*УДК 681.32*

# ОПТИМАЛЬНЫЙ ВЫБОР И ПЛАНИРОВАНИЕ РАБОТ С НЕОПРЕДЕЛЕННЫМИ ДЛИТЕЛЬНОСТЯМИ ДЛЯ ДВУХ СОТРУДНИКОВ

***Н. М. МАТВЕЙЧУК***[1), ***Ю. Н. СОТСКОВ***[2)

[1)*Белорусский государственный аграрный технический университет, пр. Независимости, 99, 220012, г. Минск, Беларусь*
[2)*Объединенный институт проблем информатики НАН Беларуси, ул. Сурганова, 6, 220012, г. Минск, Беларусь*

***Аннотация.*** Количество потенциальных пользователей тайм-менеджмента в мире неуклонно возрастает в связи с необходимостью удаленной работы (в домашних условиях), учебы, преподавания, обслуживания и в целом организации профессиональной деятельности и частной жизни с минимумом личных контактов из-за распространения в 2020 г. коронавирусной инфекции COVID-19 и других опасных инфекций. Требуются совершенствование методик тайм-менеджмента и разработка новых алгоритмов и программных средств, которые позволят учитывать особенности и потребности новых пользователей тайм-менеджмента. Такие задачи возникают в тайм-менеджменте при оптимальном выборе важных работ для двух исполнителей на определенный период времени и при составлении расписаний выполнения выбранных работ в условиях неопределенности длительностей планируемых операций. Представлены достаточные условия, алгоритмы, результаты компьютерных экспериментов по оптимальному выбору и планированию взаимосвязанных работ для двух исполнителей (руководителя и подчиненного).

***Ключевые слова:*** тайм-менеджмент; оптимальное расписание; неопределенные длительности.

***Благодарность.*** Работа выполнена при поддержке Белорусского республиканского фонда фундаментальных исследований (грант № Ф23РНФ-017).

**Образец цитирования:**
Матвейчук НМ, Сотсков ЮН. Оптимальный выбор и планирование работ с неопределенными длительностями для двух сотрудников. *Журнал Белорусского государственного университета. Математика. Информатика.* 2024;2:65–80 (на англ.).
EDN: LCOGKF

**For citation:**
Matsveichuk NM, Sotskov YuN. Optimal selection and scheduling of jobs with uncertain durations for two employees. *Journal of the Belarusian State University. Mathematics and Informatics.* 2024;2:65–80.
EDN: LCOGKF

**Авторы:**
***Наталья Михайловна Матвейчук*** – кандидат физико-математических наук, доцент; заведующий кафедрой автоматизированных систем управления производством агроэнергетического факультета.
***Юрий Назарович Сотсков*** – доктор физико-математических наук, профессор; главный научный сотрудник лаборатории математической кибернетики.

**Authors:**
***Natalja M. Matsveichuk***, PhD (physics and mathematics), docent; head of the department of automated systems of production control, agri-power faculty.
*matsveichuk@tut.by*
*https://orcid.org/0000-0002-4991-4271*
***Yuri N. Sotskov***, doctor of science (physics and mathematics), full professor; chief researcher at the laboratory of the mathematical cybernetics.
*sotskov48@mail.ru*
*https://orcid.org/0000-0002-9971-6169*

# OPTIMAL SELECTION AND SCHEDULING OF JOBS WITH UNCERTAIN DURATIONS FOR TWO EMPLOYEES

*N. M. MATSVEICHUK*[a], *Yu. N. SOTSKOV*[b]

[a]*Belarusian State Agrarian Technical University, 99 Niezaliezhnasci Avenue, Minsk 220012, Belarus*
[b]*United Institute of Informatics Problems, National Academy of Sciences of Belarus, 6 Surganava Street, Minsk 220012, Belarus*

*Corresponding author: Yu. N. Sotskov (sotskov48@mail.ru)*

***Abstract.*** The number of potential users of time-management in the world is steadily growing due to the emerging need for remote work (in a home office), distance learning, teaching, service and, in general, the organisation of professional activities and a private life with a minimum of personal contacts due to the spread of the coronavirus infection COVID-19 since 2020 and other dangerous infections. This will require the improvement of the time-management techniques and the developments of new algorithms and software for them, which will take into account the peculiarities and needs of new users of time-management. Such problems arise in time-management for optimally selecting jobs for a given time interval and for constructing optimal schedules for processing jobs under conditions of uncertain operation durations. This article presents sufficiency conditions, algorithms, and computational results for selecting and scheduling connected jobs by two employees.

***Keywords:*** time-management; optimal schedule; uncertain processing times.

***Acknowledgements.*** The work was supported by the Belarusian Republican Foundation for Fundamental Research (grant No. F23RNF-017).

## Introduction

Time-management is used for optimally choosing and planning jobs with respect to personal goals and professional activity. It includes choosing personal goals and objectives, long-term and short-term planning and the operational management of person's affairs.

**Related works and research motivations.** Article [1] examines the role of schedules in a social life. It brings into focus the main principles underlying a schedule, namely a temporal regularity involving the standardisation of the temporal locations of events and activities and their rates of recurrence and sequential. The discussion includes the constraints and the conveniences involved in using a personal schedule. As S. Eilon [2] notes, the use of time-management allows an employee to save up to 50 % of her (his) time on completing planned works, spending no more than 10 % of her (his) time on analysing and planning works during a day. Effective planning and scheduling can reduce the wasted time [3–5]. Article [3] suggests a structured approach based on the strategic and tactical time-management. The strategy is to write down a list of activities, establish priorities, and eliminate inessentials. The tactic is how to carry out essential activities with time-efficiency. Time-management is underpinned by the principle: there is no point in efficiently doing something that should not be done at all. There is a Pareto principle as follows: find out what is required and the value of alternatives. It is founded that 80 % of the value can be achieved from 20 % of the effort (80/20 rule) [3].

As it is written in [4], managers can improve their managerial performance significantly through time-management, which is a process that has to be proposed and understood by a manager since the inception of the managerial career. Prioritising tasks, preparing a to-do list, building a schedule and daily planning apart from being a good listener lead to managers who practice effective time-management and are generally successful in their profession and other domains. The 80/20 rule (a Pareto principle) is one of the most helpful of all concepts of time-management. Understanding time-management habits and practicing effective time-management techniques help in improving one's personal and managerial effectiveness.

There are articles [6–8] that examine results of time-management, and in particular, the impact of time-management on a student's academic success. The hypothesis of study [6] was that efficient time-management, under the guidance of an educational counselor, leads to significant increases in students' academic performances and so leads to academic success. Participants using time-management had above average or superior intellectual abilities. The educational counselor elaborated individualised and flexible programmes for each participant in the experimental condition according to students' learning styles, circadian and eating rhythms and daily and weekly effort curves. The results of the conducted experiments confirmed the hypothesis showing the efficiency of time-management individualised programmes [6].

Descriptive study in [7] was conducted to determine nursing and midwifery students' time-management skills in terms of their age, gender, and anxiety levels. It was demonstrated that nursing and midwifery students' time-management skills are at a mid-level point; female students were able to manage time better than male ones. The time-management skills of the students decreased as the anxiety level increased. The conclusion in [7] was that students are required to learn to manage time so that they are able to apply the same degree of efficiency in the profession they choose after completing their education.

Article [8] found that the indicator of a person's creative abilities is significantly correlated with the use of a time-management technology. Thus, time-management is a technique that almost any individual can use to increase the effectiveness and efficiency of the working time including creative and scientific activities.

**Selecting and scheduling most important jobs.** People often strive to solve several problems at the same time, combining complex tasks during working hours. Situations arise when a person puts off important and urgent work, which may be unpleasant or unusual for her (him), and strives to perfectly complete unimportant and even useless work. Such habits lead to a decrease in the likelihood of completing important tasks on time [9–11]. One of the valuable strategies often used for an effective implementation of time-management is the selection of mostly important jobs and their optimal planning [1; 4–6; 9; 12].

The importance of helping employees to plan their work from the very beginning is widely stated in the literature [4–7; 9]. A supervisor should assist the supervisee to devise a proposed schedule for activities to be undertaken and ensure that the schedule is followed. Such a plan will allow the structured and disciplined use of time of an employer [5]. It is suggested that dividing the work into smaller and more manageable units, which can be planned and controlled, makes a huge task more attainable. Identifying the expected dates for the completion of each phase is important. When devising a schedule, it is helpful to start with the expected date of the completion and work the phases from the deadline backwards [5]. Safety time can be built into the plan to allow for catch-up periods. Though it may seem tedious to plan time in such a detailed way, the results will be worthwhile.

As written in [9], one of the key components of an organisation is maintaining an individual calendar. Many employees let their schedules dictate them. The first step is to make a to-do list and prioritise each item. One needs to be realistic about what one can achieve over the next day, week, or month. If the employee is (on one's own) going over schedule, she (he) will be disappointed when she (he) fails to complete every job. The to-do list should be reviewed regularly, daily if possible, and revised as necessary. One needs to set aside time to plan, either first thing each morning or last thing in the evening to plan for the next day. It is important to do this daily or weekly as priorities may change over time [9].

Time-management makes it possible to more effectively select, plan and complete a significant number of jobs of varying complexity, which has a positive effect on the timing of necessary job, educational achievements and an increase in the quality of life [13; 14]. Optimal planning is a complex process, which requires time resources and human intellectual abilities. Organising the selected jobs can be a difficult task for the performer, requiring both additional time and certain skills. In addition, the user has to carry out the prioritisation of the planned jobs, as well as the ordering of the still unfulfilled and newly received jobs, many times over the entire planning horizon. It is advisable to use a personal computer (laptop or smartphone) as much as possible to automate the process of scheduling the planned jobs.

A problem of minimising the total (average) weighted completion time of the planned jobs by one employee is considered in [15] provided that only lower and upper bounds of the possible processing time of each job are known before scheduling. Algorithms and software have been developed for constructing a permutation of the chosen jobs with the largest relative semi-perimeter of the optimality parallelepiped. Computational experiments on the computer showed the effectiveness of the developed algorithms for time-management.

We consider the problems of creating optimal schedules for two employees. It will be shown how scheduling algorithms can be used to optimal time-management.

## Optimal selection and scheduling jobs for two employees

The discussed publications [1; 3; 4; 6; 9; 12; 14; 15] include different techniques and procedures for time-management, which are recommended to be used for planning the working time of a single employee. In our paper, we develop scheduling algorithms for two employees having a set of common jobs, e. g., for a supervisor and a subordinate. The aim of time-management is to create a job schedule for both employees during their working hours.

Consider the main features of such scheduling. The entire set of jobs consists of jobs of four types. Jobs that are performed firstly by a supervisor and then by a subordinate (e. g., a supervisor formalises a problem, outlines possible ways for solving it and delegates it to a subordinate). Jobs that are performed first by the subordinate and then by the supervisor (e. g., a supervisor checks the result of the job performed by a subordinate). There are jobs that are completely performed by a supervisor and jobs that are completely performed by a subordinate. It is naturally to assume that performing such a job consists of the execution of two or one operations. No repetition of the same concrete job is considered.

The following key peculiarity is an uncertainty of the operation durations. Indeed, it is difficult to determine an exact time, which will be required for processing a job by a human. On the other hand, one can determine a lower bound and upper bound of the operation duration. In general, the duration of each operation may remain unknown until the moment of completion of this job. At the moment of constructing an optimal schedule, a closed interval is known, which definitely contains all the possible operation durations of the planned job.

As it is written in [11; 12] and in many other papers on time-management, interruptions should be avoided while completing a job in progress (avoid unscheduled meetings, phone calls, and visitors). In addition to the direct loss of time, such interruptions cause the need to spend additional time for re-preparing the interrupted job.

The selection of jobs to perform from the entire list of available tasks can be made in accordance with their importance for the employee. The different levels of importance of the jobs can be represented in the form of the weights of the jobs to be fulfilled in the planning horizon. The criterion for the effectiveness of time-management is not only to achieve the goals set by a person, but also to complete her (his) work in the minimum possible time [16].

The constructed schedule must have a minimum length (it is the minimisation of makespan). Other criteria are to maximise the total weight of the completed jobs and to maximise the number of jobs completed in time. We use the terminology of the scheduling theory from [17] and the $\alpha|\beta|\gamma$ classification from [18] for denoting the scheduling problems, where $\alpha$ specifies machine environments, $\beta$ – job characteristics, and $\gamma$ – objective functions.

**Setting of the scheduling problem.** Let the set of jobs $\Im = (J_1, J_2, \ldots, J_n)$ have to be processed by two performers $\mathrm{M} = \{M_1, M_2\}$. A weight (an importance) $w_i$ of the job $J_i \in \Im$ is determined. The supervisor is the first performer $M_1$. The subordinate is the second performer $M_2$. Jobs in the set $\Im$ may have different (technological) routes. This processing system is called a job-shop. The number of stages (operations) $n_i$ in the route of a job $J_i \in \Im$ does not exceed two (since there are two employees). The duration of operation $O_{ij}$ is denoted by $p_{ij}$, where $J_i \in \Im,\ j \in \{1, 2\}$. The lower and upper bounds of possible duration $p_{ij}$ are denoted as $a_{ij}$ and $b_{ij}$, respectively. Thus, the uncertain (interval) job-shop problem is considered, where possible duration $p_{ij}$ of the operation $O_{ij}$ must belong to the closed interval $[a_{ij}, b_{ij}]$.

*Remark.* It is assumed that in the uncertain (interval) scheduling problem under consideration, all durations of the jobs are unknown before scheduling, i. e. the strict inequality $a_{ij} < b_{ij}$ holds for each job $J_i \in \Im$ and each machine $M_j \in \mathrm{M}$.

Let $C_i$ denote a moment of the completion of the job $J_i \in \Im$. We consider the following three ordered criteria: minimising a schedule length, i. e. makespan $C_{\max} = \max\{C_i : J_i \in \Im\}$, maximising a sum of the weights of the completed jobs $\sum w_i$ and maximising a total number of jobs $\sum U_i$ that are completed before their due dates $D_i$, where $U_i$ is equal to 1, if $C_i \le D_i$, and $U_i$ is equal to 0, if $C_i > D_i$. Using the three-field notation $\alpha|\beta|\gamma$, the problem with uncertain operation durations is denoted as follows: $J2|a_{ij} \le p_{ij} \le b_{ij},\ n_i \le 2|C_{\max}, \sum w_i, \sum U_i$, where three criteria $C_{\max}$, $\sum w_i$ and $\sum U_i$ are linearly ordered, i. e. the main criterion is $C_{\max}$, the second criterion is $\sum w_i$ and the third criterion is $\sum U_i$.

This paper continues the previous research works started in [15; 19–21] via extending the obtained results to the job-shop problem with three ordered criteria. In [15], the time-management problem for a single employee was investigated with the single criterion of minimising the weighted sum of the job completion times. The properties of optimal permutations existing for a flow-shop scheduling problem with the single criterion of the minimisation of a schedule length were investigated in [20; 21]. A similar properties of optimal permutations existing for a job-shop scheduling problem were investigated in [19]. It should be noted that papers [19–21] were devoted to the uncertain shop scheduling problems where non-strict inequalities $a_{ij} \le b_{ij}$ hold for all given jobs $J_i \in \Im,\ j \in \{1, 2\}$.

## Uncertain (interval) scheduling problems

Employees in time-management correspond to machines in the scheduling theory [15; 17–19; 22]. For scheduling jobs for a working day, we consider the uncertain two-machine job-shop problem $J2|a_{ij} \le p_{ij} \le b_{ij},\ n_i \le 2|C_{\max}$. The machine set $\mathrm{M} = \{M_1, M_2\}$ has to process the job set $\Im = \Im_1 \cup \Im_2 \cup \Im_{1,2} \cup \Im_{2,1}$, where the subset $\Im_{1,2}$ includes jobs with the machine route $(M_1, M_2)$, $|\Im_{1,2}| = n_{1,2}$. The subset $\Im_{2,1}$ includes jobs with the opposite machine route $(M_2, M_1)$, $|\Im_{2,1}| = n_{2,1}$. The subset $\Im_1$ (the subset $\Im_2$) includes jobs that must be processed by the machine $M_1$ (by the machine $M_2$, respectively). Here $|\Im_1| = n_1$, $|\Im_2| = n_2$ and $n = n_{1,2} + n_{2,1} + n_1 + n_2$. All jobs are available for processing from the initial time $t = 0$. A preemption of any operation $O_{ij}$ of the job $J_i \in \Im$ on the machine $M_j \in \mathrm{M}$ is not allowed. Probability distributions of random durations are unknown. In the realisation of a schedule, a value of the processing time $p_{ij}$ may be equal to any real number no less than the lower bound $a_{ij}$ and no larger than the upper bound $b_{ij}$.

A set of all possible vectors $p = \left(p_{1,1}, p_{1,2}, \ldots, p_{n1}, p_{n2}\right)$ of the operation durations is denoted as follows: $T = \left\{p : a_{ij} \le p_{ij} \le b_{ij},\ J_i \in \Im,\ M_j \in \mathrm{M}\right\}$. Such a vector $p \in T$ of the possible durations is called a scenario. For a fixed scenario $p \in T$, the uncertain scheduling problem $J2\left|a_{ij} \le p_{ij} \le b_{ij},\ n_i \le 2\right|C_{\max}$ turns out into the deterministic scheduling problem, which is the individual scheduling problem $J2|p,\ n_i \le 2|C_{\max}$ associated with the scenario $p$.

The deterministic problem $J2|p,\ n_i \le 2|C_{\max}$ is solvable in

$$O\left(\max\left\{n_{1,2},\ n_{2,1}\right\} \cdot \log\left(\max\left\{n_{1,2},\ n_{2,1}\right\}\right)\right)$$

time as it is noted in [22]. An optimal schedule for the individual scheduling problem $J2|p,\ n_i \le 2|C_{\max}$ may be determined by a Jackson's pair of job permutations $(\pi', \pi'')$ such that the permutation $\pi' = \left(\pi_{1,2}, \pi_1, \pi_{2,1}\right)$ determines an optimal sequence for processing jobs on the machine $M_1$ and the permutation $\pi'' = \left(\pi_{2,1}, \pi_2, \pi_{1,2}\right)$ determines an optimal sequence for processing jobs on the machine $M_2$. The job $J_i$ belongs to the permutation $\pi_h$, if the inclusion $J_i \in \Im_h$ holds.

In Jackson's pair of permutations, for the sequence $\pi_{1,2} = \left(J_{i_1}, J_{i_2}, \ldots, J_{i_{n_{1,2}}}\right)$ (and the sequence $\pi_{2,1} = \left(J_{i_1}, J_{i_2}, \ldots, J_{i_{n_{2,1}}}\right)$, respectively) of the jobs from the set $\Im_{1,2}$ (from the set $\Im_{2,1}$), the following condition must hold for all indices $k$ and $m$, $1 \le k < m \le n_{1,2}$ ($1 \le k < m \le n_{2,1}$):

$$\min\left\{p_{i_k 1},\ p_{i_m 2}\right\} \le \min\left\{p_{i_m 1},\ p_{i_k 2}\right\} \tag{1}$$

$$(\min\left\{p_{i_k 2},\ p_{i_m 1}\right\} \le \min\left\{p_{i_m 2},\ p_{i_k 1}\right\}),$$

where the permutation $\pi_{1,2}$ (and permutation $\pi_{2,1}$) is called a Johnson's permutation [22].

The optimal order of jobs from the set $\Im_1$ and jobs from the set $\Im_2$ may be arbitrary [22]. Therefore, in what follows, we consider only one permutation $\pi_1$ (one permutation $\pi_2$, respectively) of the jobs from the set $\Im_1$ that are located in the non-increasing order of their weights (from the set $\Im_2$ that are located in the non-increasing order of their weights).

Let the set $S_{1,2}$ (the set $S_{2,1}$, respectively) denote a set of all permutations of jobs from the set $\Im_{1,2}$ (the set $\Im_{2,1}$). Let $S = \left\langle S_{1,2},\ S_{2,1}\right\rangle$ denote a subset of the Cartesian product $\left(S_{1,2}, \pi_1, S_{2,1}\right) \times \left(S_{2,1}, \pi_2, S_{1,2}\right)$ such that each element in the set $S$ is a pair of job permutations $(\pi', \pi'') \in S$, where $\pi' = \left(\pi^i_{1,2}, \pi_1, \pi^j_{2,1}\right)$ and $\pi'' = \left(\pi^i_{2,1}, \pi_2, \pi^i_{1,2}\right)$, $1 \le i \le n_{1,2}!$, $1 \le j \le n_{2,1}!$.

For the uncertain (interval) job-shop scheduling problem $J2\left|a_{ij} \le p_{ij} \le b_{ij},\ n_i \le 2\right|C_{\max}$, we will consider only semi-active schedules which are determined by the set $S$.

**Definition 1** [17]. A schedule is semi-active if no operation can be processed earlier without changing the processing order or violating some given constraints.

It is known that for any regular criterion [17; 18], there exists a semi-active schedule which is optimal.

For any fixed scenario $p \in T$, there exists Jackson's pair of job permutations (belonging to the set $S$) that is optimal for the individual job-shop scheduling problem $J2|p,\ n_i \le 2|C_{\max}$. It is clear that in most cases, a single pair of job permutations, which is optimal for all possible scenarios $p \in T$ for the uncertain (interval) job-shop scheduling problem $J2\left|a_{ij} \le p_{ij} \le b_{ij},\ n_i \le 2\right|C_{\max}$, does not exist. Due to this fact, we will look for a dominant set of the job permutations based on the following definition.

**Definition 2.** A set of pairs of job permutations $DS(T) \subseteq S$ is a dominant set for the uncertain (interval) scheduling problem $J2\left|a_{ij} \le p_{ij} \le b_{ij},\ n_i \le 2\right|C_{\max}$ with the set $\Im$ of jobs, if for each scenario $p \in T$, the set $DS(T)$ contains at least one pair $(\pi', \pi'') \in S$ of the job permutations that is optimal for the individual deterministic problem $J2|p,\ n_i \le 2|C_{\max}$.

The sufficient conditions for a pair of job permutations $(\pi', \pi'') \in S$ to be an optimal pair of job permutations for any individual deterministic problem $J2|p, n_i \le 2|C_{\max}$ with any fixed scenario $p \in T$ that is feasible for the uncertain (interval) problem $J2|a_{ij} \le p_{ij} \le b_{ij}, n_i \le 2|C_{\max}$ have been investigated in [19, theorem 7, corollaries 3 and 4]. It has been proven that if one of the following conditions holds:

$$\sum_{J_i \in \Im_{1,2}} b_{i1} \le \sum_{J_j \in \Im_2 \cup \Im_{2,1}} a_{j2} \text{ and } \sum_{J_i \in \Im_{1,2}} a_{i2} \ge \sum_{J_j \in \Im_1 \cup \Im_{2,1}} b_{j1}, \tag{2}$$

$$\sum_{J_i \in \Im_{2,1}} b_{i2} \le \sum_{J_j \in \Im_1 \cup \Im_{1,2}} a_{j1} \text{ and } \sum_{J_i \in \Im_{2,1}} a_{i1} \ge \sum_{J_j \in \Im_2 \cup \Im_{1,2}} b_{j2}, \tag{3}$$

then any permutation $\pi_{1,2}$ from the set $S_{1,2}$ and any permutation $\pi_{2,1}$ from the set $S_{2,1}$ form a single-element dominant set $DS(T)$ for the uncertain problem $J2|a_{ij} \le p_{ij} \le b_{ij}, n_i \le 2|C_{\max}$.

If the first inequality in condition (2) (in condition (3), respectively) holds, then $\langle \{\pi_{1,2}\}, S_{2,1} \rangle \subseteq S$ ($\langle S_{1,2}, \{\pi_{2,1}\} \rangle \subseteq S$) is a dominant set of schedules for the uncertain problem $J2|a_{ij} \le p_{ij} \le b_{ij}, n_i \le 2|C_{\max}$. One needs to determine only orders for processing jobs from the set $\Im_{2,1}$ (the set $\Im_{1,2}$, respectively). In each this set all jobs have the same machine route.

The uncertain flow-shop problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ is a special case of the uncertain job-shop problem $J2|a_{ij} \le p_{ij} \le b_{ij}, n_i \le 2|C_{\max}$. In the flow-shop problem, all jobs have the same machine route on both machines and a schedule is determined by the permutation $\pi_k$. The uncertain (interval) flow-shop problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ with the job set $\Im = \Im_{1,2}$ ($\Im_{2,1} = \Im_1 = \Im_2 = \varnothing$) and the uncertain (interval) flow-shop problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ with the job set $\Im = \Im_{2,1}$ ($\Im_{1,2} = \Im_1 = \Im_2 = \varnothing$) are associated with the uncertain (interval) job-shop problem $J2|a_{ij} \le p_{ij} \le b_{ij}, n_i \le 2|C_{\max}$. As shown in [20], solving the uncertain (interval) job-shop problem $J2|a_{ij} \le p_{ij} \le b_{ij}, n_i \le 2|C_{\max}$ may be based on solving two associated uncertain flow-shop problems. It is sufficient to construct dominant sets for two associated uncertain (interval) flow-shop problems $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$. The dominant set for the uncertain (interval) flow-shop problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ turns out to a set of job permutations, which contains at least one optimal permutation for the deterministic flow-shop problem $F2|p|C_{\max}$ for each fixed scenario $p \in T$.

**Theorem 1** [19]. *Let the set $S'_{1,2} \subseteq S_{1,2}$ be a set of permutations from the dominant set for the uncertain flow-shop problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ with the job set $\Im = \Im_{1,2}$. And let $S'_{2,1} \subseteq S_{2,1}$ be a set of permutations from the dominant set for the uncertain flow-shop problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ with the job set $\Im = \Im_{2,1}$. Then the set $\langle S'_{1,2}, S'_{2,1} \rangle \subseteq S$ is a dominant set for the uncertain job-shop problem $J2|a_{ij} \le p_{ij} \le b_{ij}, n_i \le 2|C_{\max}$ with the job set $\Im = \Im_1 \cup \Im_2 \cup \Im_{1,2} \cup \Im_{2,1}$.*

We next consider the uncertain flow-shop problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ with the job set $\Im = \Im_{1,2}$. Due to remark, the following partition holds: $\Im_{1,2} = \Im^1_{1,2} \cup \Im^2_{1,2} \cup \Im^*_{1,2}$, where $\Im^1_{1,2} = \{J_i \in \Im_{1,2} | b_{i1} \le a_{i2}\}$, $\Im^2_{1,2} = \{J_i \in \Im_{1,2} | b_{i2} \le a_{i1}\}$, $\Im^*_{1,2} = \{J_i \in \Im_{1,2} | b_{i1} > a_{i2}, b_{i2} > a_{i1}\}$.

We prove the following necessary and sufficient conditions for the existence of a Johnson's permutation, which is optimal for any scenario $p \in T$, which is possible for the uncertain flow-shop problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ with the job set $\Im = \Im_{1,2}$.

**Theorem 2.** *There exists a Johnson's permutation, which is optimal for any scenario $p \in T$ for the uncertain (interval) flow-shop scheduling problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ with the job set $\Im = \Im_{1,2}$, if and only if, the following conditions hold:*

*a) for each pair of jobs $J_i \in \Im^1_{1,2}$ and $J_j \in \Im^1_{1,2}$ (jobs $J_i \in \Im^2_{1,2}$ and $J_j \in \Im^2_{1,2}$, respectively), either $b_{i1} \le a_{j1}$ or $b_{j1} \le a_{i1}$ (either $b_{i2} \le a_{j2}$ or $b_{j2} \le a_{i2}$, respectively);*

*b) inequality $|\Im^*_{1,2}| \le 1$ holds, and for job $J_{i^*} \in \Im^*_{1,2}$ (if any), both inequalities $a_{i^*1} \ge \max\{b_{i1} | J_i \in \Im^1_{1,2}\}$ and $a_{i^*2} \ge \max\{b_{i2} | J_i \in \Im^2_{1,2}\}$ hold.*

P r o o f. *Sufficiency*. We consider the permutation $\pi_k = \left(\pi^1,\ J_{i^*},\ \pi^2\right)$ such that, in the permutation $\pi^1$, jobs from the set $\Im^1_{1,2}$ are located in the increasing order of the values $b_{i1}$, and in the permutation $\pi^2$, jobs from the set $\Im^2_{1,2}$ are located in the decreasing order of the values $b_{i2}$. If $\left|\Im^*_{1,2}\right| = 0$, then $\pi_k = \left(\pi^1,\ \pi^2\right)$. Due to remark, the permutation $\pi_k = \left(\pi^1,\ J_{i^*},\ \pi^2\right)$ is uniquely determined.

For the considered permutation $\pi_k = \left(J_{i_1},\ J_{i_2},\ \ldots,\ J_{i_{n_{1,2}}}\right)$, condition (1) holds for any scenario $p \in T$. Indeed, for all indices $k$ and $m$, $1 \le k < m \le n_{1,2}$, both inequalities (4) hold

$$\min\left\{p_{i_k 1},\ p_{i_m 2}\right\} \le \min\left\{b_{i_k 1},\ b_{i_m 2}\right\} \text{ and } \min\left\{a_{i_m 1},\ a_{i_k 2}\right\} \le \min\left\{p_{i_m 1},\ p_{i_k 2}\right\}. \tag{4}$$

If the inclusion $J_{i_m} \in \Im^1_{1,2} \cup \Im^*_{1,2}$ holds, then the inequality $b_{i_k 1} \le a_{i_m 1}$ holds. This assertion follows from conditions *a*) or *b*) and from constructing the permutation $\pi^1$. If the inclusion $J_{i_m} \in \Im^2_{1,2}$ holds, then inequality $b_{i_m 2} \le a_{i_m 1}$ holds. Obviously, inequality $k < m$ holds as well. If inequalities (4) hold, then condition (1) holds for all feasible durations $p_{i_k 1},\ p_{i_m 1},\ p_{i_k 2},\ p_{i_m 2}$.

Similarly, one can analyse the case when $J_{i_k} \in \Im^*_{1,2} \cup \Im^2_{1,2}$ and $J_{i_m} \in \Im^2_{1,2}$, where $k < m$.

*Necessity*. Based on the contradiction method, we assume that the permutation $\pi_k = \left(J_{i_1},\ J_{i_2},\ \ldots,\ J_{i_{n_{1,2}}}\right)$ exists such that condition (1) holds for all indices $k$ and $m$, $1 \le k < m \le n_{1,2}$, for any scenario $p \in T$, and at least one condition *a*) or *b*) does not hold.

Assume that condition *a*) does not hold. If there exists a pair of jobs $J_{i_k} \in \Im^1_{1,2}$ and $J_{i_m} \in \Im^1_{1,2}$ with $k < m$, such that both inequalities $b_{i_k 1} > a_{i_m 1}$ and $b_{i_m 1} > a_{i_k 1}$ hold, we consider feasible operation durations $a_{i_m 1} \le p_{i_m 1} < p_{i_k 1} \le b_{i_k 1}$. Due to remark, there exists a real number $p_{i_m 2}$ such that inequalities $p_{i_m 1} \le b_{i_m 1} \le a_{i_m 2} < p_{i_m 2}$ hold. Condition (1) does not hold for indices $k$ and $m$. Hence, the permutation $\pi_k$ is not a Johnson's one for scenarios $T$. Similarly a contradiction may be obtained, if there exists a pair of jobs $J_{i_k} \in \Im^2_{1,2}$ and $J_{i_m} \in \Im^2_{1,2}$, $k < m$.

Now, assume that condition *b*) does not hold. If there exist two jobs $J_{i_k} \in \Im^*_{1,2}$ and $J_{i_m} \in \Im^*_{1,2}$, $k < m$, we consider feasible operation durations $p_{i_m 1} < p_{i_m 2}$ and $p_{i_k 2} < p_{i_k 1}$. Condition (1) does not hold for indices $k$ and $m$ for all scenarios $p' \in T$. For a job $J_{i_m} \in \Im^*_{1,2}$ with inequality $a_{i_m 1} < b_{i_k 1}$, where $J_{i_k} \in \Im^1_{1,2}$, we consider feasible operation durations $p_{i_m 1} < p_{i_m 2}$ and $p_{i_m 1} < p_{i_k 1}$. Condition (1) does not hold for indices $k$ and $m$ for all scenarios $p' \in T$.

Similarly, one can test the case, when for job $J_{i_k} \in \Im^*_{1,2}$, inequality $a_{i_k 2} < b_{i_m 2}$ holds, where $J_{i_m} \in \Im^2_{1,2}$. We obtain the contradiction to the assumption that for the considered permutation $\pi_k = \left(J_{i_1},\ J_{i_2},\ \ldots,\ J_{i_{n_{1,2}}}\right)$, condition (1) holds for all indices $k$ and $m$, $1 \le k < m \le n_{1,2}$, and for any fixed scenario $p \in T$. Theorem 2 is proved.

Theorem 2 implies the following claim.

**Corollary 1.** *If the conditions of theorem 2 hold, then there exists a permutation* $\pi_{1,2} \in S_{1,2}$, *which is the dominant singleton* $\left\{\pi_{1,2}\right\} = DS_{1,2}(T)$, $\left|DS_{1,2}(T)\right| = 1$, *for the uncertain (interval) flow-shop scheduling problem* $F2\left|a_{ij} \le p_{ij} \le b_{ij}\right|C_{\max}$ *with the job set* $\Im = \Im_{1,2}$.

## Uncertain (interval) two-machine flow-shop scheduling problems

We consider the binary relation $A^{1,2}_{\preceq}$ on the set $\Im_{1,2}$ based on the following definition.

**Definition 3.** For two jobs $J_u \in \Im_{1,2}$ and $J_v \in \Im_{1,2}$, $u \ne v$, inclusion $\left(J_u,\ J_v\right) \in A^{1,2}_{\preceq}$ holds if and only if for any scenario $p \in T$, condition (1) holds with $i_k = u$ and $i_m = v$.

Due to definition 3, if inclusion $\left(J_u,\ J_v\right) \in A^{1,2}_{\preceq}$ holds, then for every scenario $p \in T$, there exists a Johnson's permutation of the jobs from set $\Im_{1,2}$ such that the job $J_u$ locates before job $J_v$, $u \ne v$. In [20], it is shown that for any scenario $p \in T$, there exists a Johnson's permutation such that job $J_x \in \Im_{1,2}$ locates before the job $J_y \in \Im_{1,2}$, $x \ne y$, if and only if at least one of the following conditions holds:

$$b_{x1} \le a_{x2} \text{ and } b_{x1} \le a_{y1}, \tag{5}$$

$$b_{y2} \le a_{y1} \text{ and } b_{y2} \le a_{x2}. \tag{6}$$

For constructing the binary relation $A^{1,2}_{\preceq}$, one can check conditions (5) and (6) for pairs of jobs from the set $\Im_{1,2}$. Next, we prove two theorems about properties of the relation $A^{1,2}_{\preceq}$.

**Theorem 3.** *If $(J_u, J_v) \in A_{\prec}^{1,2}$, then there exists a dominant set for the uncertain (interval) flow-shop scheduling problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ with the job set $\mathfrak{I}_{1,2}$, such that job $J_u$ locates before job $J_v$, $u \ne v$, in all permutations from the dominant set $DS_{1,2}(T)$.*

P r o o f. Let the inclusion $(J_u, J_v) \in A_{\prec}^{1,2}$ hold. We consider an arbitrary possible scenario $p' \in T$. Due to definition 3, there exists a Johnson's permutation $\pi'$ of the jobs from the set $\mathfrak{I}_{1,2}$ with the job $J_u$ located before the job $J_v$, $u \ne v$. The permutation $\pi'$ is an optimal permutation for the individual flow-shop problem $F2|p'|C_{\max}$ with the job set $\mathfrak{I} = \mathfrak{I}_{1,2}$. Let $DS_{1,2}(T)$ be a set of all such permutation constructed for all scenarios $p \in T$. Thus, the set $DS_{1,2}(T)$ contains at least one optimal permutation for the individual deterministic problem $F2|p|C_{\max}$ for each scenario $p \in T$. Therefore, the set $DS_{1,2}(T)$ is a dominant set for the uncertain (interval) flow-shop problem $F2|a_{ij} \le p_{ij} \le b_{ij}|C_{\max}$ with the job set $\mathfrak{I} = \mathfrak{I}_{1,2}$. In each permutation from the set $DS_{1,2}(T)$, job $J_u$ locates before job $J_v$, $u \ne v$. This completes the proof of theorem 3.

**Theorem 4.** *The binary relation $A_{\prec}^{1,2}$ is a strict order.*

P r o o f. We have to show that the binary relation $A_{\prec}^{1,2}$ is anti-reflexive, asymmetric, and transitive. Due to definition 3, the binary relation $A_{\prec}^{1,2}$ is defined only for $u \ne v$. Thus, the relation $A_{\prec}^{1,2}$ is anti-reflexive. If the inclusion $(J_u, J_v) \in A_{\prec}^{1,2}$ holds, then condition (1) holds with $i_k = u$ and $i_m = v$ for any scenario $p \in T$ and the following inequalities hold:

$$\min\{p_{u1}, p_{v2}\} \le \min\{b_{u1}, b_{v2}\} \le \min\{a_{v1}, a_{u2}\} \le \min\{p_{v1}, p_{u2}\}. \tag{7}$$

On the other hand, due to remark, the following inequalities hold:

$$\min\{a_{u1}, a_{v2}\} < \min\{b_{u1}, b_{v2}\} \text{ and } \min\{a_{v1}, a_{u2}\} < \min\{b_{v1}, b_{u2}\}. \tag{8}$$

From inequalities (7) and (8), we conclude that $\min\{b_{v1}, b_{u2}\} > \min\{a_{u1}, a_{v2}\}$ and inequality (1) does not hold for $i_k = v$ and $i_m = u$ for scenarios $p \in T$. Thus, $(J_v, J_u) \notin A_{\prec}^{1,2}$ and the binary relation $A_{\prec}^{1,2}$ is asymmetric.

We prove the transitivity. Let there exist three jobs $J_u \in \mathfrak{I}_{1,2}$, $J_v \in \mathfrak{I}_{1,2}$ and $J_w \in \mathfrak{I}_{1,2}$ with the inclusions $(J_u, J_v) \in A_{\prec}^{1,2}$, $(J_v, J_w) \in A_{\prec}^{1,2}$ and condition $(J_u, J_w) \notin A_{\prec}^{1,2}$. For the jobs $J_u$ and $J_v$, similarly as for the jobs $J_v$ and $J_w$, at least one of conditions (5) and (6) holds with $x = u$ and $y = v$, and $x = v$ and $y = w$, respectively. We must consider the following four cases:

(I) $b_{u1} \le a_{u2}$, $b_{u1} \le a_{v1}$, $b_{v1} \le a_{v2}$ and $b_{v1} \le a_{w1}$;

(II) $b_{u1} \le a_{u2}$, $b_{u1} \le a_{v1}$, $b_{w2} \le a_{w1}$ and $b_{w2} \le a_{v2}$;

(III) $b_{v2} \le a_{v1}$, $b_{v2} \le a_{u2}$, $b_{v1} \le a_{v2}$ and $b_{v1} \le a_{w1}$;

(IV) $b_{v2} \le a_{v1}$, $b_{v2} \le a_{u2}$, $b_{w2} \le a_{w1}$ and $b_{w2} \le a_{v2}$.

In case (III), we obtain the contradiction to remark. Indeed, in case (III) due to remark, we obtain the following contradictory inequalities: $b_{v2} \le a_{v1} < b_{v1} \le a_{v2} < b_{v2}$.

We have to consider the remaining three cases (I), (II) and (IV).

Note that for the jobs $J_u$ and $J_w$, neither condition (5), nor condition (6) holds with $x = u$ and $y = w$, which could happen only in one of the following four cases.

1. Inequalities $b_{u1} > a_{u2}$ and $b_{w2} > a_{w1}$ contradict to the cases (I), (II) and (IV).

2. Inequalities $b_{u1} > a_{u2}$ and $b_{w2} > a_{u2}$ contradict to the cases (I) and (II).

Furthermore, from the inequalities of case (IV) and remark, we obtain the contradictory inequalities as follows: $b_{w2} > a_{u2} \ge b_{v2} > a_{v2} \ge b_{w2}$.

3. Inequalities $b_{u1} > a_{w1}$ and $b_{w2} > a_{w1}$ contradict to cases (II) and (IV).

Furthermore, from case (I) and remark, we obtain the contradictory inequalities as follows: $b_{u1} > a_{w1} \ge b_{v1} >$ $> a_{v1} \ge b_{u1}$.

4. Consider inequalities $b_{u1} > a_{w1}$ and $b_{w2} > a_{u2}$.

From the inequalities of cases (I) and (IV), we obtain the same contradictions as in the cases 3 and 2, respectively. From case (II), we obtain the following contradictory inequalities: $a_{u2} < b_{w2} \le a_{w1} < b_{u1} \le a_{u2}$.

Thus, for any three jobs $J_u \in \mathfrak{I}_{1,2}$, $J_v \in \mathfrak{I}_{1,2}$ and $J_w \in \mathfrak{I}_{1,2}$, we obtain that the inclusions $(J_u, J_v) \in A_{\prec}^{1,2}$ and $(J_v, J_w) \in A_{\prec}^{1,2}$ imply the inclusion $(J_u, J_w) \in A_{\prec}^{1,2}$. Therefore, the binary relation $A_{\prec}^{1,2}$ is transitive. Theorem 4 is proved.

**Definition 4.** Two jobs $J_x \in \Im_{1,2}$ and $J_y \in \Im_{1,2}$, $x \neq y$, are conflict jobs if the following relations hold: $\left(J_x, J_y\right) \notin A_{\prec}^{1,2}$ and $\left(J_y, J_x\right) \notin A_{\prec}^{1,2}$.

**Definition 5.** The subset $\Im_x \subseteq \Im_{1,2}$ is called a conflict set of jobs if for any job $J_y \in \Im_{1,2} \backslash \Im_x$, either relation $\left(J_x, J_y\right) \in A_{\prec}^{1,2}$ or relation $\left(J_y, J_x\right) \in A_{\prec}^{1,2}$ holds for each job $J_x \in \Im_x$, provided that any proper subset of the set $\Im_x$ does not possess such a property.

Obviously, there may exist several conflict sets in the set $\Im_{1,2}$. The permutation $\pi_k = \left(J_{[1]}, J_{[2]}, \ldots, J_{[n_{1,2}]}\right)$ is determined by the partial strict order $A_{\prec}^{1,2}$ if each inclusion $\left(J_x, J_y\right) \in A_{\prec}^{1,2}$ implies the following form of this permutation: $\pi_k = \left(\ldots, J_x, \ldots, J_y, \ldots\right)$.

Let $\Pi_{1,2}$ denote a set of all permutations determined by the partial strict order $A_{\prec}^{1,2}$.

**Theorem 5.** *There exists a dominant set $DS_{1,2}(T) = \Pi_{1,2}$ for the uncertain (interval) flow-shop scheduling problem $F2|a_{ij} \leq p_{ij} \leq b_{ij}|C_{\max}$ with the job set $\Im_{1,2}$.*

P r o o f. Based on the contradiction method, we assume that for an arbitrary scenario $p \in T$, there is no Johnson's permutation in the set $\Pi_{1,2}$ for the deterministic flow-shop scheduling problem $F2|p|C_{\max}$ with the scenario $p$.

Due to constructing permutations of the set $\Pi_{1,2}$, the above assumption means that there exists at least one pair of jobs $J_x \in \Im_{1,2}$ and $J_y \in \Im_{1,2}$ such that inclusion $\left(J_x, J_y\right) \in A_{\prec}^{1,2}$ holds, whereas condition (1) with $i_k = y$ and $i_m = x$ holds as the following strict inequality:

$$\min\left\{p_{y1}, p_{x2}\right\} < \min\left\{p_{x1}, p_{y2}\right\}. \tag{9}$$

Due to definition 3, we obtain the following non-strict inequality:

$$\min\left\{b_{x1}, b_{y2}\right\} \leq \min\left\{a_{y1}, a_{x2}\right\}. \tag{10}$$

From inequalities (9) and (10), we obtain the following contradicted inequalities:

$$\min\left\{p_{y1}, p_{x2}\right\} < \min\left\{p_{x1}, p_{y2}\right\} \leq \min\left\{b_{x1}, b_{y2}\right\} \leq \min\left\{a_{y1}, a_{x2}\right\} \leq \min\left\{p_{y1}, p_{x2}\right\}.$$

Therefore, there exists a Johnson's permutation for the problem $F2|p|C_{\max}$ in the set $\Pi_{1,2}$. Due to the arbitrariness of the choice of the possible scenario $p \in T$, the set $\Pi_{1,2}$ contains an optimal Johnson's permutation for the individual deterministic flow-shop problem $F2|p|C_{\max}$ for each fixed scenario $p \in T$. Due to definition 2, the set $\Pi_{1,2}$ is a dominant set for the uncertain problem $F2|a_{ij} \leq p_{ij} \leq b_{ij}|C_{\max}$ with the job set $\Im_{1,2}$. Theorem 5 is proved.

Let the strict order $A_{\prec}^{1,2}$ for the uncertain (interval) flow-shop scheduling problem $F2|a_{ij} \leq p_{ij} \leq b_{ij}|C_{\max}$ with the job set $\Im_{1,2}$ be represented as follows:

$$J_1, J_2, \ldots, J_k, \left\{J_{k+1}, J_{k+2}, \ldots, J_{k+r}\right\}, J_{k+r+1}, J_{k+r+2}, \ldots, J_{n_{1,2}}, \tag{11}$$

where all jobs between the brackets are conflict jobs and each of these jobs is found in relation $A_{\prec}^{1,2}$ with any job located outside the brackets. Thus, jobs in the brackets make up the conflict set. The order of jobs in the brackets may be different in the optimal permutation (depending on the used scenario $p \in T$) but they still correspond to the binary relation $A_{\prec}^{1,2}$.

The following sufficient conditions for checking the optimal order for processing jobs of the conflict set were proved in [19, theorems 10–12].

Let the strict order $A_{\prec}^{1,2}$ over the set $\Im_{1,2}$ have form (11). If for the permutation $\pi = \left(J_1, \ldots, J_k, J_{k+1}, J_{k+2}, \ldots \right.$ $\left. \ldots, J_{k+r}, J_{k+r+1}, \ldots, J_{n_{1,2}}\right) \in \Pi_{1,2}$, one of the following inequalities holds:

$$\sum_{i=1}^{k+r} b_{i,1} \leq \sum_{J_i \in \Im_2 \cup \Im_{2,1}} a_{i,2} + \sum_{j=1}^{k} a_{j,2}, \tag{12}$$

$$b_{k+s,1} \leq \sum_{J_i \in \Im_2 \cup \Im_{2,1}} a_{i,2} + \sum_{j=1}^{k+s-1} \left(a_{j,2} - b_{j,1}\right), \; s \in \{1, 2, \ldots, r\}, \tag{13}$$

$$\sum_{i=r-s+2}^{r+1} a_{k+i,1} \geq \sum_{j=r-s+1}^{r} b_{k+j,2},\ s \in \{1, 2, \ldots, r\}, \tag{14}$$

then the set $S' = \langle \{\pi\}, \Pi_{2,1} \rangle \subset S$ is a dominant set of schedules for the uncertain (interval) job-shop problem $J2 \big| a_{ij} \leq p_{ij} \leq b_{ij},\ n_i \leq 2 \big| C_{\max}$ with the job set $\Im$.

Condition (12) does not use the order of the jobs in the conflict set. So, the order of jobs in the conflict set $\{J_{k+1}, J_{k+2}, \ldots, J_{k+r}\}$ may be arbitrary. Based on the second criterion of the maximisation of the sum $\sum w_i$ of weights of the completed jobs, if condition (12) holds, we propose to locate the jobs in the conflict set in the non-increasing order of their weights.

On the other hands, to check conditions (13) and (14), one must first determine the order of jobs in the conflict set $\{J_{k+1}, J_{k+2}, \ldots, J_{k+r}\}$, e. g., as $(J_{k+1}, J_{k+2}, \ldots, J_{k+r})$. Therefore, to check conditions (13) and (14), it may be needed to consider all $r!$ possible permutations of the conflicting jobs. In [20], it is noted that it is enough to check conditions (13) and (14) for only one permutation, and it is also shown how such permutations can be constructed (see the algorithm from [19]). The following procedures 1 and 2 are close to the algorithms described in [19] and are intended to constructing permutations to check the fulfillment of conditions (13) and (14), respectively. Procedure 1 (procedure 2) constructs a permutation such that condition (13) (condition (14), respectively) are most likely to be satisfied.

**Procedure 1.** Construction of the permutation of conflict jobs by checking condition (13).
**Step 1: for** each job $J_i$ from the conflict set, test **if** inequality $a_{i,2} - b_{i,1} \geq 0$ holds **then** $J_i \in \pi_1$ **else** $J_i \in \pi_2$.
**Step 2:** construct the permutation $\pi_1$ as follows: **if** inequality $b_{i,1} \leq b_{j,1}$ holds **then** $\pi_1 = (\ldots, J_i, \ldots, J_j, \ldots)$.
**Step 3:** construct the permutation $\pi_2$ as follows: **if** inequality $a_{i,2} \geq a_{j,2}$ holds **then** $\pi_2 = (\ldots, J_i, \ldots, J_j, \ldots)$.

**Procedure 2.** Construction of the permutation of conflict jobs by checking condition (14).
**Step 1: for** each job $J_i$ from the conflict set, test **if** inequality $a_{i,1} - b_{i,2} \geq 0$ holds **then** $J_i \in \pi_1$ **else** $J_i \in \pi_2$.
**Step 2:** construct the permutation $\pi_1$ as follows: **if** inequality $b_{i,2} \geq b_{j,2}$ holds **then** $\pi_1 = (\ldots, J_i, \ldots, J_j, \ldots)$.
**Step 3:** construct the permutation $\pi_2$ as follows: **if** inequality $a_{i,1} \leq a_{j,1}$ holds **then** $\pi_2 = (\ldots, J_i, \ldots, J_j, \ldots)$.

Note that if there exist several conflict sets in the job set $\Im_{1,2}$, one can check conditions (12)–(14) sequentially for each conflict set. Indeed, conditions (12)–(14) do not use the order of jobs from the set $\{J_{k+r+2}, \ldots, J_{n_{1,2}}\}$. On the other hand, if the job set $\{J_1, J_2, \ldots, J_k\}$ (the job set $\{J_{k+r+1}, \ldots, J_{n_{1,2}}\}$, respectively) is empty, one cannot check condition (13) (condition (14), respectively).

One can consider the uncertain (interval) flow-shop problem $F2 \big| a_{ij} \leq p_{ij} \leq b_{ij} \big| C_{\max}$ with the job set $\Im = \Im_{2,1}$. The partial binary relation $A_{\prec}^{2,1}$ determined on the set $\Im_{2,1}$ of the jobs may be introduced similar to definition 3.

Note that conditions of theorems 2–5, corollary 1, and conditions (5), (6) and inequalities (12)–(14) may be reformulated similarly. The conflict jobs $J_x \in \Im_{2,1}$ and $J_y \in \Im_{2,1}$ and a conflict set of jobs $\Im_x \subseteq \Im_{2,1}$ can be investigated similarly.

Note that theorems 2–5 and corollary 1 are proved for the uncertain (interval) flow-shop problem $F2 \big| a_{ij} \leq$ $\leq p_{ij} \leq b_{ij} \big| C_{\max}$, where all jobs have interval durations unknown before scheduling. If there exists a non-empty subset of the jobs with fixed durations known before scheduling, then the binary relation $A_{\prec}^{1,2}$ on the set $\Im_{1,2}$ may become different.

## Scheduling algorithms

For scheduling jobs for a long period (e. g., for a month), we consider the uncertain (interval) two-machine job-shop scheduling problem $J2 \big| a_{ij} \leq p_{ij} \leq b_{ij},\ n_i \leq 2 \big| C_{\max}, \sum w_i, \sum U_i$ with the following ordered criteria: minimising the makespan (it is a main criterion), maximising the sum $\sum w_i$ of job weights (the second criterion) and maximising the total number of the jobs $\sum U_i$ that are completed not later their due dates (the third criterion).

Each day, both employees have 8 working hours of 800 units of time. Each unit of time corresponds to 30 s (10 min for each hour is set aside for the rest). It is assumed that 20 new jobs are arrived every day. Some of these jobs should be processed only by one of the employees (first or second), some jobs should be processed by the first employee and then by the second employee. The remaining jobs should be processed by the second employee and then by the first employee.

For every job, the lower bound $a_{ij}$ and the upper bound $b_{ij}$, $0 < a_{ij} < b_{ij}$, of the operation processing times and the job weights $w_i$ are determined before scheduling.

Due dates for all jobs are assumed to be equal to 800. The integer weight $w_i$ from 1 to 5 is assigned to each job that determines the importance of this job. The set $\Im_0$ of jobs available at the beginning of the new working day is sorted in the non-increasing order of the weights of jobs. The set $\Im(d)$ of jobs that will be completed on that day are selected as long as the following inequality holds: $\sum_{J_i \in \Im_d} \left(a_{i,1} + a_{i,2}\right) \le 2 \cdot 800$.

Based on the sufficient conditions presented in the previous sections, algorithm 1 has been developed, for constructing a daily pair of permutations of the selected jobs for both employees. As a result of executing algorithm 1, a pair of job permutations $(\pi', \pi'')$ of the form $\left(\left(\pi_{1,2}, \pi_1, \pi_{2,1}\right), \left(\pi_{2,1}, \pi_2, \pi_{1,2}\right)\right)$ will be constructed. First, we check sufficient conditions (2) and (3) and conditions of theorem 2 that the pair of job permutations $(\pi', \pi'')$ is optimal for the individual deterministic scheduling problem $J2|p, m_i \le 2|C_{\max}$ with any fixed scenario $p \in T$. If the pair of job permutations $(\pi', \pi'')$ is not constructed, the binary relation $A_{\prec}^{1,2}$ on the set $\Im_{1,2}$ (the binary relation $A_{\prec}^{2,1}$ on the set $\Im_{2,1}$, respectively) must be constructed and conflict sets of jobs are identified. Then, conditions (12)–(14) must be checked to resolve the conflicts. We arrange jobs from the sets $\Im_1$ and $\Im_2$, and in some cases jobs from the sets $\Im_{1,2}$ and $\Im_{2,1}$, in the non-increasing order of their weights to improve the value of the second criterion. The constructed pair of permutations $(\pi', \pi'')$ may be optimal for all scenarios (with the proof of the optimality, if the above sufficient conditions hold), or the constructed pair of permutations $(\pi', \pi'')$ may be optimal for the factual scenario but without the proof of the optimality, or the constructed pair of permutations $(\pi', \pi'')$ is non-optimal for the makespan criterion.

**Algorithm 1**

**Step 1:** construct the permutation $\pi_1$ of jobs of the set $\Im_1$ and the permutation $\pi_2$ of jobs of the set $\Im_1$.

**Step 2: if** the first inequality in (2) holds **then begin** to construct the permutation $\pi_{1,2}$ of jobs from the set $\Im_{1,2}$ **if** the second inequality in (2) holds **then** construct the permutation $\pi_{2,1}$ of jobs from the set $\Im_{2,1}$ **endif**.

**Step 3: if** the first inequality in (3) holds **then begin** to construct the permutation $\pi_{2,1}$ of jobs from the set $\Im_{2,1}$ **if** the second inequality in (3) holds **then** construct the permutation $\pi_{1,2}$ of jobs from the set $\Im_{1,2}$ **endif**.

**Step 4: if** both permutations $\pi_{1,2}$ and $\pi_{2,1}$ are constructed **then goto** step 13.

**Step 5: if** the permutation $\pi_{1,2}$ is constructed **then goto** step 12.

**Step 6: if** for jobs from the set $\Im_{1,2}$ conditions *a*) and *b*) hold **then** construct the permutation $\pi_{1,2} = \left(\pi_{1,2}^1, \Im_{1,2}^*, \pi_{1,2}^2\right)$, where $\pi_{1,2}^1$ is a permutation of jobs from the set $\Im_{1,2}^1$ located in the non-decreasing order of values $b_{i,1}$ and $\pi_{1,2}^2$ is a permutation of jobs from the set $\Im_{1,2}^2$ in the non-increasing order of the values $b_{i,2}$ **goto** step 11.

**Step 7:** construct binary relations $A_{\prec}^{1,2}$ over the set $\Im_{1,2}$ using conditions (5) and (6).

**Step 8:** select all conflict sets of jobs in the set $\Im_{1,2}$.

**Step 9: for** each conflict set of jobs **do if** condition (12) holds **then** construct the permutation of the conflict jobs **else begin** to implement procedure 1 and construct the permutation $(\pi_1, \pi_2)$ **if** condition (13) does not hold **then begin** to implement procedure 2 and construct the permutation $(\pi_2, \pi_1)$; **if** condition (14) does not hold **then** construct a Johnson's permutation for the conflict jobs for their processing times $p_{ij} = \frac{a_{ij} + b_{ij}}{2}$ **endif**, **endif**, **endif**, **enddo**, **endfor**.

**Step 10:** construct a permutation $\pi_{1,2}$ generated by the linear order $A_{\prec}^{1,2}$ with permutations obtained in step 9 for jobs from the conflict sets.

**Step 11: if** the permutation $\pi_{2,1}$ is constructed **then goto** step 13.

**Step 12:** repeat steps 6–11 by replacing the set $\Im_{1,2}$ by the set $\Im_{2,1}$, the machine $M_1$ by the machine $M_2$, the strict order $A_{\prec}^{1,2}$ by the strict order $A_{\prec}^{2,1}$, and *vice versa*.

**Step 13:** construct pair of permutations $(\pi', \pi'') = \left(\left(\pi_{1,2}, \pi_1, \pi_{2,1}\right), \left(\pi_{2,1}, \pi_2, \pi_{1,2}\right)\right)$.

Note that steps 2 and 3 take $\mathrm{O}\left(\max\left\{n_{1,2}, n_{2,1}\right\}\right)$ time. Step 6 takes $\mathrm{O}\left(n_{1,2}\log n_{1,2}\right)$ time. The construction of a binary relation at step 7 is based on comparing no more than $n_{1,2}\left(n_{1,2}-1\right)$ pairs of jobs from the set $\Im_{1,2}$, which takes no more than $\mathrm{O}\left(n_{1,2}\left(n_{1,2}-1\right)\right)$ time. Checking conditions (12)–(14) at step 9 requires $\mathrm{O}(r)$ time, where the conflict set contains $r$ jobs. Constructing a permutation of $r$ jobs (procedures 1 and 2) takes $\mathrm{O}(r\log r)$ time. Therefore, the total complexity of step 9 is $\mathrm{O}(r\log r)$. Since step 7 is performed at most once per set $\Im_{1,2}$ and per set $\Im_{2,1}$, we conclude that the complexity of algorithm 1 is $\mathrm{O}\left(n^2\right)$.

The jobs are processed respecting to the constructed permutations $(\pi', \pi'')$ until the beginning the next job does not go beyond the working hours for each employee.

For the uncertain (interval) job-shop problem $J2\left|a_{ij} \le p_{ij} \le b_{ij}, n_i \le 2\right|C_{\max}, \sum w_i, \sum U_i$ with ordered criteria $C_{\max}$, $\sum w_i$ and $\sum U_i$, one calculates the values of criteria $C_{\max}(d)$, $\sum w_i(d)$ and $\sum U_i(d)$ per day $d$. After a schedule realisation, the actual durations $p_{ij}^*$ of all the operations become known. From that time, it becomes possible to determine the optimal Jackson's pair of permutations and calculate the factual values of criteria $\sum w_i^*(d)$ and $\sum U_i^*(d)$ per day $d$. Relative errors of the constructed schedules respecting to the factually optimal schedules are calculated as follows:

$$\Delta C_{\max}(d) = \frac{C_{\max}(d) - C_{\max}^*(d)}{C_{\max}^*(d)},\ \ \Delta\sum w_i(d) = \frac{\sum w_i(d) - \sum w_i^*(d)}{\sum w_i^*(d)},\ \ \Delta\sum U_i(d) = \frac{\sum U_i(d) - \sum U_i^*(d)}{\sum U_i^*(d)}.$$

All jobs that were not selected, as well as jobs that were not completed during the working day are available for processing next day. New 20 jobs will be added to them.

## Computational experiments and results

We next describe the conducted computational experiments and discuss the computational results obtained for randomly generated instances of the uncertain job-shop problem $J2\left|a_{ij} \le p_{ij} \le b_{ij}, n_i \le 2\right|C_{\max}$. The following algorithm was used in the experiments.

**Algorithm 2 for computational experiments**

**Input:** job set $\Im = \Im_1 \cup \Im_2 \cup \Im_{1,2} \cup \Im_{2,1}$. Lower bound $a_{ij}$ and upper bound $b_{ij}$, $0 < a_{ij} < b_{ij}$, of feasible durations of operations $O_{ij}$ for jobs $J_i \in \Im$ and machines $M_j \in \mathrm{M}$.

**Output:** conclusion that the problem $J2\left|a_{ij} \le p_{ij} \le b_{ij}, n_i \le 2\right|C_{\max}$ was solved either exactly or heuristically. Total number of conflict sets and number of properly resolved conflict sets.

**Step 1:** set $a = 0$, $b = 0$, $c = 0$, $cs = 0$.

**Step 2:** **if** the first inequality in (2) holds **then begin** $a := a + 1$ **if** the second inequality in (2) holds **then** $b := b + 1$ **endif**.

**Step 3:** **if** the first inequality in (3) holds **then begin** $b := b + 1$; **if** the second inequality in (3) holds **then** $a := a + 1$ **endif**.

**Step 4:** **if** $a \ge 1$ and $b \ge 1$ **then goto** step 17.

**Step 5:** **if** $a \ge 1$ **then goto** step 14.

**Step 6:** **if** for jobs from the set $\Im_{1,2}$ conditions *a*) and *b*) hold, **then begin** $a := a + 1$ **goto** step 13 **endif**.

**Step 7:** construct binary relations $A_{\prec}^{1,2}$ over the set $\Im_{1,2}$ using conditions (5) and (6).

**Step 8:** select all conflict sets of jobs in the set $\Im_{1,2}$.

**Step 9:** set number of conflict sets $nc = 0$ and $n = 0$.

**Step 10:** **for** each conflict set of jobs **if** condition (12) holds **then** $n := n + 1$ **else** implement procedure 1 for constructing the permutation $(\pi_1, \pi_2)$ **if** condition (13) holds **then** $n := n + 1$ **else** implement procedure 2 for constructing the permutation $(\pi_2, \pi_1)$ **if** condition (14) holds **then** $n := n + 1$ **endif**, **endif**, **endif**, **endfor**.

**Step 11:** set $c := c + n$; $cs := cs + nc$.

**Step 12:** **if** $n = n_c$ **then** $a := a + 1$.

**Step 13:** **if** $b \ge 1$ **then goto** step 17.

**Step 14:** perform steps 6–12 by replacing the set $\Im_{1,2}$ by the set $\Im_{2,1}$, machine $M_1$ by machine $M_2$, the strict order $A_{\prec}^{1,2}$ by strict order $A_{\prec}^{2,1}$, $a$ by $b$, and *vice versa*.

**Step 15:** **if** $a \ge 1$ and $b \ge 1$ **then goto** step 17.

**Step 16: stop.** «The problem is solved heuristically; total number of conflict sets c», «number of properly resolved conflict sets cs».

**Step 17: stop.** «The problem is solved exactly; total number of conflict sets c», «number of properly resolved conflict sets cs».

Algorithm 2 is polynomial in the number $n$ of jobs and its asymptotic complexity is $O\left(n^2\right)$. All developed algorithms were coded in C# and tested on a personal computer with Intel Core i7-7700™ 4 Quad, 3.6 GHz, and 32.00 GB RAM. In the computational experiments, we tested series of randomly generated instances for the 1000-day period. Generated instance for every day consisted of 20 jobs.

The generation of lower bounds $a_{ij}$ and upper bounds $b_{ij}$ for possible values of the durations $p_{ij}$ of the operations $O_{ij}$, $p_{ij} \in \left[a_{ij}, b_{ij}\right]$, was organised as follows. A value of the lower bound $a_{ij}$ was randomly chosen from the segment [10, 1000] using the uniform distribution. With the given value of the maximum relative length δ of a segment of possible durations of the operations $O_{ij}$, the upper bound $b_{ij}$ was calculated using the following equality: $b_{ij} = a_{ij}\left(1 + \frac{\delta}{100}\right)$. A maximum relative length δ of the segment of possible durations of operations $O_{ij}$ was equal to the following values: 5 %, 10 %, 11 %, 12 %, 13 %, 14 %, 15 %, 16 %, 17 %, 18 %, 19 %, 20 %, 30 %, 40 %, 50 %. The bounds $a_{ij}$ and $b_{ij}$ were decimal fractions with the maximum possible number of digits after the decimal point.

Based on remark, for instances of the problem $J2\left|a_{ij} \le p_{ij} \le b_{ij},\ n_i \le 2\right|C_{\max}$, a strict inequality $a_{ij} < b_{ij}$ was guaranteed for each job $J_i \in \mathfrak{J}$ and each machine $M_j \in \mathrm{M}$. We tested 9 classes of the randomly generated instances of the problem $J2\left|a_{ij} \le p_{ij} \le b_{ij},\ n_i \le 2\right|C_{\max}$ with different ratios between values $n_1 : n_2 : n_{1,2} : n_{2,1}$ of jobs in the subsets $\mathfrak{J}_1$, $\mathfrak{J}_2$, $\mathfrak{J}_{1,2}$, $\mathfrak{J}_{2,1}$ of the set $\mathfrak{J}$. The computational results are presented in the following table.

**Computational results for the randomly generated instances**

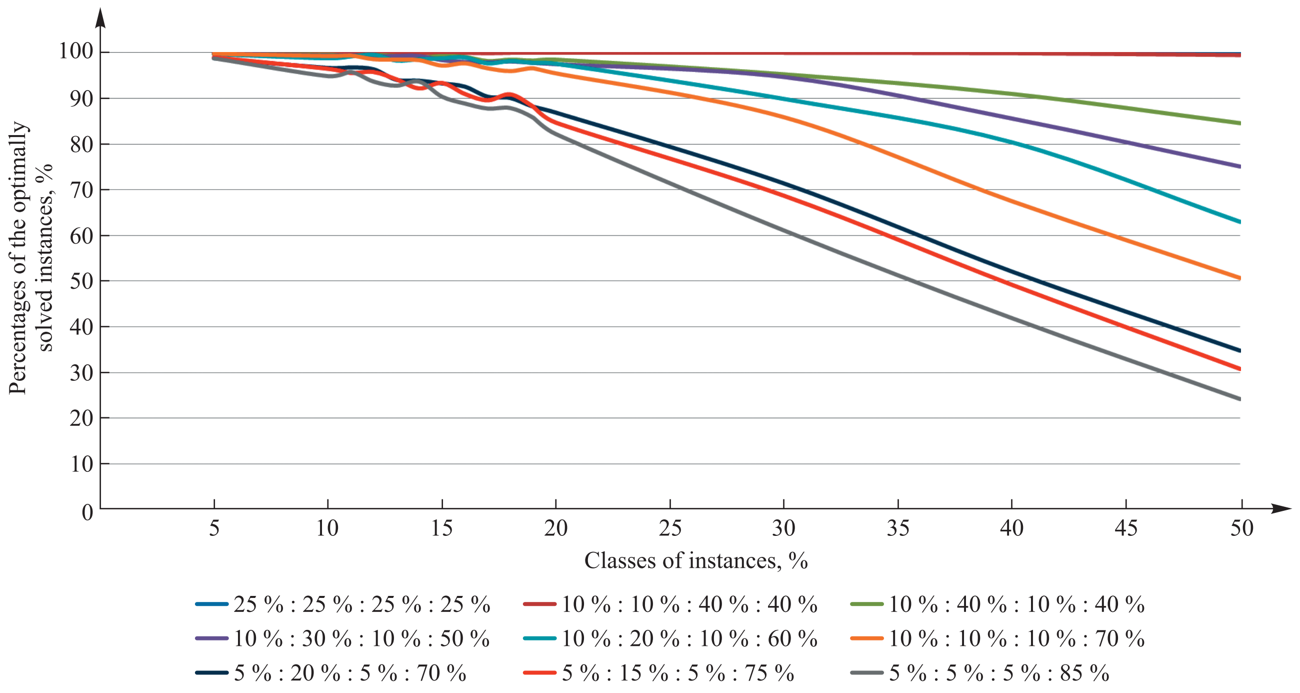
| Class of the tested instances | δ | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5 % | 10 % | 11 % | 12 % | 13 % | 14 % | 15 % | 16 % | 17 % | 18 % | 19 % | 20 % | 30 % | 40 % | 50 % |
| $n_1 : n_2 : n_{1,2} : n_{2,1}$ = 25 % : 25 % : 25 % : 25 % | | | | | | | | | | | | | | | |
| Solved tests, % | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Conflict sets | 6 | 19 | 22 | 25 | 28 | 22 | 35 | 35 | 53 | 51 | 60 | 70 | 139 | 250 | 339 |
| Solved conflicts | 6 | 19 | 22 | 25 | 28 | 22 | 35 | 35 | 53 | 51 | 60 | 70 | 139 | 250 | 339 |
| Solved conflicts, % | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| $n_1 : n_2 : n_{1,2} : n_{2,1}$ = 10 % : 10 % : 40 % : 40 % | | | | | | | | | | | | | | | |
| Solved tests, % | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 99.9 | 100 | 100 | 100 | 100 | 99.9 | 99.5 |
| Conflict sets | 235 | 531 | 567 | 645 | 692 | 743 | 811 | 807 | 851 | 952 | 992 | 1032 | 1341 | 1450 | 1424 |
| Solved conflicts | 235 | 531 | 567 | 645 | 692 | 743 | 811 | 807 | 850 | 952 | 992 | 1032 | 1341 | 1449 | 1419 |
| Solved conflicts, % | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 99.88 | 100 | 100 | 100 | 100 | 99.93 | 99.65 |
| $n_1 : n_2 : n_{1,2} : n_{2,1}$ = 10 % : 40 % : 10 % : 40 % | | | | | | | | | | | | | | | |
| Solved tests, % | 99.7 | 99.6 | 99.3 | 99.5 | 99.3 | 99 | 99.3 | 99.1 | 98.2 | 98.5 | 98.3 | 98.5 | 95.3 | 91 | 84.6 |
| Conflict sets | 466 | 830 | 884 | 887 | 1002 | 1008 | 1057 | 1116 | 1166 | 1166 | 1176 | 1244 | 1433 | 1467 | 1501 |
| Solved conflicts | 463 | 826 | 877 | 882 | 995 | 998 | 1049 | 1107 | 1148 | 1151 | 1158 | 1229 | 1385 | 1375 | 1345 |
| Solved conflicts, % | 99.36 | 99.52 | 99.21 | 99.44 | 99.30 | 99.01 | 99.24 | 99.19 | 98.46 | 98.71 | 98.47 | 98.79 | 96.65 | 93.73 | 89.61 |
| $n_1 : n_2 : n_{1,2} : n_{2,1}$ = 10 % : 30 % : 10 % : 50 % | | | | | | | | | | | | | | | |
| Solved tests, % | 99.9 | 99.4 | 99.5 | 99.2 | 99.3 | 99.3 | 98.4 | 98 | 97.9 | 98.1 | 97.8 | 97.5 | 94.7 | 85.6 | 75.1 |
| Conflict sets | 767 | 1232 | 1235 | 1361 | 1399 | 1436 | 1489 | 1596 | 1623 | 1627 | 1638 | 1680 | 1762 | 1770 | 1724 |

Ending of the table

| Class of the tested instances | δ | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5 % | 10 % | 11 % | 12 % | 13 % | 14 % | 15 % | 16 % | 17 % | 18 % | 19 % | 20 % | 30 % | 40 % | 50 % |
| Solved conflicts | 766 | 1226 | 1230 | 1351 | 1392 | 1429 | 1472 | 1576 | 1601 | 1608 | 1615 | 1653 | 1709 | 1622 | 1468 |
| Solved conflicts, % | 99.87 | 99.51 | 99.60 | 99.27 | 99.50 | 99.51 | 98.86 | 98.75 | 98.64 | 98.83 | 98.60 | 98.39 | 96.99 | 91.64 | 85.15 |
| $n_1 : n_2 : n_{1,2} : n_{2,1}$ = 10 % : 20 % : 10 % : 60 % | | | | | | | | | | | | | | | |
| Solved tests, % | 99.7 | 98.8 | 99.2 | 99.6 | 98.3 | 98.6 | 98.8 | 99 | 97.7 | 98.2 | 97.8 | 97.6 | 89.9 | 80.4 | 63 |
| Conflict sets | 1034 | 1601 | 1653 | 1758 | 1829 | 1880 | 1904 | 1949 | 2013 | 2054 | 2012 | 1984 | 2113 | 1968 | 1845 |
| Solved conflicts | 1031 | 1588 | 1645 | 1754 | 1812 | 1865 | 1891 | 1939 | 1990 | 2035 | 1989 | 1959 | 2010 | 1770 | 1471 |
| Solved conflicts, % | 99.71 | 99.19 | 99.52 | 99.77 | 99.07 | 99.20 | 99.32 | 99.49 | 98.86 | 99.07 | 98.86 | 98.74 | 95.13 | 89.94 | 79.73 |
| $n_1 : n_2 : n_{1,2} : n_{2,1}$ = 10 % : 10 % : 10 % : 70 % | | | | | | | | | | | | | | | |
| Solved tests, % | 99.8 | 99.3 | 99.4 | 98.6 | 98.5 | 98.4 | 97.2 | 97.7 | 96.6 | 96 | 96.6 | 95.5 | 85.9 | 67.5 | 50.7 |
| Conflict sets | 1432 | 1988 | 2043 | 2156 | 2192 | 2234 | 2273 | 2306 | 2324 | 2342 | 2376 | 2345 | 2222 | 2119 | 1952 |
| Solved conflicts | 1430 | 1981 | 2035 | 2142 | 2176 | 2218 | 2245 | 2283 | 2289 | 2301 | 2338 | 2299 | 2080 | 1786 | 1447 |
| Solved conflicts, % | 99.86 | 99.65 | 99.61 | 99.35 | 99.27 | 99.28 | 98.77 | 99.00 | 98.49 | 98.25 | 98.40 | 98.04 | 93.61 | 84.29 | 74.13 |
| $n_1 : n_2 : n_{1,2} : n_{2,1}$ = 5 % : 20 % : 5 % : 70 % | | | | | | | | | | | | | | | |
| Solved tests, % | 98.8 | 96.7 | 96.8 | 96.4 | 94.1 | 93.9 | 93.3 | 92.6 | 90.4 | 90.1 | 88.2 | 86.9 | 71.4 | 52.1 | 34.8 |
| Conflict sets | 1353 | 1993 | 2119 | 2165 | 2210 | 2247 | 2325 | 2337 | 2263 | 2355 | 2286 | 2343 | 2210 | 2091 | 1941 |
| Solved conflicts | 1340 | 1959 | 2081 | 2126 | 2147 | 2182 | 2257 | 2261 | 2164 | 2252 | 2164 | 2206 | 1909 | 1591 | 1268 |
| Solved conflicts, % | 99.04 | 98.29 | 98.21 | 98.20 | 97.15 | 97.11 | 97.08 | 96.75 | 95.63 | 95.63 | 94.66 | 94.15 | 86.38 | 76.09 | 65.33 |
| $n_1 : n_2 : n_{1,2} : n_{2,1}$ = 5 % : 15 % : 5 % : 75 % | | | | | | | | | | | | | | | |
| Solved tests, % | 99 | 96.5 | 95.6 | 95.8 | 94.2 | 92.2 | 93.4 | 91 | 89.6 | 90.9 | 88.2 | 84.7 | 68.7 | 49.2 | 30.8 |
| Conflict sets | 1475 | 2242 | 2330 | 2354 | 2411 | 2360 | 2450 | 2532 | 2489 | 2516 | 2580 | 2514 | 2340 | 2103 | 1952 |
| Solved conflicts | 1465 | 2207 | 2285 | 2309 | 2352 | 2278 | 2381 | 2436 | 2377 | 2419 | 2456 | 2351 | 2014 | 1577 | 1244 |
| Solved conflicts, % | 99.32 | 98.44 | 98.07 | 98.09 | 97.55 | 96.53 | 97.18 | 96.21 | 95.50 | 96.14 | 95.19 | 93.52 | 86.07 | 74.99 | 63.73 |
| $n_1 : n_2 : n_{1,2} : n_{2,1}$ = 5 % : 5 % : 5 % : 85 % | | | | | | | | | | | | | | | |
| Solved tests, % | 98.8 | 94.9 | 95.8 | 93.7 | 92.8 | 93.7 | 90.4 | 88.9 | 87.8 | 87.9 | 85.8 | 82.2 | 61.1 | 41.9 | 24.2 |
| Conflict sets | 1896 | 2585 | 2627 | 2688 | 2714 | 2810 | 2763 | 2766 | 2827 | 2808 | 2791 | 2777 | 2532 | 2220 | 2063 |
| Solved conflicts | 1883 | 2532 | 2583 | 2623 | 2638 | 2743 | 2658 | 2647 | 2701 | 2681 | 2641 | 2591 | 2111 | 1599 | 1272 |
| Solved conflicts, % | 99.31 | 97.95 | 98.33 | 97.58 | 97.20 | 97.62 | 96.20 | 95.70 | 95.54 | 95.48 | 94.63 | 93.30 | 83.37 | 72.03 | 61.66 |

For each class of the tested instances and for a fixed value of the maximum relative length δ, the computational results are presented in four rows. The row «Solved tests, %» determines the percentage of days from the 1000-day period when the pair $(\pi', \pi'')$ of the job permutations constructed using algorithm 2 for a daily schedule was optimal for all possible scenarios $p \in T$ for the generated uncertain problem $J2\left|a_{ij} \le p_{ij} \le b_{ij},\ n_i \le 2\right|C_{\max}$.

The row «Conflict sets» presents a total number of conflict sets of the jobs in the partial strict orders $A_{\prec}^{1,2}$ on the job sets $\Im_{1,2}$ and partial strict orders $A_{\prec}^{2,1}$ on the job sets $\Im_{2,1}$ constructed by algorithm 2 for 1000 days. The row «Solved conflicts» is equal to the total number of cases, where algorithm 2 constructed the permutation of all jobs from the conflict set, which was optimal for all possible scenarios $p \in T$. Note, the instance may have be more than one conflict set, and failure to resolve even one of them leads to unoptimality of the entire instance. The row «Solved conflicts, %» presents a percentage of the ratio of solved conflicts to the total number of conflict sets in 1000-day series. Average percentages of the instances solved optimally by algorithm 2 are presented in figure.



Percentages of the optimally solved instances for different classes of instances

From figure, one can conclude that if value δ does not exceed 20 %, algorithm 2 found the optimal permutation in more than 90 % of tested instances. As δ increases, this value begins to fall. At δ = 50 % for some classes, the number of optimally solved instances is more than 20 %. For some classes of problems (25 % : 25 % : 25 % : 25 % and 10 % : 10 % : 40 % : 40 %) algorithm 2 optimally solved all tested examples for all relative errors δ.

## Conclusions

We investigated the uncertain problems of constructing schedules for the execution of selected jobs by two performers. Only the lower bound $a_{ij}$ and the upper bound $b_{ij}$ for durations of any job $J_i \in \Im$ were known before scheduling. We proved theorem 2 for necessary and sufficient conditions for the existing optimal schedules for two performers and theorems 3 and 5 for sufficient conditions for the existing dominant set of schedules with a fixed order of two jobs. For the existing dominant set of schedules with fixed orders for job pairs, the binary relation was constructed. It was proven that this binary relation is a strict order (theorem 4).

Based on the proven results, efficient algorithms were developed for solving the uncertain job-shop problem $J2\left|a_{ij} \le p_{ij} \le b_{ij},\, n_i \le 2\right|C_{\max}$ either exactly or heuristically. For testing the effectiveness of the developed algorithms for time-management, the computational experiments were conducted for evaluation of a 1000-day period for drawing up daily schedules for two performers. Every day, 20 jobs were received for the execution. For planning jobs for a day, the uncertain job-shop problem $J2\left|a_{ij} \le p_{ij} \le b_{ij},\, n_i \le 2\right|C_{\max}$ was solved. The job-shop problem $J2\left|a_{ij} \le p_{ij} \le b_{ij},\, n_i \le 2\right|C_{\max},\, \sum w_i,\, \sum U_i$ was solved for time-management during a month. In the uncertain scheduling problem, three criteria $C_{\max}$, $\sum w_i$ and $\sum U_i$ were optimised in the fixed priority order. Minimisation of the schedule length $C_{\max}$ was a main criterion, maximisation of the $\sum w_i$ was a second criterion, and maximisation of the $\sum U_i$ was a third criterion. A personal computer was used for selecting important jobs for two performers and drawing up optimal schedules for their implementation.

The computational experiments conducted on randomly generated uncertain scheduling problems showed that the use of the job permutations constructed by the developed algorithms provided optimal schedules in more than 90 % tested cases (20 % tested cases, respectively) if a maximum relative length of job duration segments $\left[a_{ij},\, b_{ij}\right]$ does not exceed 20 % (50 %, respectively).

A promising research direction may be connected with the application of the mixed graph colouring method [23] to scheduling personal jobs in the time-management framework. One can assume that the scheduling problems arising in the time-management have equal processing times of the operations since breaks are needed for people after approximately equal times of the activity.

## References

1. Zerubavel E. The Benedictine ethic and the modern spirit of scheduling: on schedules and social organization. *Sociological Inquiry*. 1980;50(2):157–169. DOI: 10.1111/j.1475-682X.1980.tb00383.x.

2. Eilon S. Time-management. *Omega.* 1993;21(3):255–259. DOI: 10.1016/0305-0483(93)90084-X.

3. Reed WJ. The Pareto, Zipf and other power laws. *Economics Letters.* 2001;74(1):15–19. DOI: 10.1016/S0165-1765(01)00524-9.

4. Rastogi P. Management musings. *Colourage.* 2009;56(1):58–62.

5. Ho B. Time management of final year undergraduate English projects: supervisees' and the supervisor's coping strategies. *System.* 2003;31(2):231–245. DOI: 10.1016/S0346-251X(03)00022-8.

6. Indreica ES, Cazan AM, Truta C. Effects of learning styles and time management on academic achievement. *Procedia – Social and Behavioral Sciences*. 2011;30:1096–1102. DOI: 10.1016/j.sbspro.2011.10.214.

7. Kaya H, Kaya N, Pallos AO, Kucuk L. Assessing time-management skills in terms of age, gender and anxiety levels: a study of nursing and midwifery students in Turkey. *Nurse Education in Practice.* 2012;12(5):284–288. DOI: 10.1016/j.nepr.2012.06.002.

8. Zampetakis LA, Bouranta N, Moustakis VS. On the relationship between individual creativity and time management. *Thinking Skills and Creativity.* 2010;5(1):23–32. DOI: 10.1016/j.tsc.2009.12.001.

9. Jackson VP. Time management: a realistic approach. *Journal of the American College of Radiology.* 2009;6(6):434–436. DOI: 10.1016/j.jacr.2008.11.018.

10. Konig CJ, Oberacher L, Kleinmann M. Personal and situational determinants of multitasking at work. *Journal of Personnel Psychology.* 2010;9(2):99–103. DOI: 10.1027/1866-5888/a000008.

11. Sherwood BJ. Personal time-management allows you to work smarter with less effort. *Sherwood on Management.* 2005;July – August:44–45.

12. Ahmad NL, Yusuf ANM, Shobri NDM, Wahab S. The relationship between time management and job performance in event management. *Procedia – Social and Behavioral Sciences.* 2012;65:937–941. DOI: 10.1016/j.sbspro.2012.11.223.

13. Macan T, Gibson JM, Cunningham J. Will you remember to read this article later when you have time? The relationship between prospective memory and time management. *Personality and Individual Differences.* 2010;48(6):725–730. DOI: 10.1016/j.paid.2010.01.015.

14. Claessens BJC, van Eerde W, Rutte CG, Roe RA. A review of the time management literature. *Personnel Review*. 2007;36(2): 255–276. DOI: 10.1108/00483480710726136.

15. Sotskov YuN, Egorova NG, Matsveichuk NM. Algorithms for planning working time under interval uncertainty. *Informatics.* 2020;17(2):86–102. Russian. DOI: 10.37661/1816-0301-2020-17-2-86-102.

16. Waterworth S. Time management strategies in nursing practice. *Journal of Advanced Nursing.* 2003;43(5):432–440. DOI: 10.1046/j.1365-2648.2003.02740.x.

17. Tanaev VS, Sotskov YN, Strusevich VA. *Scheduling theory: multi-stage systems.* Dordrecht: Kluwer Academic Publishers; 1994. 406 p.

18. Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics.* 1979;5:287–326. DOI: 10.1016/S0167-5060(08)70356-X.

19. Sotskov YN, Matsveichuk NM, Hatsura VD. Two-machine job-shop scheduling problem to minimize the makespan with uncertain job durations. *Algorithms.* 2020;13(1):4. DOI: 10.3390/a13010004.

20. Ng CT, Matsveichuk NM, Sotskov YN, Cheng TCE. Two-machine flow-shop minimum-length scheduling with interval processing times. *Asia-Pacific Journal of Operational Research.* 2009;26(6):587–604. DOI: 10.1142/S0217595909002432.

21. Matsveichuk NM, Sotskov YN, Werner F. The dominance digraph as a solution to the two-machine flow-shop problem with interval processing times. *Optimization.* 2011;60(12):1493–1517. DOI: 10.1080/02331931003657691.

22. Jackson JR. An extension of Johnson's results on job lot scheduling. *Naval Research Logistics Quaterly.* 1956;3(3):201–203. DOI: 10.1002/nav.3800030307.

23. Sotskov YuN. Mixed graph colouring as scheduling multiprocessor tasks with equal processing times. *Journal of the Belarusian State University. Mathematics and Informatics*. 2021;2:67–81. DOI: 10.33581/25206508202126781.

*Received 12.04.2024 / revised 04.07.2024 / accepted 04.07.2024.*