S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports

http://semr.math.nsc.ru

Том 19, №2, стр. 528–539 (2022) УДК 519.854.2

DOI 10.33048/semi.2022.19.044 MSC 90B35

Special issue: International conference «Mathematical Optimization Theory and Operations Research» (MOTOR 2022), July 2–6, 2022, Petrozavodsk, Republic of Karelia, Russia

ON COMPLEXITY OF TWO-MACHINE ROUTING PROPOTIONATE OPEN SHOP

A.V. PYATKIN, I.D. CHERNYKH

Abstract. In the routing open shop problem a fleet of mobile machines has to traverse the given transportation network, to process immovable jobs located at its nodes and to return back to the initial location in the shortest time possible. The problem is known to be NP-hard even for the simplest case with two machines and two nodes. We consider a special *proportionate* case of this problem, in which processing time for any job does not depend on a machine. We prove that the problem in this simplified setting is still NP-hard for the same simplest case. To that end, we introduce the new problem we call 2-`Summing` and reduce it to the problem under consideration. We also suggest a $\frac{7}{6}$-approximation algorithm for the two-machine proportionate problem with at most three nodes.

Keywords: routing open shop, proportionate open shop, complexity, approximation algorithm, standard lower bound, optima localization.

Pyatkin, A.V., Chernykh, I.D., On complexity of two-machine routing propotionate open shop.

This research was was carried out within the framework of the state contract of the Sobolev Institute of Mathematics (project FWNF-2022-0019), and supported by the Russian Foundation for Basic Research, projects 20-01-00045 and 20-07-00458.

Received April, 21, 2022, published August, 25, 2022.

1. Introduction

The object of investigation of this paper is a special *proportionate* case of so-called *routing open shop problem*, which is a natural combination of the classical open shop scheduling problem and the metric traveling salesman problem (TSP).

The open shop problem was introduced in [8] and can be described as follows. The set of jobs $\mathcal{J} = \{J_1, \dots, J_n\}$ is given. Each machine M_i from the given set $\mathcal{M} = \{M_1, \dots, M_m\}$ has to perform an *operation* on every job J_j, which takes predefined processing time p_{ji}. Operations can be performed in any sequence, providing that operations of the same machine or of the same job do not overlap. The goal is to construct a feasible schedule minimizing so-called *makespan*, which is defined as the completion time of the last operation. Following the traditional three-field notation for scheduling problems (see *e.g.* [13]) the open shop problem with m machines is denoted as $Om||C_{\max}$. Notation $O||C_{\max}$ is used for the case with unbounded number of machines.

The $O2||C_{\max}$ problem can be solved to the optimum in linear time [8]. The complete review of all five known algorithms for this problem can be found in [9]. The analysis of any of these algorithm shows the important property of any instance of the $O2||C_{\max}$ problem: optimal makespan always coincides with the *standard lower bound* $\bar{C} = \max\limits_{i,j}\{\ell_i, d_j\}$, where $\ell_i = \sum\limits_j p_{ji}$ is the *load* of the machine M_i, and $d_j = \sum\limits_i p_{ji}$ is the *duration* of the job J_j. Following [11] we refer to this property as *normality*: a feasible schedule is called *normal* if its makespan equals the standard lower bound, and an instance is normal if it admits constructing a normal schedule.

The $O3||C_{\max}$ problem is NP-hard [8], however its exact complexity is still unknown. On the other hand, the problem $O||C_{\max}$ is strongly NP-hard even in the case when integer processing times do not exceed 2 [19]. For $m \geqslant 3$ normality cannot be guaranteed: the optimal makespan for $O3||C_{\max}$ can be as high as $\frac{4}{3}\bar{C}$ [18]. The NP-hardness of the $Om||C_{\max}$ problem justifies the research of its special cases. One of such relaxations is so-called *proportionate* open shop, in which the processing times are machine-independent, *i.e.* $p_{ji} = p_j$ for all i. This problem is usually denoted as $Om|prpt|C_{\max}$. However it is suggested in [17] to use clearer notation *j-prpt* to distinguish this problem from another special case with job-independent processing times, for which the name *proportionate* is also used in some papers (*e.g.* [15, 16]). In this paper we will follow the notation *j-prpt*.

The $O3|j\text{-}prpt|C_{\max}$ problem is also NP-hard. This can be easily shown by the reduction from a well-known `Partition problem` [7]; the formal proof of this fact was first published in [14]. Approximation algorithms for $Om|j\text{-}prpt|C_{\max}$ are suggested in [12, 17]. The latter paper [17] provides a pseudopolynomial algorithm for the $O3|j\text{-}prpt|C_{\max}$ problem, which gives a partial answer to the open question on exact computational complexity of $O3||C_{\max}$.

The routing open shop problem [1, 2] generalizes the open shop problem in the following manner. In addition to the open shop instance, a transportation network $G = \langle V; E\rangle$ is given. Immovable jobs are distributed among the nodes of network. One of the nodes $v_0 \in V$, called the *depot*, is the initial location of mobile machines. A weight of edge $e \in E$ represents the travel time which takes any machine to traverse this edge. In order to perform an operation on some job, machine has to reach the correspondent node first. The restrictions inherited from the open shop are in order: intervals of performing operations of the same job (same machine) cannot have common inner points. However machines can travel without any restrictions: any number of machines can traverse the same edge simultaneously in any direction and can visit any node multiple times, even without processing some job from that node. Therefore we assume, that machines take the shortest route while traveling between nodes $u, v \in V$, and the correspondent travel time is denoted as $\mathrm{dist}(u, v)$. After performing all operations, machine has to come back to the depot. The goal is to minimize the makespan $R_{\max}$, which is the completion time of the last machine's *activity* (*i.e.* traveling to the depot or processing some job located at the depot). We denote the m-machine routing open shop problem as $ROm||R_{\max}$. Notation $ROm|G = X|R_{\max}$ is used when we want to specify the structure of the network G. In this case X is substituted with some common notation from graph theory (*e.g. tree* or K_p).

The following standard lower bound for the routing open shop was introduced in [1]:

$$\bar{R} = \max\{\ell_{\max} + T^*, \max_{v \in V}(d_{\max}(v) + 2\mathrm{dist}(v_0, v))\}. \tag{1}$$

Here $\ell_{\max} = \max_i \ell_i$ is the *maximum machine load*, T^* is the optimum for the underlying TSP (*i.e.* the weight of the shortest Hamiltonian walk in G), $d_{\max}(v)$ is the maximum duration of job located at node v.

The routing open shop is strongly NP-hard for any number of machines, as it contains the metric TSP as a special case. In the contrast to $O2||C_{\max}$, the two-machine routing open shop is NP-hard even for the simplest case with two nodes ($G = K_2$) [2]. On the other hand, this problem $RO2|G = K_2|R_{\max}$ is not strongly NP-hard as it admits constructing of FPTAS [10]. The best up to date approximation algorithm for $RO2||R_{\max}$ is proposed in [4]. It constructs a feasible schedule with makespan from the interval $[\bar{R}, \frac{13}{8}\bar{R}]$ and therefore is a $\frac{13}{8}$-approximation. For a special cases with small number of nodes better approximation algorithms are given in [1] for $RO2|G = K_2|R_{\max}$ and in [5] for $RO2|G = K_3|R_{\max}$. Both algorithms construct schedules with makespan not greater than $\frac{6}{5}\bar{R}$, and this bound is tight: there exists an instance of $RO2|G = K_2|R_{\max}$ for which optimal makespan is equal to $\frac{6}{5}\bar{R}$. Therefore the interval $[\bar{R}, \frac{6}{5}\bar{R}]$ is the *tight optima localization interval* for both $RO2|G = K_2|R_{\max}$ and $RO2|G = K_3|R_{\max}$. For the relevant review on optima localization we refer the reader to [6] for routing open shop and to [17] for proportionate open shop.

In this paper we investigate the two-machine routing proportionate open shop $RO2|j\text{-}prpt|R_{\max}$. To the best of our knowledge this special case has not been studied before. In section 2 we show that the $RO2|j\text{-}prpt, G = K_2|R_{\max}$ problem is NP-hard, therefore generalizing the known result from [2]. The tight optima localization interval $[\bar{R}, \frac{7}{6}\bar{R}]$ for the case with at most three nodes is established in section 3. Some concluding remarks and directions of future research are discussed in 4.

2. The $RO2|j\text{-}prpt, G = K_2|R_{\max}$ is NP-hard

The proof of NP-hardness is organized as follows. We introduce a problem that we call 2-`Summing`, prove its NP-completeness and then show that it can be polynomially reduced to the $RO2|j\text{-}prpt, G = K_2|R_{\max}$ problem. We believe the following problem to be new: although there are some mentions of similar problems, we could not find any proof of its NP-completeness.

Problem 2-Summing. Given a set $A = \{a_1, \dots, a_N\}$ of non-negative integers and a positive integer B such that $\sum\limits_{i=1}^{N} a_i = W \geqslant 2B$, are there two disjoint (maybe empty) subsets $I_1, I_2 \subseteq \{1, \dots, N\}$ such that

$$S(I_1, I_2) \doteq \sum_{i \in I_1} a_i + 2 \sum_{i \in I_2} a_i = B?$$

The name 2-Summing goes back to the well-known NP-complete Subset Sum problem [7], where the condition $W \geqslant 2B$ is not required and no I_2 is allowed.

We need the following classic NP-complete problem [7].

Problem 3D Matching. Given three pairwise disjoint sets X, Y, Z each of size M and a family E of N triples from $X \times Y \times Z$, is there a subset $E_0 \subseteq E$ of size M such that $\bigcup\limits_{e \in E_0} e = X \times Y \times Z$?

Note that 3D Matching remains NP-complete under the additional requirement that every element $t \in X \cup Y \cup Z$ belongs to at least two triples from E. Indeed, if some t does not belong to any $e \in E$ then the answer is NO. If t lies only in one triple $e \in E$, then e must be in E_0, and the instance can be reduced.

Theorem 1. 2-Summing *problem is NP-complete.*

Proof. Note that the idea of the proof is the same as a classic NP-completeness proof of Partition problem [7]. Consider an instance of 3D Matching problem, where each element lies in at least two triples. For all $i = 1, \dots, M$ denote by x_i, y_i, and z_i the i-th element of X, Y, and Z, respectively. Put $p = \lceil \log(N+1) \rceil + 1$. Each number from A corresponds to a triple from E. Namely, if $e_s = (x_i, y_j, z_k)$ then put

$$a_s = 2^{p(i-1)} + 2^{p(M+j-1)} + 2^{p(2M+k-1)};$$

here $s = 1, \dots, N$. Also put $B = \sum\limits_{j=0}^{3M-1} 2^{pj}$. Since the total length of the constructed instance input is at most $\mathcal{O}(NMp)$, the reduction is polynomial.

If there is a subset E_0 of size M such that $\bigcup\limits_{e \in E_0} e = X \times Y \times Z$, then clearly the sum of the numbers, corresponding to the elements from E_0 is equal to B (and we take $I_2 = \emptyset$ here).

Assume that $S(I_1, I_2) = B$ for some disjoint subsets I_1 and I_2. In this case it is convenient to present the numbers in binary form. There are $3M$ groups of p bits, where each group corresponds to one element from $X \cup Y \cup Z$ (the rightmost group corresponds to x_1, the leftmost — to z_M). Each $a \in A$ assigned to a triple $e \in E$ contains in binary form exactly three 1's in the rightmost bits of groups, corresponding to the elements of E. The number B contains 1's in the rightmost bits of all groups. Since each element from $X \cup Y \cup Z$ lies in at least two triples, the condition $W \geq 2B$ holds. Note that the number N needs exactly $\lceil \log(N+1) \rceil$ binary digits; so, $2N$ needs $\lceil \log(N+1) \rceil + 1 = p$ digits. This means that even in the double sum of all numbers from A the sums of the bits in each group remains in the same group; *i. e.*, since $S(I_1, I_2) \leqslant 2W$, no shift of 1's between the groups occurs in $S(I_1, I_2)$. Therefore, the equality $S(I_1, I_2) = B$ can take place only if $|I_1| = M,\ I_2 = \emptyset$ and for each group exactly one addend in $S(I_1, I_2)$ contains 1 in

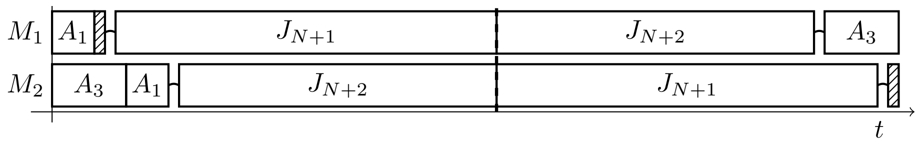

FIG. 1. A sketch of the feasible schedule in case of the positive answer to the 2-Summing problem. Hatched rectangles represent the set of jobs A_2.

the rightmost bit. But then the elements of E, corresponding to I_1 form the desired E_0. □

Remark 1. Note that for any constant k, the presented proof can be generalized for the k-Summing problem where the input is the same and the question is whether there are k pairwise disjoint subsets $I_1, \dots, I_k$ such that

$$\sum_{j=1}^{k}\left(j\sum_{i\in I_j} a_i\right) = B.$$

The only difference is that $p = \lceil \log(N+1)\rceil + \lceil \log k\rceil$ in this case.

Remark 2. Clearly, k-Summing problem is not NP-complete in the strong sense since the standard dynamic programming algorithm for the Subset Sum problem can be generalized for solving it: at each step the algorithm just has to choose the best of $k+1$ variants instead of 2.

Now we are ready to prove the main result of this section.

Theorem 2. *The $RO2|j\text{-}prpt, G = K_2|R_{\max}$ problem is NP-complete.*

Proof. Consider an instance of 2-Summing and reduce it to an instance of $RO2|j\text{-}prpt, G = K_2|R_{\max}$ in the following way. Put $n = N+2$ and let the depot v_0 contain N jobs corresponding to the numbers $a_i \in A$, processing times of both operations of the job J_j equal to a_j, $j = 1, \dots, N$. The second node v_1 contains two auxiliary jobs J_{N+1} and J_{N+2} with processing times $3W$ and $2W+B$, respectively. Choose an arbitrary $\tau > 0$ and put $\mathrm{dist}(v_0, v_1) = \tau$. Let us prove that this instance has a schedule of makespan $\overline{R} = 6W + B + 2\tau$ if and only if in the instance of 2-Summing disjoint subsets I_1, I_2 with $S(I_1, I_2) = B$ exist.

Necessity. Assume that there are disjoint subsets I_1, I_2 with $S(I_1, I_2) = B$. Denote by A_1, A_2, and A_3 the sets of numbers with indices from I_1, I_2, and $\{1, \dots, N\}\setminus(I_1 \cup I_2)$, respectively, and let $w(A_i)$ be the sum of the numbers from A_i for $i = 1, 2, 3$. Consider the following scheduling (see Fig. 1). Let the first machine process the jobs in the order $A_1, A_2, J_{N+1}, J_{N+2}, A_3$, while the second machine — in the order $A_3, A_1, J_{N+2}, J_{N+1}, A_2$ (the order of jobs inside the sets A_i can be arbitrary for each machine). Clearly, the length of the schedule is $\overline{R}$. Let us verify its correctness. It is clear that the operations of any job from $A_2 \cup A_3$ do not intersect. Since $W = w(A_1) + w(A_2) + w(A_3)$ and $B = w(A_1) + 2w(A_2)$, we have $w(A_1) + w(A_2) + 3W = w(A_3) + w(A_1) + 2W + B$, *i. e.* the completion time of job J_{N+1} at the first machine coincides with the completion time of J_{N+2} at the second machine. Finally, note that

$$w(A_3) = W - w(A_1) - w(A_2) \geq 2B - w(A_1) - w(A_2) = w(A_1) + 3w(A_2) \geq w(A_1).$$

Therefore, processing any job from A_1 at the second machine starts after finishing processing the last job from A_1 at the first machine. So, the schedule is feasible.

Sufficiency. Assume that the instance of Problem 1 has a schedule with makespan $\overline{R} = 6W + B + 2\tau$. Then each machine has only one travel to the second node and back, and process jobs without idle. We may assume that the job J_{N+1} is processed first at the machine M_1. Since $\overline{R} < 8W + B + 2\tau$, the job J_{N+2} must be processed first at the machine M_2; moreover, the absence of idle intervals implies that the machine M_1 finishes processing the job J_{N+1} at the same time when M_2 finishes processing J_{N+2}. Denote by X the set of all jobs in the depot that the machine M_1 finished before processing the job J_1, and by Y the set of all jobs in the depot that the machine M_2 started after processing the job J_2. Note that X and Y may intersect and one of them may be empty. By the above mentioned observation, $w(X)+3W = (W-w(Y))+2W+B$, *i. e.* $w(X)+w(Y) = B$. Put $A_2 = X \cap Y$ and $A_1 = (X \cup Y) \setminus A_2$. Then we have $B = w(X) + w(Y) = w(A_1) + 2w(A_2)$. Hence, the sets of indices I_1, I_2 corresponding to the sets A_1, A_2 satisfy the condition $S(I_1, I_2) = B$, as required. □

3. Optima localization for small number of nodes

In [5] a reduction procedure for the $RO2||R_{\max}$ problem is described, which transforms any instance I into simplified instance I' with the same value of the standard lower bound (1). This transformation has the following important properties:

- it is *reversible*, *i.e.* any feasible schedule for I' can be treated as a feasible schedule for the initial instance I with the same makespan,
- I' contains a single job at each node of the transportation network except maybe one node that contains two or three jobs.

This procedure is based on a *job aggregation* operation (also known as *job grouping*), which basically is replacing a subset of jobs from the same node by a new one, with processing times equal to the total processing time of the replaced jobs.

The following approximation algorithm $\mathcal{A}$ for $RO2|G = K_3|R_{\max}$, based on this procedure, is suggested:

(1) Transform given instance I into a simplified one I'. (Note that it contains at most 5 jobs.)
(2) Build an optimal schedule S' for I'. (Note that it can be done in constant time.)
(3) Treat the schedule S' as a feasible schedule S for instance I. Output S.

It is shown in [5] by means of case analysis, that $R_{\max}(S) \leqslant \frac{6}{5}\bar{R}$, and the algorithm described is therefore a $\frac{6}{5}$-approximation.

In this section we use the same idea the $RO2|j\text{-}prpt, G = K_3|R_{\max}$ problem. The following lemma establishes the lower bound on the performance guarantee of algorithm $\mathcal{A}$ for the proportionate routing open shop.

Lemma 1. *There exists an instance $\tilde{I}$ of $RO2|j\text{-}prpt, G = K_2|R_{\max}$, such that the optimal makespan for $\tilde{I}$ equals $\frac{7}{6}\bar{R}$.*

Proof. Consider the following instance $\tilde{I}$: the depot v_0 contains a single job J_1 with processing times 2, the other node v_1 contains two identical jobs J_2, J_3 with processing times 4. The travel time between the nodes is $\mathrm{dist}(v_0, v_1) = 1$. Note that

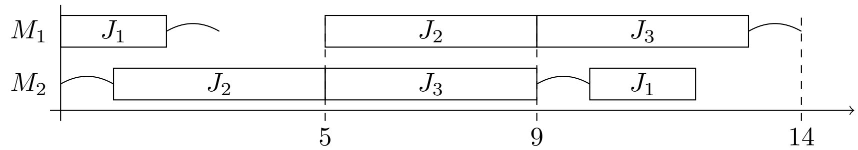

FIG. 2. An optimal schedule S for instance $\tilde{I}$.

machine loads are $\ell_1 = \ell_2 = 10$, maximum job duration in node v_1 is $d_{\max}(v_1) = 8$, and $T^* = 2$, therefore $\bar{R}(\tilde{I}) = 12$. A feasible schedule S for $\tilde{I}$ with makespan 14 is given in Fig. 2. Concave arcs represent travel times.

Let us prove that schedule S is optimal for $\tilde{I}$. Suppose otherwise and consider a schedule S' such that $R_{\max}(S') < 14$. Note that each machine in S' travels form v_0 to v_1 and back exactly once, otherwise the travel time would be at least 4 and the makespan at least 14. Without loss of generality assume that machine M_1 processes jobs in order J_1, J_2, J_3. Idle time of any machine in S' is less than 2 (otherwise $R_{\max}(S')$ is at least 14), therefore M_2 cannot start with job J_1, and hence job J_1 is performed last by machine M_2. If machine M_2 processes job J_2 after machine M_1, then is completes J_2 not earlier than at 11. In this case $R_{\max}(S') \geqslant 14$. On the other hand, if machine M_2 processed J_2 before M_1, machine M_1 idles for at least two time units before the processing of operation of job J_2. Therefore, by contradiction, such schedule S' does not exist. □

Now consider an instance I of $RO2|G = K_3|R_{\max}$ with nodes $\{v_0, v_1, v_2\}$ and corresponding simplified instance I', obtained from I by the reduction procedure from [5]. Assume that I' is *irreducible*, *i.e.* no further job aggregation in I' is possible without violating the standard lower bound $\bar{R}$. If a node in I' contains more that one job, it is called *overloaded*. A node with three jobs in irreducible instance is referred to as *superoverloaded* (see [6] for details).

Our goal is to prove that there exists a feasible schedule for I' (and therefore for I) with makespan of at most $\frac{7}{6}\bar{R}$. It is proved in [3] that any instance of $RO2||R_{\max}$ with an overloaded depot admits constructing a normal schedule (*i.e.* feasible schedule of makespan $\bar{R}$). It follows from [6] that any instance of $RO2|G = K_3|R_{\max}$ with a superoverloaded node has the same property. Therefore we only need to consider a special case of irreducible instance I' with at most two jobs at node v_1 and single jobs at nodes v_0 and v_2.

Lemma 2. *Let I be an irreducible instance of $RO2|j\text{-}prpt, G = K_3|R_{\max}$ with single jobs at v_0 and v_2 and at most two jobs at v_1. Then there exists a feasible schedule S for I such that $R_{\max}(S) \leqslant \frac{7}{6}\bar{R}(I)$.*

Proof. We consider two cases with different number of jobs at node v_1 separately.

Case 1. Node v_1 has two jobs.

Denote the processing times of four jobs J_1, J_2, J_3, J_4 by a, b, c, d respectively, J_1 at v_0, J_4 at v_2. Denote travel times as follows:

$$\text{dist}(v_0, v_1) = \tau, \text{dist}(v_0, v_2) = \mu, \text{dist}(v_1, v_2) = \nu, T^* = \tau + \mu + \nu.$$

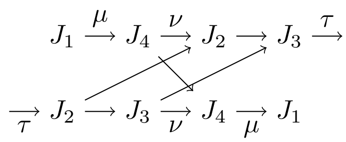

FIG. 3. Processing order for schedule S_1.

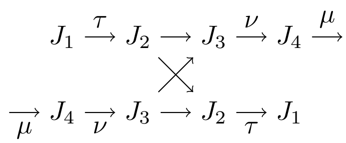

FIG. 4. Processing order for schedule S_2.

Note that, due to the irreducibility of instance I, aggregation of jobs J_2 and J_3 would violate the lower bound R, therefore

$$2b + 2c + 2\mathrm{dist}(v_0, v_1) > \bar{R}. \tag{2}$$

Without loss of generality assume

$$b \geqslant c. \tag{3}$$

Note that in this case

$$\bar{R}(I) = \max\{a + b + c + d + \tau + \mu + \nu, 2a, 2b + 2\tau, 2d + 2\mu\}. \tag{4}$$

We will describe a series of schedules for I and prove that at least one of the schedules constructed has a makespan of at most $\frac{7}{6}\bar{R}$. Each schedule will be described by specifying the linear order of operations on each machine and each job.

Let S_1 be a schedule with sequence of jobs $J_1 \to J_4 \to J_2 \to J_3$ on M_1 and $J_2 \to J_3 \to J_4 \to J_1$ on M_2, jobs J_1 and J_4 are processed first by M_1, and J_2, J_3 processed first by M_2 (Fig. 3).

Note that due to (2) we have

$$2a + 2d + 2\mu \leqslant 2\bar{R} - (2b + 2c + 2\tau) < \bar{R},$$

therefore by (3) and (4) it is sufficient to consider $R_{\max}(S_1) = R_1 = 2b + c + 2\tau$ (otherwise the makespan the schedule S_1 is normal).

Construct schedule S_2 according to the partial order of operations from Fig. 4.

Case 1.1. $R_{\max}(S_2) = R_2 = 2a + 2b + 2\tau$.

Construct schedule S_3 according to the partial order of operations from Fig. 5. It is sufficient to consider the case $R_{\max}(S_3) = R_3 = 2a + 2c + 2d + \tau + \mu + \nu$.

Construct schedule S_4 according to the partial order of operations from Fig. 6. Consider three subcases.

Case 1.1.1. $R_{\max}(S_4) = R_4 = a + 2d + 2c + 2\mu + 2\nu$.

In this case

$$2R_1 + 3R_2 + 4R_4 = 10a + 10b + 10c + 8d + 10\tau + 8\mu + 8\nu \leqslant 10\bar{R},$$

therefore $\min\{R_1, R_2, R_4\} \leqslant \frac{10}{9}\bar{R} < \frac{7}{6}\bar{R}$.

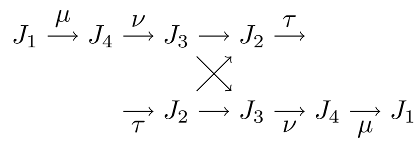

FIG. 5. Processing order for schedule S_3.

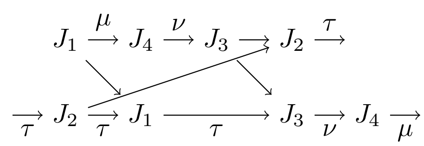

FIG. 6. Processing order for schedule S_4.

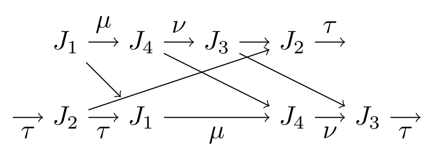

FIG. 7. Processing order for schedule S_5.

Case 1.1.2. $R_{\max}(S_4) = R_4 = 2a + c + d + \tau + \mu + \nu$.
In this case

$$R_1 + R_4 = 2a + 2b + 2c + d + 3\tau + \mu + \nu \leqslant 2\bar{R},$$

since by triangle inequality, $\tau \le \mu + \nu$. So, at least one of schedules S_1 and S_4 is normal.

Case 1.1.3. $R_{\max}(S_4) = R_4 = a + b + c + d + 3\tau + \mu + \nu$.
In this case, due to (3) and $\tau \le \mu + \nu$

$$2R_1 + 3R_3 + R_4 = 5b + 9c + 7a + 7d + 10\tau + 4\mu + 4\nu \leqslant 7\bar{R},$$

and $\min\{R_1, R_3, R_4\} \leqslant \frac{7}{6}\bar{R}$.

Case 1.2. $R_{\max}(S_2) = R_2 = 2c + 2d + 2\mu + 2\nu$.
Construct schedule S_5 with the partial order of operations as shown in Fig. 7.
Due to (3) it is sufficient to consider two cases.

Case 1.2.1. $R_{\max}(S_5) \leqslant R_5 = a + d + \max\{a, d\} + c + \tau + \mu + \nu$.
In this case

$$R_1 + R_5 = 2b + 2c + a + d + \max\{a, d\} + 3\tau + \mu + \nu \leqslant 2\bar{R}$$

and the makespan of at least one of schedules S_1 and S_5 equals $\bar{R}$.

Case 1.2.2. $R_{\max}(S_5) = R_5 = a + b + c + d + 3\tau + \mu + \nu$.
In this case, due to (3)

$$2R_1 + 3R_2 + R_5 = 5b + 9c + 7d + a + 7\tau + 7\mu + 7\nu \leqslant 7\bar{R},$$

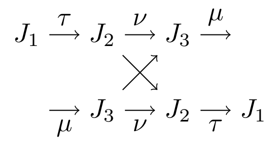

FIG. 8. Processing order for schedule S_6.

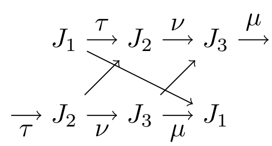

FIG. 9. Processing order for schedule S_7.

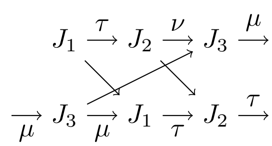

FIG. 10. Processing order for schedule S_8.

and $\min\{R_1, R_2, R_5\} \leqslant \frac{7}{6}\bar{R}$.

Case 2. Each node contains exactly one job. In this case denote processing times of three jobs J_1, J_2, J_3 by a, b, c respectively, J_1 at v_0, J_3 at v_2. Travel time are denoted as in Case 1. In this case we have

$$\bar{R}(I) = \max\{a + b + c + \tau + \mu + \nu, 2a, 2b + 2\tau, 2c + 2\mu\}. \quad (5)$$

Without loss of generality we assume

$$b + \tau \leqslant c + \mu. \quad (6)$$

Construct three schedules S_6, S_7 and S_8 according to partial orders at Figures 8, 9, 10, respectively.

Using (5) and (6), and assuming each of the schedules built has makespan greater that $\bar{R}$ we have

$$R_{\max}(S_6) = R_6 = 2a + 2b + 2\tau,$$
$$R_{\max}(S_7) = R_7 = b + c + \max\{b, c\} + \tau + \mu + \nu,$$
$$R_{\max}(S_8) = R_8 = \max\{2a + b + 2\tau, a + b + c + 2\mu + 2\tau\}.$$

Consider the following two subcases.

Case 2.1. $R_8 = 2a + b + 2\tau$.

In this case using (6), $\mu + \tau \leqslant T^*$, and $2\tau \leqslant T^*$ we obtain

$$R_7 + R_8 = 2a + 2b + c + \mu + \nu + 2\tau + \max\{b + \tau, c + \tau\} \leqslant$$
$$2a + 2b + 2c + \mu + \nu + 2\tau + \max\{\mu, \tau\} \leqslant 2a + 2b + 2c + 2T^* \leqslant 2\bar{R},$$

therefore at least one of schedules S_7 and S_8 is normal.

Case 2.2. $R_8 = a + b + c + 2\mu + 2\tau$.

In this case using (6) and $\tau \leqslant \mu + \nu$ we have

$$\Sigma = 3R_6 + 2R_7 + R_8 = 7a + 9b + 3c + 2\max\{b, c\} + 10\tau + 4\mu + 2\nu$$
$$\leqslant 7a+7b+5c+2\max\{b, c\}+8\tau+6\mu+2\nu \leqslant 7a+7b+5c+2\max\{b, c\}+7\tau+7\mu+3\nu.$$

If $b \leqslant c$ then Σ clearly does not exceed $7\bar{R}$. Assume $b \geqslant c$. Then

$$\Sigma = 7a + 9b + 5c + 7\tau + 7\mu + 3\nu \leqslant 7a + 7b + 7c + 5\tau + 9\mu + 3\nu$$
$$\leqslant 7a + 7b + 7c + 7\tau + 7\mu + 5\nu \leqslant 7\bar{R}.$$

Anyway, $3R_6 + 2R_7 + R_8 \leqslant 7\bar{R}$; so,

$$\min\{R_6, R_7, R_8\} \leqslant \frac{7}{6}\bar{R}.$$

We have considered all possible cases, and in each case the makespan of one of the schedules built does not exceed $\frac{7}{6}\bar{R}$. Lemma is proved. □

Theorem 3. *The tight optima localization interval for both $RO2|G = K_2, j-prpt|R_{\max}$ and $RO2|G = K_3, j\text{-}prpt|R_{\max}$ problems is $[\bar{R}, \frac{7}{6}\bar{R}]$.*

Proof. Straightforward from Lemmas 1 and 2. □

Remark 3. Theorem 3 together with algorithm $\mathcal{A}$ imply the existance of $\frac{7}{6}$-approximation for the $RO2|G = K_3, j\text{-}prpt|R_{\max}$ problem.

4. Conclusion

The main result of this paper is the NP-hardness of a very restricted special case of $RO2||R_{\max}$. We presume, that the auxiliary problem 2-`Summing` or its variations might also be helpful in research of the computational complexity of various scheduling problems.

To our opinion, it makes sense to continue the investigation of the proportionate routing open shop in the following directions:

(1) Approximation algorithms and optima localization for two-machine proportionate routing open shop with more complex structure of the transportation network, as well as with asymmetric distances.
(2) Approximation algorithms for the problem with $m \geqslant 3$.
(3) Approximation algorithms for the generalization of proportionate routing open shop with *prorpotional* processing times, which would finally justify the name *proportionate*.

References

[1] I. Averbakh, O. Berman, I. Chernykh, *A 6/5-approximation algorithm for the two-machine routing open-shop problem on a two-node network,* Eur. J. Oper. Res., **166**:1 (2005), 3–24. Zbl 1069.90037

[2] I. Averbakh, O. Berman, I. Chernykh, *The routing open-shop problem on a network: complexity and approximation,* Eur. J. Oper. Res., **173**:2 (2006), 531–539. Zbl 1109.90038

[3] I. Chernykh, *Two-machine routing open shop: How long is the optimal makespan?*, in Pardalos, Panos (ed.) et al., *Mathematical optimization theory and operations research. 20th international conference, MOTOR 2021, Proceedings*, Springer Lect. Notes Comput. Sci., **12755**, Springer, Cham, 2021, 253–266; Zbl 7495084

[4] I. Chernykh, A. Kononov, S. Sevastyanov, *Efficient approximation algorithms for the routing open shop problem,* Comput. Oper. Res., **40**:3 (2013), 841–847. Zbl 1349.90330

[5] I. Chernykh, E. Lgotina, *The 2-machine routing open shop on a triangular transportation network,* in Kochetov, Yury (ed.) et al., *Discrete optimization and operations research. 9th international conference, DOOR 2016, Vladivostok, Russia, September 19–23, 2016. Proceedings*, Lecture Notes in Computer Science, **9869**, 2016, 284–297. Zbl 1385.90008

[6] I. Chernykh, A. Pyatkin, *Irreducible bin packing and normality in routing open shop,* Ann. Math. Artif. Intell., **89**:8-9 (2021), 899–918. Zbl 1476.90111

[7] M. Garey, D. Johnson, *Computers and intractability: A guide to the theory of NP-completeness,* W.H. Freeman & Co., San Francisco, 1979. Zbl 0411.68039

[8] T. Gonzalez, S. Sahni, *Open shop scheduling to minimize finish time,* J. Assoc. Comput. Mach., **23** (1976), 665–679. Zbl 0343.68031

[9] A. Khramova, I. Chernykh, *A new algorithm for the two-machine open shop and the polynomial solvability of a scheduling problem with routing,* J. Sched., **24**:4 (2021), 405–412. Zbl 1479.90098

[10] A. Kononov, *On the routing open shop problem with two machines on a two-vertex network,* J. Appl. Ind. Math., **6**:3 (2012), 318–331. Zbl 1324.90059

[11] A Kononov, S. Sevastianov, I. Tchernykh, *When difference in machine loads leads to efficient scheduling in open shops,* Ann. Oper. Res., **92** (1999), 211–239. Zbl 0958.90040

[12] C. Koulamas, G. Kyparisis, *The three-machine proportionate open shop and mixed shop minimum makespan problems,* Eur. J. Oper. Res., **243**:1 (2015), 70–74. Zbl 1346.90359

[13] E. Lawler, J. Lenstra, A. Rinnooy Kan, D. Shmoys, *Sequencing and scheduling: Algorithms and complexity,* in *Logistics Of Production And Inventory*, Handbooks in Operations Research and Management Science, **4**, Elsevier, 1993, 445–522.

[14] C. Liu, R. Bulfin, *Scheduling ordered open shops,* Comput. Oper. Res., **14** (1987), 257–264. Zbl 0625.90045

[15] M. Matta, S. Elmaghraby, *Polynomial time algorithms for two special classes of the proportionate multiprocessor open shop,* Eur. J. Oper. Res., **201**:3 (2010), 720–728. Zbl 1176.90231

[16] B. Naderi, M. Zandieh, M. Yazdani, *Polynomial time approximation algorithms for proportionate open-shop scheduling,* Int. Trans. Oper. Res., **21**:6 (2014), 1031–1044. Zbl 1309.90030

[17] S. Sevastyanov, *Some positive news on the proportionate open shop problem,* Sib. Èlektron. Mat. Izv., **16** (2019), 406–426. Zbl 1414.90169

[18] S. Sevastyanov, I. Tchernykh, *Computer-aided way to prove theorems in scheduling,* in Bilardi, Gianfranco (ed.) et al., *Algorithms — ESA'98Algorithms – ESA '98. 6th annual European symposium, Venice, Italy, August 24–26, 1998. Proceedings*, Lect. Notes Comput. Sci., **1461**, Springer, Berlin, 1998, 502–513. Zbl 0929.90040

[19] D. Williamson, L. Hall, J. Hoogeveen, C. Hurkens, J. Lenstra, S. Sevast'janov, D. Shmoys, *Short shop schedules,* Oper. Res., **45**:2 (1997), 288–294. Zbl 0890.90112

Artem Valerievich Pyatkin
Sobolev Institute of Mathematics,
4, Koptyuga ave.,
Novosibirsk, 630090, Russia
Email address: artem@math.nsc.ru

Ilya Dmitrievich Chernykh
Sobolev Institute of Mathematics,
4, Koptyuga ave.,
Novosibirsk, 630090, Russia
Email address: idchern@math.nsc.ru