

Article

Schedule Execution for Two-Machine Job-Shop to Minimize Makespan with Uncertain Processing Times

Yuri N. Sotskov ^{1,*} , Natalja M. Matsveichuk ^{2,†} and Vadzim D. Hatsura ^{3,†}

¹ United Institute of Informatics Problems, National Academy of Sciences of Belarus, Surganova Street 6, 220012 Minsk, Belarus

² Belorussian State Agrarian Technical University, Nezavisimosti Avenue 99, 220023 Minsk, Belarus; matsveichuk@tut.by

³ Belorussian State University of Informatics and Radioelectronics, P. Brovki Street 6, 220013 Minsk, Belarus; vadimgatsura@gmail.com

* Correspondence: sotskov48@mail.ru; Tel.: +375-17-284-2120

Received: 4 June 2020; Accepted: 29 July 2020; Published: 7 August 2020



Abstract: This study addresses a two-machine job-shop scheduling problem with fixed lower and upper bounds on the job processing times. An exact value of the job duration remains unknown until completing the job. The objective is to minimize a schedule length (makespan). It is investigated how to best execute a schedule, if the job processing time may be equal to any real number from the given (closed) interval. Scheduling decisions consist of the off-line phase and the on-line phase of scheduling. Using the fixed lower and upper bounds on the job processing times available at the off-line phase, a scheduler may determine a minimal dominant set of schedules (minimal DS), which is based on the proven sufficient conditions for a schedule dominance. The DS optimally covers all possible realizations of the uncertain (interval) processing times, i.e., for each feasible scenario, there exists at least one optimal schedule in the minimal DS. The DS enables a scheduler to make the on-line scheduling decision, if a local information on completing some jobs becomes known. The stability approach enables a scheduler to choose optimal schedules for most feasible scenarios. The on-line scheduling algorithms have been developed with the asymptotic complexity $O(n^2)$ for n given jobs. The computational experiment shows the effectiveness of these algorithms.

Keywords: scheduling; job-shop; makespan criterion; uncertain processing times

1. Introduction

Many real-world production planning and scheduling problems have various uncertainties. Different approaches are used for solving the uncertain planning and scheduling problems. In particular, a stability approach [1–4] for solving sequencing and scheduling problems with the interval uncertainty is based on the stability analysis of the optimal job permutations (schedules) to possible variations of the job processing times (durations). In this paper, this approach is applied to the uncertain two-machine job-shop scheduling problem, where a job processing time is only known once the job is completed. Although, the exact value of the job processing time is unknown before scheduling, it is known that the processing time must have a value no less than the lower bound and no greater than the upper bound available before scheduling. It should be noted that uncertainties of the job processing times are due to some external forces in contrast to scheduling problems with controllable processing times [5–7], where the objective is to determine optimal processing times and then to find an optimal schedule for the jobs with the chosen processing times.

1.1. Research Motivation

It is not realistic to assume processing times are exactly known and fixed for many scheduling problems arising in real-world situations. For such an uncertain scheduling problem, job processing times are random variables. Moreover, it is often hard to obtain probability distributions for all random processing times of the jobs to be processed. In such cases, schedules constructed due to assuming certain probability distributions are often not close to the optimal schedule. Although, the probability distribution of the job processing time may not be known before scheduling, the upper and lower bounds on the job processing time are easy to obtain in most practical scheduling environments. The available information on these lower and upper bounds on the job processing times should be utilized in finding optimal schedules for the scheduling problem with an interval uncertainty.

Since there may not exist a unique schedule that remains optimal for all possible realizations of the job processing times (all possible scenarios), it is desirable to construct a minimal dominant set of schedules (permutations of the jobs to be processed), which dominate all other ones. At the off-line phase of scheduling (i.e., before starting an execution of the constructed schedule), a minimal dominant set of schedules may be determined based on the proven dominance relations [8].

If the constructed minimal dominant set of schedules is a singleton, then a single schedule remaining optimal for all possible scenarios exists. Otherwise, one can reduce the size of the determined minimal dominant set of schedules at the on-line phase of scheduling based on the additional information about completing some jobs. This additional on-line information allows a scheduler to find new dominance relations in order to best execute a schedule. It is clear that on-line scheduling decisions must be realized very quickly. In other words, only polynomial algorithms may be applied at the on-line phase of scheduling.

1.2. Contributions of This Research

In this paper, it is shown how to determine a minimal dominant set of schedules that would contain at least one optimal schedule for every scenario that is possible. The necessary and sufficient conditions are proven for the existence of a single pair of job permutations, which is optimal for the two-machine job-shop scheduling problem with any possible scenario. The algorithms have been developed for testing a set of the proven sufficient conditions for a schedule dominance and for the realization of a schedule, which is either optimal or very close to optimal one for the factual scenario. The developed algorithms are polynomial in the number n of the given jobs. Their asymptotic complexities do not exceed $O(n^2)$. The computational experiments on a large number of randomly generated instances of the uncertain (interval) two-machine job-shop scheduling problem show the efficiency and effectiveness of the developed off-line and on-line algorithms and programs. For different distributions of the factual job processing times, the developed on-line algorithms perform with the maximal errors of the achieved makespan less than 1% provided that $n \in \{20, 30, \dots, 100\}$. For all tested classes of the randomly generated instances, the average makespan errors $\Delta_{ave}\%$ for all tested numbers $n \in \{10, 20, \dots, 100\}$ of jobs \mathcal{J} are less than 0.02%. Each tested series of 1000 randomly generated instances was solved within no more than one second.

The paper is organized as follows. Settings of the considered scheduling problems with the interval uncertainty and main notation are introduced in Section 2. A literature review is presented in Section 3. The results published for the uncertain (interval) scheduling flow-shop problem are discussed in Section 3.2. These results are used in Section 4 for finding the optimal job permutations at the off-line phase of scheduling. In Section 4.2, the precedence digraphs are described for determining a minimal dominant set of schedules. An illustrative example is considered in Section 4.3. The on-line phase of scheduling is investigated in Section 5, where two theorems for the dominant sets of schedules have been proven. Section 6 contains the algorithms developed for the on-line phase of scheduling, illustrative examples (Section 6.2) and the discussion of the conducted computational experiments (Section 6.3). Appendix B consists of the tables with the detailed computational results. Some concluding remarks are made in Section 7.

2. Settings of Scheduling Problems and Main Notations

A set $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$ of the given jobs must be processed on different machines from a set $\mathcal{M} = \{M_1, M_2\}$. All jobs are available for processing from the same time $t = 0$. Using the standard notation $\alpha|\beta|\gamma$ [9], this deterministic two-machine job-shop scheduling problem to minimize the makespan is denoted as follows: $J2|n_i \leq 2|C_{max}$, where $\alpha = J2$ means a job-shop processing system with two available different machines and n_i a number of possible stages for processing a job $J_i \in \mathcal{J}$. The criterion $\gamma = C_{max}$ determines the minimization of a schedule length (makespan) as follows:

$$C_{max} := \min_{s \in S} C_{max}(s) = \min_{s \in S} \{\max\{C_i(s) : J_i \in \mathcal{J}\}\}, \quad (1)$$

where $C_i(s)$ denotes the completion time-point of the job $J_i \in \mathcal{J}$ in the schedule s and S denotes a set of all semi-active schedules existing for the deterministic problem $J2|n_i \leq 2|C_{max}$. (A schedule s is called a semi-active one [10–12] if the completion time-point $C_i(s)$ of any job $J_i \in \mathcal{J}$ cannot be reduced without changing an order of the jobs on some machine.)

Let O_{ij} denote an operation of the job $J_i \in \mathcal{J}$ processed on the machine $M_j \in \mathcal{M}$. Each of the available machines can process the job $J_i \in \mathcal{J}$ no more than once, a preemption of the operation O_{ij} being not allowed. The job $J_i \in \mathcal{J}$ has its own processing route through the available machines in set \mathcal{M} . The partition $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$ of the jobs is given and fixed, where each job $J_i \in \mathcal{J}_{1,2}$ must be processed first on machine M_1 and then on machine M_2 , i.e., all jobs from the set $\mathcal{J}_{1,2}$ have the same machine route (M_1, M_2) . Each job $J_i \in \mathcal{J}_{2,1}$ has an opposite machine route (M_2, M_1) . The set \mathcal{J}_j , where $j \in \{1, 2\}$, consists of all jobs, which must be processed only on one machine $M_j \in \mathcal{M}$. The following notation $m_h = |\mathcal{J}_h|$ will be used, where $h \in \{1; 2; 1, 2; 2, 1\}$.

In this research, it is investigated the uncertain (interval) two-machine job-shop scheduling problem denoted as $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$, where the duration p_{ij} of each operation O_{ij} is unknown before scheduling. It is only known that the inclusion $p_{ij} \in [l_{ij}, u_{ij}]$ holds for any possible realization of the chosen schedule, where $u_{ij} \geq l_{ij} \geq 0$. It is also assumed that a probability distribution of the random duration of a job from the set \mathcal{J} is also unknown before scheduling. Let a set T of all possible scenarios $p = (p_{1,1}, p_{1,2}, \dots, p_{n,1}, p_{n,2})$ of the job processing times be determined as follows:

$$T = \{p : l_{ij} \leq p_{ij} \leq u_{ij}, J_i \in \mathcal{J}, M_j \in \mathcal{M}\}.$$

It should be noted that the problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ is mathematically incorrect since one cannot calculate makespan $C_{max}(s)$ in the equality (1) before completing the jobs J_i in the set \mathcal{J} provided that the strict inequality $u_{ij} > l_{ij}$ holds. Moreover, in most cases there does not exist a schedule, which is optimal for all possible scenarios $p \in T$ for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$. Therefore, one cannot solve most such uncertain (interval) scheduling problems in the generally accepted sense.

In [13], it is proven that the deterministic job-shop problem $J2|n_i \leq 2|C_{max}$ is solvable in $O(n \log n)$ time. The optimal semi-active schedule for this deterministic problem is determined as the pair (π', π'') of two job permutations (called a Jackson's pair of permutations), where $\pi' = (\pi_{1,2}, \pi_1, \pi_{2,1})$ is an optimal permutation of the jobs $\mathcal{J}_1 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$ processed on machine M_1 and $\pi'' = (\pi_{2,1}, \pi_2, \pi_{1,2})$ is an optimal permutation of the jobs $\mathcal{J}_2 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$ on machine M_2 . Such an optimal semi-active schedule is presented in Figure 1. In what follows, it is assumed that job J_i belongs to the permutation π_h , if the following inclusion holds: $J_i \in \mathcal{J}_h$.

In a Jackson's pair of permutations (π', π'') , the optimal order for processing jobs from the set \mathcal{J}_1 (from the set \mathcal{J}_2 , respectively) may be arbitrary (due to this, we fix them in the increasing order of their indexes). For the permutation $\pi_{1,2}$ (permutation $\pi_{2,1}$, respectively), the following inequality holds:

$$\min\{p_{i_e1}, p_{i_f2}\} \leq \min\{p_{i_f1}, p_{i_e2}\} \quad (2)$$

for all indexes e and f provided that $1 \leq e < f \leq m_{1,2}$ ($1 \leq f < e \leq m_{2,1}$, respectively). The permutation $\pi_{1,2}$ (permutation $\pi_{2,1}$) is called a Johnson's permutation; see [14].

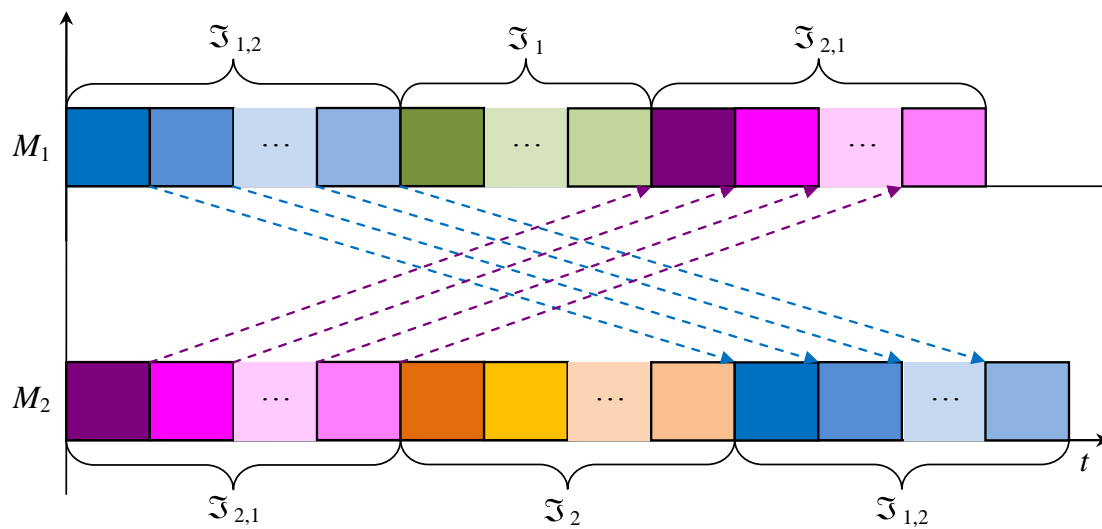


Figure 1. An example of the optimal semi-active schedule without idle times on both machines.

The deterministic scheduling problem $J2|n_i \leq 2|C_{max}$ associated with a fixed scenario p of the job processing times is an individual deterministic problem. In what follows, this problem is denoted as follows: $J2|p, n_i \leq 2|C_{max}$. For any fixed scenario $p \in T$, there exists a Jackson's pair (π', π'') of permutations, which is optimal for the problem $J2|p, n_i \leq 2|C_{max}$, i.e., the equality $C_{max}(\pi', \pi'') = C_{max}^p$ holds, where C_{max}^p denotes the optimal makespan value for the problem $J2|p, n_i \leq 2|C_{max}$.

Let $S_{1,2}$ denote a set of all permutations of $m_{1,2}$ jobs from the set $\mathcal{J}_{1,2}$, where $|S_{1,2}| = m_{1,2}!$. The set $S_{2,1}$ is a set of all permutations of $m_{2,1}$ jobs from the set $\mathcal{J}_{2,1}$, $|S_{2,1}| = m_{2,1}!$.

Let the set $S = \langle S_{1,2}, S_{2,1} \rangle$ be a subset of the Cartesian product $(S_{1,2}, \pi_1, S_{2,1}) \times (S_{2,1}, \pi_2, S_{1,2})$, each element of the set S being a pair of job permutations $(\pi', \pi'') \in S$, where $\pi' = (\pi_{1,2}^i, \pi_1, \pi_{2,1}^j)$ and $\pi'' = (\pi_{2,1}^j, \pi_2, \pi_{1,2}^i)$ with inequalities $1 \leq i \leq m_{1,2}!$ and $1 \leq j \leq m_{2,1}!$. It is known that the set S determines all semi-active schedules and vice versa; see [12]. Since index i (and index j) is the same in each permutation from the pair $(\pi', \pi'') \in S$ and it is a fixed permutation π_1 (permutation π_2), the equality $|S| = m_{1,2}! \cdot m_{2,1}!$ holds. The following definition of a J-solution is used for the uncertain (interval) job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$.

Definition 1. An inclusion-minimal set of the pairs of job permutations $S(T) \subseteq S$ is called a J-solution for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the set \mathcal{J} of the given jobs, if for each scenario $p \in T$, the set $S(T)$ contains at least one pair $(\pi', \pi'') \in S$ of job permutations that is optimal for the individual deterministic problem $J2|p, n_i \leq 2|C_{max}$ with a fixed scenario p .

From Definition 1, it follows that for any proper subset S' of the set $S(T)$, $S' \subset S(T)$, there exists a scenario $p' \in T$ such that the set S' does not contain an optimal pair of job permutations for the individual deterministic problem $J2|p', n_i \leq 2|C_{max}$ with a fixed scenario p' .

3. A Literature Review and Closed Results

It should be noted that the uncertain flow-shop scheduling problem denoted as $F2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ is well studied [15], unlike the uncertain job-shop scheduling problem.

3.1. Approaches to Scheduling Problems with Different Forms of Uncertainties

For the well-known stochastic approach, it is assumed that the job processing times are random variables with certain probability distributions determined before scheduling. There are two types of the stochastic scheduling problems [10], where one is on stochastic jobs and another is on stochastic machines. In the stochastic job scheduling problem, each job processing time is a random variable with a known probability distribution. With the objective of minimizing the expected makespan value, the flow-shop problem was studied in [16–18]. In the stochastic machine scheduling problem, each job processing time is a constant, while each completion time of the given job is a random variable due to the machine breakdown or machine non-availability. In [19–21], the flow-shop scheduling problems to stochastically minimize either makespan or total completion time were investigated.

If it is impossible to determine probability distributions for all random job processing times, other approaches have to be used [11,22–25]. In the approach of seeking a robust schedule [22,26–28], a decision-maker looks for a schedule that hedges against the worst-case possible scenario.

A fuzzy approach [29–35] allows a scheduler to find best schedules with respect to fuzzy processing times of the jobs to be processed. The work of [35] addresses to the job-shop scheduling problem with uncertain processing times modeled as triangle fuzzy numbers, where the criterion is to minimize the expected makespan value. Based on the disjunctive graph model of the job-shop problem, a definition of criticality is proposed for this job-shop problem along with neighborhood structure for a local search. It is shown that the proposed neighborhood structure has two properties: feasibility and connectivity, which allow a scheduler to improve the efficiency of the local search and to ensure asymptotic convergence (in probability) to a globally optimal solution of the uncertain job-shop problem. The conducted computational experiments supported these theoretical results.

The stability approach was developed in [1,4,36,37] for the C_{max} criterion, and in [2,38–40] for the total completion time criterion, $\sum C_i := \sum_{J_i \in \mathcal{J}} C_i(\pi')$. The aim of this approach is to construct a minimal dominant set $S(T)$ of schedules, which optimally covers all feasible scenarios T . The dominant set $S(T)$ is used in the multi-phase decision framework; see [41]. The set $S(T)$ is constructed at the first off-line phase of scheduling. Based on the set $S(T)$, it is possible to find a schedule remaining optimal for most feasible scenarios. The set $S(T)$ enables a scheduler to execute best a schedule in most cases of the uncertain flow-shop scheduling problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ [41].

The stability radius of the optimal semi-active schedule was studied in [4], where a formula for calculating the stability radius and corresponding algorithms were described and tested.

In [36], the sufficient conditions were proven when a transposition of the given jobs minimizes the makespan criterion. The work of [42] addressed the objective criterion $\sum C_i$ in the uncertain two-machine flow-shop scheduling problem. The case of separate setup times with the criterion of minimizing a total completion time or makespan was investigated in [43].

For the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$, an additional criterion is often introduced. In particular, a robust schedule minimizing the worst-case deviation from the optimal value was proposed in [44] to hedge against the interval or discrete uncertainties. In [45], a binary NP-hardness was proven for finding a pair $(\pi_q, \pi_q) \in S$ of the identical job permutations that minimizes the worst-case absolute regret for the uncertain two-machine flow-shop problem with the criterion C_{max} and only two possible scenarios. In [46], a branch and bound method was developed for the uncertain job-shop scheduling problem to minimize makespan and optimize robustness based on a mixed graph model and the propositions proposed in [47]. The effectiveness of the developed algorithm was clarified by solving test uncertain job-shop scheduling problems.

The work of [48] addresses robust scheduling for a flexible job-shop scheduling problem with a random machine breakdown. Two objectives makespan and robustness were considered. Robustness was indicated by the expected value of the relative difference between the deterministic and factual makespan values. Two measures for robustness have been developed. The first suggested measure considers the probability of machine breakdowns. The second measure considers the location of

float times and machine breakdowns. A multi-objective evolutionary algorithm is presented and experimentally compared with several other existing measures.

A function of predictive scheduling in order to obtain a stable and robust schedule for a shop floor was investigated in [49]. An innovative maintenance planning and production scheduling method has been proposed. The proposed method uses a database to collect information about failure-free times, a prediction module of failure-free times, predictive rescheduling module, a module for evaluating the accuracy of prediction and maintenance performance. The proposed approach is based on probability theory and applied for solving a job-shop scheduling problem. For unpredicted failures, a rescheduling procedure was also developed. The evaluation procedure provides information about the degradation of a performance measure and the stability of a schedule.

The simulation and experimental design methods play a useful role in solving job-shop scheduling problems with uncertain parameters (see survey [50], where many studies about dynamic and static job-shop scheduling problems with material handling are described and systematized).

In [51], a quality robustness and a solution robustness were investigated in order to compare the operational efficiency of the job-shop in the events of machine failures. Two well-known proactive approaches were compared to compute the operational efficiency of the job-shop with unpredicted machine failures. In the computational experiments, the predictive-reactive approach (without a prediction) and the proactive-reactive one (with a prediction) were applied for the job-shop model with possible disruptions. The computational results of computer simulations for the above two approaches were compared in order to select better schedules for reducing costs and waste due to machine failures.

The paper [52] presents a methodological pattern to assess the effectiveness of Order Review and Release (ORR) techniques in a job-shop environment. It is presented a comparison among three ORR approaches, i.e., a time bucketing approach, a probabilistic approach and a temporal approach. Simulation results highlighted that the performances of the ORR techniques tested depend on how perturbed the environment, where they are implemented, is. Based on a computer simulation, it was shown that the ORR techniques greatly differ in their robustness against environment perturbations.

The paper [53] presents an effective heuristic algorithm for the job-shop problem with uncertain arrival times of the jobs, processing times, due dates and part priorities. A separable problem formulation that balances modeling accuracy and solution complexity is described with the goal to minimize expected part tardiness and earliness cost. The optimization is subject to arrival times and operation precedence constraints (for each possible realization), and machine capacity constraints (in the expected value sense). The solution algorithm based on a Lagrangian relaxation and stochastic dynamic programming was developed to obtain dual solutions. The computational complexity of the developed algorithm is only slightly higher than the one without considering uncertainties of the numerical parameters. Numerical testing supported by a simulation demonstrated that near optimal solutions were obtained, and uncertainties are effectively handled for problems of practical sizes.

The published results on the application of the stability approach for the uncertain two-machine flow-shop problem are presented in Section 3.2. These results are described in detail since they are used for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ in Sections 4–6.

3.2. Closed Results for Uncertain (Interval) Flow-Shop Scheduling Problems

The uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ is a generalization of the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$, where all given jobs have the same machine route. Two uncertain flow-shop problems are associated with an uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$. In one of these flow-shop problems, an optimal schedule for processing the jobs $\mathcal{J}_{1,2}$ must be determined, i.e., it is assumed that $\mathcal{J}_{2,1} = \mathcal{J}_1 = \mathcal{J}_2 = \emptyset$. In another associated flow-shop problem, an optimal schedule for processing jobs $\mathcal{J}_{2,1}$ must be determined, i.e., it is assumed that $\mathcal{J}_{1,2} = \mathcal{J}_1 = \mathcal{J}_2 = \emptyset$. Our approach to the solution of the uncertain job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ is based on the following remark.

Remark 1. The solution of the uncertain job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ may be based on the solutions of the associated flow-shop scheduling problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_{1,2}$, where $\mathcal{J}_{2,1} = \mathcal{J}_1 = \mathcal{J}_2 = \emptyset$, and that with the job set $\mathcal{J} = \mathcal{J}_{2,1}$ (i.e., $\mathcal{J}_{1,2} = \mathcal{J}_1 = \mathcal{J}_2 = \emptyset$).

The sense of Remark 1 becomes clear from Figure 2, where the semi-active schedule s for the job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ is presented. Indeed, in Figure 2, the length $C_{max}(s)$ of the schedule s is equal to the length of the corresponding semi-active schedule determined for the associated flow-shop scheduling problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_{1,2}$. Thus, if one will solve both associated flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_{1,2}$ and associated flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_{2,1}$, then the original job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ will be also solved.

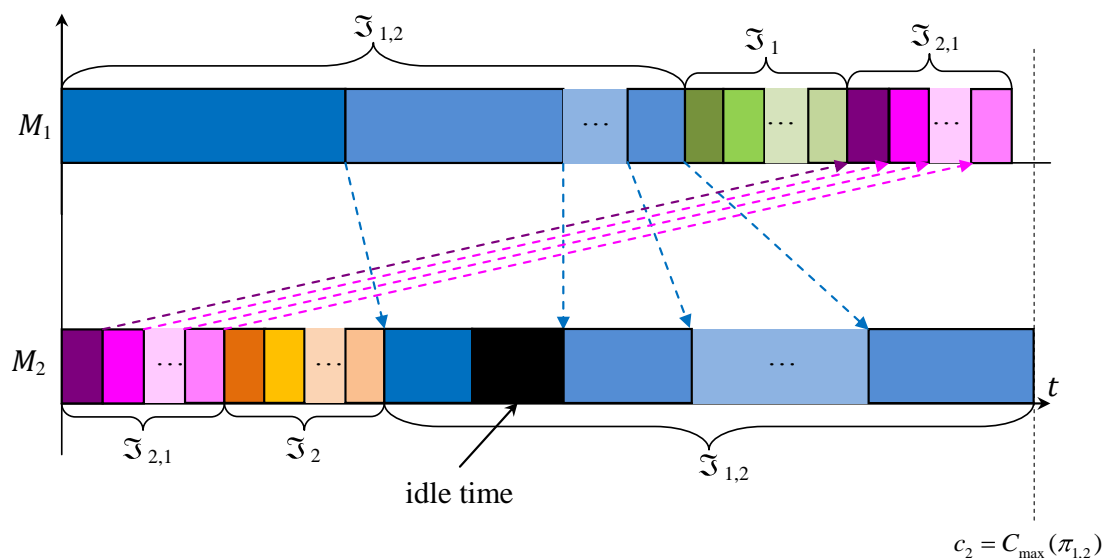


Figure 2. The optimal semi-active schedule for the job-shop scheduling problem.

We next observe in detail the results obtained for the two-machine flowshop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_{1,2}$. For using the above notations introduced for the uncertain job-shop problem, we need the following remark for the uncertain flow-shop problem.

Remark 2. The considered problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ has the following two mandatory properties:

- (i) the set S is a set of $n!$ pairs (π_q, π_q) of the identical permutations of $n = m_{1,2}$ jobs from the set $\mathcal{J} = \mathcal{J}_{1,2}$ since the machine route for processing all jobs $\mathcal{J}_{1,2}$ is the same (M_1, M_2) ;
- (ii) the J -solution (see Definition 1) is a set of Johnson's permutations of the jobs $\mathcal{J} = \mathcal{J}_{1,2}$, i.e., for each scenario $p \in T$ the set $S(T)$ contains at least one optimal pair (π_q, π_q) of identical Johnson's permutations π_q such that the inequality (2) holds for all indexes e and f .

The following Theorems 1 and 2 have been proven in [54].

Theorem 1 ([54]). There exists a J -solution $S(T)$ for the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with a fixed order $J_v \rightarrow J_w$ of the jobs J_v and J_w in all permutations $\pi_q, (\pi_q, \pi_q) \in S(T)$, if and only if at least one of the following two conditions hold:

$$u_{v1} \leq l_{v2} \text{ and } u_{v1} \leq l_{w1}; \quad (3)$$

$$u_{w2} \leq l_{w1} \text{ and } u_{w2} \leq l_{v2}. \quad (4)$$

Theorem 2 provides the necessary and sufficient conditions for existing a single-element J-solution $S(T) = \{(\pi_q, \pi_q)\}$ for the uncertain flow-shop scheduling problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$. The partition $\mathcal{J} = \mathcal{J}^0 \cup \mathcal{J}^1 \cup \mathcal{J}^2 \cup \mathcal{J}^*$ of the set $\mathcal{J} = \mathcal{J}_{1,2}$ is given, where

$$\begin{aligned}\mathcal{J}^0 &= \{J_i \in \mathcal{J} : u_{i1} \leq l_{i2}, u_{i2} \leq l_{i1}\}, \\ \mathcal{J}^1 &= \{J_i \in \mathcal{J} : u_{i1} \leq l_{i2}, u_{i2} > l_{i1}\} = \{J_i \in \mathcal{J} \setminus \mathcal{J}^0 : u_{i1} \leq l_{i2}\}, \\ \mathcal{J}^2 &= \{J_i \in \mathcal{J} : u_{i1} > l_{i2}, u_{i2} \leq l_{i1}\} = \{J_i \in \mathcal{J} \setminus \mathcal{J}^0 : u_{i2} \leq l_{i1}\}, \\ \mathcal{J}^* &= \{J_i \in \mathcal{J} : u_{i1} > l_{i2}, u_{i2} > l_{i1}\}.\end{aligned}$$

Note that for each job $J_g \in \mathcal{J}^0$, the inequalities $u_{g1} \leq l_{g2}$ and $u_{g2} \leq l_{g1}$ imply the equalities $l_{g1} = u_{g1} = l_{g2} = u_{g2}$. Thus, the equalities $p_{g1} = p_{g2} =: p_g$ hold.

Theorem 2 ([54]). *There exists a single-element J-solution $S(T) \subset S$, $|S(T)| = 1$, for the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$, if and only if the following two conditions hold:*

- (i) *for any pair of jobs J_i and J_j from the set \mathcal{J}^1 (from the set \mathcal{J}^2 , respectively), either $u_{i1} \leq l_{j1}$ or $u_{j1} \leq l_{i1}$ (either $u_{i2} \leq l_{j2}$ or $u_{j2} \leq l_{i2}$, respectively);*
- (jj) *inequality $|\mathcal{J}^*| \leq 1$ holds and for the job $J_{i^*} \in \mathcal{J}^*$ both inequalities $l_{i^*1} \geq \max\{u_{i1} : J_i \in \mathcal{J}^1\}$, $l_{i^*2} \geq \max\{u_{i2} : J_j \in \mathcal{J}^2\}$ hold with inequality $\max\{l_{i^*1}, l_{i^*2}\} \geq p_g$ valid for each job $J_g \in \mathcal{J}^0$.*

Theorem 2 characterizes the simplest case of the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$, i.e., there is a job permutation π_q dominating all others.

Let $\mathcal{J} \times \mathcal{J}$ denote a Cartesian product of the set \mathcal{J} . If $\mathcal{J}^0 = \emptyset$, then there exists the following binary relation $\mathcal{A}_< \subseteq \mathcal{J} \times \mathcal{J}$ over the set $\mathcal{J} = \mathcal{J}_{1,2}$.

Definition 2. *For the jobs $J_x \in \mathcal{J}$ and $J_y \in \mathcal{J}$, the inclusion $(J_x, J_y) \in \mathcal{A}_<$ holds if and only if at least one of the conditions (3) and (4) holds with $v = x$ and $w = y$ and neither the condition (3) nor the condition (4) holds with $v = y$ and $w = x$ (or at least one of the conditions (3) and (4) holds both with $v = x$ and $w = y$ and with $v = y, w = x$ and $x < y$).*

The above relation $(J_x, J_y) \in \mathcal{A}_<$ may be represented as follows: $J_x \prec J_y$. The binary relation $\mathcal{A}_<$ is a strict order [55] that determines the precedence digraph $\mathcal{G} = (\mathcal{J}, \mathcal{A}_<)$ with the vertex set \mathcal{J} and the arc set $\mathcal{A}_<$. The permutation $\pi_q = (J_{q_1}, J_{q_2}, \dots, J_{q_n})$, $(\pi_q, \pi_q) \in S$, is a total strict order over the set \mathcal{J} . The total strict order determined by the permutation π_q is a linear extension of the partial strict order $\mathcal{A}_<$, if the inclusion $(J_{q_x}, J_{q_y}) \in \mathcal{A}_<$ implies the inequality $x < y$. Let $\Pi(\mathcal{G})$ denote a set of all permutations $\pi_q \in S_{1,2}$ determining linear extensions of the partial strict order $\mathcal{A}_<$. The equality $\Pi(\mathcal{G}) = \{\pi_q\}$ is characterized in Theorem 2, where the strict order $\mathcal{A}_<$ over the set \mathcal{J} is represented as follows: $J_{q_1} \prec \dots \prec J_{q_i} \prec J_{q_{i+1}} \prec \dots \prec J_{q_n}$. The following two claims have been proven in [55].

Theorem 3 ([55]). *For any scenario $p \in T$, the set $\Pi(\mathcal{G})$ contains a Johnson's permutation for the deterministic flow-shop problem $F2|p|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_{1,2} = \mathcal{J}^* \cup \mathcal{J}^1 \cup \mathcal{J}^2$.*

Corollary 1 ([55]). *There exists a J-solution $S(T)$ for the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_{1,2} = \mathcal{J}^* \cup \mathcal{J}^1 \cup \mathcal{J}^2$, such that the inclusion $\pi_q \in \Pi(\mathcal{G})$ holds for all pairs of job permutations, where $(\pi_q, \pi_q) \in S(T)$.*

In [55], it is shown how to determine a minimal dominant set $S(T) = \{(\pi_q, \pi_q)\}$ with $\pi_q \in \Pi(\mathcal{G})$. The digraph $\mathcal{G} = (\mathcal{J}, \mathcal{A}_<)$ is considered as a condense form of a J-solution for the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$. The above results are used in Sections 4–6 for reducing a size of the dominant set $S(T)$ for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$.

4. The Off-Line Phase of Scheduling

The above setting of the uncertain job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ implies the following remark.

Remark 3. The factual value p_{ij}^* of the job processing time p_{ij} becomes known at the time-point $c_j(i)$ when the operation O_{ij} is completed on the machine $M_j \in \mathcal{M}$.

Due to Remark 3, if all jobs \mathcal{J} are completed on the corresponding machines from the set \mathcal{M} , the durations of all operations O_{ij} take on exact values p_{ij}^* , where $l_{ij} \leq p_{ij}^* \leq u_{ij}$, and a unique factual scenario $p^* \in T$ is realized. A pair of job permutations selected for this realization should be optimal for scenario p^* . For constructing such an optimal pair of job permutations, we propose to implement two phases, namely: the off-line phase of scheduling and the on-line phase of scheduling.

The off-line phase is completed before starting a realization of the selected semi-active schedule. At the off-line phase, a scheduler knows the exact lower and upper bounds on the job processing times and the aim is to determine a minimal dominant set of the pairs of job permutations (π', π'') .

The on-line phase is started when the corresponding machine starts the processing of the first job in the selected schedule. At this phase, a scheduler can use an additional information on the job processing time, since for each operation O_{ij} , the exact value p_{ij}^* of the processing time $p_{ij} \in T$ becomes known at the completion time $c_j(i)$ of this operation; see Remark 3.

We next consider the off-line phase of scheduling for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ and describe the sufficient conditions for existing a small dominant set of the semi-active schedules. Along with Definition 1, the following one is also used.

Definition 3. A set of the pairs of job permutations $DS(T) \subseteq S$ is a dominant set for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$, if for each scenario $p \in T$ the set $DS(T)$ contains at least one optimal pair of job permutations for the individual deterministic job-shop problem $J2|p, n_i \leq 2|C_{max}$ with scenario p .

Obviously, the J-solution is a dominant set for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$. Before processing the set \mathcal{J} of given jobs, a scheduler does not know the exact values of the job processing times. Nevertheless, it is needed to determine an optimal pair of permutations of the jobs \mathcal{J} for their processing on the machines $\mathcal{M} = \{M_1, M_2\}$.

In Section 4.1, the sufficient conditions are presented for existing a pair of job permutations (π', π'') such that the equality $DS(T) = \{(\pi', \pi'')\}$ holds. Section 4.2 contains the sufficient conditions allowing a scheduler to construct a semi-active schedule (if any), which dominates all other schedules in the set S . If a singleton $DS(T) = \{(\pi', \pi'')\}$ does not exist, a scheduler should construct partial strict orders $A_{\prec}^{1,2}$ and $A_{\prec}^{2,1}$ over set $\mathcal{J}_{1,2}$ and set $\mathcal{J}_{2,1}$; see Section 3.

4.1. Conditions for Existing a Single Optimal Pair of Job Permutations

The following conditions for existing an optimal pair of job permutations are proven in [8].

Theorem 4 ([8]). If one of the following conditions either (5) or (6) holds:

$$\sum_{J_i \in \mathcal{J}_{1,2}} u_{i1} \leq \sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_2} l_{i2} \text{ and } \sum_{J_i \in \mathcal{J}_{1,2}} l_{i2} \geq \sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_1} u_{i1}, \quad (5)$$

$$\sum_{J_i \in \mathcal{J}_{2,1}} u_{i2} \leq \sum_{J_i \in \mathcal{J}_{1,2} \cup \mathcal{J}_1} l_{i1} \text{ and } \sum_{J_i \in \mathcal{J}_{2,1}} l_{i1} \geq \sum_{J_i \in \mathcal{J}_{1,2} \cup \mathcal{J}_2} u_{i2}, \quad (6)$$

then any pair of permutations $(\pi', \pi'') \in S$ is a singleton $DS(T) = \{(\pi', \pi'')\}$ for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$.

Corollary 2 ([8]). If the following inequality holds:

$$\sum_{J_j \in \mathcal{J}_{1,2}} u_{i1} \leq \sum_{J_j \in \mathcal{J}_{2,1} \cup \mathcal{J}_2} l_{i2}, \quad (7)$$

then the set $\langle \{\pi_{1,2}\}, S_{2,1} \rangle \subseteq S$, where $\pi_{1,2}$ is an arbitrary permutation in the set $S_{1,2}$, is a dominant set for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$.

Corollary 3 ([8]). If the following inequality holds: $\sum_{j \in \mathcal{J}_{2,1}} u_{i2} \leq \sum_{j \in \mathcal{J}_{1,2} \cup \mathcal{J}_1} l_{i1}$, then the set $\langle S_{1,2}, \{\pi_{2,1}\} \rangle$, where $\pi_{2,1}$ is an arbitrary permutation in the set $S_{2,1}$, is a dominant set for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$.

In order to determine an optimal permutation for processing jobs from the set $\mathcal{J}_{2,1}$ (set $\mathcal{J}_{1,2}$, respectively), we consider the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J}_{1,2} \subseteq \mathcal{J}$ and the machine route (M_1, M_2) , and that with the job set $\mathcal{J}_{2,1} \subseteq \mathcal{J}$ and the machine route (M_2, M_1) . The following theorem has been proven in [8].

Theorem 5 ([8]). Let the set $S'_{1,2} \subseteq S_{1,2}$ be a set of permutations from the dominant set for the flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J}_{1,2}$, and the set $S'_{2,1} \subseteq S_{2,1}$ be a set of permutations from the dominant set for the flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J}_{2,1}$. Then the set $\langle S'_{1,2}, S'_{2,1} \rangle \subseteq S$ is a dominant set for the job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$.

4.2. Precedence Digraphs Determining a Minimal Dominant Set of Schedules

Based on Remark 1, the off-line phase of scheduling for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ may be based on solving the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J}_{1,2}$ and that with the job set $\mathcal{J}_{2,1}$. A criterion for the existence of a single-element J-solution for the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ is determined in Theorem 2.

In what follows, it is assumed that the equality $\mathcal{J}_{1,2} = \mathcal{J}_{1,2}^1 \cup \mathcal{J}_{1,2}^2 \cup \mathcal{J}_{1,2}^*$ holds, i.e., $\mathcal{J}_{1,2}^0 = \emptyset$. Using the results presented in Section 3, one can determine a binary relation $A_{\prec}^{1,2}$ for the uncertain flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J}_{1,2}$. For the job set $\mathcal{J}_{1,2}$, the binary relation $A_{\prec}^{1,2}$ determines the digraph $G_{1,2} = (\mathcal{J}_{1,2}, A_{\prec}^{1,2})$ with the vertex set $\mathcal{J}_{1,2}$ and the arc set $A_{\prec}^{1,2}$.

Definition 4. Two jobs $J_x \in \mathcal{J}_{1,2}$ and $J_y \in \mathcal{J}_{1,2}$, $x \neq y$, are conflict if they are not in the relation $A_{\prec}^{1,2}$, i.e., $(J_x, J_y) \notin A_{\prec}^{1,2}$ and $(J_y, J_x) \notin A_{\prec}^{1,2}$.

Due to Definition 2, for the conflict jobs $J_x \in \mathcal{J}_{1,2}$ and $J_y \in \mathcal{J}_{1,2}$, $x \neq y$, relations (3) and (4) do not hold for the case $v = x$ with $w = y$, nor for the case $v = y$ with $w = x$.

Definition 5. The inclusion-minimal set $\mathcal{J}_x \subseteq \mathcal{J}_{1,2}$ of the jobs is called a conflict set of the jobs, if for any job $J_y \in \mathcal{J}_{1,2} \setminus \mathcal{J}_x$ either relation $(J_x, J_y) \in A_{\prec}^{1,2}$ or relation $(J_y, J_x) \in A_{\prec}^{1,2}$ holds for each job $J_x \in \mathcal{J}_x$.

There may exist several conflict sets in the set $\mathcal{J}_{1,2}$. Let the strict order $A_{\prec}^{1,2}$ for the flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J}_{1,2}$ be represented as follows:

$$J_1 \prec J_2 \prec \dots \prec J_k \prec \{J_{k+1}, J_{k+2}, \dots, J_{k+r}\} \prec J_{k+r+1} \prec J_{k+r+2} \prec \dots \prec J_{m_{1,2}}. \quad (8)$$

Here, an optimal permutation for processing jobs from the set $\{J_1, J_2, \dots, J_k\}$ (for jobs from the set $\{J_{k+r+1}, J_{k+r+2}, \dots, J_{m_{1,2}}\}$) is as follows: (J_1, J_2, \dots, J_k) ($(J_{k+r+1}, J_{k+r+2}, \dots, J_{m_{1,2}})$, respectively). All jobs between braces in the presentation (8) constitute the conflict set of the jobs and they are in relation $A_{\prec}^{1,2}$ with any job located outside the braces. Due to Theorem 3, the set $\Pi(G_{1,2})$ of the permutations generated by the digraph $G_{1,2}$ includes an optimal permutation for each vector $p_{1,2}$ of the processing times of the jobs $\mathcal{J}_{1,2}$. Due to Corollary 1, the set $S_{1,2}(T) = \{(\pi_{1,2}, \pi_{1,2})\}$ with $\pi_{1,2} \in \Pi(G_{1,2})$ is a J-solution for the flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J}_{1,2}$. Analogously, the set $S_{2,1}(T) = \{(\pi_{2,1}, \pi_{2,1})\}$ with $\pi_{2,1} \in \Pi(G_{2,1})$ is a J-solution for the problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$.

with the job set $\mathcal{J}_{2,1}$. Due to Theorem 5, one can determine a dominant set for the job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set \mathcal{J} as follows: $\langle \Pi(G_{1,2}), \Pi(G_{2,1}) \rangle \subseteq S$; see Remark 1.

The following three theorems are proven in [8], where the notation $L_2 := \sum_{J_j \in \mathcal{J}_{2,1} \cup \mathcal{J}_2} l_{j2}$ is used. These theorems allow a scheduler to reduce the cardinality of a dominant set for the uncertain job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$.

Theorem 6 ([8]). Let the strict order $A_{\prec}^{1,2}$ over the set $\mathcal{J}_{1,2} = \mathcal{J}_{1,2}^* \cup \mathcal{J}_{1,2}^1 \cup \mathcal{J}_{1,2}^2$ be determined as follows: $J_1 \prec \dots \prec J_k \prec \{J_{k+1}, J_{k+2}, \dots, J_{k+r}\} \prec J_{k+r+1} \prec \dots \prec J_{m_{1,2}}$. If the following inequality holds:

$$\sum_{i=1}^{k+r} u_{i1} \leq L_2 + \sum_{i=1}^k l_{i2}, \quad (9)$$

then the set $S' = \langle \{\pi_{1,2}\}, \Pi(G_{2,1}) \rangle \subset S$, where $\pi_{1,2} \in \Pi(G_{1,2})$, is a dominant set for the job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$.

Theorem 7 ([8]). Let the partial strict order $A_{\prec}^{1,2}$ over the set $\mathcal{J}_{1,2} = \mathcal{J}_{1,2}^* \cup \mathcal{J}_{1,2}^1 \cup \mathcal{J}_{1,2}^2$ be determined as follows: $J_1 \prec \dots \prec J_k \prec \{J_{k+1}, J_{k+2}, \dots, J_{k+r}\} \prec J_{k+r+1} \prec \dots \prec J_{m_{1,2}}$. If the inequality

$$u_{k+s,1} \leq L_2 + \sum_{i=1}^{k+s-1} (l_{i2} - u_{i1}) \quad (10)$$

holds for each $s \in \{1, 2, \dots, r\}$, then the set $S' = \langle \{\pi_{1,2}\}, S_{2,1} \rangle$, where $\pi_{1,2} = (J_1, \dots, J_{k-1}, J_k, J_{k+1}, J_{k+2}, \dots, J_{k+r}, J_{k+r+1}, \dots, J_{m_{1,2}}) \in \Pi(G_{1,2})$, is a dominant set for the job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$.

Theorem 8 ([8]). Let the partial strict order $A_{\prec}^{1,2}$ over the set $\mathcal{J}_{1,2} = \mathcal{J}_{1,2}^* \cup \mathcal{J}_{1,2}^1 \cup \mathcal{J}_{1,2}^2$ have the form $J_1 \prec \dots \prec J_k \prec \{J_{k+1}, J_{k+2}, \dots, J_{k+r}\} \prec J_{k+r+1} \prec \dots \prec J_{m_{1,2}}$. If the inequality

$$\sum_{i=r-s+2}^{r-s+1} l_{k+i,1} \geq \sum_{j=r-s+1}^r u_{k+j,2} \quad (11)$$

holds for each $s \in \{1, 2, \dots, r\}$, then the set $S' = \langle \{\pi_{1,2}\}, S_{2,1} \rangle$, where $\pi_{1,2} = (J_1, \dots, J_{k-1}, J_k, J_{k+1}, J_{k+2}, \dots, J_{k+r}, J_{k+r+1}, \dots, J_{m_{1,2}}) \in \Pi(G_{1,2})$, is a dominant set for the job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$.

One can describe the analogs of Theorems 6–8 for reducing the cardinality of a dominant set for the job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ provided that for the flow-shop problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with the job set $\mathcal{J}_{2,1}$, there exists a partial strict order $A_{\prec}^{2,1}$ over the set $\mathcal{J}_{2,1} = \mathcal{J}_{2,1}^* \cup \mathcal{J}_{2,1}^1 \cup \mathcal{J}_{2,1}^2$ with the following form: $J_1 \prec \dots \prec J_k \prec \{J_{k+1}, J_{k+2}, \dots, J_{k+r}\} \prec J_{k+r+1} \prec \dots \prec J_{m_{2,1}}$.

If the set $\{J_1, \dots, J_k\}$ is empty in the constructed job permutation, then it is needed to check the conditions of Theorem 8. If the set $\{J_{k+r+1}, \dots, J_{m_{1,2}}\}$ is empty, then one needs to check the conditions of Theorem 7. Note that it is enough to test only one permutation for checking the conditions of Theorem 7 and only one permutation for checking the conditions of Theorem 8; see [8].

4.3. An Illustrative Example

To illustrate the above results, we consider Example 1 of the uncertain job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with eight jobs $\{J_1, J_2, \dots, J_8\} = \mathcal{J}$. Let three jobs J_1, J_2 and J_3 have the machine route (M_1, M_2) , jobs J_6, J_7 and J_8 have the opposite machine route (M_2, M_1) , job J_4 and job J_5 have to be processed only on machine M_1 and machine M_2 , respectively. The partition $\mathcal{J} =$

$\mathcal{J}_{1,2} \cup \mathcal{J}_{2,1} \cup \mathcal{J}_1 \cup \mathcal{J}_2$ is given, where $\mathcal{J}_{1,2} = \{J_1, J_2, J_3\}$, $\mathcal{J}_{2,1} = \{J_6, J_7, J_8\}$, $\mathcal{J}_1 = \{J_4\}$ and $\mathcal{J}_2 = \{J_5\}$. The lower and upper bounds on the job processing times are determined in Table 1.

Table 1. The numerical input data for Example 1.

J_j	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
l_{i1}	6	8	7	2	-	1	1	1
u_{i1}	7	9	9	3	-	3	3	3
l_{i2}	6	5	4	-	2	2	3	4
u_{i2}	7	6	5	-	3	4	4	4

To solve this uncertain job-shop scheduling problem, one need to determine an optimal pair (π', π'') of permutations of the eight jobs for their processing on machine M_1 and machine M_2 . These permutations π' and π'' have the following forms: $\pi' = (\pi_{1,2}, \pi_1, \pi_{2,1})$, $\pi'' = (\pi_{2,1}, \pi_2, \pi_{1,2})$.

It is necessary to find four permutations $\pi_{1,2}$, $\pi_{2,1}$, π_1 and π_2 of the jobs from the sets $\mathcal{J}_{1,2}$, $\mathcal{J}_{2,1}$, \mathcal{J}_1 and \mathcal{J}_2 , respectively. The permutations π_1 and π_2 are determined as follows: $\pi_1 = (J_4)$ and $\pi_2 = (J_5)$.

We test the sufficient conditions given in Section 4.1. The conditions (5) of Theorem 4 do not hold. For testing the conditions (6) of Theorem 4, one can obtain the following relations:

$$\sum_{J_i \in \mathcal{J}_{2,1}} u_{i2} = u_{6,2} + u_{7,2} + u_{8,2} = 4 + 4 + 4 = 12 \leq \sum_{J_i \in \mathcal{J}_{1,2} \cup \mathcal{J}_1} l_{i1} = l_{1,1} + l_{2,1} + l_{3,1} + l_{4,1} = 6 + 8 + 7 + 2 = 23,$$

$$\sum_{J_i \in \mathcal{J}_{2,1}} l_{i1} = l_{6,1} + l_{7,1} + l_{8,1} = 1 + 1 + 1 = 3 \not\geq \sum_{J_i \in \mathcal{J}_{1,2} \cup \mathcal{J}_2} u_{i2} = u_{1,2} + u_{2,2} + u_{3,2} + u_{5,2} = 7 + 6 + 5 + 3 = 21.$$

It should be noted that the case when conditions of Theorem 4 hold was considered in [8].

As the first condition in (6) holds, due to Corollary 3, one can construct permutation $\pi_{2,1} = (J_6, J_7, J_8)$ by arranging the jobs from the set $\mathcal{J}_{2,1}$ in the increasing of their indexes.

For the jobs from the set $\mathcal{J}_{1,2}$, the partition $\mathcal{J}_{1,2} = \mathcal{J}_{1,2}^1 \cup \mathcal{J}_{1,2}^2 \cup \mathcal{J}_{1,2}^*$ holds, where $\mathcal{J}_{1,2}^* = \{J_1\}$ and $\mathcal{J}_{1,2}^2 = \{J_2, J_3\}$. The condition of Theorem 2 holds for these jobs. Therefore, the following optimal permutation: $\pi_{1,2} = (J_1, J_2, J_3)$ is determined.

Thus, there exists a pair of job permutations (π', π'') , where $\pi' = (J_1, J_2, J_3, J_4, J_6, J_7, J_8)$ and $\pi'' = (J_6, J_7, J_8, J_5, J_1, J_2, J_3)$, which is optimal for all possible scenarios $p \in T$. Hence, there exists a single-element dominant set $DS(T) = \{(\pi', \pi'')\}$ for Example 1 of the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the bounds on the job processing times given in Table 1.

The optimal semi-active schedule is constructed for Example 1 at the off-line phase of scheduling, despite of the uncertainty of the job processing times. Such an issue is called as STOP 1 in the scheduling algorithms developed in [8] and used in Section 6 of this paper.

5. The On-line Phase of Scheduling

Due to Remark 3, if the job J_i is completed on the corresponding machine $M_j \in \mathcal{M}$, the duration of the operation O_{ij} takes on exact value p_{ij}^* , where $l_{ij} \leq p_{ij}^* \leq u_{ij}$. A scheduler can use this information on the duration of the operation O_{ij} for a selection of the next job for processing on machine M_j . Since it is on-line phase of scheduling, such a selection should be very quick.

It is first assumed that the set $S' = \langle \Pi(G_{1,2}), \{\pi_{2,1}^*\} \rangle \subset S$, is a dominant set for the problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set \mathcal{J} . In other words, the optimal permutations for processing all jobs from the set $\mathcal{J}_{2,1}$ are already determined at the off-line phase of scheduling.

Let the strict order $A_{\prec}^{1,2}$ over the set $\mathcal{J}_{1,2} = \mathcal{J}_{1,2}^* \cup \mathcal{J}_{1,2}^1 \cup \mathcal{J}_{1,2}^2$ be determined as follows: $J_1 \prec \dots \prec J_k \prec \{J_{k+1}, J_{k+2}, \dots, J_{k+r}\} \prec J_{k+r+1} \prec \dots \prec J_{m_{1,2}}$. At the initial time $t = 0$, machine M_1 has to start processing jobs from the set $\{J_1, \dots, J_k\}$ in the following optimal order: (J_1, \dots, J_k) . At the same time $t = 0$, machine M_2 has to start processing jobs from the set $\mathcal{J}_{2,1}$ in the order

determined by the permutation $\pi_{2,1}^*$, then jobs from the set \mathcal{J}_2 in the arbitrary order, and then jobs from the set $\{J_1, \dots, J_k\}$ in the following optimal order: (J_1, \dots, J_k) ; see Figure 3.

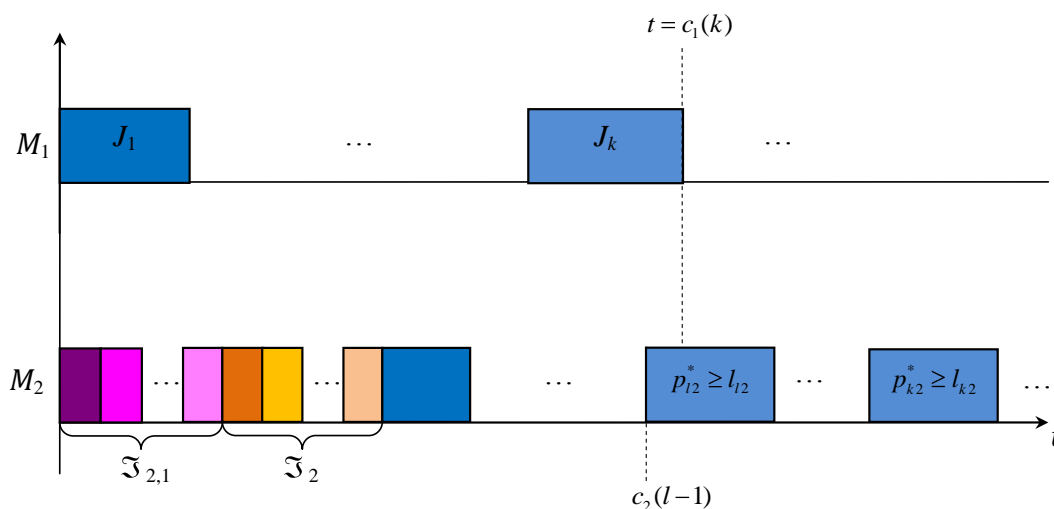


Figure 3. The initial part of the schedule execution.

At the time-point $t = c_1(k)$, machine M_1 completes the operation O_{k1} . Let $\mathcal{J}(i, j)$ denote a set of all jobs processed on machine M_j from the initial part of the schedule till the job J_i , e.g., the set of jobs $\{J_1, J_2, \dots, J_k\}$ is denoted as $\mathcal{J}(k, 1)$; see Figure 3. Due to Remark 3, at the time-point $t = c_1(k)$, the factual values p_{i1}^* of the processing times p_{i1} of all jobs J_i in the set $\mathcal{J}(k, 1)$ are already known.

Let machine M_2 process the job $J_l \in \mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \{J_1, J_2, \dots, J_k\}$ at the time-point $t = c_1(k)$, i.e., $t = c_1(k) < c_2(l)$. Let $\mathcal{J}(l-1, 2)$ denote a set of all jobs whose processing is completed on machine M_2 before time-point $t = c_1(k)$. Figure 3 depicts this situation for the job $J_{l-1} \in \{J_1, J_2, \dots, J_k\} \subset \mathcal{J}_{1,2}$.

The factual values p_{i2}^* of the processing times p_{i2} of all jobs J_i in the set $\mathcal{J}(l-1, 2)$ are known at the time-point $t = c_1(k) > c_2(l-1)$, i.e., $p_{i2} = p_{i2}^*$, while the factual values of the processing times p_{j2} of other jobs in the set \mathcal{J} remain unknown at the time-point $t = c_1(k) < c_2(l)$. Thus, at the time-point $t = c_1(k)$, the following subset of possible scenarios:

$$T(k, l-1) = \{p \in T : p_{i1} = p_{i1}^*, p_{j2} = p_{j2}^*, J_i \in \mathcal{J}(k, 1), J_j \in \mathcal{J}(l-1, 2)\}$$

may be realized instead of the initial set T of all possible scenarios; $T(k, l-1) \subseteq T$.

At the time-point $t = c_1(k)$ (it is called a decision-point), a scheduler has to make a decision about the order for processing jobs from the conflict set $\{J_{k+1}, J_{k+2}, \dots, J_{k+r}\}$. The sufficient conditions given in Theorems 6 and 7 can be reformulated in the following two theorems. (Note that Theorem 8 cannot be reformulated for the use at the on-line phase of scheduling.)

Theorem 9. Let the set $S' = \langle \Pi(G_{1,2}), \{\pi_{2,1}^*\} \rangle \subset S$ be a dominant set for the uncertain problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set \mathcal{J} . Let the strict order $A_{\mathcal{J}}^{1,2}$ over the set $\mathcal{J}_{1,2} = \mathcal{J}_{1,2}^* \cup \mathcal{J}_{1,2}^1 \cup \mathcal{J}_{1,2}^2$ be determined as follows: $J_1 \prec \dots \prec J_k \prec \{J_{k+1}, J_{k+2}, \dots, J_{k+r}\} \prec J_{k+r+1} \prec \dots \prec J_{m_{1,2}}$. If at the time-point $t = c_1(k)$, the following inequality holds:

$$c_1(k) + \sum_{i=k+1}^{k+r} u_{i1} \leq c_2(l-1) + \sum_{J_i \in (\mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}(k,1)) \setminus \mathcal{J}(l-1,2)} l_{i2}, \quad (12)$$

then at the time-point $t = c_1(k)$, the set $S' = \langle \{\pi_{1,2}\}, \{\pi_{2,1}^*\} \rangle \subset S$, where $\pi_{1,2} \in \Pi(G_{1,2})$, is a dominant set for the problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set \mathcal{J} and the set $T(k, l-1)$ of possible scenarios.

Proof. Let p be an arbitrary vector from the set $T(k, l-1)$ of possible scenarios at the time-point $t = c_1(k)$. Let C_{\max}^p denote the optimal makespan value for the deterministic job-shop problem $J2|p, n_i \leq 2|C_{\max}$ with the set \mathcal{J} of the given jobs and the vector p of the job processing times.

We consider an arbitrary permutation $\pi_{1,2} \in \Pi(G_{1,2})$ and show that the pair of job permutations $(\pi', \pi'') = ((\pi_{1,2}, \pi_1, \pi_{2,1}^*), (\pi_{2,1}^*, \pi_2, \pi_{1,2})) \in S'$ is an optimal one for the deterministic job-shop problem $J2|p, n_i \leq 2|C_{\max}$ with the set \mathcal{J} of the jobs and with any vector $p \in T(k, l-1)$ of the job processing times, i.e., the equality $C_{\max}(\pi', \pi'') = C_{\max}^p$ holds. Since the equality $C_{\max}(\pi', \pi'') = \max\{c_1(\pi'), c_2(\pi'')\}$ holds, one has to consider two possible cases (a) and (b).

Case (a): It is assumed that $c_1(\pi') \geq c_2(\pi'')$. Then, one can obtain the following equalities:

$$C_{\max}(\pi', \pi'') = c_1(\pi') = \max\left\{\sum_{J_i \in \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1} \cup \mathcal{J}_1} p_{i1}, C_{\max}(\pi_{2,1}^*)\right\}, \quad (13)$$

where $C_{\max}(\pi_{2,1}^*)$ is the value of makespan for the deterministic flow-shop problem $F2|p_{2,1}|C_{\max}$ with the job set $\mathcal{J}_{2,1}$ and the vector $p_{2,1}$ whose components are equal to the corresponding components of the vector p . Due to the conditions of Theorem 9, the permutation $\pi_{2,1}^*$ is optimal for the deterministic flow-shop problem $F2|p_{2,1}|C_{\max}$ with the set $\mathcal{J}_{2,1}$ of the given jobs and with vector $p_{2,1}$ of the job processing times. Therefore, $C_{\max}(\pi_{2,1}^*)$ is an optimal makespan value for the deterministic flow-shop problem $F2|p_{2,1}|C_{\max}$ and $C_{\max}(\pi_{2,1}^*)$ is a minimal completion time for processing all jobs from the set $\mathcal{J}_{2,1}$ on both machines. From the equalities (13), one can obtain the equality $C_{\max}(\pi', \pi'') = C_{\max}^p$.

Case (b): It is assumed that $c_1(\pi') < c_2(\pi'')$. Then, one can obtain the following equalities:

$$C_{\max}(\pi', \pi'') = c_2(\pi'') = \max\left\{\sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2}} p_{i2}, C_{\max}(\pi_{1,2})\right\}, \quad (14)$$

where $C_{\max}(\pi_{1,2})$ is an optimal value of the makespan criterion for the deterministic flow-shop problem $F2|p_{1,2}|C_{\max}$ with the job set $\mathcal{J}_{1,2}$ and with the vector $p_{1,2}$ of the job processing times (the components of this vector are equal to the corresponding components of the vector p). Since $\pi_{1,2} \in \Pi(G_{1,2})$, the initial part of the permutation $\pi_{1,2}$ has the following form: (J_1, J_2, \dots, J_k) . For every pair of jobs from the set $\{J_1, J_2, \dots, J_k\}$, at least one of the conditions, either (3) or (4), holds, see Theorem 1.

Therefore, for the job processing times determined by the vector p for the jobs $\{J_1, J_2, \dots, J_k\}$, the inequalities (2) hold. Thus, in the permutation $\pi_{1,2}^{beg} := (J_1, J_2, \dots, J_k)$, all the jobs are arranged in the Johnson's order. One can conclude that the following value

$$C_{\max}(\pi_{1,2}^{beg}) = \max_{1 \leq m \leq k} \left\{ \sum_{i=1}^m p_{i1} + \sum_{i=m}^k p_{i2} \right\} \quad (15)$$

determines an optimal makespan value for the deterministic flow-shop problem $F2|p_{1,2}^{beg}|C_{\max}$ with the job set $\{J_1, J_2, \dots, J_k\}$ and the corresponding vector $p_{1,2}^{beg}$ of the job processing times (the components of the vector $p_{1,2}^{beg}$ are equal to the corresponding components of the vector p). Therefore, $C_{\max}(\pi_{1,2}^{beg})$ is a minimal makespan value for processing jobs of the set $\{J_1, J_2, \dots, J_k\}$ on both machines. Then, for the time-point $c_2(k)$ when machine M_2 completes the operation O_{k2} , one can obtain the following equality:

$$c_2(k) = \max\left\{\sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2}(k,1)} p_{i2}, C_{\max}(\pi_{1,2}^{beg})\right\}. \quad (16)$$

Due to the inequality (12) and the equality (16), one can obtain the following inequalities for the jobs from the conflict set $\{J_{k+1}, J_{k+2}, \dots, J_{k+r}\}$:

$$c_1(k) + \sum_{i=k+1}^{k+r} p_{i1} \leq c_1(k) + \sum_{i=k+1}^{k+r} u_{i1} \leq c_2(l-1) + \sum_{J_i \in (\mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}(k,1)) \setminus \mathcal{J}(l-1,2)} l_{i2} \leq \quad (17)$$

$$\leq c_2(l-1) + \sum_{J_i \in (\mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}(k,1)) \setminus \mathcal{J}(l-1,2)} p_{i2} \leq \max \left\{ \sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2}(k,1)} p_{i2}, C_{\max}(\pi_{1,2}^{beg}) \right\} = c_2(k).$$

From the inequalities (17), one can obtain the following inequality:

$$c_1(k) + \sum_{i=k+1}^{k+r} p_{i1} \leq c_2(k). \quad (18)$$

Thus, machine M_2 processes all jobs from the conflict set $\{J_{k+1}, J_{k+2}, \dots, J_{k+r}\}$ without idle times and without an idle before processing the first job from this conflict set for any order of these conflict jobs. Using the inequality (18), one can conclude that the time-point when machine M_2 completes the processing of the last job from the conflict set $\{J_{k+1}, J_{k+2}, \dots, J_{k+r}\}$ in the permutation $\pi_{1,2}$ is determined as follows:

$$c_2 = c_2(k) + \sum_{i=k+1}^{k+r} p_{i2}, \quad (19)$$

where c_2 is an optimal makespan value for processing jobs from the set $\{J_1, J_2, \dots, J_k, J_{k+1}, J_{k+2}, \dots, J_{k+r}\}$. Next, we consider jobs from the set $\{J_{k+r+1}, \dots, J_{m_{1,2}}\}$.

Let $\pi_{1,2}^{end} := (J_{k+r+1}, \dots, J_{m_{1,2}})$ denote the permutation of the jobs $\{J_{k+r+1}, \dots, J_{m_{1,2}}\}$ in the permutation $\pi_{1,2}$. Analogously as for the job set $\{J_1, J_2, \dots, J_k\}$, one can obtain that the value of

$$C_{\max}(\pi_{1,2}^{end}) := \max_{k+r+1 \leq m \leq m_{1,2}} \left\{ \sum_{i=k+r+1}^m p_{i1} + \sum_{i=m}^{m_{1,2}} p_{i2} \right\} \quad (20)$$

is an optimal makespan value for the deterministic flow-shop problem $F2|p_{1,2}^{end}|C_{\max}$ with the job set $\{J_{k+r+1}, \dots, J_{m_{1,2}}\}$ and with the vector $p_{1,2}^{end}$ whose components are equal to the components of the vector p . Thus, $C_{\max}(\pi_{1,2}^{end})$ is a minimal makespan value for processing all jobs from the set $\{J_{k+r+1}, \dots, J_{m_{1,2}}\}$ on both machines. The time-point when machine M_2 completes the processing of the last job from the permutation π'' can be calculated as follows:

$$\begin{aligned} c_2(\pi'') &= \max \left\{ \sum_{i=1}^{k+r} p_{i1} + C_{\max}(\pi_{1,2}^{end}), c_2 + \sum_{i=k+r+1}^{m_{1,2}} p_{i2} \right\} = \\ &= \max \left\{ \sum_{i=1}^{k+r} p_{i1} + C_{\max}(\pi_{1,2}^{end}), c_2(k) + \sum_{i=k+1}^{k+r} p_{i2} + \sum_{i=k+r+1}^{m_{1,2}} p_{i2} \right\} = \\ &= \max \left\{ \sum_{i=1}^{k+r} p_{i1} + C_{\max}(\pi_{1,2}^{end}), \max \left\{ \sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2}(k,1)} p_{i2}, C_{\max}(\pi_{1,2}^{beg}) \right\} + \sum_{i=k+1}^{m_{1,2}} p_{i2} \right\} = \\ &= \max \left\{ \sum_{i=1}^{k+r} p_{i1} + C_{\max}(\pi_{1,2}^{end}), \sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2}(k,1)} p_{i2} + \sum_{i=k+1}^{m_{1,2}} p_{i2}, C_{\max}(\pi_{1,2}^{beg}) + \sum_{i=k+1}^{m_{1,2}} p_{i2} \right\} = \\ &= \max \left\{ C_{\max}(\pi_{1,2}^{beg}) + \sum_{i=k+1}^{m_{1,2}} p_{i2}, \sum_{i=1}^{k+r} p_{i1} + C_{\max}(\pi_{1,2}^{end}), \sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2}} p_{i2} \right\}, \quad (21) \end{aligned}$$

where relations (16) and (19) are used.

Due to Theorem 3, the set $\Pi(G_{1,2})$ contains a Johnson's permutation for the deterministic flow-shop problem $F2|p_{1,2}|C_{\max}$ with the job set $\mathcal{J}_{1,2}$ and with the vector $p_{1,2}$ of the job durations. We denote this Johnson's permutation as $\pi_{1,2}^*$. Since $\pi_{1,2}^* \in \Pi(G_{1,2})$, the permutation $\pi_{1,2}^*$ has the following form: $\pi_{1,2}^* = (J_1, \dots, J_k, J_{[k+1]}, J_{[k+2]}, \dots, J_{[k+r]}, J_{k+r+1}, \dots, J_{m_{1,2}})$, where the set of indexes is determined as follows: $\{[k+1], [k+2], \dots, [k+r]\} = \{k+1, k+2, \dots, k+r\}$.

The optimal makespan value $C_{\max}(\pi_{1,2}^*)$ can be calculated as follows:

$$\begin{aligned} C_{\max}(\pi_{1,2}^*) &= \max_{1 \leq m \leq m_{1,2}} \left\{ \sum_{i=1}^m p_{i1} + \sum_{i=m}^{m_{1,2}} p_{i2} \right\} = \max \left\{ \max_{1 \leq m \leq k} \left\{ \sum_{i=1}^m p_{i1} + \sum_{i=m}^k p_{i2} + \sum_{i=k+1}^{m_{1,2}} p_{i2} \right\}, \right. \\ &\quad \left. \max_{[k+1] \leq m \leq [k+r]} \left\{ \sum_{i=1}^m p_{i1} + \sum_{i=m}^{m_{1,2}} p_{i2} \right\}, \max_{k+r+1 \leq m \leq m_{1,2}} \left\{ \sum_{i=1}^{k+r} p_{i1} + \sum_{i=k+r+1}^m p_{i1} + \sum_{i=m}^{m_{1,2}} p_{i2} \right\} \right\} = \\ &= \max \left\{ C_{\max}(\pi_{1,2}^{beg}) + \sum_{i=k+1}^{m_{1,2}} p_{i2}, \max_{[k+1] \leq m \leq [k+r]} \left\{ \sum_{i=1}^m p_{i1} + \sum_{i=m}^{m_{1,2}} p_{i2} \right\}, \sum_{i=1}^{k+r} p_{i1} + C_{\max}(\pi_{1,2}^{end}) \right\}, \quad (22) \end{aligned}$$

where relations (15) and (20) are used. From relations (21) and (22), one can obtain the relations

$$\begin{aligned} c_2(\pi'') &= \max \left\{ C_{\max}(\pi_{1,2}^{beg}) + \sum_{i=k+1}^{m_{1,2}} p_{i2}, \sum_{i=1}^{k+r} p_{i1} + C_{\max}(\pi_{1,2}^{end}), \sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2}} p_{i2} \right\} \leq \\ &\leq \max \left\{ C_{\max}(\pi_{1,2}^*), \sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2}} p_{i2} \right\}. \quad (23) \end{aligned}$$

Therefore, relations (14) and (23) imply the equality $C_{\max}(\pi', \pi'') = C_{\max}^p$.

Thus, in both cases (a) and (b), the equality $C_{\max}(\pi', \pi'') = C_{\max}^p$ holds and the pair of permutations $(\pi', \pi'') = ((\pi_{1,2}, \pi_1, \pi_{2,1}^*), (\pi_{2,1}^*, \pi_2, \pi_{1,2}))$ is optimal for the deterministic job-shop problem $J2|p, n_i \leq 2|C_{\max}$ with the scenario $p \in T(k, l-1)$. Therefore, the set $S' = \langle \{\pi_{1,2}\}, \{\pi_{2,1}^*\} \rangle$ contains an optimal pair of job permutations for the job-shop problem $J2|p, n_i \leq 2|C_{\max}$ with vector $p \in T(k, l-1)$ of the job processing times. Since the vector p is arbitrarily chosen in the set $T(k, l-1)$, the set S' contains an optimal pair of job permutations for each scenario in the set $T(k, l-1)$.

Due to Definition 3, the set S' is a dominant set for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{\max}$ with the job set \mathcal{J} and with the set $T(k, l-1)$ of possible scenarios. \square

Theorem 10. Let the set $S' = \langle \Pi(G_{1,2}), \{\pi_{2,1}^*\} \rangle \subset S$ be a dominant set for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{\max}$ with the job set \mathcal{J} . Let the partial strict order $A_{\prec}^{1,2}$ over the set $\mathcal{J}_{1,2} = \mathcal{J}_{1,2}^* \cup \mathcal{J}_{1,2}^1 \cup \mathcal{J}_{1,2}^2$ be determined as follows: $J_1 \prec \dots \prec J_k \prec \{J_{k+1}, J_{k+2}, \dots, J_{k+r}\} \prec J_{k+r+1} \prec \dots \prec J_{m_{1,2}}$. If at the time-point $t = c_1(k)$, the following inequalities hold:

$$c_1(k) + \sum_{i=k+1}^{k+s} u_{i1} \leq c_2(l-1) + \sum_{J_i \in (\mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}(k-1,1)) \setminus \mathcal{J}(l-1,2)} l_{i2} + \sum_{i=k}^{k+s-1} l_{i2} \quad (24)$$

for all indexes $s \in \{1, 2, \dots, r\}$, then at the time-point $t = c_1(k)$, the set $S' = \langle \{\pi_{1,2}\}, \{\pi_{2,1}^*\} \rangle$, where $\pi_{1,2} = (J_1, \dots, J_{k-1}, J_k, J_{k+1}, J_{k+2}, \dots, J_{k+r}, J_{k+r+1}, \dots, J_{m_{1,2}}) \in \Pi(G_{1,2})$, is a dominant set for the uncertain problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{\max}$ with the job set \mathcal{J} and the set $T(k, l-1)$ of possible scenarios.

Proof. The proof of this theorem is similar to the above proof of Theorem 9 with the exception of the inequalities (17) and (18). From the condition (24) with $s = 1$, one can obtain the following inequality:

$$c_1(k) + u_{k+1,1} \leq c_2(l-1) + \sum_{J_i \in (\mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}(k-1,1)) \setminus \mathcal{J}(l-1,2)} l_{i2} + l_{k2}. \quad (25)$$

Based on the inequality (25), one can obtain the following relations:

$$c_1(k) + p_{k+1,1} \leq c_1(k) + u_{k+1,1} \leq c_2(l-1) + \sum_{J_i \in (\mathcal{J}_{2,1} \cup \mathcal{J}_2 \cup \mathcal{J}(k-1,1)) \setminus \mathcal{J}(l-1,2)} l_{i2} + l_{k2} \leq$$

$$\leq c_2(l-1) + \sum_{J_i \in (\mathcal{J}_{2,1} \cup \mathcal{J}_{2,2} \cup \mathcal{J}(k,1)) \setminus \mathcal{J}(l-1,2)} p_{i2} = c_2(k). \quad (26)$$

Due to relations (26), the following inequality holds:

$$c_1(k) + p_{k+1,1} \leq c_2(k). \quad (27)$$

Thus, machine M_2 processes the job J_{k+1} in permutation $\pi_{1,2}$ without an idle time between the jobs J_k and J_{k+1} . Analogously, using $s \in \{2, 3, \dots, r\}$, one can show that the following inequalities hold:

$$\begin{aligned} c_1(k) + p_{k+1,1} + p_{k+2,1} &\leq c_2(k+1); \\ c_1(k) + p_{k+1,1} + p_{k+2,1} + p_{k+3,1} &\leq c_2(k+2); \\ &\dots; \\ c_1(k) + \sum_{i=k+1}^{k+r} p_{i1} &\leq c_2(k+r-1). \end{aligned}$$

Therefore, machine M_2 processes jobs from the conflict set $\{J_{k+1}, J_{k+2}, \dots, J_{k+r}\}$ in permutation $\pi_{1,2}$ without idle times between the jobs J_{k+1} and J_{k+2} , between the jobs J_{k+2} and J_{k+3} and so on, between the jobs J_{k+r-1} and J_{k+r} . Then, the following relations hold:

$$c_2 = c_2(k+r) = c_2(k+r-1) + p_{k+r,2} = c_2(k+r-2) + p_{k+r-1,2} + p_{k+r,2} = \dots = c_2(k) + \sum_{i=k+1}^{k+r} p_{i2}$$

leading to the equality (19). The rest of the proof is the same as the rest of the proof of Theorem 9.

It is shown that the pair of job permutations $(\pi', \pi'') = ((\pi_{1,2}, \pi_1, \pi_{2,1}^*), (\pi_{2,1}^*, \pi_2, \pi_{1,2})) \in S'$ is optimal for the deterministic job-shop problem $J2|p, n_i \leq 2|C_{max}$ with any vector $p \in T(k, l-1)$ of the job processing times. Due to Definition 3, the set S' is a dominant set for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set \mathcal{J} and the set $T(k, l-1)$ of possible scenarios. \square

It is easy to be convinced that the sufficient conditions given in Theorems 9 and 10 may be tested in polynomial time $O(r^2)$ of the number r of the conflict jobs.

Similarly, one can prove analogs of Theorems 9 and 10 if the set $S' = \langle \{\pi_{1,2}^*\}, \Pi(G_{2,1}) \rangle \subset S$ provided that a dominant set for the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the job set \mathcal{J} and the partial strict order $A_{\prec}^{2,1}$ over the set $\mathcal{J}_{2,1} = \mathcal{J}_{2,1}^* \cup \mathcal{J}_{2,1}^1 \cup \mathcal{J}_{2,1}^2$ has the following form: $J_1 \prec \dots \prec J_k \prec \{J_{k+1}, J_{k+2}, \dots, J_{k+r}\} \prec J_{k+r+1} \prec \dots \prec J_{m_{2,1}}$.

6. Scheduling Algorithms and Computational Results

The experimental study was performed on a large number of randomly generated instances of the uncertain job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$. The off-line phase of scheduling was based on Algorithms 1 and 2 developed in [8]. Algorithms 1 and 2 are presented in Appendix A.

Algorithms 3–5 are developed for the on-line phase of scheduling. The input for each of these three algorithms includes the output of Algorithms 1 and 2 [8] applied at the off-line phase of scheduling.

Let outputs of Algorithms 1 and 2 [8] applied at the off-line phase of scheduling consist of the optimal permutation $\pi_{1,2}$ of the jobs $\mathcal{J}_{1,2}$ and the optimal permutation $\pi_{2,1}$ of the jobs $\mathcal{J}_{2,1}$. In such a case, the single-element dominant set $DS(T) = \{(\pi_{1,2}, \pi_1, \pi_{2,1}), (\pi_{2,1}, \pi_2, \pi_{1,2})\}$ is already constructed for the considered instance of the uncertain problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$. Therefore, the pair $\{(\pi_{1,2}, \pi_1, \pi_{2,1}), (\pi_{2,1}, \pi_2, \pi_{1,2})\}$ of the job permutations is optimal for the deterministic instance $J2|p, n_i \leq 2|C_{max}$ with any scenario $p \in T$. Thus, such an instance of the uncertain problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ is optimally solved by Algorithms 1 and 2 at the off-line phase of scheduling.

Hence, there is no need to use the on-line phase of scheduling for such an instance of the uncertain job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$.

In Section 6.1, it is shown how to solve instances of the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$, which cannot be optimally solved at the off-line phase of scheduling.

6.1. Algorithms 3–5 for the On-Line Phase of Scheduling

Let the considered instance of the uncertain job-shop problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ cannot be optimally solved by Algorithms 1 and 2 [8] applied at the off-line phase of scheduling. Thus, due to an application of Algorithm 1 or Algorithm 2, one can obtain one of the following three possible outputs:

- (a) the permutation $\pi_{2,1}$ of the jobs from set $\mathcal{J}_{2,1}$ and the partial strict order $A_{\prec}^{1,2}$ of the jobs $\mathcal{J}_{1,2}$;
- (b) the permutation $\pi_{1,2}$ of the jobs from set $\mathcal{J}_{1,2}$ and the partial strict order $A_{\prec}^{2,1}$ of the jobs $\mathcal{J}_{2,1}$;
- (c) the partial strict order $A_{\prec}^{1,2}$ of the jobs $\mathcal{J}_{1,2}$ and the partial strict order $A_{\prec}^{2,1}$ of the jobs $\mathcal{J}_{2,1}$.

Let B denote a number of the conflict sets in a partial strict order (in both partial strict orders) for the obtained output (a), (b) or (c). In other words, B denotes a maximal number of time-points in the decision-making at the on-line phase of scheduling. Let integer b , where $b \leq B$, denote a number of time-points in the decision-making, where optimal orders of the conflict jobs were found using Theorem 9 or Theorem 10. Using these notations, we next describe Algorithm 3 provided that there is no factual processing times of the jobs \mathcal{J} in the input of Algorithm 3; see Remark 3.

Let Algorithm 3 terminate at Step 16, i.e., it has not been constructed an optimal pair of job permutations for the factual scenario $p^* \in T$ randomly determined after completing the on-line phase of scheduling. Therefore, there is a strictly positive error $\Delta(s)$ of the objective function $C_{max}(s)$ calculated for the constructed and realized schedule s . In such a case, the proven sufficient conditions for the optimality of the schedule s do not hold in some decision-points (or in a single decision-point) at the on-line phase of scheduling. If Algorithm 3 terminates at Step 17, then an optimal pair of job permutations has been constructed for the factual scenario $p^* \in T$ randomly generated after completing the on-line phase of scheduling. The optimality of this pair of the job permutations was established only after the schedule execution, since the tested sufficient conditions for the optimality of the schedule s do not hold in some decision-points (or in a single decision-point).

If Algorithm 3 terminates at Step 18, then the tested sufficient conditions hold for all decision-points considered at the on-line phase of scheduling. Therefore, the constructed pair of job permutations is optimal for all factual scenarios $p^* \in T$ which were possible during the on-line phase of scheduling. In this case, the optimal pair of job permutations was established before the end of the schedule execution (after the last decision-point). The described Algorithm 3 must be used if the input (a) is obtained due to the application of Algorithms 1 and 2 [8] at the off-line phase of scheduling. Similarly, one can describe Algorithm 4 with the sufficient conditions from the analogs of Theorems 9 and 10 for their use in the case, when the input (b) is obtained due to the application of Algorithms 1 and 2 at the off-line phase of scheduling.

Similar Algorithm 5 must be used in the case, when the input (c) is obtained due to the application of Algorithms 1 and 2 at the off-line phase of scheduling. In Algorithm 3, a decision-point may occur on machine M_1 and on machine M_2 simultaneously. Therefore, one has to check the conditions of Theorems 9 and 10 or their analogs alternately for the corresponding conflict sets of the jobs from the set $\mathcal{J}_{1,2}$ and those from the set $\mathcal{J}_{2,1}$.

Algorithm 3 for the on-line phase of scheduling

- Input:** Lower bounds l_{ij} and upper bounds u_{ij} on the durations p_{ij} of all operations $O_{ij} \in \mathcal{J}$ processed on machines $M_j \in \mathcal{M}$;
 a permutation π_1 of the jobs \mathcal{J}_1 and a permutation π_2 of the jobs \mathcal{J}_2 ;
 an optimal permutation $\pi_{2,1}$ of the jobs from the set $\mathcal{J}_{2,1}$;
 a partial strict order $A_{\prec}^{1,2}$ of the jobs from the set $\mathcal{J}_{1,2}$;
 a number B of the conflict sets in the partial strict order $A_{\prec}^{1,2}$.
- Output:** Permutation $\pi_{1,2}$ of the jobs from the set $\mathcal{J}_{1,2}$.
- Step 1:** Set $b = 0$.
- Step 2:** **UNTIL** the completion time-point of the last job in the set \mathcal{J} ,
 process the whole linear part of the jobs in the partial strict order $A_{\prec}^{1,2}$
 on the machine M_1 till a conflict set of the jobs is met;
 let t denote a time-point of the completion of the linearly ordered set of jobs.
- Step 3:** Process jobs of the permutation $(\pi_{2,1}, \pi_2)$ and then process the linear part
 in the partial strict order $A_{\prec}^{1,2}$ on the machine M_2 up to time-point t .
- Step 4:** Check the conditions of Theorem 9 for the conflict set of the jobs.
- Step 5:** **IF** the sufficient conditions of Theorem 9 hold
THEN set $b := b + 1$ and choose an arbitrary order π_q of the conflict jobs
GOTO step 11.
- Step 6:** **ELSE** set $d_z = l_{z2} - u_{z1}$ for all conflict jobs J_z
 and partition the conflict jobs J_z into two subsets X_1 and X_2 ,
 where $J_z \in X_1$ if $d_z \geq 0$, and $J_z \in X_2$ otherwise.
- Step 7:** Construct the following order π_q of the conflict jobs:
 First, arrange the jobs from the set X_1 in the non-decreasing order
 of the values of u_{i1} , then arrange the jobs from the set X_2
 in the non-increasing order of the values of l_{i2} .
- Step 8:** Check the conditions of Theorem 10 for the constructed
 permutation of the conflict jobs.
- Step 9:** **IF** the sufficient conditions of Theorem 10 hold **THEN**
 set $b := b + 1$ **GOTO** step 11.
- Step 10:** Construct a Johnson's permutation π_q of the conflict jobs
 based on the inequalities (2) provided that $p_{ij} = (u_{ij} + l_{ij})/2$.
- Step 11:** Include the permutation π_q of the conflict jobs in the strict order
 $A_{\prec}^{1,2}$ instead of the conflict set of these jobs.
- Step 12:** **RETURN**
- Step 13:** **IF** $b = B$ **THEN GOTO** step 18.
- Step 14:** Calculate makespan $C_{\max}(s)$ for the schedule s constructed at steps 1 – 12;
 calculate makespan $C_{\max}(s^*)$ for the optimal schedule s^* polynomially
 calculated for the corresponding deterministic problem $J2|p^*, n_i \leq 2|C_{\max}$,
 where the factual processing times p^* are randomly generated for all jobs \mathcal{J} .
- Step 15:** **IF** $C_{\max}(s) = C_{\max}(s^*)$ **THEN GOTO** step 17.
- Step 16:** **STOP 4:** The constructed schedule s is not optimal for the factual
 processing times p^* of the jobs \mathcal{J} .
- Step 17:** **STOP 3:** The optimality of the constructed schedule s for the factual
 processing times p^* of the jobs \mathcal{J} was established only after
 the execution of the schedule s .
- Step 18:** **STOP 2:** The optimality of the constructed schedule s for the factual
 processing times p^* of the jobs \mathcal{J} was proven before the end
 of the execution of this schedule.

6.2. The Modified Example with Different Factual Scenarios

To demonstrate the on-line phase of scheduling based on Algorithm 3, it is considered Example 2 of the problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with the numerical input data given in Table 1 similarly as for Example 1 with the only one exception. It is assumed that $u_{3,2} = 6$.

The first part of the off-line phase of scheduling for solving Example 2 is similar to that for Example 1 till checking the conditions of Theorem 2. Indeed, the conditions of Theorem 2 do not hold for the jobs from the set $\mathcal{J}_{1,2}$ since the following strict inequalities hold: $u_{2,2} > l_{3,2}$ and $u_{3,2} > l_{2,2}$.

Due to checking the inequalities (3) and (4), one can determine the binary relation $A_{\prec}^{1,2}$ over the set $\mathcal{J}_{1,2}$ in the following form: $J_1 \prec \{J_2, J_3\}$. Thus, the set $\{J_2, J_3\}$ is a conflict set with two jobs; see Definition 5. Then, one can consecutively check the conditions of Theorems 6–8 for the jobs from the set $\mathcal{J}_{1,2}$. After letting $k = 1, r = 2$, one can calculate $L_2 = \sum_{J_i \in \mathcal{J}_{2,1} \cup \mathcal{J}_2} l_{i2} = l_{6,2} + l_{7,2} + l_{8,2} + l_{5,2} = 2 + 3 + 4 + 2 = 11$ and then obtain the following relations:

$$\sum_{i=1}^{k+r} u_{i1} = u_{1,1} + u_{2,1} + u_{3,1} = 7 + 9 + 9 = 25 \not\leq L_2 + \sum_{i=1}^k l_{i2} = L_2 + l_{1,2} = 11 + 6 = 17.$$

Thus, the condition of Theorem 6 does not hold for Example 2. Next, one can check the conditions of Theorem 7. Similarly as in the previous case, one can obtain that $L_2 = 11, k = 1$, and $r = 2$. Due to the condition (10), one can obtain two inequalities as follows: $s = 1$ and $s = 2$. Then, one can check both permutations of the jobs from the set $\mathcal{J}_{1,2}$, which satisfy the partial strict order $A_{\prec}^{1,2}$, as follows: $\Pi(\mathcal{G}_{1,2}) = \{\pi_{1,2}^1, \pi_{1,2}^2\}$, where $\pi_{1,2}^1 = \{J_1, J_2, J_3\}$ and $\pi_{1,2}^2 = \{J_1, J_3, J_2\}$.

Thus, the permutation $\pi_{1,2}^1$ must be tested. One can obtain the following relations:

$$u_{2,1} = 9 \leq L_2 + (l_{1,2} - u_{1,1}) = 11 + (6 - 7) = 10;$$

$$u_{3,1} = 9 \not\leq L_2 + \sum_{i=1}^2 (l_{i2} - u_{i1}) = L_2 + (l_{1,2} - u_{1,1}) + (l_{2,2} - u_{2,1}) = 11 + (6 - 7) + (5 - 9) = 6.$$

Hence, the condition of Theorem 7 does not hold for the permutation $\pi_{1,2}^1$.

Analogously, for the permutation $\pi_{1,2}^2$, the following relations hold:

$$u_{3,1} = 9 \leq L_2 + (l_{1,2} - u_{1,1}) = 11 + (6 - 7) = 10;$$

$$u_{2,1} = 9 \not\leq L_2 + \sum_{i=1}^2 (l_{i2} - u_{i1}) = L_2 + (l_{1,2} - u_{1,1}) + (l_{3,2} - u_{3,1}) = 11 + (6 - 7) + (4 - 9) = 5.$$

Hence, the condition of Theorem 7 does not hold for the permutation $\pi_{1,2}^2$ as well.

It is impossible to check the condition of Theorem 8, since the conflict set of the jobs $\{J_2, J_3\}$ is located at the end of the partial strict order $A_{\prec}^{1,2}$. Thus, the off-line phase of scheduling is completed, and the constructed partial strict order $A_{\prec}^{1,2}$ is not a linear order. Therefore, there does not exist a pair of permutations of the jobs, which is optimal for any scenario $p \in T$. In this case, Algorithms 1 and 2 [8] do not terminate with STOP 1. A scheduler needs to use the on-line phase of scheduling for solving Example 2 further.

The output of the off-line phase of scheduling for Example 2 contains the permutation $\pi_{2,1} = (J_6, J_7, J_8)$ of the jobs $\mathcal{J}_{2,1}$ processed on both machines M_1 and M_2 . The partial strict order $A_{\prec}^{1,2} = (J_1 \prec \{J_2, J_3\})$ of the jobs $\mathcal{J}_{1,2}$ is constructed. The obtained output (a) of the off-line phase of scheduling shows that Algorithm 3 must be used at the on-line phase of scheduling for solving Example 2.

We next show that Algorithm 3 can be stopped either with STOP 2 (Step 18) or with STOP 3 (Step 17) or with STOP 4 (Step 16) depending on the factual values of the job processing times. Note that $B = 1$; see Algorithm 3.

Case (j): Algorithm 3 is stopped at step 18 (STOP 2).

Consider Step 2 and Step 3 of Algorithm 3. The schedule execution begins as follows: at the initial time-point $t = 0$, machine M_1 starts to process operation $O_{1,1}$, while machine M_2 starts to process operation $O_{6,2}$. This process is continued until the time-point $t = 4$ when machine M_2 completes operation $O_{6,2}$. At this time-point, an exact value of the processing time $p_{6,2}^*$ becomes known, namely: $p_{6,2}^* = 4$. Then, machine M_2 starts to process operation $O_{7,2}$ and machine M_1 continues the processing of operation $O_{1,1}$. At the time-point $t = 6$, machine M_1 completes operation $O_{1,1}$. Therefore, an exact value of the duration of operation $O_{1,1}$ becomes known as follows: $p_{1,1}^* = 6$. At this time-point, a scheduler needs to choose either job J_2 or job J_3 to be processed next on machine M_1 . Note that machine M_2 continues to process the operation $O_{7,2}$ for two time units, wherein $l_{7,2} = 3$.

Consider Step 4 of Algorithm 3, where the condition (12) of Theorem 9 is checked for the conflict set of jobs $\{J_2, J_3\}$. Due to equalities $k = 1, r = 2, c_1(1) = 6, c_2(6) = 4$, one can obtain the following relations: $c_1(1) + u_{2,1} + u_{3,1} = 6 + 9 + 9 = 23 \not\leq c_2(6) + l_{7,2} + l_{8,2} + l_{5,2} + l_{1,2} = 4 + 3 + 4 + 2 + 6 = 19$.

At Steps 6 and 7 of Algorithm 3, one can obtain $d_2 = -4, d_3 = -5$ and permutation π_q having the following form: $\pi_q = (J_2, J_3)$. At Steps 8 and 9 of Algorithm 3, the conditions of Theorem 10 are checked as follows: $c_1(1) + u_{2,1} = 6 + 9 = 15 \leq c_2(6) + l_{7,2} + l_{8,2} + l_{5,2} + l_{1,2} = 4 + 3 + 4 + 2 + 6 = 19$;

$$c_1(1) + u_{2,1} + u_{3,1} = 6 + 9 + 9 = 24 \leq c_2(6) + l_{7,2} + l_{8,2} + l_{5,2} + l_{1,2} + l_{2,2} = 4 + 3 + 4 + 2 + 6 + 5 = 24.$$

At Step 11 of Algorithm 3, one can obtain the following strict order $A_{\prec}^{1,2} = (J_1 \prec J_2 \prec J_3)$ along with the permutation $\pi_{1,2} = (J_1, J_2, J_3)$. Since $b = 1 = B$ (see Step 13), Algorithm 3 is stopped at Step 18; see STOP 2. The optimal order of the conflict jobs J_2 and J_3 is found at the time-point $t = 6$ and the pair of job permutations $\pi' = (J_1, J_2, J_3, J_4, J_6, J_7, J_8)$ and $\pi'' = (J_6, J_7, J_8, J_5, J_1, J_2, J_3)$ is optimal for any scenario from the remaining set of possible scenarios $T(1, 6) = \{p \in T : p_{1,1}^* = 6, p_{6,2}^* = 4\}$.

Thus, an additional information on the exact values of the processing times $p_{6,2}^*$ and $p_{1,1}^*$ allows a scheduler to find an optimal order of all conflict jobs. It should be noted that the optimality of the constructed schedule is proven at the time-point $t = 6$, i.e., before the end of the schedule execution.

At the time-point $t = 6$, machine M_1 begins to process operation $O_{2,1}$. Note that all the above checks are performed at the time-point $t = 6$.

Case (jj): Algorithm 3 is stopped at Step 17 (STOP 3).

It is considered another possible realization of the semi-active schedule since another factual processing times are randomly generated at the on-line phase of scheduling for Example 2.

At the time-point $t = 0$, machine M_1 begins to process operation $O_{1,1}$, while machine M_2 begins to process operation $O_{6,2}$. Let machine M_2 complete operation $O_{6,2}$ at the time-point $t = 2.8$. Thus, the exact processing time $p_{6,2}^* = 2.8$ becomes known. Then, machine M_2 begins to process operation $O_{7,2}$ and completes this process at the time-point $t = 6$ (i.e., $p_{7,2}^* = 3.2$), while machine M_1 continues processing operation $O_{1,1}$. Let at the time-point $t = 6.9$, machine M_1 completes operation $O_{1,1}$ (i.e., $p_{1,1}^* = 6.9$). One needs to choose either job J_2 or job J_3 to be processed next on machine M_1 . At this time, machine M_2 continues to process the operation $O_{8,2}$ since $t = 6$ and $(6.9 - 6) = 0.9 < 4 = l_{8,2}$.

Based on the checking of the condition (12) of Theorem 9 for the conflict set of the jobs, one can obtain the following relations: $k = 1, r = 2, c_1(1) = 6.9, c_2(7) = 6$;

$$c_1(1) + u_{2,1} + u_{3,1} = 6.9 + 9 + 9 = 23.9 \not\leq c_2(7) + l_{8,2} + l_{5,2} + l_{1,2} = 6 + 4 + 2 + 6 = 18.$$

Similarly as in the previous case (j), one can obtain $d_2 = -4, d_3 = -5$, and the permutation π_q having the following form: $\pi_q = (J_2, J_3)$. The conditions of Theorem 10 are checked as follows:

$$c_1(1) + u_{2,1} = 6.9 + 9 = 15.9 \leq c_2(7) + l_{8,2} + l_{5,2} + l_{1,2} = 6 + 4 + 2 + 6 = 18;$$

$$c_1(1) + u_{2,1} + u_{3,1} = 6.9 + 9 + 9 = 24.9 \not\leq c_2(7) + l_{8,2} + l_{5,2} + l_{1,2} + l_{2,2} = 6 + 4 + 2 + 6 + 5 = 23.$$

Thus, the conditions of Theorem 10 do not hold. At Step 10 of Algorithm 3, one can construct a Johnson's permutation π_q of the conflict jobs based on the inequalities (2) for the processing times

of all conflict jobs determined as follows: $p_{ij} = (u_{ij} + l_{ij})/2$. For the jobs J_2 and J_3 , one can calculate $p_{2,1} = 8.5$, $p_{2,2} = 5.5$, $p_{3,1} = 8$, $p_{3,2} = 5$ and the Johnson's permutation π_q of the conflict jobs in the following form: $\pi_q = (J_2, J_3)$.

At the time-point $t = 6.9$, one can obtain the pair of permutations $\pi' = (J_1, J_2, J_3, J_4, J_6, J_7, J_8)$ and $\pi'' = (J_6, J_7, J_8, J_5, J_1, J_2, J_3)$ of the jobs for their processing on machines \mathcal{M} . Therefore, at the time-point $t = 6$, machine M_1 begins to process operation $O_{2,1}$. Then, at the time-point $t = 10$, machine M_2 completes operation $O_{8,2}$ (the exact processing time $p_{8,2}^* = 4$ becomes known), and then begins to process operation $O_{5,2}$ till the time-point $t = 12.4$ (thus, $p_{5,2}^* = 2.4$), and then begins to process operation $O_{1,2}$. At the time-point $t = 15.5$, machine M_1 completes operation $O_{2,1}$ (i.e., the exact processing time $p_{2,1}^* = 8.6$ becomes known), and then begins to process operation $O_{3,1}$.

Then, at the time-point $t = 18.7$, machine M_2 completes operation $O_{1,2}$ (the exact processing time $p_{1,2}^* = 6.3$ becomes known), and then begins to process operation $O_{2,2}$ till the time-point $t = 23.7$ (thus, $p_{2,2}^* = 5$). At this time-point, machine M_1 still processes operation $O_{3,1}$. As a result, machine M_2 has an idle time in the realized schedule.

At the time-point $t = 24.5$, machine M_1 completes operation $O_{3,1}$ (i.e., $p_{3,1}^* = 9$), and then begins to process operation $O_{4,1}$. Machine M_2 begins to process operation $O_{3,2}$ immediately.

At the time-point $t = 26.5$, machine M_1 completes operation $O_{4,1}$ (i.e., $p_{4,1}^* = 2$), and then begins to process operation $O_{6,1}$ till the time-point $t = 27.5$ (i.e., $p_{6,1}^* = 1$). Then, machine M_1 processes operation $O_{7,1}$ till the time-point $t = 28.5$ (i.e., $p_{7,1}^* = 1$), and then begins to process operation $O_{8,1}$.

At the time-point $t = 30.5$, machine M_2 completes operation $O_{3,2}$ (i.e., the exact processing time $p_{3,2}^* = 6$ becomes known). Thus, machine M_2 completes to process all jobs in the realized permutation π'' at the time-point $c_2(3) = 30.5$. At the time-point $t = 31.5$, machine M_1 completes operation $O_{8,1}$ (and the exact processing time $p_{8,1}^* = 3$ becomes known). Thus, machine M_1 completes to process all jobs in the realized permutation π' at the time-point $c_1(8) = 31.5$.

All uncertain processing times $p \in T$ took their factual values p_{ij}^* as follows:

$$p^* = (p_{1,1}^*, p_{1,2}^*, p_{2,1}^*, \dots, p_{7,2}^*, p_{8,1}^*, p_{8,2}^*) = (6.9, 6.3, 8.6, 5, 9, 6, 2, 0, 0, 2.4, 1, 2.8, 1, 3.2, 3, 4).$$

It should be remind that these factual processing times p^* were randomly generated at the time-points of the completions of the corresponding operations; see Remark 3.

For the constructed and realized schedule (π', π'') , the equalities $C_{\max}(\pi', \pi'') = \max\{c_1(8), c_2(3)\} = \max\{31.5, 30.5\} = 31.5$ hold; see Step 14 of Algorithm 3.

Now, one can check whether the constructed and realized schedule (π', π'') is optimal for the factual vector p^* of the job processing times. To this end, one can construct the pair of Jackson's permutations (π'_*, π''_*) for the deterministic problem $J2|p^*, n_i \leq 2|C_{\max}$ with the factual vector p^* of the job processing times. Then, one can find the optimal makespan value for the deterministic problem $J2|p^*, n_i \leq 2|C_{\max}$ as follows: $C_{\max}(\pi'_*, \pi''_*) = 31.5$; see Step 15 of Algorithm 3.

The obtained equalities $C_{\max}(\pi'_*, \pi''_*) = 31.5 = C_{\max}(\pi', \pi'')$ mean that Algorithm 3 has constructed the optimal schedule for the deterministic problem $J2|p^*, n_i \leq 2|C_{\max}$ with the factual vector p^* of the job processing times. However, the optimality of this constructed and realized schedule (π', π'') was established after the execution of the whole schedule (π', π'') . Indeed, Algorithm 3 is stopped at Step 17; see STOP 3. The constructed and realized schedule (π', π'') is presented in Figure 4 for case (jj) of the randomly generated factual processing times p^* of the jobs \mathcal{J} .

Case (jjj): Algorithm 3 is stopped at Step 16 (STOP 4).

It is considered the same process as in the previous case (jj) up to the time-point $t = 28.5$ when machine M_1 begins to process operation $O_{8,1}$ (machine M_2 processes operation $O_{3,2}$ at this time-point).

Let the equality $p_{8,1}^{**} = 1$ hold for the factual processing time $p_{8,1}^{**}$ of the operation $O_{8,1}$ and machine M_1 complete operation $O_{8,1}$. Thus, machine M_1 completes all operations of the jobs \mathcal{J} in the permutation π' at the time-point 29.5. Therefore, the equality $c_1(8) = 29.5$ holds. Similarly as in the

previous case, machine M_2 completes operation $O_{3,2}$ at the time-point $t = 30.5$. Thus, $p_{3,2}^* = 6$ and $c_2(3) = 30.5$. The factual vector of the job processing times is randomly generated as follows:

$$p^{**} = (p_{1,1}^*, p_{1,2}^*, p_{2,1}^*, \dots, p_{7,2}^*, p_{8,1}^{**}, p_{8,2}^*) = (6.9, 6.3, 8.6, 5, 9, 6, 2, 0, 0, 2.4, 1, 2.8, 1, 3.2, 1, 4).$$

The makespan value for the constructed and realized schedule (π', π'') is determined as follows: $C_{max}(\pi', \pi'') = \max\{c_1(8), c_2(3)\} = \max\{29.5, 30.5\} = 30.5$. However, the optimal makespan value for the deterministic problem $J2|p^{**}, n_i \leq 2|C_{max}$ with the factual vector p^{**} of the job processing times is equal to $29.7 < 30.5 = C_{max}(\pi', \pi'')$, since the optimal order of the jobs J_2 and J_3 is determined as follows: (J_3, J_2) . Hence, the constructed and realized schedule (π', π'') is not optimal for the factual vector $p^{**} \in T$ of the job processing times. In this case, Algorithm 3 is stopped at Step 16; see STOP 4.

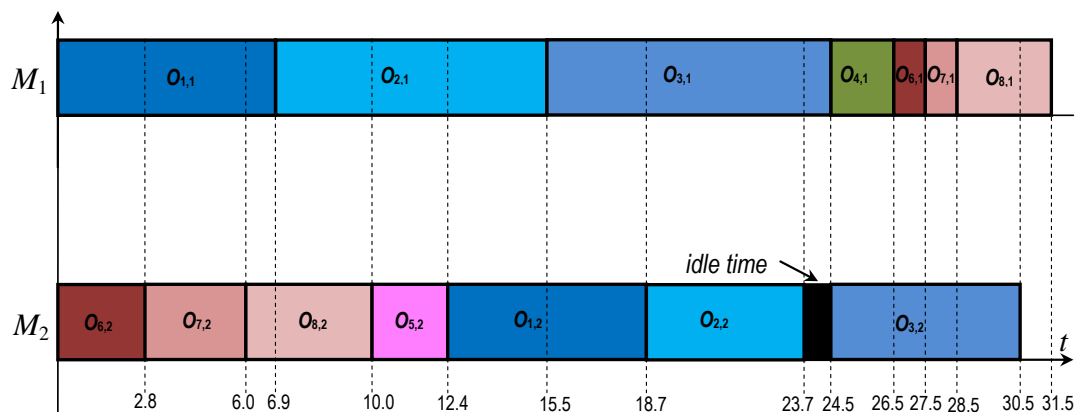


Figure 4. The optimal semi-active schedule for the Example 2 in case (jj).

6.3. Computational Experiments

We describe the computational experiments and computational results obtained for the tested randomly generated instances of the uncertain problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$. Each tested series consisted of 1000 randomly generated instances with fixed numbers $n \in \{10, 20, \dots, 100\}$ of the jobs \mathcal{J} and the maximum possible errors $\delta \in \{5\%, 10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%, 100\%\}$ of the random durations of the operations O_{ij} . The lower bounds l_{ij} and upper bounds u_{ij} on the possible values of the durations p_{ij} of operations O_{ij} , $p_{ij} \in [l_{ij}, u_{ij}]$, were randomly generated as follows. The lower bound l_{ij} was randomly chosen from the segment $[10, 100000]$ using a uniform distribution. The upper bound u_{ij} was determined using the equality $u_{ij} = l_{ij} \left(1 + \frac{\delta}{100}\right)$. The bounds l_{ij} and u_{ij} are decimal fractions with the maximum numbers of digits after the decimal points. The inequality $l_{ij} < u_{ij}$ holds for each job $J_i \in \mathcal{J}$ and each machine $M_j \in \mathcal{M}$.

Algorithms 1 and 2 developed in [8] were used at the off-line phase of scheduling. If the tested instance was not optimally solved using Algorithms 1 and 2, then corresponding Algorithms 3, 4 or 5 was used at the on-line phase of scheduling for solving further the instance of the uncertain problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$. All developed algorithms were coded in C# and tested on a PC with Intel Core i7-7700 (TM) 4 Quad, 3.6 GHz, 32.00 GB RAM.

In the computational experiments, two procedures were used to generate factual durations of the operations O_{ij} (a factual duration of the job J_i remained unknown until completing this job). In the first part of the computational experiments, the factual duration p_{ij}^* of the operation O_{ij} was randomly generated using a uniform distribution in the range $[l_{ij}, u_{ij}]$. In the second part of the computational experiments, two distribution laws were used in the experiments to determine the factual scenarios. Namely, we used the gamma distribution with parameters $(0.5; 1)$ (we call it as the distribution law with number 1) and the gamma distribution with parameters $(7.5; 1)$ (we call it as the distribution law

with number 2). For generating factual processing times for each tested instance, the number of the used distribution was randomly chosen from the possible set $\{1, 2\}$.

The sufficient conditions proven in Section 5 are verified in polynomial time $O(n^2)$ of the number n of the jobs \mathcal{J} . Therefore, all series of the tested instances in our computational experiments were solved very quickly (less than one second per a series with 1000 instances).

The experiments include testing of 14 classes of the instances of the uncertain problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ with different ratios of the numbers m_1 , m_2 , $m_{1,2}$ and $m_{2,1}$ (where $n = m_1 + m_2 + m_{1,2} + m_{2,1}$) of the jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the set \mathcal{J} , respectively. Every class of the tested instances of the problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ is characterized by the following ratio:

$$\frac{m_1}{n} \cdot 100\% : \frac{m_2}{n} \cdot 100\% : \frac{m_{1,2}}{n} \cdot 100\% : \frac{m_{2,1}}{n} \cdot 100\% \quad (28)$$

of the percentages of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the set \mathcal{J} , respectively.

Tables A1–A14 present the computational results obtained for the tested classes of instances with the following ratios (28):

0% : 0% : 10% : 90% (class 1, Table A1); 0% : 0% : 20% : 80% (class 2, Table A2);
 0% : 0% : 30% : 70% (class 3, Table A3); 0% : 0% : 40% : 60% (class 4, Table A4);
 0% : 0% : 50% : 50% (class 5, Table A5); 5% : 5% : 5% : 85% (class 6, Table A6);
 5% : 15% : 5% : 75% (class 7, Table A7); 5% : 20% : 5% : 70% (class 8, Table A8);
 10% : 10% : 10% : 70% (class 9, Table A9); 10% : 10% : 40% : 40% (class 10, Table A10);
 10% : 20% : 10% : 60% (class 11, Table A11); 10% : 30% : 10% : 50% (class 12, Table A12);
 10% : 40% : 10% : 40% (class 13, Table A13); 10% : 60% : 10% : 20% (class 14, Table A14).

All Tables A1–A14 are organized as follows. The procedure for generating factual processing times (the uniform distribution or the gamma distribution) is indicated in the first row of each table. Numbers n of the given jobs \mathcal{J} in the tested instances of the problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ are presented in the second row. The maximum possible errors δ of the randomly generated processing times (in percentages) are presented in the first column. For the fixed maximum possible error δ , the obtained computational results are presented in four rows called Stop1, Stop2, Stop3 and Stop4.

The row Stop1 determines the percentage of instances from the tested series, which were optimally solved at the off-line phase of scheduling using either Algorithms 1 or 2 developed in [8]. For such an instance, an optimal pair (π', π'') of the job permutations was constructed before the time-point of starting the first job of the realized schedule, i.e., the equality $C_{max}(\pi', \pi'') = C_{max}(\pi^*, \pi^{**})$ holds, where $(\pi^*, \pi^{**}) \in S$ is an optimal pair of job permutations for the deterministic problem $J2|p^*, n_i \leq 2|C_{max}$ with the factual scenario $p^* \in T$ that is unknown before completing the whole jobs \mathcal{J} .

The row Stop2 determines the percentage of instances, which were optimally solved at the on-line phase of scheduling using corresponding Algorithms 3, 4 or 5. For each such an instance, an optimal pair (π', π'') of job permutations for the deterministic problem $J2|p^*, n_i \leq 2|C_{max}$ associated with the factual scenario $p^* \in T$ was constructed by checking sufficient conditions in Theorem 9 or Theorem 10. Remind that the factual scenario $p^* \in T$ for the uncertain problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ remains unknown until completing the jobs \mathcal{J} .

The row Stop3 determines the percentage of instances, which were optimally solved at the on-line phase of scheduling using Algorithms 3, 4 or 5. In such a case, an optimal pair of job permutations has been constructed for the factual scenario $p^* \in T$. However, the optimality of this pair of job permutations was established only after the execution of the constructed schedule.

The row Stop4 determines the percentage of instances, for which the constructed and realized schedule is not optimal for the deterministic instance $J2|p^*, n_i \leq 2|C_{max}$ with the factual scenario p^* .

6.4. Computational Results

First of all, it is important to determine a total number of the tested instances, for which 3 (or Algorithms 4 and 5) were completed at Step 18 (STOP 2) or at Step 17 (STOP 3). This number shows

how many tested instances of the uncertain job-shop scheduling problem have been optimally solved either with the proofs of their optimality before the completion of processing all jobs \mathcal{J} (STOP 2) or the optimality of the obtained schedule was established after the realization of the constructed schedule (STOP 3). For the numbers of jobs from $n = 10$ to $n = 100$ and for each value of the tested errors δ of the processing times, average percentages of the instances optimally solved by Algorithms 1, 2, 3, 4 or 5 (these average percentages summarize the values given in rows Stop1 and Stop2 in all Tables A1–A14) are presented in Table 2 and Figure 5.

Table 2. Average percentages of the instances whose optimality of the constructed permutations was proven at the off-line and on-line phases of scheduling.

$\delta\%$	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0% : 0% : 10% : 90%	98.38	95.66	74.32	40.09	18.56	8.65	4.4	2.48	1.58	0.95	0.57
0% : 0% : 20% : 80%	99.48	98.81	94.37	74.52	45.33	24.20	12.32	6.55	3.87	2.59	1.65
0% : 0% : 30% : 70%	99.87	99.71	98.99	95.09	81.33	59.33	36.65	21.67	11.93	7.28	4.39
0% : 0% : 40% : 60%	99.97	99.94	99.73	99.02	97.09	90.86	78.5	59.76	40.85	25.93	15.2
0% : 0% : 50% : 50%	100	99.99	99.93	99.73	99.13	97.55	94.21	86.65	72.35	51.41	29.45
5% : 5% : 5% : 85%	99.67	98.45	78.97	43.19	19.26	9.11	4.31	2.02	1.1	0.58	0.27
5% : 15% : 5% : 75%	99.64	98.86	84.44	51.06	24.65	11.88	6.43	3.34	1.95	1.02	0.68
5% : 20% : 5% : 70%	99.57	98.97	86.92	55.79	29.59	14.98	7.97	4.42	2.59	1.45	1.01
10% : 10% : 10% : 70%	99.84	99.48	97.46	83.55	57.09	34.11	18.95	11.14	6.81	4.29	2.8
10% : 10% : 40% : 40%	99.99	100	99.96	99.89	99.69	99.35	98.41	96.55	92.84	85.66	73.22
10% : 20% : 10% : 60%	99.87	99.68	98.37	90.56	71.05	48.26	30.28	18.22	11.36	7.21	4.79
10% : 30% : 10% : 50%	99.9	99.72	99.11	95.33	83.85	66.15	49.34	34.35	24.13	16.64	11.44
10% : 40% : 10% : 40%	99.92	99.75	99.33	97.97	92.52	82.69	70.01	58.6	48.52	40.36	32.76
10% : 60% : 10% : 20%	99.98	99.98	99.93	99.83	99.59	99.01	97.96	96.16	93.01	89.46	85.75

Table 2 shows the total percentages of the optimally solved instances for all classes of the tested instances, for which the optimal schedules were constructed either at the off-line phase of scheduling (STOP 1) or at the on-line phase of scheduling (STOP 2). One can see that for three small values of the maximal errors $\delta \in \{5\%, 10\%, 20\%\}$ for most classes, more than 90% (up to 100%) of the tested instances were optimally solved. For all tested classes with a maximal error $\delta \leq 20\%$, more than 70% tested instances were optimally solved at the off-line or on-line phases of scheduling.

With a further increasing of the maximal error δ , the percentage of solved instances drops rapidly. For most tested classes with the maximal error δ greater than 70%, the percentage of solved instances is less than 10%. However, these indicators differ for different tested classes. For classes 4, 5, 10, 13 and 14 with maximal errors $\delta \leq 70\%$, more than 60% of the tested instances were optimally solved with the proof of the optimality before completing all the jobs. The best computational results are obtained for classes 5, 10 and 14 of the tested instances. More than 80% of the instances from these three classes were optimally solved at the off-line phase of scheduling or at the on-line phases of scheduling provided that the maximal error δ of the given job processing times was no greater than 70%, i.e., for $\delta \in \{5\%, 10\%, 15\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%\}$. For both classes 10 and 14 of the tested instances even with an error $\delta = 100\%$, more than 70% of the instances were optimally solved.

On the other hand, for both classes 1 and 6 with a maximal error $\delta = 40\%$, only less than 20% of the tested instances were optimally solved at both off-line phase and on-line phase of scheduling. For classes 1 and 6 with $\delta = 50\%$, less than 10% of the tested instances were optimally solved. Furthermore, these two classes of instances are most difficult ones to find an optimal schedule with the proof of its optimality before completing all the jobs using the on-line phase and off-line phase

of scheduling. It should be noted that all tested classes of instances demonstrate a monotonic decrease in the percentages of the optimally solved problems with an increase of the values of the maximal error δ of the job processing times; see Figure 5.

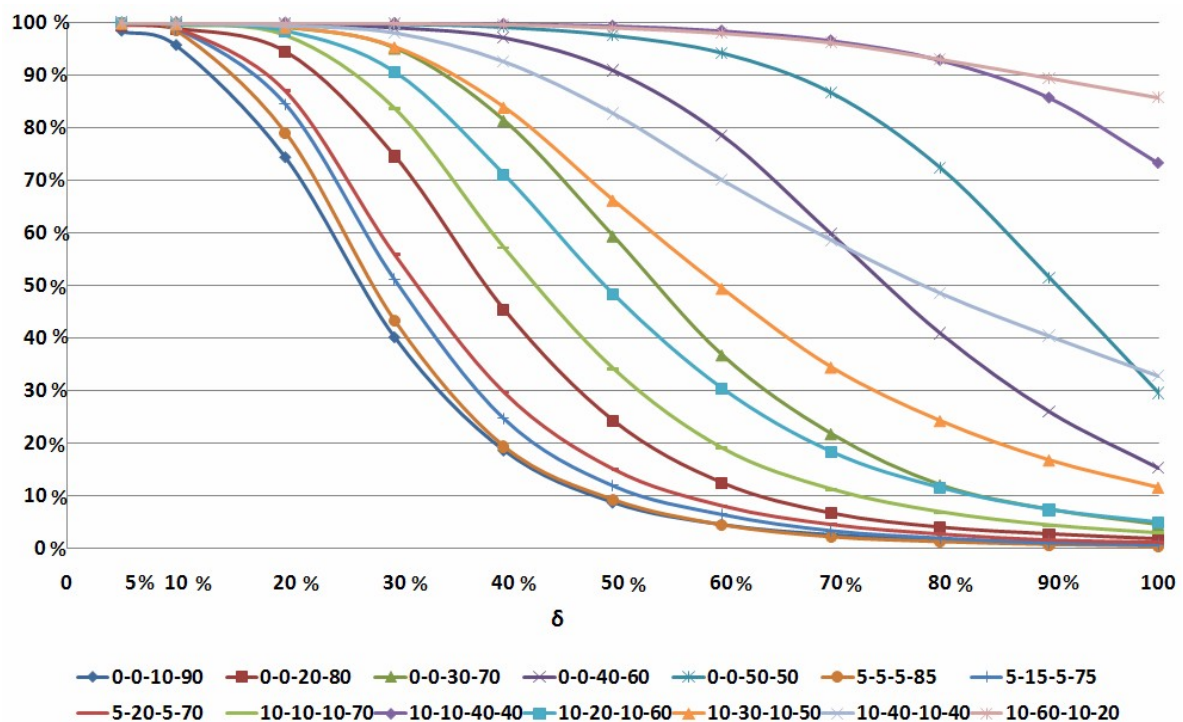


Figure 5. Average percentages of the instances whose optimality of the constructed pair of job permutations was proven at the off-line phase and on-line phase of scheduling.

Let us consider the percentages of the tested instances, for which the optimality of the constructed schedules was proven at the on-line phase of scheduling and the proofs of their optimality being obtained before completing all the jobs. Note that it is novelty of this paper; see rows Stop2 in Tables A1–A14. For all tested numbers of the jobs, $n \in \{10, 20, \dots, 100\}$, and for all maximal values of the errors $\delta \in \{5\%, 10\%, 20\%, \dots, 100\%\}$ of the job processing times, the average percentages of the instances, which were optimally solved by Algorithms 3, 4 or 5 at the on-line phase of scheduling are presented in Table 3, where only Stop2 is indicated.

It should be noted that the monotonous increase of the percentages of the optimally solved instances takes place only for classes 10 and 14 of the tested instances. For other tested classes of instances, there is a maximum, and for the different classes of the tested instances, these maximal values being achieved for different maximal values of the errors δ . Then the percentages of the optimally solved instances decrease again with the increasing of the maximal values δ . The values of the maximal numbers of instances, which optimal solutions have been proven at the on-line phase of scheduling (STOP 2), vary from 0.59% to 8.69% for different classes of instances.

Classes 1–5 are distinguished from the above classes since their maximal numbers of the instances optimally solved at the on-line phase of scheduling vary from 6% to 9%. Average percentages of the instances from these five classes, which were optimally solved by Algorithms 3, 4 or 5 at the on-line phase of scheduling (only Stop2) are shown in Figure 6.

Note that for the difficult classes 1 and 6, the percentages of instances, which were optimally solved at the on-line phase of scheduling with the proofs of their optimality, behave identically with the reaching of the maximum for the maximal error $\delta = 20\%$. However their maximal values differ, namely: from 2.96% for class 6 up to 8.69% for class 1.

Table 3. Average percentages of the instances whose optimality of the constructed permutations was proven at the on-line phase of scheduling.

$\delta\%$	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0% : 0% : 10% : 90%	0.97	3.68	8.69	7.26	3.86	1.92	1.06	0.5	0.34	0.2	0.11
0% : 0% : 20% : 80%	0.24	1.24	5.23	8.59	7.85	5.33	2.63	1.49	0.78	0.59	0.32
0% : 0% : 30% : 70%	0.03	0.22	1.33	4.27	6.82	8.13	5.78	4	2.49	1.66	1.07
0% : 0% : 40% : 60%	0.02	0.06	0.31	0.93	2.05	4.15	5.57	6.53	5.6	4.44	3.23
0% : 0% : 50% : 50%	0	0.01	0.06	0.25	0.53	1.24	2.62	4.33	6.06	7.4	6.23
5% : 5% : 5% : 85%	0.33	1.01	2.96	2.31	1.63	0.88	0.49	0.25	0.15	0.07	0.03
5% : 15% : 5% : 75%	0.27	0.84	2.64	2.39	1.78	0.93	0.47	0.3	0.19	0.09	0.08
5% : 20% : 5% : 70%	0.28	0.69	2.38	2.86	1.76	1.07	0.6	0.46	0.26	0.12	0.04
10% : 10% : 10% : 70%	0.06	0.165	0.83	1.87	1.95	1.56	1.08	0.69	0.43	0.32	0.24
10% : 10% : 40% : 40%	0	0	0.02	0.04	0.05	0.14	0.21	0.3	0.45	0.68	0.77
10% : 20% : 10% : 60%	0.05	0.13	0.6	1.29	1.77	1.6	1.22	0.82	0.57	0.41	0.25
10% : 30% : 10% : 50%	0.01	0.07	0.36	0.91	1.49	1.32	1.48	1.03	0.7	0.48	0.36
10% : 40% : 10% : 40%	0	0.03	0.16	0.41	0.78	1.17	1.29	1.01	0.9	0.62	0.46
10% : 60% : 10% : 20%	0	0	0.01	0.05	0.08	0.17	0.28	0.38	0.45	0.51	0.59

For the instances, for which the optimality of the constructed schedules was not proven before completing all the jobs \mathcal{J} , the relative errors $\Delta\%$ of the achieved objective function values for the realized schedules were calculated. Note that the positive errors $\Delta\%$ may occur only if Algorithm 3 (or Algorithms 4 and 5) have been stopped at Step 16; see STOP 4. For all tested numbers of jobs $n \in \{10, 20, \dots, 100\}$ and for all maximal values of the errors $\delta \in \{5\%, 10\%, 20\%, \dots, 100\%\}$ of the job processing times, the maximal values of $\Delta_{max}\%$ and the average values of $\Delta_{ave}\%$ were calculated separately for instances with uniform distributions (see Table 4) and gamma distributions (see Table 5).

It can be seen that the values of maximal errors $\Delta_{ave}\%$ significantly differ when applying different distribution laws. With using a uniform distribution, the maximal error Δ_{max} does not exceed 9%, while when using a gamma distribution, the maximal error Δ_{max} could reach a value more than 17%.

It can be seen that for using various distribution laws, Algorithm 3 (Algorithms 4 and 5 as well) terminates at STOP 4 with various combinations of the tested classes and maximal errors $\delta\%$. If a uniform distribution is used, then for classes 1–2, strictly positive errors $\Delta_{ave}\%$ arise for all values of the tested maximal errors $\delta\%$. For classes 9–10 and 11–13, such errors appear more often with increasing the maximal error $\delta\%$.

For a gamma distribution, for all values of $\delta\%$, the error $\Delta_{ave}\%$ arises only for class 1. For classes 2–4, 6, 8, 10, the error $\Delta_{ave}\%$ arises with the growth of maximal errors $\delta\%$. For classes 7, 9, 11–13, on the contrary, the error $\Delta_{ave}\%$ is more common for small values of the maximal errors $\delta\%$.

Table 4. Maximal errors Δ_{max} and average errors Δ_{ave} for all tested instances with factual processing times randomly generated based on a uniform distribution.

Class	$\delta\%$	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	Δ_{max}	0.031511	0.300924	0.691062	0.333292	1.110492	0.881902	2.246299	3.263145	2.729286	3.85936	5.024917
	Δ_{ave}	0.000003	0.000058	0.000225	0.000057	0.000333	0.000613	0.000733	0.00135	0.001868	0.001409	0.002253
2	Δ_{max}	0.0475	0.189872	0	0.125467	0	0.441669	0.243659	1.076096	3.127794	1.286158	1.353086
	Δ_{ave}	0.000005	0.000023	0	0.000013	0	0.000064	0.000045	0.000189	0.000976	0.000221	0.000135
3	Δ_{max}	0	0	0	0	0	0	0	0	0.8199	0	0
	Δ_{ave}	0	0	0	0	0	0	0	0	0.000082	0	0
4	Δ_{max}	0	0	0	0	0	0	0	0	0	0	0
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0	0
5	Δ_{max}	0	0	0	0	0	0	0	0	0	0	0
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0	0
6	Δ_{max}	0	0	0	0	0	0.411415	0.081623	0	0	0	0
	Δ_{ave}	0	0	0	0	0	0.000082	0.000016	0	0	0	0
7	Δ_{max}	0	0	0	0	0	0	0	0	0	0	0
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0	0
8	Δ_{max}	0	0	0	0	0	0	0	0	0	0	0
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0	0
9	Δ_{max}	0	0.068237	0.055299	0	1.31253	0	0.91223	0.893705	1.697913	2.166717	8.617851
	Δ_{ave}	0	0.000007	0.000006	0	0.000244	0	0.000144	0.000167	0.000338	0.000558	0.001348
10	Δ_{max}	0	0	0	0	0	0	0	0	0	0	0
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0	0
11	Δ_{max}	0	0	0	1.47875	3.660297	5.724288	0.810014	3.316178	0.39653	4.42828	4.666154
	Δ_{ave}	0	0	0	0.000148	0.000694	0.000572	0.000081	0.000332	0.000040	0.000799	0.000924
12	Δ_{max}	0	0	0	0	0	0	0	0	0	0	7.243838
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0	0.000724
13	Δ_{max}	0	0	0	0	0	0	0	0	0	0	5.036085
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0	0.000504
14	Δ_{max}	0	0	0	0	0	0	0	0	0	0	0
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0	0

As one can see, using the uniform distribution for the generation of the factual job processing times for classes 4, 5, 7, 10, 14, all tested instances were solved optimally using the developed algorithms and two phases of scheduling. In other words, there are no instances, for which corresponding Algorithms 3, 4 or 5 was stopped at Step 16 (STOP 4). However, for the gamma distribution, there are only two such classes 5 and 14. Thus, classes 5 and 14 can be considered as easy ones, while class 1 is the most difficult one. As for class 1, Algorithms 3, 4 and 5 are stopped at Step 16 (STOP 4) for all values of the tested maximal errors $\delta\%$. Moreover, the maximum makespan error $\Delta_{max}\%$ of more than 5% for the uniform distribution and more than 10% for the gamma distribution is found for classes 1, 9, 11 and 12 of the tested instances (these classes are difficult for the used stability approach).

Table 5. Maximal errors Δ_{max} and average errors Δ_{ave} for all tested instances with factual processing times randomly generated based on a gamma distribution.

Class	$\delta\%$	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	Δ_{max}	0.211802	0.509319	1.995284	2.439105	4.8928	2.670648	6.613935	8.782202	10.59834	9.153295	9.50327
	Δ_{ave}	0.000055	0.000131	0.00031	0.000746	0.001754	0.001746	0.00369	0.005033	0.006755	0.011687	0.01144
2	Δ_{max}	0	0	0	0	0	4.182544	1.361694	1.070905	5.634459	7.845669	7.974282
	Δ_{ave}	0	0	0	0	0	0.000595	0.000317	0.000107	0.001382	0.001737	0.001725
3	Δ_{max}	0	0	0	0	1.32533	0	0	5.566808	0	4.026352	5.385314
	Δ_{ave}	0	0	0	0	0.000133	0	0	0.000557	0	0.000511	0.001354
4	Δ_{max}	0	0	0	0	0	0	0	0	0	6.044646	0
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0.000604	0
5	Δ_{max}	0	0	0	0	0	0	0	0	0	0	0
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0	0
6	Δ_{max}	0	0	0	0.061048	0	0.387884	1.081353	1.125343	0.710307	0.643768	0.67762
	Δ_{ave}	0	0	0	0.000012	0	0.000078	0.000216	0.000401	0.000206	0.000343	0.000136
7	Δ_{max}	0	0	0.143177	0	0	0	0	0	0	0	0
	Δ_{ave}	0	0	0.000029	0	0	0	0	0	0	0	0
8	Δ_{max}	0	0	0.388797	0	0.426346	0	0.289505	0.059146	2.478004	1.167724	4.748751
	Δ_{ave}	0	0	0.000078	0	0.000085	0	0.000029	0.000006	0.000442	0.000234	0.000948
9	Δ_{max}	2.64165	6.620637	5.266738	4.163808	10.56515	0	0	0	0	0	0
	Δ_{ave}	0.000852	0.001946	0.001696	0.001615	0.003941	0	0	0	0	0	0
10	Δ_{max}	0	0	0	0	0	0	0	0.714515	0	0.334513	3.232162
	Δ_{ave}	0	0	0	0	0	0	0	0.000071	0	0.000033	0.000584
11	Δ_{max}	2.988951	10.50113	2.526341	5.632594	7.258956	0	0	0	0	0	0
	Δ_{ave}	0.0003	0.001289	0.000431	0.000887	0.001595	0	0	0	0	0	0
12	Δ_{max}	0	0.095929	1.639148	17.64929	6.3913	0	0	0	0	0	0
	Δ_{ave}	0	0.000010	0.000164	0.002948	0.001737	0	0	0	0	0	0
13	Δ_{max}	0	0.967642	0.7847	0	0	0	0	0	0	0	0
	Δ_{ave}	0	0.000097	0.000078	0	0	0	0	0	0	0	0
14	Δ_{max}	0	0	0	0	0	0	0	0	0	0	0
	Δ_{ave}	0	0	0	0	0	0	0	0	0	0	0

Class 13 of the tested instances is a rather strange one. For using the uniform distribution, a maximum makespan error $\Delta_{max}\%$ of more than 5% was obtained, while when for using the gamma distribution, the maximum makespan error $\Delta_{max}\%$ did not reach even 1%. Note that for all tested classes of the instances, the average makespan errors $\Delta_{ave}\%$ for all tested numbers $n \in \{10, 20, \dots, 100\}$ of jobs \mathcal{J} are less than 0.02%.

Maximal relative makespan errors $\Delta_{max}\%$ for each tested class and for all values of the tested maximal errors δ are shown in Figure 7 for the instances with uniform distributions and in Figure 8 for the instances with gamma distributions of the factual durations of the given operations.

Figures 7 and 8 also show that the maximal value of the makespan errors $\Delta_{max}\%$ for the constructed and realized schedule for the factual scenarios are achieved for different values of the maximal errors $\delta\%$ for different classes of the tested instances.

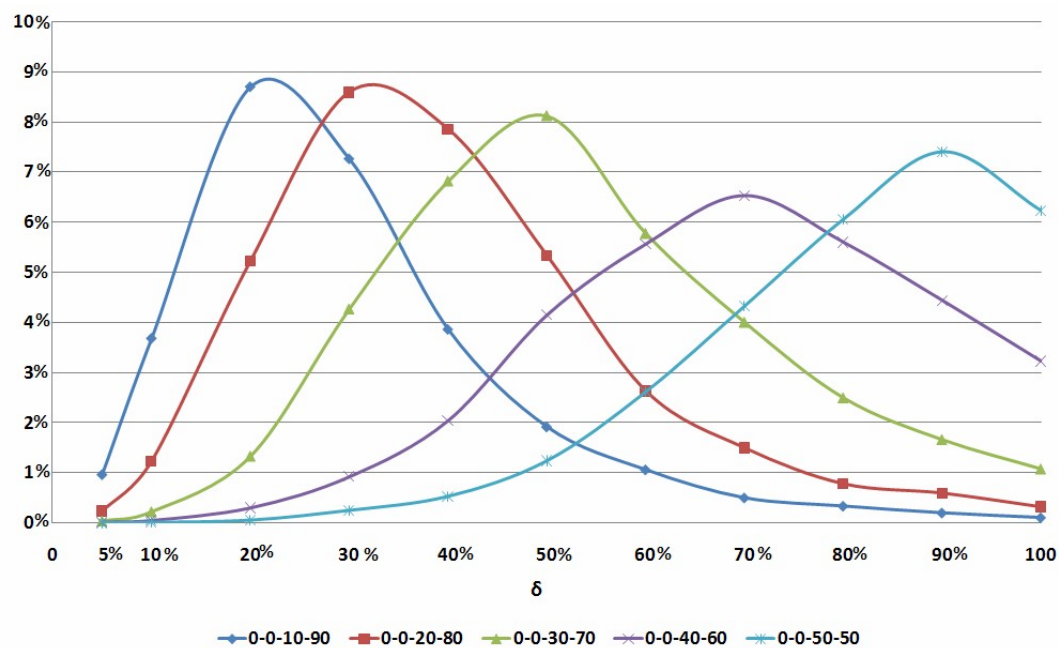


Figure 6. Average percentages of the instances whose optimality of the constructed pair of job permutations was proven at the on-line phase of scheduling.

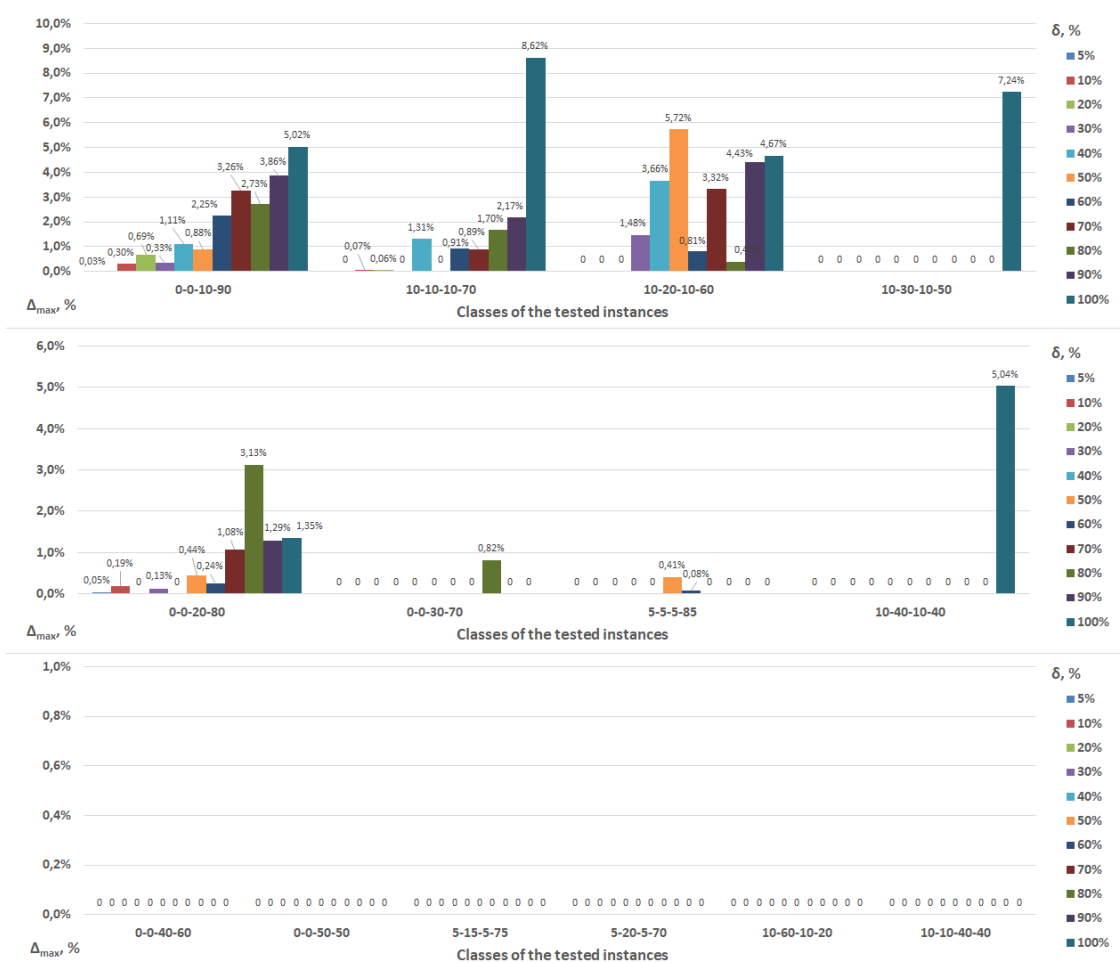


Figure 7. Maximal errors Δ_{\max} for the tested instances with a uniform distribution.

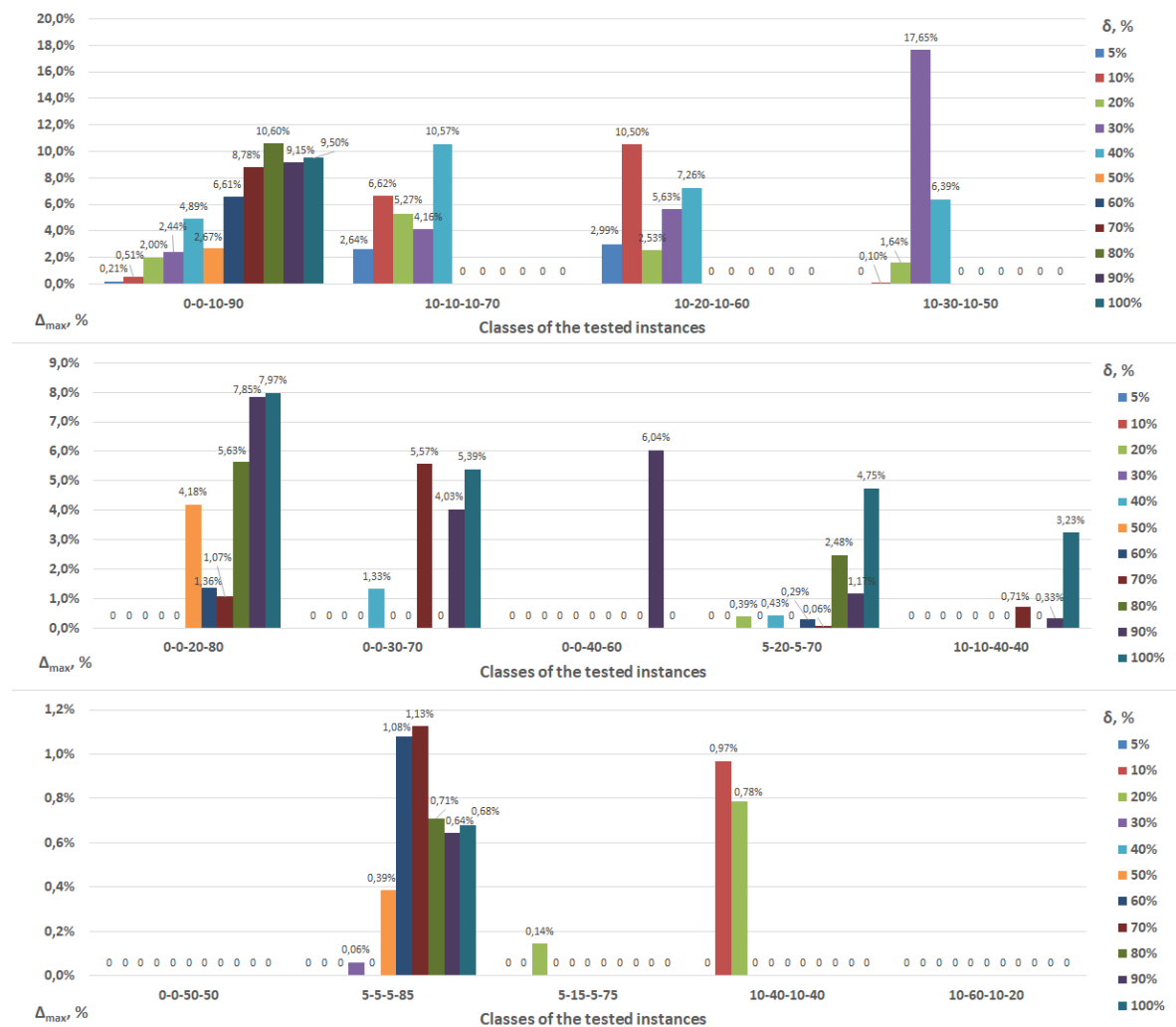


Figure 8. Maximal errors Δ_{max} for the tested instances with a gamma distribution.

7. Concluding Remarks

The uncertain job-shop scheduling problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ attract the attention of practitioners and researchers since this problem is applicable in real-life processing systems for some reduction of production costs due to a better utilization of the available machines and resources.

This paper is a continuation of our previous one [8], where only off-line phase of scheduling was investigated and tested for the uncertain problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ based on the stability approach. In [8], we tested 15 classes of the randomly generated instances $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$. A lot of instances from nine easy classes were optimally solved at the off-line phase of scheduling. If the maximal errors were no greater than 20%, i.e., $\delta \in \{5\%, 10\%, 15\%, 20\%\}$, then more than 80% of the tested instances were optimally solved at the off-line phase of scheduling. If the maximal error was equal to 50%, i.e., $\delta = 50\%$, then 45% of the tested instances were optimally solved.

However, less than 5% of the tested instances with maximal possible error $\delta \geq 20\%$ from six hard tested classes were optimally solved at the off-line phase of scheduling. There were no tested hard instances with the maximal error 50% optimally solved in [8]. All these difficulties were succeeded in Sections 4–6 of this paper, where it is shown that the on-line phase of scheduling allows a scheduler to find either optimal schedule or very close to optimal ones. Additional information on the factual value of the job processing times becomes available once the processing of the job on the machine is completed. Using this information, a scheduler can determine a smaller dominant set of semi-active schedules, which is based on sufficient conditions for schedule dominance. The smaller dominant set

enables a scheduler to quickly make an on-line scheduling decision whenever additional information on processing the job becomes available.

In Section 5, it is investigated the optimal pair (π', π'') of job permutations (Theorems 9 and 10). Using the proven analytical results, we derived Algorithms 3–5 for constructing optimal pairs (π', π'') of job permutations for all scenarios $p \in T$ or a small dominant set $S(T)$ of schedules for the uncertain problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$. At the off-line scheduling phase, Algorithms 1 and 2 [8] are used to determine the partial strict order $A_{\prec}^{1,2}$ over the job set $\mathcal{J}_{1,2}$ and the partial strict order $A_{\prec}^{2,1}$ over the job set $\mathcal{J}_{2,1}$. The constructed precedence digraphs $(\mathcal{J}_{1,2}, A_{\prec}^{1,2})$ and $(\mathcal{J}_{2,1}, A_{\prec}^{2,1})$ determine a minimal dominant set $S(T)$ of schedules.

In Sections 6, it is shown how to use Algorithms 3–5 for constructing a small dominant set of semi-active schedules that enables a scheduler to make a fast decision whenever information on completing some jobs become available. Based on these algorithms, the problem $J2|l_{ij} \leq p_{ij} \leq u_{ij}, n_i \leq 2|C_{max}$ was solved with very small errors of the obtained objective values. The computational experiments (Section 6.3) show that pairs of job permutations constructed by Algorithms 3–5 are very close to the optimal pairs of job permutations. We tested 14 classes of randomly generated instances. For the tested instances, the percentage of the optimally solved instances slowly decreases with increasing maximal errors δ of the processing times. The developed on-line algorithms perform with the maximal errors of the achieved makespan less than 1% if $n \in \{20, 30, \dots, 100\}$. For all tested classes of the instances, the average makespan errors for all numbers $n \in \{10, 20, \dots, 100\}$ of the jobs \mathcal{J} were less than 0.02%.

In a possible further research, one can continue the study of the uncertain job-shop scheduling problem based on the stability approach. It is useful to improve the developed algorithms and to extend them for other machine environments, such as a single machine or processing systems with parallel machines. It is promising to investigate an optimality region of the semi-active schedule and to develop algorithms for constructing a semi-active schedule with the largest optimality region.

It is also useful to apply the stability approach for solving the uncertain flow-shop and job-shop scheduling problems with $|\mathcal{M}| \geq 3$ different machines.

Author Contributions: Y.N.S. and N.M.M. jointly proved theoretical results; Y.N.S. and N.M.M. jointly conceived and designed the algorithm; V.H. performed the experiments; Y.N.S., N.M.M. and V.D.H. jointly analyzed the data; Y.N.S. and N.M.M. jointly wrote the paper. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: We are thankful for useful remarks and suggestions provided by four anonymous reviewers on the earlier drafts of our paper.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Algorithms 1 and 2 Developed in [8].

Algorithm 1

Input: Segments $[l_{ij}, u_{ij}]$ for all jobs $J_i \in \mathcal{J}$ and machines $M_j \in \mathcal{M}$,
a partial strict order $A_{\prec}^{1,2}$ on the set $\mathcal{J}_{1,2} = \mathcal{J}_{1,2}^* \cup \mathcal{J}_{1,2}^1 \cup \mathcal{J}_{1,2}^2$ in the form
 $J_1 \prec \dots \prec J_k \prec \{J_{k+1}, J_{k+2}, \dots, J_{k+r}\} \prec J_{k+r+1} \prec \dots \prec J_{m_{1,2}}$.

Output: **EITHER** an optimal job permutation for the problem
 $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with job set $\mathcal{J}_{1,2}$ and any scenario $p \in T$, (see STOP 0).
OR there no permutation $\pi_{1,2}$ of jobs from set $\mathcal{J}_{1,2}$, which is optimal
for all scenarios $p \in T$, (see STOP 1).

Step 1: Set $\delta_s = l_{k+s,2} - u_{k+s,1}$ for all $s \in \{1, 2, \dots, r\}$.
construct a partition of the set of conflicting jobs into two subsets X_1 and X_2 ,
where $J_{k+s} \in X_1$ if $\delta_s \geq 0$, and $J_{k+s} \in X_2$, otherwise.

Step 2: Construct a permutation $\pi^1 = (J_1, J_2, \dots, J_k, \pi_1, \pi_2, J_{k+r+1}, \dots, J_{m_{1,2}})$, where the permutation
 π_1 contains jobs from the set X_1 in the non-decreasing order of the values $u_{k+i,1}$ and the
permutation π_2 contains jobs from the set X_2 in the non-increasing order of the values
 $l_{k+i,2}$, renumber jobs in the permutations π_1 and π_2 based on their orders.

Step 3: **IF** for the permutation π^1 conditions of Theorem 7 hold **THEN GOTO** step 8.

Step 4: Set $\delta_s = l_{k+s,1} - u_{k+s,2}$ for all $s \in \{1, 2, \dots, r\}$.
construct a partition of the set of conflicting jobs into two subsets
 Y_1 and Y_2 , where $J_{k+s} \in Y_1$ if $\delta_s \geq 0$, and $J_{k+s} \in Y_2$, otherwise.

Step 5: Construct a permutation $\pi^2 = (J_1, J_2, \dots, J_k, \pi_2, \pi_1, J_{k+r+1}, \dots, J_{m_{1,2}})$, where the permutation
 π_1 contains jobs from the set Y_1 in the non-increasing order of the values $u_{k+i,2}$, and the
permutation π_2 contains jobs from the set Y_2 in the non-decreasing order of the
values $l_{k+i,1}$, renumber jobs in the permutations π_1 and π_2 based on their orders.

Step 6: **IF** for the permutation π^2 conditions of Theorem 8 hold **THEN GOTO** step 9.

Step 7: **ELSE** there is no a single dominant permutation for problem
 $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with job set $\mathcal{J}_{1,2}$ and any scenario $p \in T$ **STOP 1**.

Step 8: **RETURN** permutation π^1 , which is a single dominant permutation
for the problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with job set $\mathcal{J}_{1,2}$ **STOP 0**.

Step 9: **RETURN** permutation π^2 , which is a single dominant permutation
for the problem $F2|l_{ij} \leq p_{ij} \leq u_{ij}|C_{max}$ with job set $\mathcal{J}_{1,2}$ **STOP 0**.

Algorithm 2

Input: Lower bounds l_{ij} and upper bounds u_{ij} , $0 < l_{ij} \leq u_{ij}$, of the durations of all operations O_{ij} of jobs $J_i \in \mathcal{J}$ processed on machines $M_j \in \mathcal{M} = \{M_1, M_2\}$.

Output: **EITHER** pair of permutations $(\pi', \pi'') = ((\pi_{1,2}, \pi_1, \pi_{2,1}), (\pi_{2,1}, \pi_2, \pi_{1,2}))$, where π' is a permutation of jobs from set $\mathcal{J}_{1,2} \cup \mathcal{J}_1 \cup \mathcal{J}_{2,1}$ on machine M_1 , π'' is a permutation of jobs from set $\mathcal{J}_{1,2} \cup \mathcal{J}_2 \cup \mathcal{J}_{2,1}$ on machine M_2 , such that $\{(\pi', \pi'')\} = DS(T)$, (see STOP 0),
OR permutation $\pi_{2,1}$ of jobs from set $\mathcal{J}_{2,1}$ on machines M_1 and M_2 and a partial strict order $A_{\prec}^{1,2}$ of jobs from set $\mathcal{J}_{1,2}$,
OR permutation $\pi_{1,2}$ of jobs from set $\mathcal{J}_{1,2}$ on machines M_1 and M_2 and a partial strict order $A_{\prec}^{2,1}$ of jobs from set $\mathcal{J}_{2,1}$,
OR a partial strict order $A_{\prec}^{1,2}$ of jobs from set $\mathcal{J}_{1,2}$ and a partial strict order $A_{\prec}^{2,1}$ of jobs from set $\mathcal{J}_{2,1}$, (see STOP 1).

Step 1: Determine a partition $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{1,2} \cup \mathcal{J}_{2,1}$ of the job set \mathcal{J} , permutation π_1 of jobs from set \mathcal{J}_1 and permutation π_2 of jobs from set \mathcal{J}_2 , arrange the jobs in the increasing order of their indexes.

Step 2: **IF** the first inequality in condition (5) of Theorem 4 holds **THEN BEGIN**
Construct a permutation $\pi_{1,2}$ of jobs from set $\mathcal{J}_{1,2}$,
arrange them in the increasing order of their indexes;
IF the second inequality in condition (5) of Theorem 4 holds
THEN construct a permutation $\pi_{2,1}$ of jobs from set $\mathcal{J}_{2,1}$,
arrange them in the increasing order of their indexes **GOTO** Step 10 **END**

Step 3: **IF** the first inequality in condition (6) of Theorem 4 holds **THEN BEGIN**
Construct a permutation $\pi_{2,1}$ of jobs from set $\mathcal{J}_{2,1}$,
arrange them in the increasing order of their indexes;
IF the second inequality in condition (6) of Theorem 4 holds **THEN**
construct a permutation $\pi_{1,2}$ of jobs from set $\mathcal{J}_{1,2}$,
arrange the jobs in the increasing order of their indexes **END**

Step 4: **IF** both permutations $\pi_{1,2}$ and $\pi_{2,1}$ are constructed **THEN GOTO** Step 10.

Step 5: **IF** permutation $\pi_{1,2}$ is not constructed **THEN** fulfill Procedure 1.

Step 6: **IF** permutation $\pi_{2,1}$ is not constructed **THEN** fulfill Procedure 2.

Step 7: **IF** both permutations $\pi_{1,2}$ and $\pi_{2,1}$ are constructed **THEN GOTO** Step 10.

Step 8: **IF** permutation $\pi_{2,1}$ is constructed **THEN GOTO** Step 11.

Step 9: **IF** permutation $\pi_{1,2}$ is constructed **THEN GOTO** Step 12 **ELSE GOTO** Step 13.

Step 10: **RETURN** pair of permutations (π', π'') , where π' is the permutation of jobs from set $\mathcal{J}_{1,2} \cup \mathcal{J}_1 \cup \mathcal{J}_{2,1}$ processed on machine M_1 and π'' is the permutation of jobs from set $\mathcal{J}_{1,2} \cup \mathcal{J}_2 \cup \mathcal{J}_{2,1}$ processed on machine M_2 such that $\{(\pi', \pi'')\} = DS(T)$ **STOP 0**.

Step 11: **RETURN** the permutation $\pi_{2,1}$ of jobs from set $\mathcal{J}_{2,1}$ processed on machines M_1 and M_2 , the partial strict order $A_{\prec}^{1,2}$ of jobs from set $\mathcal{J}_{1,2}$ **GOTO** Step 14.

Step 12: **RETURN** the permutation $\pi_{1,2}$ of jobs from set $\mathcal{J}_{1,2}$ processed on machines M_1 and M_2 , the partial strict order $A_{\prec}^{2,1}$ of jobs from set $\mathcal{J}_{2,1}$ **GOTO** Step 14.

Step 13: **RETURN** the partial strict order $A_{\prec}^{1,2}$ of jobs from set $\mathcal{J}_{1,2}$ and the partial strict order $A_{\prec}^{2,1}$ of jobs from set $\mathcal{J}_{2,1}$

Step 14: **STOP 1**.

Appendix B. Tables with Computational Results

Table A1. Computational results for the randomly generated instances with the ratio 0%:0%:10%:90% of the numbers of jobs in the subsets \mathcal{I}_1 , \mathcal{I}_2 , $\mathcal{I}_{1,2}$ and $\mathcal{I}_{2,1}$ of the job set \mathcal{I} .

$\delta\%$	n	Uniform Distributions										Gamma Distributions									
		10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100
5	Stop1	93.9	97.3	98.4	97.6	96.6	97.6	97.9	98.3	98	97.5	93.4	97	97.7	97.1	98.5	98.3	97.6	98.6	98.3	98.7
	Stop2	0.9	1	0.6	1.1	1.6	1.3	1.2	1.1	0.9	1.4	0.8	0.8	0.9	1.1	0.5	0.5	1.3	0.8	0.9	0.6
	Stop3	5.1	1.7	1	1.3	1.8	1.1	0.9	0.6	1.1	1.1	5.3	2.2	1.4	1.8	1	1.2	1.1	0.6	0.8	0.7
	Stop4	0.1	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0
10	Stop1	85.8	91.8	91.6	93	92.2	92.2	91.9	92.6	93.6	93.8	83.1	93.5	92.8	92.7	93.1	92.3	92.4	93.3	94.2	93.6
	Stop2	2.1	2.5	4.5	3.5	4.1	4	4.2	4.6	4.2	4.2	3.2	2.8	3.1	3.5	3.6	4.3	4.2	3.4	3.5	4.1
	Stop3	11.7	5.7	3.9	3.5	3.7	3.8	3.9	2.8	2.2	2	12.8	3.7	4.1	3.8	3.3	3.4	3.4	3.3	2.3	2.3
	Stop4	0.4	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0	0	0	0	0	0
20	Stop1	70.6	73.2	73.3	71.9	69.6	63.4	63.6	57.7	56.3	53.4	70.2	73.7	74.1	71.5	67.2	64.2	63.4	61.5	56.1	57.5
	Stop2	4.3	6.8	9.4	8.6	8.8	11.1	9.4	10.1	9.2	9.2	4.3	8.1	8.7	10	10	11.3	9.1	9.1	8.9	7.5
	Stop3	24.2	20	17.3	19.5	21.6	25.5	27	32.2	34.5	37.4	24.9	18.2	17.2	18.5	22.8	24.5	27.5	29.4	35	35
	Stop4	0.9	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0
30	Stop1	53	55.1	47.5	41.1	31	30.7	23.4	20.3	18.5	13.4	52	52.2	44.9	34.9	33.6	26.3	27.3	18.3	18.9	14.2
	Stop2	4.8	8.8	9.1	9.7	9.8	7	6.7	5.2	4.2	4.4	6.2	8.9	10.1	10.4	11	7.9	6	4.6	5.9	4.5
	Stop3	41.9	36.1	43.4	49.2	59.2	62.3	69.9	74.5	77.3	82.2	41	38.9	45	54.7	55.4	65.8	66.7	77.1	75.2	81.3
	Stop4	0.3	0	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0
40	Stop1	41.6	32.5	24.7	16.4	10.9	8.9	5.9	3.9	2.1	2.3	37.8	31.8	22.2	18.6	10.6	8.5	6.1	4.3	3	1.9
	Stop2	4.9	7.6	7.7	6.6	4.2	2.9	2	1.1	0.7	0.3	5.2	8.5	9.1	4.8	2.9	3.5	2	1.6	1	0.6
	Stop3	52.7	59.7	67.6	77	84.9	88.2	92.1	95	97.2	97.4	55.3	59.7	68.7	76.6	86.5	88	91.9	94.1	96	97.5
	Stop4	0.8	0.2	0	0	0	0	0	0	0	0	1.7	0	0	0	0	0	0	0	0	0
50	Stop1	29	16.1	9.3	5	2.4	1.4	1.6	0.8	0.5	0	26.2	19.4	10.4	5.6	3.9	1.6	0.6	0.5	0.3	0.1
	Stop2	4.7	5.4	5	2.6	1.2	0.8	0.2	0.3	0.3	0	3.7	5.6	3.8	2.2	0.7	0.7	0.7	0.1	0.2	0.1
	Stop3	64.9	78.5	85.7	92.4	96.4	97.8	98.2	98.9	99.2	100	67.8	75	85.8	92.2	95.4	97.7	98.7	99.4	99.5	99.8
	Stop4	1.4	0	0	0	0	0	0	0	0	0	2.3	0	0	0	0	0	0	0	0	0
60	Stop1	18.1	8.4	4.6	1.4	1	0.3	0.2	0.1	0	0	16.6	9.6	3.6	1.4	1	0.3	0	0.2	0	0
	Stop2	3.8	4	1.6	0.7	0.3	0.1	0	0	0	0	3.2	3.5	2.3	0.9	0.4	0	0.3	0	0.1	0
	Stop3	76.4	87.5	93.8	97.9	98.7	99.6	99.8	99.9	100	100	77.7	86.9	94.1	97.7	98.6	99.7	99.7	99.8	99.9	100
	Stop4	1.7	0.1	0	0	0	0	0	0	0	0	2.5	0	0	0	0	0	0	0	0	0
70	Stop1	12.3	5.1	1.4	0.7	0.1	0	0	0	0	0	12.1	4.7	2.1	0.8	0.2	0	0.1	0	0	0
	Stop2	2.8	1.5	0.7	0.2	0.1	0	0	0	0	0	2	2.1	0.4	0.2	0	0	0	0	0	0
	Stop3	83.2	93.4	97.9	99.1	99.8	100	100	100	100	100	82.8	93.2	97.5	99	99.8	100	99.9	100	100	100
	Stop4	1.7	0	0	0	0	0	0	0	0	0	3.1	0	0	0	0	0	0	0	0	0
80	Stop1	8.9	1.9	0.4	0.3	0.2	0	0	0	0	0	10	2.3	0.6	0.2	0	0	0	0	0	0
	Stop2	1.8	0.8	0.5	0	0	0	0	0	0	0	2.2	1.1	0.2	0	0	0.1	0	0	0	0
	Stop3	87.1	97.1	99.1	99.7	99.8	100	100	100	100	100	84.2	96.5	99.2	99.8	100	99.9	100	100	100	100
	Stop4	2.2	0.2	0	0	0	0	0	0	0	0	3.6	0.1	0	0	0	0	0	0	0	0
90	Stop1	6.9	0.8	0.2	0.1	0	0	0	0	0	0	5.7	0.9	0.3	0	0	0	0	0	0	0
	Stop2	1.2	0.7	0.1	0	0	0	0	0	0	0	1.5	0.5	0	0	0	0	0	0	0	0
	Stop3	90.1	98.4	99.7	99.9	100	100	100	100	100	100	88	98.5	99.7	100	100	100	100	100	100	100
	Stop4	1.8	0.1	0	0	0	0	0	0	0	0	4.8	0.1	0	0	0	0	0	0	0	0
100	Stop1	4.4	0.3	0.1	0.1	0	0	0	0	0	0	4.1	0.3	0	0	0	0	0	0	0	0
	Stop2	1.1	0.2	0	0	0	0	0	0	0	0	0.8	0	0	0	0	0	0	0	0	0
	Stop3	92.1	99.4	99.9	99.9	100	100	100	100	100	100	90.2	99.4	100	100	100	100	100	100	100	100
	Stop4	2.4	0.1	0	0	0	0	0	0	0	0	4.9	0.3	0	0	0	0	0	0	0	0

Table A2. Computational results for randomly generated instances with the ratio 0%:0%:20%:80% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

		Uniform Distributions										Gamma Distributions									
$\delta\%$	n	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100
5	Stop1	96.7	98.9	98.6	99.5	99.7	99.7	99.9	99.9	99.9	99.9	96.7	98.9	99	99.3	99.6	99.4	99.7	99.8	99.9	99.8
	Stop2	0.8	0.3	0.7	0.1	0	0.2	0.1	0.1	0.1	0.1	0.3	0.4	0.1	0.3	0.1	0.4	0.2	0.1	0.1	0.2
	Stop3	2.4	0.8	0.7	0.4	0.3	0.1	0	0	0	0	3	0.7	0.9	0.4	0.3	0.2	0.1	0.1	0	0
	Stop4	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	Stop1	93.4	96.2	97.6	97.7	98	98.4	98.4	98.7	98.5	99.6	93.5	96.1	97.5	97.6	97.8	98.1	98.4	98.2	99	98.7
	Stop2	1	1.4	1.4	1.4	1.4	1.3	1.1	1.2	1	0.4	1	1.8	1.1	1.3	1.5	1.6	1.3	1.5	0.9	1.1
	Stop3	5.4	2.4	1	0.9	0.6	0.3	0.5	0.1	0.5	0	5.5	2.1	1.4	1.1	0.7	0.3	0.3	0.3	0.1	0.2
	Stop4	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	Stop1	85.4	86.9	89.5	90.2	90.8	90.5	89.1	90	90.3	90.1	83.5	87.5	88.8	88.4	90.6	89.9	91	89.4	90.9	90.1
	Stop2	2.6	4.8	5.3	5	5.8	5.8	5.7	5	5	5.4	3.1	5.7	5.8	7.4	4.7	5.8	4.9	6.5	6.2	4
	Stop3	12	8.3	5.2	4.8	3.4	3.7	5.2	5	4.7	4.5	13.4	6.8	5.4	4.2	4.7	4.3	4.1	4.1	2.9	5.9
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	Stop1	70.2	76.4	70.1	69.6	69.2	65.7	64.1	60.4	58	56	71.6	73.3	72.9	69	66	63.1	60.4	62.3	62.2	58.1
	Stop2	5.1	6.8	9.7	9.5	8.4	10.5	9.7	9.1	10.8	8.1	3.6	8.7	8.4	9.6	9.8	11.3	8.4	8.3	7.1	8.9
	Stop3	24.6	16.8	20.2	20.9	22.4	23.8	26.2	30.5	31.2	35.9	24.8	18	18.7	21.4	24.2	25.6	31.2	29.4	30.7	33
	Stop4	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	Stop1	56.7	55.3	48.8	41.5	39.6	31.2	31.1	26.5	23.5	21.5	55.1	53.9	46.5	41.5	37.5	31.6	33.2	27.4	26.2	21
	Stop2	5.1	8.2	10.5	10.6	9.6	8.2	7.6	6.9	7.5	5	5.6	9.2	11.5	7.7	8.9	8.7	7.3	7.2	6	5.7
	Stop3	38.2	36.5	40.7	47.9	50.8	60.6	61.3	66.6	69	73.5	39.3	36.9	42	50.8	53.6	59.7	59.5	65.4	67.8	73.3
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	Stop1	42.6	35.2	27.2	22.9	14.3	11.4	9.9	6.9	6.6	5.9	43.9	37.3	29.2	22	16.3	13.2	10.8	10.5	5.5	5.7
	Stop2	4.4	9.5	9.8	6.5	6	5.2	3.8	3.4	2.2	2.4	6	10	7.3	6.9	6.5	4.8	4.7	2.8	1.9	2.5
	Stop3	52.8	55.3	63	70.6	79.7	83.4	86.3	89.7	91.2	91.7	49.9	52.7	63.5	71.1	77.2	82	84.5	86.7	92.6	91.8
	Stop4	0.2	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0
60	Stop1	32.9	20.7	14.4	10.7	6.3	3.3	3.3	2.2	1.4	0.8	31.7	23.1	15.7	10.4	7.2	3.9	2.9	1.4	0.9	0.6
	Stop2	5.4	5.5	4.2	2.6	2.3	1.7	0.8	0.6	0.3	0.4	5.6	6	6.4	3.6	2.4	1.5	1.5	0.8	0.3	0.7
	Stop3	61.7	73.8	81.4	86.7	91.4	95	95.9	97.2	98.3	98.8	62.3	70.9	77.9	86	90.4	94.6	95.6	97.8	98.8	98.7
	Stop4	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0
70	Stop1	22.1	13.7	7.6	2.9	1.6	1.1	0.6	0.2	0.3	0.1	23.9	13.1	6.7	3.4	1.6	0.7	0.9	0.3	0.2	0.1
	Stop2	4.9	3.7	2.9	1.7	0.9	0.6	0.4	0.1	0	0.1	4.4	4.5	2	1.5	1.2	0.4	0	0.4	0.1	0.1
	Stop3	72.8	82.6	89.5	95.4	97.5	98.3	99	99.7	99.7	99.8	71.6	82.4	91.3	95.1	97.2	98.9	99.1	99.3	99.7	99.8
	Stop4	0.2	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0
80	Stop1	17.3	8.3	3.5	1.3	0.5	0.3	0.2	0	0	0	16.4	7.9	3.1	1.9	0.9	0.2	0	0	0	0
	Stop2	2.8	1.7	1.8	0.8	0.6	0	0	0	0	0	3.4	2.4	0.8	0.9	0.1	0.2	0.1	0	0	0
	Stop3	79.4	90	94.7	97.9	98.9	99.7	99.8	100	100	100	79.6	89.7	96.1	97.2	99	99.6	99.9	100	100	100
	Stop4	0.5	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0
90	Stop1	12.1	4.5	1.4	0.7	0	0.1	0	0	0	0	12.6	5	2.9	0.4	0.1	0.1	0	0	0	0
	Stop2	2.5	2.3	1.3	0.3	0	0.1	0	0	0	0	3.2	1.8	0.2	0.2	0	0	0	0	0	0
	Stop3	85.2	93.2	97.3	99	100	99.8	100	100	100	100	83.7	93.2	96.9	99.4	99.9	99.9	100	100	100	100
	Stop4	0.2	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0
100	Stop1	10.6	2.7	0.5	0.3	0	0	0	0	0	0	9.1	2.8	0.5	0.1	0	0	0	0	0	0
	Stop2	1.9	1.2	0.3	0	0.2	0	0	0	0	0	2	0.6	0.2	0	0	0	0	0	0	0
	Stop3	87.4	96.1	99.2	99.7	99.8	100	100	100	100	100	88.2	96.6	99.3	99.9	100	100	100	100	100	100
	Stop4	0.1	0	0	0	0	0	0	0	0	0	0.7	0	0	0	0	0	0	0	0	0

Table A3. Computational results for randomly generated instances with the ratio 0%:0%:30%:70% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

[illegible]

Table A5. Computational results for randomly generated instances with the ratio 0%:0%:50%:50% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

		Uniform Distributions										Gamma Distributions									
$\delta\%$	n	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100
5	Stop1	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	Stop2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stop3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	Stop1	99.8	100	100	100	100	100	100	100	100	100	99.9	100	100	100	100	100	100	100	100	100
	Stop2	0.1	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0
	Stop3	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	Stop1	99.1	99.6	100	100	100	100	100	100	100	100	98.9	99.9	100	100	100	100	100	100	100	100
	Stop2	0.3	0.2	0	0	0	0	0	0	0	0	0.5	0.1	0	0	0	0	0	0	0	0
	Stop3	0.6	0.2	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	Stop1	96.2	99.5	99.6	99.8	99.9	100	100	100	100	100	96.1	98.7	99.8	100	100	100	100	100	100	100
	Stop2	1.5	0.5	0.4	0.2	0.1	0	0	0	0	0	1.1	1.1	0.1	0	0	0	0	0	0	0
	Stop3	2.3	0	0	0	0	0	0	0	0	0	2.8	0.2	0.1	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	Stop1	91.3	97.4	99.2	99.2	99.6	99.9	100	100	100	100	91.4	96.7	98.9	99.1	99.7	99.6	99.9	100	100	100
	Stop2	3	1	0.2	0.5	0.1	0.1	0	0	0	0	2.1	1.7	0.6	0.7	0.3	0.2	0.1	0	0	0
	Stop3	5.7	1.6	0.6	0.3	0.3	0	0	0	0	0	6.5	1.6	0.5	0.2	0	0.2	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	Stop1	83.4	92.3	95.3	97.9	98.2	98.8	99.3	99.9	99.8	100	81.7	90	95.5	97.5	98.7	98.9	99.4	99.8	99.9	99.9
	Stop2	3.5	3.3	1.9	1.1	0.9	0.6	0.1	0.1	0.1	0	3.8	4.6	2.1	1.3	0.5	0.5	0.4	0	0	0
	Stop3	13.1	4.4	2.8	1	0.9	0.6	0.6	0	0.1	0	14.5	5.4	2.4	1.2	0.8	0.6	0.2	0.2	0.1	0.1
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	Stop1	69.4	83.1	87.1	93.8	94.5	96.1	96.4	97.8	98.6	98.3	71.2	81.5	89.2	92.6	93.5	95.1	98	98.2	98.5	98.9
	Stop2	5.7	6.7	5	2.5	2.2	1.3	0.8	0.7	0.4	0.6	6.5	6.8	3.3	3.3	2.3	2.1	0.5	0.8	0.7	0.1
	Stop3	24.9	10.2	7.9	3.7	3.3	2.6	2.8	1.5	1	1.1	22.3	11.7	7.5	4.1	4.2	2.8	1.5	1	0.8	1
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	Stop1	59.2	70.2	75.5	81.1	81.6	88.1	87.9	92.4	92.8	93.6	58.4	72.7	74.8	81.5	85.8	87.8	88	91	90.3	93.7
	Stop2	5.8	7.4	6.6	6	5.2	4.1	3.7	1.7	1.1	2.4	4.3	8.3	7.6	4.4	3.6	3.2	4.1	2.4	2.6	2
	Stop3	35	22.4	17.9	12.9	13.2	7.8	8.4	5.9	6.1	4	37.3	19	17.6	14.1	10.6	9	7.9	6.6	7.1	4.3
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	Stop1	45	57.2	61.8	62.7	67.8	68.7	71	74.9	75.5	77	46.8	54.8	61.1	64.1	69.4	71	71.4	73.8	75.8	76
	Stop2	7	8.5	7.2	7.4	7.5	6	6	5.2	4.2	3.9	5.5	7.5	7	7.7	6.4	5.3	6.4	4.5	4.4	3.5
	Stop3	48	34.3	31	29.9	24.7	25.3	23	19.9	20.3	19.1	47.7	37.7	31.9	28.2	24.2	23.7	22.2	21.7	19.8	20.5
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90	Stop1	35.2	41.2	38.7	44.2	42.5	45.6	46.2	45.2	50.3	50	37.2	37.3	41.3	43.8	43.3	46.5	48	46.7	47.5	49.4
	Stop2	6	8.3	9.8	7.7	7.8	7.4	6.4	6.2	6.4	5.5	5.4	8.7	9.8	8.9	9.3	9.2	6.4	6.9	6.2	5.7
	Stop3	58.8	50.5	51.5	48.1	49.7	47	47.4	48.6	43.3	44.5	57.4	54	48.9	47.3	47.4	44.3	45.6	46.4	46.3	44.9
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	Stop1	25.7	28.8	25.2	22.6	21.4	21.4	20.2	20.5	19	22.1	26.5	26	25.2	25.5	25.1	22.1	22.8	19.9	23.8	20.7
	Stop2	4.1	8.6	8.1	6.9	7.9	5.9	6.6	5.4	5.5	4.3	3.7	9.2	7.8	6	7.2	7.1	5.4	5.1	4.8	4.9
	Stop3	70.2	62.6	66.7	70.5	70.7	72.7	73.2	74.1	75.5	73.6	69.8	64.8	67	68.5	67.7	70.8	71.8	75	71.4	74.4
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A6. Computational results for randomly generated instances with the ratio 5%:5%:5%:85% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

$\delta\%$	n	Uniform Distributions					Gamma Distributions				
		20	40	60	80	100	20	40	60	80	100
5	Stop1	99	99.5	99.7	99.3	99.4	98.1	99.7	99.8	99.2	99.7
	Stop2	0.3	0.3	0.2	0.4	0.6	0.4	0.2	0	0.6	0.3
	Stop3	0.7	0.2	0.1	0.3	0	1.5	0.1	0.2	0.2	0
	Stop4	0	0	0	0	0	0	0	0	0	0
10	Stop1	96.4	97.4	98.4	97.8	97.1	95.4	97.6	97.7	98	98.6
	Stop2	0.9	1.1	0.7	1.2	0.7	1.5	1.4	1	1.3	0.3
	Stop3	2.7	1.5	0.9	1	2.2	3.1	1	1.3	0.7	1.1
	Stop4	0	0	0	0	0	0	0	0	0	0
20	Stop1	84.3	81.4	76.5	71.7	67.9	86.5	80.5	73.8	71	66.5
	Stop2	3.1	3.5	3.7	2.5	2.1	2.9	3.5	2.9	3.7	1.7
	Stop3	12.6	15.1	19.8	25.8	30	10.6	16	23.3	25.3	31.8
	Stop4	0	0	0	0	0	0	0	0	0	0
30	Stop1	61.8	52.9	38.4	27.1	20.9	65.1	53	40	29.3	20.3
	Stop2	5.7	2.9	2.1	2.1	0.3	3.4	2.6	1.4	1.4	1.2
	Stop3	32.5	44.2	59.5	70.8	78.8	31.4	44.4	58.6	69.3	78.5
	Stop4	0	0	0	0	0	0.1	0	0	0	0
40	Stop1	38.6	25.5	14.1	7.3	3.1	40	24.4	13.6	6.5	3.2
	Stop2	5.4	1.2	1.2	0.4	0.3	4.5	2.4	0.4	0.4	0.1
	Stop3	56	73.3	84.7	92.3	96.6	55.5	73.2	86	93.1	96.7
	Stop4	0	0	0	0	0	0	0	0	0	0
50	Stop1	27.4	8.4	3.2	1.7	0.3	25.9	10.9	3.3	0.9	0.3
	Stop2	2.3	1.4	0.3	0	0	3.7	1.1	0	0	0
	Stop3	70.2	90.2	96.5	98.3	99.7	70.4	88	96.7	99.1	99.7
	Stop4	0.1	0	0	0	0	0	0	0	0	0
60	Stop1	14.3	3.4	0.5	0.2	0	15.7	3.2	0.6	0.3	0
	Stop2	2.3	0.4	0	0	0	1.8	0.1	0.3	0	0
	Stop3	83.4	96.2	99.5	99.8	100	82.4	96.7	99.1	99.7	100
	Stop4	0	0	0	0	0	0.1	0	0	0	0
70	Stop1	8	1.1	0.3	0	0	6.5	1.7	0.1	0	0
	Stop2	0.9	0	0	0	0	1.5	0	0.1	0	0
	Stop3	91.1	98.9	99.7	100	100	91.8	98.3	99.8	100	100
	Stop4	0	0	0	0	0	0.2	0	0	0	0
80	Stop1	4	0.1	0	0.1	0	5.1	0.2	0	0	0
	Stop2	0.5	0.1	0	0	0	0.9	0	0	0	0
	Stop3	95.5	99.8	100	99.9	100	93.8	99.8	100	100	100
	Stop4	0	0	0	0	0	0.2	0	0	0	0
90	Stop1	2.6	0.3	0	0	0	2.2	0	0	0	0
	Stop2	0.3	0	0	0	0	0.4	0	0	0	0
	Stop3	97.1	99.7	100	100	100	97	100	100	100	100
	Stop4	0	0	0	0	0	0.4	0	0	0	0
100	Stop1	0.9	0	0	0	0	1.5	0	0	0	0
	Stop2	0.3	0	0	0	0	0	0	0	0	0
	Stop3	98.8	100	100	100	100	98.4	100	100	100	100
	Stop4	0	0	0	0	0	0.1	0	0	0	0

Table A8. Computational results for randomly generated instances with the ratio 5%:20%:5%:70% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

[illegible]

Table A9. Computational results for randomly generated instances with the ratio 10%:10%:10%:70% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

[illegible]

Table A10. Computational results for randomly generated instances with the ratio 10%:10%:40%:40% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

		Uniform Distributions										Gamma Distributions									
$\delta\%$	n	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100
5	Stop1	99.9	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	Stop2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stop3	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	Stop1	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	Stop2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stop3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	Stop1	99.3	100	100	100	100	100	100	100	100	100	99.5	100	100	100	100	100	100	100	100	100
	Stop2	0.2	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0
	Stop3	0.5	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	Stop1	98.4	99.9	100	100	100	100	100	100	100	100	99	99.9	100	100	100	100	100	100	100	100
	Stop2	0.5	0.1	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0
	Stop3	1.1	0	0	0	0	0	0	0	0	0	0.9	0.1	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	Stop1	97	99.8	99.9	100	100	100	100	100	100	100	96.4	99.8	100	100	100	100	100	100	100	100
	Stop2	0.1	0	0.1	0	0	0	0	0	0	0	0.5	0.2	0	0	0	0	0	0	0	0
	Stop3	2.9	0.2	0	0	0	0	0	0	0	0	3.1	0	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	Stop1	93	99	99.9	99.9	100	100	100	100	100	100	92.8	99.6	99.9	100	100	100	100	100	100	100
	Stop2	1.2	0.5	0.1	0	0	0	0	0	0	0	0.9	0.1	0	0	0	0	0	0	0	0
	Stop3	5.8	0.5	0	0.1	0	0	0	0	0	0	6.3	0.3	0.1	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	Stop1	86.6	97.9	99.6	99.5	99.8	99.8	99.8	99.9	100	100	85.8	97.1	99.3	99.6	99.8	99.7	99.9	100	100	100
	Stop2	1.4	0.5	0	0.2	0	0.1	0	0	0	0	1.5	0.4	0	0	0	0	0	0	0	0
	Stop3	12	1.6	0.4	0.3	0.2	0.1	0.2	0.1	0	0	12.7	2.5	0.7	0.4	0.2	0.3	0.1	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	Stop1	82.3	92.5	96.7	97.2	98.6	98.1	99	99.2	99.5	99.7	80	92.7	96.5	97.5	97.9	99.2	99.5	99.5	99.8	99.5
	Stop2	1.8	1.1	0.2	0.2	0	0	0.1	0	0	0.1	1.7	0.5	0	0.2	0.1	0	0	0	0	0
	Stop3	15.9	6.4	3.1	2.6	1.4	1.9	0.9	0.8	0.5	0.2	18.3	6.8	3.5	2.3	2	0.8	0.5	0.5	0.2	0.5
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	Stop1	71.8	85.9	91.5	93.8	94.4	95.3	95.6	97.7	97.3	98.1	73.1	87.2	91	93.1	95.1	95.9	96.7	97.3	98.1	98.8
	Stop2	1.9	1.2	0.6	0.2	0.5	0	0	0.1	0	0.1	2	0.9	0.2	0.7	0.1	0.2	0	0.3	0	0
	Stop3	26.3	12.9	7.9	6	5.1	4.7	4.4	2.2	2.7	1.8	24.9	11.9	8.8	6.2	4.8	3.9	3.3	2.4	1.9	1.2
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90	Stop1	63.2	77.8	81.6	84.2	85	88.7	89.9	91.5	93.6	93.6	61.7	77.2	83.3	85	86.4	88	90.4	91.4	92.7	94.4
	Stop2	2.6	1	0.8	1.1	0.2	0.2	0.6	0.2	0.1	0.2	2.6	1.9	0.9	0.2	0.5	0.2	0.1	0.1	0	0.1
	Stop3	34.2	21.2	17.6	14.7	14.8	11.1	9.5	8.3	6.3	6.2	35.7	20.9	15.8	14.8	13.1	11.8	9.5	8.5	7.3	5.5
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	Stop1	53.1	66.7	68.3	70.3	74.1	74.1	75.6	78.3	81	84.1	51.8	67	69.8	71	72.5	75.9	75.7	79.6	79.8	80.3
	Stop2	2.1	1.7	0.6	0.8	0.3	0.7	0.3	0.2	0.1	0.3	2.4	0.9	1	0.7	1.1	0.5	0.3	0.6	0.3	0.5
	Stop3	44.8	31.6	31.1	28.9	25.6	25.2	24.1	21.5	18.9	15.6	45.8	32.1	29.2	28.3	26.4	23.6	24	19.8	19.9	19.2
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A11. Computational results for randomly generated instances with the ratio 10%:20%:10%:60% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

[illegible]

Table A12. Computational results for randomly generated instances with the ratio 10%:30%:10%:50% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

[illegible]

Table A13. Computational results for randomly generated instances with the ratio 10%:40%:10%:40% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

		Uniform Distributions										Gamma Distributions									
$\delta\%$	n	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100
5	Stop1	98.9	100	100	100	100	100	100	100	100	100	99.6	99.9	100	100	100	100	100	100	100	100
	Stop2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stop3	1.1	0	0	0	0	0	0	0	0	0	0.4	0.1	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	Stop1	97.6	99.6	99.9	99.9	100	100	100	100	100	100	98.2	99.5	99.9	99.9	100	100	100	99.9	100	100
	Stop2	0	0	0.1	0.1	0	0	0	0	0	0	0	0	0.1	0.1	0	0	0	0.1	0	0
	Stop3	2.4	0.4	0	0	0	0	0	0	0	0	1.8	0.5	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	Stop1	95.8	98.6	99	99.7	99.7	99.9	99.7	100	100	99.9	95.5	97.6	99.7	99.3	99.5	99.8	100	99.8	99.9	100
	Stop2	0	0.4	0.4	0.1	0.3	0.1	0.2	0	0	0.1	0.1	0.4	0	0.5	0.2	0.2	0	0.1	0.1	0
	Stop3	4.2	1	0.6	0.2	0	0	0.1	0	0	0	4.4	2	0.3	0.2	0.3	0	0	0.1	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	Stop1	93.2	94.8	97.1	97.8	98.1	98.9	99	99	98.5	99.3	93.4	94.9	96.8	98.3	98.5	97.4	98.8	99.4	99.1	99
	Stop2	0	0.8	0.5	0.9	0.2	0.1	0	0.4	0.5	0.1	0.5	1	0.7	0.3	0.4	0.9	0.2	0.1	0.5	0
	Stop3	6.8	4.4	2.4	1.3	1.7	1	1	0.6	1	0.6	6.1	4.1	2.5	1.4	1.1	1.7	1	0.5	0.4	1
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	Stop1	88.6	91.1	91	91.9	94.5	91.2	91.7	92	93.3	93	87.7	90.5	91.6	92	92.5	93	93.1	92.6	92	91.5
	Stop2	0.3	1.4	1.1	0.7	0.2	0.8	0.9	0.7	0.9	0.5	0.2	1.7	1	1.3	0.8	0.4	0.9	0.6	0.4	0.7
	Stop3	11.1	7.5	7.9	7.4	5.3	8	7.4	7.3	5.8	6.5	12.1	7.8	7.4	6.7	6.7	6.6	6	6.8	7.6	7.8
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	Stop1	84.6	86.9	84.6	84.8	81	82	79.3	79.7	77.8	77.8	85	84.9	82	81.6	82	83.1	77.2	79.7	78.5	78.1
	Stop2	0.6	0.9	1.2	1.6	1.6	1.1	1	0.8	1.1	0.9	0	2.2	1.8	2	1	1.2	1.3	0.9	1.1	1
	Stop3	14.8	12.2	14.2	13.6	17.4	16.9	19.7	19.5	21.1	21.3	15	12.9	16.2	16.4	17	15.7	21.5	19.4	20.4	20.9
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	Stop1	79.7	77.8	75.4	71.5	71.1	69.2	63.5	61.3	59	58.8	77.5	75.1	75.4	71.8	70.7	66.4	66.3	63.5	60.1	60.3
	Stop2	0.2	1.9	2	1.9	1.9	1.2	0.8	1.1	0.9	0.6	0.8	1.7	2.1	1.5	1.4	1.8	0.5	0.8	1.7	1
	Stop3	20.1	20.3	22.6	26.6	27	29.6	35.7	37.6	40.1	40.6	21.7	23.2	22.5	26.7	27.9	31.8	33.2	35.7	38.2	38.7
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	Stop1	73.5	68.5	65.8	65.9	57.7	54.2	49.6	47.9	44.8	45.3	75.4	71.4	65.7	62.8	59.4	55.9	50.7	49.5	44.6	43.2
	Stop2	0.6	1.7	1.6	1.4	1.5	0.7	0.9	0.7	0.7	0.8	0.2	1.5	1.2	1.2	1.4	0.7	1	0.7	1	0.7
	Stop3	25.9	29.8	32.6	32.7	40.8	45.1	49.5	51.4	54.5	53.9	24.3	27.1	33.1	36	39.2	43.4	48.3	49.8	54.4	56.1
	Stop4	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0
80	Stop1	67.6	60.5	58.7	52.8	46.1	43.5	40.1	36.7	35.4	34.2	66.6	62.4	53.4	50.8	49	43	41.4	37.8	38.7	33.7
	Stop2	0.3	2.7	1.2	1.1	0.9	0.6	0.8	0.4	0.2	0.6	0.4	2.1	2	1.1	1.4	1	0.4	0.3	0.3	0.2
	Stop3	32.1	36.8	40.1	46.1	53	55.9	59.1	62.9	64.4	65.2	32.9	35.5	44.6	48.1	49.6	56	58.2	61.9	61	66.1
	Stop4	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0
90	Stop1	63.3	50.9	49.5	42.8	36.3	36.2	33.7	28.7	28.2	28.9	58.2	54.1	51.9	44.1	37.3	33.8	31.6	30.8	27	27.5
	Stop2	0.8	1.5	1.7	0.9	0.5	0.1	0.3	0.2	0.2	0.3	0.3	1.4	1.2	1	0.7	0.5	0.4	0.1	0.3	0
	Stop3	35.9	47.6	48.8	56.3	63.2	63.7	66	71.1	71.6	70.8	41.5	44.5	46.9	54.9	62	65.7	68	69.1	72.7	72.5
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	Stop1	58.1	48.4	41.1	32.5	30.9	27.2	24.2	21.2	23.1	20.4	55.5	44.4	38.2	32.2	32.3	27.7	24.1	21.8	22.4	20.3
	Stop2	0.2	1	0.6	0.8	0.3	0.4	0.2	0.4	0	0	0.7	1.5	0.7	0.8	0.6	0.5	0.2	0.1	0.2	0
	Stop3	41.6	50.6	58.3	66.7	68.8	72.4	75.6	78.4	76.9	79.6	43.8	54.1	61.1	67	67.1	71.8	75.7	78.1	77.4	79.7
	Stop4	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A14. Computational results for randomly generated instances with the ratio 10%:60%:10%:20% of the numbers of jobs in the subsets \mathcal{J}_1 , \mathcal{J}_2 , $\mathcal{J}_{1,2}$ and $\mathcal{J}_{2,1}$ of the job set \mathcal{J} .

		Uniform Distributions										Gamma Distributions									
$\delta\%$	n	10	20	30	40	50	60	70	80	90	100	10	20	30	40	50	60	70	80	90	100
5	Stop1	100	99.8	100	100	100	100	100	100	100	100	100	99.9	100	100	100	100	100	100	100	100
	Stop2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stop3	0	0.2	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	Stop1	100	99.9	100	100	100	100	100	100	100	100	100	99.8	99.9	100	100	100	100	100	100	100
	Stop2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Stop3	0	0.1	0	0	0	0	0	0	0	0	0	0.2	0.1	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	Stop1	100	99.4	99.9	100	100	100	100	100	100	100	100	99.3	99.9	100	100	100	100	100	100	100
	Stop2	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0
	Stop3	0	0.6	0.1	0	0	0	0	0	0	0	0	0.7	0	0	0	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	Stop1	99.9	98.7	99.7	99.9	100	100	100	100	100	100	100	98.6	99.5	99.5	99.9	100	100	100	100	100
	Stop2	0	0.3	0	0	0	0	0	0	0	0	0	0.2	0.1	0.3	0	0	0	0	0	0
	Stop3	0.1	1	0.3	0.1	0	0	0	0	0	0	0	1.2	0.4	0.2	0.1	0	0	0	0	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	Stop1	100	97.3	99.5	99.7	99.7	100	99.8	100	99.9	99.9	100	97.2	98.6	99.1	99.8	100	100	100	99.9	100
	Stop2	0	0.3	0.3	0	0	0	0.1	0	0	0	0	0.1	0.5	0.2	0	0	0	0	0	0
	Stop3	0	2.4	0.2	0.3	0.3	0	0.1	0	0.1	0.1	0	2.7	0.9	0.7	0.2	0	0	0	0.1	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	Stop1	99.9	96.1	97	98.3	99.2	99	99.3	99.5	99.8	100	100	95.8	96.9	98.4	99.2	99.2	99.7	99.9	99.6	100
	Stop2	0	0.2	0.5	0.2	0.3	0.5	0.2	0	0.2	0	0	0.2	0.5	0.4	0	0	0.1	0	0	0
	Stop3	0.1	3.7	2.5	1.5	0.5	0.5	0.5	0.5	0	0	0	4	2.6	1.2	0.8	0.8	0.2	0.1	0.4	0
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	Stop1	99.7	94.6	96.1	96.8	97.8	98.2	97.9	98.3	98.7	99	99.8	94.9	95.3	97.6	97.9	98.3	96.9	98.8	98.4	98.6
	Stop2	0	0.4	1.1	0.1	0.3	0.2	0.7	0	0.1	0.1	0	0.1	0.2	0.3	0.2	0.3	0.6	0.3	0.4	0.2
	Stop3	0.3	5	2.8	3.1	1.9	1.6	1.4	1.7	1.2	0.9	0.2	5	4.5	2.1	1.9	1.4	2.5	0.9	1.2	1.2
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	Stop1	99.6	94.4	94.2	94.8	95.9	94.6	95.6	95.8	96.5	96.7	99.6	94	94.9	93.4	94.5	96	97	96.2	96	96
	Stop2	0	0.4	0.6	0.5	0.6	0.6	0.3	0.4	0.4	0	0	0.2	0.7	0.5	0.4	0.8	0.2	0.2	0.4	0.3
	Stop3	0.4	5.2	5.2	4.7	3.5	4.8	4.1	3.8	3.1	3.3	0.4	5.8	4.4	6.1	5.1	3.2	2.8	3.6	3.6	3.7
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	Stop1	99.8	90.9	92	91.2	91.2	92	92	92.3	91.9	92	99.1	89.5	92	91.6	90.8	92.6	92.4	93.4	92.6	92.1
	Stop2	0	0.2	0.6	0.8	0.7	0.2	0.5	0.6	0.4	0.3	0	0.4	0.6	0.7	1.1	0.6	0.5	0.2	0.4	0.1
	Stop3	0.2	8.9	7.4	8	8.1	7.8	7.5	7.1	7.7	7.7	0.9	10.1	7.4	7.7	8.1	6.8	7.1	6.4	7	7.8
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90	Stop1	98.9	88.3	89.5	88.3	87.6	89	86.1	87.8	86.9	86.2	98.6	89.9	87.3	88.8	88.5	88.7	87.8	85.1	88.5	87.2
	Stop2	0	0.2	1	0.9	1.2	0.5	0.8	0.6	0.2	0.3	0	0.1	0.5	0.3	0.4	0.5	0.7	0.4	0.6	0.9
	Stop3	1.1	11.5	9.5	10.8	11.2	10.5	13.1	11.6	12.9	13.5	1.4	10	12.2	10.9	11.1	10.8	11.5	14.5	10.9	11.9
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	Stop1	97.8	88.4	86.6	83.1	85.7	83.4	82.8	81.3	82.1	82.4	97.6	90.2	84.4	85	84.2	81.7	80.2	82.2	80.7	83.2
	Stop2	0	0.4	1.1	1	1	0.6	0.7	0.5	0.8	0.2	0	0.2	0.7	0.9	1.2	0.6	0.6	0.5	0.4	0.5
	Stop3	2.2	11.2	12.3	15.9	13.3	16	16.5	18.2	17.1	17.4	2.4	9.6	14.9	14.1	14.6	17.7	19.2	17.3	18.9	16.3
	Stop4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

References

1. Lai, T.-C.; Sotskov, Yu.N. Sequencing with uncertain numerical data for makespan minimization. *J. Oper. Res. Soc.* **1999**, *50*, 230–243. [[CrossRef](#)]
2. Lai, T.-C.; Sotskov, Yu.N.; Sotskova, N.Y.; Werner, F. Mean flow time minimization with given bounds of processing times. *Eur. J. Oper. Res.* **2004**, *159*, 558–573. [[CrossRef](#)]
3. Sotskov, Yu.N.; Egorova, N.M.; Lai, T.-C. Minimizing total weighted flow time of a set of jobs with interval processing times. *Math. Comput. Model.* **2009**, *50*, 556–573. [[CrossRef](#)]
4. Lai, T.-C.; Sotskov, Yu.N.; Sotskova, N.Y.; Werner, F. Optimal makespan scheduling with given bounds of processing times. *Math. Comput. Model.* **1997**, *26*, 67–86.
5. Cheng, T.C.E.; Shakhlevich, N.V. Proportionate flow shop with controllable processing times. *J. Sched.* **1999**, *27*, 253–265. [[CrossRef](#)]
6. Cheng, T.C.E.; Kovalyov, M.Y.; Shakhlevich, N.V. Scheduling with controllable release dates and processing times: Makespan minimization. *Eur. J. Oper. Res.* **2006**, *175*, 751–768. [[CrossRef](#)]
7. Jansen, K.; Mastrolilli, M.; Solis-Oba, R. Approximation schemes for job shop scheduling problems with controllable processing times. *Eur. J. Oper. Res.* **2005**, *167*, 297–319. [[CrossRef](#)]
8. Sotskov, Y.N.; Matsveichuk, N.M.; Hatsura V.D. Two-machine job-shop scheduling problem to minimize the makespan with uncertain job durations. *Algorithms* **2020**, *13*, 4. [[CrossRef](#)]
9. Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Rinnooy Kan, A.H.G. Optimization and approximation in deterministic sequencing and scheduling. *Ann. Discret. Appl. Math.* **1979**, *5*, 287–326.
10. Pinedo, M. *Scheduling: Theory, Algorithms, and Systems*; Prentice-Hall: Englewood Cliffs, NJ, USA, 2002.
11. Sotskov, Yu.N.; Werner, F. *Sequencing and Scheduling with Inaccurate Data*; Nova Science Publishers: Hauppauge, NY, USA, 2014.
12. Tanaev, V.S.; Sotskov, Yu.N.; Strusevich, V.A. *Scheduling Theory: Multi-Stage Systems*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1994.
13. Jackson, J.R. An extension of Johnson's results on job lot scheduling. *Nav. Res. Logist. Q.* **1956**, *3*, 201–203. [[CrossRef](#)]
14. Johnson, S.M. Optimal two and three stage production schedules with set up times included. *Nav. Res. Logist. Q.* **1954**, *1*, 61–68. [[CrossRef](#)]
15. Gonzalez-Neira, E.M.; Montoya-Torres, J.R.; Barrera, D. Flow shop scheduling problem under uncertainties: Review and trends. *Int. J. Ind. Engin. Comput.* **2017**, *8*, 399–426. [[CrossRef](#)]
16. Elmaghraby, S.; Thoney, K.A. Two-machine flowshop problem with arbitrary processing time distributions. *IIE Trans.* **2000**, *31*, 467–477.
17. Kamburowski, J. Stochastically minimizing the makespan in two-machine flow shops without blocking. *Eur. J. Oper. Res.* **1999**, *112*, 304–309. [[CrossRef](#)]
18. Ku, P.S.; Niu, S.C. On Johnson's two-machine flow-shop with random processing times. *Oper. Res.* **1986**, *34*, 130–136. [[CrossRef](#)]
19. Allahverdi, A. Stochastically minimizing total flowtime in flowshops with no waiting space. *Eur. J. Oper. Res.* **1999**, *113*, 101–112. [[CrossRef](#)]
20. Allahverdi, A.; Mittenthal, J. Two-machine ordered flowshop scheduling under random breakdowns. *Math. Comput. Model.* **1999**, *20*, 9–17. [[CrossRef](#)]
21. Portougal, V.; Trietsch, D. Johnson's problem with stochastic processing times and optimal service level. *Eur. J. Oper. Res.* **2006**, *169*, 751–760. [[CrossRef](#)]
22. Daniels, R.L.; P. Kouvelis. Robust scheduling to hedge against processing time uncertainty in single stage production. *Manag. Sci.* **1995**, *41*, 363–376. [[CrossRef](#)]
23. Sabuncuoglu, I.; Goren, S. Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *Int. J. Comput. Integr. Manuf.* **2009**, *22*, 138–157. [[CrossRef](#)]
24. Subramaniam, V.; Raheja, A.S.; Rama Bhupal Reddy, K. Reactive repair tool for job shop schedules. *Int. J. Prod. Res.* **2005**, *1*, 1–23. [[CrossRef](#)]
25. Gur, S.; Eren, T. Scheduling and planning in service systems with goal programming: Literature review. *Mathematics* **2018**, *6*, 265. [[CrossRef](#)]
26. Pereira, J. The robust (minmax regret) single machine scheduling with interval processing times and total weighted completion time objective. *Comput. Oper. Res.* **2016**, *66*, 141–152. [[CrossRef](#)]

27. Kasperski, A.; Zielinski, P. A 2-approximation algorithm for interval data minmax regret sequencing problems with total flow time criterion. *Oper. Res. Lett.* **2008**, *36*, 343–344. [[CrossRef](#)]
28. Wu, Z.; Yu, S.; Li, T. A meta-model-based multi-objective evolutionary approach to robust job shop scheduling. *Mathematics* **2019**, *7*, 529. [[CrossRef](#)]
29. Kuroda, M.; Wang, Z. Fuzzy job shop scheduling. *Int. J. Prod. Econ.* **1996**, *44*, 45–51. [[CrossRef](#)]
30. Grabot, B.; Geneste, L. Dispatching rules in scheduling: A fuzzy approach. *Int. J. Prod. Res.* **1994**, *32*, 903–915. [[CrossRef](#)]
31. Özelkan, E.C.; Duckstein, L. Optimal fuzzy counterparts of scheduling rules. *Eur. J. Oper. Res.* **1999**, *113*, 593–609. [[CrossRef](#)]
32. Navakkoli-Moghaddam, R.; Safaei, N.; Kah, MMO. Accessing feasible space in a generalized job shop scheduling problem with the fuzzy processing times: A fuzzy-neural approach. *J. Oper. Res. Soc.* **2008**, *59*, 431–442. [[CrossRef](#)]
33. Al-Atroshi, A.M.; Azez, S.T.; Bhnem, B.S. An effective genetic algorithm for job shop scheduling with fuzzy degree of satisfaction. *Int. J. Comput. Sci. Issues* **2013**, *10*, 180–185.
34. Kasperski, A.; Zielinski, P. Possibilistic minmax regret sequencing problems with fuzzy parameteres. *IEEE Trans. Fuzzy Syst.* **2011**, *19*, 1072–1082. [[CrossRef](#)]
35. Gonzalez-Rodriguez, I.; Vela, C.R.; Puente, J.; Varela, R. A new local search for the job shop problem with uncertain durations. In Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS 2008), Sydney, Australia, 14–18 September 2008; Association for the Advancement of Artificial Intelligence: Menlo Park, CA, USA, 2008; pp. 124–131.
36. Allahverdi, A.; Sotskov, Yu.N. Two-machine flowshop minimum-lenght scheduling problem with random and bounded processing times. *Int. Trans. Oper. Res.* **2003**, *10*, 65–76. [[CrossRef](#)]
37. Matsveichuk, N.M.; Sotskov, Yu.N. A stability approach to two-stage scheduling problems with uncertain processing times. In *Sequencing and Scheduling with Inaccurate Data*; Yu, N., Sotskov, Werner, F., Eds.; Nova Science Publishers: Hauppauge, NY, USA, 2014; pp. 377–407.
38. Lai, T.-C.; Sotskov, Yu.N.; Egorova, N.G.; Werner, F. The optimality box in uncertain data for minimising the sum of the weighted job completion times. *Int. J. Prod. Res.* **2018**, *56*, 6336–6362. [[CrossRef](#)]
39. Sotskov, Yu.N.; Egorova, N.M. Single machine scheduling problem with interval processing times and total completion time objective. *Algorithms* **2018**, *75*, 66. [[CrossRef](#)]
40. Sotskov, Yu.N.; Lai, T.-C. Minimizing total weighted flow time under uncertainty using dominance and a stability box. *Comput. Oper. Res.* **2012**, *39*, 1271–1289. [[CrossRef](#)]
41. Matsveichuk, N.M.; Sotskov, Y.N.; Egorova, N.G.; Lai T.-C. Schedule execution for two-machine flow-shop with interval processing times. *Math. Comput. Model.* **2009**, *49*, 991–1011. [[CrossRef](#)]
42. Sotskov, Yu.N.; Allahverdi, A.; Lai, T.-C. Flowshop scheduling problem to minimize total completion time with random and bounded processing times. *J. Oper. Res. Soc.* **2004**, *55*, 277–286. [[CrossRef](#)]
43. Allahverdi, A.; Aldowaisan, T.; Sotskov, Yu.N. Two-machine flowshop scheduling problem to minimize makespan or total completion time with random and bounded setup times. *Int. J. Math. Math. Sci.* **2003**, *39*, 2475–2486. [[CrossRef](#)]
44. Kouvelis, P.; Yu, G. *Robust Discrete Optimization and Its Application*; Kluwer Academic Publishers: Boston, MA, USA, 1997.
45. Kouvelis, P.; Daniels, R.L.; Vairaktarakis, G. Robust scheduling of a two-machine flow shop with uncertain processing times. *IEEE Trans.* **2000**, *32*, 421–432. [[CrossRef](#)]
46. Kuwata, Y.; Morikawa, K.; Takahashi, K.; Nakamura, N. Robustness optimisation of the minimum makespan schedules in a job shop. *Int. J. Manuf. Technol. Manag.* **2003**, *5*, 1–9. [[CrossRef](#)]
47. Carlier, J.; Pinson, E. An algorithm for solving the job-shop problem. *Manag. Sci.* **1989**, *35*, 164–176. [[CrossRef](#)]
48. Xiong, J.; Xing, L.; Chen, Y.-W. Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. *Int. J. Prod. Econ.* **2013**, *141*, 112–126. [[CrossRef](#)]
49. Paprocka, I. The model of maintenance planning and production scheduling for maximising robustness. *Int. J. Prod. Res.* **2019**, *57*, 1–22. [[CrossRef](#)]
50. Xie, C.; Allen, T.T. Simulation and experimental design methods for job shop scheduling with material handling: A survey. *Int. J. Adv. Manuf. Technol.* **2015**, *80*, 233–243. [[CrossRef](#)]
51. Paprocka, I. Evaluation of the effects of a machine failure on the robustness of a job shop system—Proactive approaches. *Sustainability* **2019**, *11*, 65. [[CrossRef](#)]

52. Cigolini, R.; Perona, M.; Portioli, A. Comparison of order and release techniques in a dynamic and uncertain job shop environment. *Int. J. Prod. Res.* **1998**, *36*, 2931–2951. [[CrossRef](#)]
53. Luh, P.B.; Chen, D.; Thakur, L.S. An effective approach for job-shop scheduling with uncertain processing requirements. *IEEE Trans. Robot. Autom.* **1999**, *15*, 328–339. [[CrossRef](#)]
54. Ng, C.T.; Matsveichuk, N.M.; Sotskov, Y.N.; Cheng, T.C.E. Two-machine flow-shop minimum-length scheduling with interval processing times. *Asia-Pac. J. Oper. Res.* **2009**, *26*, 1–20. [[CrossRef](#)]
55. Matsveichuk, N.M.; Sotskov, Y.N.; Werner, F. The dominance digraph as a solution to the two-machine flow-shop problem with interval processing times. *Optimization* **2011**, *60*, 1493–1517. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).