose optimum attains this upper bound, this immediately yields the exact value of the power of preemption: $PoP_Z[3] = \frac{10}{9}$. (In fact, we found the exact function $PoP_{Z3}(\gamma)$ defined for all $\gamma \in [1/3, \infty)$.)

4. Justification of results

4.1. **The exact complexity status of the three-machine problem.** In the following theorem, the exact complexity status of $Z3$-problem is established.

Theorem 1. *There exists an exact pseudo-polynomial time algorithm for the three-machine proportionate open shop problem ($Z3$).*

Proof. Suppose, we have a feasible schedule S. We renumber the machines in the order they process job J_1. Then, looking at the schedule of machine M_2, we can observe that the set of jobs $\mathcal{J}'$ is divided into two parts: the subset $\mathcal{J}_1 \doteq \{J_j \mid j \in N_1\}$ of jobs that are processed on machine M_2 **after** J_1, and the subset $\mathcal{J}_2 \doteq \{J_j \mid j \in N_2\}$ of jobs that are processed on M_2 **before** J_1, thereby defining a partition $\mathcal{P} = \{N_1, N_2\}$ of the index-set N'.

Let now $P2(I, N'')$ denote the problem instance of the $\langle P2||C_{\max}\rangle$ problem, specified for a given instance I of $Z3$-problem by a given subset of indices $N'' \subseteq N'$ as the family of tasks with processing times $\{p_j \mid j \in N''\}$. As feasible solutions for this problem, we consider partitions $\mathcal{P} = \{N_1, N_2\}$ of the index-set N'' with the property $P(N_1) \geq P(N_2)$. The target of the problem is to find a partition $\mathcal{P} = \{N_1, N_2\}$ which minimizes $P(N_1)$. We now present

Algorithm $\mathcal{A}^*$

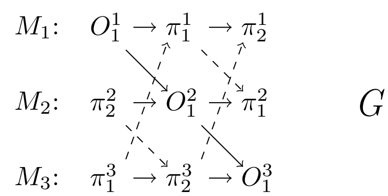

FIG. 1. Network G specifying the order of job processing.

Step 1. Find the optimal solution $\mathcal{P} = \{N_1, N_2\}$ of problem $P2(I, N')$.

Step 2. Let π_1^i, π_2^i denote the blocks of operations of jobs from N_1 and N_2, being continuously processed (i.e., without inner idle times) on the corresponding machine M_i. Let us define schedule S^* as the active schedule that meets precedence constraints depicted on Fig. 4.1 by solid arrows. ■

It can be easily seen that schedule S^* is **feasible**, since the non-simultaneity requirements for the operations of each job $j \in N'$ are met automatically. (The order of their processing obtained from schedule S^* is shown on the picture by dashed arrows.)

Indeed, the starting time of each operation O_j^1 $(j \in N')$ is shifted right with respect to the starting time of operation O_j^3 by the amount $p_1 \geq p_j$, which is sufficient for their non-overlapping. Similarly, O_j^2 $(j \in N_1)$ is shifted right with respect to O_j^1 by the amount $\max\{p_1, P(N_2)\} \geq p_j$, and O_j^3 $(j \in N_2)$ is shifted right with respect to O_j^2 by the amount $P(N_1) \geq P(N_2) \geq p_j$.

Let us show that S^* is **optimal**. It can be seen that the critical path in network G (passing through solid arrows) has one of three possible configurations: (a) passes entirely through a single machine; (b) passes entirely through job J_1; (c) $O_1^1 \to O_1^2 \to \pi_1^2$.

In the first two cases, the optimality of schedule S^* is evident, since $C_{\max}(S^*) = LB$. In the third case, we have $C_{\max}(S^*) = 2p_1 + P(N_1)$, and the optimality of S^* follows from the optimality of the partition $\mathcal{P} = \{N_1, N_2\}$ (since $P(N_1)$ cannot be smaller than obtained).

Since the problem $P2(I, N')$ can be solved in a pseudo-polynomial time (by means of a DP-procedure), the validity of the theorem follows. □

Thus, we may conclude from Theorem 1 that $Z3$-problem, which is known to be ordinary NP-hard, **is not strongly NP-hard**.

For obtaining the majority of the remaining results presented in this paper, we use a simple but efficient "gluing" procedure, which enables us to reduce the size of an instance under consideration (the number of jobs, machines, *etc.*), and thereby, helps substantially reduce the running time of the algorithm[4].

4.2. **The "gluing" procedure.** It should be noted at once that the gluing procedure admits a large number of variants of its realization, and can be applied to a wide range of scheduling problems (and not only). Consequently, presenting a "universal" description of the procedure suitable to all occasions is quite onerous. The one

[4]Another interesting application of this procedure is implementing it in computer-aided proofs of theorems (see, e.g., (Sevastianov and Tchernykh, 1998)).

presented below may be treated just as an example of one possible realization applicable to a multi-stage scheduling problem.

The main destination of the procedure is performing a transformation of the original instance I into a *reduced instance* I^* with a reduced number of elements (such as jobs, machines, *etc.*). At that, we monitor certain *control parameters* of the instance, whose values should be kept within a permitted area, while the values of some other parameters should remain unchanged.

To perform the transformation of a given instance I into a reduced instance I^*, we introduce a variable instance I' with the initial value $I' = I$. Its final value will be taken for I^*. The transformation represents a sequence of *gluing operations* applied to pairs of jobs or pairs of machines. In the case of a **job gluing procedure** (*JG-procedure*) such an operation can be defined as follows.

Assume that we consider an instance I' of a multi-stage scheduling model with a set of jobs $\mathcal{J}'$. Every job $J_j \in \mathcal{J}'$ is characterized by a vector $\bar{p}_j = (p_j^1, \ldots, p_j^m)$ of (total) processing times needed to job J_j for processing on each machine. By "gluing" a pair of jobs $J_j, J_k \in \mathcal{J}'$ we assume creating a new job $J_s \doteq J_j \oplus J_k$ with the vector of processing times $(p_j^1 + p_k^1, \ldots, p_j^m + p_k^m)$. If the new job is *certified* (*i.e.*, its control parameters meet the requirements), we include it to $\mathcal{J}'$ (synchronously eliminating the jobs J_j and J_k from $\mathcal{J}'$).

In the JG-procedure, the *maximum job length* $JL = \max_j d(J_j)$ is taken for the control parameter, keeping in mind that the length of no job in the final instance I^* should exceed a given upper bound (a *threshold*) $\bar{d}$. At that, the load of each machine remains unchanged.

The gluing procedure consists of *iterations*, where at each iteration the whole set of jobs $\mathcal{J}'$ is partitioned into three parts:
— the set $\mathcal{J}_B$ of already "formed" jobs;
— the set $\mathcal{J}_A$ of not yet considered original jobs;
— a *buffer* job J^*.

Iteration

• Take an arbitrary job $J' \in \mathcal{J}_A$ and exclude it from $\mathcal{J}_A$; then try to make a "tentative gluing" of it with job J^*: $J'' := J' \oplus J^*$.

• **If** $d(J'') \le \bar{d}$, **then** $J^* := J''$.

• **Else** [at least one of two inequalities holds: either $d(J') > \bar{d}/2$, or $d(J^*) > \bar{d}/2$]
If $d(J') > \bar{d}/2$, **then** {add job J' to $\mathcal{J}_B$; J^* remains unchanged}

• **Else** [we have $d(J') \le \bar{d}/2$, and so, $d(J^*) > \bar{d}/2$]
{add job J^* to $\mathcal{J}_B$, while J' is taken for the new value of J^*}.

The overall procedure looks as follows:
repeat Iteration **until** $\mathcal{J}_A = \varnothing$.
Find two smallest jobs $J', J'' \in \mathcal{J}' = \mathcal{J}_B \cup \{J^*\}$ and try to make a "tentative gluing" of them: $J''' := J' \oplus J''$. **If** $d(J''') \le \bar{d}$, **then** replace jobs J', J'' in $\mathcal{J}'$ by job J'''.
Put $I^* := I'$. Stop. ■

Let us estimate the number of jobs (n^*) in the final (*reduced*) instance I^*. For certainty, let us consider the version of the JG-procedure with the threshold $\bar{d} = LB(I)$.

Lemma 1. *The JG-procedure with a threshold $\bar{d}$ results in an instance I^* in which no two jobs can be glued together without violating $\bar{d}$. While being applied with a* *threshold $\bar{d} = LB(I)$ to an instance I of a multi-stage scheduling problem, it keeps unchanged the value of the standard lower bound on the optimum ($LB(I)$), and the number of jobs (n^*) in I^* does not exceed $(2m-1)$. Application of the JG-procedure to the basic Open Shop problem takes $O(nm)$ time, and only $O(n)$ time — when applied to the POS-problem.*

Proof. By the definition of the standard lower bound $LB(I) = \max\{ML(I), JL(I)\}$, the original instance I cannot contain jobs with length $d(J_j) > LB(I)$. The choice of the control parameter $\bar{d}$ ensures also that no new job J' with length $d(J') > LB(I)$ may appear in the current instance I' at any iteration of JG-procedure. Thus, the value of $JL(I')$ cannot exceed $LB(I)$. Since it is clear that any job gluing operation retains all machine loads unchanged, the value of $ML(I')$ remains unchanged, as well. Thus, the value of $LB(I')$ keeps stable.

Let us prove that the number of jobs in I^* is not greater than $2m-1$. At the end of the loop on iterations, the whole set of jobs consists of jobs from $\mathcal{J}_B$ and of job J^*. If $|\mathcal{J}_B| \leq 2m-2$, then we are done. Suppose, we have $|\mathcal{J}_B| \geq 2m-1$ (and let us make sure that this cannot be). Ok, we know that every job in $\mathcal{J}_B$ has length $d(J) > LB(I)/2$. Thus, when (at the end of an iteration t) we put into $\mathcal{J}_B$ the $(2m-2)$th job, the total length of jobs in $\mathcal{J}_B$ becomes greater than $(m-1)LB(I)$, and so, the total length of the remaining jobs does not exceed $LB(I)$. This means that the situation with $d(J'') > \bar{d} = LB(I)$ cannot appear in the algorithm any more, and so, the $(2m-1)$th job could not be put into $\mathcal{J}_B$ at the subsequent iterations. — A contradiction.

As we could see, gluing of any two jobs from $\mathcal{J}_B$ exceeds $LB(I)$. Thus, the only possibility for a further reduction of instance I' is gluing together job J^* (if $d(J^*) < \bar{d}$) and a job from $\mathcal{J}_B$. (A reasonable idea is to choose the smallest job from $\mathcal{J}_B$.) If such a gluing appears to be successful (which means, meets $\bar{d}$), this is, clearly, the last possible reduction of instance I'.

Finally, since the gluing operation of two jobs in the case of basic Open Shop problem requires $O(m)$ time (and $O(1)$ time for the POS-problem), the complete JG-procedure can be performed in $O(nm)$ and $O(n)$ time, respectively. □

4.3. **Basic algorithms.**

Procedure $\mathcal{P}_{seg}$

Step 1. Find the job J_{j^*} with $p_{j^*} = p_{\max}(I)$. Number the machines in an arbitrary order (it will correspond to the order of processing the operations of job J_{j^*} in schedule σ_{seg}).

Step 2 (defining the job order on machine M_1). To specify the job order on machine M_1, we take an arbitrary permutation π_1 of job indices starting with j^*. We then renumber the jobs according to π_1, thus obtaining $\pi_1 = (1, \ldots, n)$, with $p_1 = p_{\max}(I)$.

Step 3 (defining the job orders on machines M_i, $i = 2, \ldots, m$). The order π_i of jobs on each machine M_i, $i = 2, \ldots, m$, is defined as a cyclic shift of permutation π_{i-1} forward by a few indices representing a final segment $\widehat{\pi}_{i-1}$ of permutation π_{i-1}. Thus, $\pi_i := \widehat{\pi}_{i-1} \circ (\pi_{i-1} \setminus \widehat{\pi}_{i-1})$, where "$\circ$" denotes a concatenation of sequences. For $\widehat{\pi}_{i-1}$ we take the minimal final segment $\widehat{\pi}$ of π_{i-1} having *length* $P(\widehat{\pi}) \doteq \sum_{j \in \widehat{\pi}} p_j \geq p_1$.

If it appears that for some i segment $\widehat{\pi}_{i-1}$ has to include job J_1, we **stop** the process and output the conclusion: "The algorithm failed to construct a complete feasible schedule". (This case will be referred to as "unsuccessful".) In the alternative ("successful") case, we define all m job permutations $\pi_i,\ i = 1, \ldots, m$ and go to

Step 4 (defining a schedule). Schedule σ_{seg} is defined by specifying for each machine M_k the index j_k of the first job that should be started at time 0. It is assumed that all subsequent jobs should be processed by the machine according to permutation π_1 (cyclically shifted left by $j_k - 1$ indices), without any idle time. ■

For convenience of its further implementation, let us give an explicit definition to what we mean by the "successful case" in procedure $\mathcal{P}_{seg}$.

Definition 1. *The "successful case" in procedure $\mathcal{P}_{seg}$ occurs when, given defined a permutation π_1 of job indices (specifying the order of processing the jobs on machine M_1 and starting with job "1"), we are able to conform consecutively $(m-1)$ segments (from the end to the beginning of permutation π_1), not including job "1" and each having length at least p_1.*

Lemma 2. *Procedure $\mathcal{P}_{seg}$ runs in $O(n)$ time, and in the "successful case", computes a normal (and so, optimal) schedule σ_{seg}.*

Proof. The bound on the running time is evident. (Note that the "successful case" cannot appear, if $n < m$.) To guarantee the feasibility of the schedule in the "successful case", it is sufficient to show that no two operations of the same job overlap in time. To make certain of that, let us investigate the structure of schedule σ_{seg}, which can be seen in Table 1.

Machine	Job sequence					
M_1	(1)	X	$\widehat{\pi}_{m-1}$	$\widehat{\pi}_{m-2}$	$\ldots$	$\widehat{\pi}_1$
M_2	$\widehat{\pi}_1$	(1)	X	$\widehat{\pi}_{m-1}$	$\ldots$	$\widehat{\pi}_2$
M_3	$\widehat{\pi}_2$	$\widehat{\pi}_1$	(1)	X	$\ldots$	$\widehat{\pi}_3$
$\vdots$			$\bullet$	$\bullet$	$\bullet$	
M_{m-1}	$\widehat{\pi}_{m-2}$	$\ldots$	$\widehat{\pi}_1$	(1)	X	$\widehat{\pi}_{m-1}$
M_m	$\widehat{\pi}_{m-1}$	$\widehat{\pi}_{m-2}$	$\ldots$	$\widehat{\pi}_1$	(1)	X

TABLE 1. Schedule σ_{seg}

While comparing the schedules on two consecutive machines, M_{i-1} and M_i, we can see that all operations on machine M_i (except the segment $\widehat{\pi}_{i-1}$) are shifted to the right with respect to the operations of the same jobs on machine M_{i-1} by the amount of $P(\widehat{\pi}_{i-1}) \geq p_{\max}(I)$. Similarly, operations on machine M_1 are shifted to the right with respect to the operations on machine M_m by the amount of $p_{\max}(I)$. This guarantees the non-overlapping of all operations belonging to the same job. Thus, the schedule is feasible and normal, which implies its optimality. □

Let JG^* denote the special case of the JG-procedure with $\bar{d} = JL(I)$. As can be seen, in this case the values of JL and $p_{\max}$ are kept unchanged. Although, in this case we cannot guarantee the bound $n^* \leq (2m-1)$ on the number of jobs in the reduced instance I^* directly (by applying Lemma 1), however, as will be shown below, this, and even better bound $n^* \leq (2m-2)$ **can be assumed, w.l.o.g.**

Suppose the contrary: that the final instance I^* contains $n^* \geq 2m-1$ jobs. Among those jobs, there is the largest job J_1 (with length $d(J_1) = \bar{d}$), and there are at least $(2m-2)$ jobs distinct from J_1 such that the sum of any two is greater than $\bar{d}$ (by Lemma 1). If we put $(2m-2)$ such jobs to the end of permutation π_1 in the procedure $\mathcal{P}_{seg}$, then we definitely have a "successful case", because the segmentation procedure is able to compose from those "lengthy" jobs at least $(m-1)$ segments of length at least p_1 each one. Thus, due to Lemma 2, procedure $\mathcal{P}_{seg}$ in this "contrary" case builds an optimal (and normal) schedule in $O(n)$ time.

Furthermore, the case when the number of glued jobs is not greater than m also represents no problem for constructing a normal schedule (of length JL) in the evident way. This enables us to make the following

Assumption 1. *Without loss of generality of the problem under solution, we may further assume that for any given instance I of the POS-problem, a reduced instance I^* with $n^* \in [m+1, 2m-2]$ jobs and with the original values of the parameters $ML, JL, LB, p_{\max}$, and $\gamma^*(I)$ can be obtained by implementing the JG^*-procedure.*

(The assumption is valid since we figured out above that in the opposite case, when for the reduced instance I^*, obtained by the JG^*-procedure for a given instance I, its number of jobs n^* does not fall into the interval $[m+1, 2m-2]$, the exact solution can be found in linear time by implementing the $\mathcal{P}_{seg}$-procedure.)

Let us next describe a heuristic which, beyond its direct destination (obtaining approximate solutions with a bounded error), will be also used for deriving some exact results.

Heuristic $\mathcal{H}_{reg}$

Step 1 (gluing the jobs). Applying the JG^*-procedure, we either find an optimal schedule (by applying the $\mathcal{P}_{seg}$ procedure to the glued instance I^*; at that, heuristic $\mathcal{H}_{reg}$ stops), or transform the given instance I to a reduced instance I^* with $n^* \leq 2m-2$ jobs. At that, the values of the parameters ML, JL, LB, and $p_{\max}$ remain unchanged.

Step 2 (defining the job order on machine M_1). Let us take an arbitrary numbering of machines. Then for the job order on machine M_1 we take $\pi_1 = (1, \ldots, n^*)$, where $p_1 \geq p_2 \geq \cdots \geq p_{n^*}$.

Step 3 (defining the job orders on machines $M_i,\ i = 2, \ldots, m$). The order π_i of jobs on each machine $M_i,\ i = 2, \ldots, m$, will be defined as a cyclic shift of π_1 forward by a few indices. To determine the size of that shift, we do the following.

Let $D \doteq \max\{ML(I) + p_1, JL(I)\}$ be chosen for a *deadline* on the completion of all jobs. Let us first consider the "flow shop" schedule σ in which each machine M_i processes the jobs without an idle time in order $\pi_i = \pi_1$, starting at the time moment $(i-1)p_1$ (*i.e.*, schedule σ looks like a "regular staircase"). As can be easily seen, each job starts on machine M_i after it is completed on machine M_{i-1}, and thus, the schedule is feasible.

The set of jobs completed on M_i in schedule σ after time D constitutes a final segment $\bar{\pi}_i$ of permutation π_i. We define a new job sequence π'_i for machine M_i as

$$\pi'_i := \bar{\pi}_i \circ (\pi_i \setminus \bar{\pi}_i),$$

and define a new schedule σ' by processing the jobs from $\bar{\pi}_i$ as a "block" (without inner idle time) starting at time 0, while retaining the block of jobs $\pi_i \setminus \bar{\pi}_i$ as in σ.

We note that the above description uniquely defines the schedule, and that we need not specify for that purpose the starting times of all operations. ■

Theorem 2. *Heuristic $\mathcal{H}_{reg}$ runs in $O(n + m \log m)$ time, and for any instance I of problem Zm builds a feasible schedule σ' such that*

$$C_{\max}(\sigma') \le \max\{ML(I) + p_{\max}(I), JL(I)\}.$$

Proof. The bound on the running time, as well as the bound on schedule length are evident. Let us prove the feasibility of schedule σ'. First, let us show that each block $\bar{\pi}_i$ fits into the interval $[0, (i-1)p_1]$. Indeed, since in schedule σ all jobs from $\bar{\pi}_i$ complete after time D, they start after time $(D - p_1)$, so, $P(\bar{\pi}_i) < ML + (i-1)p_1 - D + p_1 \le (i-1)p_1$. This provides the "machine-feasibility" of schedule σ'. Next, let us be certain of its "job-feasibility".

The job-feasibility of blocks $\pi_i \setminus \bar{\pi}_i$ with respect to each other follows from that of schedule σ (which is feasible). Let us show the job-feasibility of blocks $\bar{\pi}_i$ in σ'. Let $j'(i)$ denote the index of the first job of block $\bar{\pi}_i$. Since it is the first job violating the deadline D on machine M_i in schedule σ, it completes not later than at time $D + p_1$ on machine M_i, and not later than at time D on machine M_{i-1}, which means that $j'(i)$ is not included in block $\bar{\pi}_{i-1}$. Thus, due to ordering the jobs in sequence π_i in the non-increasing order of $\{p_j\}$, the length of job $j'(i)$ is not less than that of any job $j \in \bar{\pi}_{i-1}$. Since the σ'-schedule of jobs from $\bar{\pi}_{i-1}$ on machine M_i is shifted with respect to that on machine M_{i-1} by at least the length of job $j'(i)$, this provides the job-feasibility of blocks $\bar{\pi}_i$ in σ'.

The proof of job-feasibility of schedule σ' is concluded by the evident observation that the schedule on M_1 of jobs from $\bar{\pi}_m$ is shifted with respect to the schedule of these jobs on machine M_m by at least the amount p_1, which is sufficient for its job-feasibility. □

Corollary 1. *By Sviridenko's conjecture (known for decades in the scheduling community), optimum of any instance I of the basic $\langle O \,||\, C_{\max}\rangle$ problem meets the bound*

$$OPT(I) \le LB(I) + p_{\max}(I).$$

As follows from Theorem 2, the length of schedule σ' built by heuristic $\mathcal{H}_{reg}$ meets the above bound, which confirms Sviridenko's conjecture for the case of the POS-problem.

Corollary 2. *Heuristic $\mathcal{H}_{reg}$ runs in $O(n + m \log m)$ time, and for any instance I of problem Zm provides a solution $\sigma_{\mathcal{H}_{reg}}(I)$ with the relative error: $\rho_{\mathcal{H}_{reg}}(I) \le 1/m$. The solution obtained becomes asymptotically optimal, as m tends to infinity.*

4.4. **Normal and polynomially solvable classes.** Two normal classes of problem Zm are defined for any m in terms of the parameter $\gamma^*(I)$:

$$\begin{aligned} &\mathcal{I}_{Zm}\{\gamma^*(I) \ge 2 - 2/m\} = \{I \in \mathcal{I}_{Zm} \mid \gamma^*(I) \ge 2 - 2/m\} \\ &\mathcal{I}_{Zm}\{\gamma^*(I) \le 1 - 1/m\} = \{I \in \mathcal{I}_{Zm} \mid \gamma^*(I) \le 1 - 1/m\} \end{aligned} \tag{9}$$

Theorem 3. *Subclass of instances $\mathcal{I}_{Zm}\{\gamma^*(I) \ge 2 - 2/m\}$ of Zm-problem is normal and solvable in $O(n)$ time by the $\mathcal{P}_{seg}$-procedure. The condition providing the normality of the class is tight for any $m \ge 3$.*

Proof. To prove the first statement of the theorem, it is sufficient to show that the condition defining the class provides a "successful case" for the $\mathcal{P}_{seg}$-procedure. Let us take $p_{\max}$ for the time unit. Then the defining condition takes the form: $ML \geq 2m-2$. Let $N' = \{2, \ldots, n\}$ denote the set of indices of all jobs minus J_1, then $P' \doteq \sum_{j \in N'} p_j \geq 2m-3$. That much machine load is sufficient to form $m-2$ segments $\widehat{\pi}_i$ of length ≥ 1 (since the length of each is less then 2), while the remaining load is enough to form one more segment of length ≥ 1.

To prove the tightness of the condition $ML \geq 2m-2$, it is sufficient to find (for any $m \geq 3$) an abnormal instance I^* with $ML(I^*) = 2m-2-\varepsilon$ for any $\varepsilon \in (0,1]$. The desired instance consists of job J_1 and of $2m-3$ equal-length "small" jobs with $p_j = 1 - \varepsilon/(2m-3)$. Since $ML(I^*) \geq JL(I^*)$, the normal schedule should have length $ML(I^*)$ and should have no idle time.

Assume that there exists a normal schedule for the instance I^*. Number the machines in accordance with the order of processing the operations of job J_1. Then, to provide a non-overlapping of the operations of job J_1, at least 2 "small" operations should precede the operation of job J_1 on machine M_2, at least 4 "small" operations — on machine M_3, and so on, at least $2(m-1)$ "small" jobs should be on machine M_m, but we have only $2m-3$. □

As a one more direct corollary of Theorem 2, we can formulate

Theorem 4. *Subclass of instances $\mathcal{I}_{Zm}\{\gamma^*(I) \leq 1 - 1/m\}$ of Zm-problem is normal. Heuristic $\mathcal{H}_{reg}$ finds an optimal schedule of length $JL(I)$ for any instance $I \in \mathcal{I}_{Zm}\{\gamma^*(I) \leq 1 - 1/m\}$ in $O(n + m \log m)$ time.* □

We note that the condition providing the normality of the class $\mathcal{I}_{Zm}\{\gamma^*(I) \leq 1 - 1/m\}$ **is not tight**. It can be tightened for small values of m.

Theorem 5. *Class of instances $\mathcal{I}_{Z3}\{\gamma^*(I) \leq \frac{5}{6}\}$ is normal, and the condition of normality is tight. There is a simple algorithm of running time $O(n)$ which for any instance $I \in \mathcal{I}_{Z3}\{\gamma^*(I) \leq \frac{5}{6}\}$ builds its optimal schedule of length $JL(I)$.*

Proof. Applying the JG^*-procedure, we obtain an instance I^* with $n^* \leq 3$ jobs, and we are done. Let us make sure that the cases with $n^* = 4$ or $n^* = 5$ jobs are impossible. Assuming that $n^* = 4$, number the jobs so as $p_1 \geq \cdots \geq p_4$. Under the conditions defining this class of instances, we have $ML \leq 2.5$ and $P' = ML - p_1 \leq 1.5$, which implies that $p_3 + p_4 \leq 1$. So, these two jobs must be glued together in the JG^*-procedure — a contradiction. The case $n^* = 5$ is treated similarly.

The tightness of the condition can be demonstrated by the instance I^* having, beyond job J_1, three "small" jobs with $p_2 = p_3 = p_4 = 0.5 + \varepsilon$. In this case, $ML(I^*) = 2.5 + 3\varepsilon$, $JL(I^*) = 3$, and $\gamma^*(I^*) = \frac{5}{6} + \varepsilon$, and a feasible normal schedule does not exist. The running time of this algorithm is the time needed for job gluing, i.e., $O(n)$. □

Until now, there were only known polynomial-time solvable classes of instances of the POS-problem that were normal. The following theorem presents an infinite series of such polynomial-time solvable classes that provably include infinitely many abnormal instances.

Theorem 6. *For any fixed $\varepsilon > 0$, class $\mathcal{I}_{Z3}[\varepsilon] \doteq \mathcal{I}_{Z3} \backslash \mathcal{I}_{Z3}\{\gamma^*(I) \in (1-\varepsilon, 1+\varepsilon)\}$ is solvable in time $O(n) + 2^{O(1/\varepsilon)}$. Thus, each class $\mathcal{I}_{Z3}[\varepsilon]$ is polynomial-time solvable.*

Proof.

Algorithm $\mathcal{A}_\varepsilon$

Any instance $I \in \mathcal{I}_{Z3}[\varepsilon]$ meets one of the two conditions: either (A) $ML(I) \le 3 - 3\varepsilon$, or (B) $ML(I) \ge 3 + 3\varepsilon$. Put $\Delta \doteq |ML(I) - 3|$. Then $\Delta \ge 3\varepsilon$.

Step 1. Find the subset of jobs $N'' = \{j \in N' \mid p_j > \Delta\}$ and define its subset:

$$\widetilde{N} := \begin{cases} \text{any minimal by inclusion subset } \widetilde{N} \subseteq N'' \text{ such that } P(\widetilde{N}) \ge 2, & \\ & \text{if } P(N'') \ge 2; \\ N'', & \text{otherwise.} \end{cases}$$

(So, we have $\widetilde{N} = N''$ in case (A).) Clearly, $|\widetilde{N}| \le \lceil \frac{2}{\Delta} \rceil < \frac{2}{3\varepsilon} + 1$.

Step 2. Find an optimal solution $\mathcal{P} = \{N_1, N_2\}$ of the problem $P2(I, \widetilde{N})$ (defined in the proof of Theorem 1, page 414) in $2^{O(|\widetilde{N}|)} \le 2^{O(\frac{1}{\varepsilon})}$ time.

Step 3. If $P(N_1) \ge 1$, put $N_2 := N' \backslash N_1$ and take the schedule S^* (defined by the partition $\mathcal{P} = \{N_1, N_2\}$ in the algorithm $\mathcal{A}^*$, page 414). STOP.

Step 4. Alternatively, if $P(N_1) < 1$, then we proceed with competing both "heaps", N_1 and N_2, by the rest of jobs from $N' \backslash (N_1 \cup N_2)$ (in an arbitrary order) according to the principle: put the next in turn job to the smaller heap. If in the resulting partition $\mathcal{P}' = \{N'_1, N'_2\}$ both heaps have weights $P(N'_i) \le 1$, then we have no problem with constructing a normal schedule of length $JL(I) = 3$. Alternatively, if one of the heaps has weight $P(N'_i) > 1$ (we may assume, w.l.o.g., that it is N'_1), we use the partition $\mathcal{P}' = \{N'_1, N'_2\}$ for constructing the schedule S^* according to the algorithm $\mathcal{A}^*$. STOP. ■

Let us prove that schedule S^* is optimal in any case.

If STOP occurs at Step 3, when we have $P(N_1) \ge 1$, then there may be two cases. If $P(N_2) \ge 1$, we have case (a) of the critical path in schedule S^*, resulting in $C_{\max}(S^*) = LB(I)$. Alternatively, if $P(N_2) < 1$, we have case (c) of the critical path, when $C_{\max}(S^*) = 2 + P(N_1)$, and this schedule length, clearly, cannot be smaller. Thus, in both cases we have the optimum.

If STOP occurs at Step 4, when $P(N_1) < 1$, then we have $\widetilde{N} = N''$. After finishing the completing of heaps, we again may have two cases. In one case (when both $P(N'_1) \le 1$ and $P(N'_2) \le 1$), as said above, we are done. In the alternative case (when $P(N'_1) > 1$), the difference in weights of heaps $P(N'_i)$ is, clearly, not greater than the weight of the last job (J_j) put into the largest heap (we assume that it is N'_1). Since this job is not in $\widetilde{N}$ (and hence, is not in N''), it has the weight $p_j \le \Delta$, which implies: $P(N'_2) \ge P(N'_1) - \Delta > 1 - \Delta$, whence

$$ML = P(N'_1) + P(N'_2) + 1 > 3 - \Delta.$$

Since we defined $\Delta \doteq |ML(I) - 3|$, but $\Delta \ne 3 - ML$, we deduce that $\Delta = ML - 3$. Thus,

$$2 + \Delta = ML - 1 = P(N'_1) + P(N'_2) = 2P(N'_2) + (P(N'_1) - P(N'_2)) \le 2P(N'_2) + \Delta,$$

whence $P(N'_2) \ge 1$. Since both heaps (N'_1 and N'_2) have weights ≥ 1, schedule S^* is an optimal (normal) schedule of length ML. □

It should noted that the value of ε is nowhere used in the description of the algorithm $\mathcal{A}_\varepsilon$. (So, the algorithm is applicable to ANY instance $I \in \mathcal{I}_{Z3}$.) It is used only for estimating its running time, when the set of instances is restricted to class $\mathcal{I}_{Z3}[\varepsilon]$.

4.5. **Efficient heuristic and the power of preemption.** It becomes quite evident that job gluing procedure preserving the lower bound on the optimum represents a convenient tool for investigation of the power of preemption, since the gluing of jobs definitely does not improve the optimum of an instance.

Such a gluing is also convenient for estimating the accuracy of a heuristic, if that accuracy is measured in terms of the same (unchangeable while gluing) lower bound on the optimum. (Not mentioning that the gluing decreases the running time of the heuristic.)

In the case of applying the JG^*-procedure to Z-problem, we are also guaranteed the invariability of the parameter $\gamma^*(I)$, which provides an additional possibility to investigate the dependence of both the function $PoP_{Z3}(\gamma)$, and the accuracy of the heuristic on the value of that parameter.

Three above mentioned factors are combined in this section for a joint analysis of two things: a heuristic $\mathcal{H}_m$ and the function $PoP_Z(\gamma)$ estimating the dependence of the power of preemption on the parameter $\gamma^*(I)$. We first describe

Heuristic $\mathcal{H}_m$

Step 1 (gluing the jobs). Applying the JG^*-procedure to a given instance I, we either find an optimal (and normal) schedule (at that, heuristic $\mathcal{H}_m$ stops), or obtain a reduced instance I^* with $n^* \in [m+1, 2m-2]$ jobs (due to Assumption 1). At that, the values of the parameters $ML, JL, LB, p_{\max}$, and γ^* remain unchanged.

Step 2. Find an optimal schedule σ^* for the instance I^* in time $O(T(m))$.

Step 3. Transform the schedule σ^* into a schedule $\sigma_{\mathcal{H}_m}$ for the original instance I by presenting each glued operation as a block of original operations. ■

Theorem 7. *For any instance $I \in \mathcal{I}_{Zm}$, heuristic $\mathcal{H}_m$ constructs a schedule $\sigma_{\mathcal{H}_m}$ with length $C_{\max}(\sigma_{\mathcal{H}_m}) \leq PoP_{Zm}(\gamma^*(I)) \cdot LB(I)$ (where function $PoP_{Zm}(\gamma)$ is defined in (1)) and with the relative error $\rho_{\mathcal{H}_m}(I) \leq PoP_{Zm}(\gamma^*(I))-1$. The running time of the heuristic is $O(n + T(m))$.*

Proof. The bound on schedule length follows from this chain of relations:

$$\begin{aligned} C_{\max}(\sigma_{\mathcal{H}_m}) &\stackrel{1}{=} OPT(I^*) \stackrel{2}{=} \frac{OPT(I^*)}{LB(I^*)} \cdot LB(I) \stackrel{3}{=} PoP(I^*) \cdot LB(I) \\ &\stackrel{4}{\leq} \sup_{I' \in \mathcal{I}_{Zm}\{\gamma^*(I')=\gamma^*(I)\}} PoP(I') \cdot LB(I) \stackrel{5}{=} PoP_{Zm}(\gamma^*(I)) \cdot LB(I). \end{aligned}$$

Let us justify each relation.

The first equality is a consequence of two simple facts:

(a) The length of the resulting schedule $\sigma_{\mathcal{H}_m}$ (defined for the original instance I) coincides with that of schedule σ^* built for the reduced instance I^*.

(b) Schedule σ^* is the **optimal schedule** of instance I^*.

Equality "2" just confirms the fact that JG^*-procedure preserves the parameter $LB(I)$ of a glued instance.

Equality "3" follows from the definition of $PoP(I)$ and the result by Gonzalez and Sahni.

Inequality "4" is evident, provided that JG^*-procedure preserves the parameter $\gamma^*(I)$.

Equality "5" reminds the definition of the function $PoP_{Zm}(\gamma)$.

The bound on the relative error just uses its definition (2). Finally, the bound on the running time is straightforward. □

To compute the function $PoP_{Z3}(\gamma)$ (defined for $Z3$-problem and $\gamma \in [1/3, \infty)$), we should detail the algorithm of constructing the optimal schedule for the instance I^* (at Step 2 of heuristic $\mathcal{H}_m$) in the case of $m = 3$. We should also analyze how that optimal schedule looks like, and estimate its length in the worst case.

Heuristic $\mathcal{H}_3$

Step 2 (finding an optimal schedule σ^* for the instance I^*). In the case of three machines, the only possible value of n^* at which we go to the second step of heuristic $\mathcal{H}_3$ is $n^* = 4$. In this case we proceed as follows.

We take an arbitrary numbering of machines, and number the jobs so as $p_1 \geq \cdots \geq p_4$. Next, $\pi_1 = (1, 2, 3, 4), \pi_2 = (3, 4, 1, 2)$ and $\pi_3 = (2, 3, 4, 1)$ are taken for the job orders on machines M_1, M_2, and M_3. Then we compute the active schedule σ^* respecting the prescribed job orders on machines. This schedule is, clearly, optimal for the instance I^*. ■

Theorem 8.

$$PoP_{Z3}(\gamma) = \begin{cases} 1, & \textit{for } \gamma \in \left[\frac{1}{3}, \frac{5}{6}\right]; \\ \frac{2\gamma}{3} + \frac{4}{9}, & \textit{for } \gamma \in \left[\frac{5}{6}, 1\right]; \\ \frac{2}{3} + \frac{4}{9\gamma}, & \textit{for } \gamma \in \left[1, \frac{4}{3}\right]; \\ 1, & \textit{for } \gamma \in \left[\frac{4}{3}, \infty\right). \end{cases}$$

Proof. Due to Theorems 5 and 3, any instance $I \in \mathcal{I}_{Z3}$ with $\gamma^*(I) \leq 5/6$ or $\gamma^*(I) \geq 4/3$ is normal, so, it has $PoP(I) = 1$, which yields the values of the function $PoP_{Z3}(\gamma)$ in its first and fourth intervals. Let us now consider the second and the third cases: $\gamma \in \left[\frac{5}{6}, \frac{4}{3}\right]$.

The cases, when the original instance I is transformed (at Step 1 of heuristic $\mathcal{H}_3$) into an instance I^* with $n^* \leq 3$ or $n^* = 5$ jobs (all those cases appear to be normal, and so, heave $PoP(I^*) = 1$), definitely, do not represent the worst cases, at which the value of $PoP_{Z3}(\gamma)$ is attained. Thus, it remains to consider the case with $n^* = 4$ jobs.

Taking $p_{\max}(I)$ for the time unit, we have $ML = JL \cdot \gamma = 3\gamma$, and $P' = 3\gamma - 1$. Since, in this case we have $p_3 + p_4 > 1$, the critical path in schedule σ^* passes through the operations $O_3^2, O_4^2, O_1^2, O_1^3$ and has length $p_3 + p_4 + 2$, so, $OPT(I^*) = p_3 + p_4 + 2$. The maximum of this amount is attained, when $p_3 + p_4 = \frac{2}{3}P' = \frac{2}{3}(3\gamma - 1)$, and is equal to $2\gamma + \frac{4}{3}$.

To compute $PoP(I^*) = OPT(I^*)/LB(I^*)$, we should recall that

$$LB(I^*) = \begin{cases} JL(I^*) = 3, & \text{for } \gamma \leq 1; \\ ML(I^*) = 3\gamma, & \text{for } \gamma \geq 1. \end{cases}$$

This yields the values of $PoP_{Z3}(\gamma)$ in the cases of $\gamma \in \left[\frac{5}{6}, 1\right]$ and $\gamma \in \left[1, \frac{4}{3}\right]$, respectively. □

Corollary 3. *Function $PoP_{Z3}(\gamma)$ attains its maximum, $PoP_Z[3] = \frac{10}{9}$, on an instance I with $\gamma^*(I) = 1$ (i.e., when both standard lower bounds on the optimum coincide).*

5. Conclusion

Further investigation of the proportionate Open Shop problem could be performed in the following directions:

(1) Compute the exact values of the function $PoP_Z[m]$ for any $m > 3$.

(1a) Find $PoP_Z \doteq \sup_m PoP_Z[m]$. (Is it true that $PoP_Z = PoP_Z[3]$?)

(2) The heuristic $\mathcal{H}_{reg}$ described in Section 4.3, while being asymptotically optimal, is not tight (which was convincingly demonstrated by presenting the $\mathcal{H}_3$-heuristic for $Z3$-problem with a significantly better relative error). Is it possible to design a polynomial-time heuristic (not a scheme!) for Zm-problem with a relative error less than $1/m$?

And of course, two challenging questions on the complexity status of the POS-problem still remain open:

Q1: Is it true that Zm-problem admits exact solution **in pseudo-polynomial time for any fixed** m?

Q2: Is it true that problem Z (when both n and m are parts of the input) is strongly NP-hard?

References

[1] Bárány, I., and Fiala, T., *Nearly optimum solution of multimachine scheduling problems*, Szigma, **15**:3 (1982), 177–191 (in Hungarian). MR0694207

[2] I. Chernykh, S. Sevastyanov, *On the abnormality in open shop scheduling*, In: 17th Baikal International Triannual School-Seminar "Methods of Optimization and their Applications", July 31—August 6, 2017, Maksimikha, Buryatia (2017), URL: `http://isem.irk.ru/conferences/mopt2017/Abstracts/BAIKAL\_2017\_paper\_95.pdf`

[3] I.G. Drobouchevitch, V.A. Strusevich, *A polynomial algorithm for the three-machine open shop with a bottleneck machine*, Ann. Oper. Res., **92** (1999), 185–214. MR1730744

[4] T. Fiala, *An Algorithm for the Open-Shop Problem*, Mathematics of Operations Research, **8**:1 (1983), 100—109. MR0703829

[5] T. Gonzalez, S. Sahni, *Open shop scheduling to minimize finish time*, J. Assoc. Comput. Mach., **23**:4 (1976), 665–679. MR0429089

[6] Kononov, A., Sevastianov, S. and Tchernykh, I., *When difference in machine loads leads to efficient scheduling in open shops*, Annals of Operations Research, **92** (1999), 211–239. MR1730745

[7] Koulamas, C. and Kyparisis, George J., *The Three-Machine Proportionate Open Shop and Mixed Shop Minimum Makespan Problems*, European Journal of Operational Research, **243**:1 (2015), 70–74. MR3300910

[8] G.J. Kyparisis, C. Koulamas, *Open shop scheduling with maximal machines*, Discrete Applied Mathematics, **78** (1997) 175–187. MR1475825

[9] C.Y. Liu, R.L. Bulfin, *Scheduling ordered open shops*, Computers and Operations Research, **14** (1987), 257–264. MR0885579

[10] M.E. Matta, S.E. Elmaghraby, *Polynomial time algorithms for two special classes of the proportionate multiprocessor open shop*, European Journal of Operational Research, **201**:3 (2010), 720–728. MR2552490

[11] B. Naderi, M. Zandieh, M. Yazdani, *Polynomial time approximation algorithms for proportionate open-shop scheduling*, International Transactions in Operational Research, **21**:6 (2014), 1031–1044. MR3271087

[12] S.V. Sevastyanov, *On the Compact Vector Summation*, Diskretnaya Matematika, **3**:3 (1991), 66–72. (in Russian) MR1138092

[13] S.V. Sevast'yanov, *A polynomial-time open-shop problem with an arbitrary number of machines*, Cybernetics and Systems Analysis, **28**:6 (1992), 918–933; doi: 10.1007/BF01291296 MR1215381

[14] S.V. Sevastyanov, *Vector summation in Banach space and polynomial algorithms for flow shops and open shops*, Mathematics of Operations Research, **20**:1 (1995), 90–103; doi: 10.1287/moor.20.1.90 MR1320448

[15] Sevastianov, S., *Nonstrict vector summation in multi-operation scheduling*, Annals of Operations Research, **83** (1998), 179–211; doi: 10.1023/A:1018908013582 MR1661682

[16] S.V. Sevastianov, I.D. Tchernykh, *Computer-Aided Way to Prove Theorems in Scheduling*, in: Bilardi, a.o. (Eds.), Algorithms - ESA'98, 6th Annual European Symposium, Venice, Italy, August 1998. Proceedings, Lecture Notes in Computer Science, **1461** (1998), Springer-Verlag, 502–513; doi: 10.1007/3-540-68530-8_42 Zbl 0929.90040

[17] S.V. Sevastianov, G.J. Woeginger, *Makespan minimization in open shops: a polynomial time approximation scheme*, Mathematical Programming, **82**1–2 (1998), 191–198; doi: 10.1007/BF01585871 Zbl 0920.90075

[18] Sevastianov S.V. Woeginger G.J., *Linear time approximation scheme for the multiprocessor open shop problem*, Discrete Applide Mathematics, **114** (2001), 273–288; doi: 10.1016/S0166-218X(00)00375-9 Zbl 1020.68016

Sergey Vasilyevich Sevastyanov
Sobolev Institute of Mathematics,
4, Koptyuga ave.,
Novosibirsk, 630090, Russia
E-mail address: seva@math.nsc.ru