Using a Stability Method for Scheduling and Sequencing with Interval Data

Yuri N. Sotskov
United Institute of Informatics Problems,
Surganova St. 6, 220012 Minsk, Belarus
email: sotskov@newman.bas-net.by

Frank Werner
Otto-von-Guericke-Universität, Fakultät für Mathematik,
39106 Magdeburg, Germany
email: frank.werner@ovgu.de

June 5, 2013

Abstract

Optimal scheduling is considered as sequencing the given activities over time in order to meet some given objective. Activities (jobs, operations) represent processes which use resources (machines) to produce goods or to provide services. This work presents a survey of scheduling and sequencing problems with interval or uncertain data which can be solved by a stability method. It is assumed that the job processing times (and other numerical parameters) may take any values from given closed intervals. For different types of problems, we discuss mathematical models, known results, and developed algorithms, which are based on a stability analysis of an optimal solution with respect to possible variations of the input parameters. The main attention is paid to multi-stage processing systems with a fixed job route through the given machines, but we also deal with single-stage scheduling problems. The surveyed stability method combines a stability analysis, a multi-stage scheduling decision framework (the off-line planning stage and the on-line scheduling stages), and the solution concept of a minimal dominant set of schedules that optimally covers all scenarios in the sense that for any scenario such a set contains at least one solution (an optimal sequence). The results discussed in this survey have been obtained in the period from 1988 to 2013. This work will provide the reader with a comprehensive understanding of the stability method and the models appropriate for scheduling and sequencing with interval data. This survey shows how one can use the stability of an optimal (or near optimal) sequence of the given activities to overcome the uncertainty of the numerical input data.

Keywords: Uncertainty, General shop, Job shop, Flow shop, Single machine scheduling

1 Introduction

The idea of a stability approach for solving scheduling problems with *uncertain* parameters arose at the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, when the first author investigated the *stability* of an optimal schedule [121]. This approach was further developed at the Institute of Mathematical Optimization of the Otto-von-Guericke University of Magdeburg in Germany, where both authors took part in several scientific projects in the period from 1993 till 2013. The stability approach was described in the monograph [122], which was written at the Department of Industrial and Business Management of the National Taiwan University, where the first author took part in the scientific projects headed by T.-C. Lai from 1997 till 2011.

This survey is about optimal *scheduling* with uncertain (or interval) numerical data. A schedule is considered as a mathematical object and so it does not matter, where the schedule is used. Optimal scheduling is considered as sequencing the given activities over *time* in order to meet a given objective. Activities (or jobs, operations) represent processes that use resources (or machines) to produce goods or provide services. This survey addresses the *stability* of an optimal schedule with respect to *uncertainty* of the input data. It is shown how one can use the stability of an optimal (or near optimal) sequence of the given activities to overcome the uncertainty of the numerical input data.

There is no unique method that will fit all the different types of uncertainties arising in the real world, and the method described in this survey aims to complement but not to replace other methods to deal with uncertainty in sequencing and scheduling. In particular, a *stochastic* method is useful when one has enough prior information to characterize the probability distributions of the random activity durations and there is a sufficiently large number of repetitions of similar processes. However, a stochastic method may have a limited significance for a small number of realizations of the process. In contrast to a stochastic model and a fuzzy model, we assume that in spite of the *uncertainty* of the input data, the desired *schedule* must fix the unique place in the *sequence* for each given activity. Contrary to most methods for dealing with uncertainty in scheduling, the main aim of the stability method is to construct really an optimal schedule for the actual numerical input data. Of course, this is only possible when the level of uncertainty is not very high. Otherwise, a decision-maker must look for a heuristic solution using the stability method or another method for scheduling under high uncertainty.

In order to outline the stability approach, let us discuss a connection between *time*, which is necessary for solving a scheduling problem and *time*, which is necessary for the realization of a schedule. What is the main role of *time* in practical scheduling? Since *time* has only one direction for variation (from present to future), actually a decision-maker may have reliable information only about the activities that are already realized, and there may be not enough information about the realization of the future activities involved in a real schedule. In practice, it is useful to take into account that a unit of *time* may have a different "price" while solving a scheduling problem. With this in mind, we distinguish several *time* phases in decision-making depending on a *static* or *dynamic scheduling* environment. The first, *off-line, proactive,* phase of scheduling is used before realization of the given activities (static scheduling) while the subsequent, *on-line, reactive* phases of scheduling are used when a part of the schedule has already been realized (dynamic scheduling). Actually, at the off-line phase of scheduling, there may be more *time* for decision-making and so the "price" of a *time* unit may be low. However, before the realization of the activities, there is usually not enough reliable information to construct an optimal schedule for the future realization of the activities. Thus, at the off-line phase of scheduling, it is necessary to create a solution (multiple schedules) to the scheduling problem provided that the main numerical input data are *uncertain*. At the off-line phase of scheduling, we propose to realize a *stability* analysis of the constructed schedules to deal with different contingencies that may arise during the schedule execution.

At the on-line phases (in the dynamic scheduling when a part of the schedule has already been realized), there is more reliable information for the decision-making (the situation becomes more *certain*, e.g., due to the known durations of the activities already executed). However, there is often not enough *time* for using this additional information in order to construct a post-optimal schedule. The main idea of the approach described in this work is to prepare for the on-line phases of scheduling when the "price" of a *time* unit will be higher than that at the off-line phase of scheduling [121, 122].

The processing systems considered in this survey are those arising in practice for scheduling in continuous manufacturing industries and in discrete manufacturing industries (e.g., cars, semiconductors). Continuous manufacturing industries often have two types of machines: Machines for the main operations (e.g., paper mills, steel mills, aluminum mills, chemical plants, refineries) and machines for finishing operations such as cutting of the material, bending, folding, painting and printing. Medium and short term scheduling of the latter operations often leads to *single machine* scheduling problems; see Section 6. Scheduling the main production operations leads to *flow shop* and *job shop* scheduling problems; see Section 5. The most general processing system considered in this survey is a *general shop*; see Section 3. Within this survey, we use mixed graphs for modeling a two-stage scheduling environment with different types of uncertainties; see Subsection 3.1. Problem settings, notations, and a scheme of the stability method are described in Section 2. Following the monographs [121, 122], most sections of this survey are written as self-sufficient for understanding. Remarks and bibliographic notes are given at the end of each section in the subsection entitled "Notes and references", where publications are cited, in which the results presented in the section have been obtained. We attempted to minimize the number of references used within other subsections.

2 Problem settings, notations, and the stability method

Mathematical problems arising in scheduling are studied in *scheduling theory* containing the two main parts based either on deterministic models or on non-deterministic models (stochastic, fuzzy, or uncertain). *Deterministic models* have been introduced for scheduling environments (see [15, 79, 135, 136] among others) in which the processing time (duration) of each job processed by a machine and other numerical parameters are supposed to be given in advance (before applying a scheduling procedure) and assumed to be fixed (constants) during the realization of a schedule. Often in real life, however, the exact numerical data are not known in advance, and difficulties arise when some job processing times (which were assumed to be fixed) will vary due to a change in a dynamic environment. Even if all processing times are fixed before scheduling, one is forced to take into account possible errors within the realization of a schedule, the precision of the equipment for calculating the numerical data, round-off errors in the calculation of a schedule on the computer, machine breakdowns, additionally arriving jobs, and so on.

More general scheduling settings have been considered using *stochastic models* [20, 79] (and *fuzzy models* [89]), where the job processing time is assumed to be a *random variable (a fuzzy number)* with a known probability distribution (with known membership function). In practice, difficulties may still arise in many scenarios. First, one may not have enough prior information to characterize the probability distribution (the membership function) of a random (fuzzy) numerical data. Second, even if the probability distributions (membership functions) of all random (fuzzy) numerical data are a priori known, these distributions (the membership functions) are really useful for a large number of realizations of similar scheduling environments, but they can be of little practical sense for a unique or small number of similar realizations.

In this survey, a model of one of the more realistic scheduling scenarios is considered: It is assumed that in the realization of a schedule, a processing time (or other numerical parameters) may take any real value between the *lower* and *upper bounds* given before scheduling. Obviously, a deterministic model is a special case of such a model (namely, if lower and upper bounds are identically given for a processing time). The model considered can also be interpreted as a stochastic one under *strict uncertainty* when there is no sufficient a priori information about the probability distribution of a random processing time (or more precisely, it is only known that the random processing time will fall between the given lower and upper bounds with probability one).

The general scheme of the stability approach for dealing with uncertainty in scheduling may be described as follows. The original scheduling problem is decomposed into two (or more) sequential scheduling problems (phases). At the off-line phase, a set of potentially optimal schedules (namely, a minimal dominant set of schedules) has to be constructed under the conditions of *uncertainty* of the given available numerical input data. It is assumed that only lower and upper bounds for the activity duration are known at the first, off-line, phase of scheduling. Moreover, the probability distributions of the random activity durations are assumed to be unknown between their lower and upper bounds. For solving a scheduling problem with such an *uncertain* input data, we propose to use a *stability* analysis of an optimal schedule with respect to the possible variations of the given parameters. Since the "price" of *time* is not high at the off-line phase of scheduling, a decision-maker can use even a time-consuming algorithm for solving a scheduling problem with uncertain numerical data. When some activities will be realized, a decision-maker will have more reliable information that may be used to find an optimal schedule. So, at the on-line phase of scheduling, it is necessary to choose a schedule (from the minimal dominant set of schedules constructed at the off-line phase) that must be realized in an optimal way. Such a schedule must be optimal for the actual activity durations. To solve a scheduling problem at the on-line phase, a decision-maker needs to use fast (i.e., polynomial-time) algorithms.

We consider a multi-stage processing system (for brevity, a *shop*), which consists of a set of machines $M = \{M_1, M_2, \dots, M_m\}$ that must process a set of jobs $J = \{J_1, J_2, \dots, J_n\}$. For the shop, there are assumed the following conditions.

Condition 1: *At any time, a machine $M_k \in M$ either processes one job or is idle.*

Condition 2: *At any time, a job $J_i \in J$ is either processed by one machine from the set M, waits for processing or is already completed.*

Condition 3: *The machine order $(M_{i_1}, M_{i_2}, \dots, M_{i_{n_i}})$ for processing a job $J_i \in J$, called the technological (or machine) route of job J_i, is fixed before scheduling.*

The processing of a job $J_i \in J$ by a machine $M_{i_k} \in M$ at the *stage* $k \in \{1, 2, \dots, n_i\}$ of the technological route is called an *operation* denoted as O_{ik}. Let Q be the set of all operations for processing all jobs from the set J: $Q = \{O_{ik} : J_i \in J,\ k = 1, 2, \dots, n_i\}$. If the machine routes are *identically* given for all jobs from the set J, e.g., $(M_1, M_2, \dots, M_m)$, then we have a *flow shop*, otherwise (if the machine routes may be given *differently* for different jobs), we have a *job shop*. In the former case, each job has to be processed once by each machine while in the latter case, repetitions and (or) absence of a machine in the technological route of a job are allowed. In both cases each operation is assigned to a certain machine, and the technological route $(M_{i_1}, M_{i_2}, \dots, M_{i_{n_i}})$ of job $J_i \in J$ defines linearly ordered operations (i.e., *a sequence*) $(O_{i1}, O_{i2}, \dots, O_{in_i})$ such that operation O_{ik} has to be processed by machine $M_{i_k} \in M$ after operation $O_{i,k-1}$ processed by machine $M_{i_{k-1}} \in M$ and before operation $O_{i,k+1}$ processed by machine $M_{i_{k+1}} \in M$, $k \in \{2, 3, \dots, n_i - 1\}$. For a flow shop, the equality $n_i = m$ holds for each job $J_i \in J$ while in a job shop, the value n_i may be smaller or larger than m or equal to m for a job J_i. For both flow and job shops, it is assumed that there are no other *precedence constraints* given on the set Q except those defined by the machine routes $(O_{i1}, O_{i2}, \dots, O_{in_i})$, $J_i \in J$.

Condition 4: *Preemption of an operation is forbidden.*

Thus, in any schedule, operation $O_{ij} \in Q$ being started at time s_{ij} has to be processed up to its completion time $c_{ij} = s_{ij} + p_{ij}$, where p_{ij} is the processing time of operation O_{ij}. Let Q_k denote the set of all operations from the set Q that have to be processed by machine $M_k \in M$. In a *deterministic model*, the processing times p_{ij} are fixed for all operations O_{ij}, $J_i \in J$, $j = 1, 2, \dots, n_i$. Therefore, a *schedule* may be defined as the set of starting times s_{ij} (or the set of completion times c_{ij}) of all operations Q. Such a set of the starting (or completion) times of the operations Q defines a unique *sequence* for processing the operations Q_k by each machine M_k, $k = 1, 2, \dots, m$. A schedule uniquely defines m sequences (one sequence of the operations Q_k for one machine $M_k \in M$) and vice versa. The objective of a shop-scheduling problem is to find such a *schedule* (i.e., to find such m sequences of the operations Q_k^J on the machines M_k, $k = 1, 2, \dots, m$) for which the value of the given objective function $\Phi(C_1, C_2, \dots, C_n)$ is minimal. Hereafter, C_i is equal to the completion time of job $J_i \in J$, i.e., $C_i = c_{in_i}$.

If the objective function $\Phi(C_1, C_2, \dots, C_n)$ is non-decreasing, such a criterion is called *regular*. In multi-stage scheduling, the most popular regular criteria are the minimization of maximum flow time (makespan)

$$\Phi(C_1, C_2, \dots, C_n) = \max\{C_i : J_i \in J\} = \mathcal{C}_{max}$$

and the minimization of mean flow time

$$\Phi(C_1, C_2, \dots, C_n) = \sum_{i=1}^{n} C_i = \sum \mathcal{C}_i.$$

Scheduling problems are classified by triplets $\alpha/\beta/\gamma$. Using such a triplet, the deterministic shop problems considered in Sections 3 – 6 are denoted by $\mathcal{J}//\mathcal{C}_{max}$ and $\mathcal{J}//\sum \mathcal{C}_i$ for the job shop and by $\mathcal{F}//\mathcal{C}_{max}$ and $\mathcal{F}//\sum \mathcal{C}_i$ for the flow shop. The symbols $\mathcal{J}$ and $\mathcal{F}$ are used to indicate a job shop and a flow shop, respectively.

The job shop problem is NP-hard for most criteria considered in scheduling theory even for a small number of machines and jobs. The following shop-scheduling problems are unary NP-hard: $\mathcal{F}3//\mathcal{C}_{max}$, $\mathcal{F}3//\sum \mathcal{C}_i$, $\mathcal{J}3/p_{ij} = 1/\mathcal{C}_{max}$ and $\mathcal{J}2/p_{ij} \in \{1, 2\}/\mathcal{C}_{max}$ [32, 61, 62]. Problems $\mathcal{F}/n = 3/\mathcal{C}_{max}$, $\mathcal{F}/n = 3/\sum \mathcal{C}_i$, $\mathcal{J}3/n = 2/\mathcal{C}_{max}$ and $\mathcal{J}3/n = 2/\sum \mathcal{C}_i$ are binary NP-hard [16, 95, 116].

The job shop problem $\mathcal{J}//\Phi$ is a special case of a *general shop* problem $\mathcal{G}//\Phi$, in which arbitrary precedence constraints may be given on the set of operations Q. Hereafter, Φ denotes any given regular criterion: $\Phi = \Phi(C_1, C_2, \ldots, C_n)$. A general shop is defined via a partially ordered set of the given operations.

In Section 3, we present results for the calculation of the stability radius of an optimal schedule for general and job shops. The main attention is paid to the results on a stability analysis that are used in the stability method. Section 3 deals with a mathematical model for scheduling scenarios in which the processing time of each operation $O_{ij} \in Q$ (or other numerical parameters like job release times, set-up times, transportation times, etc.) may be *uncertain* before applying a scheduling procedure and may take any real value between a given *lower bound* $a_{ij} \geq 0$ and an *upper bound* $b_{ij} \geq a_{ij}$.

Condition 5: *The actual processing time p_{ij} of the operation $O_{ij} \in Q$ may take any real value between given lower and upper bounds: $a_{ij} \leq p_{ij} \leq b_{ij}$.*

It is clear that not only the duration of the processing of operations may be a source of uncertainty in practical scheduling. Uncertainty may also arise from machine breakdowns, unexpected arrivals of new jobs, late arrivals of raw materials, modifications of setup times, release dates and (or) due dates. As it will be shown in Section 3, all these and some other sources of uncertainty may be treated in terms of problem $\mathcal{G}/a_{ij} \leq p_{ij} \leq b_{ij}/\Phi$. So, problem $\mathcal{G}/a_{ij} \leq p_{ij} \leq b_{ij}/\Phi$ seems to be rather realistic, at least, it is not very restrictive: Even if there is no prior information on the possible perturbations of the processing times p_{ij}, one can consider 0 as a given lower bound of p_{ij} and a sufficiently large number (e.g., the *planning horizon*) as a given upper bound of p_{ij}.

3 Stability radius of an optimal schedule

The usual assumption that the processing times of all operations, setup times, release dates, due dates, and other numerical parameters are exactly known before scheduling restricts practical aspects of deterministic models of scheduling theory since it is often not valid for real-world processes. This section is devoted to the results obtained for the stability analysis of an optimal schedule that may help to extend the significance of scheduling theory for real production scheduling problems. The terms "stability analysis", "sensitivity analysis" or "post-optimal analysis" are used for the phase of an optimization process at which a solution of an optimization problem has already been found, and additional calculations are performed in order to investigate how this solution depends on the numerical input data. The stability radius of an optimal schedule denotes the largest quantity of independent variations of the processing times of the operations (and other changeable numerical parameters) such that this schedule remains optimal.

The main reason for calculating the stability radii is that in most practical cases, the processing times of the operations (and other numerical input data) are inaccurate before applying a scheduling procedure. In such cases, a stability analysis is necessary to investigate the credibility of an optimal schedule. On the one hand, if the possible errors of the numerical parameters are larger than the stability radius of an optimal schedule, this schedule may not be the best in a practical realization, and so there is not much sense in likely large efforts to construct an optimal schedule: It may be more advisable to restrict the scheduling procedure to the construction of an approximate or heuristic solution. On the other hand, this is not the case when the possible change of each numerical parameter is less than or equal to the stability radius of an optimal schedule: An a priori constructed optimal schedule will remain optimal (i.e., the best for the given objective) in any practical realization. Another reason for calculating the stability radius is connected with the need to solve a set of similar scheduling problems. In reality, the main characteristics of a shop (such as the number of machines, the technological routes, the range of variations of the processing times, and so on) do not change quickly, and it may be possible to use previous computations of an optimal schedule for solving a new similar scheduling problem. Since the majority of scheduling problems is NP-hard, enumeration schemes such as branch-and-bound are often used for finding an optimal schedule. To this end, it is necessary to construct a solution tree, which is often huge. Unfortunately, most of the information contained in the solution tree is lost after having solved the problem. In such a situation, the stability radius of the optimal schedule constructed gives the possibility to use a large part of this information for solving further similar scheduling problems.

The rest of this section is organized as follows. In Subsection 3.1, we introduce a mixed (disjunctive) graph model to represent the input data of a general shop scheduling problem $\mathcal{G}//\Phi$. It is shown that various scheduling problems may be represented as extremal problems on mixed graphs, and the only requirement for such a representation is the prohibition of operation preemptions (Condition 4 on page 5). In Subsection 3.2, we describe how the stability radius of an optimal schedule (digraph) for problem $\mathcal{G}//\Phi$ can be calculated via the reduction to a non-linear mathematical programming problem. The advantage of studying the stability of an optimal digraph instead of the stability of an optimal schedule is demonstrated in Subsections 3.1 and 3.2. The calculation of the stability radius along with characterizations of its extreme values for the problems $\mathcal{G}//\mathcal{C}_{max}$ and $\mathcal{G}//\sum \mathcal{C}_i$ are considered in Subsection 3.3 and in Subsection 3.4, respectively.

3.1 Mixed graphs for modeling a general shop

We consider a general shop in which the given partially ordered set of operations Q has to be processed without interruptions by a set of machines $M = \{M_1, M_2, \ldots, M_m\}$.

Let p_{ij} denote the processing time (duration) of operation $O_{ij} \in Q$ and c_{ij} denote the completion time of operation O_{ij}. Operation preemption is not allowed (Condition 4 on page 5): If an operation O_{ij} starts at time s_{ij}, its processing is not interrupted until the time $c_{ij} = s_{ij} + p_{ij}$ of completion. The problem of finding an optimal schedule minimizing the given objective function Φ of the job completion times is denoted as $\mathcal{G}//\Phi$. The set of operations Q is partially ordered by the given precedence constraints (temporal constraints) $\to$. Given two operations $O_{ij} \in Q$ and $O_{uv} \in Q$, where operation O_{ij} is a predecessor of operation O_{uv}, i.e., $O_{ij} \to O_{uv}$, the inequality

$$c_{ij} + p_{uv} \leq c_{uv} \tag{3.1}$$

must hold for any schedule. Let Q_k be the set of operations processed by machine $M_k \in M$, and so the partition $Q = \bigcup_{k=1}^{m} Q_k$ of the set Q defines resource constraints. Since at any time each machine $M_k \in M$ can process at most one operation (Condition 1) and operation preemptions are not allowed (Condition 4), the two inclusions $O_{ij} \in Q_k$ and $O_{uv} \in Q_k$ imply one of the following disjunctive inequalities:

$$c_{ij} + p_{uv} \leq c_{uv} \quad \text{or} \quad c_{uv} + p_{ij} \leq c_{ij}. \tag{3.2}$$

For the case of a job shop, the set of operations Q is also partitioned into n chains (linearly ordered sets): $Q = \bigcup_{i=1}^{n} Q^{J_i}$, where each chain includes the set Q^{J_i} of operations for processing job J_i, $1 \leq i \leq n$.

For the problem $\mathcal{G}//\Phi$, the processing time p_{ij} of each operation $O_{ij} \in Q$ is fixed before scheduling, and therefore, a schedule of the operations Q on the machines M may be defined by the completion times c_{ij} or by the starting times $s_{ij} = c_{ij} - p_{ij}$ of the operations $O_{ij} \in Q$. If the operation processing times are not fixed before scheduling (Condition 5), it is not possible to define s_{ij} and c_{ij} for all operations $O_{ij} \in Q$ before the schedule execution. Therefore, in the general case of problem $\mathcal{G}/a_{ij} \leq p_{ij} \leq b_{ij}/\Phi$, the goal is to determine a sequence for processing the set of operations Q_k on each machine $M_k \in M = \{M_1, M_2, \dots, M_m\}$. Such a set of m sequences satisfying both the given precedence constraints (3.1) and the given capacity constraints (3.2) is defined as a schedule for the problem $\mathcal{G}/a_{ij} \leq p_{ij} \leq b_{ij}/\Phi$. The general shop problem is to find a schedule that minimizes the value of a given non-decreasing objective function $\Phi(C_1, C_2, \dots, C_n)$.

A mixed (disjunctive) graph is often used to model a deterministic scheduling problem. We also represent the structural input data for a general shop problem by means of a mixed graph $G = (Q, A, E)$, where the set Q of operations is the set of vertices; the precedence constraints (3.1) are represented by the set of non-transitive arcs A. If operation O_{ij} must be processed before operation O_{uv} starts, i.e., a precedence constraint $O_{ij} \to O_{uv}$ is given, then the arc (O_{ij}, O_{uv}) has to belong to the set

$$A = \{(O_{ij}, O_{uv}) : O_{ij} \to O_{uv}\}.$$

The capacity constraints (3.2) are represented by the set E of edges $[O_{ij}, O_{uv}]$ connecting the unordered operations O_{ij} and O_{uv}, which have to be processed by the same machine:

$$E = \{[O_{ij}, O_{uv}] \ : \ O_{ij} \in Q_k, \ O_{uv} \in Q_k \ \text{ neither } O_{ij} \to O_{uv} \text{ nor } O_{uv} \to O_{ij} \text{ holds}\}.$$

For a deterministic problem $\mathcal{G}//\Phi$, one can associate a non-negative weight p_{ij} with each vertex $O_{ij} \in Q$ in the mixed graph $G = (Q, A, E)$ constructed for the problem $\mathcal{G}//\Phi$. The weighted mixed graph $G(p) = (Q(p), A, E)$ with $p = (p_{1,1}, p_{1,2}, \dots, p_{nn_n})$, represents both the structural and numerical input data for the general shop scheduling problem $\mathcal{G}//\Phi$. For solving the problem $\mathcal{G}//\Phi$, we replace the edge $[O_{ij}, O_{uv}] \in E$ by an arc incident to the vertexes O_{ij} and O_{uv}. Indeed, due to Condition 1, if the edge $[O_{ij}, O_{uv}]$ belongs to the set E, then for the operations O_{ij} and O_{uv}, there exist two possibilities: To provide the first inequality from (3.2) (in this case the edge $[O_{ij}, O_{uv}]$ has to be replaced by the arc (O_{ij}, O_{uv})), or to provide the second inequality from (3.2) (in this case the edge $[O_{ij}, O_{uv}]$ has to be replaced by the arc (O_{uv}, O_{ij})). The digraph $G_s = (Q, A \cup E_s, \emptyset)$ generated from the mixed graph $G = (Q, A, E)$ by an orientation of all edges E is *feasible* if and only if G_s contains no circuit. Let $\Lambda(G) = \{G_1, G_2, \dots, G_\lambda\}$ be the set of all feasible (or circuit-free) digraphs $G_s = (Q, A \cup E_s, \emptyset)$ generated from G. We remind that *a schedule is called semiactive if no operation $O_{ij} \in Q$ can start earlier without altering the processing sequence of the operations on any of the machines M.*

Lemma 1 *If the objective function $\Phi(C_1, C_2, \dots, C_n)$ is non-decreasing, it is sufficient to check only semiactive schedules while solving the problem $\mathcal{G}//\Phi$.*

Lemma 2 *Each feasible digraph $G_s = (Q, A \cup E_s, \emptyset) \in \Lambda(G)$ uniquely defines a semiactive schedule $s \in S$ for the problem $\mathcal{G}//\Phi$, and vice versa.*

Due to Lemma 1, for solving the problem $\mathcal{G}//\Phi$, we can restrict ourselves to the consideration of the set S of all semiactive schedules. Given a vector $p = (p_{1,1}, p_{1,2}, \dots, p_{nn_n})$, the digraphs $G_s \in \Lambda(G)$ and $G_s(p)$ are called optimal if and only if schedule $s \in S$ is optimal. Due to Lemma 2, we can use a digraph $G_s \in \Lambda(G)$ (an optimal digraph G_s) instead of a schedule $s \in S$ (instead of an optimal schedule s). The digraph $G_s \in \Lambda(G)$ uniquely defines a set of m sequences for processing the operations Q_k by the machine $M_k \in M = \{M_1, M_2, \dots, M_m\}$, and vice versa. It should be noted that the digraph $G_s \in \Lambda(G)$ is more appropriate when performing a stability analysis for the problem $\mathcal{G}//\Phi$ since it is 'more stable' than the corresponding schedule $s \in S$ with respect to the variations of the operation durations (and other numerical parameters involved in the schedule s). It is more useful to consider a stability analysis for an optimal digraph $G_s \in \Lambda(G)$ (defining the set of m optimal sequences for processing the operations Q_k by the machine $M_k \in \{M_1, M_2, \dots, M_m\}$) than for the optimal schedule $s \in S$. Frequently, a production process is controlled by the operation sequences and not the actual starting and completion times of the operations provided by a schedule. Also, even any small change in a processing time definitely changes a schedule. But the practical need to reschedule occurs when the operation sequence changes. Large and more meaningful ranges of numerical parameter changes are obtained when considering the digraph $G_s \in \Lambda(G)$ instead of the schedule $s \in S$. It is often more important in practice to keep in mind not the calendar times when the operations have to be started and have to be completed, but only the m sequences in which the operations $Q_k, k = 1, 2, \dots, m$, have to be processed on each machine $M_k \in M$ (these sequences are uniquely defined by the digraph G_s). Note also that the starting and completion times of the operations Q, the value of the objective function and other characteristics of a semiactive schedule s, corresponding to an acyclic weighted digraph $G_s(p)$, can be easily determined using *critical path calculations*.

Given a fixed vector $p = (p_{1,1}, p_{1,2}, \dots, p_{nn_n})$ of the processing times, in order to construct an optimal schedule for the problem $\mathcal{G}//\Phi$, one may enumerate (at least explicitly) the feasible digraphs $G_1(p), G_2(p), \dots, G_\lambda(p)$ generated by the mixed graph G and then select an optimal digraph, i.e., a feasible digraph with minimal value of the objective function. Although problem $\mathcal{G}//\Phi$ is unary NP-hard for any regular criterion Φ, [58, 136], the running time of calculating an optimal schedule $s = (c_1(s), c_2(s), \dots, c_q(s))$ may be restricted by an $O(q^2)$-algorithm after having constructed an optimal digraph $G_s(p)$. Thus, the main difficulty of the problem $\mathcal{G}//\Phi$ (in terms of the mixed graph approach) is to construct an optimal digraph $G_s = (Q, A \cup E_s, \emptyset)$, i.e., to find the best set E_s of arcs generated by orienting the edges of the set E. Each feasible digraph $G_s = (Q, A \cup E_s, \emptyset)$ is uniquely defined by the set of arcs E_s replacing the edge set E.

If the operation durations (or other parameters) given before scheduling may vary in the realization of a schedule, it is not sufficient to construct only an optimal digraph $G_s \in \Lambda(G)$ for solving the problem $\mathcal{G}//\Phi$. It is important to analyze the question of how much the operation durations (or other parameters) may vary so that the digraph G_s remains optimal. Next, we show that, due to the generality of the above mixed graph model, one can analyze other changeable parameters of a general shop problem (like *release dates, due dates, job weights, setup times*, etc.) in terms of the mixed graph G.

In particular, including a dummy operation 0 into the mixed graph G which proceeds the first operation of each job $J_i \in J$ allows one to consider the *processing time* p_0 of this dummy operation as a *release date* of job J_i. Let d be a sufficiently large number. Including a dummy operation q into the mixed graph G which succeeds the last operation of job J_i allows one to consider the *processing time* of the dummy operation q as a *due date* d_i of job J_i. If machine

$M_k \in M$ needs a sequence-independent setup time $\tau_{ij} \geq 0$ before starting the operation $O_{uv} \in Q_k$ after the completion of the operation $O_{ij} \in Q_k$, $[O_{ij}, O_{uv}] \in E$, then the weight (processing time) of the operation O_{ij} has to be increased by the value τ_{ij} ($p_{ij} := p_{ij} + \tau_{ij}$). If job $J_i \in J$ needs a sequence-independent transportation time $\vartheta_{ij} \geq 0$ before starting the operation $O_{i,j+1} \in Q^{J_i}$ after the completion of the operation $O_{ij} \in Q^{J_i}$, then the weight (processing time) of operation O_{ij} has to be increased by the value ϑ_{ij} ($p_{ij} := p_{ij} + \vartheta_{ij}$). If the given setup times $\tau_{ij,uv} \geq 0$ (or transportation times $\vartheta_{ij} \geq 0$) are sequence-dependent, then we have to consider also weighted arcs and edges in the mixed graph G. Namely, the arc $(O_{ij}, O_{uv}) \in A$ must have the weight $\tau_{ij,uv}$ (weight $\vartheta_{ij,uv}$, respectively). The edge $[i,j] \in E$ must have two weights: the weight τ_{ij} (weight $\vartheta_{O_{ij},O_{uv}}$) for an orientation $(O_{ij}, O_{uv}) \in E_s$ of the edge $[O_{ij}, O_{uv}]$, and the weight $\tau_{uv,ij}$ (weight $\vartheta_{uv,ij}$) for an orientation $(O_{uv}, O_{ij}) \in E_s$ of this edge. Job weights may also be taken into account in the objective function Φ.

Remark 1 The mixed graph model G allows one to consider other parameters (like *release dates, due dates, setup times*, etc.) than the operation processing times.

In the following subsections, we present some results for the stability ball of an optimal digraph $G_s(p)$, i.e., a closed ball in the space of the numerical input data such that within this ball a schedule s remains optimal. For simplicity, due to Remark 1, we will consider the operation durations (processing times) as the only changeable parameters.

3.2 Regular criterion

This subsection is devoted to the stability of an optimal digraph $G_s(p)$, which represents a solution to problem $\mathcal{G}//\Phi$. The main question is as follows. Under which largest simultaneous and independent changes in the components of the vector $p = (p_{1,1}, p_{1,2}, \dots, p_{nn_n})$ of the operation processing times remains digraph $G_s(p)$ optimal? Let $|Q| = q$ and R^q be the space of all q-dimensional real vectors p with the maximum metric, i.e., the distance $d(p, p')$ between the vectors $p \in R^q$ and $p' = (p'_{1,1}, p'_{1,2}, \dots, p'_{nn_n}) \in R^q$ is defined as $d(p, p') = \max_{O_{ij} \in Q} |p_{ij} - p'_{ij}|$, where $|p_{ij} - p'_{ij}|$ denotes the absolute value of the difference $p_{ij} - p'_{ij}$. Let R^q_+ be space of q-dimensional non-negative real vectors:

$$R^q_+ = \{x = (x_{1,1}, x_{1,2}, \dots, x_{nn_n}) : \ x_{ij} \geq 0, \ O_{ij} \in Q\}$$

and schedule $s \in S$ be optimal for the problem $\mathcal{G}//\Phi$ with the vector $p \in R^q_+ \subset R^q$ of the operation durations.

Definition 1 *A closed ball $O_\varrho(p)$ with the radius $\varrho \in R^1_+$ and the center $p \in R^q_+$ in the space of the q-dimensional real vectors R^q is a stability ball of schedule $s \in S$ if for any vector $p' \in O_\varrho(p) \cap R^q_+$ of the processing times, schedule s remains optimal. The maximum value $\varrho_s(p)$ of the radius ϱ of the stability ball $O_\varrho(p)$ of schedule s is the stability radius of schedule s (of digraph G_s):*

$$\varrho_s(p) = \max\{\varrho \in R^1_+ : \ \textit{If } p' \in O_\varrho(p) \cap R^q_+, \ \textit{digraph } G_s \textit{ is optimal}\}. \tag{3.3}$$

We denote the stability radius by $\varrho_s(p)$ for an arbitrarily given regular criterion Φ. For the criterion $\mathcal{C}_{max}$, the stability radius is denoted by $\widehat{\varrho}_s(p)$, and for the criterion $\sum \mathcal{C}_i$ by $\overline{\varrho}_s(p)$. In what follows, we use whenever appropriate the notion "stability radius of an optimal digraph

$G_s \in \Lambda(G)$" instead of "stability radius of an optimal schedule $s \in S$". Due to the maximum metric, the set $O_\varrho(p) \cap R^q_+$ is a polytope for any positive $\varrho \in R^1_+$. Definition 1 implies a general approach for calculating $\varrho_s(p)$, which is discussed in this subsection for any given regular criterion Φ and in Subsection 3.3 and in Subsection 3.4 for $\Phi = \mathcal{C}_{max}$ and for $\Phi = \sum \mathcal{C}_i$, respectively.

The calculation of the stability radius $\varrho_s(p)$ is reduced to the solution of a non-linear programming problem. We give this reduction for the problem $\mathcal{G}//\Phi$ provided that the set of all operations $Q = \{O_{1,1}, O_{1,2}, \dots, O_{nn_n}\}$ is partitioned into n linearly ordered subsets of the operations Q^{J_i} defining the machine routes of the jobs $J_i \in J$. We denote by $\{\mu\}$ the set of vertices that form a path μ in the digraph G_k and by $l^p(\mu)$ the weight of this path: $l^p(\mu) = \sum_{O_{ij} \in \{\mu\}} p_{ij}$. Let $\tilde{H}^i_k$ denote the set of all paths in the digraph $G_k = (Q, A \cup E_k, \emptyset) \in \Lambda(G)$ ending in the last operation O_{ii_n} of job J_i. The completion time $C_i(k) = c_{ii_n}(k)$ of the job J_i in a semiactive schedule $k \in S$ is equal to the value $\max_{\mu \in \tilde{H}^i_k} l^p(\mu)$ of the largest weight of a path in the set $\tilde{H}^i_k$. While calculating $c_{ii_n}(k), J_i \in J$, it is sufficient to consider a subset of the set $\tilde{H}^i_k$ due to the following dominance relation.

Definition 2 *The path $\mu \in \tilde{H}^i_s$ is dominant if there is no path $\nu \in \tilde{H}^i_s$ such that $\{\mu\} \subset \{\nu\}$. Otherwise, if $\{\mu\} \subset \{\nu\}$, path μ is dominated by path ν.*

The dominance relation given in Definition 2 is a *strict order*. Let H^i_k denote the set of all dominant paths in the set $\tilde{H}^i_k$. Since $p_{ij} \geq 0$ for any operation $O_{ij} \in Q$, we obtain $C_i(k) = c_{ii_n}(k) = \max_{\mu \in H^i_k} l^p(\mu)$. The objective function value $\Phi(C_1, C_2, \dots, C_n)$ for the semiactive schedule $s = (c_{1,1}(s), c_{1,2}(s), \dots, c_{nn_n}(s)) \in S$ may be calculated as follows:

$$\Phi(\max_{\mu \in H^1_s} l^p(\mu), \max_{\mu \in H^2_s} l^p(\mu), \dots, \max_{\mu \in H^n_s} l^p(\mu)).$$

Therefore: *The semiactive schedule $s = (c_{1,1}(s), c_{1,2}(s), \dots, c_{nn_n}(s)) \in S$ is optimal for problem $\mathcal{G}//\Phi$ with a regular criterion Φ if and only if*

$$\Phi(\max_{\mu \in H^1_s} l^p(\mu), \dots, \max_{\mu \in H^n_s} l^p(\mu)) = \min_{k=1,2,\dots,\lambda} \Phi(\max_{\nu \in H^1_k} l^p(\nu), \dots, \max_{\nu \in H^n_k} l^p(\nu)). \tag{3.4}$$

We denote

$$\Phi^p_s = \Phi(\max_{\mu \in H^1_s} l^p(\mu), \dots, \max_{\mu \in H^n_s} l^p(\mu)).$$

Let the subset $S^\Phi(p)$ of the set S denote the set of all optimal semiactive schedules for the problem $\mathcal{G}//\Phi$ with the vector $p = (p_{1,1}, p_{1,2}, \dots, p_{nn_n}) \in R^q_+$ of the operation durations and let the inclusion $s \in S^\Phi(p)$ hold. Due to Definition 1, the stability radius $\varrho_s(p)$ may be defined as follows:

$$\varrho_s(p) = \inf\{d(p, x) : x \in R^q_+, s \notin S^\Phi(x)\}. \tag{3.5}$$

While equality (3.3) defines the stability radius $\varrho_s(p)$ as the maximal real number, which may be a radius of a stability ball $O_\varrho(p)$, equality (3.5) defines the value of $\varrho_s(p)$ as the infimum of the non-negative real numbers that cannot be a radius of a stability ball $O_\varrho(p)$. From (3.5), it follows that, in order to calculate the stability radius $\varrho_s(p)$, it is sufficient to calculate the optimal value of the objective function $f(x_{1,1}, x_{1,2}, \dots, x_{nn_n})$ of the following non-linear programming problem:

$$\text{Minimize } f(x_{1,1}, x_{1,2}, \dots, x_{nn_n}) = \max_{O_{ij} \in Q} |x_{ij} - p_{ij}| \tag{3.6}$$

$$\text{subject to } \Phi_s^x > \min\{\Phi_k^x \; : \; k = 1, 2, \ldots, \lambda; \; k \neq s\}; \tag{3.7}$$

$$x_{ij} \geq 0, \quad O_{ij} \in Q. \tag{3.8}$$

If condition (3.7) is not satisfied for any non-negative vector $x \in R_+^q$, then the digraph $G_s(p)$ is optimal for all vectors $x \in R_+^q$: $s \in S^\Phi(x)$, $x \in R^q$, and we obtain

$$\begin{cases} \Phi_s^x \leq \min\{\Phi_k^x \; : \; k = 1, 2, \ldots, \lambda; \; k \neq s\}, \\ x_{ij} \geq 0, \quad O_{ij} \in Q. \end{cases}$$

In this case, we say that the stability radius $\varrho_s(p)$ is *infinitely large*. To indicate that the stability radius is infinitely large we write: $\varrho_s(p) = \infty$.

In all other cases, there exists an optimal value f^* of the objective function of problem (3.6) – (3.8):

$$f^* = \inf \max_{O_{ij} \in Q} |x_{ij} - p_{ij}|,$$

where the infimum is taken over all vectors x satisfying conditions (3.7) and (3.8). To find the value f^*, it is sufficient to calculate a solution $x^0 = (x_{1,1}^0, x_{1,2}^0, \ldots, x_{nn_n}^0)$ of the non-linear programming problem that is obtained from problem (3.6) – (3.8) due to the replacement of the sign $>$ in inequality (3.7) by the sign $\geq$. The equalities

$$f^* = \max_{O_{ij} \in Q} |x_{ij}^0 - p_{ij}| = d(x^0, p) = \varrho_s(p)$$

hold. For any small $\epsilon > 0$, there exists a vector $x^\epsilon = (x_{1,1}^\epsilon, x_{1,2}^\epsilon, \ldots, x_{nn_n}^\epsilon) \in R_+^q$ such that $d(x^\epsilon, p) = \varrho_s(p) + \epsilon$ and $s \notin S^\Phi(x^\epsilon)$. It may occur that the given vector p is itself a solution to the non-linear programming problem (3.6) – (3.8) with the sign $\geq$ in inequality (3.7). In the latter case, the equalities $\varrho_s(p) = d(p, p) = 0$ hold, and it means that the optimality of the digraph $G_s(p)$ is *unstable*: For any small real $\epsilon > 0$, there exists a vector $p' \in R_+^q$ such that $s \notin S^\Phi(p')$ and $d(p, p') = \epsilon$. In this case, we say that the optimal digraph $G_s(p)$ is unstable. If the given vector p is not a solution of such a problem, we have the strict inequality $\varrho_s(p) > 0$. In this case, the optimal digraph $G_s(p)$ is *stable*.

3.3 Maximum flow time (makespan)

The best studied case of the shop-scheduling problem $\mathcal{G}//\Phi$ is the one with $\Phi = \mathcal{C}_{max}$, where the objective is to find a makespan *optimal* schedule, i.e., to find a schedule

$$s = (c_{1n_1}(s), c_{2n_2}(s), \ldots, c_{nn_n}(s))$$

with a minimum value of the maximum flow time $\max\{c_{in_i}(s) : J_i \in J\}$. Let $\tilde{H}_k$ ($\tilde{H}$, respectively) be the set of all paths in the digraph $G_k(p) \in \Lambda(G)$ (in the digraph $(Q, A, \emptyset)$) constructed for the general shop problem $\mathcal{G}//\mathcal{C}_{max}$. Let H and H_k denote the set of all dominant paths in the digraph $(Q, A, \emptyset)$ and in the digraph $G_k \in \Lambda(G)$, respectively (see Definition 2). Thus, we can write $H_k = \{\nu \in \tilde{H}_k :$ Inclusion $\{\nu\} \subset \{\mu\}$ does not hold for any path $\mu \in \tilde{H}_k\}$. The set $H \subseteq \tilde{H}$ is defined similarly. The value of $\max_{i=1}^n C_i$ of a schedule s is given by the weight of a maximum-weight path (a *critical path*) in the weighted digraph $G_s(p)$. Obviously, at least one critical path in $G_s(p)$ is dominant and for any path $\mu \in H$, there exists a path $\nu \in H_s$ such that either ν dominates path μ or the inclusion $\mu \in H_s$ holds. The equality (3.4) for the problem $\mathcal{G}//\mathcal{C}_{max}$ is converted into the following one:

$$\max_{\mu \in H_s} l^p(\mu) = \min_{k=1,2,\ldots,\lambda} \max_{\nu \in H_k} l^p(\nu). \tag{3.9}$$

Therefore: *The schedule $s = (c_1(s), c_2(s), \ldots, c_q(s)) \in S$ is optimal for the problem $\mathcal{G}//\mathcal{C}_{max}$ if and only if equality (3.9) holds.*

Let $H_k(p)$ denote the set of all critical dominant paths in the digraph $G_k(p)$.

Lemma 3 *There exists a real $\epsilon > 0$ such that the set $H_k \backslash H_k(p)$ contains no critical path of the digraph $G_k \in P(G)$ for any vector of the processing times $p^\epsilon = (p^\epsilon_{1,1}, p^\epsilon_{1,2}, \ldots, p^\epsilon_{nn_n}) \in O_\epsilon(p) \cap R^q_+$, i.e., $H_k(p^\epsilon) \subseteq H_k(p)$.*

Theorem 1 *For an optimal schedule $s \in S^\Phi(p)$, $\Phi = \mathcal{C}_{max}$, of the problem $\mathcal{G}//\mathcal{C}_{max}$, the equality $\widehat{\varrho}_s(p) = 0$ holds if and only if there exist another optimal schedule $k \in S^\Phi(p)$, $k \neq s$, and a path $\mu^* \in H_s(p)$ such that there is no a path $\nu^* \in H_k(p)$ with $\{\mu^*\} \subseteq \{\nu^*\}$.*

Since the condition of Theorem 1 is violated when $H_s(p) \subseteq H$, we obtain

Corollary 1 *If s is an optimal schedule for the problem $\mathcal{G}//\mathcal{C}_{max}$ and $H_s(p) \subseteq H$, then $\widehat{\varrho}_s(p) > 0$.*

For the following corollary from Theorem 1, it is not necessary to know the set $H_s(p)$.

Corollary 2 *If s is a unique optimal schedule for the problem $\mathcal{G}//\mathcal{C}_{max}$, then $\widehat{\varrho}_s(p) > 0$.*

A characterization of an infinitely large stability radius is given by the following

Theorem 2 *For an optimal schedule $s \in S^\Phi(p)$, $\Phi = \mathcal{C}_{max}$, of the problem $\mathcal{G}//\mathcal{C}_{max}$, the stability radius $\widehat{\varrho}_s(p)$ is infinitely large if and only if for any path $\mu \in H_s \backslash H$ and for any digraph $G_t(p) \in \Lambda(G)$, there exists a path $\nu \in H_t$ such that $\{\mu\} \subseteq \{\nu\}$.*

Corollary 3 *If $\widehat{\varrho}_s(p) < \infty$, then $\widehat{\varrho}_s(p) \leq \max_{O_{ij} \in Q} p_{ij}$.*

Due to Theorem 2, one can identify a general shop problem $\mathcal{G}//\mathcal{C}_{max}$ whose optimal schedule is implied only by the precedence constraints given on the set of operations Q and by the given distribution of the operations Q to the machines M, but it is independent of the processing times $p \in R^q_+$. Because of the generality of the problem $\mathcal{G}//\mathcal{C}_{max}$, it is practically difficult to check the condition of Theorem 2. However, for a job shop problem $\mathcal{J}//\mathcal{C}_{max}$, there are necessary and sufficient conditions for $\widehat{\varrho}_s(p) = \infty$, which can be verified in $O(q^2)$ time. To present the latter conditions, we need the following notations. Let A_k (B_k, respectively) be the set of all operations $O_{ij} \in Q$ such that $O_{ij} \to O_{uv}$ ($O_{uv} \to O_{ij}$) and $O_{uv} \in Q_k$, $O_{ij} \notin Q_k$: $A_k = \{O_{ij} \ : \ O_{ij} \to O_{uv},\ O_{ij} \in Q \cap Q_k,\ O_{uv} \in Q_k\}$; $B_k = \{O_{uv} \ : \ O_{ij} \to O_{uv},\ O_{uv} \in Q \cap Q_k,\ O_{ij} \in Q_k\}$. For the operation set B, let $n(B)$ denote the number of jobs in the set J having at least one operation in the set B.

Theorem 3 *For the problem $\mathcal{J}//\mathcal{C}_{max}$, there exists an optimal digraph $G_s(p)$ with an infinitely large stability radius if and only if both the following conditions hold:*

(1) The inequality $\max\{|A_k|, |B_k|\} \leq 1$ *holds for any machine M_k with $n(Q_k) > 1$;*

(2) If there exist two operations $O_{gh} \in A_k$ and $O_{ab} \in B_k$ of job J_l, then there exists a path from vertex O_{ab} to vertex O_{gh} in the digraph $(Q, A, \emptyset)$ (possibly $O_{ab} = O_{gh}$).

From Theorem 3, it follows that there are job shop problems $\mathcal{J}//\mathcal{C}_{max}$ with an optimal schedule having an infinitely large stability radius for any given n and m. Testing the condition of Theorem 3 takes $O(q^2)$ time. Note that for a flow shop problem, such a schedule can exist only if n or m is equal to 1. Indeed, for the problem $\mathcal{F}//\mathcal{C}_{max}$, we have $n(A_k) > 1$ for any machine M_k with $k \geq 2$ provided that $n \geq 2$ and $m \geq 2$.

Corollary 4 *For the problem $\mathcal{F}//\mathcal{C}_{max}$ with $n \geq 2$ and $m \geq 2$, the inequality $\widehat{\varrho}_s(p) > 0$ holds.*

There does not exist an optimal schedule s with $\varrho_s(p) = \infty$ for a problem $\mathcal{J}//\Phi$ with other regular criteria Φ presented in [58]. For calculating the stability radius $\widehat{\varrho}_s(p)$, one can consider the operations of the set $\{\nu\} \setminus \{\mu\}$ in non-decreasing order of their processing times. Let $p^{\nu\mu}_{(0)}$ be equal to zero and let $(p^{\nu\mu}_{(0)}, p^{\nu\mu}_{(1)}, \ldots, p^{\nu\mu}_{(\omega_{\nu\mu})})$ denote a non-decreasing sequence of the processing times of the operations from the set $\{\nu\} \setminus \{\mu\}$, where $w_{\nu\mu} = |\{\nu\} \setminus \{\mu\}|$.

Theorem 4 *If G_s is an optimal digraph for the problem $\mathcal{G}//\mathcal{C}_{max}$, $s \in S^{\Phi}(p)$, $\Phi = \mathcal{C}_{max}$, and the inequality $\widehat{\varrho}_s(p) < \infty$ holds, then*

$$\widehat{\varrho}_s(p) = \min_{k=1,2,\ldots,\lambda; k \neq s} \widehat{r}_{ks}, \ \ \text{where} \tag{3.10}$$

$$\widehat{r}_{ks} = \min_{\mu \in H_{sk}} \ \max_{\nu \in H_k, \ l^p(\nu) \geq l^p_s} \ \max_{\beta=0,1,\ldots,w_{\nu\mu}} \frac{l^p(\nu) - l^p(\mu) - \sum_{\alpha=0}^{\beta} p^{\nu\mu}_{(\alpha)}}{|\{\mu\} \cup \{\nu\}| - |\{\mu\} \cap \{\nu\}| - \beta}. \tag{3.11}$$

Equalities (3.10) and (3.11) mean that one has to compare an optimal digraph $G_s(p)$ with other feasible digraphs $G_k(p)$. Note that the formulas in Theorem 4 turn into $\widehat{\varrho}_s(p) = \infty$ if $H_{sk} = \emptyset$ for any $k = 1, 2, \ldots, \lambda$, $k \neq s$ (Theorem 2). Moreover, if only a subset of the processing times (say, $P \subseteq \{p_{1,1}, p_{1,2}, \ldots, p_{nn_n}\}$) can be changed but the other ones cannot be changed, Theorem 4 remains valid provided that the difference $|\{\mu\} \cup \{\nu\}| - |\{\mu\} \cap \{\nu\}|$ is replaced by $|\{\{\mu\} \cup \{\nu\}\} \cap P| - |\{\{\mu\} \cap \{\nu\}\} \cap P|$ in equality (3.11).

3.4 Mean flow time minimization

In this subsection, we consider the stability radius $\overline{\varrho}_s(p)$ of an optimal schedule for the problem $\mathcal{G}//\sum \mathcal{C}_i$ with the criterion $\sum \mathcal{C}_i$. If $\Phi = \sum \mathcal{C}_i$, the conditions (3.4) and (3.5) for the general shop problem $\mathcal{G}//\Phi$ are converted into the following ones:

$$\sum_{i=1}^{n} \max_{\mu \in H^i_s} l^p(\mu) = \min_{k=1,2,\ldots,\lambda} \ \sum_{i=1}^{n} \max_{\nu \in H^i_k} l^p(\nu),$$

$$\overline{\varrho}_s(p) = \inf \Big\{ d(p, x) \ : \ x \in R^q_+, \ \sum_{i=1}^{n} \max_{\mu \in H^i_s} l^x(\mu) > \min_{k=1,2,\ldots,\lambda; \ k \neq s} \ \sum_{i=1}^{n} \max_{\nu \in H^i_k} l^x(\nu) \Big\}.$$

Let Ω_k^u be a set of representatives of the family of sets $(H_k^i)_{1\le i\le n}$. More precisely, the set Ω_k^u includes exactly one path from each set $H_k^i, 1 \le i \le n$. Since $H_k^i \cap H_k^j = \emptyset$ for each pair of different jobs J_i and J_j, we have $|\Omega_k^u| = n$ and there exist $\omega_k = \prod_{i=1}^n |H_k^i|$ different sets of representatives for each digraph G_k, namely: $\Omega_k^1, \Omega_k^2, \ldots, \Omega_k^{\omega_k}$. For each set Ω_k^u, we can calculate the integer vector

$$n(\Omega_k^u) = (n_{i_1,j_1}(\Omega_k^u), n_{i_2,j_2}(\Omega_k^u), \ldots, n_{i_q,j_q}(\Omega_k^u)),$$

where for each operation $O_{i_d,j_d} \in Q = \{O_{i_1,j_1}, O_{i_2,j_2}, ..., O_{i_q,j_q}\}$, the number $n_{i_d,j_d}(\Omega_k^u)$ is equal to the number of paths in the set Ω_k^u, which include the vertex O_{i_d,j_d}. Since a path $\nu \in H_k^i$ includes any vertex $O_{i_d,j_d} \in Q$ at most once, the value $n_{i_d,j_d}(\Omega_k^u)$ is equal to the number of copies of the vertex O_{i_d,j_d} contained in the multiset $\{\{\nu\} : \nu \in \Omega_k^u\}$.

Let the set of operations Q be ordered in the following way:

$$O_{i_1j_i}, O_{i_2j_2}, \ldots, O_{i_mj_m}, O_{i_{m+1}j_{m+1}}, \ldots, O_{i_qj_q}, \tag{3.12}$$

where $n_{i_\alpha j_\alpha}(\Omega_k^u) \le n_{i_\alpha j_\alpha}(\Omega_s^v)$ for each $\alpha = 1, 2, \ldots, m$ and $n_{i_\alpha j_\alpha}(\Omega_k^u) > n_{i_\alpha j_\alpha}(\Omega_s^v)$ for each $\alpha = m+1, m+2, \ldots, q$. For the sequence (3.12), the inequalities

$$p_{i_{m+1}j_{m+1}} \le p_{i_{m+2}j_{m+2}} \le \ldots \le p_{i_qj_q}$$

have to be satisfied. Using the sequence of operations (3.12), we can present the following formula for calculating $\overline{\varrho}_s(p)$.

Theorem 5 *If G_s is an optimal digraph for the problem $\mathcal{G}//\sum \mathcal{C}_i$, then*

$$\overline{\varrho}_s(p) = \min_{k=1,2,\ldots,\lambda; k\neq s} \overline{r}_{ks}, \ \ \textit{where} \tag{3.13}$$

$$\overline{r}_{ks} = \min_{\Omega_s^v \in \Omega_{s,k}} \ \max_{u=1,2,\ldots,\omega_k} \ \max_{\beta=0,1,\ldots,q-m} \frac{\sum_{\alpha=1}^{m+\beta} p_{i_\alpha j_\alpha} |n_{i_\alpha j_\alpha}(\Omega_k^u) - n_{i_\alpha j_\alpha}(\Omega_s^v)|}{\sum_{\alpha=1}^{m+\beta} |n_{i_\alpha j_\alpha}(\Omega_k^u) - n_{i_\alpha j_\alpha}(\Omega_s^v)|}. \tag{3.14}$$

If only a subset of the processing times can be changed but the other ones cannot be changed, formulas similar to (3.13) and (3.14) can be derived (see the remark after Theorem 4). Next, we consider the case of $\overline{\varrho}_s(p) = 0$. Similarly to the notions of a critical path and a critical weight, which is important for the problem $\mathcal{G}//\mathcal{C}_{max}$ (see Section 3.3), we introduce the notions of a critical set of paths $\Omega_k^{u^*}$ and a critical sum of weights for the problem $\mathcal{G}//\sum \mathcal{C}_i$. The set $\Omega_k^{u^*}, u^* \in \{1, 2, \ldots, \omega_k\}$, is called a *critical set* if the value of the objective function

$$L_k^p = \max_{u\in\{1,2,\ldots,\omega_k\}} \sum_{\nu\in\Omega_k^u} l^p(\nu)$$

for the weighted digraph $G_k(p)$ is reached on this set:

$$\sum_{\nu\in\Omega_k^{u^*}} l^p(\nu) = \max_{u\in\{1,2,\ldots,\omega_k\}} \sum_{\nu\in\Omega_k^u} l^p(\nu) = L_k^p.$$

The value L_k^p is the *critical sum of weights* for the digraph $G_k(p)$. A critical set $\Omega_k^{u^*}$ may include a path $\nu \in H_k^i, \ i = 1, 2, \ldots, n$, if and only if $l^p(\nu) = \max_{\mu\in H_k^i} l^p(\mu)$ and so for different vectors $p \in R_+^q$ of the processing times, different sets $\Omega_k^u, \ u \in \{1, 2, \ldots, \omega_k\}$, may be critical. Let $\Omega_k(p)$ denote the set of all critical sets $\Omega_k^{u^*}$ of the digraph $G_k(p)$ for the vector $p = (p_{1,1}, p_{1,2}, \ldots, p_{nn_n}) \in R_+^q$ and let Ω_k denote the set $\{\Omega_k^u : u = 1, 2, \ldots, \omega_k\}$.

Lemma 4 *There exists a real number $\epsilon > 0$ such that the set $\Omega_k \backslash \Omega_k(p)$ contains no critical set of the digraph $G_k(p) \in \Lambda(G)$ for any processing time vector $p' \in O_\epsilon(p) \cap R^q_+$.*

The next claims define necessary and sufficient conditions for the equality $\overline{\varrho}_s(p) = 0$.

Theorem 6 *Let G_s be an optimal digraph for the problem $\mathcal{G}//\sum \mathcal{C}_i$ with positive processing times $p_{ij} > 0$ of all operations $O_{ij} \in Q$. The equality $\overline{\varrho}_s(p) = 0$ holds if and only if the following three conditions hold:*

(a) There exists another optimal schedule $k \in S^\Phi(p)$, $\Phi = \sum \mathcal{C}_i$, $k \neq s$;

(b) There exists a set $\Omega_s^{v^*} \in \Omega_s(p)$ such that for any set $\Omega_k^{u^*} \in \Omega_k(p)$, there exists an operation $O_{ij} \in Q$ for which condition $n_{ij}(\Omega_s^{v^*}) \geq n_{ij}(\Omega_k^u)$, $\Omega_k^u \in \Omega_k(p)$, holds (or condition $n_{ij}(\Omega_s^{v^*}) \leq n_{ij}(\Omega_k^u)$, $\Omega_k^u \in \Omega_k(p)$, holds);

(c) At least one of the above non-strict inequalities in (a) or (b) is satisfied as a strict inequality for the set $\Omega_k^{u^*}$.

Corollary 5 *If $s \in S$ is a unique optimal schedule for the problem $\mathcal{G}//\sum \mathcal{C}_i$, then $\overline{\varrho}_s(p) > 0$.*

Theorem 7 *If $s \in S$ is an optimal schedule for the problem $\mathcal{G}//\sum \mathcal{C}_i$ with $\lambda > 1$ and $p_{ij} > 0$ for at least one operation $O_{ij} \in Q$, then $\overline{\varrho}_s(p) \leq \max_{O_{ij} \in Q} p_{ij}$.*

Remark 2 *As follows from Theorem 7, the problem $\mathcal{J}//\sum \mathcal{C}_i$ with $\lambda > 1$ cannot have an optimal schedule with an infinitely large stability radius.*

3.5 Notes and references

Next, we present some references, where the results of this section have been obtained and discuss different aspects of a stability analysis for scheduling problems. Surveys on stability analysis in sequencing and scheduling were given in [22, 43, 44, 97, 103, 104, 109, 114, 124, 125, 126, 127]. An annotated bibliography for a stability analysis in programming and optimization was given in [39, 40]. This section was based on the definition of the stability radius of an optimal schedule being introduced in [6, 92, 93, 94, 98, 106]. Any shop-scheduling problem with a regular criterion and forbidden operation preemptions can be represented as an extremal problem on a mixed (disjunctive) graph [1, 15, 58, 80, 85, 101, 102, 133, 136]. As it was proven in [34], the set of semiactive schedules contains at least one optimal schedule provided that the criterion is regular. It should also be noted that such a digraph is more stable than the corresponding schedule with respect to variations of the job processing times. The advantage of studying the stability of an optimal digraph instead of the stability of an optimal schedule was justified in Subsection 3.2. For this reason, the mixed graph model (described in Subsection 3.1) was used for the shop-scheduling problems considered in this survey.

The papers [10, 53, 57, 98, 123] were devoted to the stability of an optimal digraph $G_s(p)$, which represents an optimal solution to the problem $\mathcal{G}//\Phi$. Subsection 3.2 was based on the papers [93, 98]. In [98], the calculation of the stability radius $\varrho_s(p)$ of an optimal schedule s was reduced to a non-linear programming problem. Formulas for calculating the stability radius of an optimal schedule with the $\mathcal{C}_{max}$ criterion and the characterization of the extreme values of $\widehat{\varrho}_s(p)$ were proven in [98]; see Subsection 3.3.

The same questions for the mean flow time criterion were investigated in [10]. Necessary and sufficient conditions for the equality $\widehat{\varrho}_s(p) = 0$ and the characterization of an infinitely large stability radius were proven in [98]. In [52, 53], it was shown that for the problem $\mathcal{J}//\mathcal{C}_{max}$, there exist necessary and sufficient conditions for $\widehat{\varrho}_s(p) = \infty$ which can be verified in $O(q^2)$ time. In [53], the analogies to Theorems 2 and 3 for the job shop problem $\mathcal{J}//\mathcal{L}_{max}$ with minimizing maximum lateness (with respect to the due dates given for the jobs $J_i \in J$) were proven. It was also shown that there does not exist an optimal schedule s with $\varrho_s(p) = \infty$ for all other regular criteria (see [58]) that are considered in scheduling theory. The extreme values of $\overline{\varrho}_s(p)$ were studied in [10, 125]. Necessary and sufficient conditions for the equality $\overline{\varrho}_s(p) = 0$ were derived in [10]; see Subsection 3.4. The stability of an optimal schedule for a job-shop problem with two jobs was investigated in [118, 119], where polynomial geometric algorithms developed in [2, 14, 42, 91, 95, 134] were used. In [75], the stability of an optimal schedule for the two-machine flow shop problem with the makespan criterion was considered.

Note that it is often more complicated to analyze $\sum \mathcal{C}_i$ than $\mathcal{C}_{max}$, e.g., the two-machine problems $\mathcal{F}2//\mathcal{C}_{max}$, $\mathcal{J}2/n_i \leq 2/\mathcal{C}_{max}$ and $\mathcal{O}2//\mathcal{C}_{max}$ are polynomially solvable (i.e., *there exist polynomial-time algorithms for them*) but the problems $\mathcal{F}2//\sum \mathcal{C}_i$ and $\mathcal{O}2//\sum \mathcal{C}_i$ are NP-hard (see, e.g., [58, 136]), which means that *up to now, there do not exist efficient (polynomial-time) algorithms for their solution and moreover, the construction of a polynomial-time algorithm for at least one of them in the future is unlikely.*

In this section, we considered the main stability analysis question: *What are the limits to the processing time changes such that the schedule at hand remains optimal?* Of course, other numerical parameters of a practical scheduling problem may also be changeable. It is easy to see that, due to the generality of the mixed graph model with any regular criterion considered in Subsection 3.1, one can analyze other changeable parameters (like *release times, due dates, deadlines, job weights, setup* and *removal times*, etc.) in terms of the mixed graph G. (E.g., the introduction of a dummy operation in G which proceeds the first operation of a job allows one to consider the *processing* time of this dummy operation as a *release* time of this job.) Nevertheless, for simplicity in what follows, we shall continue to consider the operation durations as the only changeable parameters. It is assumed that all the processing times are simultaneously and independently changeable. In order to study specific changes of different numerical parameters (e.g., the case when only one parameter is changeable) in Section 4, we introduce and study the more general notion of the so-called *relative stability radius*.

Two other sensitivity analysis questions are both of theoretical and practical importance (see [41]). *Given a specific change of one numerical parameter of a scheduling problem, what is the new optimal value of the objective function?* Answering this question identifies the effect of parameter changes on the objective function value. This answer is a solution to the *evaluation version* of a scheduling problem, while an answer to the following classical sensitivity analysis question is a solution to the *optimization version* of a scheduling problem: *Given a specific change of a numerical parameter, what is the new optimal schedule?* It is obvious that a solution to the latter question (which is a subject of Sections 2 and 3) implies a solution to the former question, however, the converse is not true. Regarding the latter question, how a schedule changes may be of interest. Similarly, as for the stability radius, both these questions may be used when several numerical parameters of a scheduling problem change simultaneously. In Section 4, we consider the problem $\mathcal{G}/a_{ij} \leq p_{ij} \leq b_{ij}/\Phi$ the solution of which gives an answer to all the above sensitivity analysis questions.

Decision-makers often consider multiple objectives when making scheduling decisions. However, very little research has been done in multiple machine environments with *multiple objectives*. Various aspects of the stability of vector discrete optimization problems were studied in the papers [29, 30, 31] and in many other papers written by V.A. Emelichev et al. In these papers and in many other papers, the stability of a vector discrete optimization problem was considered as a discrete analogy of the property of an optimal mapping specifying the Pareto choice function being upper semi-continuous in the Hausdorf sense, i.e., there exists a neighborhood in the space of the initial parameters such that new Pareto optima are impossible to arise inside it. This stability approach was initiated by V.K. Leontev [64, 65].

4 General shop with interval processing times

This section deals with general shop and job shop problems with the objective to minimize the makespan or mean flow time provided that the numerical input data are uncertain. To be more specific, we consider the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$ and the problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$. The 'strategy' of the stability approach is to separate the 'structural' input data from the 'numerical' input data. The precedence and capacity constraints (i.e., the structural input data) are given by the mixed graph G, which completely defines the set of semiactive schedules. The set of optimal semiactive schedules is defined by the weighted mixed graph $G(p)$ which presents both the structural and numerical input data. In Subsection 4.1, we define a solution of the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$ as a minimal dominant set $\Lambda^*(G)$: For any fixed vector p of the processing times with the components p_{ij} in the segments $[a_{ij}, b_{ij}]$, $O_{ij} \in Q$, there exists at least one optimal digraph in the set $\Lambda^*(G)$. Subsection 4.2 deals with the mathematical background for later presentations. In Subsection 4.3, we present the main formulas and an algorithm for solving the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$. In Subsections 4.1 – 4.3, we present an approach to deal with the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$ based on an improved stability analysis of an optimal schedule. In the course of this section, an *optimal* schedule (digraph), a *better* and a *best* schedule (digraph) are considered with respect to the criterion $\mathcal{C}_{max}$ or $\sum \mathcal{C}_i$.

4.1 Minimal dominant set (MDS)

Let the input data of a general shop scheduling problem be presented by a mixed graph $G = (Q, A, E)$ introduced in Section 3.1. It is worthwhile to note that the *feasibility* of a schedule s and that of a weighted digraph $G_s(p) \in \Lambda(G))$ are independent of the vector $p = (p_{1,1}, p_{1,2}, \dots, p_{nn_n})$ of the processing times, while the *optimality* of a weighted digraph $G_s(p)$ depends on the vector p. Actually, the set $\Lambda(G) = \{G_1, G_2, \dots, G_\lambda\}$ of feasible digraphs is completely defined by the mixed graph $G = (Q, A, E)$ (without the weights p) while the information on the vector p of the processing times is needed to determine whether a schedule $k \in S$ is optimal or not.

If the processing time vector p is not known before scheduling, different schedule realizations may result in different critical paths in the digraph $G_s(p)$. For practical problems, the cardinality λ of the set $\Lambda(G)$ may be huge. However, one need to consider only a subset B of the set $\Lambda(G) : B \subseteq \Lambda(G)$. Since $p_{ij} \ge 0$ for all $O_{ij} \in Q$, we obtain $\max_{O_{ij} \in Q} c_{ij}(s) = \max_{\mu \in H_s} l^p(\mu)$ and from equality (3.9), it follows that the digraph $G_s(p)$ has the minimal critical weight within the set $B \subseteq \Lambda(G)$ if and only if

$$\max_{\mu \in H_s} l^p(\mu) = \min_{G_k \in B} \max_{\nu \in H_k} l^p(\nu). \tag{4.1}$$

For the case $B = \Lambda(G)$, the equality (4.1) provides an *optimality* criterion for a schedule $s \in S$ (if p is fixed). The problem $\mathcal{G}//\mathcal{C}_{max}$ is a special case of the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$. We call the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$ uncertain in contrast to the problem $\mathcal{G}//\mathcal{C}_{max}$ called deterministic.

The stability approach for solving the uncertain problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$ is based on an improved stability analysis of an optimal digraph. As follows from Section 3, an optimal digraph $G_s \in \Lambda(G)$ provides a solution of the deterministic problem $\mathcal{G}//\mathcal{C}_{max}$. Since the vector $p \in T$ of processing times is unknown in the uncertain problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$, the completion time of the jobs $J_i \in J$ cannot be calculated before scheduling. Therefore, mathematically, the uncertain problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$ is not correct.

In the OR literature, different approaches for correcting an uncertain problem like $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$ have been used. In particular, an additional criterion of minimizing the maximal regret is introduced to make the uncertain problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$ correct in the robust approach. In this section, another approach is proposed for solving the uncertain problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$. The proposed approach is based on a *minimal dominant set* of schedules (digraphs), which may be considered as a solution to such an uncertain problem.

Definition 3 *A set of digraphs $\Lambda^*(G) \subseteq \Lambda(G)$ is called a dominant set of the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$ if for each fixed vector $p \in T$ of the processing times the set $\Lambda^*(G)$ contains at least one optimal digraph. If any proper subset of the set $\Lambda^*(G)$ is no longer a dominant set of the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$, it is a minimal dominant set $\Lambda^T(G)$, which is called a G-solution for the above uncertain problem.*

To find a G-solution for the problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$, we need to determine the largest ball within which digraph G_s is 'the best' for a subset B of the set $\Lambda(G)$. In Subsection 4.2, we propose another definition (more general than Definition 1) of the stability radius (this stability radius $\widehat{\varrho}_s^B(p \in T)$ is called relative). The relative stability radius is used for solving the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$ in the sense of Definition 3. The formulas (3.10) and (3.11) being valid for the case of calculating the stability radius with $0 \le p_{ij} < \infty$, $O_{ij} \in Q$, are generalized to the case when the variations of the processing times are restricted by lower and upper bounds (as in Condition 5) and some feasible digraphs have to be excluded from the comparisons with 'the best' one.

4.2 Relative stability radius

In Section 3, the stability radius $\widehat{\varrho}_s(p)$ of an optimal digraph G_s has been investigated which denotes the largest quantity of independent variations of the processing times p_{ij} of the operations $O_{ij} \in Q$ within the interval $[0, \infty)$ such that the digraph G_s remains 'the best' (i.e., the weighted digraph $G_s(p)$ has the minimal critical weight) among all feasible digraphs $\Lambda(G)$ (see Definition 1). The following generalization of the stability radius $\widehat{\varrho}_s(p)$ (we call it *relative* stability radius) is defined by considering the closed interval $[a_{ij}, b_{ij}]$ instead of $[0, \infty)$ and considering a set $B \subseteq \Lambda(G)$ instead of the whole set $\Lambda(G)$. In Definition 4, l_s^p denotes the critical weight of the weighted digraph $G_s(p)$, $p \in T$.

Definition 4 *Let the digraph $G_s \in B \subseteq \Lambda(G)$ have the minimal critical weight $l_s^{p'}$ for each vector $p' \in O_\varrho(p) \cap T$ among all digraphs from the given set B: $l_s^{p'} = \min\{l_k^{p'} : G_k \in B\}$. The* *maximal value of the radius ϱ of such a ball $O_\varrho(p)$ is denoted by $\widehat{\varrho}_s^B(p \in T)$ and is called the relative stability radius of the digraph G_s with respect to polytope T.*

Note that the relativity of $\widehat{\varrho}_s^B(p \in T)$ is defined not only by the polytope T of feasible vectors, but also by the set B of feasible digraphs. From Definition 1 and Definition 4, it follows: $\widehat{\varrho}_s(p) = \widehat{\varrho}_s^{\Lambda(G)}(p \in R_+^q)$. Thus, the relative stability radius is equal to the *maximal error* of the given processing times p_{ij} $(a_{ij} \leq p_{ij} \leq b_{ij},\ O_{ij} \in Q)$ within which the 'superiority' of the digraph G_s is still preserved over the given subset B of feasible digraphs. The following two extreme cases of the relative stability radius are of particular importance for solving problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$. On the one hand, if for any positive real number $\epsilon > 0$ which may be as small as desired, there exist a vector $p' \in O_\epsilon(p) \cap T$ and a digraph $G_k \in B$ such that $l_s^{p'} > l_k^{p'}$, we obtain a zero relative stability radius: $\widehat{\varrho}_s^B(p \in T) = 0$. On the other hand, if $l_s^{p'} \leq l_k^{p'}$ for any vector $p' \in T$ and for any digraph $G_k \in B$, we obtain an infinitely large relative stability radius: $\widehat{\varrho}_s^B(p \in T) = \infty$. Note that even in the case of finite upper bonds $(b_i < \infty,\ i \in Q)$, i.e., when the maximal error of the processing time p_i for each operation $i \in Q$ is restricted by

$$\epsilon_{max} = \max\{\{p_{ij} - a_{ij},\ b_{ij} - p_{ij}\} : O_{ij} \in Q\}, \tag{4.2}$$

the value of $\widehat{\varrho}_s^B(p \in T)$ may be infinitely large as it follows from Definition 4. The deterministic problem $\mathcal{G}//\mathcal{C}_{max}$ with the vector p of processing times and the optimal digraph G_s provides such a trivial example with an infinitely large relative stability radius $\widehat{\varrho}_s^B(p \in T)$. Indeed, if $a_i = p_i = b_i$ for each operation $i \in Q$, then the polytope T degenerates into a single point: $T = \{p\}$ and therefore, from the inclusion $p' \in O_\varrho(p) \cap T$, it follows that the vector p' mentioned in Definition 4 is definitely equal to vector p, for which the digraph G_s is optimal. To characterize the extreme values of $\widehat{\varrho}_s^B(p \in T)$, we define the following binary relation which generalizes the dominance relation used in Section 3.

Definition 5 *The path ν dominates the path μ in the set T if and only if for any vector $x = (x_{1,1}, x_{1,2}, \ldots, x_{nn_n}) \in T$ the following inequality holds:*

$$l^x(\mu) \leq l^x(\nu). \tag{4.3}$$

The following lemma gives a criterion for the above dominance relation.

Lemma 5 *The path ν dominates the path μ in the set T if and only if the following inequality holds:*

$$\sum_{O_{ij} \in [\mu] \setminus [\nu]} b_{ij} \leq \sum_{O_{uv} \in [\nu] \setminus [\mu]} a_{uv}. \tag{4.4}$$

Definition 6 *The set of paths H_k dominates the set of paths H_s in T if and only if for any path $\mu \in H_s$, there exists a path $\nu \in H_k$, which dominates the path μ in the set T.*

Lemma 6 gives a condition when a domination does not hold.

Lemma 6 *The set of paths H_k does not dominate the set of paths H_s in T if there exists a path $\mu \in H_s$ such that the system*

$$\begin{cases} \sum_{O_{ij} \in [\nu] \setminus [\mu]} a_{ij} < \sum_{O_{uv} \in [\mu] \setminus [\nu]} b_{uv}, \\ a_{ij} \leq x_{ij} \leq b_{ij},\ O_{ij} \in Q, \end{cases}$$

has a solution for any path $\nu \in H_k$.

If $H_k = H_k(p)$, we have $H_k(p') \subseteq H_k = H_k(p)$ for any vector $p' \in R^q_+$ of the processing times. The following lemma shows that the set of critical paths is not expanded for small variations of the processing times.

Lemma 7 *If $H_k \neq H_k(p)$, the inclusion $H_k(p') \subseteq H_k(p)$ holds for any vector $p' \in O_\epsilon(p) \cap R^q_+$ with the real number $\epsilon_k > \epsilon > 0$ defined as follows:*

$$\epsilon_k = \frac{1}{q}\left[l^p_k - \max\{l^p(\nu) : \nu \in H_k \setminus H_k(p)\}\right].$$

Theorem 8 *Let the digraph G_s have the minimal critical weight l^p_s, $p \in T$, within the given subset $B \subseteq \Lambda(G)$ of feasible digraphs. Then equality $\widehat{\varrho}^B_s(p \in T) = 0$ holds if and only if there exists a digraph $G_k \in B$ such that $l^p_s = l^p_k$, $k \neq s$, and the set of paths $H_k(p)$ does not dominate the set of paths $H_s(p)$ in T.*

Corollary 6 *If $G_s \in B$ is the unique optimal digraph for the vector $p \in T$ of processing times, then $\widehat{\varrho}^B_s(p \in T) > 0$.*

Theorem 8 identifies a digraph $G_s \in \Lambda(G)$ whose 'superiority' within the set B is *unstable*: Even a very small change in the processing times can make another digraph from the set B to be 'better' than G_s. The following theorem identifies a digraph G_s whose 'superiority' within the set B in the polytope T is 'absolute': Any changes of the processing times within the polytope T cannot make another digraph from the set B to be 'better' than the digraph G_s.

Theorem 9 *For the digraph $G_s \in B$, $\widehat{\varrho}^B_s(p \in T) = \infty$ if and only if for any digraph $G_t \in B$, $t \neq s$, the set of paths H_t dominates the set of paths $H_s \setminus H$ in T.*

Corollary 7 *If $\widehat{\varrho}^B_s(p \in T) < \infty$, then $\widehat{\varrho}^B_s(p \in T) \leq \epsilon_{max}$, where the value ϵ_{max} is calculated according to (4.2).*

In the following subsection, we use Theorem 9 as a stopping rule in the algorithm developed for solving the problem $\mathcal{G}/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ since the optimality of the digraph $G_s \in B$ with $\widehat{\varrho}^B_s(p \in T) = \infty$ does not depend on the vector $p \in T$.

4.3 Algorithms for the problem $\mathcal{G}/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$

From Subsections 4.1 and 4.2, it follows that a G-solution for the uncertain problem $\mathcal{G}/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ may be obtained on the basis of a repeated calculation of the relative stability radii $\widehat{\varrho}^B_s(p \in T)$. Thus, we need formulas for calculating $\widehat{\varrho}^B_s(p \in T)$. In Subsection 3.3, formulas (3.10) and (3.11) were presented for calculating the stability radius $\widehat{\varrho}_s(p) = \widehat{\varrho}^{\Lambda(G)}_s(p \in R^q_+)$ (see Theorem 4 on page 14). Theorem 10, which follows, generalizes these formulas for any given subset $B \subseteq \Lambda(G)$ of feasible digraphs and for any given polytope $T \subseteq R^q_+$ of feasible vectors of the processing times. To present the new formulas, we need the following notations. If μ and ν are paths in the digraphs from the set $\Lambda(G)$, then $[\mu] + [\nu]$ denotes the *symmetric difference* $[\mu] \cup [\nu] \setminus [\mu] \cap [\nu]$ of the sets $[\mu]$ and $[\nu]$. We calculate the values $\Delta^{ij}(\mu, \nu)$ equal to $b_{ij} - p_{ij}$, if $O_{ij} \in [\mu] \setminus [\nu]$, and equal to $p_{ij} - a_{ij}$, if $O_{ij} \in [\nu] \setminus [\mu]$. Let $\Delta^{ij}_0(\mu, \nu)$ be equal to zero. We order the set of values $\Delta^{ij}(\mu,\nu)$ for the operations O_{ij} from the symmetric difference $[\mu]+[\nu]$ in the following way:

$$\Delta_1^{i_1j_1}(\mu,\nu) \le \Delta_2^{i_2j_2}(\mu,\nu) \le \ldots \le \Delta_{|[\mu]+[\nu]|}^{i_{|[\mu]+[\nu]|}j_{|[\mu]+[\nu]|}}(\mu,\nu), \tag{4.5}$$

where the subscript $k \in \{1,2,\ldots,[\mu]+[\nu]\}$ indicates the location of $\Delta_k^{i_kj_k}(\mu,\nu)$ in the sequence (4.5), and the superscript i_kj_k indicates the operation $O_{i_kj_k} \in [\mu]+[\nu]$ for which the value $\Delta_k^{i_kj_k}(\mu,\nu)$ is calculated. For any two feasible digraphs G_s and G_k, we introduce the following set of paths:

$$H_{sk}(T)=\{\mu \in H_s : \textit{There is no path } \nu \in H_k \textit{ which dominates the path } \mu \textit{ in the polytope } T\}.$$

Theorem 10 *If the digraph G_s has the minimal critical weight l_s^p, $p \in T$, in the given set $B \subseteq \Lambda(G)$ of feasible digraphs, then*

$$\widehat{\varrho}_s^B(p \in T) = \min_{G_k \in B} \widehat{r}_{ks}^B, \tag{4.6}$$

$$\widehat{r}_{ks}^B = \min_{\mu \in H_{sk}(T)} \ \max_{\nu \in H_k,\ l^p(\nu) \ge l_s^p} \ \max_{\beta=0,1,\ldots,|[\mu]+[\nu]|-1} \frac{l^p(\nu) - l^p(\mu) - \sum_{\alpha=0}^{\beta} \Delta^{i_\alpha j_\alpha}(\mu,\nu)}{|[\mu]+[\nu]| - \beta}. \tag{4.7}$$

Remark 3 The formulas (4.6) and (4.7) defined in Theorem 10 turn into $\widehat{\varrho}_s^B(p \in T) = \infty$ if $H_{sk}(T) = \emptyset$ for each digraph $G_k \in B$.

The problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$ may be solved as follows. Let B denote a subset of feasible digraphs which contains a G-solution $\Lambda^*(G)$ for the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$. On the basis of the algorithm, which follows, we can expand the set $\Lambda' \subseteq \Lambda^*(G)$ starting with $\Lambda' = \emptyset$ and finishing with $\Lambda' = \Lambda^*(G)$.

Algorithm $SOL\_\mathcal{C}_{max}(1)$

Input: Set $\Lambda(G)$ of feasible digraphs, polytope T of the processing times.
Output: G-solution $\Lambda^*(G)$ of the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$.

Step 1: Find a set $B \subseteq \Lambda(G)$ such that $\Lambda^*(G) \subseteq B$.
Step 2: Set $\Lambda' = \emptyset$.
Step 3: Fix the vector p of processing times, $p \in T$.
Step 4: Find an optimal digraph $G_s(p) \in B$ for the problem $\mathcal{G}//\mathcal{C}_{max}$ with the vector p of processing times.
Step 5: Calculate the relative stability radius $\widehat{\varrho}_s^B(p \in T)$.
Step 6: **IF** $\widehat{\varrho}_s^B(p \in T) < \infty$ and $B \setminus \{G_s\} \ne \emptyset$ **THEN**
BEGIN
Step 7: Select a digraph $G_k(p) \in B$ which is competitive to digraph $G_s(p)$.
Step 8: Find a vector $p^* \in T$ of the processing times closest to p such that $l_s^{p^*} = l_k^{p^*}$ and for any small $\epsilon > 0$, there exists a vector p^ϵ with $l_s^{p^\epsilon} > l_k^\epsilon$ and $d(p^*, p^\epsilon) \le \epsilon$.
Step 9: Set $\Lambda' := \Lambda' \cup \{G_s\}$, $B := B \setminus \{G_s\}$, $s = k$; $p = p^*$ **GOTO** *Step 5*
END
Step 10: **ELSE** $\Lambda^*(G) = \Lambda' \cup \{G_s\}$ **STOP**

We substantiate some steps of Algorithm $SOL\_\mathcal{C}_{max}(1)$. In Step 1, the determination of the set $B = \Lambda(G)$ of all feasible digraphs by an explicit enumeration is possible only for a small number of edges in the mixed graph G. Using the simple bound (4.8), one can considerably restrict the number of feasible digraphs, with which a comparison of an optimal digraph G_s has to be done while calculating the relative stability radius $\widehat{\varrho}_s^B(p \in T)$. For a large cardinality of the set E, one can use a branch-and-bound algorithm for the construction of the k best digraphs. In Step 3, we have to fix the processing times as any vector from the set T. For example, we can use a 'historical' vector p of the processing times which helps to simplify Steps 3, 4 or 5. Step 4 may be realized by an explicit enumeration or by an implicit enumeration (e.g., by a branch-and-bound method) of the feasible digraphs B. In Step 4, we can apply Theorem 8 to guarantee that the selected optimal digraph G_s is stable. If $\widehat{\varrho}_s^B(p \in T) = 0$, we can take another optimal digraph (the latter exists due to Theorem 8) which is stable, or we can change the initial vector p of the processing times. Steps 5, 7, and 8 may be done on the basis of Theorem 9 or Theorem 10. If $\widehat{\varrho}_s^B(p \in T) = \infty$, Theorem 9 is used as a 'stopping rule' of the algorithm, otherwise, Theorem 10 is used which is time-consuming. A competitive digraph and a new vector p^* of the processing times are calculated in Algorithm $SOL\_\mathcal{C}_{max}(1)$ in parallel with the calculation of the relative stability radius $\widehat{\varrho}_s^B(p \in T)$. Note that a competitive digraph is not necessarily uniquely determined, and one can take any of them. Steps 5 and 7 are rather complicated. In Algorithm $SOL\_\mathcal{C}_{max}(1)$ we must anew construct a set $H_{sk}(T)$ in each iteration based on a direct comparison of the paths in a new optimal digraph G_s and in each other digraph G_k from the set B, so it may be very time-consuming.

We present Algorithm $SOL\_\mathcal{C}_{max}(2)$, which may be more efficient. This algorithm focuses on one of the optimal digraphs G_1 and on one vector $p \in T$. Let $\{\Gamma_i : i = 1, 2, \dots, I\}$ be the set of competitive digraphs of the digraph G_1 with respect to the set B, where i is a counter of the current iteration and I is the whole number of iterations.

Algorithm $SOL\_\mathcal{C}_{max}(2)$

Input: Set $\Lambda(G)$ of feasible digraphs, polytope T of the processing times.
Output: G-solution $\Lambda^*(G)$ of the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$.

Step 1: Find a set $B \subseteq \Lambda(G)$ such that $\Lambda^*(G) \subseteq B$.
Step 2: Set $\Lambda' = \emptyset$; $i = 1$ and $\Gamma_i = \emptyset$.
Step 3: Fix the vector p of processing times, $p \in T$.
Step 4: Find an optimal digraph $G_1(p) := G_s(p) \in B$ for the problem $\mathcal{G}//\mathcal{C}_{max}$ with the vector p of processing times.
Step 5: Calculate $\widehat{\varrho}_1^B(p \in T)$.
Step 6: **IF** $\widehat{\varrho}_1^B(p \in T) < \infty$ **THEN**
BEGIN
Step 7: Select the set of competitive digraphs Γ_i of the digraph $G_1(p)$ with respect to the set B.
Step 8: Set $\Lambda' := \Lambda' \cup \Gamma_i$, $B := B \setminus \Gamma_i$, $i := i + 1$. **GOTO** *Step 5*
END
Step 10: **ELSE** $\Lambda^*(G) := \Lambda' \cup \{G_1\}$ **STOP**

Using Algorithm $SOL\_\mathcal{C}_{max}(2)$, we construct an increasing sequence of the relative stability radii $\widehat{\varrho}_1 < \widehat{\varrho}_2 < \ldots < \widehat{\varrho}_I$ of the stability balls $O_{\widehat{\varrho}_i}(p)$, $i \in \{1, 2, \dots, I\}$, with the same center $p \in T$ and different sets of feasible digraphs $B = \Lambda(G) \setminus \bigcup_{j=1}^{i} \Gamma_j$. As a result, we construct a sequence of 'nested sets' of the competitive digraphs

$$\Gamma_1,\ \Gamma_1 \bigcup \Gamma_2,\ \ldots,\ \bigcup_{i=1}^{I} \Gamma_i$$

of the digraph G_1, where the set $\{G_1\} \bigcup \{\bigcup_{i=1}^{I} \Gamma_i\}$ is a G-solution $\Lambda^*(G)$ to the scheduling problem on the mixed graph $(Q, A \cup E_1, \emptyset)$, and G_1 is one of the optimal digraphs in the set $\Lambda(G)$ for the vector $p \in T$ of processing times. The most difficult part of Algorithm $SOL\_\mathcal{C}_{max}(2)$ is to find the stability radius $\widehat{\varrho}_1^B(p \in T)$ (Step 5 and Step 6) and to find the sets of competitive digraphs (Step 7). However, one can use the following remark.

Remark 4 It is not necessary to perform Steps 1 - 10 since one can construct a G-solution $\Lambda^*(G)$ in one scan. Namely, from Remark 3, it follows that all digraphs $G_k, k \neq 1$, for which a set $H_{1k}(T) \neq \emptyset$ was constructed in Step 5 are united with the optimal digraph G_1 and compose a G-solution:

$$\Lambda^*(G) = \{G_1\} \bigcup \{\bigcup_{i=1}^{I} \Gamma_i\} = \{G_1\} \bigcup \{G_k :\ H_{1k}(T) \neq \emptyset\}.$$

Thus, one can use the software developed for the problems discussed in Section 3 with the following modification: We add the loop with Steps 6 – 9. An increasing sequence of the relative stability radii of the stability balls with the same center $p \in T$ corresponds to an increasing sequence of the values $\widehat{r}_{k1}^B$ calculated by (4.7) for the optimal digraph $G_1(p)$ in Step 5. A competitive digraph (or a set of competitive digraphs Γ_i) of the digraph $G_1(p)$ is constructed in one scan as well.

Remark 5 For both algorithms, fixing the initial vector p in Step 3 and the choice of an optimal digraph $G_s(p)$ in Step 4 (and also in Step 7 for Algorithm $SOL\_\mathcal{C}_{max}(1)$) have a large influence on the further calculations and the resulting G-solution.

Next, we show how to restrict the number of digraphs G_k (the cardinality of the set B) with which an optimal digraph G_s has to be compared in the process of the calculation of the relative stability radius $\widehat{\varrho}_s^B(p \in T)$.

Due to formulas (4.6) on page 22, the calculation of the relative stability radius is reduced to a complicated calculation on the set of digraphs $B \subseteq \Lambda(G)$. The main objects for the calculation of $\widehat{\varrho}_s^B(p \in T)$ are the sets of paths in the digraphs $G_k \in B$. In the worst case, the calculation of $\widehat{\varrho}_s^B(p \in T)$ implies to have an optimal digraph G_s and to construct all digraphs from the subset B of the set $\{G_1, G_2, \ldots, G_\lambda\}$. In order to restrict the number of digraphs G_k with which a comparison of the optimal digraph G_s has to be done during the calculation of $\widehat{\varrho}_s^B(p \in T)$, one can use the upper bound $\widehat{\varrho}_s^B(p \in T) \leq \widehat{r}_{ks}^B$ on the relative stability radius, where $\widehat{r}_{ks}^B$ is defined according to formula (4.7) on page 22.

Lemma 8 *If $\widehat{\varrho}_s^B(p \in T) < \infty$ and there exists a digraph $G_k \in B$ such that*

$$\widehat{r}_{ks}^B \leq \frac{l_t^p - l_s^p}{q} \tag{4.8}$$

for a digraph $G_t \in B$, then it is not necessary to consider the digraph G_t during the calculation of the relative stability radius $\widehat{\varrho}_s^B(p \in T)$.

Corollary 8 *Let the set* $B = \{G_s = G_{i_1}, G_{i_2}, \ldots, G_{i_{|B|}}\}$ *be sorted in non-decreasing order of the objective function values:*

$$l^p_{i_1} \le l^p_{i_2} \le \ldots \le l^p_{i_{|B|}}.$$

If for the currently compared digraph G_{i_k} *from the set* $B \subseteq \Lambda(G)$*, inequality*

$$\widehat{r}^{B}_{i_k s} \le \frac{l^p_{i_t} - l^p_{i_1}}{q}$$

holds for the digraph $G_{i_t} \in B$ *with* $l^p_{i_k} \le l^p_{i_t}$*, then it is possible to exclude the digraphs* $G_{i_t}, G_{i_{t+1}}, \ldots, G_{i_{|B|}}$ *from further considerations during the calculation of the relative stability radius* $\widehat{\varrho}^B_s(p \in T)$.

Using Corollary 8, one can compare the optimal digraph $G_s = G_{i_1}$ consecutively with the digraphs $G_{i_2}, G_{i_3}, \ldots, G_{i_{|B|}}$ from the set B in non-decreasing order of the objective function values: $l^p_{i_1} \le l^p_{i_2} \le \ldots \le l^p_{i_{|B|}}$. If for the currently compared digraph $G_k = G_{i_r}$ inequality (4.8) holds, one can exclude the digraphs $G_{i_r}, G_{i_{r+1}}, \ldots, G_{i_{|B|}}$ from further considerations. The bound (4.8) is tight. Since $\widehat{\varrho}_s(p) = \widehat{\varrho}^{\Lambda(G)}_s(p \in R^q_+)$, Corollary 8 implies Corollary 9 which allows us to restrict the number of feasible digraphs while calculating the stability radius $\widehat{\varrho}_s(p)$ (see Definition 1 on page 10).

Corollary 9 *Let the set* $\Lambda(G) = \{G_s = G_{i_1}, G_{i_2}, \ldots, G_{i_\lambda}\}$ *be sorted in non-decreasing order of the objective function values:*

$$l^p_{i_1} \le l^p_{i_2} \le \ldots \le l^p_{i_\lambda}.$$

If for the currently compared digraph G_{i_k} *from the set* $\Lambda(G) = \{G_s = G_{i_1}, G_{i_2}, \ldots, G_{i_k}, \ldots, G_{i_t}, \ldots G_{i_\lambda}\}$*, inequality*

$$\widehat{r}^{\Lambda(G)}_{i_k s} \le \frac{l^p_{i_t} - l^p_{i_1}}{q}$$

holds for the digraph $G_{i_t} \in \Lambda(G)$ *with* $l^p_{i_k} \le l^p_{i_t}$*, then it is possible to exclude the digraphs* $G_{i_t}, G_{i_{t+1}}, \ldots, G_{i_\lambda}$ *from further considerations during the calculation of the stability radius* $\widehat{\varrho}_s(p)$.

4.4 Notes and references

In [79], it was noted that one "source of uncertainty is processing times, which, typically, are not known in advance. Thus, a good model of a scheduling problem would need to address these forms of uncertainty." The results presented in this section were proven and published in [55, 56, 57, 125, 128, 129, 131]. The proof of the lower bound for $\widehat{\varrho}^B_s(p \in T)$ can be found in [55]. The calculation of $\widehat{\varrho}_s(p) = \widehat{\varrho}^{\Lambda(G)}_s(p \in R^q_+)$ was proposed in [98]. In [123], a bound on the stability radius was used to restrict the number of digraphs considered for calculating the stability radius (these results were given in Subsection 2.4). The main results presented in this section were published in [55, 123]. In [128, 131], an approach for dealing with 'strict uncertainty' based on a stability analysis of an optimal semiactive schedule was generalized to an uncertain job shop problem with any given regular criterion Φ. Necessary and sufficient conditions for a G-solution of the problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\sum \mathcal{C}_i$ were derived in [56]. A theorem proven in [56] characterizes a single-element G-solution of the problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\Phi$, which is necessarily a minimal G-solution. A similar theorem formulated for the problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\sum \mathcal{C}_i$ was proven in [55]. Similar theorems and lemmas formulated for the problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\sum \mathcal{C}_i$ were proven in [57].

Since the optimality of a schedule s depends on the critical path in the digraph G_s, we focused on the set of paths in the digraph G_s which may be critical (see Lemma 7 and Theorem 8). To restrict the set of paths which may be critical, one can use a dominance relation on the set of paths (see Definition 2). Although this relation is based only on the structural input data, its use may considerably reduce the set of paths which may be critical. To deal with the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$ in Subsection 4.2, we generalized the dominance relation (see Definition 5) due to the numerical input data as well. On the basis of this dominance relation, we presented a characterization of a zero relative stability radius (Theorem 8) and an infinite relative stability radius (Theorem 9). In Subsection 4.3, we have given a formula for calculating the exact value of the relative stability radius (Theorem 10). These results may be considered as a mathematical background for developing algorithms for solving the problem $\mathcal{G}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$.

The network presentation of the structural input data (precedence and capacity constraints) and a minimal G-solution were discussed in Section 3, where the decision process was presented as the construction of a set of schedules (digraphs) which dominate other schedules. To solve the problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\sum \mathcal{C}_i$, we developed an approach for calculating the relative stability radius. In [10], the stability radius $\overline{\varrho}_s^B(p \in T)$ was investigated for the case when $B = \Lambda(G)$ and the whole space R_+^q being used instead of the polytope T. Upper bounds for $\widehat{\varrho}_s^B(p \in T)$ and $\overline{\varrho}_s^B(p \in T)$ were used to restrict the number of digraphs compared with an optimal digraph for calculating the relative stability radii. These bounds were derived for $\widehat{\varrho}_s(p)$ and $\overline{\varrho}_s(p)$ in [123].

Lai and Sotskov [54] used a weighted mixed graph G for representing the input data of a job shop problem which implies a one-to-one correspondence between the set of semiactive schedules S and circuit-free digraphs $\Lambda(G)$. Since the optimality of a schedule $s \in S$ for the makespan criterion depends on the critical path in the corresponding digraph G_s, the analysis in [54] was focused on the set of paths in $G_s \in \Lambda(G)$ which may be critical. In [54], the critical path method [23] was modified for constructing a minimal digraph containing only possible candidates of critical paths. A minimal set of makespan optimal schedules for uncertain numerical input data was characterized in [54], where an exact and a heuristic algorithm were developed for problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$. Note that the approach developed in [54] is based on the stability property of a makespan optimal schedule, which was theoretically investigated in [53, 98, 127] and in some other papers (see [114, 125] with surveys of a stability analysis for scheduling problems). Briefly, the main issue of the research presented in [54] was to simplify the digraph G_s due to the existence of two types of dominance relations between its paths; see Subsection 4.2. A formula for calculating the stability radius of an optimal schedule was provided in [55] for the job-shop problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$. The stability radius of an optimal schedule was investigated in [53, 123, 127] for the problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$ and in [10] for the problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\sum \mathcal{C}_i$. In the work of [55], the stability analysis of a schedule minimizing the total flow time was exploited in a branch-and-bound method for solving the job-shop problem $\mathcal{J}/a_{ij} \le p_{ij} \le b_{ij}/\sum \mathcal{C}_i$.

In the PhD thesis [130], a further step in this direction was performed by focusing on two types of dominance relations between feasible digraphs (schedules). This step is useful for shop scheduling problems under 'strict uncertainty' with both C_{max} and $\sum C_i$ criteria since it allows one to reduce significantly the number of schedules which are sufficient to be considered as candidates for a G-solution. However, for the $\sum C_i$ criterion, this step seems to be even more important since the comparison of digraphs (schedules) with respect to $\sum C_i$ is essentially more complicated than that for C_{max} (see [10, 123]). Note also that, while the simplification of a feasible digraph G_s may be done in polynomial time [54], to find a G-solution of problem $\mathcal{J}/a_{ij} \leq p_{ij} \leq b_{ij}/\sum \mathcal{C}_i$ or problem $\mathcal{J}/a_{ij} \leq p_{ij} \leq b_{ij}/C_{max}$ is NP-hard. The minimization of a project network with given bounds of the activity durations was studied in [117]. This approach seems to be particularly useful when the structural input data are fixed before applying a scheduling algorithm but the numerical input data are uncertain, especially when a lot of scheduling problems with the same (or close) structural input data have to be solved.

As follows from [7, 8, 9], a reduction of the digraphs may be essential even for all non-negative perturbations of the processing times: $0 \leq p_{ij} < \infty$. Bräsel et al. [7], Bräsel et al. [8] and Bräsel and Kleinau [9] introduced the set of so-called *'irreducible' schedules* for a classical job shop problem $\mathcal{J}//\mathcal{C}_{max}$ and for an open shop problem $\mathcal{O}//\mathcal{C}_{max}$: For any processing times, this set contains at least one optimal schedule. On the basis of computations for $n \leq 3$ and $m \leq 7$, it was shown that only a relatively small part of semiactive schedules is irreducible for an open shop, and this part becomes even relatively smaller when the size of the problem grows. By computational experiments [7], it was demonstrated that the hardness of a classical job shop problem essentially depends on the cardinality of the set of irreducible schedules. Using the above extension of the three-field notation, we can say that the classical job shop problem $\mathcal{J}/0 \leq p_{ij} < \infty/\mathcal{C}_{max}$ was a subject of [7] and the open shop problem $\mathcal{O}/0 \leq p_{ij} < \infty/\mathcal{C}_{max}$ was a subject of [7, 9].

We have to emphasize that the random processing times $p_{ij}, O_{ij} \in Q$, in the problem $\mathcal{G}/a_{ij} \leq p_{ij} \leq b_{ij}/\Phi$ are due to external forces in contrast to scheduling problems with *controllable* processing times, see e.g. [45, 47, 132, 137], where the objective is to choose both the optimal processing times (which are under the control of a decision-maker) and an optimal schedule for the chosen processing times. Both of the above parts of a G-solution are supposed to be arguments in the objective function which is non-decreasing in the job completion times and non-increasing in the operation processing times. The objective is to choose suitable processing times in order to minimize a given function trading off between the profits due to the reduction of the makespan [45, 47, 132, 137] (or mean flow time [137]) and the costs for increasing the machine speeds.

5 Two-stage scheduling

First, we consider the following two-stage job-shop scheduling problem denoted as $\mathcal{J}2/n_i \leq 2/\mathcal{C}_{max}$, where n_i denotes the number of processing stages. Let a set $\mathcal{J}$ of n jobs $\mathcal{J} = \{J_1, J_2, ..., J_n\}$ have be processed in a job-shop with two different machines $\mathcal{M} = \{M_1, M_2\}$. Each of the two machines $M_j \in \mathcal{M}$ can process a job $J_i \in \mathcal{J}$ no more than once provided that operation preemptions are not allowed. Let $\mathcal{J}_{12} \subseteq \mathcal{J}$ denote the subset of all jobs with the machine route (M_1, M_2) (i.e., job $J_i \in \mathcal{J}_{12}$ must be processed on machine $M_1 \in \mathcal{M}$ and then on machine $M_2 \in \mathcal{M}$). Let $\mathcal{J}_{21} \subseteq \mathcal{J}$ denote the subset of jobs with the opposite machine route (M_2, M_1) and $\mathcal{J}_m \subseteq \mathcal{J}$ denote the set of jobs that have to be processed on exactly one machine $M_j \in \mathcal{M}$. We obtain $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{12} \cup \mathcal{J}_{21}$ and denote $n_h = |\mathcal{J}_h|$, where $h \in \{1, 2, 12, 21\}$. All n jobs are available for processing from time $t = 0$. Let O_{ij} denote the operation of the job $J_i \in \mathcal{J}$ processed on the machine $M_j \in \mathcal{M}$. Let the processing time p_{ij} of the operation O_{ij} be fixed before scheduling. The criterion C_{max} under consideration denotes the minimization of the schedule length (makespan) defined as follows: $C_{max} = \min_s C_{max}(s) = \min_s\{\max\{C_j(s) \mid J_i \in \mathcal{J}\}\}$, where $C_j(s)$ denotes the completion time of the job $J_i \in \mathcal{J}$ in the schedule s. Jackson [46] proved that the problem $\mathcal{J}2/n_i \leq 2/\mathcal{C}_{max}$ is polynomially solvable and an optimal schedule for this problem may be defined by a pair of permutations (π', π'') (we call them a *Jackson pair of permutations* [48]), where π' is the sequence of the jobs $\mathcal{J}_1 \cup \mathcal{J}_{12} \cup \mathcal{J}_{21}$ processed on the machine M_1 and π'' is a sequence of the jobs $\mathcal{J}_2 \cup \mathcal{J}_{12} \cup \mathcal{J}_{21}$ processed on the machine M_2. Namely, $(\pi' = (\pi_{12}, \pi_1, \pi_{21}), \pi'' = (\pi_{21}, \pi_2, \pi_{12}))$, where job J_j belongs to the permutation π_h if and only if the inclusion $J_j \in \mathcal{J}_h$ holds, where $h \in \{1, 2, 12, 21\}$. Note that the processing order of the jobs from the set $\mathcal{J}_1$ (set $\mathcal{J}_2$) may be arbitrary in an optimal schedule while the permutations π_{12} and π_{21} are *Johnson permutations* of the jobs from set $\mathcal{J}_{12}$ and set $\mathcal{J}_{21}$, respectively. The permutation $\pi_{12} = (J_{i_1}, J_{i_2}, \ldots, J_{i_{n_{12}}})$ (permutation $\pi_{21} = (J_{i_1}, J_{i_2}, \ldots, J_{i_{n_{21}}})$, respectively) is called a *Johnson permutation* of the jobs from the set $\mathcal{J}_{12}$ (from set $\mathcal{J}_{21}$) if the condition

$$\min\{p_{i_k 1}, p_{i_m 2}\} \leq \min\{p_{i_m 1}, p_{i_k 2}\}, \tag{5.1}$$

holds for all inequalities $1 \leq k < m \leq n_{12}$ (the condition

$$\min\{p_{i_k 2}, p_{i_m 1}\} \leq \min\{p_{i_m 2}, p_{i_k 1}\}, \tag{5.2}$$

holds for all inequalities $1 \leq k < m \leq n_{21}$, respectively).

In this section, we consider a more general non-preemptive job-shop scheduling problem with uncertain (interval) processing times. Assume that the processing time p_{ij} of the job $J_i \in \mathcal{J}$ on the machine $M_j \in \mathcal{M}$ is not fixed before scheduling. Actually, in a realization of the schedule, the value p_{ij} may be equal to any real value between a lower bound l_{ij} and an upper bound b_{ij} being given before scheduling. Moreover, the probability distribution of the processing time is unknown before scheduling. This problem is denoted as $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$. We call the problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$ an *uncertain* problem in contrast to the problem $\mathcal{J}2/n_i \leq 2/\mathcal{C}_{max}$ called a *deterministic* problem. If the equality $a_{ij} = b_{ij}$ holds for any job $J_i \in \mathcal{J}$ and any machine $M_j \in \mathcal{M}$, the problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$ turns into a deterministic job-shop problem $\mathcal{J}2/n_i \leq 2/\mathcal{C}_{max}$. Let the set of all feasible vectors $p = (p_{1,1}, p_{1,2}, \ldots, p_{N,1}, p_{N,2})$ of the job processing times be denoted by $T = \{p \mid a_{ij} \leq p_{ij} \leq b_{ij},\ J_j \in \mathcal{J},\ M_m \in \mathcal{M}\}$. For a fixed vector $p \in T$, the uncertain problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$ becomes the deterministic problem $\mathcal{J}2/n_i \leq 2/\mathcal{C}_{max}$ associated with the vector p of the job processing times. Thus, for any vector $p \in T$ of the job processing times, there exists a Jackson pair of permutations that is optimal for the deterministic problem $\mathcal{J}2/n_i \leq 2/\mathcal{C}_{max}$ associated with the vector p of the job processing times.

We denote by S_{12} the set of all permutations (or sequences) of the n_{12} jobs from the set $\mathcal{J}_{12}$, where $|S_{12}| = n_{12}!$, and by S_{21} the set of all permutations (sequences) of the n_{21} jobs from the set $\mathcal{J}_{21}$, where $|S_{21}| = n_{21}!$. Let $S =< S_{12}, S_{21} >$ be a subset of the Cartesian product $(S_{12}, \pi_1, S_{21}) \times (S_{21}, \pi_2, S_{12})$ such that the elements of the set S are ordered pairs of the two permutations (π', π''), where $\pi' = (\pi^i_{12}, \pi_1, \pi^j_{21})$ and $\pi'' = (\pi^j_{21}, \pi_2, \pi^i_{12})$, $1 \leq i \leq n_{12}!$, $1 \leq j \leq n_{21}!$. The processing order of the jobs of the set $\mathcal{J}_1$ (set $\mathcal{J}_2$) may be arbitrary in an optimal schedule, and we can fix both sequences π_1 and π_2 in increasing order of the job numbers. Since both permutations π_1 and π_2 are fixed, and the index i (index j) is the same in each permutation from the pair $(\pi', \pi'') \in P$, we obtain $|P| = n_{12}! \cdot n_{21}!$. We use the following definition of a solution to the uncertain problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$.

Definition 7 *A set of permutations $S(T) \subseteq S$ is called a J-solution to the uncertain problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$, if for each vector $p \in T$, the set $S(T)$ contains at least* *one pair of permutations that is a Jackson one for the deterministic problem $\mathcal{J}2/n_i \leq 2/\mathcal{C}_{max}$ associated with the vector p of the job processing times, provided that for any proper subset S' of the set $S(T)$, there exists at least one vector p' of job processing times such that the set S' does not contain any Jackson pair of permutations for the deterministic problem $\mathcal{J}2/n_i \leq 2/\mathcal{C}_{max}$ associated with the vector p' of the job processing times.*

Let us consider now a flow-shop problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ with the set of jobs $\mathcal{J} = \mathcal{J}_{12}$, $n = n_{12}$ (i.e., all jobs have the same machine route (M_1, M_2)), which is a sub-problem of the problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$. So, for the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$, the set of all feasible permutations is the set P_{12}. Let the set of all feasible vectors $p = (p_{1,1}, p_{1,2}, \dots, p_{n,1}, p_{n,2})$ of the job processing times for the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ be also denoted by T. If the equality $a_{ij} = b_{ij}$ holds, the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ turns into a deterministic flow-shop problem $\mathcal{F}2//\mathcal{C}_{max}$. Thus, for any vector $p \in T$ of the job processing times, there exists a Johnson permutation that is optimal for the deterministic problem $\mathcal{F}2//\mathcal{C}_{max}$ associated with the vector p of the job processing times. For the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$, we use the following definition instead of Definition 7.

Definition 8 *A set of permutations $S_{12}(T) \subseteq S_{12}$ is called a J-solution to the uncertain problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$, if for each vector $p \in T$, the set $S_{12}(T)$ contains at least one permutation that is a Johnson one for the deterministic problem $\mathcal{F}2//\mathcal{C}_{max}$ associated with the vector p of the job processing times provided that for any proper subset S'_{12} of set $S_{12}(T)$, there exists at least one vector p' of job processing times such that the set S' does not contain any Johnson permutation for the deterministic problem $\mathcal{F}2//\mathcal{C}_{max}$ associated with the vector p' of the job processing times.*

Definitions 7 and 8 imply the following three claims.

Lemma 9 *If S_{12} is a J-solution to the flow-shop problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ with the job set $\mathcal{J}_{12}$, then $S =< S_{12}, P_{21} >$ contains at least one Jackson pair of permutations for any vector $p \in T$ of the job processing times for the job-shop problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$ with the job set $\mathcal{J}$.*

Lemma 10 *If S_{21} is a J-solution to the flow-shop problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ with the job set $\mathcal{J}_{21}$, then $S =< P_{12}, S_{21} >$ contains at least one Jackson pair of permutations for any vector $p \in T$ of the job processing times of the job-shop problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$ with the job set $\mathcal{J}$.*

Theorem 11 *If S_{12} is a J-solution to the flow-shop problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ with the job set $\mathcal{J}_{12}$, and S_{21} is a J-solution to the flow-shop problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ with the job set $\mathcal{J}_{12}$, then $S =< S_{12}, S_{21} >$ is a J-solution to the job-shop problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$ with the job set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{12} \cup \mathcal{J}_{21}$.*

5.1 J-solution to the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$

Let us consider the case when there exists a permutation $\pi_i \in S$ constituting a single-element set (singleton) that is a J-solution $S(T) = \{\pi_i\}$ to the uncertain problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$.

Due to Definition 8, such a permutation π_i must be a Johnson permutation for the deterministic problem $\mathcal{F}2//\mathcal{C}_{max}$ associated with any possible vector $p \in T$ of the job processing times. In other words, the permutation π_i has to *dominate* each permutation $\pi_k \in S$, i.e., the inequality $C_{max}(\pi_i, p) \leq C_{max}(\pi_k, p)$ must hold for any vector $p \in T$ and any permutation $\pi_k \in S$. We construct a partition $\mathcal{J} = \mathcal{J}_0 \cup \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}^*$ of the n jobs defined as follows: $\mathcal{J}_0 = \{J_i \in \mathcal{J} \mid b_{i1} \leq u_{i2}, b_{i2} \leq a_{i1}\}$; $\mathcal{J}_1 = \{J_i \in \mathcal{J} \mid b_{i1} \leq a_{i2}, b_{i2} > a_{i1}\} = \{J_i \in \mathcal{J} \setminus \mathcal{J}_0 \mid b_{i1} \leq a_{i2}\}$; $\mathcal{J}_2 = \{J_i \in \mathcal{J} \mid b_{i1} > a_{i2}, b_{i2} \leq a_{i1}\} = \{J_i \in \mathcal{J} \setminus \mathcal{J}_0 \mid b_{i2} \leq a_{i1}\}$; $\mathcal{J}^* = \{J_i \in \mathcal{J} \mid b_{i1} > a_{i2}, b_{i2} > a_{i1}\}$; where some subsets $\mathcal{J}_0$, $\mathcal{J}_1$, $\mathcal{J}_2$, and $\mathcal{J}^*$ may be empty. For each job $J_k \in \mathcal{J}_0$, from the inequalities $b_{k1} \leq a_{k2}$ and $b_{k2} \leq a_{k1}$, we obtain the equalities $a_{k1} = b_{k1} = a_{k2} = b_{k2}$. Since both intervals for the processing times of the job J_k on the machines M_1 and M_2 become a point, the processing times p_{k1} and p_{k2} are fixed in the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$, and they are equal for both machines: $p_{k1} = p_{k2} =: p_k$.

Remark 6 *Note that the sets $\mathcal{J}_1$ and $\mathcal{J}_2$ are defined in a way such that both inclusions $\mathcal{J}_1 \subseteq N_1$ and $\mathcal{J}_2 \subseteq N_2$ may hold for any vector $p \in T$ (the sets N_1 and N_2 are those used in the Johnson algorithm). The jobs from the set $\mathcal{J}_0$ may be either in the set N_1 or in the set N_2 regardless of the vector $p \in T$. The jobs from the set $\mathcal{J}^*$ may be either in the set N_1 or in the set N_2 depending on the vector $p \in T$.*

Theorem 12 *There exists a single-element J-solution $S(T) \subset S$, $|S(T)| = 1$, to the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ if and only if*

(a) for any pair of jobs J_i and J_j from the set $\mathcal{J}_1$ (from the set $\mathcal{J}_2$, respectively) either $b_{i1} \leq a_{j1}$ or $b_{j1} \leq a_{i1}$ (either $b_{i2} \leq a_{j2}$ or $b_{j2} \leq a_{i2}$),

(b) $|\mathcal{J}^*| \leq 1$ and for the job $J_{i^*} \in \mathcal{J}^*$ (if any), the following inequalities hold:

$$a_{i^*1} \geq \max\{b_{i1} \mid J_i \in \mathcal{J}_1\}; \tag{5.3}$$

$$a_{i^*2} \geq \max\{b_{j2} \mid J_j \in \mathcal{J}_2\}; \tag{5.4}$$

$$\max\{a_{i^*1},\ a_{i^*2}\} \geq p_k \ \ \textit{for each job } J_k \in \mathcal{J}_0. \tag{5.5}$$

The opposite extreme case is when a J-solution to the uncertain problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ has a maximal possible cardinality: $|S(T)| = n!$. We use the notations: $a_{max} = \max\{a_{im} \mid J_i \in \mathcal{J}, M_m \in \mathcal{M}\}$ and $b_{min} = \min\{b_{im} \mid J_i \in \mathcal{J}, M_m \in \mathcal{M}\}$.

Theorem 13 *If for the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ the inequality*

$$a_{max} < b_{min} \tag{5.6}$$

holds, then $S(T) = S$.

Corollary 10 *If inequality (5.6) holds, then for each permutation $\pi_k \in S$, there exists a vector $p \in T$ such that π_k is the unique Johnson permutation for the deterministic problem $\mathcal{F}2//\mathcal{C}_{max}$ associated with the vector p of the job processing times.*

Testing condition (a) of Theorem 12 takes $O(n \log n)$ time and testing condition (b) takes $O(n)$ time. Thus, the conditions of Theorem 12 may be tested in $O(n \log n)$ time. Testing the conditions of Theorem 13 and Corollary 10 takes $O(n)$ time.

5.2 General case of the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$

We show how to delete redundant permutations from the set S for constructing a J-solution to the uncertain problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$. To this end, we will fix the order of the jobs $J_v \in \mathcal{J}$ and $J_w \in \mathcal{J}$ in the desired J-solution if there exists at least one Johnson permutation of the form $\pi_k = (s_1, J_v, s_2, J_w, s_3) \in S$ for any vector $p \in T$ of the job processing times. In [5], the following sufficient conditions were proven for such a fixing of the order of two jobs. If

$$b_{v1} \leq a_{v2} \text{ and } b_{w2} \leq a_{w1}, \tag{5.7}$$

then for each vector $p \in T$, there exists a permutation $\pi_k = (s_1, J_v, s_2, J_w, s_3) \in S$, which is a Johnson one for the deterministic problem $\mathcal{F}2//\mathcal{C}_{max}$ associated with the vector p of the job processing times. If

$$b_{v1} \leq a_{v2} \text{ and } b_{v1} \leq a_{w1}, \tag{5.8}$$

then for each vector $p \in T$, there exists a permutation $\pi_k = (s_1, J_v, s_2, J_w, s_3) \in S$, which is a Johnson one for the deterministic problem $\mathcal{F}2//\mathcal{C}_{max}$ associated with the vector p of the job processing times. If

$$b_{w2} \leq a_{w1} \text{ and } b_{w2} \leq a_{v2}, \tag{5.9}$$

then for each vector $p \in T$, there exists a permutation $\pi_k = (s_1, J_v, s_2, J_w, s_3) \in S$, which is a Johnson one for the deterministic problem $\mathcal{F}2//\mathcal{C}_{max}$ associated with the vector p of the job processing times. The subsequences s_1, s_2 and s_3 may be empty, e.g., the jobs J_v and J_w may be adjacent in the above Johnson permutation: $\pi_k = (s_1, J_v, J_w, s_3) \in S$. Note that, if condition (5.7) holds, then the job J_v belongs to the set N_1 and the job J_w belongs to the corresponding set N_2 for all possible realizations of the processing times. Note that, if condition (5.8) holds, then the job J_v belongs to the set N_1 for all possible realizations of the processing times, while the job J_w may be either in the set N_1 or in the set N_2 for different realizations of the job processing times. If condition (5.9) holds, then the job J_w belongs to the set N_2 for all possible realizations of the processing times, while the job J_v may be either in the set N_1 or in the set N_2 for different realizations of the job processing times. If at least one condition (5.7)–(5.9) holds, then there exists a J-solution $S(T)$ to the uncertain problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ with the fixed order $J_v \to J_w$, i.e., the job $J_v \in \mathcal{J}$ has to be located before the job $J_w \in \mathcal{J}$ in any permutation $\pi_i \in S(T)$. It can also be proven that, if both conditions (5.8) and (5.9) do not hold, then there is no J-solution $S(T)$ with the fixed order $J_v \to J_w$ in all permutations $\pi_i \in S(T)$. If in addition, no analogous condition holds for the opposite order $J_w \to J_i$, then at least one permutation with the job J_i located before job J_w and that with the job J_w located before the job J_i must be included in any J-solution $S(T)$ to the uncertain problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$. Summarizing, we present the following claim.

Theorem 14 *There exists a J-solution $S(T)$ to the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ with the fixed order $J_v \to J_w$ of the jobs $J_v \in \mathcal{J}$ and $J_w \in \mathcal{J}$ if and only if at least one condition (5.8) or (5.9) holds.*

5.3 A binary relation defined by the set $S(T)$

Let $\mathcal{J} \times \mathcal{J}$ denote the Cartesian product of the set $\mathcal{J}$. Due to Theorem 14, by testing inequalities (5.8) and (5.9) for each pair of jobs $J_v \in \mathcal{J}$ and $J_w \in \mathcal{J}$, one can construct the following binary relation $\mathcal{A}_{\preceq} \subseteq \mathcal{J} \times \mathcal{J}$ on the set $\mathcal{J}$.

Definition 9 *The inclusion* $(J_v, J_w) \in \mathcal{A}_{\preceq}$ *with* $v \neq w$ *holds if and only if there exists a J-solution* $S(T)$ *to the uncertain problem* $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ *such that the job* $J_v \in \mathcal{J}$ *is located before the job* $J_w \in \mathcal{J}$ *(i.e.,* $J_v \to J_w$*) in all permutations* $\pi_k \in S(T)$*.*

The binary relation $\mathcal{A}_{\preceq}$ defines a dominance digraph $(\mathcal{J}, \mathcal{A}_{\preceq})$ with the vertex set $\mathcal{J}$ and the arc set $\mathcal{A}_{\preceq}$. The relation $(J_v, J_w) \in \mathcal{A}_{\preceq}$ will be also represented as follows: $J_v \preceq J_w$. It takes $O(n^2)$ time to construct the digraph $(\mathcal{J}, \mathcal{A}_{\preceq})$ by testing inequalities (5.8) and (5.9) for each pair of jobs from the set $\mathcal{J}$. In the general case, the binary relation $\mathcal{A}_{\preceq}$ may be not transitive.

Theorem 15 *If the binary relation* $\mathcal{A}_{\preceq}$ *is not transitive, then* $\mathcal{J}_0 \neq \emptyset$*.*

Theorem 15 may be represented in a contrapositive form as follows.

Corollary 11 *If* $\mathcal{J}_0 = \emptyset$*, then the binary relation* $\mathcal{A}_{\preceq}$ *is transitive.*

Next, we consider the case when $\mathcal{J}_0 \neq \emptyset$. In the general case, the binary relation $\mathcal{A}_{\preceq}$ defined over the set $\mathcal{J}$ is a pseudo-order relation since the binary relation $\mathcal{A}_{\preceq}$ possesses only transitivity. The digraph $(\mathcal{J}, \mathcal{A}_{\preceq})$ defined by the binary relation $\mathcal{A}_{\preceq}$ may contain circuits and loops. However, a loop in the digraph $(\mathcal{J}, \mathcal{A}_{\preceq})$ has no sense while defining a J-solution $S(T)$ to the uncertain problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$.

Theorem 16 *(A) If* $J_k \in \mathcal{J}_0$*,* $J_i \in \mathcal{J}_1$*, and* $a_{i1} < p_k$*, then* $J_i \preceq J_k$ *and* $J_k \not\preceq J_i$*.*

(B) If $J_k \in \mathcal{J}_0$*,* $J_i \in \mathcal{J}_2$*, and* $a_{i2} < p_k$*, then* $J_k \preceq J_i$ *and* $J_i \not\preceq J_k$*.*

(C) If $J_k \in \mathcal{J}_0$*,* $J_i \in \mathcal{J}^*$*, inequality* $a_{i1} < p_k$*, holds, and inequality* $J_k \not\preceq J_i$ *does not hold, then* $J_k \preceq J_i$ *and* $J_i \not\preceq J_k$*.*

(D) If $J_k \in \mathcal{J}_0$*,* $J_i \in \mathcal{J}^*$*, inequality* $J_k \not\preceq J_i$ *holds, and inequality* $a_{i1} < p_k$*, does not hold, then* $J_k \preceq J_i$ *and* $J_i \not\preceq J_k$*.*

Remark 7 *If inequality* $p_k > \max\{a_{i1},\ a_{i2}\}$ *holds, then the order of these two jobs cannot be fixed in any J-solution* $S(T)$ *to the uncertain problem* $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$*, i.e., both relations* $J_i \not\preceq J_k$ *and* $J_k \not\preceq J_i$ *hold.*

Theorem 17 *If* $J_k \in \mathcal{J}_0$*,* $J_i \in \mathcal{J}_1$ *(*$J_i \in \mathcal{J}_2$*), and* $p^L_{i1} \geq p_k$ *(*$p^L_{i2} \geq p_k$*, respectively), then both relations* $J_i \preceq J_k$ *and* $J_k \preceq J_i$ *hold.*

Using Theorems 14, 16, and 17 for the case $\mathcal{J}_0 \neq \emptyset$, we can construct a digraph $(\mathcal{J}, \mathcal{A}_{\preceq})$ which may contain circuits. If the conditions of Theorem 17 do not hold for any job $J_i \in \mathcal{J}_1$ ($J_i \in \mathcal{J}_2$), then we can construct a digraph $(\mathcal{J}, \mathcal{A}_{\prec})$ without circuits. *If there exists a job* $J_i \in \mathcal{J}_1$ *(*$J_i \in \mathcal{J}_2$*) such that inequality* $l_{i1} \geq p_k$ *(inequality* $l_{i2} \geq p_k$*) holds, then we can construct a family of solutions to the uncertain problem* $F2|l_{jm} \leq p_{jm} \leq u_{jm}|C_{max}$ *via fixing different feasible positions for the job* $J_k \in \mathcal{J}_0$ *in the permutations from a J-solution. Such a family of J-solutions may be used by a decision-maker for selecting the best permutation to be realized using additional information obtained after processing some jobs from the set* $\mathcal{J}$*.*

Since the cardinality of a J-solution $S(T)$ may vary for different uncertain problems $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ in the range $[1, n!]$, there is no polynomial algorithm for a direct enumeration of all permutations of the set $S(T)$. However, due to Theorems 14, 16, and 17, one can construct a digraph $(\mathcal{J}, \mathcal{A}_{\preceq})$ or a digraph $(\mathcal{J}, \mathcal{A}_{\prec})$ in $O(n^2)$ time. The digraph $(\mathcal{J}, \mathcal{A}_{\prec})$ (digraph $(\mathcal{J}, \mathcal{A}_{\preceq})$) defines a set $S(T)$ (a family of sets $S(T)$) and may be considered as a condense form of a J-solution (family of J-solutions) to the uncertain problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$. Of course, the more job pairs involved in the binary relation $\mathcal{A}_{\preceq}$, the more redundant permutations will be deleted from the set S while constructing a J-solution $S(T) \subseteq S$ to the uncertain problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$. In Subsections 6.4 and 6.5, the case $\mathcal{J}_0 = \emptyset$ is considered.

5.4 Properties of the dominance digraph

Let $\mathcal{J}_0 = \emptyset$, i.e., $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$. For the jobs $J_v \in \mathcal{J}_1$ and $J_w \in \mathcal{J}_1$ (for the jobs $J_v \in \mathcal{J}_2$ and $J_w \in \mathcal{J}_2$), it may happen that there exist both a J-solution $S(T) \subset S$ with the job J_v located before the job J_w in all permutations $\pi_k \in S(T)$ (i.e., $J_v \to J_w$) and a J-solution $S'(T) \subset S$ with the job J_w located before the job J_v in all permutations $\pi_l \in S'(T)$ (i.e., $J_w \to J_v$). In such a case, the digraph $(\mathcal{J}, \mathcal{A}_{\preceq})$ has the circuit (J_v, J_w, J_v).

Theorem 18 *The digraph $(\mathcal{J}, \mathcal{A}_{\preceq})$ has no circuits if and only if the set $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$ includes no pair of jobs $J_i \in \mathcal{J}_k$ and $J_j \in \mathcal{J}_k$ with $k \in \{1, 2\}$ such that*

$$a_{ik} = b_{ik} = a_{jk} = b_{jk}. \tag{5.10}$$

Corollary 12 *Let $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$. The dominance digraph $(\mathcal{J}, \mathcal{A}_{\preceq})$ has the circuit (J_i, J_j, J_i) if and only if for the jobs $J_i \in \mathcal{J}_k$ and $J_j \in \mathcal{J}_k$, $k \in \{1, 2\}$, equalities (5.10) hold.*

In the case $\mathcal{J}_0 = \emptyset$, we can define a binary relation $\mathcal{A}_{\prec} \subseteq \mathcal{A}_{\preceq} \subseteq \mathcal{J} \times \mathcal{J}$.

Definition 10 *If $J_v \preceq J_w$ and $J_w \not\preceq J_v$, then $J_v \prec J_w$. If $J_v \preceq J_w$ and $J_w \preceq J_v$ with $v < w$, then $J_v \prec J_w$ and $J_w \not\prec J_v$.*

Since the set $\mathcal{J}_0$ is empty, we obtain an antireflective (due to Definition 9), antisymmetric (due to Theorem 18 and Definition 10), and transitive (due to Theorem 15) binary relation $\mathcal{A}_{\prec}$ on the set $\mathcal{J}$, i.e., we obtain a strict order. The strict order $\mathcal{A}_{\prec}$ defines a dominance digraph $\mathcal{G} = (\mathcal{J}, \mathcal{A}_{\prec})$ with the vertex set $\mathcal{J}$ and the arc set $\mathcal{A}_{\prec}$.

Corollary 13 *The relation $J_u \prec J_v$ implies $J_u \preceq J_v$ and at least one condition (5.8) or (5.9) must hold. The relation $J_u \preceq J_v$ implies exactly one of the relations $J_u \prec J_v$ or $J_v \prec J_u$.*

Theorem 19 *Let $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$. Then there exists at most one component with a cardinality greater than one in the dominance digraph $\mathcal{G}$.*

Corollary 14 *If $J_i \prec J_j$ and $J_k \prec J_l$, $i \neq k$, $i \neq l$, $j \neq k$, $j \neq l$, then at least one of the relations $J_i \prec J_l$, $J_l \prec J_i$, $J_k \prec J_j$ or $J_j \prec J_k$ must hold.*

Lemma 11 *Let the relation $J_i \prec J_j$ hold and assume that there exists a vector $p \in T$ such that a permutation of the form $\pi_k = (\ldots, J_j, \ldots, J_i, \ldots) \in S$ is a Johnson permutation for the vector p of job processing times. Then condition (5.1) with $i_k = j$, $i_m = i$ must hold as an equality.*

Lemma 12 *Let for a Johnson permutation of the form $\pi_i = (\ldots, J_u, \ldots,\ J_v, \ldots, J_w, \ldots)$ condition (5.1) with $i_k = u$, $i_m = w$ be an equality. Then condition (5.1) must be an equality either with $i_k = u$, $i_m = v$ or with $i_k = v$, $i_m = w$.*

Lemma 13 *Let for a Johnson permutation of the form $\pi_i = (\ldots, J_u, \ldots,\ J_v, \ldots, J_w, \ldots)$ condition (5.1) both with $i_k = u$, $i_m = v$ and with $i_k = v$, $i_m = w$ be a strict inequality. Then condition (5.1) with $i_k = u$, $i_m = w$ must be a strict inequality.*

A permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$ may be considered as a total order of the jobs $\mathcal{J}$. A total order defined by the permutation π_k is called a *linear extension* of a partial order $\mathcal{A}_{\prec}$, if each inclusion $(J_{k_u}, J_{k_v}) \in \mathcal{A}_{\prec}$ implies the inequality $u < v$. Let $\Pi(\mathcal{G})$ denote the set of permutations $\pi_k \in S$ defining all linear extensions of the partial order $\mathcal{A}_{\prec}$. In particular, if $\mathcal{G} = (\mathcal{J}, \emptyset)$, then $\Pi(\mathcal{G}) = S$. On the other hand, if $|\mathcal{A}_{\prec}| = \frac{n(n-1)}{2}$, then $\Pi(\mathcal{G}) = \{\pi_k\}$. A criterion for such a case was given in Theorem 12.

Theorem 20 *Let $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$. For any vector $p \in T$, the set $\Pi(\mathcal{G})$ contains a Johnson permutation for the problem $F2||C_{max}$ associated with the vector p of job processing times.*

Corollary 15 *If $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$, then there exists a J-solution $S(T)$ to the problem $\mathcal{F}2/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$ such that $S(T) \subseteq \Pi(\mathcal{G})$.*

5.5 How to construct a minimal dominant set $S(T) \subseteq \Pi(\mathcal{G})$?

The pair of jobs $J_i \in \mathcal{J}$ and $J_j \in \mathcal{J}$ is called a *conflict pair* of jobs, if neither the relation $J_i \preceq J_j$ nor $J_j \preceq J_i$ holds. Due to Corollary 13, both conditions $(J_i, J_j) \notin \mathcal{A}_{\prec}$ and $(J_j, J_i) \notin \mathcal{A}_{\prec}$ hold for a conflict pair of jobs J_i and J_j.

Lemma 14 *Let $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$. A permutation of the form $\pi_g = (\ldots, J_i, \ldots, J_r,\ \ldots, J_j, \ldots)$ $\in \Pi(\mathcal{G})$ is redundant, if for a pair of jobs $J_i \in \mathcal{J}_k$ and $J_j \in \mathcal{J}_k$ with $k \in \{1, 2\}$, equalities (5.10) hold and the job J_r creates a conflict pair both with the job J_i and with the job J_j.*

A redundant permutation defined in Lemma 14 is called a *redundant permutation of type 1*. Due to Lemma 14, testing whether the set $\Pi(\mathcal{G})$ contains a redundant permutation of type 1 takes $O(n^2)$ time. In order to delete all redundant permutations of type 1 from set $\Pi(\mathcal{G})$, it is sufficient to treat each pair of jobs J_i and J_j, for which the condition of Lemma 14 holds, as one (any) job from the set $\{J_i, J_j\}$. E.g., if $i < j$, then only job J_i will be treated. We delete (successively, one by one) a vertex J_j from the digraph $\mathcal{G}$ along with all arcs incident to vertex J_j provided that $i < j$. Let $\mathcal{G}^I$ denote the digraph obtained from $\mathcal{G} = (\mathcal{J}, \mathcal{A}_{\prec})$ after such a deletion of all corresponding vertices and arcs. We denote the set of vertices deleted as $\mathcal{J}^I$. The set $\Pi(\mathcal{G}^I)$ has no redundant permutations of type 1 for the instance $\mathcal{F}2/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$ obtained from the original one via deleting the set of jobs $\mathcal{J}^I$. The desired subset of the set $\Pi(\mathcal{G})$ without a redundant permutation of type 1 will be obtained from the set $\Pi(\mathcal{G}^I)$ via inserting each job $J_j \in \mathcal{J}^I$ just after the corresponding job J_i in each permutation from the set $\Pi(\mathcal{G}^I)$.

Let the inclusion $J_j \in \mathcal{J}^*$ hold. We define two sets of jobs as follows:

$$\mathcal{J}'_j = \{J_q \in \mathcal{J}_2 \mid \min\{b_{j1}, b_{j2}\} < b_{q2}\} \bigcup \{J_r \in \mathcal{J}_1 \cup \mathcal{J}^* \mid \min\{b_{j1}, b_{j2}\} \le a_{r1}\};$$

$$\mathcal{J}''_j = \{J_w \in \mathcal{J}_1 \mid \min\{b_{j1}, b_{j2}\} < b_{w1}\} \bigcup \{J_u \in \mathcal{J}_2 \cup \mathcal{J}^* \mid \min\{b_{j1}, b_{j2}\} \le l_{u2}\}.$$

Lemma 15 *Let $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$. If $J_j \in \mathcal{J}^*$, $J_q \in \mathcal{J}'_j$, $J_w \in \mathcal{J}''_j$, then each permutation of the form $\pi_g = (\dots, J_q, \dots, J_j, \dots, J_w, \dots) \in \Pi(\mathcal{G})$ is redundant.*

A redundant permutation defined by the condition of Lemma 15 will be called a *redundant permutation of type 2*. Due to Lemma 15, testing whether a permutation $\pi_g \in \Pi(\mathcal{G})$ is a redundant permutation of type 2 takes $O(n)$ time. It is easy to prove the following claim: *If for the sets $\mathcal{J}'_j$ and $\mathcal{J}''_j$ constructed for job $J_j \in \mathcal{J}^*$ one of the following conditions (a) or (b) holds: (a) either $\mathcal{J}'_j = \emptyset$ or $\mathcal{J}''_j = \emptyset$; (b) $\mathcal{J}'_j = \mathcal{J}''_j$ and $|\mathcal{J}'_j| = |\mathcal{J}''_j| = 1$, then the job J_j does not generate a redundant permutation of type 2 in the set* $\Pi(\mathcal{G})$. Let $\Pi^*(\mathcal{G})$ denote the set of permutations remaining in the set $\Pi(\mathcal{G})$ after deleting all redundant permutations of type 1 and type 2.

Lemma 16 *Let $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$. If the permutation $\pi_t \in \Pi(\mathcal{G})$ is redundant in the set $\Pi(\mathcal{G})$, then π_t is a redundant permutation either of type 1 or type 2.*

Theorem 21 *If the set $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$ does not contain a pair of jobs $J_i \in \mathcal{J}^*$ and $J_j \in \mathcal{J}^*$such that the condition*

$$\max\{p^L_{i,3-k}, p^L_{j,3-k}\} < p^L_{ik} = p^U_{ik} = p^L_{jk} = p^U_{jk} < \min\{p^U_{i,3-k}, p^U_{j,3-k}, \} \tag{5.11}$$

holds, $k \in \{1, 2\}$, then $\Pi^*(\mathcal{G}) = S(T)$.

Testing the condition of Theorem 21 takes $O(n)$ time. Due to Theorem 21 and Lemma 16, if there are no jobs such that condition (5.11) holds, a J-solution can be constructed by deleting all redundant permutations of type 1 and type 2 from the set $\Pi(\mathcal{G})$. Since the obtained set $\Pi^*(\mathcal{G})$ is uniquely defined, the following claim is correct.

Corollary 16 *If the set $\mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$ does not contain a pair of jobs J_i and J_j such that condition (5.11) holds, then the relation $\mathcal{A}_\prec$ defines a unique J-solution $\Pi^*(\mathcal{G}) = S(T)$ to problem $\mathcal{F}2/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$.*

The condition of Theorem 21 is sufficient (but not necessary) for the uniqueness of a J-solution $\Pi^*(\mathcal{G}) = S(T)$ to problem $\mathcal{F}2/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$.

5.6 Conditions for schedule domination

Now we provide sufficient conditions for the existence of a *dominant* set of pairs of permutations in the following sense.

Definition 11 *The pair of permutations $(\pi'_u, \pi''_u) \in \mathcal{S}$ dominates the pair of permutations $(\pi'_v, \pi''_v) \in \mathcal{S}$ with respect to T if the inequality $C_{max}(\pi_u, p) \le C_{max}(\pi_v, p)$ holds for any vector $p \in T$ of the job processing times. The set of permutations $\mathcal{S}' \subseteq \mathcal{S}$ is dominant with respect to T if for each pair of permutations $(\pi'_v, \pi''_v) \in \mathcal{S}$, there exists a permutation $(\pi'_u, \pi''_u) \in \mathcal{S}'$ that dominates the pair of permutations (π'_v, π''_v) with respect to T.*

It is clear that the set of pairs of permutations $\mathcal{S}(T)$ used in Definition 7 is dominant with respect to T. It should be noted that Definition 11 does not use Jackson pairs of permutations in contrast to Definition 7. In what follows, we will relax (if it will be useful) the requirement for a dominant pair of permutations (π'_u, π''_u) to be a Jackson one.

Lemma 17 *Let the problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$ with the job set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{12} \cup \mathcal{J}_{21}$ be considered. If*

$$\sum_{J_j \in \mathcal{J}_{12}} b_{j1} \leq \sum_{J_j \in \mathcal{J}_{21} \cup \mathcal{J}_2} a_{j2},$$

then in an optimal schedule, the order of the jobs of the set $\mathcal{J}_{12}$ may be arbitrary.

Let $\mathcal{P} = \{J_i \in \mathcal{J}_{12} | (J_i \prec J_j) \text{ or } (J_j \prec J_i) \text{ for all } J_j \in \mathcal{J}_{12} \setminus \{J_i\}\}$. Consider two jobs $J_q \in \mathcal{P}$ and $J_w \in \mathcal{P}$, $J_q \prec J_w$, such that the set $\mathcal{P}$ does not contain any job J_s with $J_q \prec J_s \prec J_w$. If there exists a set $\mathcal{J}_k \subset \mathcal{J}_{12} \setminus \mathcal{P}$ such that for every job $J_r \in \mathcal{J}_k$ the relations $J_q \prec J_r \prec J_w$ hold, then the set $\mathcal{J}_k$ is called a *conflict set* of jobs. Obviously, the set $\mathcal{J}_{12}$ may contain more than one conflict set. If $\mathcal{P}_1 \subset \mathcal{J}_{12}$ and $\mathcal{P}_2 \subset \mathcal{J}_{12}$ are two conflict sets, then $\mathcal{P}_1 \cup \mathcal{P}_2 = \emptyset$. If the partial strict order $\mathcal{A}$ over the set $\mathcal{J}_{12}$ is

$$(J_1 \prec \ldots \prec J_k \prec \{J_{k+1}, J_{k+2}, \ldots, J_{k+r}\} \prec J_{k+r+1} \prec \ldots \prec J_{n_{12}}),$$

the notation $\{J_{k+1}, J_{k+2}, \ldots, J_{k+r}\}$ denotes the conflict set. Analogously, we can define a conflict set for the jobs from the set $\mathcal{J}_{21}$.

Lemma 18 *Let the problem $\mathcal{J}2/a_{ij} \leq p_{ij} \leq b_{ij}, n_i \leq 2/\mathcal{C}_{max}$ with the job set $\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2 \cup \mathcal{J}_{12} \cup \mathcal{J}_{21}$ be considered. Let the partial strict order $\mathcal{A}_\prec$ over the set $\mathcal{J}_{12}$ be*

$$(J_1 \prec \ldots \prec J_k \prec \{J_{k+1}, J_{k+2}, \ldots, J_{k+r}\} \prec J_{k+r+1} \prec \ldots \prec J_{n_{12}}).$$

If

$$\sum_{J_i \in \mathcal{J}_{12}, i < k+1} b_{i1} \leq \sum_{J_j \in \mathcal{J}_{21} \cup \mathcal{J}_2} a_{j2} + \sum_{J_i \in \mathcal{J}_{12}, i < k+r+1} a_{i2},$$

then in an optimal schedule, the order of the jobs of the conflict set $\mathcal{J}_k \subseteq \{J_{k+1}, J_{k+2}, \ldots, J_{k+r}\}$ may be arbitrary.

If the conditions of Lemma 18 (Lemma 17) hold, we can order the jobs of the conflict set $\mathcal{J}_k$ (the whole set $\mathcal{J}_{12}$), in increasing order of the job numbers, for example, which will decrease the cardinality of the set of the pairs of permutations under consideration. On the other hand, if the conditions of Lemma 17 do not hold, machine M_2 may have an idle time when it processes the jobs from the conflict set $\mathcal{J}_k$. Next, we describe and justify sufficient conditions and the formal algorithms for constructing a dominant permutation (if it is possible) for problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ in the following sense.

Definition 12 *The permutation $\pi_u \in S$ dominates the permutation $\pi_k \in S$ with respect to T if $C_{max}(\pi_u, p) \leq C_{max}(\pi_k, p)$ for any vector $p \in T$ of the job processing times. The set of permutations $S' \subseteq S$ is dominant with respect to T if for each permutation $\pi_k \in S$, there exists a permutation $\pi_u \in S'$ that dominates permutation π_k with respect to T.*

If conditions (a) – (b) of Theorem 12 hold, then the singleton $\{\pi_u\}$ is dominant with respect to T (we say that such a permutation π_u is dominant with respect to T). It is also clear that the set of permutations S used in Definition 8 is dominant with respect to T. It should be noted that Definition 12 does not exploit the Johnson rule in contrast to Definition 8. In what follows, we will relax (if it will be useful) the demand for a dominant permutation π_u to be a Johnson one. Note that, instead of condition (5.1) for the optimality of permutation $\pi_i \in S$, one can also consider more general conditions proven in [13, 71, 74]. The larger subset of optimal permutations constructed in [13, 70, 71, 74] may provide more choices to the decision-maker to find an optimal schedule under uncertainty conditions.

Lemma 19 *Let the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ with the job set $\mathcal{J}_{12}$ be considered. Let the partial strict order $\mathcal{A}_{\prec}$ over the set $\mathcal{J}_{12}$ be as follows:*

$$(J_1 \prec \ldots \prec J_k \prec \{J_{k+1}, J_{k+2}, \ldots, J_{k+r}\} \prec J_{k+r+1} \prec \ldots \prec J_{n_{12}}).$$

If

$$\sum_{J_i \in \mathcal{J}_{12}, i<k+1} b_{i1} \leq \sum_{J_i \in \mathcal{J}_{12}, i<k+r+1} a_{i2},$$

then in an optimal schedule, the order of the jobs of the conflict set $\mathcal{J}_k \subseteq \{J_{k+1}, J_{k+2}, \ldots, J_{k+r}\}$ may be arbitrary.

5.7 On-line scheduling

Let $\mathcal{J}_0 = \emptyset$. As far as the on-line scheduling phase is concerned, the actual value p^*_{jm} of the job processing time p_{jm} is available at the time-point $t_j = c_m(j)$ when job J_j is completed by machine M_m. Let the partial strict order $\mathcal{A}_{\prec}$ over the set $\mathcal{J}_{12}$ be as follows:

$$(J_1 \prec \ldots \prec J_k \prec \{J_{k+1}, J_{k+2}, \ldots, J_{k+r}\} \prec J_{k+r+1} \prec \ldots \prec J_{n_{12}}).$$

Then at time-point $t_0 = 0$, machine M_1 can start processing jobs from the set $\{J_1, \ldots, J_k\}$ till the conflict set of jobs $\mathcal{J}_k$. An additional decision has to be used at the time-point $t_k = c_1(k)$. It is clear that at time-point t_k, the actual processing times of the jobs from the set $\mathcal{J}(t_k, 1) = \{J_1, J_2, \ldots, J_k\}$ on machine M_1 are already known. Let these actual values of the processing times be as follows: $p_{1,1} = p^*_{1,1},\ p_{2,1} = p^*_{2,1}, \ldots, p_{k1} = p^*_{k1}$.

At time-point t_0, machine M_2 starts to process jobs from the set $\mathcal{J}_{12}$. Let at time-point t_k, machine M_2 already operates jobs from the set $\mathcal{J}(t_k, 2) \subset \mathcal{J}_{21} \cup \mathcal{J}_2 \cup \{J_1, J_2, \ldots, J_k\}$. The actual values p^*_{j2} of the processing times p_{j2} of the jobs J_j from the set $\mathcal{J}(t_k, 2)$ are available at time-point $t_k = c_1(k)$, i.e., $p_{j2} = p^*_{j2}$, while the actual values of the processing times p_{l2} of the jobs J_l from the set $\mathcal{J}_{21} \cup \mathcal{J}_2 \cup \{J_1, J_2, \ldots, J_k\} \setminus \mathcal{J}(t_k, 2)$ are unavailable at time-point $t_k = c_1(k)$. Thus, at time-point $t_k = c_1(k)$, the set of feasible vectors

$$T(k) = \{p \in T \mid p_{i1} = p^*_{i1},\ p_{j2} = p^*_{j2},\ J_i \in \mathcal{J}(t_k, 1),\ J_j \in \mathcal{J}(t_k, 2)\} \tag{5.12}$$

will be utilized instead of set T defined by equality (6.1). We can calculate a lower bound $c^L_2(k)$ for the actual value $c_2(k)$ in the following way:

$$c^L_2(k) = c_2(l-1) + \max\{a_{l2},\ c_1(k) - c_2(l-1)\} + \sum_{j=l}^{k} a_{j2}.$$

Next, we consider the following question. When will one of the permutations of the set S be optimal for all vectors $p \in T(k)$ of the job processing times? To answer this question, we have to consider all possible orders of the jobs of the conflict set. At time-point t_{k-1}, a scheduler can choose the order of the jobs in the conflict set being processed next (immediately after job J_{k-1}) on machine M_1. Let a set of r jobs be conflicting at time-point $t_k = c_1(k) \geq 0$. W.l.o.g. we assume that the jobs from the set $\{J_{k_1}, J_{k_2}, \ldots, J_{k_r}\} \subset \mathcal{J} = \mathcal{J}^* \cup \mathcal{J}_1 \cup \mathcal{J}_2$ are conflicting. Then we need to test the $r!$ possible orders of the conflicting jobs. We can calculate the following upper bound $c_2^U(k)$ for the actual value $c_2(k)$:

$$c_2^U(k) = c_2(l-1) + \sum_{j=l}^{k-1} b_{j2}.$$

We consider sufficient conditions for an optimal ordering of the conflict jobs at the on-line decision-making time-points. Let the set $\mathcal{J}_k$ of r jobs be conflicting at time-point $t_k = c_1(k) > 0$. Then we need test no more than $r!$ possible orders of conflicting jobs (according to the partial order of the jobs of the conflict set $\mathcal{J}_k$).

Lemma 20 *Let the partial strict order $\mathcal{A}_{\prec}$ over the set $\mathcal{J}_{12}$ be as follows:*

$$(J_1 \prec \ldots \prec J_k \prec \{J_{k_1}, J_{k_2}, \ldots, J_{k_r}\} \prec J_{k+1} \prec \ldots \prec J_n).$$

If inequality

$$\sum_{i=1}^{s+1} b_{k_i 1} \leq \sum_{j=0}^{s} a_{k_j 2}$$

holds for each $s = 0, 1, \ldots, r$, where $a_{k_0 2} = c_2^L(k) - c_1(k)$, then the permutation $\{J_1, \ldots, J_k, J_{k_1}, J_{k_2}, \ldots, J_{k_r}, J_{k+1}, \ldots, J_n\}$ is dominant with respect to $T(k)$.

Let δ_m be defined recursively as follows:

$$\delta_m = \max\{0,\ p^U_{m,2} - p^L_{m+1,1} + \delta_{m-1}\},$$

where $\delta_1 = \max\{0,\ p^U_{1,2} - p^L_{2,1}\}$. Using this notation, we can generalize the above lemmas for the case of r conflicting jobs at the time-point $t_0 = 0$.

Lemma 21 *Let the partial strict order $\mathcal{A}_{\prec}$ over the set $\mathcal{J}_{12}$ be as follows:*

$$(J_1 \prec \ldots \prec J_k \prec \{J_{k_1}, J_{k_2}, \ldots, J_{k_r}\} \prec J_{k+1} \prec \ldots \prec J_n).$$

If the conditions

$$\sum_{i=m}^{s} a_{k_i 1} > \sum_{j=m-1}^{s-1} b_{k_j 2}, \qquad m = 1, 2, \ldots, s,$$

$$\sum_{i=s+1}^{m} b_{k_i 1} \leq \sum_{j=s}^{m-1} a_{k_j 2}, \qquad m = s+1, s+2, \ldots, r,$$

$$\sum_{i=s+1}^{r+1} a_{k_i 1} \geq \sum_{j=s}^{r} b_{k_j 2}$$

hold for at least one $s = 1, 2, \ldots, r$, where

$$b_{k_0 2} = c_2^U(k) - c_1(k),$$

then the permutation $\{J_1, \ldots, J_k, J_{k_1}, J_{k_2}, \ldots, J_{k_r}, J_{k+1}, \ldots, J_n\}$ is dominant with respect to $T(k)$.

Lemma 22 *Let the partial strict order* $\prec$ *be as follows:*

$$(\{J_1, J_2, \ldots, J_r\} \prec J_{r+1} \prec \ldots).$$

If

$$p^L_{r+1,1} \geq p^U_{r,2} + \delta_{r-1},$$

then the order of the jobs $J_1,\ J_2,\ \ldots,\ J_r$ *in the optimal permutation is as follows:* $J_1 \rightarrow J_2 \rightarrow \ldots \rightarrow J_r$.

5.8 Off-line and on-line scheduling algorithms

The following algorithms were coded in C++: Algorithm 1 for the off-line scheduling and Algorithm 2 for the on-line scheduling provided that the set $\mathcal{J}_0$ is empty.

Algorithm 1 for off-line scheduling

Input: Lower and upper bounds l_{jm} and u_{jm} for the processing times p_{jm} of the jobs $J_j \in \mathcal{J}$ on the machines $M_m \in \mathcal{M}$.
Output: Solution $S(T)$ to the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$; binary relation $\prec$ defining the solution $S(T)$, if $|S(T)| > 1$.
Step 1: Test conditions $(a) - (b)$ of Theorem 12. **IF** conditions $(a) - (b)$ hold
Step 2: **THEN** construct a dominant permutation of the jobs $\mathcal{J}$ **GOTO** step 10;
Step 3: **ELSE** using Theorem 14 construct the digraph $G = (\mathcal{J}, A)$; set $k := 0$;
Step 4: Take the first conflict subset of jobs in the set $\mathcal{J}$;
Step 5: **IF** the conditions of Lemma 19 hold for the conflict set **THEN** order the jobs of the conflict set arbitrarily, **ELSE** set $k := k + 1$;
Step 6: **IF** there is no conflict set **THEN IF** $k = 0$ **GOTO** step 10.
Step 7: **ELSE GOTO** step 9.
Step 8: **ELSE** Take the next subset of conflict jobs in the set $\mathcal{J}$ **GOTO** step 5;
Step 9: **STOP:** Binary relation defining solution $S(T)$, $|S(T)| > 1$.
Step 10: **STOP:** Dominant permutation with respect to T, $|S(T)| = 1$.

In Algorithm 2, the integer $k,\ 1 \leq k \leq n$, denotes the number of decision-making time-points $t_i = c_1(i),\ J_i \in \mathcal{J}$, in the on-line scheduling phase. The integer $m,\ 1 \leq m \leq k$, denotes the number of decision-making time-points for which the optimal orders of the conflicting jobs were found using the above sufficient conditions.

Algorithm 2 for on-line scheduling ($\mathcal{J}_0 = \emptyset$)

Input: Lower and upper bounds l_{jm} and u_{jm} for the processing times p_{jm} of the jobs $J_j \in \mathcal{J}$ on the machines $M_m \in \mathcal{M}$; solution $S^*(T)$, $|S^*(T)| > 1$, to the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ defined by the partial strict order $\prec$.
Output: Either a dominant permutation $\pi_u \in S^*(T)$ with respect to $T(i),\ J_j \in \mathcal{J}$, or a permutation from the solution $S^*(T)$ without optimality proof.
Step 1: Set $k := 0$, $m := 0$.
Step 2: **IF** the first jobs in the partial strict order $\prec$ are conflicting **THEN BEGIN** check the condition of Lemma 22 for all conflict permutations

IF the condition of Lemma 22 holds for at least one conflict permutation **THEN** Set $k := k + 1$, $m := m + 1$, choose an optimal permutation of the conflicting jobs **ELSE** Set $k := k + 1$, choose an arbitrary permutation of the conflict jobs and process the jobs of the conflict set in the chosen permutation. **END**

Step 3: **UNTIL** completing the last job in the actual schedule process of the linear part of the partial strict order $\prec$.

Step 4: Check the conditions of Lemmas 20 and 21 for all orders of the conflict jobs.

Step 5: **IF** at least one sufficient condition from Lemmas 20 and 21 holds for at least one order of the jobs of the conflict set **THEN** $k := k + 1$, $m := m + 1$, choose an optimal order of the conflict jobs **ELSE** $k := k + 1$, choose an arbitrary order of the jobs of the conflict set,

Step 6: process the jobs of the conflict set in the chosen order. **RETURN**

Step 7: **IF** $k = m$ **THEN GOTO** step 10.

Step 8: Calculate the length $C_{\max}$ of the schedule that was constructed via steps $1 - 7$ and the length $C^*_{\max}$ of the optimal schedule constructed for the actual processing times. **IF** $C_{\max} = C^*_{\max}$ **THEN GOTO** step 9.

Step 9: **STOP:** the constructed schedule is not optimal for the actual job processing times.

Step 10: **STOP:** the optimality of the permutation is defined after the schedule execution.

Step 11: **STOP:** the optimality of the permutation is proven before the schedule execution.

The computational experiments have shown the efficiency of the stability approach on many randomly generated instances of the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$. The average relative error of the makespan

$$[(C_{max} - C^*_{max})/C^*_{max}] \cdot 100\%$$

obtained for all actual schedules was less than 2.9% for all randomly generated instances with $n = 10$ jobs. The average relative error of the makespan obtained for all actual schedules was less than 1.67% for all randomly generated instances with n jobs, where $20 \leq n \leq 1000$. The sufficient conditions for the existence of a dominant permutation given in Lemmas $20 - 21$ was very effective for on-line scheduling. In should also be noted that the number of decision-making time-points $t_i = c_1(i)$ when the order of the conflicting jobs has to be decided is rather high for some instances with $n \geq 50$. However, these decisions made in Algorithm 2 were fast: There was no randomly generated instance which takes a running time of more than 0.05 seconds for a processor with 1200 MHz.

5.9 Notes and references

In this section, the complete version of the stability approach was implemented for the two-stage scheduling problem with the makespan criterion. At the off-line stage of the stability method, a minimal dominant set (MDS) of the job permutations was constricted, which provided either the final optimal solution for the problem $\mathcal{F}2/a_{ij} \leq p_{ij} \leq b_{ij}/\mathcal{C}_{max}$ or a minimal set of optimal permutations that contained at least one solution for each possible scenario. In the latter case, an MDS was used at the on-line stage of the stability method in order to construct a factually optimal schedule for the uncertain problem $\mathcal{F}2/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$. All the results presented in this section were proven in the papers [4, 5, 27, 66, 67, 68, 69, 72, 73, 77, 105].

A minimal dominant set (MDS) of semiactive schedules was investigated in [4, 5, 54, 55] for the C_{max} criterion, and in [4, 57, 105] for the total flow time criterion $\sum C_i$. In particular, the paper [105] addressed the total flow time criterion in a two-machine flow-shop problem $\mathcal{F}2/a_{ij} \le p_{ij} \le b_{ij}/\sum \mathcal{C}_i$. A geometrical algorithm has been developed for solving the flow-shop problem $\mathcal{F}m/a_{ij} \le p_{ij} \le b_{ij}, n = 2/\sum \mathcal{C}_i$ with m machines and two jobs. For an uncertain flow-shop problem with two or three machines, sufficient conditions have been identified when the transposition of two jobs minimizes the total flow time. The work of [4] dealt with the case of separate setup times with the criterion of minimizing the makespan or total flow time. Namely, the processing times were fixed while each setup time was relaxed to be a distribution-free random variable within a given lower and upper bound. Dominance relations were identified for an uncertain flow-shop problem with two machines. In [5], for a two-machine flow-shop problem $\mathcal{F}2/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$ sufficient conditions were identified when the transposition of two jobs minimizes the makespan. In [54, 55], necessary and sufficient conditions were proven for the case when a single semiactive schedule dominates all the others, and necessary and sufficient conditions were proven for the case when it is possible to fix the optimal order of two jobs for the makespan criterion with interval processing times.

The stability approach was originally proposed in [55] and developed in [54] for the C_{max} criterion and in [55] for the total completion time criterion $\sum C_i$. In particular, a formula for calculating the stability radius of an optimal schedule (i.e., the largest value of simultaneous independent variations of the job processing times for the schedule to remain optimal) has been provided in [55]. In the work of [55], the stability analysis of a schedule minimizing the total completion time was exploited in a branch and bound method for solving the job-shop problem $Jm|p^L_{ij} \le p_{ij} \le p^U_{ij}|\sum C_i$ with m machines. In [5], for the two-machine flow-shop problem $\mathcal{F}2/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$, sufficient conditions were identified when the transposition of two jobs minimizes the makespan. The article [105] addressed total completion time in a flow-shop with interval processing times. In particular, a geometrical algorithm was developed for solving the flow-shop problem $\mathcal{F}m/a_{ij} \le p_{ij} \le b_{ij}, n = 2/\sum \mathcal{C}_i$ with m machines and two jobs. For the flow-shop problem with two and three machines, sufficient conditions were identified when the transposition of two jobs minimizes total completion time. The work of [4] was devoted to the case of separate setup times with the criterion of minimizing the makespan or total completion time. Namely, the processing times are fixed while each setup time is relaxed to be a distribution-free random variable within a given lower and upper bound. Local and global dominance relations were identified for such a flow-shop problem with two machines. In [54], necessary and sufficient conditions were proven for the case when a single schedule dominates all the others, and necessary and sufficient conditions were proven for the case when it is possible to fix the optimal order of two jobs for the makespan criterion with interval processing times. In contrast to the papers [54, 55], where exponential algorithms based on an exhausting enumeration of the feasible schedules were derived for a job shop problem $\mathcal{J}m/a_{ij} \le p_{ij} \le b_{ij}/\mathcal{C}_{max}$, in this section, we presented polynomial procedures for solving the special case of the problem when $m = 2$ and all jobs have the same technological route.

The stability approach was also implemented in [3] for the two-machine flow shop problem with the total completion time objective, interval processing and setup times. In [26], a general model motivated by a concrete industrial application was developed. The authors proposed a two-phase method based on solving a mixed-integer programming problem and improving the initial schedule by a tabu search heuristic. Such an approach may be used to handle various scheduling problems when there are sequence-dependent set-up times, the jobs have to be processed in batches and the machines have non-availability intervals. In [11, 12], the stability of a Johnson permutation (a Jackson pair of permutations) was used for solving the two-machine flow shop (job shop) scheduling problem with non-availability intervals given for the machines.

A mass uncertain problem $\alpha/a_{ij} \le p_{ij} \le b_{ij}/\gamma$ cannot be easier (in an asymptotical sense) than its deterministic counterpart $\alpha//\gamma$ since the latter is a special case of the former (if $a_{ij} = b_{ij}$ for each $J_i \in J$ and $M_j \in M$). So, the statement by Lenstra [60] about "the mystical power of twoness" that a shop scheduling problem may be solved in polynomial time only if at least one number of the machines or jobs is restricted by two (and $\mathcal{P} \ne \mathcal{NP}$) is applicable to the uncertain problems $\alpha/a_{ij} \le p_{ij} \le b_{ij}/\gamma$ as well. As it was proven in [95, 100], the deterministic problems $\mathcal{F}/n = 3/\mathcal{C}_{max}$ and $\mathcal{F}/n = 3/\sum \mathcal{C}_i$ and all other deterministic flow shop scheduling problems $\mathcal{F}/n = 3/\Phi$ with a non-trivial regular criterion Φ are binary NP-hard. Even the deterministic problems $\mathcal{J}3/n = 3/\mathcal{C}_{max}$ and $\mathcal{J}3/n = 3/\sum \mathcal{C}_i$ and so the other job shop scheduling problems $\mathcal{F}/n = 3/\Phi$ are binary NP-hard [116]. The complexity of deterministic shop scheduling problems with a fixed number of jobs or (and) machines was surveyed in [17, 60, 88].

6 Single-stage scheduling

The single machine scheduling problem with interval processing times, which we consider in this section, can be described as follows. There are $n \ge 2$ jobs $\mathcal{J} = \{J_1, J_2, ..., J_n\}$ to be processed on a single machine. Associated with a job $J_i \in \mathcal{J}$, there is a weight $w_i > 0$ reflecting the job importance. All jobs are available for processing at time zero. The processing time p_i of a job $J_i \in \mathcal{J}$ can take any real value between a lower bound $a_i > 0$ and an upper bound $b_i \ge a_i$. The exact value of the processing time remains unknown until job completion. Let T denote the set of vectors $p = (p_1, p_2, \dots, p_n)$ of the processing times in the space R^n_+ of non-negative n-dimensional real vectors. The set T is the Cartesian product of the closed intervals (or segments) $[a_i, b_i]$:

$$T = \{p \mid p \in R^n_+,\ a_i \le p_i \le b_i,\ i \in \{1, 2, \dots, n\}\} = \times_{i=1}^n [a_i, b_i]. \tag{6.1}$$

A vector $p \in T$ is called a scenario. Let $S = \{\pi_1, \pi_2, \dots, \pi_{n!}\}$ denote the set of all permutations $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n})$ of the n jobs in $\mathcal{J}$. Given a permutation $\pi_k \in S$ and a scenario $p \in T$, let $C_i = C_i(\pi_k, p)$ denote the completion time of the job $J_i \in \mathcal{J}$ in the semi-active schedule [79, 122] defined by the permutation π_k. The criterion $\sum w_i C_i$ denotes the minimization of the sum of the weighted completion times:

$$\sum_{J_i \in \mathcal{J}} w_i C_i(\pi_t, p) = \min_{\pi_k \in S} \{ \sum_{J_i \in \mathcal{J}} w_i C_i(\pi_k, p)\},$$

where the permutation $\pi_t = (J_{t_1}, J_{t_2}, \dots, J_{t_n}) \in S$ is optimal. This problem is denoted as $1/a_i \le p_i \le b_i/\sum w_i \mathcal{C}_i$. Next, we adopt the *stability approach* [54, 55, 122] to the problem $1/a_i \le p_i \le b_i/\sum w_i \mathcal{C}_i$.

6.1 Fixed processing times

In [90], it was proven that the deterministic problem $1//\sum w_i\mathcal{C}_i$ can be solved in $O(n \log_2 n)$ time due to the following sufficient condition for the optimality of a permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$:

$$\frac{w_{k_1}}{p_{k_1}} \geq \frac{w_{k_2}}{p_{k_2}} \geq \ldots \geq \frac{w_{k_n}}{p_{k_n}}. \tag{6.2}$$

Hereafter, the inequality $p_{k_i} > 0$ must hold for each job $J_{k_i} \in \mathcal{J}$. Note that inequalities (6.2) are also a necessary condition for the optimality of a permutation $\pi_k \in S$.

Theorem 22 *The permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$ is optimal for the problem $1//\sum w_i\mathcal{C}_i$ if and only if inequalities* (6.2) *hold.*

To obtain the corresponding result for the uncertain problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$, we need the following two corollaries from Theorem 22.

Corollary 17 *If the inequality*

$$\frac{w_u}{p_u} > \frac{w_v}{p_v}$$

holds, then in all optimal permutations for the problem $1//\sum w_i\mathcal{C}_i$ the job J_u precedes the job J_v.

Corollary 18 *If the equality*

$$\frac{w_u}{p_u} = \frac{w_v}{p_v}$$

holds, then the problem $1//\sum w_i\mathcal{C}_i$ has both an optimal permutation $\pi_l \in S$ of the form $\pi_l = (\ldots, J_u, J_v, \ldots)$ and an optimal permutation $\pi_m \in S$ of the form $\pi_m = (\ldots, J_v, J_u, \ldots)$.

6.2 Minimal dominant set (MDS)

Let the notation $1/p/\sum w_i\mathcal{C}_i$ be used for indicating an instance with a fixed scenario $p \in T$ of the deterministic problem $1//\sum w_i\mathcal{C}_i$.

Definition 13 [110] *The set of permutations $S(T) \subseteq S$ is a minimal dominant set (MDS for short) for the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$, if for any fixed scenario $p \in T$, the set $S(T)$ contains at least one optimal permutation for the instance $1/p/\sum w_i\mathcal{C}_i$, provided that any proper subset of set $S(T)$ loses such a property.*

The above results valid for the deterministic problem $1//\sum w_i\mathcal{C}_i$ will be used for finding a minimal dominant set $S(T)$ for the uncertain problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$ in the general case and in the two special cases, i.e., when $|S(T)| = 1$ or $|S(T)| = n!$. A minimal dominant set $S(T)$ for the uncertain problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$ will be obtained by constructing a precedence-dominance relation on the set of jobs $\mathcal{J}$. We define a precedence-dominance relation as follows.

Definition 14 *The job J_u dominates the job J_v with respect to T (denoted by $J_u \mapsto J_v$) if there exists a minimal dominant set $S(T)$ for the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$ such that the job J_u precedes the job J_v in every permutation of the set $S(T)$.*

From Definition 14, it follows that a minimal dominant set $S(T)$ constructed for the deterministic problem $1//\sum w_i\mathcal{C}_i$ associated with the vector $p \in T$ of the job processing times is a singleton: $S(T) = \{\pi_k\}$, where the set T is also a singleton, $T = \{p\}$. As a result, the precedence-dominance relations

$$J_{k_1} \mapsto J_{k_2} \mapsto J_{k_3} \mapsto \ldots \mapsto J_{k_{n-1}} \mapsto J_{k_n}$$

with respect to the set $T = \{p\}$ hold for the deterministic problem $1//\sum w_i\mathcal{C}_i$.

Theorem 23 *For the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$, the job J_u dominates the job J_v with respect to T if and only if the following inequality holds:*

$$\frac{w_u}{b_u} \geq \frac{w_v}{a_v}. \tag{6.3}$$

The cardinality of a minimal dominant set $S(T)$ may be considered as a measure of the uncertainty of the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$: The uncertainty in the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$ is less, if the cardinality of the set $S(T)$ is smaller.

The inclusion $S(T) \subseteq S$ implies the inequalities $1 \leq |S(T)| \leq n!$. Next, we present characterizations of two extreme cases of a minimal dominant set.

6.2.1 Dominant singleton: $|S(T)| = 1$

The simplest case of the uncertain problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$ arises when the equality $|S(T)| = 1$ holds. In such a case, a minimal dominant set for the uncertain problem is a singleton: $\{\pi_k\} = S(T)$ (as well as an ordinary solution to the deterministic problem $1//\sum w_i\mathcal{C}_i$).

Theorem 24 *For the existence of a dominant singleton $S(T) = \{\pi_k\} = \{(J_{k_1}, J_{k_2}, \ldots, J_{k_n})\}$ for the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$, it is necessary and sufficient that the following set of inequalities holds:*

$$\frac{w_{k_1}}{b_{k_1}} \geq \frac{w_{k_2}}{a_{k_2}}, \frac{w_{k_2}}{b_{k_2}} \geq \frac{w_{k_3}}{a_{k_3}}, \ldots, \frac{w_{k_{n-1}}}{b_{k_{n-1}}} \geq \frac{w_{k_n}}{a_{k_n}}. \tag{6.4}$$

If a dominant singleton $S(T) = \{\pi_k\}$ for the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$ exists, then the dominant permutation π_k can be obtained using Theorem 23 in $O(n^2)$ time. Next, we show that, as an application for this purpose, Theorem 24 allows us either to find a singleton $S(T)$ (if it exists) faster or to prove that a dominant singleton does not exist. For a fast checking of condition (6.4) of Theorem 24, we sort the jobs of the set $\mathcal{J}$ in non-increasing order of the fractions $\frac{w_{k_j}}{\overline{p}_{k_j}}$, where $\overline{p}_{k_j}$ denotes the midpoint of the segment $[a_{k_j}, b_{k_j}]$:

$$\overline{p}_{k_j} = \frac{b_{k_j} - a_{k_j}}{2}.$$

As a result, we obtain a permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$ in $O(n \log_2 n)$ time. Due to the sufficiency in Theorem 24, if condition (6.4) holds, then this permutation π_k defines a dominant singleton $S(T) = \{\pi_k\}$ for the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$. On the other hand, it is easy to show that, if conditions (6.4) does not hold for all pairs of consecutive jobs in the permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n})$, then a dominant singleton for the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$ does not exist. It takes $O(n)$ time to check conditions (6.4) for a fixed permutation π_k of n jobs. Thus, the asymptotic complexity of checking the condition of Theorem 24 can be estimated as $O(n \log_2 n)$.

6.2.2 MDS with maximal cardinality: $|S(T)| = n!$

The most uncertain case of the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$ arises when $|S(T)| = n!$.

Theorem 25 *Let $a_i < b_i$ for each job $J_i \in \mathcal{J}$. For the existence of a minimal dominant set $S(T)$ for the problem $1/a_i \leq p_i \leq b_i/\sum w_i\mathcal{C}_i$ with the maximal cardinality $|S(T)| = n!$, it is sufficient that the following inequality holds:*

$$\max\{\frac{w_i}{b_i} \mid J_i \in \mathcal{J}\} < \min\{\frac{w_i}{a_i} \mid J_i \in \mathcal{J}\}. \quad (6.5)$$

It takes $O(n)$ time to check the condition (6.5) of Theorem 25 since $O(n)$ is the complexity of finding a maximum (minimum) in the set $\{\frac{w_i}{b_i} \mid J_i \in \mathcal{J}\}$ of real numbers (set $\{\frac{w_i}{a_i} \mid J_i \in \mathcal{J}\}$, respectively).

6.3 Stability box

In [112], the stability box $\mathcal{SB}(\pi_k, T)$ within a set of scenarios T has been defined for a permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$. We denote $\mathcal{J}^-[k_i] = \{J_{k_1}, J_{k_2}, \ldots, J_{k_{i-1}}\}$ and $\mathcal{J}^+[k_i] = \{J_{k_{i+1}}, J_{k_{i+2}}, \ldots, J_{k_n}\}$. Let S_{k_i} be the set of permutations $(\pi(\mathcal{J}^-[k_i]), J_{k_i}, \pi(\mathcal{J}^+[k_i])) \in S$ of the jobs $\mathcal{J}$, $\pi(\mathcal{J}')$ being a permutation of the jobs $\mathcal{J}' \subset \mathcal{J}$.

Definition 15 [112] *Let the permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$ be optimal for at least one scenario $p \in T$. The maximal segment $[l_{k_i}, u_{k_i}] \subseteq [a_{k_i}, b_{k_i}]$ is called the maximum variation of the processing time of the job J_{k_1} in the permutation π_k if for any permutation $\pi_e \in S_{k_i}$ and any scenario $p = (p_1, p_2, \ldots, p_n) \in T$, at which the permutation π_e is optimal, the permutation π_e remains optimal for any scenario p' from the set*

$$\times_{j=1}^{k_i-1}[p_j, p_j] \times [l_{k_i}, u_{k_i}] \times_{j=k_i+1}^{n} [p_j, p_j]$$

and for any scenario $p'' = (p''_1, p''_2, \ldots, p''_n) \in T$ with $p''_{k_i} \in [l_{k_i}, u_{k_i}]$, there exists an optimal permutation $\pi_v \in S_{k_i}$ for the instance $1/p''/\sum w_i\mathcal{C}_i$. Let N_k denote the set of indexes i of all jobs with their non-empty maximum variations of the processing times. The Cartesian product

$$\mathcal{SB}(\pi_k, T) = \times_{k_i \in N_k}[l_{k_i}, u_{k_i}] \subseteq T$$

is a stability box for the permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$. If there does not exist a scenario $p \in T$ such that the permutation π_k is optimal for the instance $1/p/\sum w_i\mathcal{C}_i$, then $\mathcal{SB}(\pi_k, T) = \emptyset$.

The maximality of the closed rectangular box $\mathcal{SB}(\pi_k, T) = \times_{k_i \in N_k}[l_{k_i}, u_{k_i}]$ in Definition 15 means that the box $\mathcal{SB}(\pi_k, T) \subseteq T$ has both a maximal possible dimension $|N_k|$ and a maximal possible volume. In Definition 15, the maximality of a rectangular box $\mathcal{SB}(\pi_k, T)$ means that for each position $i \in \{1, 2, \ldots, n\}$ in permutation π_k, the lower bound l_{k_i} (the upper bound u_{k_i}) for the variation of the processing time p_{k_i} of the job J_{k_i}, which is located at position i in the permutation π_k, preserving the optimality of the permutation π_k has to be as small (as large) as possible provided that the processing time of each other job J_{k_j}, $j \in \{1, 2, \ldots, n\} \setminus \{i\}$, may vary independently and simultaneously within the whole given segment $[a_{k_j}, b_{k_j}]$. We call the dimension of a stability box $\mathcal{SB}(\pi_k, T)$ the cardinality $|N_k|$ of the set $\{p_{k_i} \mid k_i \in N_k\}$ of the processing times in the scenario p' which may be modified in the vector p with preserving the optimality of the permutation π_k. The cardinality $|N_k|$ of the set N_k is an important characteristic of the stability box $\mathcal{SB}(\pi_k, T)$: It defines the maximum number of the processing times in p' which are modifiable in the scenario p without violating the optimality of the permutation π_k. Note that the processing times of the remaining set $\{p'_{k_j} \mid k_j \in N \setminus N_k\}$ have to remain the same as those in the original vector $p : p'_{k_j} = p_{k_j}$.

In [98, 127], the stability region of an optimal semi-active schedule was investigated for the job-shop problem with the makespan and mean flow time criterion, respectively. Using the notations introduced for the problem $1/a_i \le p_i \le b_i / \sum w_i \mathcal{C}_i$, the stability region $\mathcal{K}(\pi_k, T)$ of a permutation $\pi_k \in S$ with respect to T is defined as follows:

$$\mathcal{K}(\pi_k, T) = \left\{ p \mid p \in T, \sum_{J_i \in \mathcal{J}} w_i C_i(\pi_k, p) = \min_{\pi_l \in S} \left\{ \sum_{J_i \in \mathcal{J}} w_i C_i(\pi_l, p) \right\} \right\}. \quad (6.6)$$

Definition 15 of a stability box and Definition (6.6) of a stability region imply the inclusion $\mathcal{SB}(\pi_k, T) \subseteq \mathcal{K}(\pi_k, T)$. In [112], the following properties of the stability box and the stability region were derived.

Theorem 26 [112] *The stability box $\mathcal{SB}(\pi_k, T)$ (the stability region $\mathcal{K}(\pi_k, T)$) is empty, if and only if there is no scenario $p \in T$ such that permutation π_k is optimal for the instance $1/p/\sum w_i \mathcal{C}_i$.*

Theorem 27 [112] *There exists a scenario $p \in T$ such that permutation $\pi_k = (J_{k_1}, J_{k_2}, \ldots, J_{k_n}) \in S$ is optimal for the instance $1/p/\sum w_i \mathcal{C}_i$ if and only if there is no job J_{k_i}, $i \in \{1, 2, \ldots, n-1\}$, such that inequality*

$$\frac{w_{k_i}}{a_{k_i}} < \frac{w_{k_j}}{b_{k_j}} \quad (6.7)$$

holds for at least one job J_{k_j}, where $j \in \{i+1, i+2, \ldots, n\}$.

Theorem 28 [112] *The stability box $\mathcal{SB}(\pi_k, T)$ (stability region $\mathcal{K}(\pi_k, T)$) is empty, if and only if there exists a job J_{k_i}, $i \in \{1, 2, \ldots, n-1\}$, such that inequality (6.7) holds for at least one job J_{k_j}, where $j \in \{i+1, i+2, \ldots, n\}$.*

Definitions 15 and (6.6) imply the following claim.

Theorem 29 [112] *If there exists exactly one scenario $p \in T$ such that the permutation $\pi_k \in S$ is optimal for the instance $1/p/\sum w_i \mathcal{C}_i$, then $\mathcal{SB}(\pi_k, T) = \{p\} = \mathcal{K}(\pi_k, T)$.*

Theorem 30 characterizes another extreme case for the stability box and region.

Theorem 30 [112] $\mathcal{SB}(\pi_k, T) = T = \mathcal{K}(\pi_k, T)$ *if and only if inequalities (6.4) hold.*

As follows from Definition 13, one can restrict the search by the permutations of a minimal dominant set $S(T)$ while solving the problem $1/a_i \le p_i \le b_i / \sum w_i \mathcal{C}_i$.

Theorem 31 [112] *If $\pi_k \in S(T)$, then $\mathcal{SB}(\pi_k, T) \neq \emptyset$ and $\mathcal{K}(\pi_k, T) \neq \emptyset$.*

In [110, 112], an $O(n \log n)$ algorithm was developed for calculating the stability box.

6.4 A job permutation with the largest stability box

A job permutation in the set S with a larger dimension and a larger volume of the stability box seems to be more efficient than one with a smaller dimension and (or) a smaller volume of the stability box. We present some properties of a stability box, which allow us to derive an $O(n \log n)$ algorithm for choosing a permutation $\pi_t \in S$ which has (a) the largest dimension $|N_t|$ of the stability box $\mathcal{SB}(\pi_t, T) = \times_{t_i \in N_t}[l_{t_i}, u_{t_i}] \subseteq T$ among all permutations $\pi_k \in S$ and (b) the largest relative volume $Vol\mathcal{SB}(\pi_t, T)$ of the stability box $\mathcal{SB}(\pi_t, T)$ among all permutations $\pi_k \in S$ having the largest dimension $|N_k| = |N_t|$ of their stability boxes $\mathcal{SB}(\pi_k, T)$ and the minimal number n^k of the maximum variations with zero length. Definition 15 implies the following claim.

Property 1 *For any jobs $J_i \in \mathcal{J}$ and $J_v \in \mathcal{J}$, $v \neq i$, we have*

$$\left(\frac{w_i}{u_i}, \frac{w_i}{l_i}\right) \bigcap \left(\frac{w_v}{b_v}, \frac{w_v}{a_v}\right) = \emptyset.$$

Let S^{max} denote the subset of all permutations π_t in the set S possessing properties (a) and (b). Using Property 1, we can define the relative order of a job $J_i \in \mathcal{J}$ with respect to a job $J_v \in \mathcal{J}$ for any $v \neq i$ in a permutation $\pi_t = (J_{t_1}, J_{t_2}, \dots, J_{t_n}) \in S^{max}$. To this end, we have to treat all three possible cases (I)–(III) for the intersection of the open interval $\left(\frac{w_i}{b_i}, \frac{w_i}{a_i}\right)$ and the closed interval $\left[\frac{w_v}{b_v}, \frac{w_v}{a_v}\right]$. The order of the jobs J_i and J_v in the desired permutation $\pi_t \in S^{max}$ may be defined in the cases (I)–(III) using the following rules. Case (I) is defined by the inequalities

$$\frac{w_v}{b_v} < \frac{w_i}{b_i}, \frac{w_v}{a_v} < \frac{w_i}{a_i}. \tag{6.8}$$

In case (I), the desired order of the jobs J_v and J_i in the permutation $\pi_t \in S^{max}$ may be defined by a strict inequality from (6.8): Job J_v proceeds job J_i in the permutation π_t.

Property 2 *For the case (I), there exists a permutation $\pi_t \in S^{max}$, in which the job J_v proceeds the job J_i.*

Case (II) is defined by the equalities

$$\frac{w_v}{b_v} = \frac{w_i}{b_i}, \frac{w_v}{a_v} = \frac{w_i}{a_i}. \tag{6.9}$$

Property 3 *For the case (II), there exists a permutation $\pi_t \in S^{max}$, in which the jobs J_i and J_v are located adjacently: $i = t_r$ and $v = t_{r+1}$.*

If equalities (6.9) hold, one can restrict the search for a permutation $\pi_t \in S^{max}$ by a subset of permutations in the set S with adjacently located jobs J_i and J_v (Property 3). Moreover, the order of such jobs $\{J_i, J_v\}$ does not influence the volume of the stability box and its dimension.

Remark 8 *Due to Property 3, while looking for a permutation $\pi_t \in S^{max}$, we shall treat a pair of jobs $\{J_i, J_v\}$ satisfying (6.9) as one job (either job J_i or J_v).*

Case (III) is defined by the strict inequalities

$$\frac{w_v}{b_v} \geq \frac{w_i}{b_i}, \frac{w_v}{a_v} \leq \frac{w_i}{a_i}. \tag{6.10}$$

provided that at least one of the inequalities (6.10) is strict. For a job $J_i \in \mathcal{J}$ satisfying case (III), let $\mathcal{J}(i)$ denote the set of all jobs $J_v \in \mathcal{J}$, for which the strict inequalities (6.10) hold.

Property 4 *(i) For a fixed permutation $\pi_k \in S$, job $J_i \in \mathcal{J}$ may have at most one maximal segment $[l_i, u_i]$ of the possible variations of the processing time $p_i \in [a_i, b_i]$ preserving the optimality of permutation π_k.*

(ii) For the whole set of permutations S, only in case (III), a job $J_i \in \mathcal{J}$ may have more than one (namely: $|\mathcal{J}(i)| + 1 > 1$) maximal variation $[l_i, u_i]$ of the time $p_i \in [a_i, b_i]$ preserving the optimality of this or that particular permutation from the set S.

Let $\mathcal{L}$ denote the set of pairs $(i, [l_i, u_i])$ consisting of all non-empty maximal variations $[l_i, u_i]$ of the processing times p_i for all jobs $J_i \in \mathcal{J}$. Let $\mathcal{L}'$ denote the set of pairs $(i, [l_i, u_i])$ consisting of the greatest maximal variations $[l_i, u_i]$ of the processing times p_i for all jobs $J_i \in \mathcal{J}$ on all permutations from the set S.

Property 5 $|\mathcal{L}'| \leq n$.

Property 6 *There exists a permutation $\pi_t \in S$ with the set of pairs $(i, [l_i, u_i])$, where i is the job number, $[l_i, u_i]$ denotes the of maximal variations of the processing time $p_i, J_i \in \mathcal{J}$, such that the set $(i, [l_i, u_i])$ coincides with the set $L' \subseteq L$.*

The above properties allow us to derive an $O(n \log n)$ algorithm for calculating a permutation $\pi_t \in S^{max}$ with the largest dimension $|N_t|$ and the largest volume of a stability box $\mathcal{SB}(\pi_t, T)$. This algorithm constructs a permutation $\pi_t \in S$ such that the dimension $|N_t|$ of the stability box

$$\mathcal{SB}(\pi_t, T) = \times_{t_i \in N_t} [l_{t_i}, u_{t_i}] \subseteq T$$

is the largest one for all permutations S, and the volume of the stability box $\mathcal{SB}(\pi_t, T)$ is the largest one for all permutations $\pi_k \in S$ having the largest dimension $|N_k| = |N_t|$ of their stability boxes $\mathcal{SB}(\pi_t, T)$ and the minimal counter n^k of the *maximal variations* with zero length.

6.5 Notes and references

All the results presented in this section have been obtained and (or) generalized in the papers [6, 27, 28, 105, 110, 111, 112, 113, 115, 118, 120, 121, 122, 125].

In spite of obvious practical importance, the literature on a stability and sensitivity analysis in scheduling is rather small. Outside the considered stability approach, one can mention [41, 49, 78]. In [49], the sensitivity of a heuristic algorithm with respect to the variation of the processing time of one job was investigated. In [78], the results for the traveling salesman problem were used for a single machine scheduling problem with minimizing tardiness (see [58]). A sensitivity analysis for some single-stage scheduling problems was developed by Hall and Posner [41]. In particular, they developed a sensitivity analysis for the problems $1//\sum w_i \mathcal{C}_i$,

$P//\sum \mathcal{C}_i$, $1//\mathcal{L}_{max}$ and other *list scheduling problems*, i.e., those that are optimally solvable by a sequence of jobs generated due to a known priority rule. In [41], it was also shown that a modified instance (when a concrete change of one processing time or one job weight is given) of some polynomially solvable problems (among them the problems $1//\sum w_i\mathcal{C}_i$, $1/r_i, pmtn/\sum \mathcal{C}_i$, and $1/r_i, d_i, p_i = 1/\sum w_i\mathcal{C}_i$) can be solved more efficiently using the optimal schedule known for the original problem instance. A systematic study of sensitivity analysis for single-stage scheduling problems was initiated in [41]. Hall and Posner [41] described a sufficient condition under which an increase in the set of feasible schedules does not change the optimal non-preemptive schedule for a regular criterion and without given precedence constraints. *If the deadline d_k of job J_k is increased when there is a consecutive idle time of at least* $\max_{J_i \in J}\{P_k : C_i(s) > C_k(s)\}$ *in the closed interval $[C_k(s), d_k]$, then the originally optimal schedule s remains optimal.* For a bottleneck criterion (like minimizing the maximum lateness), the following sufficient condition was given in [41] as well: *If an increase in operation processing time does not create a new bottleneck or increase the current bottleneck, then the originally optimal schedule remains optimal.* In Sections 2 – 7 of this survey, it was assumed that multiple processing time changes are independent. However, in some real world scheduling problems, multiple parameter changes are related. As it was mentioned in [41], this may be due to the relationships between the numerical parameters, such as the correlation between the value and the work required. It may also be due to the modelling process. In a systematic study [41], Hall and Posner demonstrated for some single-stage scheduling problems that a sensitivity analysis may depend on the position of the jobs with the changed parameters. They also identified single-stage scheduling problems where performing additional computations during optimization facilitates a sensitivity analysis.

The stability of an optimal line balance for a fixed number of stations was investigated in [25, 107, 108, 109]. The assembly line balancing problem with variable operation times was considered, e.g., in [33, 59, 86, 138]. In [33, 138], fuzzy set theory was used to represent uncertainty of the operation times. Genetic algorithms were used either to minimize the total operation time for each station [33] or to minimize the efficiency of the fuzzy line balance [138]. In [59], the entire decision process was decomposed into two parts: The deterministic problem and the stochastic problem. For the former problem, integer programming was used to minimize the number of stations. For the latter problem, which takes into account the variations of the operation times over different products, queuing network analysis was used to determine the necessary capacity of the material-handling system. In [86], dynamic programming and branch-and-bound methods were used to minimize the total labor cost and the expected incompletion cost arising from the operations not completed within the cycle time c. Branch-and-bound algorithms were developed in [18, 81, 86, 87], integer programming algorithms in [24, 59, 81].

Kouvelis et al. [50] focused on manufacturing environments, where the job processing times are uncertain. In these settings, scheduling decision-makers are exposed to the risk that an optimal schedule with respect to a deterministic or stochastic model will perform poorly when evaluated relative to the *actual* processing times. *Robust scheduling*, i.e., determining a schedule whose performance (compared to the associated optimal schedule) is relatively insensitive to the potential realizations of the job processing times. A similar robust scheduling approach was developed for a single-machine problem by Daniels and Kouvelis [21]. Other robust decision-making formulations were presented by Rosenblatt and Lee [84], Kouvelis et al. [51] and Mulvey et al. [76]. Leon et al. [63] considered robustness measures and robust scheduling methods that generate job shop schedules that maintain a high performance over a range of system disturbances. Wu et al. [139] proposed a solution approach to the weighted tardiness job shop problem as follows. To identify a critical subset of the scheduling decisions at the beginning of the planning horizon and to relegate the rest of the scheduling decisions to future points in time. The above research presents a philosophy of great practical importance: Local scheduling should be allowed sufficiently flexible without losing a global view of the system.

The complexity of a sensitivity analysis was investigated in the paper [19] which is based largely on the unpublished work [82], see also [83]. The complexity of calculating the stability radius of the optimal schedule and of a solution of the minimization of a linear form was considered in [35, 36, 37, 38, 93, 95, 96, 97, 99, 100].

In spite of a large number of papers and books published about scheduling, the utilization of numerous results of scheduling theory in most production environments is far from the desired volume. One of the reasons for such a gap between scheduling theory and practice is connected with the usual assumption that the processing times of the jobs are known exactly before scheduling or that they are random values with a priori known probability distributions. In this work, a model of one of the more realistic scheduling scenarios was considered. It is assumed that in the realization of a schedule, the job processing time may take any real value between given lower and upper bounds (within the given polytope T), and there is no prior information about the probability distributions of the random processing times. For such an uncertain scheduling problem, there does usually not exist a unique schedule that remains optimal for all possible realizations of the processing times and a set of schedules has to be considered which dominates all other schedules for the given criterion. To find such a set of schedules, our idea was to use the schedule domination and the stability analysis of an optimal schedule with respect to the perturbations of the processing times; see Section 3.

7 Conclusion

The stability approach considered in this work combines a stability analysis, a multi-stage decision framework, and the solution concept of a minimal dominant set of semi-active schedules. The general scheme of the stability method for an uncertain scheduling problem is presented in Fig. 1. The realization of this scheme for the uncertain two-stage scheduling problem was given in Section 5 in the form of Algorithm 1 for off-line scheduling (see page 39) and Algorithm 2 for on-line scheduling (see page 39).

The material of this survey is presented in an order from hard scheduling problems to easier scheduling problems. Sections 3 and 4 deal with general and job shop problems without any restriction on the number of processing stages, Section 5 deals with a two-machine flow shop problem and a job shop problem with two stages, and Section 6 deals with a single-machine scheduling problem. Such an order of the problems allows us to present mathematical results in the opposite order: from simple results to more complicated and deep ones. The notion of the relative stability radius of an optimal schedule s was introduced as the maximal value of the radius of a stability ball within which a schedule s remains the best among the given set B of schedules; see Section 4 The relativity was considered with respect to the polytope T of the feasible vectors of the processing times and with respect to the set B of semiactive schedules for which the superiority of a schedule s at hand has to be guaranteed. In Sections 3 and 4, we used the mixed (disjunctive) graph model which is suitable for the whole scheduling process from the initial mixed graph G representing the input data until a final digraph G_s representing a semiactive schedule s. The mixed graph model may be used for different requirements on the numerical input data. Most results of Sections 3 and 4 were formulated in terms of paths in the digraphs G_s.

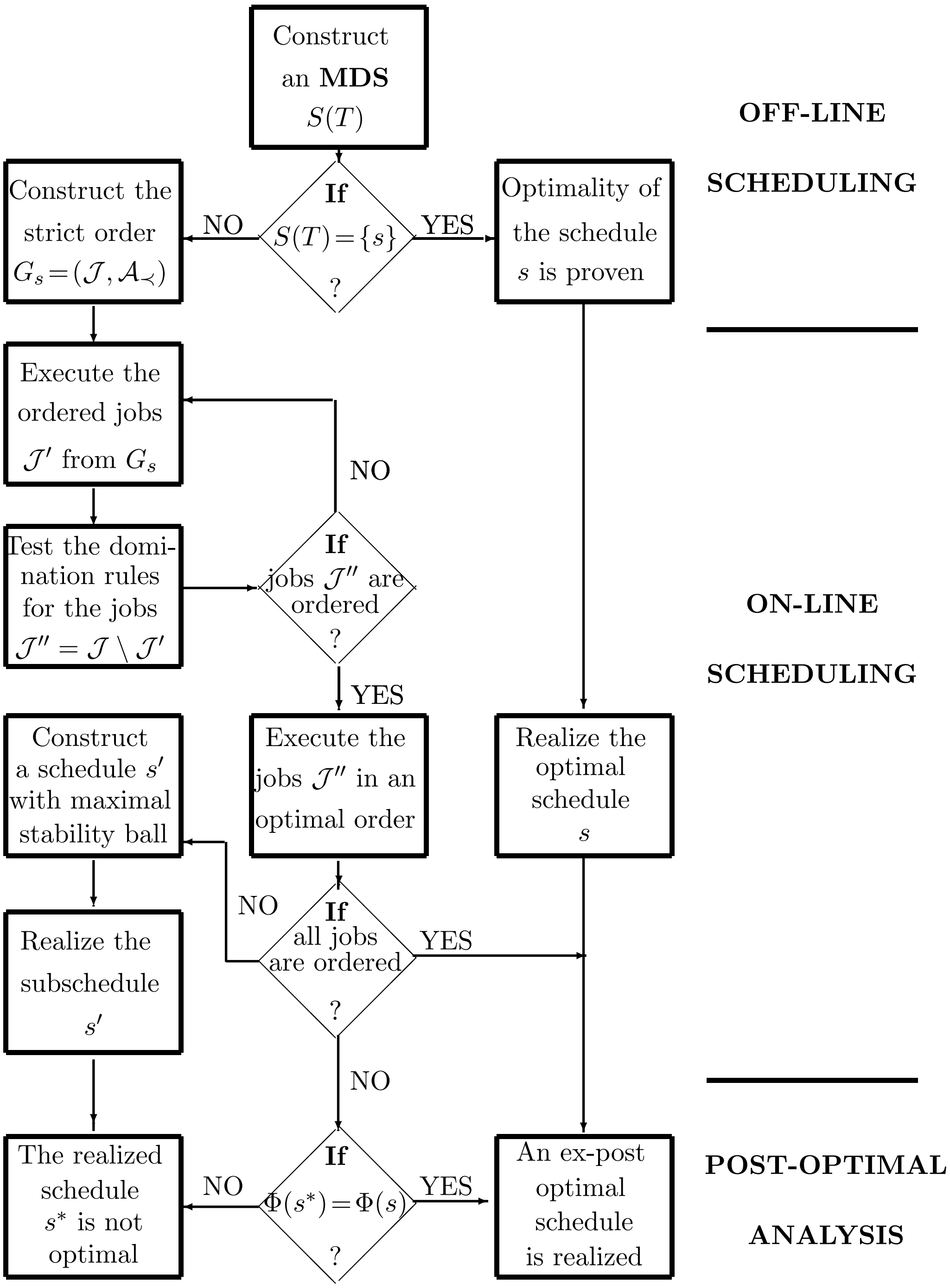

Figure 1: Scheme of the stability method for an uncertain scheduling problem

In Section 4, we focused on dominance relations between feasible schedules taking into account the given polytope T (Subsection 4.2). We established necessary and sufficient conditions for the case of an infinitely large relative stability radius of an optimal schedule s for the maximum flow time criterion (see Theorem 9 on page 21). Under such conditions, schedule s remains optimal for any feasible perturbations of the processing times. We established also necessary and sufficient conditions for the case of a zero relative stability radius of an optimal schedule s (see Theorem 8 on page 21). Under such conditions, the optimality of schedule s is unstable: there are some small changes of the given processing times which imply that another schedule from the set B will be better (will have a smaller length) than schedule s. Formulas for calculating the exact value of the relative stability radius were based on a comparison of an optimal schedule s with other schedules from the set B (see Theorem 10 on page 22), and we showed how it is possible to restrict the number of schedules from the set B examined for such a calculation of the relative stability radius (see Lemma 8 on page 24). To this end, we considered the schedules from the set B in non-decreasing order of the values of the objective function until some inequalities hold (see Corollary 8 on page 25). In Section 4, analogous results were obtained for the mean flow time criterion, and the focus was on dominance relations between the feasible schedules taking into account the given criterion. Formulas for calculating the exact value of the relative stability radius were given. We established necessary and sufficient conditions for an infinitely large relative stability radius of an optimal schedule for the mean flow time criterion and necessary and sufficient conditions for a zero relative stability radius of an optimal schedule. Using these results, we developed several exact and heuristic algorithms for constructing a solution (MDS) (see Definition 3 on page 19) of a scheduling problem with uncertain processing times.

If no additional criterion is introduced (contrary to a robust method), there is no information on the probability distributions of the random processing times (contrary to a stochastic method) and there are no membership functions for non-script processing times (contrary to a fuzzy method), then a stability method may be used for a problem $\alpha/a_{ij} \leq p_{ij} \leq b_{ij}/\gamma$. This method combines a stability analysis, a multi-stage decision framework and the solution concept of a minimal dominant set (MDS). In [113, 115], an MDS was used to measure the uncertainty of the problem $\alpha/a_{ij} \leq p_{ij} \leq b_{ij}/\gamma$. This measure allows a scheduler to select a suitable method for solving an instance of the uncertain problem $\alpha/a_{ij} \leq p_{ij} \leq b_{ij}/\gamma$.

References

[1] Abdelmaguid, T.: Permutation-induced acyclic networks for the job shop scheduling problem. Applied Mathematics Modelling **V. 33**, 1560–1572 (2009)

[2] Akers, S.: A graphical approach to production scheduling problems. Operations Research **V. 4**, 244–245 (1956)

[3] Allahverdi, A.: Two-machine flowshop scheduling problem to minimize total completion time with bounded setup and processing times. International Journal of Production Economics **V. 103**, 386–400 (2006)

[4] Allahverdi, A., Aldowaisan, T., Sotskov, Y.: Two-machine flowshop scheduling problem to minimize makespan or total completion time with random and bounded setup times. International Journal of Mathematics and Mathematical Sciences **V. 39**(11), 2475–2486 (2003)

[5] Allahverdi, A., Sotskov, Y.: Two-machine flowshop minimum-length scheduling problem with random and bounded processing times. International Transactions in Operations Research **V. 10**(1), 65–76 (2003)

[6] Alyushkevich, V., Sotskov, Y.: Stability in production scheduling problem. Vesti Akad. Navuk BSSR, Ser. Fiz.-Techn. Navuk (3), 102–107 (1989). (In Russian)

[7] Bräsel, H., Harborth, M., Tautenhahn, T., Willenius, P.: On the hardness of the classical job shop problem. Annals of Operations Research **V. 92**, 265–279 (1999)

[8] Bräsel, H., Harborth, M., Tautenhahn, T., Willenius, P.: On the set of solutions of the open shop problem. Annals of Operations Research **V. 92**, 241–263 (1999)

[9] Bräsel, H., Kleinau, M.: New steps in the amazing world of sequences and schedules. Mathematical Methods of Operations Research **V. 43**, 195–214 (1996)

[10] Bräsel, H., Sotskov, Y., Werner, F.: Stability of a schedule minimizing mean flow time. Mathematical and Computer Modelling **V. 24**(10), 39–53 (1996)

[11] Braun, O., Lai, T.C., Schmidt, G., Sotskov, Y.: Stability of Johnson's schedule with respect to limited machine availability. International Journal of Production Research **V. 40**(17), 4381–4400 (2002)

[12] Braun, O., Leshchenko, N., Sotskov, Y.: Optimality of jackson's permutation with respect to limited machine availability. International Transactions in Operations Research **V. 13**, 59–74 (2006)

[13] Briand, C., La, H., Erschler, J.: A new sufficient condition of optimality for the two-machine flowshop problem. European Journal of Operational Research **V. 169**, 712–722 (2006)

[14] Brucker, P.: An efficient algorithm for the job-shop problem with two jobs. Computing **V. 40**, 353–359 (1988)

[15] Brucker, P.: Scheduling Algorithms. Springer, Berlin (1995)

[16] Brucker, P., Kravchenko, S., Sotskov, Y.: Preemptive job-shop scheduling problems with fixed number of jobs. Mathematical Methods of Operations Research **V. 49**, 41–76 (1999)

[17] Brucker, P., Sotskov, Y., Werner, F.: Complexity of shop-scheduling problems with fixed number of jobs: a survey. Mathematical Methods of Operations Research **V. 65**, 461–481 (2007)

[18] Bukchin, J., Tzur, M.: Design of flexible assembly line to minimize equipment cost. IIE Transactions **V. 32**, 585–598 (2000)

[19] Chakravarti, N., Wagelmans, A.: Calculation of stability radii for combinatorial optimization problems. Operations Research Letters **V. 23**(1), 1–7 (1998)

[20] Chretienne, P., Coffman, E., Lenstra, J., Liu, Z.e.: Scheduling Theory and its Applications. John Wiley & Sons (1995)

[21] Daniels, R., Kouvelis, P.: Robust scheduling to hedge against processing time uncertainty in single stage production. Management Science **V. 41**(2), 363–376 (1995)

[22] Dhamala, T.: On the potentially optimal solutions of classical shop scheduling ptoblems. International Journal of Operations Research **V. 4**(2), 80–89 (2007)

[23] Dijkstra, E.: A note on two problems in connection with graphs. Numerische Mathematik **V. 1**, 269–271 (1959)

[24] Dolgui, A., Guschinski, N., Harrath, Y., Levin, G.: Une approche de programmation linaire pour la conception des lignes de transfert. European Journal of Automated Systems (APII-JESA) **V. 36**(1), 11–33 (2002)

[25] Dolgui, A., N.Y., S.: Stability analysis for assembly lines sequencing with uncertain processing times. In: Proceedings of the Fifth Workshop on Models and Algorithms for Planning and Scheduling Problems MAPSP'01. Technische Universität Bergakademie Freiberg, Aussois, France (2001)

[26] Drotos, M., Erdos, G., Kis, T.: Computing lower and upper bounds for a large-scale industrial job shop scheduling problem. European Journal of Operational Research **V. 197**, 296–306 (2009)

[27] Egorova, N., Matsveichuk, N., Sotskov, Y.N.: Selection of an optimal order for processing jobs on two machines during scheduling realization. Vesti Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk **V. 5**, 20–24 (2006). (In Russian)

[28] Egorova, N., Sotskov, Y.: Minimization weighted sum of job completion times with interval durations. Informatics (3), 5–16 (2006). (In Russian)

[29] Emelichev, V., Girlih, E., Nikulin, Y., Podkopaev, D.: Stability and regularization of vector problem of integer linear programming. Optimization **V. 51**(4), 645–676 (2002)

[30] Emelichev, V., Kuz'min, K., Nikulin, Y.: Stability analysis of the Pareto optimal solutions for some vector boolean optimaization problem. Optimization **V. 54**(6), 545–561 (2005)

[31] Emelichev, V., Podkopaev, D.: On a quantitative measure of the stability of a vector problem of integer programming. U.S.S.R. Computational Mathematics and Mathamatical Physics **V. 38**(11), 1727–1731 (1998). (Translated from Russian)

[32] Garey, M., Johnson, D., Sethi, R.: The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research **V. 1**, 117–129 (1976)

[33] Gen, M., Tsujimura, Y., Li, Y.: Fuzzy assembly line balancing using genetic algorithms. Computers & Industrial Engineering **V. 31**(3/4), 631–634 (1996)

[34] Giffler, B., Thompson, G.: Algorithms for solving production-scheduling problems. Operations Research **V. 8**(4), 487–503 (1960)

[35] Gordeev, E.: Algorithms of polynomial complexity for computing the stability radius in two classes of trajectory problems. U.S.S.R. Computational Mathematics and Mathematical Physics **V. 27**(4), 14–20 (1987)

[36] Gordeev, E.: Solution stability of the shortest path problem. Discrete Mathematics **V. 1**(3), 45–56 (1989). (In Russian)

[37] Gordeev, E., Leontev, V.: The complexity of the tabulation of trajectory problems. U.S.S.R. Computational Mathematics and Mathematical Physics **V. 25**(4), 199–201 (1985)

[38] Gordeev, E., Leontev, V., Sigal, I.: Computational algorithms for finding stability radius in choice problems. U.S.S.R. Computational Mathematics and Mathematical Physics **V. 23**(4), 128–132 (1983)

[39] Greenberg, H.: A bibliography for the development of an intelligent mathematical programming system. Annals of Operations Research **V. 65**, 55–90 (1996). http://orcs.bus.okstate.edu/itorms

[40] Greenberg, H.: An annotated bibliography for post-solution analysis in mixed integer programming and combinatorial optimization. In: D. Woodruff (ed.) Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search, pp. P. 97–148. Kluwer Academic Publishers, Boston, MA (1998)

[41] Hall, N., Posner, M.: Sensitivity analysis for scheduling problems. Journal of Scheduling **V. 7**, 49–83 (2004)

[42] Hardgrave, W., Nemhauser, G.: A geometric model and graphical algorithm for a sequencing problem. Operations Research **V. 11**, 889–900 (1963)

[43] Herroelen, W., Leus, R.: Robust and reactive project scheduling: a review and classification procedures. International Journal of Production Research **V. 42**(8), 1599–1620 (2004)

[44] Herroelen, W., Leus, R.: Project scheduling under uncertainty: Survey and research potentials. European Journal of Operational Research **V. 165**, 289–306 (2005)

[45] Ishii, H., Masuda, T., Nishida, T.: Two machine mixed shop scheduling problems with controllable machine speeds. Discrete Applied Mathematics **V. 17**, 29–38 (1987)

[46] Jackson, J.: An extension of johnson's results on job lot scheduling. Naval Research Logistics Quarterly **V. 3**(3), 201–203 (1956)

[47] Janiak, A.: General flow-shop scheduling with resource constraints. International Journal of Production Research **V. 26**, 125–138 (1988)

[48] Johnson, S.: Optimal two and three stage production schedules with set up times included. Naval Research Logistics Quarterly **V. 1**(1), 61–68 (1954)

[49] Kolen, A., Rinnooy Kan, A., Hoesel, C., Wagelmans, A.: Sensitivity analysis of list scheduling algorithms. Discrete Applied Mathematics **V. 55**, 145–162 (1994)

[50] Kouvelis, P., Daniels, R., Vairaktarakis, G.: Robust scheduling of a two-machine flow shop with uncertain processing times. IIE Transactions **V. 32**, 421–432 (2000)

[51] Kouvelis, P., Kurawarwala, A., Gutierrez, G.: Algorithms for robust single- and multiple-period layout planning for manufacturing systems. European Journal of Operational Research **V. 63**, 287–303 (1992)

[52] Kravchenko, S., Sotskov, Y.: Makespan optimal schedule with infinitely large stability radius. Vesti Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk (4), 85–91 (1993). (In Russian)

[53] Kravchenko, S., Sotskov, Y., Werner, F.: Optimal schedules with infinitely large stability radius. Optimization **V. 33**, 271–280 (1995)

[54] Lai, T.C., Sotskov, Y.: Sequencing with uncertain numerical data for makespan minimization. Journal of the Operations Research Society **V. 50**, 230–243 (1999)

[55] Lai, T.C., Sotskov, Y., Sotskova, N., Werner, F.: Optimal makespan scheduling with given bounds of processing times. Mathematical and Computer Modelling **V. 26**(3), 67–86 (1997)

[56] Lai, T.C., Sotskov, Y., Sotskova, N., Werner, F.: Mean flow time minimization with uncertain processing times. Otto-von-Guericke-Universität, FMA, Preprint No. 15/98, Magdeburg, Germany (1998)

[57] Lai, T.C., Sotskov, Y., Sotskova, N., Werner, F.: Mean flow time minimization with given bounds of processing times. European Journal of Operational Research **V. 159**, 558–573 (2004)

[58] Lawler, E., Lenstra, J., Rinnooy Kan, A., Shmoys, D.: Sequencing and scheduling: Algorithms and complexity. In: S. Graves, A.R. Kan, P. Zipkin (eds.) Handbooks in Operations Research and Management Science, vol. V. 4, pp. P. 445–522. Logistics of Production and Inventory, North-Holland, New York (1993)

[59] Lee, H., Johnson, R.: A line-balancing strategy for designing flexible assembly systems. International Journal of Flexible Manufacturing Systems **V. 3**, 91–120 (1991)

[60] Lenstra, J.: The mystical power of twoness: In memoriam eugene l. lawler. Journal of Scheduling **V. 1**, 3–14 (1998)

[61] Lenstra, J., Rinnooy Kan, A.: Computational complexity of discrete optimization problems. Annals of Discrete Mathematics **V. 4**, 121–140 (1979)

[62] Lenstra, J., Rinnooy Kan, A., Brucker, P.: Complexity of machine scheduling problems. Annals of Discrete Mathematics **V. 1**, 343–362 (1977)

[63] Leon, V., Wu, S., Storer, R.: Robustness measures and robust scheduling for job shops. IIE Transactions **V. 26**, 32–43 (1994)

[64] Leontev, V.: The stability of the traveling salesman problem. U.S.S.R. Computational Mathematics and Mathematical Physics **V. 15**(5), 199–213 (1975)

[65] Leontev, V.: Stability in combinatorial choice problems. Soviet Mathematics Doklady **V. 17**, 635–638 (1976)

[66] Leshchenko, N., Sotskov, Y.: Two-machine minimum-length shop-scheduling problems with uncertain processing times. In: Proceedings of XI-th International Conference "Knowledge-Dialogue-Solution", pp. P. 375–381. Varna, Bulgaria (2005)

[67] Leshchenko, N., Sotskov, Y.: A dominant schedule for the uncertain two-machine shop-scheduling problem. In: Proceedings of XII-th International Conference "Knowledge-Dialogue-Solution", pp. P. 291–297. Varna, Bulgaria (2006)

[68] Leshchenko, N., Sotskov, Y.N.: Finding the optimal order for processing two conflict jobs. Vesti Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk **V. 4**, 103–110 (2006). (In Russian)

[69] Leshchenko, N., Sotskov, Y.N.: Realization of an optimal schedule for the two-machine flow-shop with interval job processing times. International Journal: Information, Theories & Applications **V. 14**(2), 182–189 (2007)

[70] Lin, Y., Deng, J.: On the structure of all optimal solutions of the two-machine flowshop scheduling problem. Operational Research Transactions **V. 3**(2), 10–20 (1999)

[71] Lin, Y., Wang, X.: Necessary and sufficient conditions of optimality for some classical scheduling problems. European Journal of Operational Research **V. 176**, 809–818 (2007)

[72] Matsveichuk, N., Sotskov, Y., Egorova, N., Lai, T.C.: Schedule execution for two-machine flow-shop with interval processing times. Mathematical and Computer Modelling **V. 49**(5-6), 991–1011 (2009)

[73] Matsveichuk, N., Sotskov, Y., Werner, F.: Partial job order for solving the two-machine flow-shop minimum-length problem with uncertain processing times. Optimization **V. 60**(12), 1493–1517 (201)

[74] Melnikov, O.: Properties of the optimal schedules for the Bellman-Johnson problem with two machines. Vesti Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk (3), 19–25 (1976). (In Russian)

[75] Melnikov, O.: Optimal schedule stability for the Bellman-Johnson problem. Vesti Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk (6), 99–101 (1978). (In Russian)

[76] Mulvey, J., Vanderbei, R., Zenios, S.: Robust optimization of large-scale systems. Operations Research **V. 43**, 264–281 (1995)

[77] Ng, C., Matsveichuk, N., Sotskov, Y., Cheng, T.: Two-machine flow-shop minimum-length scheduling with interval processing times. Asia-Pacific Journal of Operational Research **V. 26**(6), 715–734 (2009)

[78] Picard, J., Queyranne, M.: The time-dependent traveling salesman problem and its application to the tardiness problem in one machine scheduling. Operations Research **V. 26**, 86–110 (1978)

[79] Pinedo, M.: Scheduling: Theory, Algorithms, and Systems. Prentice-Hall, Englewood Cliffs, NJ, USA (2002)

[80] Pinson, E.: The job shop scheduling problem: A concise survey and some recent development. In: P. Chretienne, E. Coffman, J. Lenstra, Z. Liu (eds.) Scheduling Theory and Its Applications, pp. 277–293. John Wiley & Sons (1995)

[81] Pinto, P., Dannenbring, D., Khumawala, B.: Assembly line balancing with processing alternatives: An application. Management Science **V. 29**, 817–830 (1983)

[82] Ramaswamy, R., Chakravarti, N.: Complexity of determining exact tolerances for min-sum and min-max combinatorial optimization problems. Tech. Rep. WPS-247/95, Indian Institute of Management, Calcutta (1995)

[83] Ramaswamy, R., Orlin, J., Chakravarti, N.: Sensitivity analysis for shortest path problems and maximum capacity path problems in undirected graphs. Mathematical Programming. Series A **V. 102**, 355–369 (2005)

[84] Rosenblatt, J., Lee, H.: A robustness approach to facilities design. International Journal of Production Research **V. 25**, 479–486 (1987)

[85] Roy, B., Sussmann, B.: Les problèmes d'ordonnancement avec contraintes disjonctives. Note DS No. 9 bis, SEMA, Montrouge (1964)

[86] Sarin, S., Erel, E., Dar-El, E.: A methodology for solving single-model, stochastic assembly line balancing problem. OMEGA - International Journal of Management Science **V. 27**, 525–535 (1999)

[87] Scholl, A., Klein, R.: Balancing assembly lines effectively: A computational comparison. European Journal of Operational Research **V. 114**, 51–60 (1998)

[88] Shakhlevich, N., Sotskov, Y., Werner, F.: Complexity of mixed shop scheduling problems: A survey. European Journal of Operational Research **V. 120**, 343–351 (2000)

[89] Slowinski, R., Hapke, M.: Scheduling Under Fuzziness. Physica-Verlag, Heidelberg, New York, Heidelberg, New York (1999)

[90] Smith, W.: Various optimizers for single-stage production. Naval Research Logistics Quarterly **V. 3**(1), 59–66 (1956)

[91] Sotskov, Y.: Optimal scheduling two jobs with regular criterion. In: Design Processes Automating, pp. P. 86–95. Institute of Engineering Cybernetics, Belarus, Minsk (1985). (In Russian)

[92] Sotskov, Y.: Stability of optimal schedule for a set of jobs. Vesti Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk (6), 99–104 (1988). (In Russian)

[93] Sotskov, Y.: The stability of high-speed optimal schedules. U.S.S.R. Computational Mathematics and Mathematical Physics **V. 29**(3), 57–63 (1989)

[94] Sotskov, Y.: Usage of stability of optimal schedule for design of informational computer systems. Automation and Remote Control (3), 12–19 (1990). (Translated from Russian)

[95] Sotskov, Y.: The complexity of shop-scheduling problems with two or three jobs. European Journal of Operational Research **V. 53**, 326–336 (1991)

[96] Sotskov, Y.: ϵ-approximate stability radius of Boolean linear form minimization problem. Institute of Engineering Cybernetics, Preprint No. 21, Minsk, Belarus (1991). (In Russian)

[97] Sotskov, Y.: A review of solution stability analysis in sequencing and scheduling. Institute of Engineering Cybernetics, Preprint No. 22, Minsk, Belarus (1991)

[98] Sotskov, Y.: Stability of an optimal schedule. European Journal of Operational Research **V. 55**, 91–102 (1991)

[99] Sotskov, Y.: The stability of the approximate Boolean minimization of a linear form. U.S.S.R. Computational Mathematics and Mathematical Physics **V. 33**(5), 699–707 (1993)

[100] Sotskov, Y.: Two, three, many or the complexity of scheduling with fixed number of jobs. In: A.D. Ulrich Derigs Achim Bachem (ed.) Operations Research Proceedings, pp. P. 168–172. Springer-Verlag Berlin, Heidelberg (1995)

[101] Sotskov, Y.: Software for production scheduling based on the mixed (multi)graph approach. Computing & Control Engineering Journal **V. 7**(5), 240–246 (1996)

[102] Sotskov, Y.: Mixed multigraph approach to scheduling jobs on machines of different types. Optimization **V. 42**, 245–280 (1997)

[103] Sotskov, Y.: Investigation of the optimal schedule stability. Informatics (4), 65–75 (2004). (In Russian)

[104] Sotskov, Y.: Recent results on stability analysis of an optimal assembly line balance. International Journal: Information, Theories & Applications **V. 14**(2), 174–181 (2007)

[105] Sotskov, Y., Allahverdi, A., Lai, T.C.: Flowshop scheduling problem to minimize total completion time with random and bounded processing times. Journal of the Operations Research Society **V. 55**, 277–286 (2004)

[106] Sotskov, Y., Alyushkevich, V.: Stability of an optimal orientation of mixed graph edges. Doklady Akademii Navuk BSSR **V. 32**(4), 108–111 (1988). (In Russian)

[107] Sotskov, Y., Dolgui, A., Portmann, M.C.: Stability analysis of optimal balance for assembly line with fixed cycle time. European Journal of Operational Research **V. 168**(3), 783–797 (2006)

[108] Sotskov, Y., Dolgui, A., Sotskova, N., Werner, F.: Stability of the optimal line balance for a fixed number of stations. In: A. Dolgui, J. Soldek, Z. O. (eds.) Proceedings of the 9th International Multi-Conference Advanced Computer Systems, Production System Design, Supply Chain Management and Logistics, ACS'02 - SCM, pp. 21–28. Miedzyzdroje, Poland (2002)

[109] Sotskov, Y., Dolgui, A., Sotskova, N., Werner, F.: Stability of optimal line balance with given station set. In: A. Fiat, G. Woeginger (eds.) Supply Chain Optimization.Applied Optimization, vol. V. 94, pp. 135–149. Springer, USA, New York (2005)

[110] Sotskov, Y., Egorova, N., Lai, T.C.: Minimizing total weighted flow time of a set of jobs with interval processing times. Mathematical and Computer Modelling **V. 50**, 556–573 (2009)

[111] Sotskov, Y., Egorova, N., Werner, F.: Minimizing total weighted completion time with uncertain data: A stability approach. Automation and Remote Control **V. 71**(10), 2038–2057 (2010)

[112] Sotskov, Y., Lai, T.C.: Minimizing total weighted flow under uncertainty using dominance and a stability box. Computers & Operations Research **V. 39**, 1271–1289 (2012)

[113] Sotskov, Y., Lai, T.C., Werner, F.: Measures of problem uncertainty for scheduling with interval processing times. Operations Research Spektrum **V. 39**, 1–31 (2012). (In Press: DOI 10 1007/s00291-012-0306-3)

[114] Sotskov, Y., Leontev, V., Gordeev, E.: Some concepts of stability analysis in combinatorial optimization. Discrete Applied Mathematics **V. 58**, 169–190 (1995)

[115] Sotskov, Y., Matsveichuk, N.: Measure of uncertainty for bellman-johnson problem with interval data. Cybernetics and System Analysis **V. 48**(5), 3–17 (2012)

[116] Sotskov, Y., Shakhlevich, N.: NP-hardness of shop-scheduling problems with three jobs. Discrete Applied Mathematics **V. 59**, 237–266 (1995)

[117] Sotskov, Y., Shilak, A.: Minimization of project-network with given bounds of activity durations. In: B. Fleischmann, R. Lasch, U. Derigs, W. Domschke, U. Rieder (eds.) Operations Research Proceedings 2000, pp. P. 384–387. Springer-Verlag Heidelberg, Drezden, Germany (2001)

[118] Sotskov, Y., Sotskova, N.: Stability of an optimal schedule for a job-shop problem with two jobs. International Journal: Information Theories & Applications **V. 10**(3), 321–326 (2003)

[119] Sotskov, Y., Sotskova, N.: Stability of an optimal schedule for a job-shop problem with two jobs. In: Proceedings of the X-th International Conference "Knowledge -Dialogue - Solution", pp. 502–507. Commerce Bulgaria Publishers, Varna, Bulgaria (2003)

[120] Sotskov, Y., Sotskova, N.: Stability of an optimal schedule for a job-shop problem with two jobs. International Journal: Information, Theories & Applications **V. 10**(3), 321–326 (2003)

[121] Sotskov, Y., Sotskova, N.: Scheduling Theory. Systems with Uncertain Numerical Parameters. National Academy of Sciences of Belarus, United Institute of Informatics Problems, Minsk, Belarus (2004). (In Russian)

[122] Sotskov, Y., Sotskova, N., Lai, T.C., Werner, F.: Scheduling under Uncertainty. Theory and Algorithms. Publishing House "Belorusskaya Nauka" National Academy of Sciences of Belarus, Minsk, Belarus (2010)

[123] Sotskov, Y., Sotskova, N., Werner, F.: Stability of an optimal schedule in a job shop. OMEGA - International Journal of Management Science **V. 25**(4), 397–414 (1997)

[124] Sotskov, Y., Tanaev, V.: Scheduling theory and practice: Minsk group results. Intelligent Systems Engineering **V. 1**, 1–8 (1994)

[125] Sotskov, Y., Tanaev, V., Werner, F.: Stability radius of an optimal schedule: A survey and recent developments. In: G. Yu (ed.) Industrial Applications of Combinatorial Optimization, vol. V. 16, pp. P. 72–108. Kluwer Academic Publishers, Boston, MA (1998)

[126] Sotskov, Y., Wagelmans, A., Werner, F.: On the calculation of the stability radius of an optimal or an approximate schedule. Tech. Rep. 9718/A, Econometric Institute, Erasmus University Rotterdam, The Netherlands, Rotterdam (1997)

[127] Sotskov, Y., Wagelmans, A., Werner, F.: On the calculation of the stability radius of an optimal or an approximate schedule. Annals of Operations Research **V. 83**, 213–252 (1998)

[128] Sotskova, N.: Optimization in a job shop with given bounds of processing times. In: J. Soldek, J. Pejas (eds.) Proceedings of the Sixth International Conference on Advanced Computer Systems ACS'99, pp. 462–467. Technical University of Szczecin, Szczecin - Poland (1999)

[129] Sotskova, N.: Solution of an uncertain job-shop problem with regular criterion. In: Book of Abstracts. Symposium on Operations Research SOR'99, p. 103. Otto-von-Guericke-Universität Magdeburg, Magdeburg - Germany (1999)

[130] Sotskova, N.: Opimal scheduling with unsertainty in the numerical data on the basis of a stability analysis. Ph.D. thesis, Faculty of Mathematic / Otto-von-Guericker-University, Magdeburg, Germany (2001). http://diglib.uni-magdeburg.de/Dissertationen/2001/nadsotskova.pdf

[131] Sotskova, N., Werner, F.: Dominance relations on the set of schedules for uncertain job-shop. In: J.M. Proth, V. Tanaev (eds.) Proceedings of the International Workshop on Discrete Optimization Methods an Scheduling and Computer-aided Design, pp. 99–101. Minsk, Belarus (2000)

[132] Strusevich, V.: Two machine flow shop scheduling problem with no wait in process: Controllable machine speeds. Discrete Applied Mathematics **V. 59**, 75–86 (1995)

[133] Sussmann, B.: Scheduling problems with interval disjunctions. Mathematical Methods of Operations Research **V. 16**, 165–178 (1972)

[134] Szwarc, W.: Solution of the Akers-Friedman scheduling problem. Operations Research **V. 8**, 782–788 (1960)

[135] Tanaev, V., Gordon, V., Shafransky, Y.: Scheduling theory. Single-Stage Systems. Kluwer Academic Publishers, Dordrecht, The Netherlands (1994)

[136] Tanaev, V., Sotskov, Y., Strusevich, V.: Scheduling Theory: Multi-Stage Systems. Kluwer Academic Publishers, Dordrecht, The Netherlands (1994)

[137] Trick, M.: Scheduling multiple variable-speed machines. Operations Research **V. 42**, 234–248 (1994)

[138] Tsujimura, Y., Gen, M., Kubota, E.: Solving fuzzy assembly-line balancing problem with genetic algorithms. Computers and Industrial Engineering **V. 29**(1-4), 543–547 (1995)

[139] Wu, S., Byeon, E.S., Storer, R.: A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. Operations Research **47**(1), 113–124 (1999)