Optimal Scheduling with Uncertainty in the Numerical Data on the Basis of a Stability Analysis

DISSERTATION

zur Erlangung des akademischen Grades

doctor rerum naturalium
(**Dr. rer. nat.**),

genehmigt durch die Fakultät für Mathematik
der Otto-von-Guericke-Universität Magdeburg

von Diplom-Wirtschaftsmathematikerin Nadezhda Sotskova

geb. am 21. Mai 1975
in Minsk, Belarus

Gutachter: Prof. Dr. Frank Werner
Prof. Dr. Ingo Althöfer
Prof. Dr. Chengbin Chu

Eingereicht am: 03.01.2001
Verteidigung am: 09.04.2001

No *time* like the present

To my parents

Acknowledgements

This research began as I was a student of the Faculty of Applied Mathematics at the Belarussian State University at Minsk and continued at the Institute of Mathematical Optimization of the University of Magdeburg.

I would like to thank the members of the Promotion Committee, who gave me the chance to work on and to complete my thesis in Germany, Prof. DSc. Michail Kovalev from the Belarussian State University and Prof. DSc. Mikalai Miatselski from the Institute of Mathematics of National Academy of Sciences of Belarus, who recommended me.

During the last three years, I wrote this dissertation under the supervision of Prof. Dr. Frank Werner. It is a great pleasure for me to express my gratitude to him for his kind agreement to manage my work, and the continuous support of my study and research, for his useful suggestions for improvements and for his help in the English and German languages. Moreover, I would like to gratefully acknowledge the other collaborators of the Department of Mathematics for their good attitude and for creating the *optimal* atmosphere for studies and research.

During the time that I worked on this thesis, I have received support of several people from the University of Magdeburg. Among them I would like to mention: Prof. Dr. Eberhard Girlich, who helped me with all my questions and requests; Dr. Udo Uebrick, who found time for *numerous* conversations with me; Dr. Burkhard Thiele, who shared his useful software expertise with me; Volker Lauff, who went with me hand-by-hand to the PhD *basis*; Krister Johnson for help in the English language; Vladimir Nalivaiko and Ivan Mihov, who helped me with my real life questions abroad. It is a great pleasure for me to cordially thank them here.

As a large part of my *schedule* in Germany, I had the opportunity to learn German, English and French. So, I am grateful to many teachers from the Language Center of the University of Magdeburg, who helped me to learn foreign languages.

A special thanks also goes to Viktoriya Kryzhanyuk for her moral support and her likemindedness of worldly questions. She always was with me in up minutes and in down ones, during the moments of *analysis* and frustrations, and small victories that are inherently connected with doing scientific research. Since the time we got to know each other in Germany mainly Viktoriya's company helps me to get over the loneliness.

I like to thank Prof. Dr. Ingo Althöfer from the University of Jena for his willingness to read my manuscript and for his useful suggestions for improvements.

I also acknowledge and thank Georgii Andreev from the Institute of Engineering Cybernetics of the National Academy of Sciences of Belarus for his qualified help in programming in Fortran and in *data* processing.

Furthermore, I thank my friend Andrei Tratsevskii for his *stable* understanding, love, faithfulness, support and patience.

Finally, it seems almost superfluous to mention the encouragement and understanding that I have always received from my mother and father. Their enthusiasm, faith and love were of great help for me during the period of separation and I have benefitted in many ways from their advices. I thank my father for his various ideas which have always been very stimulating for me and for a great deal of useful suggestions to improve both my style of writing papers and the layout of my thesis. I hope I meet up to their expectations and hopes.

Certainly, I would not have been able to write this thesis in the present form and *in the present time* without support of all people mentioned! You have all been a great support to me. Without your kind help and continuous assistance this thesis would have never been written the way it is.

Magdeburg, January 2001

Nadezhda Sotskova

Contents

List of Tables

List of Figures

Introduction

Scheduling is one of the common steps of decision-making which often plays a crucial role in manufacturing as well as in service industries. Scheduling is concerned with the allocation of given *machines* to given *jobs* over *time*. It is a decision-making process that has as a goal the optimization of a given objective. The machines (and jobs, respectively) may take different forms: Machines in a workshop (and operations in a production process), runways at an airport (and take-offs and landings at an airport), crews at the constructions site (and stages in a construction project), processing units in a computing environment (and executions of computer programs), teachers at the university (and student groups), and so on. A job may have a different priority, release time, due date, and processing time. Modern scheduling theory contains two main parts, based on deterministic and stochastic models.

Deterministic models are introduced for scheduling environments (see e.g. [All97, BDP96, LLRKS93, TSS94]) in which the processing time (duration) of each operation processed by a machine is supposed to be given in advance (before applying a scheduling procedure) and assumed to be a *constant* during the practical realization of a schedule. Unfortunately, exact information is not often known in advance, and difficulties arise when the given processing time of some operation may vary due to a change in a dynamic environment. Even if the processing times are given in advance, OR workers are forced to take into account errors within the practical realization of a schedule, the precision of the equipment for calculating the processing times, round-off errors in the calculation of a schedule on the computer, machine breakdowns, additionally arriving jobs with high priority and so on. The inadequacy of a deterministic scheduling problem in modelling real-world situations was emphasized in several publications (see e.g. [ML93, PL94, Pin95a]).

More general scheduling settings have been considered using a *stochastic* model (see [CCLe95, pp. 33-59], [Pin95a]), where the duration of an operation is assumed to be a *random variable* with a known probability distribution. However, in practice difficulties may still arise in some scenarios. First, we may not have enough prior information to characterize the probability distribution of a random duration. Second, even if the probability distribution of a random duration is known a priori, this distribution is useful only for a large number of realizations of similar scheduling environments but is of little practical sense for a unique realization or for a small number of similar realizations.

In this dissertation, a model of one of the more realistic scheduling scenarios is considered: It is assumed that in a practical realization of a schedule the processing time of an operation may take any value between the *lower* and *upper bounds* given before applying a scheduling procedure. Obviously, a deterministic model is a special case of the model considered, i.e. when given lower and upper bounds for each processing time are equal. The model considered can also be interpreted as a stochastic model under such 'strict uncertainty', when there is no sufficient a priori information about the probability distribution of a random duration, or more precisely, it is only known that the random duration will fall between given lower and upper bounds with probability one. In spite of obvious practical importance, the model under 'strict uncertainty' attracts a very limited attention in the OR literature so far.

Next, we introduce our model more formally. Let us 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 have to process a set of given jobs $J = \{J_1, J_2, \dots, J_n\}$. For a shop under consideration, the following three assumptions are fulfilled.

Assumption 1: *At any time each machine $M_k \in M$ can process not more than one job and each job $J_i \in J$ can be processed at most on one machine from the set M.*

The processing of a job on a machine is called an *operation*, and it is assumed that the processing of a job includes the execution of the given set of operations in the given order. The machine order for processing job $J_i \in J$ is called (*technological*) *route* of job J_i, and the distribution of all given operations Q to the machines M is fixed via the technological routes of the jobs J. If the routes may be given differently for different jobs, we have a **job shop**, otherwise we have its special case called a **flow shop**. In the latter case, each job has to be processed once on each machine while in the former case, both repetitions and absence of a machine in the route are allowed. In both cases each operation is assigned to a certain machine, and the route of job $J_i \in J$ defines linearly ordered operations (a sequence) $O_{i1}, O_{i2}, \dots, O_{in_i}$. For a flow shop, the equality $n_i = m$ holds for each job $J_i \in J$, while in the general case of the job shop, the value n_i may be smaller or larger than m or equal to m for different jobs $J_i \in J$.

The following assumption is also fulfilled for the shop considered in this dissertation.

Assumption 2: *Preemptions of an operation are forbidden.*

Assumption 2 means that 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} denotes the processing time of an operation O_{ij}.

In a *deterministic* model, the processing times p_{ij} are assumed to be known exactly for all operations $O_{ij},\ J_i \in J,\ j = 1, 2, \dots, n_i$, and a schedule may be defined as the start times s_{ij} (or completion times c_{ij}) of all operations Q provided that both Assumption 1 and Assumption 2 are fulfilled. Such a set of start (completion) times of operations Q defines m sequences of the corresponding operations from the set Q on the corresponding machines $M_i \in M,\ i = 1, 2, \dots, m$, and vice versa. The objective is to find such sequences of the set of operations Q on the machines $M_i \in M$ (i.e. to find such a *schedule*) for which the value of the given objective function $\Phi(C_1, C_2, \dots, C_n)$ is minimal. Hereafter, $C_i = c_{in_i}$ denotes the completion time of job $J_i \in J$. If the function $\Phi(C_1, C_2, \dots, C_n)$ is a non-decreasing one, such a criterion is called regular [LLRKS93]. The most popular regular criteria are the minimization of maximum flow time (makespan) $\mathcal{C}_{max} = \Phi(C_1, C_2, \dots, C_n) = \max\{C_i : J_i \in J\}$, and the minimization of mean flow time $\sum \mathcal{C}_i = \Phi(C_1, C_2, \dots, C_n) = \sum_{i=1}^{n} C_i$.

Scheduling problems are classified by a triplet $\alpha/\beta/\gamma$ (see [LLRKS93]). The α field describes the machine environment and usually contains a single entry. The β field provides details of the processing characteristics and may contain no entries, a single entry, or multiple entries. The γ field contains the objective function to be minimized and it usually contains a single entry. Using such a three-field notation, the deterministic job shop problems considered in Chapter 1 are denoted by $\mathcal{J}//\mathcal{C}_{max}$ and $\mathcal{J}//\sum\mathcal{C}_i$.

Along with a job shop, in which the technological routes are fixed for all jobs, the **open shop** is also considered in scheduling theory as a multi-stage processing system in which the actual route $O_{i1}, O_{i2}, \dots, O_{im}$ may be arbitrary for job $J_i \in J$, i.e. the route is not fixed a priori but has to be found in an optimal way by a decision-maker. In an open shop, each job has to be processed once on each machine (similarly to a flow shop). To indicate an open shop problem, the letter $\mathcal{O}$ is used instead of $\mathcal{J}$ or $\mathcal{F}$ in the first field of the three-field notation.

The job shop problem is a special case of the so-called **general shop** problem $\mathcal{G}//\mathcal{C}_{max}$, in which *arbitrary* precedence constraints may be given on the set of operations Q. For the general shop, it is not necessary to use a double subscript for the designation of the operations O_{ij} since the notion of a job may lose its sense for the general shop problem. Let $Q = \{1, 2, \dots, q\}$ denote the set of all given operations in the general shop and Q_k denote the set of all operations from set Q, $Q_k \subseteq Q$, which have to be processed on machine $M_k \in M$. If $i \in Q_k$, then the non-negative real value p_i denotes the processing time of operation i on machine $M_k \in M$. In this dissertation, along with common notations (see Table 4.11 at page 130) we use specific notations for the general shop (see Table 2.7 at page 53) and specific notations for the job shop (see Table 3.10 at page 94).

In Chapter 1, we survey known results on the calculation of the stability radius of an optimal schedule for general and job shops. The *stability radius* denotes the largest quantity of independent variations of the processing times of the operations such that the given schedule remains optimal. In the survey, the main attention is paid to the results on a stability analysis which are used further in this dissertation. Some other related results and approaches are briefly given in the last section of Chapter 1.

Chapter 2 deals with a mathematical model for scheduling scenarios in which the processing time of each operation $i \in Q$ is *uncertain* before applying a scheduling procedure and may take any value between a given *lower bound* $a_i \geq 0$ and an *upper bound* $b_i \geq a_i$. More precisely, in Chapter 2 we consider the general shop problem when the *structural input data* are fixed, while only a lower bound $a_i \geq 0$ and an upper bound $b_i \geq a_i$ for the processing time of operation $i \in Q$, are given as *numerical input data* before applying a scheduling procedure. In other words, the following assumption is fulfilled.

Assumption 3: *The actual processing time p_i of operation $i \in Q$ may take any real value between given lower and upper bounds, i.e.*

$$a_i \leq p_i \leq b_i, \ i \in Q. \tag{1}$$

It should be noted that while Assumption 1 and Assumption 2 are commonly used in scheduling theory, Assumption 3 is rather new for the OR literature. The main aim of this thesis is to introduce Assumption 3 in the settings of scheduling problems.

A general shop problem which satisfies Assumptions 1, 2 and 3 will be denoted by $\mathcal{G}/a_i \leq p_i \leq b_i/\Phi$. On the one hand, problem $\mathcal{G}/a_i \leq p_i \leq b_i/\Phi$ can be considered as a *stochastic* general shop problem under 'strict uncertainty', when there is no prior information about the probability distributions of the random processing times. On the other hand, if $a_i = b_i$ for each operation $i \in Q$, problem $\mathcal{G}/a_i \leq p_i \leq b_i/\Phi$ turns out to be a *deterministic* general shop problem $\mathcal{G}//\Phi$.

Problem $\mathcal{G}/a_i \leq p_i \leq b_i/\Phi$ seems to be rather realistic, at least, it is not restrictive: Even if there is no prior information on the possible perturbations of the processing times p_i, one can consider 0 as lower bound of p_i and a sufficiently large number (e.g. the planning horizon) as upper bound for p_i. Moreover, for a flow and open shop problem fixing the structural input data means only to fix the number n of jobs and the number m of machines. Consequently, any two flow (open) shop problems with the same number n of jobs and the same number m of machines, i.e. problems $\mathcal{F}m/n = k/\Phi$ (problems $\mathcal{O}m/n=k/\Phi$, respectively) may differ one from another only in their processing times.

Chapter 3 deals with the job shop problem $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$ of minimizing the sum of completion times of n jobs $J = \{J_1, J_2, \dots, J_n\}$ processed on m machines $M = \{M_1, M_2, \dots, M_m\}$ when only the technological routes of the jobs are known before applying a scheduling procedure, while the processing times are uncertain.

Chapter 4 is devoted to some computational results of the calculation of the stability radii of optimal schedules for randomly generated job shop problems, when the objective is to minimize mean or maximum flow times. We test algorithms coded in Fortran-77 for an *a posteriori* analysis, in which an optimal schedule has already been constructed and the question is to determine such changes in the processing times of operations, which do not destroy the optimality of the schedule at hand. We present also computational results for solving randomly generated problems $\mathcal{J}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ and $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$.

In Conclusions, we summarize the results obtained and outline some topics for future research.

Each chapter is written mainly as an independent one from the others. To this end, a short abstract and the main notations are given at the beginning of each chapter, and a summary and some remarks are given at the end of chapter. The independence of the chapters implies some repetitions in definitions, notations and argumentation.

The titles of the papers and abstracts, where the results of this dissertation were published, are printed in bold face in the bibliography.

Chapter 1

Stability Analysis in Scheduling Theory: A Survey

The usual assumption that the processing times of the operations are known before scheduling restricts practical aspects of the modern scheduling theory since it is often not valid for real-world processes. The main part of this chapter (i.e. Sections 1.1 - 1.4) is devoted to a survey of the known results for the stability analysis of an optimal schedule. The term 'stability analysis' is used for the phase of an algorithm at which a solution (or solutions) of an optimization problem has (have) already been found, and additional calculations are performed in order to investigate how this solution depends on the numerical input data. In this chapter, we survey the known results on job shop and general shop problems for the calculation of the stability radius, when the objective is to minimize mean or maximum flow time. The extreme values of the stability radius are considered in more detail.

1.1 Mixed Graph Models for General and Job Shops

The results from [BSW96, KSW95, Sot91, STW98] on the stability analysis may be considered as an investigation of scheduling problems *under conditions of uncertainty*, when the aim is to study the influence of round-off errors of the processing times on the property of a schedule to be optimal. The main reason for performing a stability analysis is that in most practical cases the processing times of the operations are inexact or uncertain before applying a scheduling procedure. In such cases a stability analysis is necessary to investigate the credibility of an optimal schedule at hand. On the one hand, if possible errors of the processing times are larger than the stability radius of an optimal schedule, this schedule may not be the best in a practical realization and there is not much sense in 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 each real change of the processing time is less than or equal to the stability radius of an optimal schedule: An a priori constructed optimal schedule will remain optimal (the best) in the practical realization as well. 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 (the number of machines, the technological routes, the range of variations of the processing times and so on) do not change quickly, and it is possible to use previous computations 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. However, 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 part of this information for solving further similar scheduling problems.

Different scheduling problems may be represented as extremal problems on disjunctive graphs (see e.g. [LLRKS93, RS64, Sus72, TSS94]). As it was mentioned in [CCLe95, pp. 277-293], the disjunctive graph approach is the most suitable one for traditionally difficult scheduling problems. In Section 1.1, we describe the disjunctive graph to represent the input data of the **general shop** problem $\mathcal{G}//\Phi$. A small example of a job shop problem illustrates the disjunctive graph approach. In Section 1.2, we survey some results concerning the calculation of the stability radius of an optimal schedule for problem $\mathcal{G}//\Phi$ via the reduction to a non-linear mathematical programming problem. The calculation of the stability radius along with characterizations of its extreme values for problems $\mathcal{G}//\mathcal{C}_{max}$ and $\mathcal{G}//\sum\mathcal{C}_i$ are surveyed in Sections 1.3 and 1.4, respectively. In Section 1.5, we survey related approaches to the stability analysis in combinatorial optimization.

First, we consider a general shop problem in which the given set of partially ordered operations $Q = \{1, 2, \ldots, q\}$ has to be processed by a given set of machines $M = \{M_1, M_2, \ldots, M_m\}$. We assume that each operation is assigned exactly to one machine, and at any time each machine can process at most one operation (see Assumption 1). Let p_j denote the processing time (duration) of operation $j \in Q$ and c_j denote the completion time of an operation j. Preemptions of operations are not allowed (see Assumption 2): If an operation i starts at time s_i, its processing is not interrupted until operation i is completed (up to time $c_i = s_i + p_i$). This problem is denoted by $\mathcal{G}//\Phi$.

The set of operations Q is supposed to be partially ordered by the given *precedence constraints* $\rightarrow$, which are defined as follows. Given two operations $i \in Q$ and $j \in Q$, we assume that the notation $i \rightarrow j$ means that operation i is a predecessor of operation j, i.e. if $i \rightarrow j$, then the inequality

$$c_i + p_j \leq c_j \tag{1.1}$$

holds for any feasible schedule.

Given that $\{Q_k : \ k = 1, 2, \ldots, m\}$ is a partition of the set Q, i.e.

$$Q = \cup_{k=1}^{m} Q_k; \ Q_k \neq \emptyset \ \text{ and } \ Q_k \cap Q_l = \emptyset; \ \text{ if } \ k \neq l; \ k = 1, 2, \ldots, m; \ l = 1, 2, \ldots, m,$$

we have *capacity constraints*. Since at any time machine $M_k \in M$ can process at most one operation (see Assumption 1), the inclusions $i \in Q_k$ and $j \in Q_k$ imply one of the following inequalities:

$$c_i + p_j \leq c_j \quad \text{ or } \quad c_j + p_i \leq c_i. \tag{1.2}$$

For the **job shop** problem $\mathcal{J}//\Phi$, along with the above partition the set of operations Q is also partitioned into n chains

$$Q = \bigcup_{i=1}^{n} Q^{(i)}; \ Q^{(i)} \neq \emptyset \ \text{ and } \ Q^{(i)} \cap Q^{(j)} = \emptyset; \ \text{ if } \ i \neq j; \ i = 1, 2, \ldots, n; \ j = 1, 2, \ldots, n,$$

where each chain includes the set $Q^{(i)}$ of all operations of a job J_i, $1 \leq i \leq n$, and this chain represents the technological route of job J_i. Note that for the job shop and flow shop problems the sets $Q^{(i)}$ are a priori known.

For problem $\mathcal{G}//\Phi$, the processing time p_i of each operation $i \in Q$ is known, and therefore a schedule of the operations Q on the machines M may be defined by the completion times c_i or by the start times $s_i = c_i - p_i$ of the operations $i \in Q$.

If the operation processing times are not known, it is not possible to define s_i and c_i for all operations $i \in Q$. Therefore, in the general case of problem $\mathcal{G}/a_i \le p_i \le b_i/\Phi$, the goal is to determine a processing sequence of 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 given precedence constraints (1.1) and capacity constraints (1.2) may be considered as a *schedule* for problem $\mathcal{G}/a_i \le p_i \le b_i/\Phi$. The general shop problem is to find such a schedule, which minimizes the value of the given non-decreasing objective function $\Phi(c_1, c_2, \dots, c_q)$.

A mixed (or *disjunctive*) graph is often introduced to model a deterministic scheduling problem (see [BDP96, Pin95b, RS64, TSS94]). We follow this approach and 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 (1.1) are represented by the set of directed (*conjunctive*) non-transitive arcs A: If operation i has to be processed before operation j starts and there is no other path from i to j, the arc (i, j) has to be included into the set A:

$$A = \{(i,j) : i \to j;\ i \in Q, j \in Q;$$
$$\text{there is no operation } k \in Q \text{ such that } i \to k \text{ and } k \to j \text{ simultaneously hold}\},$$

- the capacity constraints (1.2) are represented by the set E of undirected edges (pairs of *disjunctive arcs*) connecting operations, which have to be processed on the same machine:

$$E = \{[i,j] :\ i \in Q_k,\ j \in Q_k;\quad k = 1, 2, \dots, m;\quad i \not\to j;\ j \not\to i,$$
$$\text{i.e. neither } i \to j \text{ nor } j \to i \text{ holds } \}.$$

For a deterministic setting, the processing times p_i of all operations $i \in Q$ are known before scheduling and we associate a non-negative weight p_i with each vertex $i \in Q$ in $G = (Q, A, E)$ to obtain the *weighted* mixed graph denoted by $G(p) = (Q(p), A, E)$, which represents both the structural and numerical input data.

While solving the scheduling problem, each edge $[i, j] \in E$ has to be oriented. Indeed, for each pair of operations i and j, for which the edge $[i, j]$ belongs to the set E, there exist two possibilities: To complete operation $i \in Q_k$ before operation $j \in Q_k$ starts on their common machine $M_k \in M$ and to provide the first inequality from (1.2) (in this case edge $[i, j]$ is replaced by the arc (i, j)), or to complete operation j before operation i starts and to provide the second inequality from (1.2) (in this case edge $[i, j]$ is replaced by the arc (j, i)) (see [LLRKS93, Sus72, TSS94]). Let $E^* = \cup_{[i,j] \in E}\{(i, j), (j, i)\}$. The term 'disjunctive graph' is associated with the selection of one of these two possibilities for each pair of arcs $\{(i, j), (j, i)\} \subseteq E^*$. It means that one of these arcs must be added to a subset $E_s \subset E^*$ of *chosen arcs* and the other one must be rejected from the mixed (disjunctive) graph:

(*) $(i, j) \in E_s$ if and only if $(j, i) \in E^* \backslash E_s$.

Not each of such subsets E_s may be feasible for the scheduling procedure since E_s may cause a contradiction. A *feasible* schedule s is defined by a subset $E_s \subset E^*$ such that along with the above condition (*) the following condition is satisfied:

(**) the digraph $G_s = (Q, A \cup E_s, \emptyset)$ has no circuits.

In what follows, the adjective 'feasible' is often omitted before 'schedule'. Let $G_s = (Q, A \cup E_s, \emptyset)$ denote the digraph generated from the mixed graph G by orienting all edges of the set E. Digraph G_s is called *feasible* if and only if G_s contains no circuits. Let $\Lambda(G) = \{G_1, G_2, \ldots, G_\lambda\}$ be the set of all feasible digraphs G_s, i.e. digraphs which satisfy both conditions (*) and (**).

Since the objective function $\Phi(c_1, c_2, \ldots, c_q)$ is non-decreasing, one may consider only *semiactive* schedules: A schedule is called semiactive if no operation can start earlier without delaying the processing of some other operation and/or without violating the sequence of operations on some machine [LLRKS93, TSS94]. In the following, we consider only the set S of semiactive schedules. For any non-decreasing objective function, an optimal semiactive schedule exists, and there exists a one-to-one correspondence between all semiactive schedules $S = \{1, 2, \ldots, \lambda\}$ and all digraphs $\Lambda(G) = \{G_1, G_2, \ldots, G_\lambda\}$ generated from the mixed graph G: *Each feasible digraph $G_s \in \Lambda(G)$ uniquely defines a feasible schedule $s \in S$, and vice versa.*

On the one hand, given a vector $p = (p_1, p_2, \ldots, p_q)$ of processing times, a feasible digraph $G_s \in \Lambda(G)$ corresponding to $G_s(p) = (Q(p), A \cup E_s, \emptyset)$ uniquely defines the *earliest completion time* $c_i(s)$ of each operation $i \in Q$ and a unique semiactive schedule

$$s = (c_1(s), c_2(s), \ldots, c_q(s)).$$

On the other hand, each semiactive schedule $s \in S$ defines a unique digraph $G_s(p) \in \Lambda(G)$. In the following, we call the digraph $G_s(p)$ *optimal* if and only if schedule $s \in S$ is *optimal*.

Hereafter we often use an optimal digraph G_s instead of an optimal schedule s since digraph $G_s \in \Lambda(G)$ uniquely defines a set of m optimal sequences, and vice versa. The start times and completion times of the operations, 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 *longest path calculations* (see e.g. [TSS94, p. 285]). Given a fixed vector $p = (p_1, p_2, \ldots, p_q)$ of processing times, in order to construct an optimal schedule, one may enumerate (explicitly or implicitly) all feasible digraphs $G_1(p), G_2(p), \ldots, G_\lambda(p)$ generated by orienting all edges of the mixed graph G and selecting an optimal digraph, i.e. a feasible digraph with minimal value of the objective function. Unfortunately, the number λ of such feasible digraphs (the number of semiactive schedules) grows exponentially in the edge number $|E|$, and an overall enumeration of feasible digraphs is practically impossible for large numbers of jobs and machines. Nevertheless, for our computational experiments presented in Chapter 4, we use an explicit enumeration of feasible digraphs for small job shop problems in order to calculate the stability radii for all optimal schedules.

Although problem $\mathcal{G}//\Phi$ is NP-hard in the strong sense for any given regular criterion Φ considered in scheduling theory [LLRKS93, TSS94], the running time of calculating an optimal schedule $s = (c_1(s), c_2(s), \ldots, c_q(s))$ may be restricted by an $O(q^2)$-algorithm (see [[TSS94], p. 285]) after having constructed an optimal digraph $G_s(p)$. Thus, the main difficulty of problem $\mathcal{G}//\Phi$ is to construct an optimal digraph $G_s = (Q, A \cup E_s, \emptyset)$, i.e. to define the best set E_s of arcs generated by orienting the edges of the set E.

Due to the particular importance of the set E_s, it is called the *signature* of a schedule s [BSW96, Sot91, SLG95, STW98, Sus72]. Each feasible digraph $G_s = (Q, A \cup E_s, \emptyset)$ is uniquely defined by its signature, i.e. by the set of arcs E_s which replace the set of edges E.

As it was noted in [BDP96], the disjunctive graph model "has mostly replaced the solution representation by Gantt charts as described in [Gan19]". Next, we give additional comments to elaborate this kind of preference. First, while a Gantt chart is useful for the graphical presentation of a particular solution, the mixed graph model is suitable for the whole scheduling process from the initial mixed graph $G(p)$ (representing the input data) until a final digraph G_s (representing a solution $s \in S$) has been found. Second, a Gantt chart is a representation of one particular situation when there are no changes both in the a priori known processing times and in the calculated start times. However, such a situation is 'ideal' (at least, it occurs rather seldom). Thus, a Gantt chart seems to be more appropriate 'after realization' of the process (when all processing times, start times and completion times are known) while 'before realization' a mixed graph $G(p)$ and a digraph G_s seem to be more useful, since they are stable with respect to possible changes of the above 'times'. Third, while a Gantt chart is simply a picture in the plane, a digraph is a mathematical (i.e. abstract) object and can assume different graphical presentations. In particular, one can view a Gantt chart as a diagram of the weighted digraph $G_s(p)$ in the plane.

Next, we show how a mixed graph G may be introduced in the case of a **job shop** problem. Note that for a more convenient notation for the job shop, we use a double subscript designated to operations. To present the structural input data for problem $\mathcal{J}//\Phi$, we use the following mixed graph $G = (Q, A, E)$, where

- $Q = \{O_{ij} : J_i \in J,\ j = 1, 2, \ldots, n_i\}$,
- $A = \{(O_{ij}, O_{i,j+1}) : J_i \in J,\ j = 1, 2, \ldots, n_i - 1\}$,
- $E = \{[O_{ij}, O_{uv}] : O_{ij} \in Q_k, O_{uv} \in Q_k\ J_i \neq J_u\}$.

The set of arcs A defines *precedence constraints* (i.e. technological routes) as follows. Since each job may be processed on at most one machine at a time (see Assumption 1), operation $O_{ij} \in Q^{(i)}$ has to be completed before operation $O_{i,j+1} \in Q^{(i)}$ starts: $c_{ij} \leq s_{i,j+1}$. The route of job $J_i \in J$ defines linearly ordered operations (a sequence) $\{O_{i1}, O_{i2}, \ldots, O_{in_i}\} = Q^{(i)}$. At the stage $j \in \{1, 2, \ldots, n_i\}$ of the technological route of job J_i, operation $O_{ij} \in Q_k$ has to be processed on machine $M_k \in M$. The set of edges E defines *capacity constraints* as follows. Since any machine $M_k \in M$ can process at most one operation at a time (see Assumption 1) and operation preemptions are not allowed (see Assumption 2), operation $O_{ij} \in Q_k$ has to precede operation $O_{uv} \in Q_k$ or vice versa: $c_{ij} \leq s_{uv}$ or $c_{uv} \leq s_{ij}$.

Since a job shop is a special case of a general shop, we can use the notations of a general shop for the job shop, too. Using general shop notations, we can assume that the first job J_1 has the operations $\{1, 2, \ldots, n_1\} = Q^{(1)}$, the second job J_2 has the operations $\{n_1+1, n_1+2, \ldots, n_1+n_2\} = Q^{(2)}$, and so on, the last job J_n has the operations $\{\sum_{j=1}^{n-1} n_j + 1, \sum_{j=1}^{n-1} n_j + 2, \ldots, \sum_{j=1}^{n} n_j = q\} = Q^{(n)}$. Let us consider a small example to demonstrate most of the above notations.

Example 1.1 Figure 1.1 shows an example of a weighted mixed graph $G(p)$ for a job shop problem $\mathcal{J}4/n = 3/\Phi$ with three jobs and four machines. Also, for a job shop we can use the notations of operations with double subscript. In Figure 1.1 the set of all operations Q is $\{O_{11}, O_{12}, \ldots, O_{33}\}$, job J_1 consists of operations O_{11}, O_{12} and O_{13}, job J_2 of operations O_{21} and O_{22}, job J_3 of operations O_{31}, O_{32} and O_{33}. Machine M_1 has to process operations O_{11} and O_{33}, machine M_2 operations O_{12} and O_{32}, machine M_3 operations O_{13} and O_{22}, and machine M_4 operations O_{21} and O_{31}.

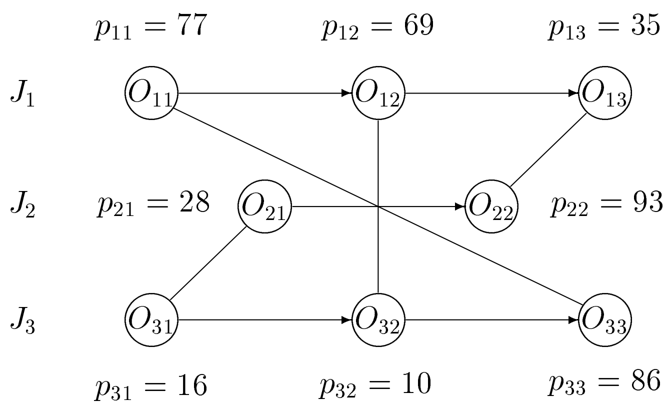


Figure 1.1: *Weighted mixed graph $G(p) = (Q(p), A, E)$ for problem $\mathcal{J}4/n = 3/\Phi$*

The goal of this problem $\mathcal{J}4/n = 3/\Phi$ is to sequence the four sets of operations $Q_1 = \{O_{11}, O_{33}\}, Q_2 = \{O_{12}, O_{32}\}, Q_3 = \{O_{13}, O_{22}\}$ and $Q_4 = \{O_{21}, O_{31}\}$. There are $2^4 = 16$ possible digraphs which can be generated from the mixed graph G, and 12 of them are feasible. The maximal weight of a path in the digraph $G_s(p)$ (called *critical weight*) defines the makespan, $\mathcal{C}_{max} = \max\{c_{n_i}(s) : J_i \in J\}$, of schedule $s \in S$. The path in $G_s(p)$ with a critical weight is called a *critical path.*

It is easy to see that there are two optimal digraphs for criterion $\mathcal{C}_{max}$ with the critical weight $\mathcal{C}_{max} = 181$, and only one optimal digraph for the mean flow time criterion with the value $\sum \mathcal{C}_i = \sum_{J_i \in J} c_{n_i}(s) = 465$. The digraph G_1, represented in Figure 1.2, is optimal for both criteria $\mathcal{C}_{max}$ and $\sum \mathcal{C}_i$, while digraph $G_2 = (Q, A \cup E_2, \emptyset)$ with the signature $E_2 = \{(O_{11}, O_{33}), (O_{22}, O_{13}), (O_{31}, O_{21}), (O_{32}, O_{12})\}$ is optimal for criterion $\mathcal{C}_{max}$, but is not optimal for criterion $\sum \mathcal{C}_i$.

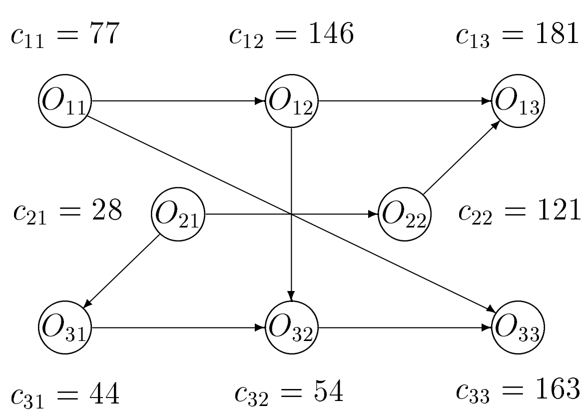


Figure 1.2: *Digraph $G_1 = (Q, A \cup E_1, \emptyset)$ which is optimal for both criteria $\mathcal{C}_{max}$ and $\sum \mathcal{C}_i$*

If operation durations are not known exactly before applying a scheduling procedure, it is not enough to construct only an optimal digraph G_s. It is also important to analyze the question of how much the durations of the operations may vary so that the digraph G_s remains optimal. In the following sections of this chapter, we survey known 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. Section 1.2 contains a formal definition of the stability radius, which is the maximal value of the radius of such a stability ball (see [Sot91]). Example 1.1 is used to demonstrate the notations and results.

1.2 Stability Radius

In Sections 1.2 - 1.4, the main question is as follows. How can one vary the processing times $p_i, i \in Q$, such that a given schedule $s \in S$ for problem $\mathcal{G}//\Phi$ remains optimal and how can one calculate the largest quantity of such variations of the processing times?

Note that any variation $p_i \pm \epsilon,\ \epsilon > 0$, of a processing time p_i changes at least the completion time $c_i(s)$ of operation i in an optimal schedule $s = (c_1(s), \dots, c_i(s), \dots, c_q(s))$ and, as a result, we obtain another schedule: $(\dots, c_i(s) + \epsilon, \dots)$ or $(\dots, c_i(s) - \epsilon, \dots)$. However, the optimal digraph $G_s(p) = (Q(p), A \cup E_s, \emptyset)$ for the new problem obtained due to such a variation of p_i remains the same if ϵ is sufficiently small.

The results of [BSW96, KSW95, Sot91, SSW97] were devoted to the stability of an optimal digraph $G_s(p)$ which represents an optimal solution of problem $\mathcal{G}//\Phi$. The above question may be concretized as follows. Under which largest independent changes in the components of the vector $p = (p_1, p_2, \dots, p_q)$, remains digraph $G_s(p)$ optimal? Next, we introduce these notions in a formal way (see [Sot91]).

Let R^q be the space of all q-dimensional real vectors p with the Chebyshev (maximum) metric: The distance $d(p, p')$ between the vectors $p \in R^q$ and $p' = (p'_1, p'_2, \dots, p'_q) \in R^q$ is defined as follows:

$$d(p, p') = \max_{i \in Q} |p_i - p'_i|,$$

where $|p_i - p'_i|$ denotes the absolute value of the difference $p_i - p'_i$. Let R^q_+ be the space of all q-dimensional non-negative real vectors:

$$R^q_+ = \{x = (x_1, x_2, \dots, x_q) :\ x_i \geq 0,\ i \in Q\}.$$

Let schedule $s \in S$ be optimal for the non-negative real vector $p \in R^q_+ \subset R^q$ of the processing times.

Definition 1.1 *The 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 called a stability ball of the schedule $s \in S$ (of digraph $G_s \in \Lambda(G)$) if for any vector $p' \in O_\varrho(p) \cap R^q_+$ of the processing times schedule s (digraph $G_s(p')$) remains optimal. The maximum value $\varrho_s(p)$ of the radius ϱ of a stability ball $O_\varrho(p)$ of the schedule s (of digraph G_s) is called the stability radius of the schedule s (of digraph G_s), where*

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

We denote the stability radius by $\varrho_s(p)$ for an arbitrarily given regular criterion. For criterion $\mathcal{C}_{max}$, the stability radius is denoted by $\widehat{\varrho}_s(p)$, and for criterion $\sum \mathcal{C}_i$ by $\overline{\varrho}_s(p)$.

In what follows, we use whenever appropriate the notion "stability radius of the optimal digraph $G_s \in \Lambda(G)$" instead of "stability radius of the optimal schedule $s \in S$".

Due to the maximum metric, the set $O_\varrho(p) \cap R^q_+$ is a polytope for any positive $\varrho \in R^1_+$. Formulas for calculating the stability radius for the makespan criterion and the characterization of the extreme values of $\widehat{\varrho}_s(p)$ have been proven in [Sot91, TSS94]. The same questions for the mean flow time criterion have been considered in [BSW96].

Example 1.1 (continued). Returning to the Example 1.1 presented in Figure 1.1, one can calculate the stability radii: $\widehat{\varrho}_1(p) = 5.75$, $\widehat{\varrho}_2(p) = 1.8$ and $\overline{\varrho}_1(p) = 1.17$. In particular, the equality $\widehat{\varrho}_1(p) = 5.75$ means that digraph $G_1(p)$ remains optimal for criterion $\mathcal{C}_{max}$ if no processing time changes its value by more than 5.75. On the other hand, there exist such changes $p_i \pm (5.75 + \epsilon)$ of the processing times $p_i, i \in Q$, for which $G_1(p)$ is no longer optimal and this statement is valid for any small positive real ϵ. Obviously, if we have both optimal schedules (the first defined by digraph $G_1(p)$ and the second defined by digraph $G_2(p)$), the first one is preferable for practice since its stability radius is larger.

Definition 1.1 implies a general approach for calculating $\varrho_s(p)$, which is discussed in the rest of this section and which is concretized for $\Phi = \mathcal{C}_{max}$ and for $\Phi = \sum \mathcal{C}_i$ in Section 1.3 and in Section 1.4, respectively. In [Sot91], the calculation of $\varrho_s(p)$ has been reduced to a non-linear programming problem. Next, we give the presentation from [Sot91, STW98] for the case of the general shop problem $\mathcal{G}//\Phi$ when the set of all operations Q is partitioned into n technological routes $Q^{(i)}$ of a job $J_i, i \in \{1, 2, \dots, n\}$.

Let $[\mu]$ denote the set of vertices which form a path μ in the digraph G_k and $l^p(\mu)$ be the weight of this path:

$$l^p(\mu) = \sum_{i \in [\mu]} p_i.$$

Let $\tilde{H}^i_k$ denote the set of all paths in the digraph $G_k = (Q, A \cup E_k, \emptyset)$ ending in the vertex j_i, where operation $j_i \in Q^{(i)} \subseteq Q$ is the last operation of job J_i, $1 \le i \le n$. The value of $c_{j_i}(k)$ for a schedule $k \in S$ is equal to the largest weight of a path in $\tilde{H}^i_k$.

Given a digraph $G_k(p)$, $c_i(k)$ is equal to the maximum weight among all paths in $\tilde{H}_k$ ending in vertex $i \in Q$. While calculating $c_i(k), i \in Q$, we can consider only a subset of $\tilde{H}_k$ due to the following binary relation.

Definition 1.2 *The path ν dominates path μ if the set $[\mu]$ is a proper subset of the set $[\nu]$.*

The above dominance relation is a *strict order* binary relation, where *transitivity* and *antireflexivity* hold.

Let H^i_k denote the set of all dominant paths in $\tilde{H}^i_k$. Since

$$c_{j_i}(k) = \max_{j \in Q^{(i)}} c_j(k) = \max_{\mu \in H^i_k} l^p(\mu),$$

the schedule $s = (c_1(s), c_2(s), \dots, c_{j_i}(s)) \in S$ is optimal if

$$\begin{aligned} \Phi^p_s &= \Phi(\max_{\mu \in H^1_s} l^p(\mu), \max_{\mu \in H^2_s} l^p(\mu), \dots, \max_{\mu \in H^n_s} l^p(\mu)) = \\ &= \min_{k=1,2,\dots,\lambda} \Phi(\max_{\nu \in H^1_k} l^p(\nu), \max_{\nu \in H^2_k} l^p(\nu), \dots, \max_{\nu \in H^n_k} l^p(\nu)) = \min_{k=1,2,\dots,\lambda} \Phi^p_k. \end{aligned} \tag{1.3}$$

Let $S^\Phi(p) \subseteq S$ denote the set of all optimal semiactive schedules for the vector p of the processing times with respect to criterion Φ and let $s \in S^\Phi(p)$. From Definition 1.1 it follows that

$$\varrho_s(p) = \inf\{d(p, x) : x \in R^q_+, s \notin S^\Phi(x)\}. \tag{1.4}$$

In order to calculate $\varrho_s(p)$, it is sufficient to know the optimal value of the objective function $f(x_1, x_2, \ldots, x_q)$ of the following non-linear programming problem (see [Sot91]):

$$\text{Minimize } f(x_1, x_2, \ldots, x_q) = \max_{i=1,2,\ldots,q} |x_i - p_i| \tag{1.5}$$

subject to

$$\begin{aligned} \Phi_s^x &= \Phi(\max_{\mu \in H_s^1} l^x(\mu), \max_{\mu \in H_s^2} l^x(\mu), \ldots, \max_{\mu \in H_s^n} l^x(\mu)) > \\ &> \min_{\substack{k=1,2,\ldots,\lambda,\\ k\neq s}} \Phi(\max_{\nu \in H_k^1} l^x(\nu), \max_{\nu \in H_k^2} l^x(\nu), \ldots, \max_{\nu \in H_k^n} l^x(\nu)) = \min_{\substack{k=1,2,\ldots,\lambda,\\ k\neq s}} \Phi_k^x, \end{aligned} \tag{1.6}$$

$$x_i \geq 0, \quad i = 1, 2, \ldots, q. \tag{1.7}$$

If condition (1.6) is not satisfied for any vector $x \in R_+^q$, then digraph $G_s(p)$ is optimal for all vectors $x \in R_+^q$ of the processing times:

$$\begin{cases} \Phi_s^x \leq \min\{\Phi_k^x \, : \, k = 1, 2, \ldots, \lambda; k \neq s\}, \\ x_i \geq 0, i = 1, 2, \ldots, q. \end{cases}$$

In this case the stability radius is *infinitely large*:

$$\varrho_s(p) = \infty.$$

In all other cases, there exists an optimal value f^* of the objective function of problem (1.5)-(1.7):

$$f^* = \inf \max_{i=1,2,\ldots,q} |x_i - p_i|,$$

where the infimum is taken over all vectors x satisfying conditions (1.6) and (1.7). To find the value f^*, it is sufficient to know a solution $x^0 = (x_1^0, x_2^0, \ldots, x_q^0)$ of problem (1.5)-(1.7), where the sign $>$ in inequality (1.6) is replaced by the sign $\geq$:

$$\text{Minimize} \quad f(x_1, x_2, \ldots, x_q) = \max_{i=1,2,\ldots,q} |x_i - p_i|,$$

$$\text{subject to} \quad \begin{cases} \Phi_s^x \geq \min_{k=1,2,\ldots,\lambda; k\neq s} \Phi_k^x, \\ x_i \geq 0, \; i = 1, 2, \ldots, q. \end{cases}$$

Thus,

$$f^* = \max_{i=1,2,\ldots,q} |x_i^0 - p_i| = d(x^0, p) = \varrho_s(p)$$

and for any small $\epsilon > 0$, there exists a vector $x^\epsilon = (x_1^\epsilon, x_2^\epsilon, \ldots, x_q^\epsilon) \in R_+^q$ such that $d(x^\epsilon, p) = \varrho_s(p) + \epsilon$ and $s \notin S^\Phi(x^\epsilon)$. It may occur that the above solution x^0 of the non-linear programming problem is equal to p. In this case equalities

$$\varrho_s(p) = d(p, p) = 0$$

hold and it means that the optimal 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 $d(p, p') = \epsilon$.

1.3 Maximum Flow Time

In [Sot91, TSS94], the stability radius for criterion $\mathcal{C}_{max}$ has been considered and here we survey these results. Let for the general shop problem, H and H_k denote the set of all dominant paths in digraph $(Q, A, \emptyset)$ and the set of all dominant paths in digraph $G_k \in \Lambda(G)$, respectively (see Definition 1.2). Thus,

$$H_k = \{\nu \in \tilde{H}_k : \text{Inclusion } [\nu] \subset [\mu] \text{ does not hold for any path } \mu \in \tilde{H}_k\},$$

where $\tilde{H}_k$ ($\tilde{H}$, respectively) is the set of all paths in digraph $G_k(p) \in \Lambda(G)$ (in digraph $(Q, A, \emptyset)$) for the general shop.

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 the critical path in the weighted digraph $G_s(p)$. The equality (1.3) for problem $\mathcal{G}//\mathcal{C}_{max}$ is converted to the following one:

$$\max_{\mu \in H_s} l^p(\mu) = \min_{k=1,2,\dots,\lambda} \max_{\nu \in H_k} l^p(\nu). \tag{1.8}$$

Let $H_k(p)$ denote the set of all critical dominant paths in the digraph $G_k(p) \in \Lambda(G)$ (with respect to the vector p). Obviously, we have $H_k(p) \subseteq H_k$. Next, we present necessary and sufficient conditions for equality $\widehat{\varrho}_s(p) = 0$ proven in [Sot91].

Theorem 1.1 *For an optimal schedule $s \in S$ of problem $\mathcal{G}//\mathcal{C}_{max}$, 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 does not exist any path $\nu^* \in H_k(p)$ with $[\mu^*] \subseteq [\nu^*]$.*

From Theorem 1.1, the following two corollaries are obtained (see [Sot91]).

Corollary 1.1 *If s is an optimal schedule for problem $\mathcal{G}//\mathcal{C}_{max}$ and $H_s(p) \subseteq H$, then $\widehat{\varrho}_s(p) > 0$.*

Corollary 1.2 *If s is a unique optimal schedule for problem $\mathcal{G}//\mathcal{C}_{max}$, then $\widehat{\varrho}_s(p) > 0$.*

In [Sot91], the following characterization of an infinitely large stability radius was proven.

Theorem 1.2 *For an optimal schedule $s \in S$ of 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 $[\mu] \subseteq [\nu]$.*

The following corollary gives a simple upper bound for the stability radius $\widehat{\varrho}_s(p)$.

Corollary 1.3 *If $\widehat{\varrho}_s(p) < \infty$, then $\widehat{\varrho}_s(p) \leq \max_{i=1}^{q} p_i$.*

Due to Theorem 1.2, one can identify a problem whose optimal schedule is implied only by the given precedence constraints and by the given distribution of the operations to the machines, but independent from the processing times of the operations. However, it is difficult to check the conditions of Theorem 1.2.

In [KSW95], it has been shown that for 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 X_k (Y_k, respectively) be the set of all operations $i \in Q$ such that $i \rightarrow j$ ($j \rightarrow i$) and $j \in Q_k$, $i \notin Q_k$. For a set X of operations, let $n(X)$ be the number of jobs having at least one operation in X.

Theorem 1.3 *For problem $\mathcal{J}//\mathcal{C}_{max}$, there exists an optimal digraph $G_s(p)$ with an infinitely large stability radius if and only if the following two conditions hold:*
1) inequality $\max\{|X_k|, |Y_k|\} \leq 1$ holds for any machine M_k with $n(Q_k) > 1$ and
2) if there exist two operations $g \in X_k$ and $f \in Y_k$ of job J_l, then there exists a path from f to g in the digraph $(Q, A, \emptyset)$ (possibly $f = g$).

In [KSW95], the analogies to Theorems 1.2 and 1.3 for the job shop problem $\mathcal{J}//\mathcal{L}_{max}$ with minimizing maximum lateness (see [LLRKS93]) have been proven and it has been shown that there does not exist an optimal schedule s with $\varrho_s(p) = \infty$ for all other regular criteria (see [LLRKS93]), which are considered in classical scheduling theory.

Formulas for calculating $\widehat{\varrho}_s(p)$ have been derived in [Sot91]. The calculation of the stability radius is reduced to an extremal problem on the given set of digraphs $\Lambda(G) = \{G_1, G_2, \ldots, G_\lambda\}$ with a variable vector of weights assigned to the vertices of digraph $G_k \in \Lambda(G)$. The main objects for the calculation are the sets of dominant paths $H_k,\ k = 1, 2, \ldots, \lambda$, and the following sets as well:

$$H_{sk} = \Big\{\mu \in H_s : \text{ There is no path } \nu \in H_k \text{ such that } [\mu] \subseteq [\nu]\Big\}.$$

Let $p^{\nu\mu}_{(0)}$ be equal to zero and let $(p^{\nu\mu}_{(1)}, p^{\nu\mu}_{(2)}, \ldots, p^{\nu\mu}_{(w_{\nu\mu})})$ denote a non-decreasing sequence of the processing times of $w_{\nu\mu}$ operations from the set $[\nu]\backslash[\mu]$, where $w_{\nu\mu} = |[\nu]\backslash[\mu]|$. Let l^p_s be the critical weight of digraph $G_s(p) \in \Lambda(G)$ at the vector p of the processing times: $l^p_s = \max_{\mu \in H_s} l^p(\mu) = l^p(\mu^*)$, where $\mu^* \in H_s(p)$.

Formulas for calculating the stability radius for criterion $\mathcal{C}_{max}$ have been derived in [Sot91, TSS94]. We code these formulas in Fortran-77 (see Chapter 4).

Theorem 1.4 *If G_s is an optimal digraph for problem $\mathcal{C}_{max}$ and $0 < \widehat{\varrho}_s(p) < \infty$, then*

$$\widehat{\varrho}_s(p) = \min_{k=1,2,\ldots,\lambda; k \neq s} \widehat{r}_{ks}, \tag{1.9}$$

where

$$\widehat{r}_{ks} = \min_{\mu \in H_{sk}} \quad \max_{\nu \in H_k,\ l^p(\nu) \geq l^p_s} \quad \max_{\beta=0,1,\ldots,w_{\nu\mu}} \frac{l^p(\nu) - l^p(\mu) - \sum_{\alpha=0}^{\beta} p^{\nu\mu}_{(\alpha)}}{|[\mu] \cup [\nu]| - |[\mu] \cap [\nu]| - \beta}. \tag{1.10}$$

Equality (1.9) means that one has to compare an optimal digraph $G_s(p)$ with all other feasible digraphs $G_k(p)$. In Section 2.4, we show how it is possible to restrict this enumeration and the comparisons.

1.4 Mean Flow Time

In this section, we survey results from [BSW96, STW98], where the stability radius $\overline{\varrho}_s(p)$ for criterion $\sum \mathcal{C}_i$ has been studied. If $\Phi = \sum \mathcal{C}_i$, conditions (1.3) and (1.4) for the job shop problem are converted to the following conditions (1.11) and (1.12), respectively.

$$\sum_{i=1}^{n} \max_{\mu \in H^i_s} l^p(\mu) = \min_{k=1,2,\ldots,\lambda} \sum_{i=1}^{n} \max_{\nu \in H^i_k} l^p(\nu), \tag{1.11}$$

$$\overline{\varrho}_s(p) = \inf\Big\{d(p, x)\ :\ x \in R^q_+,\ \sum_{i=1}^{n} \max_{\mu \in H^i_s} l^x(\mu) > \min_{k=1,2,\ldots,\lambda;\ k \neq s} \sum_{i=1}^{n} \max_{\nu \in H^i_k} l^x(\nu)\Big\}, \tag{1.12}$$

where $H_k^i \subseteq \tilde{H}_k^i$ is the set of all dominant paths in the digraph G_k ending in the *fixed* vertex $O_{in_i} \in Q^{(i)}$ and starting from *different* vertices $O_{j1} \in Q^{(j)}, j = 1, 2, \dots, n$.

Obviously, the value C_i for a digraph $G_s(p)$ is equal to the largest weight of a path from the set H_s^i, and hence, to solve problem $\mathcal{G}//\sum \mathcal{C}_i$, one must find a digraph $G_s(p)$ such that $L_s^p = \min\{L_k^p : \ k = 1, 2, \dots, \lambda\}$, where $L_k^p = \sum_{i=1}^n \max_{\nu \in H_k^i} l^p(\nu)$ is the sum of the job completion times for the digraph represented by $G_k(p)$ with fixed processing times $p \in R_+^q$. To find the stability radius $\overline{\varrho}_s(p)$, it is possible to consider sets of representatives of the family of sets H_k^i, $1 \leq i \leq n$, which may be denoted as follows. Let Ω_k^u be a set of representatives of the family of sets $(H_k^i)_{J_i \in J}$. Each of these sets Ω_k^u includes exactly one path from each set H_k^i, $J_i \in J$. Since $H_k^i \cap H_k^j = \emptyset$ for any pair of different jobs J_i and J_j, we have the equality $|\Omega_k^u| = n$ and there exist $\omega_k = \prod_{i=1}^n |H_k^i|$ different sets of representatives for digraph G_k, namely: $\Omega_k^1, \Omega_k^2, \dots, \Omega_k^{\omega_k}$. For each set Ω_k^u, one can calculate the integer vector $n(\Omega_k^u) = (n_{11}(\Omega_k^u), n_{12}(\Omega_k^u), \dots, n_{nn_n}(\Omega_k^u))$, where $n_{ij}(\Omega_k^u), i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, n_i\}$, denotes the number of paths in Ω_k^u which include vertex O_{ij}. The value $n_{ij}(\Omega_k^u)$ is equal to the number how often vertex O_{ij} is contained in the multiset $\{[\nu] : \nu \in \Omega_k^u\}$. We denote

$$\begin{aligned}\Omega_{sk} = \{\Omega_s^v :\ & \text{There does not exist a set } \Omega_k^u \text{ such that} \\ & n_{ij}(\Omega_s^v) \leq n_{ij}(\Omega_k^u) \text{ for each } i = 1, 2, \dots, n,\ j = 1, 2, \dots, n_i\}.\end{aligned}$$

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

$$O_{ij_{(1)}}, O_{ij_{(2)}}, \dots, O_{ij_{(m)}}, O_{ij_{(m+1)}}, \dots, O_{ij_{(q)}}, \tag{1.13}$$

where $n_{ij_{(\alpha)}}(\Omega_k^u) \leq n_{ij_{(\alpha)}}(\Omega_s^v)$ for each $\alpha = 1, 2, \dots, m$ and $n_{ij_{(\alpha)}}(\Omega_k^u) > n_{ij_{(\alpha)}}(\Omega_s^v)$ for each $\alpha = m+1, m+2, \dots, q$. For sequence (1.13), the inequalities

$$p_{ij_{(m+1)}} \geq p_{ij_{(m+2)}} \geq \dots \geq p_{ij_{(q)}}$$

have to be satisfied. The following formula was proven in [BSW96].

Theorem 1.5 *If G_s is an optimal digraph for problem $\mathcal{G}//\sum \mathcal{C}_i$ and $0 < \overline{\varrho}_s(p) < \infty$, then*

$$\overline{\varrho}_s(p) = \min_{k=1,2,\dots,\lambda; k \neq s} \overline{r}_{ks}, \tag{1.14}$$

where

$$\overline{r}_{ks} = \min_{\Omega_s^v \in \Omega_{sk}} \max_{\substack{u=1,2,\dots,\omega_k, \\ \sum_{\nu \in \Omega_k^u} l^p(\nu) \geq L_s^p}} \max_{\beta=1,2,\dots,q-m} \frac{\sum_{\alpha=1}^{m+\beta} p_{ij_{(\alpha)}}(n_{ij_{(\alpha)}}(\Omega_k^u) - n_{ij_{(\alpha)}}(\Omega_s^v))}{\sum_{\alpha=1}^{m+\beta} |n_{ij_{(\alpha)}}(\Omega_k^u) - n_{ij_{(\alpha)}}(\Omega_s^v)|}. \tag{1.15}$$

The extreme values of $\overline{\varrho}_s(p)$ were considered in [BSW96, STW98]. Similarly to the notion of a critical path and the critical weight for problem $\mathcal{G}//\mathcal{C}_{max}$, the notion of a critical set $\Omega_k^{u^*}$ and the critical sum of weights for problem $\mathcal{G}//\sum \mathcal{C}_i$ were introduced in [BSW96]. The set $\Omega_k^{u^*}, u^* \in \{1, 2, \dots, \omega_k\}$, is called *critical set* if the value of the objective function

$$L_k^p = \max_{u \in \{1,2,\dots,\omega_k\}} \sum_{\nu \in \Omega_k^u} l^p(\nu) \tag{1.16}$$

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

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

The value L_k^p defined in (1.16) is called *critical sum of weights* for digraph $G_k(p)$. Let $\Omega_k(p)$ denote the set of all critical sets $\Omega_k^{u^*}$ of digraph $G_k(p)$ at the vector $p = (p_{11}, p_{12}, \dots, p_{nn_n}) \in R_+^q$ of the processing times. The following necessary and sufficient conditions for equality $\overline{\varrho}_s(p) = 0$ have been derived in [BSW96].

Theorem 1.6 *Let s be an optimal schedule of 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:*
1) there exists another optimal schedule $k \in S, k \neq s$, and
2) 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 the condition

$$n_{ij}(\Omega_s^{v^*}) \geq n_{ij}(\Omega_k^u), \;\; \Omega_k^u \in \Omega_k(p), \tag{1.17}$$

holds (or the condition

$$n_{ij}(\Omega_s^{v^*}) \leq n_{ij}(\Omega_k^u), \;\; \Omega_k^u \in \Omega_k(p), \tag{1.18}$$

holds) and
3) inequality (1.17) (or inequality (1.18), respectively) is satisfied as a strict one for the set $\Omega_k^{u^*}$.

Corollary 1.4 *If $s \in S$ is a unique optimal schedule for problem $\mathcal{G}//\sum\mathcal{C}_i$, then $\overline{\varrho}_s(p) > 0$.*

The following upper bound for the stability radius of an optimal schedule for problem $\mathcal{G}//\sum\mathcal{C}_i$ was proven in [BSW96].

Theorem 1.7 *If $s \in S$ is an optimal schedule for 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 1.1 As it follows from Theorem 1.7, problem $\mathcal{J}//\sum\mathcal{C}_i$ with $\lambda > 1$ cannot have an optimal schedule with an infinitely large stability radius in contrast to problem $\mathcal{J}//\mathcal{C}_{max}$ and problem $\mathcal{J}//\mathcal{L}_{max}$ (see Section 1.3).

Table 1.1: *Special cases of the shop scheduling problem*

<table>
<tr><th>Characterization
of machine service</th><th>Shop scheduling
problem</th><th>Characterization of the
technological routes of the jobs</th></tr>
<tr><td rowspan="3">Each job has to be processed exactly once on each of the m machines $M = \{M_1, M_2, \dots, M_m\}$</td><td>Open shop
$\mathcal{O}//\Phi$</td><td>Different jobs may have different routes, which are not given a priori</td></tr>
<tr><td>Flow shop
$\mathcal{F}//\Phi$</td><td>All jobs have the same route, which is given a priori</td></tr>
<tr><td>Classical
job shop</td><td rowspan="2">Different jobs may have different routes, which are given a priori</td></tr>
<tr><td>A job may visit a machine more than once</td><td>Job shop
$\mathcal{J}//\Phi$</td></tr>
</table>

Note that all of the results in this section and in Section 1.2 are valid for any general shop scheduling problem. However, we use the partition of the set of operations Q into n chains $Q^{(i)}, i = 1, 2, \ldots, n$, (which is necessary for the job shop and flow shop but is not necessary for the general shop) for a better presentation of the results. Table 1.1 collects special cases of the shop scheduling problem which are characterized by the machine service and the technological routes of the jobs. In scheduling theory often the classical job shop is considered for which each job has to be processed exactly once on each machine (see problem in the third row in Table 1.1). For our consideration this restriction is not important. We will consider the job shop problem $\mathcal{J}m//\Phi$ with *recirculation* (see [Pin95a]), which may occur when a job may visit a machine more than once.

1.5 Related Approaches and Results

The scheduling theory has received a lot of attention among OR practitioners, management scientists, production and operations research workers and mathematicians since the early 1950s. However, the utilization of classical scheduling theory in most production environments is minimal (see [ML93, PL94, Pin95a]). MacCarthy and Liu [ML93] aim the gap between scheduling theory and scheduling practice. They also discuss some research issues which attempt to make scheduling theory more useful in practice. Next, we describe some recent trends in scheduling research which try to make it more relevant and applicable.

For an uncertain scheduling environment *stochastic* models are introduced, where the processing times (and some other parameters) are assumed to be random variables with known probability distributions. For example, such stochastic models for a single machine with the minimization of mean flow time are considered by Chand et al. [CTU96], by Li and Cao [LC95], and with the minimization of earliness-tardiness penalties by Cai and Tu [CT96] as well as by Robb and Rohleder [RR96]. Since it is possible for a company to estimate the times at which jobs are expected to arrive, Chand et al. [CTU96] develop a decomposition approach such that a large problem can be solved by combining optimal solutions of several smaller problems. The model of Robb and Rohleder [RR96] consists of a *probabilistic dynamic scheduling problem* with non-regular performance measures. Using simulation, they explore the robustness of the heuristics with respect to uncertainty in the durations of the operations.

Schmidt [Sch00] reviews some results related to *deterministic* scheduling problems where the machines are not continuously available for processing. The complexity of single and multi-machine problems is analyzed considering criteria depending on the completion times and the due dates. Chu and Gordon [CG00] consider a single machine problem including both due date assignment and the scheduling decision. It is assumed that the due dates are proportional to the job processing times. The objective is to minimize the weighted earliness-tardiness and the penalty related to the size of the dates with respect to the processing times. Jain and Meeran [JM99] present a concise overview of job shop scheduling techniques and the best computational results obtained.

The scheduling problem with an *availability constraint* is very important, as it happens often in the industry. For example, a machine may not be available during the scheduling horizon due to a breakdown (stochastic) or preventive maintenance (deterministic). In an *on-line* setting, machine availabilities are not known in advance. Unexpected machine breakdowns are a typical example of events that arise on-line. Sometimes schedulers have partial knowledge of the availabilities, i.e. they have some 'look-ahead' information. They might know the next time interval where a machine requires maintenance or they might know when a broken machine will be available again [San95]. In an *off-line* setting, one assumes complete information, i.e. all machine availabilities are known prior to the schedule generation [Sch00].

Several *on-line models* have been proposed, and the main difference between these models are the assumptions on the information that becomes available to the scheduler. For a description of these on-line models, we refer to the survey by Sgall [Sga98]. According to [CV97], *on-line* means that jobs arrive over time, and all job characteristics become known at their arrival time [CV97]. Jobs do not have to be scheduled immediately upon arrival. At each time a machine is idle and a job is available, the algorithm decides which one of the available jobs is scheduled, if any. An on-line algorithm for the problem of scheduling jobs on identical parallel machines with the objective of minimizing the makespan is proposed and analyzed by Chen and Vestjens [CV97]. This problem is NP-hard when the *off-line* version is considered, although it can be solved in polynomial time by an on-line algorithm if preemption is allowed [CV97].

Seiden [Sei98] studies on-line scheduling of jobs with fixed start and completion times. Jobs must be scheduled on a single machine which runs at most one job at a given time. The problem is on-line since jobs are unknown until their start times. Each job must be started or rejected immediately when it becomes known. The goal is to maximize the sum of the value the *payoff* (the sum of the values of those jobs which run to completion).

Scheduling problems with *controllable processing times* have received an increasing attention during the last decade. It is often assumed that the actual possible processing time of a job can be continuously controlled, i.e. it can be any number in a given interval. Recent results are presented in [DHM96, KDV00, Str95, Tri94].

Traditional scheduling procedures consider static and deterministic future conditions even though this may not be the case in actual scheduling problems. After a description, the preplanned schedule can become inapplicable to the new conditions. As Graves [Gra81] stated, there is no scheduling problem but rather a *rescheduling problem*. Responding to such dynamic factors immediately as they occur is called *real-time scheduling*. An on-line simulation methodology is proposed by Davis and Jones [DJ88] to analyze several scheduling rules in a stochastic job shop. The job shop rescheduling problem is considered as a particularly hard combinatorial optimization problem (Parunak and van Dyke [PD91]). The production rescheduling problem deals with uncertainty caused by the exterior business environment and interior production conditions. Since it has practical applications, the rescheduling problem is studied by many authors (see e.g. [LLLH00, PD91, SK94]).

A *reactive scheduling approach* is developed by Smith et al. [SOM+90], which uses different knowledge sources and aims to make decisions faster with less emphasis on optimality. For the knowledge-based systems, the most difficult operation is to decide which knowledge source has to be activated. A discussion of knowledge-based reactive scheduling systems can be found in Blazewicz et al. [BESW93] as well as Szelke and Kerr [SK94]. Bean et al. [BBMN91] propose a 'match-up' heuristic method for scheduling problems with disruptions. They show that assuming enough idle time is present in the original schedule and disruptions are sufficiently spaced over time, the optimal rescheduling strategy is to match-up with the preschedule at some time in the future. The objective in [AG99] is to create a new schedule that is consistent with the order production planning decisions like material flow, tooling and purchasing. When a machine breakdown forces a modified flow shop out of the prescribed state, the proposed strategy reschedules a part of the initial schedule to match-up with the preschedule at some point.

Fuzzy scheduling techniques proposed in the literature either fuzzify directly the existing scheduling rules, or solve mathematical programming problems to determine the optimal schedules. The optimality of a fuzzy logic alternative to the usual treatment of uncertainties in a scheduling system using probability theory was examined by Ozelkan and Duckstein [OD99]. The purpose of the latter paper was to investigate necessary optimality conditions of fuzzy counterparts of 'classical' dispatching rules, such as the shortest processing time (SPT) and the earliest due date (EDD). Essentially, *any element of a scheduling problem may be uncertain.*

Dumitru and Lubau [DL82] propose fuzzy mathematical models to solve the job shop problem. Grabot and Geneste [GG94] use a fuzzy rule-based approach to find a compromise between different job shop dispatching rules. Kuroda and Wang [KW96] also analyze fuzzy job shop problems using a branch-and-bound algorithm to obtain results for lateness related criteria. A mathematical programming approach to a single machine scheduling problem with *fuzzy precedence relation* is given in [IT95]. Job shop scheduling with both fuzzy processing times and fuzzy due dates are proposed in [SK00]. Sakawa and Kubota [SK00] formulate a multiobjective fuzzy job shop problem as three-objective ones which not only maximizes the minimum agreement index but also maximizes the average agreement index and minimizes the maximum fuzzy completion time. Generally, the topic of fuzzy scheduling has received much attention during the last decade. Slowinski and Hapke [SH99] collect the main works.

In most of the classical shop scheduling models, it is assumed that an individual processing time incorporates all other time parameters (lags) attached to a job or to an operation. In practice, however, such parameters often have to be viewed separately from the actual processing times. For example, if for an operation some pre-processing and/or post-processing is required, then we obtain a scheduling model with *set-up* and/or removal times separated. Strusevich [Str99] considers a two-machine open shop problem with involved interstage transportation times. He assumes that there is a known *time lag* (transportation time) between the completion of an operation and the beginning of the next operation of the same job.

The majority of scheduling research assumes *set-up* as negligible or as a part of the processing time. While this assumption simplifies the analysis, it adversely affects the solution quality for many applications which require an explicit treatment of set-up times. Such applications, coupled with the emergence of production concepts like time-based competition and group technology, have motivated an increasing interest to include set-up considerations in scheduling problems. The paper [AGA98] provides a comprehensive review of the literature on scheduling problems involving set-up times (set-up costs). In [All97], Allahverdi considers a two-machine flow shop problem with the objective to minimize the expected makespan where machines suffer breakdowns and the job set-up and removal times are separated from the processing times. The same author [All95] proposes a dominance relation where no assumption about the breakdown processes is made. In general, such a dominance relation does not yield optimal schedules. However, if certain assumptions about the breakdowns distributions and counting processes hold, it is possible to obtain an optimal schedule.

Decision-makers often consider multiple objectives when making scheduling decisions. However, very little research has been done in multiple machine environments with *multiple objectives.* Allahverdi and Mittenthal [AM98] consider a two-machine flow shop scheduling problem, where machines suffer random breakdowns and processing times are constant, with respect to both the makespan and the maximum lateness objective functions. Kyparisis and Koulamas [KK00] study the two-machine open shop problem with a hierarchical objective: Minimize the total completion time subject to minimum makespan $\mathcal{O}2//\sum \mathcal{C}_i|\mathcal{C}_{max}$.

Cheng and Shakhlevich [CS98a] consider a special class of flow shop problems, known as the *proportionate flow shop*. In such a shop, each job flows through the machines in the same order and has equal processing times on the machines. It is assumed that all operations of a job may be compressed by the same amount which will incur an additional cost. The objective is to minimize the makespan of the schedule together with a compression cost function which is nondecreasing with respect to the amount of compression. A *bicriterion approach* to solve the single machine scheduling problem in which the job release dates can be compressed while incurring additional costs, is considered in [CS98b].

Stein and Wein [SJ97] give a proof that, for any instance of a rather general class of scheduling problems, there exists a schedule with a makespan at most twice that of the optimal value and of a total weighted completion time at most twice that of the optimal value.

Brucker and Krämer [BK96b] derive complexity results for *resource-constrained scheduling problems* with a fixed number of operation types in which either the processing times are bounded or the number of processors is fixed. They consider shop problems with multiprocessor operations, in which either the number of jobs or the number of stages is fixed. They present polynomial time algorithms for these problems with makespan, mean flow time, weighted number of tardy operations, and sum of tardiness as objective functions.

The papers above address problems of practical importance in planning, scheduling, and control. It is therefore important to produce schedules that are both stable (robust) and adaptable to system disturbances. More importantly, it offers unique properties that lead to a more effective planning and control methods for systems under uncertainty. There exist a lot of papers presenting different approaches to stability analysis of *discrete optimization problems*, and in the last part of this section, we provide a sketch of some approaches to stability analysis, which are close to the subject of this dissertation.

A related approach to stability analysis for *linear trajectory problems* (such as the *traveling salesman problem*, the *assignment problem*, the *shortest path problem*) and some other discrete optimization problems has been initiated in [GL80, Leo75, Leo76, Lib91, SW80, Tar82] and developed in some other papers (see Sotskov et al. [SLG95] for an extensive survey). Most results have been obtained for the stability radius of the whole set of solutions (optimal trajectories), i.e. for the largest radius $\varrho(p)$ of an open ball in the space of the numerical input data p such that a new optimal trajectory does not arise. A formula for calculating the stability radius $\varrho(p)$ of the set of all solutions of the traveling salesman problem is obtained by Leontev [Leo75, Leo76] and the extreme values of $\varrho(p)$ are determined. Gordeev and Leontev [GL80] derive analogous results for a similar problem with a *bottleneck objective function*. A specific transformation of a branch-and-bound algorithm for the traveling salesman problem for calculating $\varrho(p)$ is suggested by Gordeev et al. [GLS83]. Gordeev [Gor89] proposes a polynomial algorithm for calculating the stability radius of the whole set of solutions of *extremal problems on matroids* and on the intersection of two matroids.

It should be noted that related approaches to stability analysis for the traveling salesman problem, the shortest path problem, and some others, which can be represented as a *binary optimization problem* with a linear objective function, are developed in [GL80, Lib99, Lib91, LvdPSvdV96, Tar82].

The complexity of calculating the stability radius $\varrho(p)$ of a solution of a discrete optimization problem is studied in [GL85, RC95]. Ramaswamy and Chakravarti [RC95] show that the problem of determining the *arc tolerance* for a discrete optimization problem is as hard as the problem itself (the arc tolerance is the maximum change, i.e. increase or decrease, of a single weight, which does not destroy the optimality of a solution). This means that in the case of the traveling salesman problem, the arc tolerance problem is NP-hard even if an optimal tour is given. Gordeev [Gor89] proved the NP-hardness of the problem of calculating $\varrho(p)$ for the polynomially solvable shortest path problem in a digraph without negative circuits. Sotskov et al. [SWW98] show that the stability radius of an approximate solution may be calculated in polynomial time if the number of unstable components grows rather slowly, namely as $O(\log N)$, where N is the number of cities in the traveling salesman problem. Libura et al. [LvdPSvdV96] argue that it is rather convenient from a computational point of view to use the set of k shortest tours when applying a stability analysis to the *symmetric traveling salesman problem.*

An extensive survey of the obtained results within such an a posteriori analysis is given in [SLG95]. Greenberg [Gre97] categorizes types of postoptimal sensitivity analyses and gives a survey of the literature started in the late 1970's. A primary concern of sensitivity analysis is how the optimal solution values change when the data changes. The subject of post-solution analysis includes debugging a scenario, such as when it is anomalous, unbounded or infeasible.

In spite of obvious practical importance, the literature on stability analysis in scheduling is rather small. Outside the considered approach, one can mention [KRKvHW94, Mel78, PQ78]: In [KRKvHW94], the sensitivity of a heuristic algorithm with respect to the variation of the processing time of one job is investigated, in [Mel78] the stability of an optimal schedule for the flow shop problem $\mathcal{F}//\mathcal{C}_{max}$ is considered, and in [PQ78] the results for the traveling salesman problem are used for a one machine scheduling problem with minimizing tardiness (see [LLRKS93]).

In general, studying a scheduling problem with *uncertain processing times* and *its sensitivity analysis* is of importance. The reasons can be illustrated by giving references to practical applications. In many cases the data used are imprecise due to uncertainty with respect to the exact parameter values or due to errors in the measurement. In industrial applications of mathematical programming models, there are almost always uncertain elements that are assumed away or suppressed in the formal description of the model (see [Wag95]).

We have to emphasize that the random processing times $p_i, i \in Q$, in problem $\mathcal{G}/a_i \leq p_i \leq b_i/\Phi$ are due to external forces in contrast to scheduling problems with *controllable* processing times, see e.g. [DHM96, IMN87, IN86, Jan88, Str95, Tri94], 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 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.

To model scheduling in an uncertain environment, a *two-person non-zero sum game* is introduced by Chrysslouris et al. [CDL94], where the decision-maker was considered as player 1 and the 'nature' as player 2.

Next, we observe known results for *makespan* minimization under 'strict uncertainty' of the numerical input data. Lai and Sotskov [LS99] use 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 [LS99] is focused on the set of paths in $G_s \in \Lambda(G)$ which may be critical.

In [LS99], the critical path method [Dij59] is 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 is characterized in [LS99], where an exact and a heuristic algorithm are developed for problem $\mathcal{J}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$. Note that the approach developed in [LS99] is based on the stability property of a makespan optimal schedule, which is theoretically investigated in [KSW95, Sot91, SWW98] and in some other papers (see [SLG95, STW98] for surveys of stability analysis for scheduling problems).

Briefly, the main issue of the research presented in [LS99] is to simplify the digraph G_s due to the existence of two types of dominance relations between its paths (see Section 1.2 and 2.2). In this dissertation, we perform a further step in this direction by focusing on two types of dominance relations between feasible digraphs (schedules) (see Section 3.1 below). This step is useful for shop scheduling problems under 'strict uncertainty' with both C_{max} and $\sum C_i$ criteria since it allows to reduce significantly the number of schedules which are sufficient to consider as candidates for a solution.

As follows from [BHTW99a, BHTW99b, BK96a], a reduction of the digraphs may be essential even for all non-negative perturbations of the processing times: $0 \leq p_i < \infty$. Bräsel et al. [BHTW99a], Bräsel et al. [BHTW99b] and Bräsel and Kleinau [BK96a] introduce 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 non-negative processing times, this set contains at least one optimal schedule. On the basis of computations with $n \leq 3$ and $m \leq 7$, it is 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 [BHTW99a], it is 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 $\mathcal{J}/0 \leq p_i < \infty/\mathcal{C}_{max}$ is a subject of [BHTW99a] and the open shop problem $\mathcal{O}/0 \leq p_i < \infty/\mathcal{C}_{max}$ is a subject of [BHTW99a, BK96a].

Kouvelis et al. [KDV00] focus on manufacturing environments where 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 *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 job processing times. The paper [KDV00] focuses on a two-machine flow shop problem and the performance measure of interest is the makespan criterion. A similar robust scheduling approach is developed for a single-machine problem by Daniels and Kouvelis [DK95]. Other robust decision-making formulations are presented by Rosenblatt and Lee [RL87], Kouvelis et al. [KKG92] and Mulvey et al. [MVZ95].

Leon et al. [LWS94] consider robustness measures and robust scheduling methods that generate job shop schedules that maintain high performance over a range of system disturbances. Wu et al. [WBS99] study the weighted tardiness job shop problem. A basic thesis of the latter paper is that "global scheduling performance is determined primarily by a subset of the scheduling decisions to be made". Wu et al. [WBS99] propose to identify a critical subset of the scheduling decisions at the beginning of the planning horizon and relegate the rest of the scheduling decisions to future points in time. Our approach considered in Chapters 2, 3 and 4 uses a similar idea.

Chapter 2

General Shop Problem with Makespan Criterion

This chapter deals with the general shop problem with the objective to minimize the makespan provided that the numerical input data are uncertain. In a stochastic setting, the random processing time of an operation is assumed to take a known probability distribution. The scheduling environments that we consider are so uncertain that all information available about the processing time of an operation is its upper and lower bounds. We present an approach to deal with problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ based on an improved stability analysis of an optimal schedule and demonstrate this approach on an example of the job shop problem $\mathcal{J}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$. In the course of this chapter, an *optimal* schedule (digraph), a *better* and a *best* schedule (digraph) are considered with respect to criterion $\mathcal{C}_{max}$.

All necessary notions from the paper [LSSW97] are generalized for problem $\mathcal{G}/a_i \leq p_i \leq b_i/\Phi$ for a more effective use. Some propositions which are not proven here are presented for a complete account of the theory. In [SSW97], a bound of the stability radius was used to restrict the number of digraphs considered for calculating the stability radius (these results are given in Section 2.4).

2.1 Solution and Minimal Solution

Let us consider a general shop problem as described in Section 1.1. In a *deterministic* setting, the processing times p_i are assumed to be known exactly for all operations $i \in Q$, and as it was mentioned in Chapter 1, a schedule may be defined by the start times s_i or completion times c_i of all operations $i \in Q$. Given a fixed vector $p = (p_1, p_2, \ldots, p_q)$ of the processing times, in order to construct an optimal schedule for the general shop problem $\mathcal{G}//\mathcal{C}_{max}$, one may first enumerate (explicitly or implicitly) all *feasible* digraphs $G_1(p), G_2(p), \ldots, G_\lambda(p)$ and then select an *optimal* digraph, i.e. one with a minimal value of the critical weight among all λ feasible digraphs. It is worthwhile to note that the *feasibility* of a digraph $G_s(p)$ is independent of the vector $p = (p_1, p_2, \ldots, p_q)$ of the processing times, while the *optimality* of a digraph depends on the vector p. In other words, the set $\Lambda(G) = \{G_1, G_2, \ldots, G_\lambda\}$ of feasible digraphs is completely defined by the mixed graph $G = (Q, A, E)$ (without 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, i.e. the optimality of a schedule is defined by the weighted mixed graph $G(p) = (Q(p), A, E)$.

If the vector p of the processing times is not known exactly before applying a scheduling procedure (e.g., the processing times may *vary* in a practical realization), different realizations may result in different critical paths in the weighted digraph $G_s(p)$. For practical problems, the cardinality λ of the set $\Lambda(G)$ may be huge. However, as we will show, we often need only to consider some subset B of the set $\Lambda(G) : B \subseteq \Lambda(G)$. From the equality (1.8) it follows that digraph $G_s(p)$ has the minimal critical weight within the set $B \subseteq \Lambda(G)$ if and only if

$$\max_{\mu \in H_s} l^p(\mu) = \min_{G_k \in B} \max_{\nu \in H_k} l^p(\nu). \tag{2.1}$$

For the case $B = \Lambda(G)$, the equality (2.1) provides an *optimality* criterion of a schedule $s \in S$ (if vector p is fixed).

In this chapter, we allow the duration p_i of an operation $i \in Q$ to assume any value in the fixed closed interval $[a_i, b_i]$, where $0 \leq a_i \leq b_i$, (see Assumption 3). As it was already mentioned, the deterministic problem $\mathcal{G}//\mathcal{C}_{max}$ is a special case of a general shop problem with *uncertain* processing times $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ when $a_i = b_i$ for each operation $i \in Q$. Also, one can interpret p_i in problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ as a random variable x_i with the following cumulative probability distribution function:

$$F_i(t) = P(x_i < t) = \begin{cases} 0, & \text{if } t < a_i, \\ 1, & \text{if } t = b_i, \end{cases}$$

where the density function of a cumulative probability distribution is uncertain in the closed interval $[a_i, b_i]$ for operation $i \in Q$.

In the framework of stochastic scheduling ([Pin95a], pp. 167 – 252), each random variable x_i associated with the processing time of the operation $i \in Q$ is assumed to have a *known* probability distribution. For example, a stochastic variant of problem $\mathcal{G}//\mathcal{C}_{max}$ with exponential continuous time distributions with rates α_i, $i \in Q$, is denoted by $\mathcal{G}/p_i \sim exp(\alpha_i)/E\mathcal{C}_{max}$, where the density function of an exponentially distributed random variable x_i is $f_i(t) = \alpha_i e^{-\alpha t}$, the corresponding probability distribution function is $F_i(t) = P(x_i < t) = 1 - e^{-\alpha t}$, and

$$E_i(x_i) = \int_0^\infty t f_i(t) dt = \int_0^\infty t dF_i(t) = \frac{1}{\alpha_i}$$

is the mean or expected value of x_i. The objective of problem $\mathcal{G}/p_i \sim exp(\alpha_i)/E\mathcal{C}_{max}$ is to minimize the expected makespan $E\mathcal{C}_{max}$ of a schedule using an appropriate scheduling policy.

The approach we present in this chapter for solving problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ is based on an improved stability analysis of an optimal digraph (see Section 1.3). This chapter is organized as follows. In the rest of this section, we demonstrate some preliminary ideas using an example of a job shop problem. Section 2.2 deals with the required mathematical background for later presentations. In Section 2.3, we present the main formula and an algorithm for solving problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ and in Section 2.4, we show how to restrict the number of digraphs which we have to consider during the calculation of the stability radius. A summary and some remarks to this chapter are given in Section 2.5.

As follows from Chapter 1, an optimal digraph $G_s \in \Lambda(G)$ provides a solution of problem $\mathcal{G}//\mathcal{C}_{max}$. In other words, an optimal digraph defines a set of m sequences of the operations Q_k processed on machine $M_k, k = 1, 2, \ldots, m$, with a minimal value of the makespan among all feasible schedules when the vector $p = (p_1, p_2, \ldots, p_q)$ of the processing times is given.

Next, we define a solution of problem $\mathcal{G}/a_i \le p_i \le b_i/\Phi$ for the general case $a_i \le b_i, i \in Q$. Let T denote the polytope in the space R^q_+ defined by inequalities (1) (see page 3):

$$T = \{x = (x_1, x_2, \ldots, x_q) : \ a_i \le x_i \le b_i, \ i \in Q\}.$$

A set $\Lambda^*(G) \subseteq \Lambda(G)$ of feasible digraphs is called a *solution* of problem $\mathcal{G}/a_i \le p_i \le b_i/\Phi$ if this set contains at least one optimal digraph for each fixed vector $x \in T$ of the processing times. Obviously, the whole set $\Lambda(G)$ may be considered as a solution of problem $\mathcal{G}/a_i \le p_i \le b_i/\Phi$ with any given polytope $T \subseteq R^q_+$, i.e. for each pair of vectors $a = (a_1, a_2, \ldots, a_q) \in R^q_+$ and $b = (b_1, b_2, \ldots, b_q) \in R^q_+$ with $a_i \le b_i, \ i = 1, 2, \ldots, q$. However, such a solution is usually redundant: Some digraph from the set $\Lambda(G)$ cannot be optimal for any point p from the polytope T. Moreover, the construction of the whole set $\Lambda(G)$ is only possible for a small problem size since the cardinality λ of the set $\Lambda(G)$ could be equal to $2^{|E|}$. Note also that during the realization of a schedule, a decision-maker may have difficulties dealing with such a huge set of possible candidates of schedules for realization. Therefore, it is practically important to look for a *'minimal solution'* $\Lambda^*(G) \subseteq \Lambda(G)$ for problem $\mathcal{G}/a_i \le p_i \le b_i/\Phi$. A set $\Lambda^*(G)$ is called a minimal solution if any proper subset of $\Lambda^*(G)$ is not a solution. Note that $\Lambda^*(G)$ may be not unique since there may exist two or more optimal digraphs for some vector $p \in T$ of the processing times. We combine these arguments as follows.

Definition 2.1 *A set of digraphs $\Lambda^*(G) \subseteq \Lambda(G)$ is called a solution of problem $\mathcal{G}/a_i \le p_i \le b_i/\Phi$ if for each fixed vector $p \in T$ of the processing times the set $\Lambda^*(G)$ contains an optimal digraph. If any proper subset of the set $\Lambda^*(G)$ is no longer a solution of problem $\mathcal{G}/a_i \le p_i \le b_i/\Phi$, it is called a minimal solution and we denote it by $\Lambda^T(G)$.*

Table 2.1 summarizes the mixed graph approach to the general shop problem with criterion $\mathcal{C}_{max}$ in accordance with the availability of the information on the vector p of the processing times. Note that row 1 of Table 2.1 refers to the *mass* general shop problem, where the only information requirement on p is that p belongs to the vector space R^q_+.

Table 2.1: *Scheduling with different requirements on the numerical data*

	Scheduling problem	Input data	Feasible solutions	Optimal solution
1	Mass general shop scheduling problem	$G = (Q, A, E)$; $0 \le p_i \le \infty, \ i \in Q$	$\Lambda(G)$	$\Lambda(G)$
2	Individual problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$	$G(p) = (Q(p), A, E)$; $a_i \le p_i \le b_i, \ i \in Q$	$\Lambda(G)$	$\Lambda^*(G) \subseteq \Lambda(G)$
3	Individual problem $\mathcal{G}/p_i \sim F_i(t)/E\mathcal{C}_{max}$	$G(x) = (Q(x), A, E)$; $F_i(t) = P(x_i < t)$	$\Lambda(G)$	$\{G_s\} \subseteq \Lambda^*(G) \subseteq \Lambda(G)$
4	Individual problem $\mathcal{G}//\mathcal{C}_{max}$	$G(p) = (Q(p), A, E)$; $a_i = p_i = b_i, \ i \in Q$	$\Lambda(G)$	$G_s \in \Lambda(G)$

Note that *any digraph $G_s \in \Lambda(G)$ may become optimal in some realization of the process.* Indeed, we can set p_i equal to a sufficiently small real $\epsilon \ge 0$ for each $i \in Q^0 =$

$[\mu] \setminus \cup_{k \neq s} \cup_{\nu \in H_k(p)} [\nu]$, where $\mu \in H_s$ is a critical path in digraph G_s. For such a setting of the processing times, equality (2.1) is satisfied with $B = \Lambda(G)$. In particular, if $Q^0 = Q$, we get a trivial individual problem $\mathcal{G}//\mathcal{C}_{max}$ with $p_i = \epsilon = 0,\ i \in Q$, where any feasible digraph G_s is optimal.

In this chapter, we consider an individual problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$ which is very general (see row 2 in Table 2.1). In one extreme case when $a_i = 0$ and $b_i = \infty$ for each $i \in Q$, problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$ coincides with the whole mass problem presented in row 1. In the other extreme case when $a_i = b_i$ for each $i \in Q$, problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$ reduces to problem $\mathcal{G}//\mathcal{C}_{max}$ (see row 4), which is the basic problem studied in deterministic scheduling. The more information about the processing times is available before applying a scheduling procedure, the 'better' the solution obtained may be: The cardinality of the minimal solution $\Lambda^T(G)$ is smaller if the polytope T is defined by smaller closed intervals $[a_i, b_i]$. For example, a minimal solution set reduces to a single optimal digraph $G_s \in \Lambda(G)$ in the case of problem $\mathcal{G}//\mathcal{C}_{max}$ (see row 4).

Row 3 refers to the individual problem $\mathcal{G}/p_i \sim F_i(t)/E\mathcal{C}_{max}$, a basic problem studied in stochastic scheduling, where each operation $i \in Q$ is assumed to be a random variable with a probability distribution $F_i(t)$ known before applying a scheduling procedure. For problem $\mathcal{G}/p_i \sim F_i(t)/E\mathcal{C}_{max}$, the optimal solution may be a single digraph G_s when one adopts a static scheduling policy ([Pin95a], p.178) or a subset of feasible digraphs $\Lambda^*(G)$ when one adopts a dynamic scheduling policy ([Pin95a], p.179). When a static scheduling policy is adopted, a decision-maker constructs and uses an optimal schedule $s \in S$ which minimizes the expected makespan $E\mathcal{C}_{max}$ and schedule s remains unchanged during the entire process. In the case of a dynamic scheduling policy, an initial schedule s is constantly revised during the process based on the updated information available [Pin95a]. We can note that the minimal solution set $\Lambda^T(G)$ for problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$, may be calculated exactly before the realization of the process, while for problem $\mathcal{G}/p_i \sim F_i(t)/E\mathcal{C}_{max}$ the solution may vary and may even be the whole set $\Lambda(G)$ for a lot of probability distributions $F_i(t)$.

It is worth to note that for all four formulations presented in Table 2.1, the set of feasible solutions remains the same and therefore the properties of feasible digraphs $\Lambda(G)$ are of particular importance. Our approach for solving problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$ is based on the stability of an optimal digraph which guarantees that a feasible digraph remains optimal after some possible variations of the processing times.

Example 2.1 To facilitate the presentation of the main ideas of our approach, let us

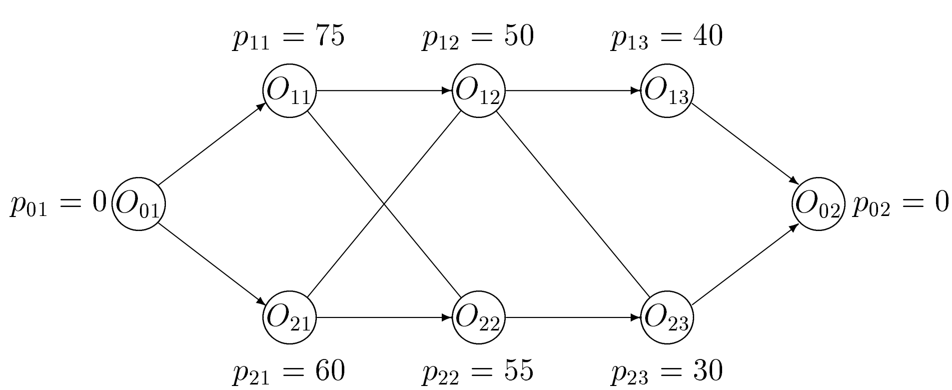


Figure 2.1: *Weighted mixed graph $G(p) = (Q(p), A, E)$*

consider the following job shop problem with two jobs J_1 and J_2 and five machines $M = \{M_1, M_2, \dots, M_5\}$, where job J_1 (job J_2) consists of the set of ordered operations $\{O_{11}, O_{12}, O_{13}\} = Q^{(1)}$ (ordered operations $\{O_{21}, O_{22}, O_{23}\} = Q^{(2)}$, respectively). The assignment of operations $Q = \{O_{01}, O_{02}, O_{11}, O_{12}, O_{13}, O_{21}, O_{22}, O_{23}\}$ to the set of machines M is as follows: $Q_1 = \{O_{11}, O_{22}\}$, $Q_2 = \{O_{12}, O_{21}, O_{23}\}$, $Q_3 = \{O_{13}\}$, $Q_4 = \{O_{01}\}$ and $Q_5 = \{O_{02}\}$. Operation $O_{ij} \in Q$ is called *dummy* operation if $p_{ij} = 0$. To accommodate dummy operations in the framework of the mixed graph, we assume that each dummy operation 'has to be processed' by a special dummy machine with a zero processing time, and we assume that the number of dummy machines is equal to the number of dummy operations. Therefore, each dummy operation is an isolated vertex in the graph $(Q, \emptyset, E)$. Operations O_{01} and O_{02} and machines M_4 and M_5 are dummy, where operation O_{01} (operation O_{02}) denotes the start (the end) of a schedule and so it precedes all other operations (all other operations precede operation O_{02}).

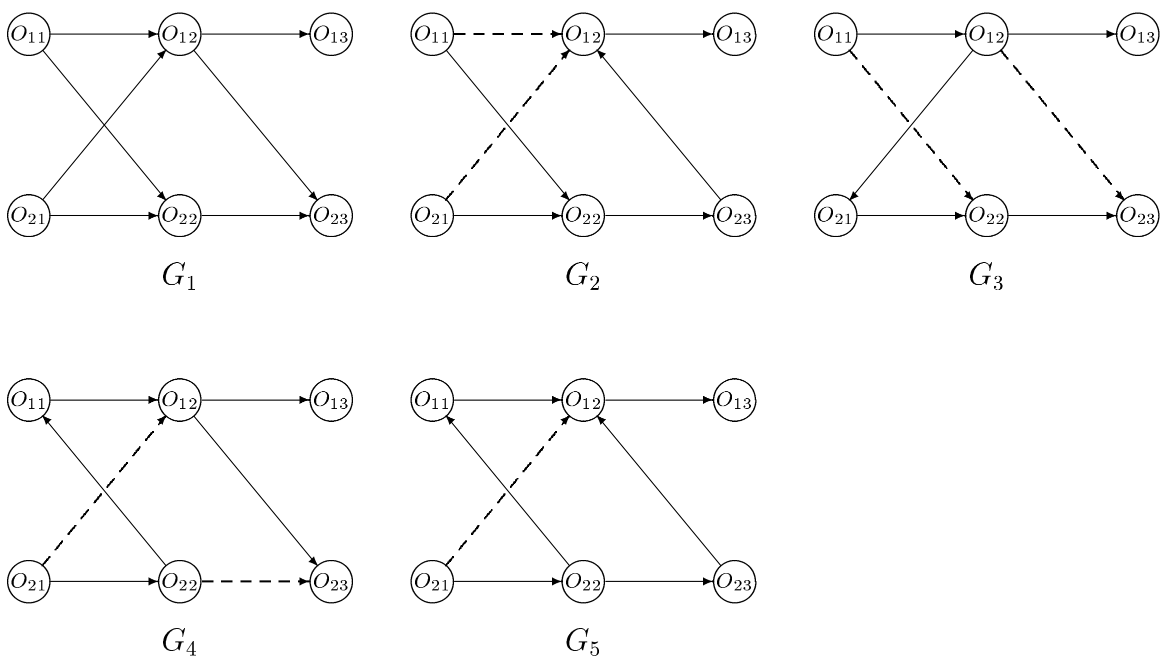


Figure 2.2: *Set of feasible digraphs $\Lambda(G) = \{G_1, G_2, \dots, G_5\}$ for problem $\mathcal{J}3/n{=}2/\mathcal{C}_{max}$*

The input data of Example 2.1 is represented by the mixed graph $G(p) = (Q(p), A, E)$ in Figure 2.1, where each processing time p_{ij} is presented near the vertex $O_{ij} \in Q$, and the vector p of the processing times is as follows: $p = (75, 50, 40, 60, 55, 30)$ (without dummy operations). For this small example, one can explicitly enumerate all feasible digraphs $\Lambda(G) = \{G_1, G_2, \dots, G_5\}$ (these digraphs but without dummy operations O_{01} and O_{02} are represented in Figure 2.2), calculate their makespans:

$$\begin{aligned}
l_1^p &= \max\{c_{in_i}(1) : J_i \in J\} = 165, \\
l_2^p &= \max\{c_{in_i}(2) : J_i \in J\} = 250, \\
l_3^p &= \max\{c_{in_i}(3) : J_i \in J\} = 270, \\
l_4^p &= \max\{c_{in_i}(4) : J_i \in J\} = 280, \\
l_5^p &= \max\{c_{in_i}(5) : J_i \in J\} = 280,
\end{aligned}$$

and select an optimal digraph $G_1(p) = (Q(p), A \cup E_1, \emptyset)$ with the signature $E_1 = \{(O_{11}, O_{22}), (O_{21}, O_{12}), (O_{12}, O_{23})\}$. Digraph $G_1(p)$ has a minimal critical weight equal to 165 (see Figure 2.3).

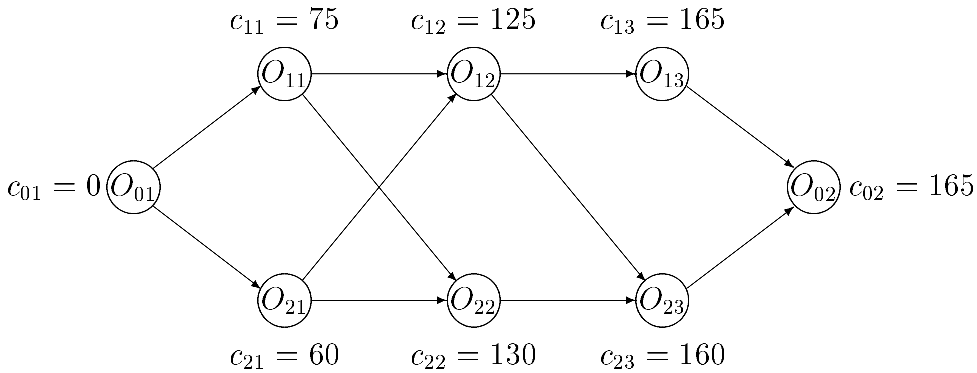


Figure 2.3: *Optimal digraph $G_1 = (Q, A \cup E_1, \emptyset)$ with the completion times c_{ij} presented near the vertices $O_{ij} \in Q$*

In our theoretical results, the job shop problem is considered without the start and end operations O_{01} and O_{02}, which are dummy and not used in our calculation. (However, in the software developed we also use dummy start and end operations for a better organization of the programmed algorithms.) For problem $\mathcal{J}2/n{=}3, a_i \le p_i \le b_i/\mathcal{C}_{max}$, the set of feasible digraphs is presented in Figure 2.2, where the transitive arcs are indicated as dotted lines.

Using formulas (1.9) and (1.10), one can calculate the stability radius of the optimal digraph $G_1(p)$. In Table 2.2, we present our calculations according to formula (1.9) in detail. To this end, we compare digraph $G_1(p)$ with each digraph $G_k \in \Lambda(G) \setminus \{G_1\}$. More exactly, each path $\mu \in H_{1k} \subseteq H_1$ in $G_1(p)$ presented in column 3, is compared with each path $\nu \in H_k$ presented in column 4, for which $l^p(\nu) \ge l_1^p = 165$. The cardinalities of the sets $H_{1k}, k = 2, 3, \dots, \lambda$, are given in column 2. Since $H_{12} = \emptyset$ and $H_{14} = \emptyset$, digraphs G_2 and G_4 are not involved in the computations. The non-decreasing sequence of processing times $(p^{\nu\mu}_{(0)}, p^{\nu\mu}_{(1)}, \dots, p^{\nu\mu}_{(\omega_{\nu\mu})})$ defined at page 15, is given in column 5. In column 6 we present the calculations according to the fraction in the formula (1.9) consecutively for each $\beta = 0, 1, \dots, \omega_{\nu\mu}$. In columns 7, 8 and 9, we extract the maximum for $\beta = 0, 1, \dots, w_{\nu\mu}$, the maximum for $\nu \in H_k, l^p(\nu) \ge l_1^p$, and the minimum for $\mu \in H_{1k}$, respectively, from the values obtained in column 6. In other words, column 9 presents the values of r_{k1} for the digraphs G_k. The last step is to adapt the formula (1.10) from Theorem 1.4 (see page 15). The minimum value $\widehat{r}_{ks}$ is given in column 9.

Thus, for Example 2.1 one calculates due to formula (1.9) that $\widehat{\varrho}_1(p) = \min\{30, 30\} = 30$. Thus, digraph G_1 remains optimal if the variation x_{ij} of each processing time p_{ij} from the set $\{p_{11} = 75, p_{12} = 50, p_{13} = 40, p_{21} = 60, p_{22} = 55, p_{23} = 30\}$ is no more than 30, $p_{ij} - 30 \le x_{ij} \le p_{ij} + 30$. Therefore, when solving problem $\mathcal{J}2/n{=}3, a_i \le p_i \le b_i/\mathcal{C}_{max}$, digraph $G_1 = (Q, A \cup E_1, \emptyset)$ remains optimal if for all possible variations of the processing times $x = (x_{11}, x_{12}, \dots, x_{23}) \in O_{\widehat{\varrho}_1(p)}(p) = O_{30}(p)$, the following inequality holds:

$$\max_{O_{ij} \in Q} \{x_{ij} - a_{ij}, b_{ij} - x_{ij}\} \le 30. \tag{2.2}$$

In such a case, the given polytope T defined by inequalities (1) in the space R^q_+ is completely contained in the stability ball $O_{30}(p)$ of the optimal digraph $G_1 : \ T \subseteq O_{30}(p)$.

Table 2.2: *Calculation of the stability radius $\hat{\varrho}_1(p)$ for problem $\mathcal{J}3/n=2/\mathcal{C}_{max}$*

G_k	$\|H_{1k}\|$	$\mu\in H_{1k}$, $l^p(\mu)$	$\nu\in H_k$: $l^p(\nu)\geq l^p_1$	$p^{\nu\mu}_{(\beta)}$, $0\leq\beta\leq w_{\nu\mu}$	$\frac{l^p(\nu)-l^p(\mu)-\sum_{\alpha=0}^{\beta}p^{\nu\mu}_{(\alpha)}}{\|[\mu]\cup[\nu]\|-\|[\mu]\cap[\nu]\|-\beta}$	$\max_\beta$	$\max_\nu$	$\min_\mu$
1	2	3	4	5	6	7	8	9
G_2	0							
G_3	1	(O_{21},O_{12},O_{13}), $l^p(\mu)=150$	(O_{11},O_{12},O_{13}): $l^p(\nu)=165=l^p_1$	$p^{\nu\mu}_{(0)}=0$ $p^{\nu\mu}_{(1)}=75$	$\frac{165-150-0}{2-0}=7.5$ $\frac{165-150-75}{2-1}<0$	7.5	30	30
			$(O_{11},O_{12},O_{21},$ $O_{22},O_{23})$: $l^p(\nu)=270>165$	$p^{\nu\mu}_{(0)}=0$ $p^{\nu\mu}_{(1)}=30$ $p^{\nu\mu}_{(2)}=55$ $p^{\nu\mu}_{(3)}=75$	$\frac{270-150-0}{4-0}=30$ $\frac{270-150-30}{4-1}=30$ $\frac{270-150-(30+55)}{4-2}=17.5$ $\frac{270-150-(30+55+75)}{4-3}<0$	30		
G_4	0							
G_5	2	(O_{11},O_{12},O_{23}), $l^p(\mu)=155$	$(O_{21},O_{22},O_{11},$ $O_{12},O_{13})$: $l^p(\nu)=280>165$	$p^{\nu\mu}_{(0)}=0$ $p^{\nu\mu}_{(1)}=40$ $p^{\nu\mu}_{(2)}=50$ $p^{\nu\mu}_{(3)}=60$	$\frac{280-155-0}{4-0}=31.25$ $\frac{280-155-40}{4-1}=28\frac{1}{3}$ $\frac{280-155-(40+55)}{4-2}=15$ $\frac{280-155-(40+55+60)}{4-3}<0$	31.25	31.25	30
			$(O_{21},O_{22},O_{23},$ $O_{12},O_{13})$: $l^p(\nu)=235>165$	$p^{\nu\mu}_{(0)}=0$ $p^{\nu\mu}_{(1)}=40$ $p^{\nu\mu}_{(2)}=55$ $p^{\nu\mu}_{(3)}=60$	$\frac{235-155-0}{4-0}=20$ $\frac{235-155-40}{4-1}=13\frac{1}{3}$ $\frac{235-155-(40+55)}{4-2}<0$ $\frac{235-155-(40+55+60)}{4-3}<0$	20		
		(O_{11},O_{22},O_{23}), $l^p(\mu)=160$	$(O_{21},O_{22},O_{11},$ $O_{12},O_{13})$: $l^p(\nu)=280>165$	$p^{\nu\mu}_{(0)}=0$ $p^{\nu\mu}_{(1)}=40$ $p^{\nu\mu}_{(2)}=50$ $p^{\nu\mu}_{(3)}=60$	$\frac{280-160-0}{4-0}=30$ $\frac{280-160-40}{4-1}=26\frac{2}{3}$ $\frac{280-160-(40+50)}{4-2}=15$ $\frac{280-160-(40+50+60)}{4-3}<0$	30	30	
			$(O_{21},O_{22},O_{23},$ $O_{12},O_{13})$: $l^p(\nu)=235>165$	$p^{\nu\mu}_{(0)}=0$ $p^{\nu\mu}_{(1)}=40$ $p^{\nu\mu}_{(2)}=50$ $p^{\nu\mu}_{(3)}=60$	$\frac{235-160-0}{4-0}=18.75$ $\frac{235-160-40}{4-1}=11\frac{2}{3}$ $\frac{235-160-(40+50)}{4-2}=7.5$ $\frac{235-160-(40+50+60)}{4-3}<0$	18.75		

In other words, digraph G_1 provides a solution of problem $\mathcal{J}2/n=3, a_i\leq p_i\leq b_i/\mathcal{C}_{max}$ as long as inequality (2.2) is satisfied:

$$\Lambda^*(G) = \{G_1\}.$$

In this case, a decision-maker needs to use only one digraph G_1 from the set $\Lambda(G) = \{G_1, G_2, G_3, G_4, G_5\}$ (see Figure 2.2) in any possible realization and so the solution of problem $\mathcal{J}2/n=3, a_i\leq p_i\leq b_i/\mathcal{C}_{max}$ turns out to be the same as for problem $\mathcal{J}2/n=3/\mathcal{C}_{max}$ with the fixed vector of the processing times $p \in T$. In this case, the minimal solution consists of one digraph: $\{G_1\} = \Lambda^T(G) \subseteq \Lambda^*(G)$.

Otherwise (i.e. if inequality (2.2) does not hold), the optimality of digraph G_1 is not guaranteed within the given polytope T: There exists another feasible digraph G_k, $k \neq 1$, (we call it a *competitive* digraph for G_1) with a critical weight being smaller than that of digraph G_1 in some realization of the process. If such a 'superiority' of the competitive digraph G_k occurs when the processing times are equal to $p^* = (p^*_{11}, p^*_{12}, \ldots, p^*_{23}) \in T$ (i.e. digraph G_k instead of G_1 is optimal for the vector p^*), one can calculate the stability radius $\widehat{\varrho}_k(p^*)$ of the digraph G_k at the new vector p^* of the processing times. In the case when $\widehat{\varrho}_k(p^*)$ is strictly positive, one can consider the union $O_{30}(p) \cup O_{\widehat{\varrho}_k(p^*)}(p^*)$ of the two balls. If the inclusion $T \subseteq O_{30}(p) \cup O_{\widehat{\varrho}_k(p^*)}(p^*)$ holds, problem $\mathcal{J}2/n{=}3, a_i{\le}p_i{\le}b_i/\mathcal{C}_{max}$ is solved. In such a case, a decision-maker needs to use either digraph G_1 or digraph G_k for a practical realization of an optimal schedule:

$$\Lambda^*(G) = \{G_1, G_k\}.$$

Otherwise, we have to calculate the stability radius of a competitive digraph of digraph G_k at a new vector of the processing times.

Continuing in this manner, we may cover the given polytope T by the union of the stability balls of some feasible digraphs. As a result, for any vector of the processing times from the polytope T (i.e. whenever inequalities (1) hold), we have at least one optimal schedule.

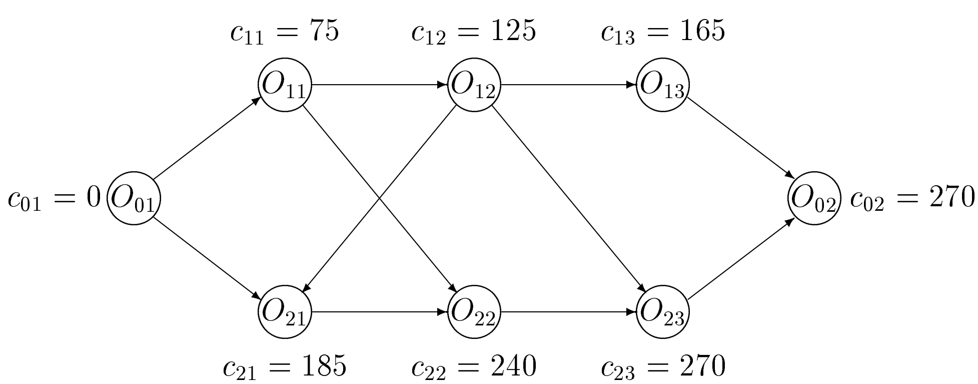


Figure 2.4: *Competitive digraph* $G_3 = (Q, A \cup E_3, \emptyset)$ *for digraph* $G_1 = (Q, A \cup E_1, \emptyset)$ *which is optimal for* $p = (75, 50, 40, 60, 55, 30)$

For Example 2.1 with the original vector $p = (75, 50, 40, 60, 55, 30)$, competitive digraphs for the optimal digraph G_1 are the digraphs $G_3 = (Q, A \cup E_3, \emptyset)$ and $G_5 = (Q, A \cup E_5, \emptyset)$, where $E_3 = \{(O_{11}, O_{22}), (O_{12}, O_{21}), (O_{12}, O_{23})\}$ and $E_5 = \{(O_{22}, O_{11}), (O_{21}, O_{12}), (O_{23}, O_{12})\}$. Digraph G_3 with the completion times of the operations Q is presented in Figure 2.4. As the calculation of the stability radius shows, at the boundary of the ball $O_{30}(p)$ (namely, at the point $p^* = (p^*_{11}, p^*_{12}, \ldots, p^*_{23}) = (45, 80, 70, 90, 25, 0) \in R^6_+$) both digraphs G_1 and G_3 are optimal. Note that vector p^* is determined during the calculation of the stability radius on the basis of the formulas (1.9) and (1.10). Specifically, vector p^* is obtained from vector p by decreasing the processing times of the operations O_{11}, O_{22}, and O_{23} by the value $\widehat{\varrho}_1(p)$, where $p^*_{11} = 75 - 30 = 45$, $p^*_{22} = 55 - 30 = 25$, and $p^*_{23} = 30 - 30 = 0$, and by increasing the processing times of the operations O_{12}, O_{13}, O_{21} by the same value $\widehat{\varrho}_1(p)$, where

$p^*_{12} = 50+30 = 80$, $p^*_{13} = 40+30 = 70$, $p^*_{21} = 60+30 = 90$, i.e. according to formula (2.3):

$$p^*_i = \begin{cases} p_i + r, & \text{if } i \in [\mu], \\ \max\{0, p_i - r\}, & \text{if } i \in [\nu]\backslash[\mu], \\ p_i, & \text{if } i \notin [\mu] \cup [\nu], \mu \in H_{sk}, \nu \in H_k, \end{cases} \tag{2.3}$$

where $[\mu] = \{O_{12}, O_{13}, O_{21}\}, \mu \in H_{13}$, $[\nu] = \{O_{11}, O_{12}, O_{21}, O_{22}, O_{23}\}, \nu \in H_3$, and $r = \widehat{\varrho}_1(p) = 30$. Due to such changes in the processing times, the critical weight of digraph G_1 is increased from 165 to 240 and the critical weight of digraph G_3 is decreased from 270 to 240.

From [Sot91] it follows that the existence of two or more optimal digraphs is a necessary condition (but not a sufficient one) for the stability radius to be equal to zero (see Theorem 1.1). Nevertheless, the 'unstability' of an optimal digraph happens at the boundary of a *stability region* (the stability region of the digraph G_s is the whole set of the vectors $p \in R^q_+$ with the schedule s being optimal [Sot91]), where at least two optimal digraphs exist. Such a situation occurs for Example 2.1, namely: $\widehat{\varrho}_1(p^*) = \widehat{\varrho}_3(p^*) = 0$. According to Theorem 1.1, there exists a path $\mu^* \in H_1(p)$, $[\mu^*] = \{O_{12}, O_{13}, O_{21}\}$, such that there does not exist any path $\nu \in H_3(p)$ with $[\mu^*] \subseteq [\nu]$. On the other hand, there exists a path $\nu^* \in H_3(p)$, $[\nu^*] = \{O_{11}, O_{12}, O_{21}, O_{22}, O_{23}\}$, such that there does not exist any path $\mu \in H_1(p)$ with $[\nu^*] \subseteq [\mu]$. Note also that for the point p^*, the only competitive digraph for digraph G_3 is digraph G_1 (and vice versa), where the stability radius of G_1 for the original point $p \in R^q_+$ has been already calculated.

Considering the competitive digraph G_5 instead of the competitive digraph G_3 gives zero stability radii for both digraphs G_1 and G_5 at the corresponding vector $p' = (105, 20, 10, 30, 85, 60)$ of processing times, constructed due to (2.3) with $r = \widehat{\varrho}_1(p) = 30$ for the paths $[\mu'] = \{O_{11}, O_{22}, O_{23}\}, \mu' \in H_{15}$, and $[\nu'] = \{O_{11}, O_{12}, O_{13}, O_{21}, O_{22}\}, \nu' \in H_5$.

From the above discussion it follows that another type of the stability radius is required for solving problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$. While $\widehat{\varrho}_s(p)$ denotes the largest radius of a ball $O_{\widehat{\varrho}}(p)$ within which digraph G_s is 'the best' for the whole set $\Lambda(G)$ (see Definition 1.1), we need to determine the largest ball within which digraph G_s is 'the best' for some subset B of the set of feasible digraphs $\Lambda(G)$. Indeed, for Example 2.1 we need to calculate the largest radius of the ball within which digraph G_3 has the minimal critical weight among the feasible digraphs $\Lambda(G)$ except digraph G_1, which is optimal within the ball $O_{\widehat{\varrho}_1(p)}(p)$ and which is already contained in the set of candidates for a practical realization: $G_1 \in \Lambda^*(G)$. Thus, in this case we need to consider the set $B = \Lambda(G)\backslash\{G_1\}$.

In Section 2.2, we propose a new definition of the stability radius. Note also that the given bounds a_i and b_i for possible variations of the processing time x_i, $i \in Q$, may enlarge the stability ball of the optimal digraph G_s. E.g. this is true for Example 2.1 since inequality (2.2) becomes only a sufficient condition for the optimality of digraph G_1 (but not a necessary one). In Section 2.2, we provide both necessary and sufficient conditions for a zero (and for an infinitely large) stability radius. In Section 2.3, the formulas (1.9) and (1.10) from [Sot91] given for the case of calculating the stability radius with $0 \leq p_i < \infty$, $i \in Q$, are generalized to the case when the variations of the processing times are given by inequalities (1) and some feasible digraphs have to be excluded from the comparisons with 'the best' one.

2.2 Relative Stability Radius

In [Sot91, SSW97], the stability radius $\widehat{\varrho}_s(p)$ of an optimal digraph has been investigated which denotes the largest quantity of independent variations within the interval $[0,\infty)$ of the processing times p_i of the operations $i \in Q$ such that 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 1.1). For solving problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$, we need a more general notion of a stability radius since the processing time of operation $i \in Q$ falls within the given closed interval $[a_i, b_i]$, $0 \leq a_i \leq b_i$, and competitive digraphs only belong to some subset B of the set $\Lambda(G)$. The following generalization of the stability radius (we call it *relative* stability radius) is defined by considering the closed interval $[a_i, b_i]$ instead of $[0,\infty)$ and by considering the set $B \subseteq \Lambda(G)$ instead of the whole set $\Lambda(G)$. In the following Definition 2.2, l_s^p is the critical weight of digraph $G_s \in \Lambda(G)$ at the vector $p \in T$, defined at page 15.

Definition 2.2 *Assume that for each vector $p' \in O_\varrho(p) \cap T$ digraph $G_s \in B \subseteq \Lambda(G)$ with the vector p' of weights has the minimal critical weight $l_s^{p'}$ among all digraphs of the set 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 the polytope T for criterion $\mathcal{C}_{max}$.*

Note that the relativity of the stability radius in Definition 2.2 is not only considered with respect to the polytope T of the feasible vector of the processing times, but also with respect to the set B of semiactive schedules. However, to avoid a too complicated notion, we omit here and in the following the phrase "with respect to the set B".

From Definition 1.1 and Definition 2.2, if follows that $\widehat{\varrho}_s(p) = \widehat{\varrho}_s^{\Lambda(G)}(p \in R_+^q)$. The relative stability radius is equal to the *maximal error* of the given processing times p_i $(a_i \leq p_i \leq b_i,\ i \in Q)$ within which the 'superiority' of digraph G_s is still preserved over the given set B of feasible digraphs. The following two extreme cases of such an error are of particular importance for problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$. On the one hand, if for any positive real $\epsilon > 0$ 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 have 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 have an infinitely large relative stability radius:

$$\widehat{\varrho}_s^B(p \in T) = \infty.$$

Even if in the case of $b_i < \infty$ the maximal error of p_i for each $i \in Q$ is restricted by

$$\epsilon_{max} = \max\{\{p_i - a_i, b_i - p_i\} \ : \ i \in Q\}, \tag{2.4}$$

it is still possible that $\widehat{\varrho}_s^B(p \in T)$ is infinitely large as implied by Definition 2.2. E.g. the deterministic problem $\mathcal{G}//\mathcal{C}_{max}$ is such a trivial example with an infinitely large relative stability radius of the optimal digraph G_s. Indeed, if $a_i = p_i = b_i$ for each $i \in Q$, then the polytope T degenerates into a single point: $T = \{p\}$, and so from the inclusion $p' \in O_\varrho(p) \cap T$ it follows that vector p' is equal to vector p, for which 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 introduced in [STW98] (see Section 1.2) and which is an improved formulation of the dominance relations given in the paper [LSSW97].

Definition 2.3 *Path ν dominates path μ in the polytope T if and only if for any vector $x = (x_1, x_2, \ldots, x_q) \in T$ the following inequality holds:*

$$l^x(\mu) \leq l^x(\nu). \tag{2.5}$$

The binary relation introduced in Definition 2.3 is an extension of the dominance relation introduced in Section 1.2 in the sense that path ν dominates path μ in any polytope $T \subseteq R_+^q$ if path ν dominates path μ (according to Definition 1.2). Indeed, if $[\mu] \subset [\nu]$, then the inequality $l^x(\mu) \leq l^x(\nu)$ holds for any vector $x \in R_+^q$. Note also that both dominance relations coincide at least when $a_i = 0$ and $b_i = \infty$ for each $i \in Q$ (it is easy to see that inclusion $[\mu] \subset [\nu]$ holds if and only if inequality (2.5) holds for $a_i = 0$ and $b_i = \infty,\ i \in Q$). Moreover, in this case equality $l^x(\mu) = l^x(\nu)$ is achieved only if $x_i = a_i = 0$ for any operation $i \in [\nu] \setminus [\mu]$.

Thus, we conclude that the *dominance relation introduced in Definition 1.2 is a special case of the dominance relation defined by the inequality (2.5) when T is equal to the space $R_+^q\ :\ a_i = 0$ and $b_i = \infty$ for each $i \in Q$*. Hence, the phrase “path ν dominates path μ” is identical to the phrase “path ν dominates path μ in R_+^q”.

The following lemma gives a simple criterion for the dominance relation defined by inequality (2.5) in Definition 2.3.

Lemma 2.1 *Path ν dominates path μ in the polytope T if and only if inequality (2.6) holds:*

$$\sum_{i \in [\mu] \setminus [\nu]} b_i \leq \sum_{j \in [\nu] \setminus [\mu]} a_j. \tag{2.6}$$

PROOF. By subtracting all common variables from the left- and right-hand sides of the inequality (2.5) and taking into account that $a_i \leq b_i$ for each $i \in Q$, we obtain that inequality (2.5) is equivalent to the following inequality:

$$\sum_{i \in [\mu] \setminus [\nu]} x_i \leq \sum_{j \in [\nu] \setminus [\mu]} x_j \ \text{ for any } x_i \ \text{ with } \ a_i \leq x_i \leq b_i,\ i \in [\nu] \cup [\mu]. \tag{2.7}$$

It is easy to see that any vector $x \in T$ satisfies the inequality (2.7) if and only if inequality (2.6) holds:

$$\sum_{i \in [\mu] \setminus [\nu]} a_i \leq \sum_{i \in [\mu] \setminus [\nu]} b_i \leq \sum_{j \in [\nu] \setminus [\mu]} a_j \leq \sum_{j \in [\nu] \setminus [\mu]} b_j.$$

$\diamond$

On the basis of the above path domination, we introduced in [LSSW97] a domination of the sets of paths.

Definition 2.4 *The set of paths H_k dominates the set of paths H_s in the polytope T if and only if for any path $\mu \in H_s$, there exists a path $\nu \in H_k$, which dominates path μ in the polytope T.*

The following statement gives a simple sufficient condition, when the domination of sets of paths does not hold (the idea of the proof was taken from [LSSW97]).

Corollary 2.1 *The set of paths H_k does not dominate the set of path H_s with respect to the polytope T if there exists a path $\mu \in H_s$ such that system*

$$\begin{cases} \sum_{i \in [\nu] \setminus [\mu]} a_i < \sum_{j \in [\mu] \setminus [\nu]} b_j, \\ a_i \leq x_i \leq b_i, \ i \in Q, \end{cases} \tag{2.8}$$

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

PROOF. From Definition 2.4 it follows that the set of paths H_k does not dominate the set of paths H_s in the polytope T if there exists a path $\mu^* \in H_s$ such that there is no path $\nu \in H_k$ which dominates path μ^* in the polytope T. This means that inequality (2.5) is violated for the path $\mu^* \in H_s$ for some vector $x^0 \in T$, i.e. the system

$$\begin{cases} l^x(\nu) < l^x(\mu), \\ a_i \leq x_i \leq b_i, \ i \in Q, \end{cases} \tag{2.9}$$

has a solution for any path $\nu \in H_k$. Furthermore, system (2.9) is compatible if and only if it has the following solution:

$$x_i = x_i^0 = \begin{cases} a_i, & \text{if } i \in [\mu^*] \setminus [\nu], \\ b_i, & \text{if } i \in [\nu] \setminus [\mu^*]. \end{cases} \tag{2.10}$$

It is easy to see that vector x according to (2.10) is a solution of system (2.9) if and only if condition (2.6) does not hold for any vertex $i \in [\nu] \cup [\mu^*]$. In other words, the vector $x^0 = (x_1^0, x_2^0, \ldots, x_q^0) \in T$ and the path $\mu^* \in T$ are a solution of the equivalent system (2.8), too.

◇

Obviously, 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 which was proven in [STW98, LSSW97] shows that in general the set of the critical paths is not expanded for small variations of the processing times.

Lemma 2.2 *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 $\epsilon_k > \epsilon > 0$ defined as follows:*

$$\epsilon_k = \frac{1}{q}\Big(l_k^p - \max\{l^p(\nu) \ : \ \nu \in H_k \setminus H_k(p)\}\Big). \tag{2.11}$$

Next, we present a generalization of the necessary and sufficient conditions for a zero stability radius (see Theorem 1.1) and an infinitely large stability radius (see Theorem 1.2) to the case of a zero relative stability radius and an infinitely large relative stability radius, respectively.

Theorem 2.1 *For digraph G_s, which has the minimal critical weight l_s^p, $p \in T$, within the set $B \subseteq \Lambda(G)$ of feasible digraphs, the equality $\widehat{\varrho}_s^B(p \in T) = 0$ holds if and only if there exists a digraph $G_k \in B$ such that $l_s^p = l_k^p$, $k \neq s$, and the set of paths $H_k(p)$ does not dominate the set of paths $H_s(p)$ in the polytope T.*

PROOF. *Sufficiency (if).* Let the conditions of Theorem 2.1 be satisfied: There exists a digraph $G_k \in B$ such that $l^p_s = l^p_k,\ k \neq s$, and $H_k(p)$ does not dominate the set $H_s(p)$ in T. We show that $\widehat{\varrho}^B_s(p \in T) < \epsilon$ for any given $\epsilon > 0$ which may be as small as desired.

Since the set $H_k(p)$ does not dominate the set $H_s(p)$ in the polytope T, there exists a path $\mu^* \in H_s(p)$ such that no path $\nu \in H_k(p)$ dominates path μ^* in the polytope T, i.e. system (2.9) has a solution for any path $\nu \in H_k(p)$. First, we make the following remark.

Remark 2.1 From the compatibility of (2.9), it follows that for the considered problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$, the trivial case with $a_i = b_i$ for each $i \in Q$ does not hold, since in this case the first inequality in (2.9) is transformed into inequality $l^p(\nu) < l^p(\mu^*)$ which is wrong: $l^p(\nu) = l^p_k = l^p_s = l^p(\mu^*)$.

We construct a vector $p' = (p'_1, p'_2, \dots, p'_q)$ with the following components:

$$p'_i = \begin{cases} p_i + \epsilon', & \text{if } i \in [\mu^*],\ p_i \neq b_i, \\ p_i - \epsilon', & \text{if } i \in \{\cup_{\nu \in H_k(p)}[\nu]\}\backslash[\mu^*],\ p_i \neq a_i, \\ p_i, & \text{otherwise}, \end{cases} \tag{2.12}$$

where ϵ' is chosen as a strictly positive real number less than both value ϵ and value

$$\epsilon_{min} = \max\{0,\ \min\{\min\{p_i - a_i :\ p_i > a_i, i \in Q\},\ \min\{b_i - p_i :\ b_i > p_i, i \in Q\}\}\}.$$

We can also choose ϵ' less than $\epsilon_k > 0$ defined in (2.11). More precisely, if $H_k \neq H_k(p)$, then $\epsilon_k > 0$, and we can choose ϵ' such that $0 < \epsilon' < \min\{\epsilon, \epsilon_k, \epsilon_{min}\}$. Otherwise, if $H_k = H_k(p)$, we choose ϵ' such that $0 < \epsilon' < \min\{\epsilon, \epsilon_{min}\}$. Such choices are possible since in both above cases, inequality $\epsilon_{min} > 0$ holds due to the Remark 2.1. The following arguments are the same for both cases of the choice of ϵ' except the 'last step' since $H_k \backslash H_k(p) = \emptyset$ in the latter case.

Since system (2.9) has a solution for each path $\nu \in H_k$, the first inequality in (2.9)

$$l^x(\nu) < l^x(\mu^*)$$

has a solution for $x \in T$ which implies that inclusion $[\mu^*] \subset [\nu]$ does not hold for any path $\nu \in H_k(p)$. Therefore, from the equalities $l^p(\nu) = l^p_k = l^p_s = l^p(\mu^*)$ and (2.12), we can conclude that vector p' is a solution of system (2.9) for each path $\nu \in H_k(p)$. In other words, vector p' is a solution of the following system of inequalities:

$$\begin{cases} l^x(\nu) < l^x(\mu^*),\ \nu \in H_k(p), \\ a_i \leq x_i \leq b_i,\ i \in Q. \end{cases}$$

Thus, we have $l^{p'}(\nu) < l^{p'}(\mu^*)$ for each $\nu \in H_k(p)$, and therefore

$$\max\{l^{p'}(\nu)\ :\ \nu \in H_k(p)\} < l^{p'}(\mu^*). \tag{2.13}$$

The 'last step' in the proof of sufficiency is as follows. Since $p' \in O_{\epsilon'}(p) \cap R^q_+$ with $0 < \epsilon' < \epsilon_k$, due to Lemma 2.2 we have $H_k(p') \subseteq H_k(p)$ and, as a result,

$$l^{p'}(\tau) < l^{p'}_k = \max\{l^{p'}(\nu)\ :\ \nu \in H_k(p)\} \tag{2.14}$$

for each path $\tau \in H_k \backslash H_k(p)$. From inequalities (2.13) and (2.14), it follows that $l^{p'}_k < l^{p'}_s$. Taking into account that $d(p', p) = \epsilon' < \epsilon$, we conclude that $\widehat{\varrho}^B_s(p \in T) < \epsilon$.

Necessary (only if). We prove necessity by contradiction. Let us suppose that $\widehat{\varrho}^B_s(p \in T) = 0$ but the conditions of Theorem 2.1 do not hold. The following cases i and ii of violating these conditions may hold.

i) There does not exist a digraph $G_k \in B$ such that $l^p_s = l^p_k,\ k \neq s$.

In the trivial case when $B = \{G_s\}$, we have $\widehat{\varrho}^B_s(p \in T) = \infty$ due to Definition 2.2.

If $B \backslash \{G_s\} \neq \emptyset$, we can calculate the following real number:

$$\epsilon^* = \frac{1}{q} \min\{l^p_t - l^p_s \ :\ G_t \in B,\ t \neq s\} \tag{2.15}$$

which is strictly positive since $l^p_s < l^p_t$ for each $G_t \in B,\ t \neq s$. Next, we show that the difference $l^p_t - l^p_s$ cannot become negative when vector p is replaced by an arbitrary vector $p^0 \in O_{\epsilon^*}(p) \cap T \subseteq R^q_+$ with $0 < \epsilon^* < \epsilon_k$.

From (2.15) it follows that $l^{p^0}_k - l^{p^0}_s \geq q \cdot \epsilon^*$, and therefore, to make the difference $l^{p^0}_k - l^{p^0}_s$ equal to zero, one need a vector p' with a distance from the vector p greater than or equal to $\epsilon_k :\ d(p^0, p') \geq \epsilon_k$. But due to the conditions of Lemma 2.2, we have $d(p, p') \leq \epsilon^* < \epsilon_k$. Since for any digraph $G_t \in B$, the difference $l^{p^0}_k - l^{p^0}_s$ is still greater than the product $q \cdot \epsilon^*$, we conclude that digraph G_s remains 'the best' (perhaps one of the 'best') within the set B for any vector p^0 of the processing times. Due to Definition 2.2, we have $\widehat{\varrho}^B_s(p \in T) \geq \epsilon^* > 0$ which contradicts the above assumption of $\widehat{\varrho}^B_s(p \in T) = 0$.

ii) There exists a digraph $G_k \in B$ such that $l^p_s = l^p_k,\ k \neq s$, and for any such digraph G_k, the set of paths $H_k(p)$ dominates the set of paths $H_s(p)$ in the polytope T.

In this case we can take any ϵ that satisfies the following inequalities:

$$0 < \epsilon < \min\Big\{ \min\{\epsilon_k \ :\ l^p_k = l^p_s,\ G_k \in B\},\ \frac{1}{q} \min\{l^p_t - l^p_s \ :\ l^p_t > l^p_s,\ G_t \in B\}\Big\}.$$

Due to inequality $\epsilon > \epsilon_s$, we get from Lemma 2.2 the equalities:

$$l^{p^0}_s = \max_{\mu \in H_s(p^0)} l^{p^0}(\mu) = \max_{\mu \in H_s(p)} l^{p^0}(\mu) \tag{2.16}$$

for any vector $p^0 \in O_\epsilon(p) \cap R^q_+$. The statement that for any digraph $G_k \in B, k \neq s$, with $l^p_s = l^p_k$ the set of paths $H_k(p)$ dominates the set of paths $H_s(p)$ in the polytope T means that for any path $\mu \in H_s(p)$, there exists a path $\nu^* \in H_k(p)$ such that inequality

$$l^x(\mu) \leq l^x(\nu^*) \tag{2.17}$$

holds for any vector $x \in T$. Due to inequality (2.17) and taking into account that $\epsilon < \epsilon_k$ and $\epsilon < \epsilon_s$, we obtain the following inequality using Lemma 2.2:

$$\max_{\mu \in H_s(p)} l^{p^0}(\mu) \leq \max_{\nu \in H_k(p)} l^{p^0}(\nu). \tag{2.18}$$

Thus, due to (2.16) and (2.18), we have

$$l^{p^0}_s \leq \max_{\nu \in H_k(p)} l^{p^0}(\nu) \tag{2.19}$$

for any digraph $G_k \in B,\ l^p_s = l^p_k,\ k \neq s$. Since

$$\epsilon < \frac{1}{q} \min\{l^p_t - l^p_s :\ l^p_t > l^p_s,\ G_t \in B\},$$

inequality $l_t^p > l_s^p$ implies inequality $l_t^{p^0} > l_s^{p^0}$. Taking into account (2.19), we conclude that $l_s^{p^0} \leq l_k^{p^0}$ for any digraph $G_k \in B$ and for any vector $p^0 \in T$ with $d(p, p^0) \leq \epsilon$. Consequently, $\widehat{\varrho}_s^B(p \in T) \geq \epsilon > 0$, which contradicts the assumption of $\widehat{\varrho}_s^B(p \in T) = 0$.
$\diamond$

Theorem 2.1 directly implies the following statements.

Corollary 2.2 *If $G_s \in B$ is a unique optimal schedule for the vector $p \in T$, then $\widehat{\varrho}_s^B(p \in T) > 0$.*

Corollary 2.3 *If $G_s \in B$ and $l_s^p = \min\{l_k^p \ : \ G_k \in B\}$, then $\widehat{\varrho}_s^B(p \in T) \geq \epsilon^*$ with ϵ^* calculated according to (2.15).*

The proof of such a lower bound for $\widehat{\varrho}_s^B(p \in T)$ can be found in [LSSW97].

Theorem 2.1 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 digraph G_s.

Theorem 2.2 *For digraph $G_s \in B$, we have $\widehat{\varrho}_s^B(p \in T) = \infty$ 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 $H_s \backslash H$ in the polytope T.*

PROOF. *Sufficiency.* If ϱ is a positive number as large as desired, we take any vector $p \in O_\varrho(p) \cap T \subseteq R_+^q$ and consider a path $\mu \in H_s$ such that $l_s^p = l^p(\mu)$.

j) If $\mu \in H$, then inequality $l_s^p = l^p(\mu) \leq l_t^p$ holds for any digraph $G_t \in \Lambda(G)$.

jj) If $\mu \in H_s \backslash H$, then due to the condition of Theorem 2.2, it follows that for any digraph $G_t \in B, t \neq s$, there exists a path $\nu^* \in H_t$ such that the inequality

$$l^x(\mu) \leq l^x(\nu^*)$$

holds for any vector $x \in T$ (and for the vector p, too). Therefore, we have $l_s^p = l^p(\mu) < l^p(\nu^*) \leq l_t^p$. Thus, in both above cases j and jj we have $l_s^p = \min\{l_t^p \ : \ G_t \in B\}$.

Necessity. We prove necessity by contradiction. Let us suppose that $\widehat{\varrho}_s^B(p \in T) = \infty$, but there exists a digraph $G_t \in B, t \neq s$, such that the set of paths H_t does not dominate the set of paths $H_s \backslash H$ in the polytope T. Thus, there exists a path $\mu^0 \in H_s \backslash \ H$ such that for any path $\nu \in H_t$, the system

$$\begin{cases} l^x(\nu) < l^x(\mu^0), \\ a_i \leq x_i \leq b_i, \ i \in Q, \end{cases} \tag{2.20}$$

has a solution. Therefore, due to Corollary 2.1, the inequality

$$\sum_{i \in [\nu] \backslash [\mu^0]} a_i < \sum_{j \in [\mu^0] \backslash [\nu]} b_j \tag{2.21}$$

holds. We consider the vector $p^* = (p_1^*, p_2^*, \ldots, p_q^*) \in T$ with

$$p_i^* = \begin{cases} a_i, & \text{if } i \in \{\cup_{[\nu] \in H_t}[\nu]\} \setminus [\mu^0], \\ b_i, & \text{if } i \in [\mu^0], \\ p_i & \text{otherwise.} \end{cases}$$

Adding to the left-hand side and to the right-hand side of (2.21) the value $\sum_{j\in[\nu]\cap[\mu^0]} b_j$, we obtain that inequality

$$\sum_{i\in[\nu]\setminus[\mu^0]} a_i + \sum_{j\in[\nu]\cap[\mu^0]} b_j < \sum_{j\in[\mu^0]} b_j$$

holds. Thus, we can conclude that vector p^* is a solution of the system of linear inequalities obtained by joining systems (2.20) for all paths $\nu \in H_t$, i.e. we have

$$\begin{cases} l^{p^*}(\nu) < l^{p^*}(\mu^0), \ \nu \in H_t, \\ a_i \le x_i \le b_i, \ i \in Q. \end{cases}$$

Therefore, $l_t^{p^*} < l^{p^*}(\mu^0) \le l_s^{p^*}$, and hence, we get a contradiction to the above assumption: $\widehat{\varrho}_s^B(p \in T) < d(p^*, p) \le \epsilon_{max} < \infty$.

◇

From Theorem 2.2 we obtain the following upper bound for the relative stability radius.

Corollary 2.4 *If $\widehat{\varrho}_s^B(p \in T) < \infty$, then $\widehat{\varrho}_s^B(p \in T) \le \epsilon_{max}$, where value ϵ_{max} is calculated according to (2.4).*

PROOF. The desired bound immediately follows from the proof of necessity in Theorem 2.2.

◇

In the following section, we use Theorem 2.2 as a stopping rule in the algorithm developed for solving problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$ since the optimality of digraph $G_s \in B$ with $\widehat{\varrho}_s^B(p \in T) = \infty$ does not depend on the vector $p \in T$ of the processing times.

2.3 Algorithms for Problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$

From Sections 2.1 and 2.2, it follows that problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$ may be solved on the basis of a repeated calculation of the relative stability radii $\widehat{\varrho}_s^B(p \in T)$. The formulas for calculating $\widehat{\varrho}_s(p) = \widehat{\varrho}_s^{\Lambda(G)}(p \in R_+^q)$ were given in [Sot91] and discussed in Section 1.3. Theorem 2.3, which follows, generalizes these formulas for any given set $B \subseteq \Lambda(G)$ and for any given polytope $T \subseteq R_+^q$. To present the new formula, we need the following notations.

Let μ and ν be paths in the digraphs from the set $\Lambda(G)$. We denote the 'symmetric difference' $[\mu] \cup [\nu] \setminus [\mu] \cap [\nu]$ of the sets $[\mu]$ and $[\nu]$ by $[\mu] + [\nu]$ and calculate the following values:

$$\Delta^i(\mu, \nu) = \begin{cases} b_i - p_i, & \text{if } \ i \in [\mu] \setminus [\nu], \\ p_i - a_i, & \text{if } \ i \in [\nu] \setminus [\mu]. \end{cases} \tag{2.22}$$

Let $\Delta_0^i(\mu, \nu)$ be equal to zero. We order the set of values $\Delta^i(\mu, \nu)$ for all operations i from the set $[\mu] + [\nu]$ in the following way:

$$\Delta_1^{i_1}(\mu, \nu) \le \Delta_2^{i_2}(\mu, \nu) \le \ldots \le \Delta_{|[\mu]+[\nu]|}^{i_{|[\mu]+[\nu]|}}(\mu, \nu), \tag{2.23}$$

where the subscript $j \in \{1, 2, \ldots, [\mu] + [\nu]\}$ indicates the location of $\Delta^i(\mu, \nu)$ in the above inequalities, and the superscript i_j denotes operation $i_j \in [\mu] + [\nu]$ for which the value $\Delta_j^{i_j}(\mu, \nu)$ was calculated (in the following formulas, we will reduce the superscript to one letter i (operation i), and we hope it will not cause a misunderstanding). For any two feasible digraphs G_s and G_k, let

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

Theorem 2.3 *Given digraph G_s with the minimal critical weight l_s^p, $p \in T$, within the set $B \subseteq \Lambda(G)$ of feasible digraphs, we have*

$$\widehat{\varrho}_s^B(p \in T) = \min_{G_k \in B} \widehat{r}_{ks}^B, \tag{2.24}$$

where

$$\widehat{r}_{ks}^B = \min_{\mu \in H_{sk}(T)} \quad \max_{\nu \in H_k,\ l^p(\nu) \geq l_s^p} \quad \max_{\beta=0,1,\ldots,|[\mu]+[\nu]|-1} \frac{l^p(\nu) - l^p(\mu) - \sum_{\alpha=0}^{\beta} \Delta_\alpha^i(\mu,\nu)}{|[\mu]+[\nu]| - \beta}. \tag{2.25}$$

PROOF. From Definition 2.2 it follows that

$$\widehat{\varrho}_s^B(p \in T) = \inf\{d(p,x) :\ x \in T,\ l_s^x > \min\{l_k^x :\ G_k \in B\}\}.$$

Therefore, to find the relative stability radius $\widehat{\varrho}_s^B(p \in T)$, it is sufficient to construct a vector $x \in T$ which satisfies the following three conditions.
1) There exists a digraph $G_k(p) \in B,\ k \neq s$, such that $l_s^x = l_k^x$, i.e.

$$\max_{\mu \in H_s} l^x(\mu) = \max_{\nu \in H_k} l^x(\nu). \tag{2.26}$$

2) For any given real $\epsilon > 0$, which may be as small as desired, there exists a vector $p^\epsilon \in T$ such that $d(x, p^\epsilon) = \epsilon$ and $l_s^{p^\epsilon} > l_k^{p^\epsilon}$, i.e. inequality

$$\max_{\mu \in H_s} l^{p^\epsilon}(\mu) > \max_{\nu \in H_k} l^{p^\epsilon}(\nu) \tag{2.27}$$

is satisfied for at least one digraph $G_k(p) \in B$.
3) The distance $d(p,x)$ achieves the minimal value among the distances between the vector p and the other vectors in the polytope T which satisfy both above conditions 1 and 2.

After having constructed such a vector $x \in T$, one can define the relative stability radius of the digraph G_s:

$$\widehat{\varrho}_s^B(p \in T) = d(p,x),$$

since the critical path of digraph G_s becomes larger than that of digraph G_k for any vector $p^\epsilon \in T$ with positive real ϵ, which may be as small as desired (see condition 2), and so digraph G_s has no longer the minimal critical weight among all other feasible digraphs, while in the ball $O_{d(p,x)}(p \in T)$ digraph G_s has the minimal critical weight (see condition 3). Digraph G_k satisfying conditions 1, 2 and 3 is a *competitive* digraph for the optimal digraph G_s.

To satisfy conditions 1, 2 and 3 (except the inclusion $x \in T$), we first search for a vector $x = p(r) = (p_1(r), p_2(r), \ldots, p_q(r)) \in R^q$ with the components $p_i(r) \in \{p_i, p_i + r, p_i - r\}$ on the basis of a direct comparison of the paths from the set H_s and the paths from the sets H_k, where $G_k \in B$.

Let the value $l^p(\nu)$ be greater than the weight of a critical path in an optimal digraph G_s. To satisfy equality (2.26), the weight of a path $\nu \in H_k$ must be smaller than or equal to the weight of at least one path $\mu \in H_s$, and there must exist a path $\nu \in H_k$ with a weight equal to the weight of a critical path of G_s. Thus, if we have calculated

$$r_\nu = \min_{\mu \in H_s} \frac{l^p(\nu) - l^p(\mu)}{|[\mu]+[\nu]|}, \tag{2.28}$$

we obtain the equality

$$\max_{\mu \in H_s} l^{p(r)}(\mu) = l^{p(r)}(\nu) \tag{2.29}$$

for the vector $p(r) = p(r_\nu)$ with the components

$$p_i(r) = p_i(r_\nu) = \begin{cases} p_i + r_\nu, & \text{if } i \in [\mu], \\ p_i - r_\nu, & \text{if } i \in [\nu] \backslash [\mu], \\ p_i, & \text{if } i \notin [\mu] + [\nu]. \end{cases} \tag{2.30}$$

We can make the following remark.

Remark 2.2 Due to (2.28), the vector $p(r)$ calculated in (2.30) is the closest one to the given vector p among all vectors x for which equality (2.29) with $p(r) = x$ holds. Indeed, to make the difference $l^x(\nu) - \max_{\mu \in H_s} l^x(\mu)$ equal to zero, one needs a q-dimensional vector x with a distance from the vector p greater than or equal to r_ν : $d(p, x) \geq r_\nu$.

To reach equality (2.26) for the whole digraph G_k, we have to repeat the calculation (2.28) for each path $\nu \in H_k$ with $l^p(\nu) \geq l^p_s$. Thus, instead of the vector $p(r_\nu)$, we have to consider the vector $p(r) = p(r_{G_k})$ calculated according to formula (2.30), where

$$r_{G_k} = \min_{\mu \in H_s} \quad \max_{\nu \in H_k;\ l^p(\nu) \geq l^p_s} \frac{l^p(\nu) - l^p(\mu)}{|[\mu] + [\nu]|}. \tag{2.31}$$

Next, we consider inequality (2.27). Since the processing times have to belong to the polytope $T \subseteq R^q_+$, this inequality may not be valid for a vector $p^\epsilon \in T$ if path ν dominates path μ in the polytope T. Thus, we can restrict our consideration to the subset $H_{sk}(T)$ of the set H_s of all paths, which are not dominated by paths from the set H_k in the polytope T and for which there does not exist a path $\nu \in H_k$ such that $[\nu] = [\mu]$. Hence, we can replace H_s in equality (2.31) by $H_{sk}(T)$.

To obtain the desired vector $x \in R^q$, we have to use equality (2.31) for each digraph $G_k \in \Lambda(G), k \neq s$. Let r denote the minimum of such a value r_{G_k}:

$$r = r_{G_{k^*}} = \min\{r_{G_k} : G_k,\ k \neq s\}$$

and let $\nu^* \in H_{k^*}$ and $\mu^* \in H_{sk^*}$ be paths at which value $r_{G_{k^*}}$ has been reached:

$$r_{G_{k^*}} = r_{\nu^*} = \frac{l^p(\nu^*) - l^p(\mu^*)}{|[\mu] + [\nu]|}.$$

Due to the Remark 2.2, we have obtained a lower bound for the stability radius:

$$\widehat{\varrho}^B_s(p \in T) \geq r = \min_{G_k \in B} \quad \min_{\mu \in H_{sk}(T)} \quad \max_{\nu \in H_k;\ l^p(\nu) \geq l^p_s} \frac{l^p(\nu) - l^p(\mu)}{|[\mu] + [\nu]|}. \tag{2.32}$$

The bound (2.32) is tight: If $\widehat{\varrho}^B_s(p \in T) \leq \min\{\Delta^i(\mu^*, \nu^*) \ : \ i \in [\mu^*] \cup [\nu^*]\}$, then $\widehat{\varrho}^B_s(p \in T) = r$. In particular, we have $\widehat{\varrho}^B_s(p \in T) = r$ in (2.32) if $\widehat{\varrho}^B_s(p \in T) \leq \epsilon_{min}$.

To obtain the exact value of $\widehat{\varrho}^B_s(p \in T)$ in the general case, we can use the vector $x = p^*(r) = (p^*_1(r), p^*_2(r), ..., p^*_q(r))$ with the components

$$p^*_i(r) = \begin{cases} p_i + \min\{r, b_i - p_i\}, & \text{if } i \in [\mu], \\ p_i - \min\{r, p_i - a_i\}, & \text{if } i \in [\nu] \setminus [\mu], \\ p_i, & \text{if } i \notin [\mu] + [\nu], \end{cases} \tag{2.33}$$

instead of the vector $p(r)$ defined in (2.30). As it follows from Remark 2.2, such a vector $p^*(r) \in T$ is the closest one to the vector p among all vectors $x \in T$ which satisfy both conditions 1 and 2.

For calculating the maximal value r for the vector $p^*(r)$, we can consider each operation i from the set $[\mu] \cup [\nu]$ one by one in non-decreasing order (2.23) of the values $\Delta^i(\mu, \nu)$ defined in (2.22). As a result, formula (2.32) will be transformed into the formulas given in Theorem 2.3.

$\diamond$

Remark 2.3 Note that the formulas in Theorem 2.3 turn into $\widehat{\varrho}_s^B(p \in T) = \infty$ if $H_{sk}(T) = \emptyset$ for each $G_k \in B$.

Example 2.1 (continued). Returning to the Example 2.1 in Section 2.1, let us consider problem $\mathcal{J}3/n = 2, a_i \le p_i \le b_i/\mathcal{C}_{max}$ whose input data are given by the weighted mixed graph $G(p)$ in Figure 2.1 together with the vectors $a = (a_{11}, a_{12}, \dots, a_{23})$ and $b = (b_{11}, b_{12}, \dots, b_{23})$ of lower and upper bounds for the possible variations of the processing times p, where $a = (35, 40, 20, 50, 45, 20)$ and $b = (100, 90, 110, 80, 80, 40)$. So, the numerical input data for this problem are given in Table 2.3.

Table 2.3: *Numerical data for problem $\mathcal{J}3/n\!=\!2, a_i \le p_i \le b_i/\mathcal{C}_{max}$*

i	1	1	1	2	2	2
j	1	2	3	1	2	3
a_{ij}	35	40	20	50	45	20
b_{ij}	100	90	110	80	80	40

Since the mixed graph G is the same for the above problem $\mathcal{J}3/n\!=\!2/\mathcal{C}_{max}$ considered in Section 2.1 and for the new problem $\mathcal{J}3/n = 2, a_i \le p_i \le b_i/\mathcal{C}_{max}$, we have the same set $\Lambda(G)$ of feasible digraphs (see Figure 2.2). Moreover, if we start with the same initial vector $p = (75, 50, 40, 60, 55, 30)$ of the processing times, we obtain the same optimal digraph G_1, presented in Figure 2.3 with the dummy operations and in Figure 2.2 without the dummy operations. Using Theorem 2.3, we can calculate the relative stability radius of this digraph: $\widehat{\varrho}_1^{\Lambda(G)}(p \in T) = 60$, where the polytope $T \in R^6_+$ is defined by the above vectors a and b (see Table 2.3). Note that, due to these bounds a_{ij} and b_{ij} for the possible variations of the processing times p_{ij}, $O_{ij} \in Q = \{O_{11}, O_{12}, \dots, O_{23}\}$, the stability radius of the digraph G_1 increased from 30 to 60 (remind that in Section 2.1, we calculated $\widehat{\varrho}_1^{\Lambda(G)}(p \in R^6_+) = \varrho_1(p) = 30$).

In Table 2.4, one can observe the calculation of $\widehat{\varrho}_1^{\Lambda(G)}(p \in T)$. The set $H_{1k}(T)$ is empty for each digraph $G_k, k \in \{2, 4, 5\}$. Note that $H_{sk}(T) \subseteq H_{sk}$, therefore we have $H_{12}(T) = \emptyset$ and $H_{14}(T) = \emptyset$ and for both paths $\mu_1 = (O_{11}, O_{12}, O_{23}) \in H_{15}$ and $\mu_2 = (O_{11}, O_{22}, O_{23}) \in H_{15}$, there exists a path $\nu_2 = (O_{21}, O_{22}, O_{11}, O_{12}, O_{13}) \in H_5$ which dominates both paths μ_1 and μ_2 in the polytope T, i.e. inequality (2.6) holds. Table 2.4 has an analogous design as Table 2.2 from Section 2.1 with the exception of column 5, which contains the values $\Delta^{ij}_\beta(\mu, \nu)$, $\beta = 0, 1, \dots, |[\mu] + [\nu]| - 1$, defined by formula (2.22) at page 40 in non-decreasing order (2.23). Let us consider a path $\mu = (O_{21}, O_{12}, O_{13}) \in H_{13}(T)$ and a path $\nu_1 = (O_{11}, O_{12}, O_{13}) \in H_3$. For each vertex from the set $[\mu] + [\nu_1]$ (the symmetric difference of the sets $[\mu]$ and $[\nu_1]$), $|[\mu]+[\nu_1]| = 2$, we calculate the values $\Delta^{11}(\mu,\nu_1) = p_{11} - a_{11} = 75 - 35 = 40$, $\Delta^{21}(\mu,\nu_1) = b_{21} - p_{21} = 80 - 60 = 20$. By a comparison of the path μ with the path $\nu_2 = (O_{11}, O_{12}, O_{21}, O_{22}, O_{23}) \in H_3$, we find the values $\Delta^{11}(\mu,\nu_2) = p_{11} - a_{11} = 75 - 35 = 40$, $\Delta^{13}(\mu,\nu_2) = b_{13} - p_{13} = 110 - 40 = 70$, $\Delta^{22}(\mu,\nu_2) = p_{22} - a_{22} = 55 - 45 = 10$, $\Delta^{23}(\mu,\nu_2) = p_{23} - a_{23} = 30 - 20 = 10$. The sequential calculations of the fraction from the formula (2.25) are represented in column 6 of Table 2.4. Column 9 (see Table 2.2) is redundant for this small example.

Table 2.4: *Calculation of the relative stability radius $\widehat{\varrho}_1^{\Lambda(G)}(p \in T)$ for problem $\mathcal{J}3/n = 2, a_i \le p_i \le b_i/\mathcal{C}_{max}$*

G_k	$\lvert H_{1k}(T)\rvert$	$\mu \in H_{1k}(T)$, $l^p(\mu)$	$\nu \in H_k$: $l^p(\nu) \ge l_1^p$	$\Delta_\beta^{ij}(\mu,\nu)$, $0 \le \beta \le \lvert[\mu]+[\nu]\rvert - 1$	$\frac{l^p(\nu) - l^p(\mu) - \sum_{\alpha=0}^{\beta} \Delta_\alpha^{ij}(\mu,\nu)}{\lvert[\mu]+[\nu]\rvert - \beta}$	$\max_\beta$	$\max_\nu$
1	2	3	4	5	6	7	8
G_2	0						
G_3	1	(O_{21}, O_{12}, O_{13}), $l^p(\mu) = 150$	(O_{11}, O_{12}, O_{13}): $l^p(\nu_1) = 165 = l_1^p$	$\Delta_0^{ij}(\mu,\nu_1) = 0$ $\Delta_1^{21}(\mu,\nu_1) = 20$	$\frac{165-150-0}{2-0} = 7.5$ $\frac{165-150-20}{2-1} < 0$	7.5	60
			$(O_{11}, O_{12}, O_{21}, O_{22}, O_{23})$: $l^p(\nu_2) = 270 > 165$	$\Delta_0^{ij}(\mu,\nu_2) = 0$ $\Delta_1^{22}(\mu,\nu_2) = 10$ $\Delta_2^{23}(\mu,\nu_2) = 10$ $\Delta_3^{11}(\mu,\nu_2) = 40$	$\frac{270-150-0}{4-0} = 30$ $\frac{270-150-10}{4-1} = 36\frac{2}{3}$ $\frac{270-150-(10+10)}{4-2} = 50$ $\frac{270-150-(10+10+40)}{4-3} = 60$	60	
G_4	0						
G_5	0						

So, one of the two competitive digraphs, namely digraph G_3 (see Figure 2.2 at page 29 or Figure 2.4 at page 32), remains also a competitive digraph of G_1 for problem $\mathcal{J}3/n = 2, a_i \le p_i \le b_i/\mathcal{C}_{max}$. However, the new vector of the processing times $p^* = p^{(2)}$, calculated due to (2.33) with $r = \widehat{\varrho}_1^{\Lambda(G)}(p \in T) = 60$, $\mu = (O_{21}, O_{12}, O_{13}) \in H_1$ and $\nu = (O_{11}, O_{12}, O_{21}, O_{22}, O_{23}) \in H_{13}(T) \subseteq H_3$, is as follows: $p^{(2)} = (35, 90, 100, 80, 45, 20)$.

Next, we follow the scheme proposed at pages 30–33 for obtaining a solution of problem $\mathcal{J}3/n = 2, a_i \le p_i \le b_i/\mathcal{C}_{max}$. We calculate $\widehat{\varrho}_3^{\Lambda(G)\setminus\{G_1\}}(p^{(2)} \in T) = 32.5$ on the basis of Theorem 2.3 and obtain the competitive digraph G_2 of digraph G_3. For digraph G_2, the minimum in (2.24) is reached on the set $B = \Lambda(G) \setminus \{G_1\}$, and thus digraph G_2 becomes optimal at least for one point $p^{(3)}$ of the stability sphere (the boundary of the stability ball $O_{32.5}(p^{(2)})$). Then we calculate the stability radius $\widehat{\varrho}_2^{\Lambda(G)\setminus\{G_1,G_3\}}(p^{(3)} \in T) = 27.5$ for the new optimal digraph G_2 and for the new set $B := B \setminus \{G_3\} = \Lambda(G) \setminus \{G_1, G_3\}$, and so on. Solving problem $\mathcal{J}3/n = 2, a_i \le p_i \le b_i/\mathcal{C}_{max}$ takes four iterations (see Table 2.5). On the basis of Theorem 2.2 or Theorem 2.3, we obtain $\widehat{\varrho}_4^{\Lambda(G)\setminus\{G_1,G_2,G_3\}}(p^{(4)} \in T) = \infty$.

Thus, the set of digraphs $\{G_1, G_2, G_3, G_4\}$ is a solution of problem $\mathcal{J}3/n = 2, a_i \le p_i \le b_i/\mathcal{C}_{max}$:

$$\Lambda^*(G) = \{G_1, G_2, G_3, G_4\}.$$

Table 2.5: *Solution of problem $\mathcal{J}3/n=2, a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ by Algorithm $SOL\_\mathcal{C}_{max}(1)$*

i	Center $p^{(i)} \in T$ of the stability ball	Set B of feasible digraphs	Optimal digraph G_s	$\widehat{\varrho}_s^B(p^{(i)} \in T)$	Competitive digraph of G_s
1	$(75, 50, 40, 60, 55, 30)$	$\Lambda(G)$	G_1	60	G_3
2	$(35, 90, 100, 80, 45, 20)$	$\Lambda(G) \setminus \{G_1\}$	G_3	32.5	G_2
3	$(67.5, 90, 67.5, 80, 77.5, 40)$	$\Lambda(G) \setminus \{G_1, G_3\}$	G_2	27.5	G_4
4	$(40, 90, 95, 80, 80, 40)$	$\Lambda(G) \setminus \{G_1, G_2, G_3\}$	G_4	∞	–

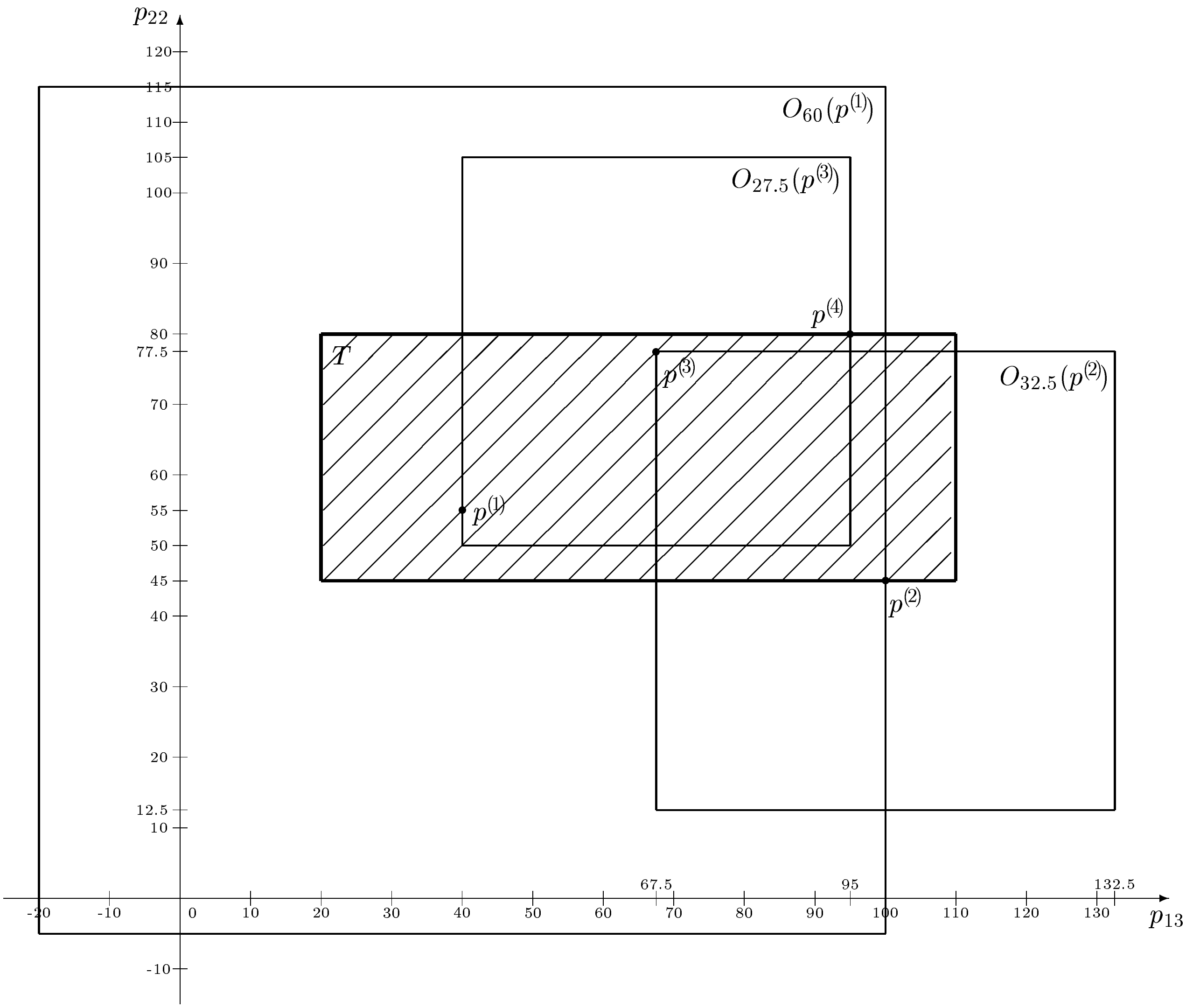


Figure 2.5: *Projections of the stability balls on the plane for problem $\mathcal{J}3/n=2, a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ constructed by Algorithm $SOL\_\mathcal{C}_{max}(1)$*

In other words, we cover the given polytope T by the union of the stability balls of the feasible digraphs from the set $\Lambda^*(G)$.

The projections of these stability balls on the plane for the component p_{13} of the vector p given at the axis of x-coordinates and for the component p_{22} of the vector p given at the axis of y-coordinates are drawn in Figure 2.5. The last stability ball $O_\infty(p^{(4)})$ with the radius $\widehat{\varrho}_4(p^{(4)} \in T) = \infty$ covers the given polytope T and all other stability balls. The stability ball $O_\infty(p^{(4)})$ cannot be shown in Figure 2.5. For some suitable changes of the processing times $p_i \pm (\widehat{\varrho}_s(p) + \epsilon)$ (where ϵ is positive real, and it may be as small as desired) at least one of the four digraphs $\{G_1, G_2, G_3, G_4\}$ becomes optimal. Therefore, a decision-maker can use one of the schedules from the set $\Lambda^*(G)$ for the possible realization of the processing times.

From Table 2.5 and Figure 2.5, one can see that the set $\{G_1, G_2, G_3, G_4\}$ cannot be inclusion minimal: The stability ball $O_{60}(p^{(1)})$ with the radius $\widehat{\varrho}_1^{\Lambda(G)}(p^{(1)} \in T) = 60$ 'covers' the intersection $O_{27.5}(p^{(3)}) \cap T$ of the stability ball with the radius $\widehat{\varrho}_2^{\Lambda(G)\setminus\{G_1,G_3\}}(p^{(3)} \in T) = 27.5$ and polytope T. As it will be shown at the end of this section, the solution $\{G_1, G_2, G_3, G_4\}$ is not minimal since at least digraph G_2 is redundant.

In general, problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$ may be solved as follows. Let B denote the set of feasible digraphs which contains an optimal set $\Lambda^*(G)$ for problem $\mathcal{G}/a_i \le p_i \le b_i/\mathcal{C}_{max}$. On the basis of the algorithm developed in [LSSW97], 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: A set $\Lambda(G)$, a polytope T.
Output: A solution $\Lambda^*(G)$.

Step 1: Find the set $B \subseteq \Lambda(G)$ of possible candidates for the set $\Lambda^*(G)$;

Step 2: set $\Lambda' = \emptyset$;

Step 3: fix the vector p of the processing times, $p \in T$;

Step 4: find an optimal digraph $G_s(p) \in B$ for problem $\mathcal{G}//\mathcal{C}_{max}$ with the vector p of the 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 $B \setminus \{G_s\} \neq \emptyset$ **THEN**
begin

Step 7: select a digraph $G_k(p) \in B$ which is a competitive digraph for 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 $\Lambda' := \Lambda' \cup \{G_s\}$;

Step 10: set $B := B \setminus \{G_s\}$;

Step 11 set $s = k$; $p = p^*$; **GOTO** *Step 5*
end

Step 12: **ELSE** $\Lambda^*(G) = \Lambda' \cup \{G_s\}$ **stop**.

Now we concretize 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. In the computational experiments discussed in [SSW97] (see Section 4.1), a direct enumeration has been used for $|E| \leq 30$. These experiments have shown that a competitive digraph has a critical weight that is usually very close to that of an optimal digraph. Moreover, using the simple bound from Section 2.4 below, 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 $\widehat{\varrho}_s^B(p \in T)$. So, for a larger cardinality of the set E, one can use a branch-and-bound algorithm for the construction of the k best digraphs (see Section 4.1). As it was shown for the traveling salesman problem [Lib99, LvdPSvdV96] and for linear binary programming [WJ88], the running time of such a branch-and-bound algorithm grows rather slowly with k.

In Step 3 one can fix the processing times as any vector from T. For example, one can use a 'historical' vector p of the processing times which helps to simplify the Steps 3, 4 or 5 (as it was in Example 2.1). If the input data of the problem are new, one can set $p_i = \frac{1}{2}(b_i - a_i),\ i \in Q$.

Step 4 may be realized by an explicit enumeration or by an implicit enumeration (e.g. by branch-and-bound method) of the feasible digraphs B. In Step 4 one can apply Theorem 2.1 to guarantee that the selected optimal digraph G_s is stable. If $\widehat{\varrho}_s^B(p \in T) = 0$, one can take another optimal digraph (the latter exists due to Theorem 2.1) which is stable, or one can change the initial vector p of the processing times.

Steps 5, 7, and 8 may be done on the basis of Theorem 2.2 and/or Theorem 2.3. If $\widehat{\varrho}_s^B(p \in T) = \infty$, Theorem 2.2 can be used as a 'stopping rule'. Otherwise, we are forced to use Theorem 2.3 which is more 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 $\widehat{\varrho}_s^B(p \in T)$. Note that a competitive digraph is not necessarily uniquely determined, so we can take one 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 is very time-consuming. Next, we propose a more efficient Algorithm $SOL\_\mathcal{C}_{max}(2)$, which focuses on one of the optimal digraphs G_1 and on one vector p from T.

Let $\Gamma_i, i = 1, 2, \ldots, I$, be a set of competitive digraphs of digraph G_1 with respect to the set B, where i is a counter of the current iteration and I is the number of the last iteration.

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

Input: A set $\Lambda(G)$, a polytope T.
Output: A solution $\Lambda^*(G)$.

Step 1: Find the set $B \subseteq \Lambda(G)$ of possible candidates for the set $\Lambda^*(G)$;

Step 2: set $\Lambda' = \emptyset$; $i = 1$ and $\Gamma_i = \emptyset$;

Step 3: fix the vector p of the processing times, $p \in T$;

Step 4: find an optimal digraph $G_1(p) := G_s(p) \in B$ for problem $\mathcal{G}//\mathcal{C}_{max}$ with the vector p of the 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 a set of competitive digraphs Γ_i of digraph $G_1(p)$ with respect to the set B;

Step 8: set $\Lambda' := \Lambda' \cup \Gamma_i$;

Step 9: set $B := B \setminus \Gamma_i$ and $i := i + 1$; **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 relative stability radii $\widehat{\varrho}_1 < \widehat{\varrho}_2 < \ldots < \widehat{\varrho}_I$ of the stability balls $O_{\widehat{\varrho}_i}(p),\ i \in \{1, 2, \ldots, I\}$, with the same center $p \in T$ and different sets of feasible digraphs $B = \Lambda(G) \setminus \cup_{j=1}^{i} \Gamma_j$. Moreover, we construct a sequence of 'nested sets' of the competitive digraphs $\Gamma_1,\ \Gamma_1 \cup \Gamma_2,\ \ldots,\ \cup_{i=1}^{I} \Gamma_i$ of digraph G_1, where the set $\{G_1\} \cup \{\cup_{i=1}^{I} \Gamma_i\}$ is a solution $\Lambda^*(G)$ of the scheduling problem for 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. Since 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), we should make the following remark.

Remark 2.4 It is not necessary to perform Steps 1 - 11 since we can construct a solution $\Lambda^*(G)$ in one scan. Namely, from Remark 2.3 it follows that all digraphs $G_k, k \neq 1$, for which a set $H_{1k}(T) \neq \emptyset$ was constructed in Step 5, compose a solution: $\Lambda^*(G) = \{G_1\} \cup \{\cup_{i=1}^{I} \Gamma_i\} = \{G_1\} \cup \{G_k :\ H_{1k}(T) \neq \emptyset\}$. We use the software developed for the problems discussed in Chapter 1 with the following modification: We add the loop of 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 for the optimal digraph $G_1(p)$ in Step 5. A more effective strategy (without adding the above loop) is described in Chapter 4.

Example 2.1 (continued). Solving the above problem takes only two iterations by Algorithm $SOL\_\mathcal{C}_{max}(2)$ (see Table 2.6). Thus, the set of digraphs

$$\Lambda^*(G) = \{G_1, G_3\}$$

Table 2.6: *Solution of problem $\mathcal{J}3/n\!=\!2, a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ by Algorithm $SOL\_\mathcal{C}_{max}(2)$*

i	Set B	$\widehat{\varrho}_1^B(p \in T)$	Set Γ_i of competitive digraphs of the optimal digraph G_1
1	$\Lambda(G)$	60	$\{G_3\}$
2	$\Lambda(G) \setminus \{G_3\}$	∞	$\emptyset$

is also a solution of problem $\mathcal{J}3/n=2, a_i \le p_i \le b_i/\mathcal{C}_{max}$. Using Algorithm $SOL\_\mathcal{C}_{max}(2)$, one can construct two stability balls $O_{60}(p)$ and $O_{\infty}(p)$, which cover the polytope T (see Figure 2.6). Again, the stability ball with an infinite radius cannot be shown. So, it is clear that Algorithm $SOL\_\mathcal{C}_{max}(1)$ did not construct a minimal solution. In general case, we do not know whether Algorithm $SOL\_\mathcal{C}_{max}(2)$ constructs a minimal solution. However, it is easy to show that there is no one-element solution of this problem. Hence the solution $\{G_1, G_3\}$ presented in Table 2.6 is a minimal one with respect to cardinality of the set $\Lambda^*(G)$.

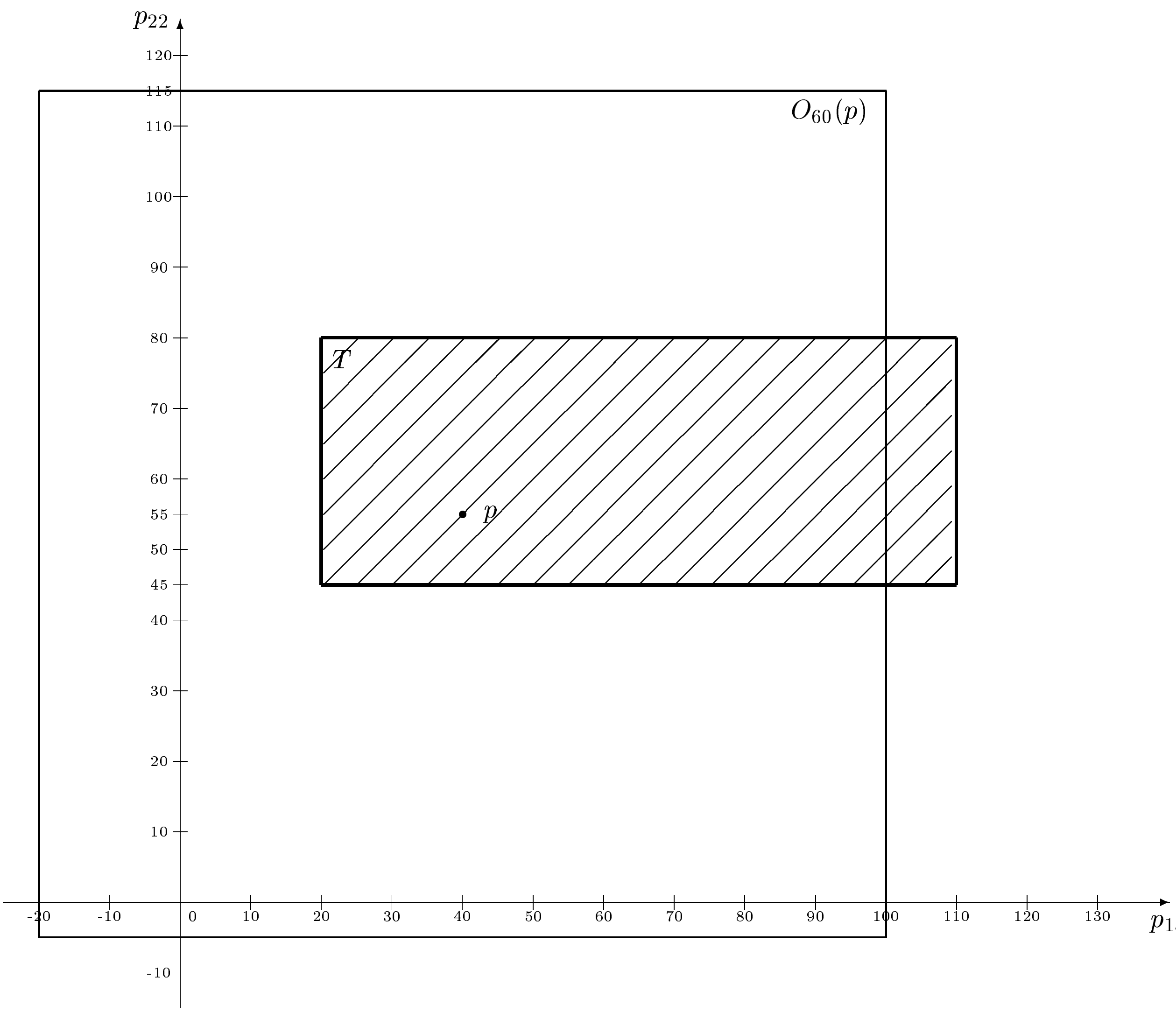


Figure 2.6: *Projections of the stability balls on the plane for problem $\mathcal{J}3/n=2, a_i \le p_i \le b_i/\mathcal{C}_{max}$ constructed by Algorithm $SOL\_\mathcal{C}_{max}(2)$*

Remark 2.5 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 solution.

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

2.4 Redundant Digraphs for Calculating $\widehat{\varrho}_s^B(p \in T)$

Due to formulas (2.24) at page 41, the calculation of the relative stability radius is reduced to a complicated calculation on the set of all 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 $G_k \in B$. At 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 $B \subseteq \{G_1, G_2, \ldots, G_\lambda\}$. 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 the stability radius $\widehat{\varrho}_s^B(p \in T)$, we can use the upper bound of the relative stability radius $\widehat{\varrho}_s^B(p \in T) \leq \widehat{r}_{ks}^B$, where $\widehat{r}_{ks}^B$ is defined according to formula (2.25) at page 41.

Lemma 2.3 *If $\widehat{\varrho}_s^B(p \in T) < \infty$ and there exists a digraph $G_k \in B$ such that*

$$\widehat{r}_{ks}^B \leq \frac{l_t^p - l_s^p}{q} \quad \textit{for some } t \textit{ with } G_t \in B, \tag{2.34}$$

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

PROOF. To calculate the stability radius $\widehat{\varrho}_s^B(p \in T)$, one can compare the optimal digraph G_s consecutively with each feasible digraph $G_i, i \neq s$, from the set B. The value $\widehat{r}_{ks}^B$ calculated according to (2.25) shows that there exists a feasible digraph G_k, which becomes better than digraph G_s for some vector $p' \in T$ if

$$d(p, p') = \widehat{r}_{ks}^B + \epsilon,$$

where ϵ is a positive real number and it may be as small as desired (see condition 2 introduced at page 41). Let us show that, if the condition of Lemma 2.3 is satisfied, i.e. inequality (2.34) holds, then the value $\widehat{r}_{ts}^B$ calculated for the digraph G_t does not improve the minimum in formula (2.24) (i.e. inequalities $\widehat{\varrho}_s^B(p \in T) \leq \widehat{r}_{ks}^B \leq \widehat{r}_{ts}^B$ hold).

Let us compare the optimal digraph G_s with a feasible digraph G_t, $t \neq k$. From the condition 1 at page 41 (condition 2, respectively) it follows that digraph G_t is a competitive digraph for G_s if the weight of each path $\nu \in H_t$ of digraph G_t becomes equal to (smaller than) the weight of at least one path $\mu^* \in H_s$ of digraph G_s for some new vector $\widehat{x} \in T$ (for some new vector $\widehat{p}^\epsilon = \widehat{x} \pm \epsilon \in T$, where $\epsilon = d(\widehat{x}, \widehat{p}^\epsilon) > 0$ may be as small as desired). Hence, the inequality

$$\max_{\mu^* \in H_s} l^{\widehat{x}}(\mu^*) > \max_{\nu \in H_t} l^{\widehat{x}}(\nu)$$

$$\left(\max_{\mu^* \in H_s} l^{\widehat{p}^\epsilon}(\mu^*) > \max_{\nu \in H_t} l^{\widehat{p}^\epsilon}(\nu) \right)$$

holds. It means that the critical weight of digraph G_t becomes smaller than that of digraph G_s in some feasible realization of the process. Such a 'superiority' of the competitive digraph G_t occurs for some suitable changes of the processing times $\widehat{p}_i^\epsilon = p_i \pm (\widehat{r}_{ts}^B + \epsilon) = \widehat{x}_i \pm \epsilon$, when the value $\widehat{r}_{ts}^B = d(p, \widehat{x})$ calculated in (2.25) reaches the minimum value in (2.24) (see condition 3). To this end, one must increase the weights of the vertices, which form a path $\mu^* \in H_s$, by the minimal value $\widehat{r}_{ts}^B$ and decrease the weights of vertices from the set $[\nu^*] \setminus [\mu^*]$, $\nu^* \in H_t$, by the same value $\widehat{r}_{ts}^B$ (according to formula (2.33)). Note that we must take such a path $\nu^* \in H_t$ for which the maximum in (2.25) is reached.

So, for the competitive digraph G_t, the distance $d(p, \widehat{x}) = \widehat{r}_{ts}^B$ must achieve its minimal value in (2.24) among the distances between the vector p and the other vectors in the polytope T (i.e. the non-strict inequality $\widehat{r}_{ts}^B \leq \widehat{r}_{ks}^B$ is also satisfied). However, we show further that, due to (2.34), value $\widehat{r}_{ts}^B$ cannot be smaller than $\widehat{r}_{ks}^B$ during the calculation of the relative stability radius $\widehat{\varrho}_s^B(p \in T)$ (see Theorem 1.4):

$$\begin{aligned}
\widehat{r}_{ks}^B &\leq \frac{l_t^p - l_s^p}{q} \leq \\
&\leq \frac{l_t^p - l^p(\mu^*)}{q} \leq \\
&\leq \frac{l_t^p - l^p(\mu^*)}{|[\nu^*]\backslash[\mu^*] + [\mu^*]\backslash[\nu^*]|} \leq \\
&\leq \max_{\nu \in H_t} \frac{l^p(\nu) - l^p(\mu^*)}{|[\mu^*] \cup [\nu]| - |[\mu^*] \cap [\nu]|} \leq \\
&\leq \min_{\mu \in H_s} \max_{\nu \in H_t} \frac{l^p(\nu) - l^p(\mu)}{|[\mu] + [\nu]|} \leq \widehat{r}_{ts}^B.
\end{aligned}$$

Since $\widehat{\varrho}_s^B(p \in T) \leq \widehat{r}_{ks}^B \leq \widehat{r}_{ts}^B$, the value $\widehat{r}_{ts}^B$ cannot decrease the value $\widehat{r}_{ks}^B$ in (2.24) and therefore digraph G_t need not to be considered during the calculation of the stability radius.

◇

Lemma 2.3 directly implies the following corollary.

Corollary 2.5 *Let the set $\Lambda(G) = \{G_s = G_{i_1}, G_{i_2}, \ldots, G_{i_\lambda}\}$ be sorted in non-decreasing order $G_{i_1}, G_{i_2} \ldots, G_{i_\lambda}$ of the objective function values $l_{i_1}^p \leq l_{i_2}^p \leq \ldots \leq l_{i_\lambda}^p$. If for the currently compared digraph G_{i_k} from the set $B \subseteq \Lambda(G) = \{G_s = G_{i_1}, G_{i_2}, \ldots, G_{i_k}, \ldots, G_{i_t}, \ldots, G_{i_\lambda}\}$ the inequality*

$$\widehat{r}_{i_k s}^B \leq \frac{l_{i_t}^p - l_{i_1}^p}{q} \tag{2.35}$$

holds for digraph $G_{i_t} \in B \subseteq \Lambda(G)$ with $l_{i_k}^p \leq l_{i_t}^p$, 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 $\widehat{\varrho}_s^B(p \in T)$.

PROOF. Since the digraphs in the set $B \subseteq \Lambda(G)$ are sorted in non-decreasing order of the objective function values and inequality (2.35) holds for digraph G_{i_t}, inequality

$$\widehat{r}_{i_k s}^B \leq \frac{l_{i_j}^p - l_{i_1}^p}{q}$$

holds for each digraph $G_{i_j}, j = t+1, t+2, \ldots, |B|$, and due to Lemma 2.3, these digraphs need not to be considered during the calculation of the relative stability radius (since we have the upper bound $\widehat{\varrho}_s^B(p \in T) \leq \widehat{r}_{i_k s}^B \leq \frac{l_{i_j}^p - l_{i_1}^p}{q}$).

◇

Using Corollary 2.5, we can compare the optimal digraph $G_s = G_{i_1}$ consecutively with the digraphs $G_{i_2}, G_{i_3}, \ldots, G_{i_\lambda}$ from the set $\Lambda(G)$ in non-decreasing order of the objective function values: $l_{i_1}^p \leq l_{i_2}^p \leq \ldots \leq l_{i_\lambda}^p$. If for the currently compared digraph $G_k = G_{i_r}$ inequality (2.34) holds, we can exclude the digraphs $G_{i_r}, G_{i_{r+1}}, \ldots, G_{i_\lambda}$ from further considerations.

2.5 Resume and Notations

In [Pin95a], 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." In this chapter, we considered problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ for dealing with uncertain scheduling environments in which only lower and upper bounds for the processing times are known before scheduling. Such a problem may arise in many practical situations since, even if no specific bounds for an uncertain processing time p_i are known, we can set $a_i = 0$ and b_i equal to the horizon of planning.

As far as we know, such a type of scheduling problem was not considered in the OR literature so far. In Section 2.1, we defined a solution of problem $\mathcal{G}/a_i \leq p_i \leq b_i/\Phi$ as a minimal (with respect to inclusion) set of schedules such that at least one of them is optimal for any fixed processing time p_i in the closed interval $[a_i, b_i]$, $i \in Q$. We used a mixed graph model for representing the input data, the scheduling process and the final solution. Our 'strategy' was to separate the 'structural' input data from the 'numerical' input data as much as possible. The precedence and capacity constraints (i.e. the structural input data) are given by the mixed graph G, which completely defines the set of feasible schedules. The set of optimal schedules is defined by the weighted mixed graph $G(p)$ which presents both the structural and numerical input data.

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 digraph G_s which may be critical (see Lemma 2.2 and Theorem 2.1). To restrict the set of paths which may be critical, one can use a dominance relation for the set of paths reduced in Section 1.2 (see Definition 1.2). 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 problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ in Section 2.2, we generalized the dominance relation (see Definition 2.3) 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 2.1) and an infinite relative stability radius (Theorem 2.2). In Section 2.3, we have given a formula for calculating the exact value of the relative stability radius (Theorem 2.3). These results may be considered as a mathematical background for developing algorithms for solving problem $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$.

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.

Table 2.7 combines the main notations used in this chapter for the general shop problem. The common notations for job shop and general shop problems are given in Table 4.11 at page 130.

Table 2.7: *Notations for the general shop problem*

Symbols	Description
Q	Set of operations: $Q = \{1, 2, \ldots q\}$
q	Number of operations: $q = \vert Q \vert$
j_i	Last operation of job J_i, $1 \leq i \leq n$
s_i	Starting time of operation i
c_i	Completion time of operation i
p_i	Processing time of operation i
a_i	Lower bound for the processing time of operation i, given before scheduling
b_i	Upper bound for the processing time of operation i, given before scheduling
$Q^{(i)}$	Set of all operations of job J_i, $i \in \{1, 2, \ldots, n\}$
$c_i(s)$	Earliest completion time of operation $i \in Q$ in the digraph $G_s(p)$
$s \in S$	Semiactive schedule $s = (c_1(s), c_2(s), \ldots, c_q(s))$, defined by the digraph G_s and the vector p
$\tilde{H}$	Set of all paths in the digraph $(Q, A, \emptyset)$
$\tilde{H}_s$	Set of all paths in the digraph $G_s \in \Lambda(G)$
H	Set of all dominant paths in the digraph $(Q, A, \emptyset)$
H_s	Set of all dominant paths in the digraph $G_s \in \Lambda(G)$
$H_k(p)$	Set of all critical dominant paths in the digraph $G_k \in \Lambda(G)$ (with respect to the vector p); $H_k(p) \subseteq H_k$
l_k^p	Critical weight of digraph $G_k \in \Lambda(G)$ with the vector p of processing times: $\Phi_k^p = l_k^p = \max_{\mu \in H_s} l^p(\mu) = l^p(\mu^*)$, where $\mu^* \in H_s(p)$
$\tilde{H}_k^i$	Set of paths in digraph G_k ending in the last vertex j_i (operation) of job J_i
H_k^i	Set of all dominant paths in the set $\tilde{H}_k^i$
H_{sk}	Subset of the set H_s of all paths, which are not dominated by paths from the set H_k: $H_{sk} = \{\mu \in H_s :$ There is no path $\nu \in H_k$ such that $\{\mu\} \subseteq \{\nu\}\}$
$H_{sk}(T)$	$H_{sk}(T) = \{\mu \in H_s :$ There is no path $\nu \in H_k$ which dominates path μ in the polytope $T\}$
$[\mu] + [\nu]$	'Symmetric difference' $[\mu] + [\nu] = [\mu] \cup [\nu] \setminus [\mu] \cap [\nu]$ of sets $[\mu]$ and $[\nu]$

Chapter 3

Job Shop Problem with Mean Flow Time Criterion

In this chapter, the job shop problem with the objective of minimizing the sum of job completion times under uncertain numerical input data is modeled in terms of a mixed graph. It is assumed that only the structural input data (i.e. precedence and capacity constraints) are fixed while for the operation processing times only their lower and upper bounds are known before scheduling and the probability distribution functions of the random processing times are unknown. The structural input data are defined by the technological routes of the jobs, e.g. for a flow or open shop fixing the structural input data simply means to fix the number of jobs and the number of machines. Two variants of a branch-and-bound method are developed. The first one constructs a set of k schedules which are the best with respect to the mean flow time criterion for some vector of processing times. The second variant constructs a set of potentially optimal schedules for all perturbations of the processing times within the given lower and upper bounds. To exclude redundant schedules, we use a stability analysis based on the pairwise comparison of schedules. Along with implicit enumerations based on a branch-and-bound method, we realize an explicit enumeration of all feasible schedules. The results which are given in this chapter have been published in [LSSW98, Sotskova99b, Sotskova99c, SW00].

3.1 Dominance Relations

Assume that n jobs $J = \{J_1, J_2, \ldots, J_n\}$ have to be processed on m machines $M = \{M_1, M_2, \ldots, M_m\}$ when only the technological routes of the jobs are given before scheduling. At the stage $j \in \{1, 2, \ldots, n_i\}$ of job J_i, operation $O_{ij} \in Q_k$ has to be processed on machine $M_k \in M$. The distribution of the operations to the machines M is fixed via the given technological routes of the jobs J. Each machine can process at most one operation at a time (see Assumption 1) and preemptions of an operation are forbidden (see Assumption 2).

We consider the job shop problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ with fixed technological routes and uncertain processing times p_{ij}, which have to satisfy only the inequalities $a_{ij} \le p_{ij} \le b_{ij}$, $J_i \in J$; $j = 1, 2, \ldots, n_i$ (see Assumption 3). The sum of the job completion times (mean flow time) is considered as the objective function $\Phi = \Phi(C_1, C_2, \ldots, C_n) = \sum_{i=1}^{n} C_i = \sum \mathcal{C}_i$, where $C_i = c_{in_i}$ is the completion time of job $J_i \in J$.

To present the structural input data for problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$, we use the mixed graph $G = (Q, A, E)$ introduced at page 9. Such a mixed graph G defines the structural input data (precedence and capacity constraints) which are supposed to be known before scheduling. A schedule is defined as a circuit-free digraph $G_s = (Q, A \cup E_s, \emptyset)$ generated from the mixed graph G by replacing each edge $[O_{ij}, O_{uv}] \in E$ by one of the arcs (O_{ij}, O_{uv}) or (O_{uv}, O_{ij}). In this chapter, we use the terms of an *optimal schedule (digraph)*, a *better* and a *best schedule (digraph)* with respect to the mean flow time criterion $\sum \mathcal{C}_i$. However, the makespan criterion $\mathcal{C}_{max}$ and a regular criterion Φ are considered in this chapter as well.

Due to Definition 2.1, a *solution* $\Lambda^*(G)$ of problem $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$ is a set of digraphs containing at least one optimal digraph for each feasible vector $p = (p_{11}, p_{12}, \ldots, p_{nn_n})$ of the processing times, i.e. for each vector $p \in T$, where

$$T = \{x = (x_{11}, x_{12}, \ldots, x_{nn_n}) : \ a_{ij} \leq x_{ij} \leq b_{ij};\ i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, n_i\}$$

is the polytope of all feasible vectors of the processing times in the space R^q_+, with $q = |Q| = \sum_{i=1}^{n} n_i = \sum_{k=1}^{m} |Q_k|$. It is practically important to look for a *minimal solution* $\Lambda^T(G)$ of problem $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$, i.e. for a minimal subset of the set $\Lambda(G)$ containing at least one optimal digraph for each fixed vector $p \in T$ of the processing times such that any proper subset of the set $\Lambda^T(G)$ is not a solution (see Definition 2.1).

If the processing times p_{ij} of all operations $O_{ij} \in Q$ are fixed, one can calculate the value of the objective function for a digraph $G_s \in \Lambda(G)$ using the critical path method (CPM) [Dij59]. As it follows from Section 1.2, to solve problem $\mathcal{J}//\Phi$ we must find a digraph G_s such that $\Phi^p_s = \min\{\Phi^p_k : \ k = 1, 2, \ldots, \lambda\}$ (see formula (1.3)), where

$$\Phi^p_k = \Phi(\max_{\nu \in H^1_k} l^p(\nu), \max_{\nu \in H^2_k} l^p(\nu), \ldots, \max_{\nu \in H^n_k} l^p(\nu))$$

is the value of the objective function of the job completion times for the digraph $G_k \in \Lambda(G)$ with fixed processing times $p \in R^q_+$, and $l^p(\mu)$ is the weight of path μ: $l^p(\mu) = \sum_{O_{ij} \in [\mu]} p_{ij}$. Remind that $\Phi^p_s = l^p_s$ for criterion $\mathcal{C}_{max}$ and $\Phi^p_s = L^p_s$ for criterion $\sum \mathcal{C}_i$. As it has been shown in [KSW95] (see Theorem 1.3), for criterion $\mathcal{C}_{\max}$ there exist problems $\mathcal{J}//\mathcal{C}_{\max}$ for which the optimality of a schedule s does not depend on the numerical input data, i.e. $\widehat{\varrho}_s(p) = \infty$, which means that this schedule s minimizes the makespan for all non-negative processing times. However, such a schedule cannot exist for criterion $\sum \mathcal{C}_i$. In other words, each mean flow time optimal schedule loses its optimality for some vectors $p \in R^q_+$ of the processing times, i.e. $\overline{\varrho}_s(p) < \infty$ (see Theorem 1.7 and Remark 1.1). As it will be shown in the proof of Theorem 3.4 for the case of the relative stability radius $\overline{\varrho}^B_s(p \in T)$ (see Definition 3.2 below) when $T \subset R^q_+$ and $B \subset \Lambda(G)$, an unrestricted value of $\overline{\varrho}^B_s(p \in T)$ is possible.

In [Sotskova99b, SW00], an approach for dealing with 'strict uncertainty' based on a stability analysis of an optimal semiactive schedule was generalized for an uncertain job shop problem with any given regular criterion Φ. For problem $\mathcal{J}/a_i \leq p_i \leq b_i/\Phi$, we introduce the following two transitive *dominance relations* which define partial orderings on the set of digraphs $\Lambda(G)$.

Definition 3.1 *Digraph G_s dominates (strongly dominates) digraph G_k in domain $D \subseteq R^q_+$ if inequality $\Phi^p_s \leq \Phi^p_k$ (inequality $\Phi^p_s < \Phi^p_k$, respectively) holds for any vector $p \in D$ of the processing times, and we denote the dominance relation by $G_s \preceq_D G_k$ (and the strong dominance relation by $G_s \prec_D G_k$).*

If $a_{ij} = b_{ij}$ for each operation $O_{ij} \in Q$ (i.e. if T turns into a point which implies that problem $\mathcal{J}/a_i \le p_i \le b_i/\Phi$ turns into a deterministic problem $\mathcal{J}//\Phi$), the dominance relation $\preceq_T$ defines a total ordering on the set of digraphs $\Lambda(G)$, and consequently the set $\Lambda^T(G)$ consists of a single digraph: $\Lambda^T(G) = \{G_s\}$, where G_s is any optimal digraph for problem $\mathcal{J}//\Phi$ with processing times p_{ij} being equal to $a_{ij} = b_{ij}$ for each operation $O_{ij} \in Q$. In other words, digraph G_s dominates all digraphs $G_k \in \Lambda(G)$ at the point $a \in R^q_+$, i.e. $G_s \preceq_a G_k$. Moreover, if the strong dominance relation holds for each digraph $G_k \in \Lambda(G)$ at the point $a = b$, i.e. if $G_s \prec_a G_k$, then digraph G_s is the unique optimal one for the processing times p_{ij} equal to $a_{ij} = b_{ij}$. As it follows from the computational results carried out in [SSW97] (see Section 4.2 below), an optimal digraph for problem $\mathcal{J}//\Phi$ is usually uniquely determined. In other words, if the dominance relation $G_s \preceq_a G_k$ is valid for each digraph $G_k \in \Lambda(G)$, then generally the strong dominance relation $G_s \prec_a G_k$ is valid for each digraph $G_k \in \Lambda(G)$ with $k \neq s$. Note that this is not the case for the makespan criterion: For almost all job shop problems randomly generated in Section 4.2, makespan optimal digraphs are not uniquely determined.

For problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$, the operation processing times may vary between given lower and upper bounds and therefore it is a priori unknown which path from the set H^i_k will have the largest weight in a practical realization of the schedule G_k. Thus, we have to consider the whole set Ω^u_k of representatives of the family of sets $(H^i_k)_{J_i \in J}$ in a similar way to the approach considered for the problem $\mathcal{J}//\sum \mathcal{C}_i$ (see Section 1.4).

Next, we show how to restrict the number of sets of representatives which have to be considered while solving problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$. For different vectors $p \in R^q_+$ of the processing times, different sets Ω^u_k, $u \in \{1, 2, \dots, \omega_k\}$, may be *critical*, however a path $\nu \in H^i_k$, $J_i \in J$, may belong to a critical set only if $l^p(\nu) = \max_{\mu \in H^i_k} l^p(\mu)$. Therefore, while solving problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$, it is sufficient to consider only paths from the set H^i_k which may have the largest weight for at least one vector $p \in T$ of the processing times. Moreover, if there are two or more paths in H^i_k which have the largest weight at the same vector $p \in T$, it is sufficient to consider only one of them. Thus, it is sufficient to consider only *dominant* paths which were defined in Section 2.2 (see Definition 2.5).

Using Corollary 2.1, one can simplify digraph G_s while solving problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ or problem $\mathcal{J}/a_i \le p_i \le b_i/\mathcal{C}_{max}$. First, we delete all transitive arcs, then we delete some arcs based on the domination of sets of paths (see Definition 2.4). Let $H^i_s(T)$ denote the set of all dominant paths in H^i_s with respect to the polytope T. Since $H_s \subseteq \cup^n_{i=1} H^i_s$ for problem $\mathcal{J}/a_i \le p_i \le b_i/\mathcal{C}_{max}$, one can construct a set $H^i_s(T)$ as a subset of the set of all dominant paths H_s by selecting all paths ending in vertex O_{in_i} (if they exist). Let $G^T_s = (Q^T_s, E^T_s, \emptyset)$ be a *minimal subgraph* of digraph G_s such that, if $\mu \in \cup^n_{i=1} H^i_s(T)$, then digraph G^T_s contains path μ. To construct the digraph G^T_s, one can use the following straightforward modification of CPM [Dij59].

Assume that the path μ has the maximal weight among all paths in digraph G_s ending in vertex O_{ij} when the processing times are defined by the vector $p \in R^q_+$. As usual, the weight of path μ minus p_{ij} is called *earliest start time* of operation O_{ij} and we denote it by $l^p_s(O_{ij})$:

$$l^p_s(O_{ij}) = \sum_{O_{uv} \in [\mu] \setminus \{O_{ij}\}} p_{uv}.$$

The following recursive relations are obvious:

$$l^a_s(O_{ij}) = \max\{l^a_s(O_{uv}) + a_{uv} \ : \ (O_{uv}, O_{ij}) \in A \cup E_s\},$$

$$l^b_s(O_{ij}) = \max\{l^b_s(O_{uv}) + b_{uv} \ : \ (O_{uv}, O_{ij}) \in A \cup E_s\}.$$

Starting with a vertex in digraph G_s which has a zero in-degree and following the CPM approach, we define values $l_s^a(O_{ij})$ and $l_s^b(O_{ij})$ for each vertex $O_{ij} \in Q$. Then, using backtracking, we define vertices Q_s^T and arcs E_s^T of digraph G_s^T as follows. Initially, we set $Q_s^T = \{O_{in_i} : \ J_i \in J\}$ for problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ (if $H_s^i(T) \neq \emptyset$, $H_s^i(T) \subseteq H_s$, for problem $\mathcal{J}/a_i \le p_i \le b_i/\mathcal{C}_{max}$) and we set $E_s^T = \emptyset$. Then we add vertex O_{uv} to the set Q_s^T, and we add arc (O_{uv}, O_{in_i}) to the set E_s^T:

$$Q_s^T := Q_s^T \cup \{O_{uv}\}, \ E_s^T := E_s^T \cup \{(O_{uv}, O_{in_i})\}$$

if and only if the following two conditions hold:
1) there is no arc $(O_{u_1v_1}, O_{in_i})$ such that $l_s^b(O_{u_1v_1}) < l_s^a(O_{uv})$, and
2) inequality $l_s^b(O_{uv}) + b_{in_i} \geq l_s^a(O_{in_i})$ holds.
Continuing in a similar way for each vertex which is already included in the set Q_s^T, we construct the digraph G_s^T (see Lemma 2.1).

Thus, instead of digraphs $G_k, \ k = 1, 2, \ldots, \lambda$, one can consider digraphs G_k^T which contain all dominant paths $\cup_{i=1}^n H_k^i(T)$ and which are often essentially simpler than the corresponding digraphs G_k. The transformation of digraph G_k into digraph G_k^T by testing inequality (2.6)

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

(see page 35) takes $O(q^2)$ elementary steps (q is the number of operations).

Let for criterion $\sum \mathcal{C}_i$ the superscripts of the sets $\Omega_k^1, \Omega_k^2, \ldots, \Omega_k^{\omega_k^T}, \ldots, \Omega_k^{\omega_k}$ be such that for a path μ the inclusion $\mu \in \cup_{i=1}^n H_k^i(T)$ holds if and only if $\mu \in \cup_{i=1}^{\omega_k^T} \Omega_k^i, \ \omega_k^T = \prod_{i=1}^n |H_k^i(T)|$.

Example 3.1 To illustrate the above notions and definitions, we introduce a job shop problem $\mathcal{J}3/n{=}3, a_i \le p_i \le b_i/\sum \mathcal{C}_i$ with $Q_1 = \{O_{11}, O_{13}, O_{32}\}, Q_2 = \{O_{12}, O_{21}, O_{33}\}$ and $Q_3 = \{O_{22}, O_{31}\}$. The mixed graph $G = (Q, A, E)$ represented in Figure 3.1 defines the structural input data. The numerical input data are defined by the polytope $T \in R_+^8$ within which the actual vector p of the processing times has to be contained, and they are given in Table 3.1. For this small example, one can explicitly enumerate all digraphs of the set $\Lambda(G)$. Since not all digraphs may be optimal for the given segments $[a_{ij}, b_{ij}]$ of possible variations of the processing times, we construct a subset B of the set $\Lambda(G)$ of possible candidates of competitive digraphs (optimal digraphs) using the algorithms from Section 3.4 below. The cardinality of the set B is equal to 12, at the same time the cardinality λ of the set $\Lambda(G)$ is equal to 22.

Table 3.1: *Numerical data for problem* $\mathcal{J}3/n{=}3, a_i \le p_i \le b_i/\sum \mathcal{C}_i$

i	1	1	1	2	2	3	3	3
j	1	2	3	1	2	1	2	3
a_{ij}	60	20	45	10	50	60	30	30
b_{ij}	80	40	60	30	70	80	50	40

Before finding a minimal solution $\Lambda^T(G)$ of this problem $\mathcal{J}3/n{=}3, a_i \le p_i \le b_i/\sum \mathcal{C}_i$, we consider its deterministic version $\mathcal{J}3/n{=}3/\sum \mathcal{C}_i$ by setting the vector of the processing times to be equal to $p^0 = (p^0_{11}, p^0_{12}, \dots, p^0_{33}) \in T$ with $p^0_{11} = 70, p^0_{12} = 30, p^0_{13} = 60, p^0_{21} = 20, p^0_{22} = 60, p^0_{31} = 70, p^0_{32} = 40$, and $p^0_{33} = 30$ (this vector can be chosen arbitrarily from the polytope T). We number the digraphs $G_1, G_2, \dots, G_{12}$ in accordance with non-decreasing values of the function $\sum C_i$ calculated for the vector p^0 of the processing times: $L_1^{p^0} = 440, L_2^{p^0} = 470, L_3^{p^0} = 500, L_4^{p^0} = 500, L_5^{p^0} = 520, L_6^{p^0} = 530, L_7^{p^0} = 540, L_8^{p^0} = 550, L_9^{p^0} = 570, L_{10}^{p^0} = 610, L_{11}^{p^0} = 610, L_{12}^{p^0} = 620$.

For the vector $p^0 \in T$, the digraph $G_1 = (Q, A \cup A_1, \emptyset)$ with the signature $A_1 = \{(O_{11}, O_{32}), (O_{32}, O_{13}), (O_{21}, O_{12}), (O_{12}, O_{33}), (O_{21}, O_{33}), (O_{31}, O_{22})\}$ is the only optimal digraph. Therefore, for the initial problem $\mathcal{J}3/n{=}3, a_i \le p_i \le b_i / \sum \mathcal{C}_i$, we have to include the digraph G_1 in the desired minimal solution $\Lambda^T(G)$.

Using the above modification of CPM, we simplify the digraphs G_1, $G_2, \dots, G_{12}$. Then we compare the sets of representatives $\Omega_1^1, \Omega_1^2, \dots, \Omega_1^{\omega_1^T}$ for the digraph G_1 with the sets of representatives $\Omega_k^1, \Omega_k^2, \dots, \Omega_k^{\omega_k^T}$ for the other digraphs $G_k,\ k = 2, 3, \dots, 12$. Due to a pairwise comparison of these sets, we find that only two digraphs may be better than the digraph G_1 (provided that the vector of the processing times belongs to the polytope T defined in Table 3.1). These two digraphs are as follows: Digraph $G_2 = (Q, A \cup A_2, \emptyset)$ with the signature $A_2 = \{(O_{11}, O_{32}), (O_{13}, O_{32}), (O_{21}, O_{12}), (O_{12}, O_{33}), (O_{21}, O_{33}), (O_{22}, O_{31})\}$ and digraph $G_5 = (Q, A \cup A_5, \emptyset)$ with the signature $A_5 = \{(O_{11}, O_{32}), (O_{13}, O_{32}), (O_{21}, O_{12}), (O_{12}, O_{33}), (O_{21}, O_{33}), (O_{31}, O_{22})\}$. Moreover, the digraph G_2 is the only optimal one, for example, for the vector $p' = (60, 20, 60, 10, 60, 80, 40, 30) \in T$, and the digraph G_5 is the only optimal one for the vector $p'' = (60, 20, 45, 30, 70, 80, 50, 30) \in T$. Consequently, a minimal solution of problem $\mathcal{J}3/n{=}3, a_i \le p_i \le b_i / \sum \mathcal{C}_i$ consists of three digraphs, namely: $\Lambda^T(G) = \{G_1, G_2, G_5\}$. The corresponding digraphs G_1^T, G_2^T and G_5^T are represented in Figure 3.2 a), b) and c), respectively. Note that while digraph G_1 has $\omega_1 = 4 \text{ x } 2 \text{ x } 5 = 40$ sets of representatives, digraph G_1^T has only $\omega_1^T = 3 \text{ x } 1 \text{ x } 3 = 9$ sets of representatives. For the digraphs G_2 and G_2^T, these numbers are $\omega_2 = 16$ and $\omega_2^T = 2$, and for the digraphs G_5 and G_5^T, these numbers are $\omega_5 = 28$ and $\omega_5^T = 1$.

The above full enumeration of the digraphs $\Lambda(G)$ is only possible for a small number of edges in the mixed graph G, and for a practical use one has to reduce the number of digraphs which have to be constructed. E.g. for the illustrative example under consideration, it is sufficient to construct only $k = 5$ digraphs, which are the best for the

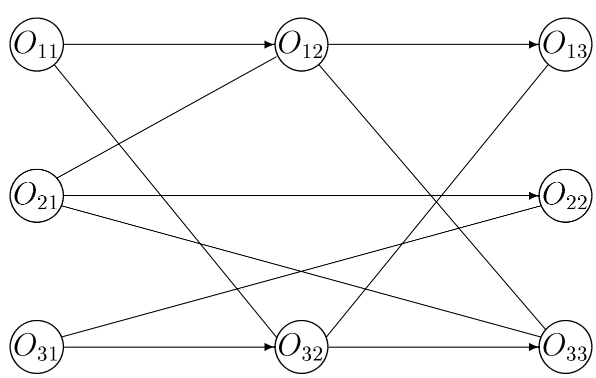


Figure 3.1: *Mixed graph $G = (Q, A, E)$ for problem $\mathcal{J}3/n{=}3, a_i \le p_i \le b_i / \sum \mathcal{C}_i$*

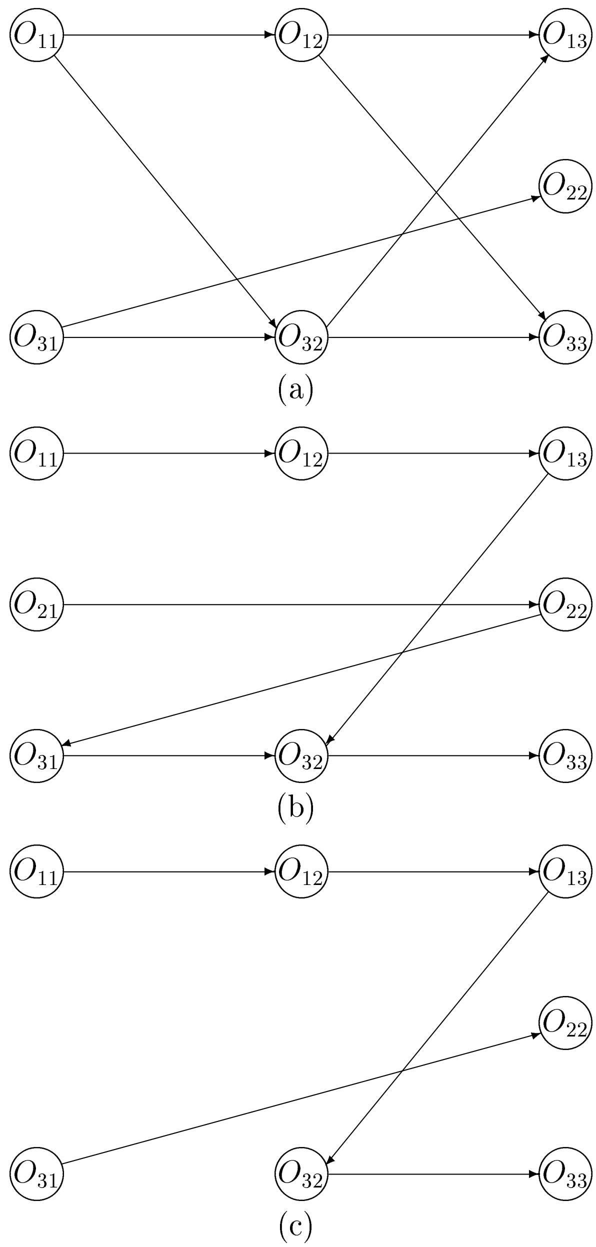


Figure 3.2: *Digraphs G_1^T, G_2^T and G_5^T which define a minimal solution $\Lambda^T(G)$ for Example 3.1*

initial vector p^0 of the processing times. Further, in Section 3.4 such a calculation will be developed on the basis of a branch-and-bound method for constructing k best digraphs. Moreover, the digraphs G_3 and G_4 in the set of the $k = 5$ best digraphs are also redundant. In Section 3.4, we present a branch-and-bound method for constructing all digraphs which are the only ones that may be optimal for feasible vectors of the processing times. We also show how to calculate the stability radius of an optimal digraph on the basis of an explicit enumeration of the digraphs $\Lambda(G)$. The calculation of the stability radius will be used in Sections 3.4 and 4.4 as the main procedure for finding a minimal solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\sum C_i$.

The remainder of this chapter is organized as follows. Necessary and sufficient conditions for a set of digraphs to be a solution respectively a minimal solution of the job shop problem with uncertain processing times are proven in Section 3.2. Three exact and four heuristic algorithms for problems $\mathcal{J}/a_i \le p_i \le b_i/\mathcal{C}_{max}$ and $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ are given in Section 3.4. In the algorithm based on an explicit schedule enumeration, we generalize the results from [BSW96] for the stability radius of an optimal schedule. Section 3.4 contains a branch-and-bound method ($B\&B1$) for constructing k schedules which are the best for problem $\mathcal{J}/a_i \le p_i \le b_i/\Phi$ with fixed (e.g. expected) processing times. As it has been shown for the traveling salesman problem [Lib99, LvdPSvdV96, vdP97] and for linear binary programming [PZ76, WJ88], the running time of such a branch-and-bound variant grows relatively slowly with k.

We develop also a branch-and-bound method ($B\&B2$) for constructing all schedules which may be optimal if the processing times vary between given lower and upper bounds. Unfortunately, both algorithms $B\&B1$ and $B\&B2$ may construct some redundant schedules, which are not necessarily contained in a minimal solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\Phi$. To reject such redundant schedules, we use a deeper (but more time-consuming) stability analysis of an optimal schedule. The last section of this chapter contains some concluding remarks.

3.2 Characterization of a Solution

A characterization of a solution Λ of problem $\mathcal{J}/a_i \le p_i \le b_i/\Phi$ which is a proper subset of the set $\Lambda(G), \Lambda \subset \Lambda(G)$, may be obtained on the basis of the dominance relation $\preceq_D$ introduced in Section 3.1. Necessary and sufficient conditions for a solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ have been derived in [LSSW98].

Theorem 3.1 *The set $\Lambda \subset \Lambda(G)$ is a solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\Phi$ if and only if there exists a finite covering of the polytope T by convex closed domains $D_j \subset R^q_+$: $T \subseteq \cup_{j=1}^d D_j,\ d \le |\Lambda|$, such that for any digraph $G_k \in \Lambda(G)$ and for any domain $D_j,\ j = 1, 2, \dots, d$, there exists a digraph $G_s \in \Lambda$ for which dominance relation $G_s \preceq_{D_j} G_k$ holds.*

PROOF. *Sufficiency.* For any fixed vector $p \in T$, one can find a domain $D_j, 1 \le j \le d$, such that $p \in D_j$. From the condition of Theorem 3.1, it follows that for any digraph $G_k \in \Lambda(G)$, there exists a digraph G_s such that dominance relation $G_s \preceq_{D_j} G_k$ holds. Hence, we have $\Phi^p_s \le \Phi^p_k$ and so inequality $\min\{\Phi^p_s : G_s \in \Lambda \subseteq \Lambda(G)\} \le \Phi^p_k$ holds for each $k = 1, 2, \dots, \lambda$. Consequently, for any vector $p \in T$ of the processing times, set Λ contains an optimal digraph.

Necessity. Let the set $\Lambda \subseteq \Lambda(G)$ be a solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\Phi$. We define a subset Λ' of the set Λ such that each digraph $G_s \in \Lambda'$ is optimal for at least one vector $p \in T$ of the processing times. For each digraph $G_s \in \Lambda$, one can define its *stability region*, i.e. the set of all vectors $p \in T \subseteq R^q_+$ for which digraph G_s is optimal. Let D_s be the intersection of the stability region of the digraph G_s with the polytope T:

$$D_s = \{p \in R^q_+ \ : \ \Phi^p_s \le \Phi^p_k,\ k = 1, 2, \dots, \lambda\} \cap T. \tag{3.1}$$

Since Λ' is a solution, we have $T \subseteq \cup_{j=1}^{|\Lambda'|} D_j \subset R^q_+$ and for each digraph $G_k \in \Lambda(G)$ and each domain D_s, the dominance relation $G_s \preceq_{D_s} G_k$ holds. The inclusion $G_s \in \Lambda'$ implies $D_s \neq \emptyset$. From inequality (3.1) it follows that D_s is a closed set.

Note that, if digraph G_s is optimal for the vector p, it remains optimal for a feasible vector αp with any positive real number $\alpha > 0$. Consequently, the stability region is convex and thus D_s is convex, too (as the intersection of convex sets).

$\diamond$

Theorem 3.1 implies the following corollary from [LSSW98] which characterizes a single-element solution of problem $\mathcal{J}/a_i \leq p_i \leq b_i/\Phi$, which is necessarily a minimal solution.

Corollary 3.1 *The equality $\Lambda^T(G) = \{G_s\}$ holds if and only if dominance relation $G_s \preceq_T G_k$ holds for any digraph $G_k \in \Lambda(G)$.*

A minimal solution which includes more than one digraph may be characterized on the basis of the strong dominance relation $\prec_D$ as follows. (A similar theorem formulated for problem $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$ has been proven in [LSSW98].)

Theorem 3.2 *Let the set $\Lambda^*(G)$ be a solution of problem $\mathcal{J}/a_i \leq p_i \leq b_i/\Phi$ with $|\Lambda^*(G)| > 1$. This solution is minimal if and only if for each digraph $G_s \in \Lambda^*(G)$, there exists a vector $p^{(s)} \in T$ such that the strong dominance relation $G_s \prec_{p^{(s)}} G_k$ holds for each digraph $G_k \in \Lambda^*(G) \setminus \{G_s\}$.*

Proof. *Sufficiency.* If the condition of Theorem 3.2 holds, then for any digraph $G_s \in \Lambda^*(G)$, the set $\Lambda^*(G) \setminus \{G_s\}$ is no longer a solution of problem $\mathcal{J}/a_i \leq p_i \leq b_i/\Phi$ since for the above vector $p^{(s)} \in T$, inequality $\Phi_s^{p^{(s)}} < \Phi_k^{p^{(s)}}$ holds for each digraph $G_k \in \Lambda^*(G) \setminus \{G_s\}$.

Necessity. We assume that $\Lambda^*(G)$ is a minimal solution but the condition of Theorem 3.2 does not hold, i.e. there exists a digraph $G_s \in \Lambda^*(G)$ such that for each vector $p^{(s)} \in T$, there exists a digraph $G_k \in \Lambda^*(G) \setminus \{G_s\}$ for which the strong dominance relation $G_s \prec_{p^{(s)}} G_k$ does not hold, i.e. we have $\Phi_s^p \geq \Phi_k^p$. It follows that the set $\Lambda^*(G) \setminus \{G_s\}$ is also a solution of problem $\mathcal{J}/a_i \leq p_i \leq b_i/\Phi$ (since the set $\Lambda^*(G)$ is supposed to be a solution). Thus, we get a contradiction to the assumption that solution $\Lambda^*(G)$ is minimal.

$\diamond$

Section 3.4 deals with different algorithms for finding a solution and a minimal solution on the basis of an explicit or an implicit schedule enumeration. All algorithms developed are based on the fact that a digraph $G_s \in \Lambda(G)$ being optimal for the fixed vector $p \in R_+^q$ of the processing times, generally remains optimal within some neighborhood of the point p in the space R_+^q (see Section 1.2). In other words, digraph G_s dominates all digraphs in a neighborhood of the point p. We consider the closed ball $O_r(p) \subset R^q$ with the center $p \in T$ and the radius $r > 0$ as the neighborhood of the point $p \in T \subset R_+^q$ in the space R^q. Next, we rewrite some basis notions using dominance relation $\preceq_D$.

The closed ball $O_r(p)$ is called a *stability ball* of the digraph G_s if this digraph dominates all digraphs $G_k \in \Lambda(G)$ in the polytope $T^* = O_r(p) \cap T$, i.e. if $G_s \preceq_{T^*} G_k$ for each $G_k \in \Lambda(G)$ (in this case, from Corollary 3.1 it follows that $\Lambda^{T^*}(G) = \{G_s\}$). As it was noted in Section 2.2, the radius r of a stability ball may be interpreted as the *error* of the given processing times $p = (p_{11}, p_{12}, \ldots, p_{nn_n}) \in R_+^q$ such that for all variable processing times $x = (x_{11}, x_{12}, \ldots, x_{nn_n}) \in R_+^q$ with $p_{ij} - r \leq x_{ij} \leq p_{ij} + r$ digraph G_s remains the best. The maximal value of such a radius is of particular importance for finding a minimal solution $\Lambda^T(G)$. Similarly to Definition 2.2 of the relative stability radius for the makespan criterion, we give the definition of the relative stability radius for the mean flow time criterion.

Definition 3.2 *Assume that for each vector $p' \in O_\varrho(p) \cap T$ digraph $G_s \in B \subseteq \Lambda(G)$ with the vector p' of weights has the minimal critical sum of weights $L_s^{p'}$ among all digraphs of the set B. The maximal value of the radius ϱ of such a ball $O_\varrho(p)$ is denoted by $\overline{\varrho}_s^B(p \in T)$ and is called the relative stability radius of the digraph G_s with respect to the polytope T for criterion $\sum \mathcal{C}_i$.*

Remark 3.1 From Definition 3.1 and Definition 3.2, it follows that the *relative stability radius* $\overline{\varrho}_s^B(p \in T)$ of the digraph $G_s \in B$ is equal to the maximal value of the radius ϱ of a ball $O_\varrho(p)$ such that for each digraph $G_k \in B \subseteq \Lambda(G)$ dominance relation $G_s \preceq_{T^*} G_k$ holds, if $T^* = O_\varrho(p) \cap T$.

As it follows from Section 1.4 (which deals with the stability radius $\overline{\varrho}_s(p)$ (see conditions (1.11) and (1.12))), to find the relative stability radius $\overline{\varrho}_s^B(p \in T)$ for problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ it is sufficient to construct a vector $\overline{x} = (\overline{x}_{11}, \overline{x}_{12}, \dots, \overline{x}_{nn_n}) \in T \subseteq R_+^q$ which satisfies the following three conditions.
1') There exists a digraph $G_k(p) \in B, k \ne s$, such that $L_s^{\overline{x}} = L_k^{\overline{x}}$, i.e.

$$\sum_{i=1}^{n} \max_{\mu \in H_s^i} l^{\overline{x}}(\mu) = \sum_{i=1}^{n} \max_{\nu \in H_k^i} l^{\overline{x}}(\nu). \tag{3.2}$$

2') For any given real $\epsilon > 0$, which may be as small as desired, there exists a vector $\overline{p}^\epsilon \in T$ such that $d(\overline{x}, \overline{p}^\epsilon) = \epsilon$ and $L_s^{\overline{p}^\epsilon} > L_k^{\overline{p}^\epsilon}$, i.e. inequality

$$\sum_{i=1}^{n} \max_{\mu \in H_s^i} l^{\overline{p}^\epsilon}(\mu) > \sum_{i=1}^{n} \max_{\nu \in H_k^i} l^{\overline{p}^\epsilon}(\nu) \tag{3.3}$$

is satisfied for at least one digraph $G_k(p) \in B$.
3') The distance $d(p, \overline{x})$ achieves its minimal value among the distances between the vector p and the other vectors in the polytope T which satisfy both above conditions 1' and 2'.

Next, we describe the calculation of the relative stability radius $\overline{\varrho}_s^B(p \in T)$ using the above notation of the dominance relation. To this end, we prove Lemma 3.1 below about the dominance relation $\preceq_T$, and then we derive a formula for the calculation of the relative stability radius $\overline{\varrho}_s^B(p \in T)$ which is presented in Theorem 3.3.

If $\Lambda^T(G) = \{G_s\}$, then digraph G_s dominates all digraphs in the polytope T (see Corollary 3.1). In such a case, we assume that $\overline{\varrho}_s^{\Lambda(G)}(p \in T) = \infty$, since digraph G_s remains the best for all variable feasible vectors $x \in T$ of the processing times. Otherwise, there exists a digraph $G_k \in \Lambda(G)$ such that dominance relation $G_s \preceq_T G_k$ does not hold, and from Corollary 3.1 and Remark 3.1, it follows that the stability radius $\overline{\varrho}_s^{\Lambda(G)}(p \in T)$ has to be finite, i.e. there exists a vector $\overline{p}^\epsilon \in T$ such that inequality (3.3) holds. To calculate the stability radius $\overline{\varrho}_s^B(p \in T)$, $B \subseteq \Lambda(G)$, we will consider digraphs $G_k \in B$ such that dominance relation $G_s \preceq_T G_k$ does not hold, and for each of these digraphs G_k, we will look for the vector $\overline{p}^\epsilon \in T$ which is the closest to p, among all vectors for which inequality (3.3) holds (see condition 3'). The following lemma allows to restrict the set of digraphs $G_k \in B$ which have to be considered for any regular criterion.

Lemma 3.1 *Digraph $G_s \in B$ dominates digraph $G_k \in B$ in the polytope T if (only if) the following inequality (3.4) holds (inequalities (3.5) hold, respectively):*

$$\Phi_s^b \le \Phi_k^a \tag{3.4}$$

$$(\Phi_s^a \le \Phi_k^a, \ \Phi_s^b \le \Phi_k^b). \tag{3.5}$$

PROOF. *Sufficiency.* Since the objective function is non-decreasing, it follows from inequality (3.4) that

$$\Phi_s^x \leq \Phi_s^b \leq \Phi_k^a \leq \Phi_k^x$$

for any vector $x \in T$. Therefore, dominance relation $G_s \preceq_T G_k$ holds.

Necessity. Dominance relation $G_s \preceq_T G_k$ means that inequality $\Phi_s^x \leq \Phi_k^x$ holds for any vector $x \in T$ and thus for both vectors $a \in T$ and $b \in T$, too, i.e. inequalities (3.5) hold.

$\diamond$

Similar theorems and the above lemma formulated for the special case of problem $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$ have been proven in [LSSW98].

The test of inequalities (3.4) and (3.5) takes $O(q^2)$ elementary steps, however, there is a 'gap' between the necessary and sufficient conditions of Lemma 3.1, if $\Phi_s^a \neq \Phi_s^b$. To overcome this gap for problem $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$, we are forced to compare the sets $\Omega_s^v,\ v = 1, 2, \ldots, \omega_s^T$, with the sets $\Omega_k^u,\ u = 1, 2, \ldots, \omega_k^T$, since we do not know a priori which set will be critical. First, we will find a vector $\overline{x} = (\overline{x}_{11}, \overline{x}_{12}, \ldots, \overline{x}_{nn_n}) \in T$, which is the closest to the vector $p \in T$ such that $L_s^{\overline{x}} = L_k^{\overline{x}}$ (see condition 1' above). For the desired vector $\overline{x}$, the value $\sum_{\nu \in \Omega_k^u} l^{\overline{x}}(\nu)$ for each set $\Omega_k^u,\ u = 1, 2, \ldots, \omega_k^T$, has to be not greater than the value $\sum_{\mu \in \Omega_s^v} l^{\overline{x}}(\mu)$ for at least one set $\Omega_s^v,\ v = 1, 2, \ldots, \omega_s^T$. If the opposite inequality holds for the given vector $p \in T$, i.e. if $\sum_{\mu \in \Omega_s^v} l^p(\mu) < \sum_{\nu \in \Omega_k^u} l^p(\nu)$, we can calculate the value

$$r = \frac{\sum_{\nu \in \Omega_k^u} l^p(\nu) - \sum_{\mu \in \Omega_s^v} l^p(\mu)}{\sum_{O_{ij} \in Q} |n_{ij}(\Omega_k^u) - n_{ij}(\Omega_s^v)|} \tag{3.6}$$

(where $n_{ij}(\Omega_k^u)$ is the number of copies of operation O_{ij} in the multiset $\{[\nu] :\ \nu \in \Omega_k^u\}$) in order to obtain vector $\overline{x}$ with

$$\sum_{\mu \in \Omega_s^v} l^{\overline{x}}(\mu) = \sum_{\nu \in \Omega_k^u} l^{\overline{x}}(\nu). \tag{3.7}$$

It is easy to convince that equality (3.7) holds for the vector $\overline{x}$ obtained from the vector p by adding the value r calculated in (3.6) to all components p_{ij} with $n_{ij}(\Omega_k^u) < n_{ij}(\Omega_s^v)$ and by subtracting the same value r from all components p_{ij} with $n_{ij}(\Omega_k^u) > n_{ij}(\Omega_s^v)$. Note that for the above vector $\overline{x}$, the inclusion $\overline{x} \in T$ need not hold. To guarantee this inclusion, we have to look for a vector $\overline{x}$ in the form $\overline{x} = p(r) = (p_{11}(r), p_{12}(r), \ldots, p_{nn_n}(r))$, where

$$\overline{x}_{ij} = p_{ij}(r) = \begin{cases} p_{ij} + \min\{r, b_{ij} - p_{ij}\}, & \text{if } n_{ij}(\Omega_k^u) < n_{ij}(\Omega_s^v), \\ p_{ij} - \min\{r, p_{ij} - a_{ij}\}, & \text{if } n_{ij}(\Omega_k^u) > n_{ij}(\Omega_s^v), \\ p_{ij}, & \text{if } n_{ij}(\Omega_k^u) = n_{ij}(\Omega_s^v). \end{cases} \tag{3.8}$$

Let $r_{\Omega_k^u, \Omega_s^v}$ denote the minimal distance between the given vector $p \in T$ and the desired vector $\overline{x} = p(r) \in T$ for which equality (3.7) holds: $r_{\Omega_k^u, \Omega_s^v} = d(p, p(r))$. Next, we show how to calculate this value $r_{\Omega_k^u, \Omega_s^v}$. To this end, we define the value

$$\Delta^{ij}(\Omega_s^v, \Omega_k^u) = \begin{cases} b_{ij} - p_{ij}, & \text{if } n_{ij}(\Omega_k^u) < n_{ij}(\Omega_s^v), \\ p_{ij} - a_{ij}, & \text{if } n_{ij}(\Omega_k^u) > n_{ij}(\Omega_s^v), \end{cases} \tag{3.9}$$

for each operation $O_{ij} \in N(\Omega_k^u, \Omega_s^v) = \{\cup_{\mu \in \Omega_k^u \cup \Omega_s^v} [\mu] :\ n_{ij}(\Omega_k^u) \neq n_{ij}(\Omega_s^v)\}$, which we put in non-decreasing order:

$$\Delta_1^{ij}(\Omega_s^v, \Omega_k^u) \leq \Delta_2^{ij}(\Omega_s^v, \Omega_k^u) \leq \ldots \leq \Delta_{|N(\Omega_s^v, \Omega_k^u)|}^{ij}(\Omega_s^v, \Omega_k^u). \tag{3.10}$$

Note that each value $\Delta^{ij}_{\alpha}(\Omega^v_s, \Omega^u_k)$ is calculated according to (3.9) for all different operations O_{ij}, and the subscript $\alpha = 1, 2, \ldots, |N(\Omega^v_s, \Omega^u_k)|$ indicates the location of value (3.9) in the above order. Let us define also the value

$$N_\alpha(\Delta) = |n_{ij}(\Omega^u_k) - n_{ij}(\Omega^v_s)|$$

for each $\Delta^{ij}_{\alpha}(\Omega^v_s, \Omega^u_k)$, $\alpha = 1, 2, \ldots, |N(\Omega^v_s, \Omega^u_k)|$, and let $\Delta^{ij}_0(\Omega^v_s, \Omega^u_k) = 0$ and $N_0(\Delta) = 0$.

From (3.8) and (3.10), it follows that equality (3.11) holds:

$$r_{\Omega^v_s, \Omega^u_k} = \max_{\beta = 0,1,\ldots,|N(\Omega^v_s, \Omega^u_k)|-1} \frac{\sum_{\nu \in \Omega^u_k} l^p(\nu) - \sum_{\mu \in \Omega^v_s} l^p(\mu) - \sum_{\alpha=0}^{\beta} \Delta^{ij}_{\alpha}(\Omega^v_s, \Omega^u_k)\ N_\alpha(\Delta)}{\sum_{O_{ij} \in Q} |n_{ij}(\Omega^u_k) - n_{ij}(\Omega^v_s)| - \sum_{\alpha=0}^{\beta} N_\alpha(\Delta)}. \quad (3.11)$$

To ensure equality $L^{\overline{x}}_s = L^{\overline{x}}_k$ for the digraph G_k and the vector $\overline{x} = p(r) \in T$, we have to repeat the calculations (3.8) - (3.11) for each set Ω^u_k, $u \in \{1, 2, \ldots, \omega^T_k\}$, with

$$\sum_{\nu \in \Omega^u_k} l^p(\nu) \geq L^p_s.$$

Then we have to take the maximum of $r_{\Omega^v_s, \Omega^u_k}$, for each set Ω^u_k, $u \in \{1, 2, \ldots, \omega^T_k\}$, and to take the minimum of the maximum obtained:

$$\overline{r}^B_{ks} = \min_{v \in \{1,2,\ldots,\omega^T_s\}} \ \max_{\substack{u \in \{1,2,\ldots,\omega^T_k\}, \\ \sum_{\nu \in \Omega^u_k} l^p(\nu) \geq L^p_s}} r_{\Omega^v_s, \Omega^u_k}. \quad (3.12)$$

Note that, if there exists a vector $\overline{x} \in T$ such that equality $L^{\overline{x}}_s = L^{\overline{x}}_k$ holds (see condition 1' above), nevertheless it may be that there exists no vector $\overline{p}^\epsilon \in T$ defined as in condition 2' such that $L^{\overline{p}^\epsilon}_s > L^{\overline{p}^\epsilon}_k$. However, as follows from Definition 3.1, only inequality (3.3) implies that digraph G_s does not dominate digraph G_k in the polytope T. Therefore, we look for a vector $\overline{p}^\epsilon \in T$ such that inequality (3.3) holds which may be rewritten in the following equivalent form:

$$\max_{v \in \{1,2,\ldots,\omega^T_s\}} \sum_{\mu \in \Omega^v_s} l^{\overline{p}^\epsilon}(\mu) > \max_{u \in \{1,2,\ldots,\omega^T_k\}} \sum_{\nu \in \Omega^u_k} l^{\overline{p}^\epsilon}(\nu). \quad (3.13)$$

Remark 3.2 It is easy to see that there exists a vector $\overline{p}^\epsilon \in T$ such that

$$\sum_{\mu \in \Omega^v_s} l^{\overline{p}^\epsilon}(\mu) > \sum_{\nu \in \Omega^u_k} l^{\overline{p}^\epsilon}(\nu) \quad (3.14)$$

if and only if inequality (3.14) holds for the vector $\overline{p}^\epsilon = p^* = (p^*_{11}, p^*_{12}, \ldots, p^*_{nn_n}) \in T$, where

$$p^*_{ij} = \begin{cases} b_{ij}, & \text{if } n_{ij}(\Omega^u_k) < n_{ij}(\Omega^v_s), \\ a_{ij}, & \text{if } n_{ij}(\Omega^u_k) > n_{ij}(\Omega^v_s), \\ p_{ij}, & \text{if } n_{ij}(\Omega^u_k) = n_{ij}(\Omega^v_s). \end{cases} \quad (3.15)$$

Indeed, all components of the vector $p^* \in T$ with $n_{ij}(\Omega^u_k) < n_{ij}(\Omega^v_s)$ are as large as possible in the polytope T and all components with $n_{ij}(\Omega^u_k) > n_{ij}(\Omega^v_s)$ are as small as possible in the polytope T (obviously, changing components with $n_{ij}(\Omega^u_k) = n_{ij}(\Omega^v_s)$ does not influence the difference $\sum_{\nu \in \Omega^u_k} l^x(\nu) - \sum_{\mu \in \Omega^v_s} l^x(\mu)$). Thus, we have to restrict the consideration of the sets Ω^v_s in inequality (3.13) to the subset Ω^*_{sk} of the set $\{\Omega^v_s :$

$v = 1, 2, \ldots, \omega_s^T\}$ defined as follows: Ω^*_{sk} is the set of all sets of representatives Ω_s^v, $v \in \{1, 2, \ldots, \omega_s^T\}$, for which inequality

$$\sum_{\mu \in \Omega_s^v} l^{p^*}(\mu) > \sum_{\nu \in \Omega_k^u} l^{p^*}(\nu) \tag{3.16}$$

holds for each set of representatives Ω_k^u, $u \in \{1, 2, \ldots, \omega_k^T\}$. Thus, sufficiency in Lemma 3.1 formulated for the mean flow time criterion ($\Phi_s^p = L_s^p$) was generalized in [LSSW98] as follows.

Lemma 3.2 *Digraph $G_s \in B$ dominates digraph $G_k \in B$ in the polytope T if $\Omega^*_{sk} = \emptyset$.*

Due to Lemma 3.2, we can rewrite equality (3.12) as follows:

$$\overline{r}_{ks}^B = \min_{\Omega_s^v \in \Omega^*_{sk}} \max_{\substack{u \in \{1,2,\ldots,\omega_k^T\},\\ \sum_{\nu \in \Omega_k^u} l^p(\nu) \geq L_s^p}} r_{\Omega_s^v, \Omega_k^u}. \tag{3.17}$$

To obtain the desired vector $\overline{p}^\epsilon \in T$, we have to calculate $\overline{r}_{ks}^B$ according to (3.17) for each digraph $G_k \in B$ which is not dominated by digraph G_s (i.e. if $G_s \not\preceq_T G_k$) and to take the minimum over all such digraphs G_k. We summarize the above discussion in the following theorem.

Theorem 3.3 *If we assume that digraph $G_s \in B \subseteq \Lambda(G)$ dominates all digraphs $G_k \in B$ at the vector $p \in T$ of the processing times, then equality*

$$\begin{aligned} \overline{\varrho}_s^B(p \in T) &= \min\{ \min_{\Omega_s^v \in \Omega^*_{sk}} \max_{\substack{u \in \{1,2,\ldots,\omega_k^T\},\\ \sum_{\nu \in \Omega_k^u} l^p(\nu) \geq L_s^p}} r_{\Omega_s^v, \Omega_k^u} \ : \ G_s \not\preceq_T G_k\} = \\ &= \min\{\overline{r}_{ks}^B : \ G_s \not\preceq_T G_k\} \end{aligned} \tag{3.18}$$

holds, where value $r_{\Omega_s^v, \Omega_k^u}$ is calculated according to (3.11).

The following corollary will help us to prove Theorem 3.5 below.

Corollary 3.2 *The value $r_{\Omega_s^{v^0}, \Omega_k^u}$ calculated according to (3.11) for the set $\Omega_s^{v^0} \in \Omega^*_{sk} \setminus \Omega_s(p)$ is strongly positive.*

Proof. Due to formula (3.18), we have to repeat the calculation (3.11) for each set $\Omega_s^v \in \Omega^*_{sk}$ and each set $\Omega_k^{u^0}, u^0 \in \{1, 2, \ldots, \omega_k^T\}$, such that $\sum_{\nu \in \Omega_k^{u^0}} l^p(\nu) \geq L_s^p$. Since there exists a set $\Omega_s^{v^0} \in \Omega^*_{sk} \setminus \Omega_s(p)$, i.e. $\sum_{\mu \in \Omega_s^{v^0}} l^p(\mu) < L_s^p$, the inequalities

$$\begin{aligned} & \sum_{\nu \in \Omega_k^{u^0}} l^p(\nu) - \sum_{\mu \in \Omega_s^{v^0}} l^p(\mu) \geq \\ \geq \ & \min_{\substack{u^0 \in \{1,2,\ldots,\omega_k^T\},\\ \sum_{\nu \in \Omega_k^{u^0}} l^p(\nu) \geq L_s^p}} \sum_{\nu \in \Omega_k^{u^0}} l^p(\nu) - \max_{\substack{v^0 \in \{1,2,\ldots,\omega_s^T\},\\ \sum_{\nu \in \Omega_s^{v^0}} l^p(\mu) < L_s^p}} \sum_{\mu \in \Omega_s^{v^0}} l^p(\mu) \geq \\ \geq \ & L_s^p - \max_{\substack{v^0 \in \{1,2,\ldots,\omega_s^T\},\\ \sum_{\nu \in \Omega_s^{v^0}} l^p(\mu) < L_s^p}} \sum_{\mu \in \Omega_s^{v^0}} l^p(\mu) > 0 \end{aligned}$$

hold. Therefore, due to the calculation of the value $r_{\Omega_s^{v^0}, \Omega_k^{u^0}}$, the numerator in (3.11) is strongly positive at least for $\beta = 0$. Since we have to take the maximum value among all values calculated for each $\beta = 0, 1, \dots, |N(\Omega_s^{v^0}, \Omega_k^{u^0})| - 1$ (see formula (3.11)), we get $r_{\Omega_s^{v^0}, \Omega_k^{u^0}} > 0$.

$\diamond$

Next, we present necessary and sufficient conditions for an infinitely large relative stability radius $\overline{\varrho}_s^B(p \in T)$ for problem $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$ if $B \subset \Lambda(G)$ and $T \subset R_+^q$, although problem $\mathcal{J}//\sum \mathcal{C}_i$ with $\lambda > 1$ cannot have an optimal digraph with an infinitely large stability radius $\overline{\varrho}_s(p)$ (see Remark 1.1). Recall that $\overline{\varrho}_s(p) = \overline{\varrho}_s^{\Lambda(G)}(p \in R_+^q)$.

Theorem 3.4 *For digraph $G_s \in B \subseteq \Lambda(G)$, we have $\overline{\varrho}_s^B(p \in T) = \infty$ if and only if $\Omega_{sk}^* = \emptyset$ for each digraph $G_k \in B$.*

PROOF. *Necessity.* Following the contradiction method, we suppose that $\overline{\varrho}_s^B(p \in T) = \infty$ but there exists a digraph $G_k \in B$ such that the set of representatives $\Omega_s^{v^0}, v^0 \in \{1, 2, \dots, \omega_s^T\}$, belongs to the set Ω_{sk}^*. It follows that the inequality

$$\sum_{\mu \in \Omega_s^{v^0}} l^{p^*}(\mu) > \sum_{\nu \in \Omega_k^u} l^{p^*}(\nu)$$

holds for the vector p^* calculated according to formula (3.15) for the set of representatives $\Omega_k^u, u \in \{1, 2, \dots, \omega_k^T\}$. Thus, due to Remark 3.2 there exists a vector $p' \in T$ such that inequality

$$\sum_{\mu \in \Omega_s^{v^0}} l^{p'}(\mu) > \sum_{\nu \in \Omega_k^u} l^{p'}(\nu)$$

holds.

Since this inequality holds for all sets Ω_k^u, $u \in \{1, 2, \dots, \omega_k^T\}$, this inequality holds for a critical set $\Omega_k^{u^*} \in \Omega_k(p)$, too. Therefore, we obtain

$$\sum_{\mu \in \Omega_s^{v^0}} l^{p'}(\mu) > \sum_{\nu \in \Omega_k^{u^*}} l^{p'}(\nu) = L_k^{p'}$$

and hence digraph G_s cannot be optimal for the processing times given by vector $p' \in T$.

We get a contradiction:

$$\overline{\varrho}_s^B(p \in T) < d(p, p') \leq \max_{O_{ij} \in Q} \{b_{ij} - p_{ij}, p_{ij} - a_{ij}\} < \infty.$$

Sufficiency. Due to Lemma 3.2, equality $\Omega_{sk}^* = \emptyset$ (valid for each digraph $G_k \in B$) implies that digraph $G_s \in B$ dominates all digraphs $G_k \in B$ in polytope T. Hence, inequality $L_s^{p'} \leq L_k^{p'}$ holds for each vector $p' \in T$ and so $\overline{\varrho}_s^B(p \in T) = \infty$.

$\diamond$

From the above proof of the necessity, we obtain an upper bound for the relative stability radius $\overline{\varrho}_s^B(p \in T)$.

Corollary 3.3 *If $\overline{\varrho}_s^B(p \in T) < \infty$, then $\overline{\varrho}_s^B(p \in T) \leq \max\{\{b_{ij} - p_{ij}, p_{ij} - a_{ij}\} : O_{ij} \in Q\}$.*

Moreover, we can strengthen Corollary 3.1 as follows.

Corollary 3.4 *The following propositions are equivalent:*

1) $\Lambda^T(G) = \{G_s\}$,

2) $\overline{\varrho}_s^{\Lambda(G)}(p \in T) = \infty$,

3) $G_k \in \Lambda(G) \;\Rightarrow\; G_s \preceq_T G_k$,

4) $G_k \in \Lambda(G) \;\Rightarrow\; \Omega^*_{sk} = \emptyset$.

To present necessary and sufficient conditions for $\overline{\varrho}_s^B(p \in T) = 0$, we need the following auxiliary lemma proven in [BSW96]. Let Ω_k denote the set $\{\Omega_k^u : \; u = 1, 2, \ldots, \omega_k\}$.

Lemma 3.3 *If* $\Omega_k \neq \Omega_k(p)$, *the inclusion* $\Omega_k(p') \subseteq \Omega_k(p)$ *holds for any vector* $p' \in O_\epsilon(p) \cap R_+^q$ *with* $\overline{\epsilon}_k > \epsilon > 0$ *defined as follows:*

$$\overline{\epsilon}_k = \frac{1}{qn} \min\Big\{L_k^p - \sum_{\nu \in \Omega_k^u} l^p(\nu) : \; \Omega_k^u \in \Omega_k \backslash \Omega_k(p)\Big\}. \tag{3.19}$$

The following theorem is a generalization of Theorem 1.6.

Theorem 3.5 *Let* G_s *be an optimal digraph of problem* $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$ *with the minimal objective function value* L_s^p, $p \in T$, *within the set* $B \subseteq \Lambda(G)$ *of feasible digraphs. The equality* $\overline{\varrho}_s^B(p \in T) = 0$ *holds if and only if the following three conditions hold:*
1) there exists a digraph $G_k \in B$ *such that* $L_s^p = L_k^p$, $k \neq s$,
2) the set $\Omega_s^{v^*} \in \Omega^*_{sk} \cap \Omega_s(p)$ *is such that for any set* $\Omega_k^{u^*} \in \Omega_k(p)$, *there exists an operation* $O_{ij} \in Q$ *for which condition*

$$n_{ij}(\Omega_s^{v^*}) \geq n_{ij}(\Omega_k^{u^*}) \tag{3.20}$$

holds (or condition

$$n_{ij}(\Omega_s^{v^*}) \leq n_{ij}(\Omega_k^{u^*}) \tag{3.21}$$

holds) and
3) inequality (3.20) (or inequality (3.21), respectively) is satisfied as a strict one for at least one set $\Omega_k^{u^0} \in \Omega_k(p)$: $n_{ij}(\Omega_s^{v^*}) > n_{ij}(\Omega_k^{u^0})$ *(or* $n_{ij}(\Omega_s^{v^*}) < n_{ij}(\Omega_k^{u^0})$*).*

Proof. *Necessity.* We prove necessity by contradiction. Assume that $\overline{\varrho}_s^B(p \in T) = 0$ but the conditions of the theorem are not satisfied. We consider four cases *i*, *ii*, *iii* and *iv* of violating these conditions.

i) Assume that there does not exist another optimal digraph $G_k \in B$ such that $L_s^p = L_k^p, k \neq s$. If $B \setminus \{G_s\} \neq \emptyset$, we can calculate the value

$$\overline{\epsilon}^* = \frac{1}{qn} \min_{t \neq s}(L_t^p - L_s^p), \tag{3.22}$$

which is strictly positive since $L_s^p < L_t^p$ for each $G_t \in B, t \neq s$. Using Lemma 3.3, one can verify that for any real ϵ, which satisfies the inequalities $0 < \epsilon < \overline{\epsilon}^*$, the difference in the right-hand side of equality (3.22) remains positive when vector p is replaced by any vector $p^0 \in O_\epsilon(p) \cap T$. Indeed, for any $v \in \{1, 2, \ldots, \omega_s^T\}$, the cardinality of the set Ω_s^v may be at most equal to qn: $|\Omega_s^v| \leq qn$. Thus, the difference $L_t^p - L_s^p = L_t^p - \max_{v \in \{1,2,\ldots,\omega_s^T\}} \sum_{\nu \in \Omega_s^v} l^p(\nu)$ may not be 'overcome' by a vector p^0 if $d(p, p^0) < \overline{\epsilon}^*$. Hence, we conclude that digraph G_s remains optimal for any vector $p^0 = (p_{11}^0, p_{12}^0, \ldots, p_{nn_n}^0) \in T$

of the processing times provided that $d(p, p^0) \leq \epsilon < \overline{\epsilon}^*$. Therefore, we have $\overline{\varrho}_s^B(p \in T) \geq \overline{\epsilon}^* > \epsilon > 0$ which contradicts the assumption $\overline{\varrho}_s^B(p \in T) = 0$.

ii) Assume that there exists a digraph $G_k \in B$ such that $L_s^p = L_k^p, k \neq s$, and $\Omega_{sk}^* \cap \Omega_s(p) = \emptyset$. Note that $\Omega_{st}^* \neq \emptyset$ for all digraphs $G_t \in B, t \neq s$. Otherwise, we get $\overline{\varrho}_s^B(p \in T) = \infty$ due to Theorem 3.4.

Assume that there exists a set $\Omega_{st}^* \neq \emptyset$ for the digraphs G_s and G_t with $L_t^p > L_s^p$, i.e. there exists a set $\Omega_s^{v'} \in \Omega_{st}^*$. Similarly to the proof of Corollary 3.2, we can show that all values $r_{\Omega_s^{v'}, \Omega_t^u}$ calculated for each digraph G_t with $L_t^p > L_s^p$ cannot be equal to zero: We obtain a strongly positive numerator in formula (3.11) at least for $\beta = 0$:

$$\sum_{\nu \in \Omega_t^u} l^p(\nu) - \sum_{\mu \in \Omega_s^{v'}} l^p(\mu) > 0.$$

Therefore, the maximum taken according to (3.11) is also strongly positive, i.e. $r_{\Omega_s^{v'}, \Omega_t^u} > \epsilon > 0$, where we can choose any ϵ such that the inequality

$$\epsilon < \min\Big\{\overline{\epsilon}_s,\, \overline{\epsilon}_k,\, \frac{1}{qn} \min_{\substack{G_t \in B, \\ L_t^p > L_s^p}} (L_t^p - L_s^p)\Big\} \tag{3.23}$$

is satisfied. This means that only in the case of the calculation of the value $r_{\Omega_s^v, \Omega_k^u}$ for the optimal digraphs $G_k \in B, L_k^p = L_s^p$, with $\Omega_{sk}^* \neq \emptyset$ we can obtain $r_{\Omega_s^v, \Omega_k^u} = 0$.

Assume that there exists a set $\Omega_{sk}^* \neq \emptyset$ for the digraphs G_s and G_k with $L_k^p = L_s^p$, i.e. there exists a set $\Omega_s^{v''} \in \Omega_{sk}^*$. In this case, we can set

$$\epsilon' = \min\Big\{\overline{\epsilon}_s, \overline{\epsilon}_k, \frac{1}{qn} \min\Big\{L_s^p - \max \sum_{\mu \in \Omega_s^v} l^p(\mu) : \sum_{\nu \in \Omega_s^v} l^p(\mu) < L_s^p\Big\}\Big\}.$$

Taking into account our assumption that for each digraph $G_k \in B, L_s^p = L_k^p, k \neq s$, the set $\Omega_{sk}^* \cap \Omega_s(p)$ is empty, it follows from the proof of Corollary 3.2 that $r_{\Omega_s^{v''}, \Omega_k^u} > \epsilon' > 0$.

Hence, for all digraphs $G_t, L_t^p \geq L_s^p$, inequality $\overline{r}_{ts}^B > \min\{\epsilon, \epsilon'\}$ holds, where the value $\overline{r}_{ts}^B$ is calculated due to formula (3.17) using the value $r_{\Omega_s^v, \Omega_t^u} > 0$. Therefore, the relative stability radius satisfies the following inequalities: $\overline{\varrho}_s^B(p \in T) > \min\{\epsilon, \epsilon'\} > 0$, which contradicts the above assumption $\overline{\varrho}_s^B(p \in T) = 0$.

iii) Assume that there exists a digraph $G_k \in B$ such that $L_s^p = L_k^p, k \neq s$, and for any set $\Omega_s^{v^*} \in \Omega_{sk}^* \cap \Omega_s(p)$ there exists a set $\Omega_k^{u^*} \in \Omega_k(p)$ such that $n_{ij}(\Omega_s^{v^*}) = n_{ij}(\Omega_k^{u^*})$ for any operation $O_{ij} \in Q$.

In this case, we can take any ϵ that satisfies the inequality (3.23). Due to $\epsilon < \overline{\epsilon}_s$, we get from Lemma 3.3 that equality

$$L_s^{p^0} = \max_{\Omega_s^{v^*} \in \Omega_s(p^0)} \sum_{\mu \in \Omega_s^{v^*}} l^{p^0}(\mu) = \max_{\Omega_s^{v^*} \in \Omega_s(p)} \sum_{\mu \in \Omega_s^{v^*}} l^{p^0}(\mu) \tag{3.24}$$

holds for any vector $p^0 \in O_\epsilon(p) \cap T$.

On the other hand, since there exists a set $\Omega_k^{u^*} \in \Omega_k(p)$ such that $n_{ij}(\Omega_s^{v^*}) = n_{ij}(\Omega_k^{u^*})$, $O_{ij} \in Q$, for any set $\Omega_s^{v^*} \in \Omega_{sk}^* \cap \Omega_s(p)$ and for any digraph $G_k, L_s^p = L_k^p$, we obtain the inequality

$$\max_{\Omega_s^{v^*} \in \Omega_s(p)} \sum_{\mu \in \Omega_s^{v^*}} l^{p^0}(\mu) \leq \max_{\Omega_k^u \in \Omega_k(p)} \sum_{\nu \in \Omega_k^{u^*}} l^{p^0}(\nu),$$

because of $\epsilon < \overline{\epsilon}_s$ and $\epsilon < \overline{\epsilon}_k$. Therefore, due to (3.24) we have

$$L_s^{p^0} \leq \max_{\Omega_k^u \in \Omega_k(p)} \sum_{\nu \in \Omega_k^{u^*}} l^{p^0}(\nu) \tag{3.25}$$

for any optimal digraph $G_k,\ k \neq s$. Since

$$\epsilon < \frac{1}{qn} \min_{L_t^p \neq L_s^p} \{L_t^p - L_s^p\},$$

the condition $L_t^p \neq L_s^p$ implies $L_t^{p^0} \neq L_s^{p^0}$. So taking into account (3.23) and the latter implication, we conclude that the digraph G_s becomes an optimal digraph for any vector $p^0 \in T$, provided that $d(p, p^0) \leq \epsilon$. Consequently, we have $\overline{\varrho}_s^B(p \in T) \geq \epsilon > 0$, which contradicts the assumption $\overline{\varrho}_s^B(p \in T) = 0$.

iv) Assume that conditions 1 and 2 of Theorem 3.5 hold. More exactly, there exists a digraph $G_k \in B$ such that $L_s^p = L_k^p, k \neq s$, and one of the two cases of condition 2 and one of the two cases of condition 3 hold. Assume that for any set $\Omega_s^{v^*} \in \Omega_{sk}^* \cap \Omega_s(p)$, there exists a set $\Omega_k^{u^*} \in \Omega_k(p)$ such that for any operation $O_{ij} \in Q$ with $n_{ij}(\Omega_s^{v^*}) > n_{ij}(\Omega_k^{u^*})$, there exists a set $\Omega_k^{u^0} \in \Omega_k(p)$ with $n_{ij}(\Omega_s^{v^*}) < n_{ij}(\Omega_k^{u^0})$.

Arguing in the same way as in case *iii*, we can show that $\overline{\varrho}_s^B(p \in T) \geq \epsilon > 0$, where ϵ is as in (3.23), since for any vector $p^0 \in O_\epsilon(p) \cap T$, the value $\sum_{\mu \in \Omega_s^{v^*}} l^{p^0}(\mu)$ is less than or equal to the value $\sum_{\nu \in \Omega_k^{u^*}} l^{p^0}(\nu)$ or to the value $\sum_{\nu \in \Omega_k^{u^0}} l^{p^0}(\nu)$.

Sufficiency. We show that, if the conditions of Theorem 3.5 are satisfied, then $\overline{\varrho}_s^B(p \in T) < \epsilon$ for any given $\epsilon > 0$. First, we make the following remark.

Remark 3.3 In the trivial case of $a_{ij} = b_{ij}$ for each operation $O_{ij} \in Q$, the set $\Omega_{sk}^* \cap \Omega_s(p)$ is empty, since in this case the vector p is equal to the vector p^* constructed according to (3.15), and the strong inequality (3.16) does not hold.

We construct a vector $p' = (p'_{11}, p'_{12}, \ldots, p'_{nn_n}) \in T$ with components $p'_{ij} \in \{p_{ij}, p_{ij} + \epsilon', p_{ij} - \epsilon'\}$, where $\epsilon' = \min\{\epsilon, \overline{\epsilon}_k, \overline{\epsilon}_{min}\}$ with the value $\overline{\epsilon}_k > 0$ defined in (3.19), and

$$\overline{\epsilon}_{min} = \max\{0,\ \min\{\min\{p_{ij} - a_{ij} :\ p_{ij} > a_{ij}, O_{ij} \in Q\},\ \min\{b_{ij} - p_{ij} :\ b_{ij} > p_{ij}, O_{ij} \in Q\}\}\},$$

using the following rule: For each $\Omega_k^{u^*} \in \Omega_k(p)$, mentioned in Theorem 3.5, we set $p'_{ij} = p_{ij} + \epsilon'$, if inequalities (3.20) hold, or we set $p'_{ij} = p_{ij} - \epsilon'$, if inequalities (3.21) hold.

More precisely, we can choose ϵ' as follows: If $\Omega_k \neq \Omega_k(p)$, then $\overline{\epsilon}_k > 0$, and we can choose ϵ' such that $0 < \epsilon' < \min\{\epsilon, \overline{\epsilon}_k, \overline{\epsilon}_{min}\}$. Otherwise, if $\Omega_k = \Omega_k(p)$, we choose ϵ' such that $0 < \epsilon' < \min\{\overline{\epsilon}, \overline{\epsilon}_{min}\}$. Such choices are possible since in both above cases, inequality $\overline{\epsilon}_{min} > 0$ holds due to Remark 3.3. Note that $\epsilon' > 0$ since $p_{ij} > 0, O_{ij} \in Q$. The following arguments are the same for both cases of the choice of ϵ'.

After changing at most $|\Omega_k(p)|$ components of the vector p according to this rule, we obtain a vector p' of the processing times for which inequality

$$\sum_{\mu \in \Omega_s^{v^*}} l^{p'}(\mu) > \sum_{\nu \in \Omega_k^{u^*}} l^{p'}(\nu)$$

holds for each set $\Omega_k^{u^*} \in \Omega_k(p)$. Due to $\epsilon' \leq \overline{\epsilon}_{min}$, we have $p' \in T$. Since $\epsilon' \leq \overline{\epsilon}_k$, we have

$$L_k^{p'} \ = \max_{u \in \{1,2,\ldots,\omega_k^T\}} \sum_{\nu \in \Omega_k^u} l^{p'}(\nu) \ =$$

$$= \max_{\Omega_k^u \in \Omega_k(p)} \sum_{\nu \in \Omega_k^u} l^{p'}(\nu) =$$

$$= \sum_{\nu \in \Omega_k^{u^*}} l^{p'}(\nu) < \sum_{\mu \in \Omega_s^{v^*}} l^{p'}(\mu) \leq L_s^{p'}.$$

Thus, we conclude that digraph G_s is not optimal for the vector $p' \in T$ with $d(p, p') = \epsilon'$ which implies $\overline{\varrho}_s^B(p \in T) < \epsilon' \leq \epsilon$.

$\diamond$

Corollary 3.5 *If $G_s \in B$ is a unique optimal digraph for the vector $p \in T$, then $\overline{\varrho}_s^B(p \in T) > 0$.*

From Theorem 3.5 we obtain the following lower bound for the relative stability radius $\overline{\varrho}_s^B(p \in T)$.

Corollary 3.6 *If $G_s \in B$ is an optimal digraph, then $\overline{\varrho}_s^B(p \in T) \geq \overline{\epsilon}^*$, where $\overline{\epsilon}^*$ is calculated according to (3.22).*

PROOF. If there exists a digraph $G_k \in B$ such that $L_s^p = L_k^p$, $k \neq s$, the equality $\overline{\varrho}_s^B(p \in T) \geq \overline{\epsilon}^* = 0$ holds due to Definition 3.2. Otherwise, inequality $\overline{\varrho}_s^B(p \in T) \geq \overline{\epsilon}^*$ follows from the above proof of necessity (see case i).

$\diamond$

Example 3.1 (continued). Returning to the Example 3.1 and using Theorem 3.3, we can calculate the relative stability radius of the digraph $G_1 \in B \subseteq \Lambda(G), |B| = 12$, for the vector $p = p^0 = (70, 30, 60, 20, 60, 70, 40, 30)$ of the processing times according to formula (3.18). After a pairwise comparison of the sets of representatives for the digraph G_1^T with those for the digraphs $G_2^T, G_3^T, \ldots, G_{12}^T$, we obtain the equality $\overline{\varrho}_1^B(p^0 \in T) = 3$, which means that digraph G_1 remains optimal at least for all vectors $p \in O_3(p^0) \cap T$ of the processing times. Due to the calculation of the stability radius, we show that only digraphs G_2 and G_5 may be better than digraph G_1 provided that vector p of the processing times belongs to the polytope T, and for all digraphs $G_k \in \Lambda(G)$ with $k \neq 2$ and $k \neq 5$, dominance relation $G_1 \preceq_T G_k$ holds. We also obtain the following equalities: $\overline{\varrho}_1^B(p^0 \in T) = \overline{r}_{21}^B = 3$, $\overline{\varrho}_1^{B \setminus \{G_2\}}(p^0 \in T) = \overline{r}_{51}^{B \setminus \{G_2\}} = 10$, where the values $\overline{r}_{k1}^B$ are calculated according to (3.17). Next, it follows from Theorem 3.4 that $\overline{\varrho}_1^{B \setminus \{G_2, G_5\}}(p^0 \in T) = \infty$.

Due to Theorem 3.1, the set $\Lambda^*(G) = \{G_1, G_2, G_5\}$ is a solution of problem $\mathcal{J}3/n = 3, a_i \leq p_i \leq b_i / \sum \mathcal{C}_i$, since there exists a covering of the polytope T by the domains $D_s = \{p \in R_+^8 : L_s^p \leq L_k^p, k = 1, 2 \ldots, \lambda\} \cap T$ with $s \in \{1, 2, 5\}$. More exactly, for any digraph $G_k \in \Lambda(G)$ and for any domain D_s, $s \in \{1, 2, 5\}$, there exists a digraph $G_s \in \Lambda^*(G)$ for which dominance relation $G_s \preceq_{D_s} G_k$ holds (since the dominance relation $G_1 \preceq_T G_k$ holds for each digraph $G_k \in \Lambda(G), k \neq 2, k \neq 5$, it follows that set $\{D_1, D_2, D_5\}$ is indeed a covering of the polytope T). Moreover, since for each digraph $G_s \in \Lambda^*(G)$ there exists a point (see vectors p^0, $\overline{p}^\epsilon$ and $\overline{x}$, given in Section 3.1), for which this digraph is the unique optimal one, it follows from Theorem 3.2 that solution $\Lambda^*(G) = \{G_1, G_2, G_5\}$ is minimal.

Note that from a practical point of view, it is more useful to consider a covering of the polytope T by nested balls $O_3(p^0), O_{10}(p^0)$ and $O_{r^*}(p^0)$, where r^* may be any real number no less than $\max\{b_{ij} - p^0_{ij}, p^0_{ij} - a_{ij} : i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, n_i\}$. Indeed, due to the calculation of the stability radius $\overline{\varrho}^B_1(p^0 \in T)$, we know that for each vector $p \in O_3(p^0)$ digraph G_1 is optimal. Moreover, for each vector $p \in O_{10}(p^0)$ at least one digraph G_1 or G_2 is optimal since $\overline{\varrho}^{B\setminus\{G_2\}}_1(p^0 \in T) = 10$. Finally, for each vector $p \in O_{r^*}(p^0)$ at least one digraph G_1, G_2 or G_5 is optimal since $\overline{\varrho}^{B\setminus\{G_2,G_5\}}_1(p^0 \in T) = \infty$.

Table 3.2: *Solution of problem $\mathcal{J}3/n{=}3, a_i \le p_i \le b_i/\sum \mathcal{C}_i$ with the initial vector $p^0 \in T$*

i	Set B	$\overline{\varrho}^B_1(p^0 \in T)$	Set Γ_i of competitive digraphs of digraph G_1
1	$B = \{G_1, G_2, \ldots, G_{12}\}$	3	$\{G_2\}$
2	$B \setminus \{G_2\}$	10	$\{G_5\}$
3	$B \setminus \{G_2, G_5\}$	∞	$\emptyset$

Remark 3.4 Solving problem $\mathcal{J}3/n = 3, a_i \le p_i \le b_i/\sum \mathcal{C}_i$ takes three iterations by the above algorithm (see Table 3.2). But similarly to the calculation of the relative stability radius and the construction of a solution of the scheduling problem with the makespan criterion (see Remark 2.4), we can construct a solution $\Lambda^*(G)$ for the mean flow time criterion in one scan as follows. We union one of the optimal digraphs G_s with all digraphs $G_k, k \neq s$, for which a nonempty set $\Omega^*_{sk} \neq \emptyset$ exists, i.e. for which the dominance relation $G_s \preceq_T G_k$ does not hold, and the union of these digraphs composes such a solution $\Lambda^*(G)$. In other words, a solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ is the union of an optimal digraph and of all its competitive digraphs $\Lambda^*(G) = \{G_s\} \cup \{G_k : G_s \not\preceq_T G_k\} = \{G_s\} \cup \{\cup^I_{i=1} \Gamma_i\}$, where Γ_i is the set of competitive digraphs of digraph G_s with respect to the set B in the iteration $i = 1, 2, \ldots, I$.

Next, we consider a small problem $\mathcal{J}3/n{=}2/\sum \mathcal{C}_i$ to illustrate the calculation of $\overline{\varrho}_1(p)$ by formulas (1.14) and (1.15). Then we calculate the relative stability radius for problem $\mathcal{J}3/n{=}2, a_i{\le}p_i{\le}b_i/\sum \mathcal{C}_i$. Notice that we use the same notations for different examples: Example 3.1 above and Example 2.1 below (without causing any confusion).

Example 2.1 (continued). Returning to the Example 2.1 from Section 2.1, let us consider the job shop problem with the mean flow time criterion $\mathcal{J}3/n{=}2/\sum \mathcal{C}_i$, whose input data are given by the weighted mixed graph $G(p)$ with $p = (75, 50, 40, 60, 55, 30)$, presented in Figure 2.1. Obviously, the set of all feasible digraphs $\Lambda(G)$ is the same (see Figure 2.2), but we number these digraphs in non-decreasing order of the objective function values: $L^p_1 \le L^p_2 \le \ldots \le L^p_5$ (see Figure 3.3). As we can see, digraph $G_1(p)$ is optimal for both criteria $\mathcal{C}_{max}$ and $\sum \mathcal{C}_i$. Next, we determine the stability radius $\overline{\varrho}_1(p)$ of this digraph.

To this end, we construct an auxiliary Table 3.3, where for each feasible digraph $G_k, k = 1, 2, \ldots, 5$, column 2 presents the sets Ω^u_k of representatives of the family of sets $(H^i_k)_{J_i \in J}$, column 3 presents the integer vector $n(\Omega^u_k) = (n_{11}(\Omega^u_k), n_{12}(\Omega^u_k), \ldots, n_{23}(\Omega^u_k))$, where the value $n_{ij}(\Omega^u_k)$ is equal to the number of vertices O_{ij} in the multiset $\{[\nu] : \nu \in \Omega^u_k\}$, and column 4 presents the value

$$\sum_{\nu \in \Omega^u_k} l^p(\nu) = \sum_{O_{ij} \in [\nu],\ \nu \in \Omega^u_k} p_{ij} \cdot n_{ij}(\Omega^u_k).$$

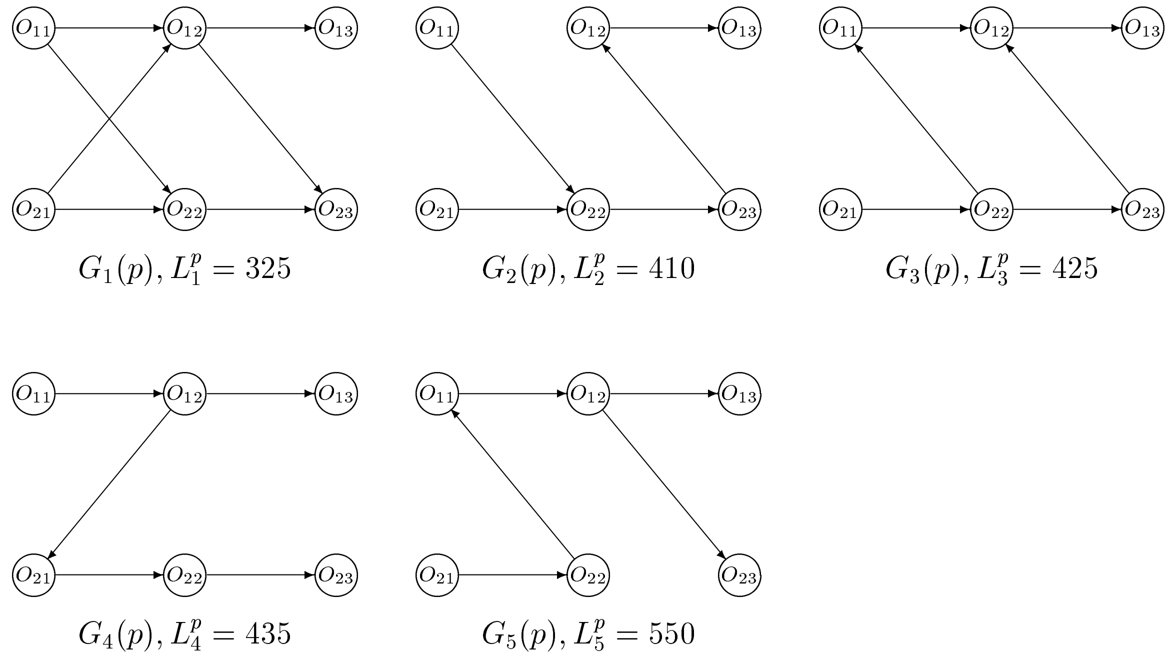


Figure 3.3: *Digraphs $\Lambda(G) = \{G_1, G_2, \ldots, G_5\}$ numbered in non-decreasing order of the objective function values $\sum \mathcal{C}_i$*

Table 3.3: *Auxiliary information for problem $\mathcal{J}3/n{=}2/\sum \mathcal{C}_i$*

G_k	$\Omega_k^u, u = 1, 2, \ldots, \omega_k$	$n_{11}(\Omega_k^u)$,	$n_{12}(\Omega_k^u)$,	$n_{13}(\Omega_k^u)$,	$n_{21}(\Omega_k^u)$,	$n_{22}(\Omega_k^u)$,	$n_{23}(\Omega_k^u)$	$\sum_{\nu \in \Omega_k^u} l^p(\nu)$
1	2	3						4
G_1	$\Omega_1^1 = \{O_{11}, O_{12}, O_{13}; O_{11}, O_{12}, O_{23}\}$	2	2	1	0	0	1	320
	$\Omega_1^2 = \{O_{11}, O_{12}, O_{13}; O_{21}, O_{12}, O_{23}\}$	1	2	1	1	0	1	305
	$\Omega_1^3 = \{O_{11}, O_{12}, O_{13}; O_{11}, O_{22}, O_{23}\}$	2	1	1	0	1	1	325
	$\Omega_1^4 = \{O_{11}, O_{12}, O_{13}; O_{21}, O_{22}, O_{23}\}$	1	1	1	1	1	1	310
	$\Omega_1^5 = \{O_{21}, O_{12}, O_{13}; O_{11}, O_{12}, O_{23}\}$	1	2	1	1	0	1	305
	$\Omega_1^6 = \{O_{21}, O_{12}, O_{13}; O_{21}, O_{12}, O_{23}\}$	0	2	1	2	0	1	290
	$\Omega_1^7 = \{O_{21}, O_{12}, O_{13}; O_{11}, O_{22}, O_{23}\}$	1	1	1	1	1	1	310
	$\Omega_1^8 = \{O_{21}, O_{12}, O_{13}; O_{21}, O_{22}, O_{23}\}$	0	1	1	2	1	1	295
G_2	$\Omega_2^1 = \{O_{11}, O_{22}, O_{23}, O_{12}, O_{13}; O_{11}, O_{22}, O_{23}\}$	2	1	1	0	2	2	410
	$\Omega_2^2 = \{O_{11}, O_{22}, O_{23}, O_{12}, O_{13}; O_{21}, O_{22}, O_{23}\}$	1	1	1	1	2	2	395
	$\Omega_2^3 = \{O_{21}, O_{22}, O_{23}, O_{12}, O_{13}; O_{11}, O_{22}, O_{23}\}$	1	1	1	1	2	2	395
	$\Omega_2^4 = \{O_{21}, O_{22}, O_{23}, O_{12}, O_{13}; O_{21}, O_{22}, O_{23}\}$	0	1	1	2	2	2	380
G_3	$\Omega_3^1 = \{O_{21}, O_{22}, O_{11}, O_{12}, O_{13}; O_{21}, O_{22}, O_{23}\}$	1	1	1	2	2	1	425
	$\Omega_3^2 = \{O_{21}, O_{22}, O_{23}, O_{12}, O_{13}; O_{21}, O_{22}, O_{23}\}$	0	1	1	2	2	2	380
G_4	$\Omega_4^1 = \{O_{11}, O_{12}, O_{13}; O_{11}, O_{12}, O_{21}, O_{22}, O_{23}\}$	2	2	1	1	1	1	435
G_5	$\Omega_5^1 = \{O_{21}, O_{22}, O_{11}, O_{12}, O_{13};$ $O_{21}, O_{22}, O_{11}, O_{12}, O_{23}\}$	2	2	1	2	2	1	550

The calculation of $\overline{\varrho}_1(p)$ by formula (1.15) is given in Table 3.4, which presents the results of the computations for each $\beta = 1, 2, \ldots, q - m$, where m is the number of

Table 3.4: *Calculation of the stability radius $\overline{\varrho}_1(p)$ for problem $\mathcal{J}3/n=2/\sum \mathcal{C}_i$*

G_k	$\lvert\Omega_{1k}\rvert$	$\Omega_1^v \in \Omega_{1k}$	Ω_k^u, $1 \le u \le \omega_k$	β	$p_{ij_{(m+\beta)}}$, $1 \le \beta \le q-m$	$\frac{\sum_{\alpha=1}^{m+\beta} p_{ij_{(\alpha)}}(n_{ij_{(\alpha)}}(\Omega_k^u)-n_{ij_{(\alpha)}}(\Omega_1^v))}{\sum_{\alpha=1}^{m+\beta} \lvert n_{ij_{(\alpha)}}(\Omega_k^u)-n_{ij_{(\alpha)}}(\Omega_1^v)\rvert}$	$\max_{\beta}$	$\max_{\Omega_k^u}$	$\min_{\Omega_1^v}$
1	2	3	4	5	6	7	8	9	10
G_2	4	Ω_1^1	Ω_2^1	1	$p_{ij_{(5)}}=55$	$\frac{50(1-2)+55(2-0)}{\lvert 1-2\rvert+\lvert 2-0\rvert}=20$	22.5	22.5	22.5
				2	$p_{ij_{(6)}}=30$	$\frac{50(1-2)+55(2-0)+30(2-1)}{\lvert 1-2\rvert+\lvert 2-0\rvert+\lvert 2-1\rvert}=22.5$			
			Ω_2^2,Ω_2^3	1	$p_{ij_{(4)}}=60$	$\frac{75(1-2)+50(1-2)+60(1-0)}{1+1+1}=\frac{-65}{3}$	12.5		
				2	$p_{ij_{(5)}}=55$	$\frac{-65+55(2-0)}{3+2}=\frac{45}{5}=9$			
				3	$p_{ij_{(6)}}=30$	$\frac{45+30(2-1)}{5+1}=12.5$			
			Ω_2^4	1	$p_{ij_{(4)}}=60$	$\frac{75(0-2)+50(1-2)+60(2-0)}{2+1+2}=\frac{-80}{5}$	7.5		
				2	$p_{ij_{(5)}}=60$	$\frac{-80+55(2-0)}{5+2}=\frac{30}{7}=4\frac{2}{3}$			
				3	$p_{ij_{(6)}}=30$	$\frac{30+30(2-1)}{7+1}=7.5$			
		Ω_1^2,Ω_1^5	Ω_2^1	1	$p_{ij_{(4)}}=75$	$\frac{50(1-2)+60(0-1)+75(2-1)}{1+1+1}=\frac{-35}{3}$	17.5	22.5	
				2	$p_{ij_{(5)}}=55$	$\frac{-35+55(2-0)}{3+2}=\frac{75}{5}=15$			
				3	$p_{ij_{(6)}}=30$	$\frac{75+30(2-1)}{5+1}=17.5$			
			Ω_2^2,Ω_2^3	1	$p_{ij_{(5)}}=55$	$\frac{50(1-2)+55(2-0)}{1+2}=\frac{60}{3}=20$	22.5		
				2	$p_{ij_{(6)}}=30$	$\frac{60+30(2-1)}{3+1}=22.5$			
			Ω_2^4	1	$p_{ij_{(4)}}=60$	$\frac{75(0-1)+50(1-2)+60(2-1)}{1+1+1}=\frac{-65}{3}$	12.5		
				2	$p_{ij_{(5)}}=55$	$\frac{-65+55(2-0)}{3+2}=\frac{45}{5}=9$			
				3	$p_{ij_{(6)}}=30$	$\frac{45+30(2-1)}{5+1}=12.5$			
		Ω_1^6	Ω_2^1	1	$p_{ij_{(4)}}=75$	$\frac{50(1-2)+60(0-2)+75(2-0)}{1+2+2}=\frac{-20}{5}$	15	22.5	
				2	$p_{ij_{(5)}}=55$	$\frac{-20+55(2-0)}{5+2}=\frac{90}{7}=12\frac{6}{7}$			
				3	$p_{ij_{(6)}}=30$	$\frac{90+30(2-1)}{7+1}=15$			
			Ω_2^2,Ω_2^3	1	$p_{ij_{(4)}}=75$	$\frac{50(1-2)+60(1-2)+75(1-0)}{1+1+1}=\frac{-35}{3}$	17.5		
				2	$p_{ij_{(5)}}=55$	$\frac{-35+55(2-0)}{3+2}=\frac{75}{5}=15$			
				3	$p_{ij_{(6)}}=30$	$\frac{75+30(2-1)}{5+1}=17.5$			
			Ω_2^4	1	$p_{ij_{(5)}}=55$	$\frac{50(1-2)+55(2-0)}{1+2}=\frac{60}{3}=20$	22.5		
				2	$p_{ij_{(6)}}=30$	$\frac{60+30(2-1)}{3+1}=22.5$			
G_3	5	Ω_1^1	Ω_3^1	1	$p_{ij_{(5)}}=60$	$\frac{75(1-2)+50(1-2)+60(2-0)}{1+1+2}=\frac{-5}{4}$	17.5	17.5	17.5
				2	$p_{ij_{(6)}}=55$	$\frac{-5+55(2-0)}{4+2}=17.5$			
			Ω_3^2	1	$p_{ij_{(4)}}=60$	$\frac{75(0-2)+50(1-2)+60(2-0)}{2+1+2}=\frac{-80}{5}$	7.5		
				2	$p_{ij_{(5)}}=55$	$\frac{-80+55(2-0)}{5+2}=\frac{30}{7}=4\frac{2}{7}$			
				3	$p_{ij_{(6)}}=30$	$\frac{30+30(2-1)}{7+1}=7.5$			
		Ω_1^2,Ω_1^5	Ω_3^1	1	$p_{ij_{(5)}}=60$	$\frac{50(1-2)+60(2-1)}{1+1}=\frac{10}{2}=5$	30	30	
				2	$p_{ij_{(6)}}=55$	$\frac{10+55(2-0)}{2+2}=30$			
			Ω_3^2	1	$p_{ij_{(4)}}=60$	$\frac{75(0-1)+50(1-2)+60(1+1-1)}{1+1+1}=\frac{-65}{3}$	12.5		
				2	$p_{ij_{(5)}}=55$	$\frac{-65+55(2-0)}{3+2}=\frac{45}{5}=9$			
				3	$p_{ij_{(6)}}=30$	$\frac{45+30(2-1)}{5+1}=12.5$			

Table 3.4 (continuation): *Calculation of the stability radius $\overline{\varrho}_1(p)$ for problem $\mathcal{J}3/n{=}2/\sum\mathcal{C}_i$*

G_k	$\lvert\Omega_{1k}\rvert$	$\Omega_1^v\in\Omega_{1k}$	Ω_k^u, $1\le u\le\omega_k$	β	$p_{ij_{(m+\beta)}}$, $1\le\beta\le q-m$	$\frac{\sum_{\alpha=1}^{m+\beta} p_{ij_{(\alpha)}}(n_{ij_{(\alpha)}}(\Omega_k^u)-n_{ij_{(\alpha)}}(\Omega_1^v))}{\sum_{\alpha=1}^{m+\beta}\lvert n_{ij_{(\alpha)}}(\Omega_k^u)-n_{ij_{(\alpha)}}(\Omega_1^v)\rvert}$	$\max\limits_{\beta}$	$\max\limits_{\Omega_k^u}$	$\min\limits_{\Omega_1^v}$
1	2	3	4	5	6	7	8	9	10
		Ω_1^3	Ω_3^1	1	$p_{ij_{(5)}}=60$	$\frac{75(1-2)+60(2-0)}{1+2}=\frac{45}{3}=15$	25	25	
				2	$p_{ij_{(6)}}=55$	$\frac{45+55(2-1)}{3+1}=25$			
			Ω_3^2	1	$p_{ij_{(4)}}=60$	$\frac{75(0-2)+60(2-0)}{2+2}=\frac{-30}{4}$	$9\frac{1}{6}$		
				2	$p_{ij_{(5)}}=55$	$\frac{-30+55(2-1)}{4+1}=\frac{25}{5}=5$			
				3	$p_{ij_{(6)}}=30$	$\frac{25+30(2-1)}{5+1}=9\frac{1}{6}$			
		Ω_1^6	Ω_3^1	1	$p_{ij_{(5)}}=75$	$\frac{50(1-2)+75(1-0)}{1+1}=\frac{25}{2}=12.5$	33.75	33.75	
				2	$p_{ij_{(6)}}=55$	$\frac{25+55(2-0)}{2+2}=33.75$			
			Ω_3^2	1	$p_{ij_{(5)}}=55$	$\frac{50(1-2)+55(2-0)}{1+2}=\frac{60}{3}=20$	22.5		
				2	$p_{ij_{(6)}}=30$	$\frac{60+30(2-1)}{3+1}=22.5$			
G_4	2	Ω_1^6	Ω_4^1	1	$p_{ij_{(5)}}=75$	$\frac{60(1-2)+75(2-0)}{1+2}=\frac{90}{3}=30$	36.25	36.25	35
				2	$p_{ij_{(6)}}=55$	$\frac{90+55(1-0)}{3+1}=36.25$			
		Ω_1^8	Ω_4^1	1	$p_{ij_{(5)}}=75$	$\frac{60(1-2)+75(2-0)}{1+2}=\frac{90}{3}=30$	35	35	
				2	$p_{ij_{(6)}}=55$	$\frac{90+55(2-1)}{3+1}=35$			
G_5	0								

operations $O_{ij}\in\Omega_1^v\cup\Omega_k^u$, $\Omega_1^v\in\Omega_{1k}$, for which $n_{ij}(\Omega_1^v)<n_{ij}(\Omega_k^u)$. The cardinality of the set $\Omega_{1k}, k=1,2,\dots,5$, and the elements Ω_1^ν of this set are presented in column 2 and column 3, respectively. The elements of the set Ω_k^u, $u=1,2,\dots,\omega_k$, for which $\sum_{\nu\in\Omega_k^u} l^p(\nu)\ge L_1^p=325$ are presented in column 4.

Since the vector $n(\Omega_k^u)=(n_{11}(\Omega_k^u),n_{12}(\Omega_k^u),\dots,n_{23}(\Omega_k^u))$ is the same for both sets Ω_1^2 and Ω_1^5, for both sets Ω_1^4 and Ω_1^7, and for both sets Ω_2^2 and Ω_2^3 (see Table 3.3), the results calculated by formula (1.15) are the same for these pairs of sets, too. Therefore, we combine these calculations in column 7 in Table 3.4. In column 6 we give the sequence of processing times of the operations $O_{ij}\in\Omega_1^v\cup\Omega_k^u$ with $n_{ij}(\Omega_1^v)<n_{ij}(\Omega_k^u)$ ordered in the following way: $p_{ij_{(m+1)}}\ge p_{ij_{(m+2)}}\ge\dots\ge p_{ij_{(q)}}$. Note that in column 7 we do not write components with $n_{ij}(\Omega_1^v)=n_{ij}(\Omega_k^u)$ in the fraction from formula (1.15). For the sets Ω_1^1 and Ω_2^1, we give a more detailed computation and for each other pair of the sets Ω_1^v and Ω_k^u at each following iteration, we use the value of the fraction obtained at the previous iteration. From the derived values in column 7, we write their maximum for $\beta=1,2,\dots,q-m$, the maximum for Ω_k^u, $u=1,2,\dots,\omega_k$, and the minimum for $\Omega_1^v\in\Omega_{1k}$, respectively, in columns 8, 9 and 10. Using formula (1.14), we take the minimum value from column 10. Therefore, we obtain $\overline{\varrho}_1(p)=17.5$.

Let us consider an uncertain job shop problem $\mathcal{J}3/n{=}2,a_i\le p_i\le b_i/\sum\mathcal{C}_i$ to illustrate the idea of constructing a solution set mentioned in Remark 3.4. The structural input data are given by the mixed graph G in Figure 2.1 and the numerical input data are given in Table 2.3. Obviously, the set of all feasible digraphs $\Lambda(G)$ is identical for $\mathcal{C}_{max}$ and

$\sum \mathcal{C}_i$, and here we number these digraphs in non-decreasing order of the values $\sum \mathcal{C}_i$ with the same initial vector $p = (75, 50, 40, 60, 55, 30)$ as for problem $\mathcal{J}3/n{=}2, a_i \le p_i \le b_i/\mathcal{C}_{max}$ considered in Chapter 2: $L_1^p \le L_2^p \le \ldots \le L_5^p$ (see Figure 3.3). Using the modification of CPM described at page 57, we can simplify the digraphs $G_1, G_2, \ldots, G_5$, but for these numerical input data (see Table 2.3) the corresponding digraphs $G_1^T, G_2^T, \ldots, G_5^T$ are the same. It means that the number of sets of representatives ω_k^T is equal to the number ω_k for each digraph $G_k, k = 1, 2, \ldots, 5$, (see Table 3.3). Let us find the relative stability radius $\overline{\varrho}_1^{\Lambda(G)}(p \in T)$ of the optimal digraph $G_1(p)$ presented in Figure 3.3.

Table 3.5: *Auxiliary information for the construction of the sets $\Omega_{1k}^*, k \in \{2, 3, 4, 5\}$, for problem $\mathcal{J}3/n{=}2, a_i \le p_i \le b_i/\sum \mathcal{C}_i$*

G_k, $2 \le k \le 5$	Ω_1^v, $v \le 1 \le \omega_1^T$	Ω_k^u, $u \le 1 \le \omega_k^T$	$p^* \in T$	$\sum_{\mu \in \Omega_1^v} l^{p^*}(\mu) > \sum_{\mu \in \Omega_1^v} l^{p^*}(\mu)$	Ω_{1k}^*
1	2	3	4	5	6
G_2	Ω_1^1	Ω_2^1	$(75, 90, 40, 60, 45, 20)$	$390 \ngtr 410$	$\Omega_1^1 \not\in \Omega_{12}^*$
		Ω_2^2, Ω_2^3	$(100, 90, 40, 50, 45, 20)$	$440 > 410$	
		Ω_2^4	$(100, 90, 40, 50, 45, 20)$	$440 > 410$	
	Ω_1^2, Ω_1^5	Ω_2^1	$(35, 90, 40, 80, 45, 20)$	$400 > 330$	$\{\Omega_1^2, \Omega_1^5\} \not\subseteq \Omega_{12}^*$
		Ω_2^2, Ω_2^3	$(75, 90, 40, 60, 45, 20)$	$375 \ngtr 395$	
		Ω_2^4	$(100, 90, 40, 50, 45, 20)$	$390 > 360$	
	Ω_1^3	Ω_2^1	$(75, 50, 40, 60, 45, 20)$	$305 \ngtr 370$	$\Omega_1^3 \not\in \Omega_{12}^*$
		Ω_2^2, Ω_2^3	$(100, 50, 40, 50, 45, 20)$	$355 \ngtr 370$	
		Ω_2^4	$(100, 50, 40, 50, 45, 20)$	$355 > 320$	
	$\{\Omega_1^4, \Omega_1^7\} \not\subseteq \Omega_{12}$				$\{\Omega_1^4, \Omega_1^7\} \not\subseteq \Omega_{12}^*$
	Ω_1^6	Ω_2^1	$(35, 90, 40, 80, 45, 20)$	$400 > 330$	$\Omega_1^6 \not\in \Omega_{12}^*$
		Ω_2^2, Ω_2^3	$(35, 90, 40, 80, 45, 20)$	$400 > 375$	
		Ω_2^4	$(75, 90, 40, 60, 45, 20)$	$360 \ngtr 380$	
	$\Omega_1^8 \not\in \Omega_{12}$				$\Omega_1^8 \not\in \Omega_{12}^*$
G_3	Ω_1^1	Ω_3^1	$(100, 90, 40, 50, 45, 30)$	$450 \ngtr 450$	$\Omega_1^1 \not\in \Omega_{13}^*$
		Ω_3^2	$(100, 90, 40, 50, 45, 20)$	$440 > 360$	
	Ω_1^2, Ω_1^5	Ω_3^1	$(75, 90, 40, 50, 45, 30)$	$375 \ngtr 425$	$\{\Omega_1^2, \Omega_1^5\} \not\subseteq \Omega_{13}^*$
		Ω_3^2	$(100, 90, 40, 50, 45, 20)$	$390 > 360$	
	Ω_1^3	Ω_3^1	$(100, 50, 40, 50, 45, 30)$	$365 \ngtr 410$	$\Omega_1^3 \not\in \Omega_{13}^*$
		Ω_3^2	$(100, 50, 40, 50, 45, 20)$	$355 > 320$	
	$\{\Omega_1^4, \Omega_1^7\} \not\subseteq \Omega_{13}$				$\{\Omega_1^4, \Omega_1^7\} \not\subseteq \Omega_{13}^*$
	Ω_1^6	Ω_3^1	$(35, 90, 40, 60, 45, 30)$	$370 \ngtr 405$	$\Omega_1^6 \not\in \Omega_{13}^*$
		Ω_3^2	$(75, 90, 40, 60, 45, 20)$	$360 \ngtr 380$	
	$\Omega_1^8 \not\in \Omega_{13}$				$\Omega_1^8 \not\in \Omega_{13}^*$
G_4	$\Omega_1^1 \not\in \Omega_{14}$				$\Omega_1^1 \not\in \Omega_{14}^*$
	$\{\Omega_1^2, \Omega_1^5\} \not\subseteq \Omega_{14}$				$\{\Omega_1^2, \Omega_1^5\} \not\subseteq \Omega_{14}^*$
	$\Omega_1^3 \not\in \Omega_{14}$				$\Omega_1^3 \not\in \Omega_{14}^*$
	$\{\Omega_1^4, \Omega_1^7\} \not\subseteq \Omega_{14}$				$\{\Omega_1^4, \Omega_1^7\} \not\subseteq \Omega_{14}^*$
	Ω_1^6	Ω_4^1	$(35, 50, 40, 80, 45, 30)$	$330 \ngtr 365$	$\Omega_1^6 \not\in \Omega_{14}^*$
	Ω_1^8	Ω_4^1	$(35, 40, 40, 80, 55, 30)$	$325 \ngtr 355$	$\Omega_1^8 \not\in \Omega_{14}^*$
G_5	$\{\Omega_{15}\} = \emptyset$				$\{\Omega_{15}^*\} = \emptyset$

First, due to Remark 3.2 we have to construct the set Ω_{1k}^* for each digraph $G_k, k = 2, 3, 4, 5$. To this end, we construct an auxiliary Table 3.5, where for each combination of the sets $\Omega_1^v, v = 1, 2, \ldots, \omega_1^T$, and $\Omega_k^u, u = 1, 2, \ldots, \omega_k^T, k = 2, 3, 4, 5$, we obtain the vector

p^* according to formula (3.15) (see column 4) and check inequality (3.16) (see column 5). As we did in Table 3.4, we combine the same calculations for each pair of sets Ω_1^2 and Ω_1^5, Ω_1^4 and Ω_1^7, Ω_2^2 and Ω_2^3. Since $\Omega_{sk}^* \subseteq \Omega_{sk}$, we do not perform such a calculation for the sets Ω_k^u, which do not belong to the sets $\Omega_{1k}, k = 2,3,4,5$, (see Table 3.4). So, it follows from column 5 that there is no set of representatives Ω_1^v, $v \in \{1,2,\dots,\omega_1^T\}$, such that the inequality (3.16) holds for each set of representatives Ω_k^u, $u \in \{1,2,\dots,\omega_k^T\}$. Thus, $\Omega_{1k}^* = \emptyset$ for each digraph $G_k \in B = \Lambda(G) \setminus \{G_1\}$. Therefore, from Theorem 3.3 and Theorem 3.4, it follows that $\overline{\varrho}_1^{\Lambda(G)}(p \in T) = \infty$. (For the numerical input data presented in Table 2.3, digraph G_1 dominates all digraphs $G_k \in \Lambda(G)$ in the polytope T and remains the best for all feasible vectors $x \in T$ of the processing times.) In such a case, we obtain a single-element minimal solution $\Lambda^T(G) = \{G_1\}$.

To illustrate the case of formula (3.18) from Theorem 3.3, we give the following example.

Example 3.2 Let us consider a similar job shop problem $\mathcal{J}3/n=2, a_i \le p_i \le b_i/\sum C_i$ with the same structural input data (see Figure 2.1), but with different numerical input data (see Table 3.6). We do not simplify the digraphs $G_1, G_2, \dots, G_5$ for the new numerical input data, i.e. the corresponding digraphs $G_1^T, G_2^T, \dots, G_5^T$ have the same sets of representatives $\{\Omega_k^u : u = 1,2,\dots,\omega_k^T,\ \omega_k^T = \omega_k,\ k = 1,2,\dots,5\}$. For the same initial vector $p = (75,50,40,60,55,30)$, we have the same optimal digraph $G_1(p)$ and all feasible digraphs $\Lambda(G)$ are numbered as for the above problem (see Figure 3.3). Let us calculate the relative stability radius $\overline{\varrho}_1^{\Lambda(G)}(p \in T)$ on the basis of Theorem 3.3.

Table 3.6: *Numerical data for problem* $\mathcal{J}3/n=2, a_i \le p_i \le b_i/\sum C_i$

i	1	1	1	2	2	2
j	1	2	3	1	2	3
a_{ij}	35	30	40	45	10	15
b_{ij}	105	100	75	85	65	50

For the combination of the numerical input data given in Table 3.6 we construct the following sets:

$$\begin{aligned}
\Omega_{12}^* &= \{\Omega_1^1, \Omega_1^2, \Omega_1^5, \Omega_1^6\},\\
\Omega_{13}^* &= \{\Omega_1^1, \Omega_1^2, \Omega_1^5, \Omega_1^6\},\\
\Omega_{14}^* &= \{\Omega_1^6\},\\
\Omega_{15}^* &= \emptyset.
\end{aligned}$$

It means that digraph G_1 does not dominate digraphs G_2, G_3, G_4 in the polytope T and due to Theorem 3.4, we have $\overline{\varrho}_1^{\Lambda(G)}(p \in T) \ne \infty$.

In Table 3.7 one can observe the calculation of the relative stability radius $\overline{\varrho}_1^{\Lambda(G)}(p \in T)$ for the vector $p = (75,50,40,60,55,30)$. Following Theorem 3.3, we must compare digraph G_1 with each digraph G_k, $k = 2,3,4$, for which $\Omega_{1k}^* \ne \emptyset$. Thus, we perform the calculations due to formulas (3.11) and (3.18) for each set $\Omega_1^v \in \Omega_{1k}^*$ (see column 2) and each set Ω_k^u (see column 3). For the sets Ω_k^u in column 3, inequality $\sum_{\nu \in \Omega_k^u} l^p(\nu) \ge L_1^p = 325$ holds (see Table 3.3). Column 5 contains the values $\Delta_\beta^{ij}(\Omega_1^v, \Omega_k^u), \beta = 0,1,\dots,|N(\Omega_1^v, \Omega_k^u)|-1$,

Table 3.7: *Calculation of the relative stability radius* $\overline{\varrho}_1^{\Lambda(G)}(p \in T)$ *for problem* $\mathcal{J}3/n = 2, a_i \le p_i \le b_i / \sum C_i$

G_k	$\Omega_1^v \in \Omega_{1k}^*$	Ω_k^u, $1 \le u \le \omega_k^T$	β	$\Delta_\beta^{ij}(\Omega_1^v, \Omega_k^u)$, $0 \le \beta \le \lvert N(\Omega_1^v, \Omega_k^u)\rvert - 1$	$N_\beta(\Delta)$	$\dfrac{\sum_{\nu\in\Omega_k^u} l^p(\nu) - \sum_{\mu\in\Omega_1^v} l^p(\mu) - \sum_{\alpha=0}^{\beta}\Delta_\alpha^{ij}(\Omega_1^v,\Omega_k^u)\cdot N_\alpha(\Delta)}{\sum_{O_{ij}\in Q}\lvert n_{ij}(\Omega_k^u) - n_{ij}(\Omega_1^v)\rvert - \sum_{\alpha=0}^{\beta}\lvert N_\alpha(\Delta)\rvert}$	$\max\limits_{\beta}$	$\max\limits_{\Omega_k^u}$	$\min\limits_{\Omega_1^v}$
1	2	3	4	5	6	7	8	9	10
G_2	Ω_1^1	Ω_2^1	0	$\Delta_0^{ij}(\Omega_1^1,\Omega_2^1)=0$	0	$\frac{410-320-0}{4-0}=\frac{90}{4}=22.5$	25	25	25
			1	$\Delta_1^{23}(\Omega_1^1,\Omega_2^1)=15$	1	$\frac{90-15}{4-1}=\frac{75}{3}=25$			
			2	$\Delta_2^{22}(\Omega_1^1,\Omega_2^1)=45$	2	$\frac{75-45\cdot 2}{3-2}<0$			
		Ω_2^2,Ω_2^3	0	$\Delta_0^{ij}(\Omega_1^1,\Omega_2^2)=0$	0	$\frac{395-320-0}{6-0}=\frac{75}{6}=12.5$	12.5		
			1	$\Delta_1^{21}(\Omega_1^1,\Omega_2^2)=15$	1	$\frac{75-15}{6-1}=\frac{60}{5}=12$			
			2	$\Delta_2^{23}(\Omega_1^1,\Omega_2^2)=15$	1	$\frac{60-15}{5-1}=\frac{45}{4}=11.25$			
			3	$\Delta_3^{11}(\Omega_1^1,\Omega_2^2)=30$	1	$\frac{45-30}{4-1}=\frac{15}{3}=5$			
			4	$\Delta_4^{22}(\Omega_1^1,\Omega_2^2)=45$	2	$\frac{15-45\cdot 2}{3-2}<0$			
		Ω_2^4	0	$\Delta_0^{ij}(\Omega_1^1,\Omega_2^4)=0$	0	$\frac{380-320-0}{8-0}=\frac{60}{8}=7.5$	7.5		
			1	$\Delta_1^{23}(\Omega_1^1,\Omega_2^4)=15$	1	$\frac{60-15}{8-1}=\frac{45}{7}=6\frac{3}{7}$			
			2	$\Delta_2^{21}(\Omega_1^1,\Omega_2^4)=15$	2	$\frac{45-15\cdot 2}{7-2}=\frac{15}{5}=3$			
			3	$\Delta_3^{11}(\Omega_1^1,\Omega_2^4)=30$	2	$\frac{15-30\cdot 2}{5-2}<0$			
	Ω_1^2,Ω_1^5	Ω_2^1	0	$\Delta_0^{ij}(\Omega_1^2,\Omega_2^1)=0$	0	$\frac{410-305-0}{6-0}=\frac{105}{6}=17.5$	18	25	
			1	$\Delta_1^{23}(\Omega_1^2,\Omega_2^1)=15$	1	$\frac{105-15}{6-1}=\frac{90}{5}=18$			
			2	$\Delta_2^{21}(\Omega_1^2,\Omega_2^1)=25$	1	$\frac{90-25}{5-1}=\frac{65}{4}=16.25$			
			3	$\Delta_3^{11}(\Omega_1^2,\Omega_2^1)=40$	1	$\frac{65-40}{4-1}=\frac{25}{3}=8\frac{1}{3}$			
			4	$\Delta_4^{22}(\Omega_1^2,\Omega_2^1)=45$	2	$\frac{25-44\cdot 2}{3-2}<0$			
		Ω_2^2,Ω_2^3	0	$\Delta_0^{ij}(\Omega_1^2,\Omega_2^2)=0$	0	$\frac{395-305-0}{4-0}=\frac{90}{4}=22.5$	25		
			1	$\Delta_1^{23}(\Omega_1^2,\Omega_2^2)=15$	1	$\frac{90-15}{4-1}=\frac{75}{3}=25$			
			2	$\Delta_2^{22}(\Omega_1^2,\Omega_2^2)=45$	2	$\frac{75-45\cdot 2}{3-2}<0$			
		Ω_2^4	0	$\Delta_0^{ij}(\Omega_1^2,\Omega_2^4)=0$	0	$\frac{380-305-0}{6-0}=\frac{75}{6}=12.5$	12.5		
			1	$\Delta_1^{21}(\Omega_1^2,\Omega_2^4)=15$	1	$\frac{75-15}{6-1}=\frac{60}{5}=12$			
			2	$\Delta_2^{23}(\Omega_1^2,\Omega_2^4)=15$	1	$\frac{60-15}{5-1}=\frac{45}{4}=11.25$			
			3	$\Delta_3^{11}(\Omega_1^2,\Omega_2^4)=30$	1	$\frac{45-30}{4-1}=\frac{15}{3}=5$			
			4	$\Delta_4^{22}(\Omega_1^2,\Omega_2^4)=45$	2	$\frac{15-45\cdot 2}{3-2}<0$			
	Ω_1^6	Ω_2^1	0	$\Delta_0^{ij}(\Omega_1^6,\Omega_2^1)=0$	0	$\frac{410-290-0}{8-0}=\frac{120}{8}=15$	15	25	
			1	$\Delta_1^{23}(\Omega_1^6,\Omega_2^1)=15$	1	$\frac{120-15}{8-1}=\frac{105}{7}=15$			
			2	$\Delta_2^{21}(\Omega_1^6,\Omega_2^1)=25$	2	$\frac{105-25\cdot 2}{7-2}=\frac{55}{5}=11$			
			3	$\Delta_3^{11}(\Omega_1^6,\Omega_2^1)=40$	2	$\frac{55-40\cdot 2}{5-2}<0$			
		Ω_2^2,Ω_2^3	0	$\Delta_0^{ij}(\Omega_1^6,\Omega_2^2)=0$	0	$\frac{395-290-0}{6-0}=\frac{105}{6}=17.5$	18		
			1	$\Delta_1^{23}(\Omega_1^6,\Omega_2^2)=15$	1	$\frac{105-15}{6-1}=\frac{90}{5}=18$			
			2	$\Delta_2^{21}(\Omega_1^6,\Omega_2^2)=25$	1	$\frac{90-25}{5-1}=\frac{65}{4}=16.25$			
			3	$\Delta_3^{11}(\Omega_1^6,\Omega_2^2)=40$	1	$\frac{65-40}{4-1}=\frac{25}{3}=8\frac{1}{3}$			
			4	$\Delta_4^{22}(\Omega_1^6,\Omega_2^2)=45$	2	$\frac{25-45\cdot 2}{3-2}<0$			
		Ω_2^4	0	$\Delta_0^{ij}(\Omega_1^6,\Omega_2^4)=0$	0	$\frac{380-290-0}{4-0}=\frac{90}{4}=22.5$	25		
			1	$\Delta_1^{23}(\Omega_1^6,\Omega_2^4)=15$	1	$\frac{90-15}{4-1}=\frac{75}{3}=25$			
			2	$\Delta_2^{22}(\Omega_1^6,\Omega_2^4)=45$	2	$\frac{75-45\cdot 2}{3-2}<0$			

Table 3.7 (continuation): *Calculation of the relative stability radius* $\overline{\varrho}_1^{\Lambda(G)}(p \in T)$ *for problem* $\mathcal{J}3/n\!=\!2, a_i \le p_i \le b_i / \sum \mathcal{C}_i$

G_k	$\Omega_1^v \in \Omega_{1k}^*$	Ω_k^u, $1 \le u \le \omega_k^T$	β	$\Delta_\beta^{ij}(\Omega_1^v, \Omega_k^u)$, $0 \le \beta \le \lvert N(\Omega_1^v, \Omega_k^u)\rvert - 1$	$N_\beta(\Delta)$	$\dfrac{\sum_{\nu \in \Omega_k^u} l^p(\nu) - \sum_{\mu \in \Omega_1^v} l^p(\mu) - \sum_{\alpha=0}^{\beta} \Delta_\alpha^{ij}(\Omega_1^v, \Omega_k^u) \cdot N_\alpha(\Delta)}{\sum_{O_{ij} \in Q} \lvert n_{ij}(\Omega_k^u) - n_{ij}(\Omega_1^v)\rvert - \sum_{\alpha=0}^{\beta} \lvert N_\alpha(\Delta)\rvert}$	$\max_\beta$	$\max_{\Omega_k^u}$	$\min_{\Omega_1^v}$
1	2	3	4	5	6	7	8	9	10
G_3	Ω_1^1	Ω_3^1	0	$\Delta_0^{ij}(\Omega_1^1, \Omega_3^1) = 0$	0	$\frac{425-320-0}{6-0} = \frac{105}{6} = 17.5$	18.75	18.75	18.75
			1	$\Delta_1^{21}(\Omega_1^1, \Omega_3^1) = 15$	2	$\frac{105-15\cdot 2}{6-2} = \frac{75}{4} = 18.75$			
			2	$\Delta_2^{11}(\Omega_1^1, \Omega_3^1) = 30$	1	$\frac{75-30}{4-1} = \frac{45}{3} = 15$			
			3	$\Delta_3^{22}(\Omega_1^1, \Omega_3^1) = 45$	2	$\frac{45-45\cdot 2}{3-2} < 0$			
		Ω_3^2	0	$\Delta_0^{ij}(\Omega_1^1, \Omega_3^2) = 0$	0	$\frac{380-290-0}{8-0} = \frac{60}{8} = 7.5$	7.5		
			1	$\Delta_1^{21}(\Omega_1^1, \Omega_3^2) = 15$	1	$\frac{60-15}{8-1} = \frac{45}{7} = 6\frac{3}{7}$			
			2	$\Delta_2^{23}(\Omega_1^1, \Omega_3^2) = 15$	2	$\frac{45-15\cdot 2}{7-2} = \frac{15}{5} = 3$			
			3	$\Delta_3^{11}(\Omega_1^1, \Omega_3^2) = 30$	2	$\frac{15-30\cdot 2}{5-2} < 0$			
	Ω_1^2, Ω_1^5	Ω_3^1	0	$\Delta_0^{ij}(\Omega_1^2, \Omega_3^1) = 0$	0	$\frac{425-305-0}{4-0} = \frac{120}{4} = 30$	35	35	
			1	$\Delta_1^{21}(\Omega_1^2, \Omega_3^1) = 15$	1	$\frac{120-15}{4-1} = \frac{105}{3} = 35$			
			2	$\Delta_2^{22}(\Omega_1^2, \Omega_3^1) = 45$	2	$\frac{105-45\cdot 2}{3-2} = 15$			
		Ω_3^2	0	$\Delta_0^{ij}(\Omega_1^2, \Omega_3^2) = 0$	0	$\frac{380-305-0}{6-0} = \frac{75}{6} = 12.5$	12.5		
			1	$\Delta_1^{21}(\Omega_1^2, \Omega_3^2) = 15$	1	$\frac{75-15}{6-1} = \frac{60}{5} = 12$			
			2	$\Delta_2^{22}(\Omega_1^2, \Omega_3^2) = 45$	1	$\frac{60-15}{5-1} = \frac{45}{4} = 11.25$			
			3	$\Delta_3^{11}(\Omega_1^2, \Omega_3^2) = 30$	1	$\frac{45-30}{4-1} = \frac{15}{3} = 5$			
			4	$\Delta_4^{22}(\Omega_1^2, \Omega_3^2) = 45$	2	$\frac{15-45\cdot 2}{3-2} < 0$			
	Ω_1^6	Ω_3^1	0	$\Delta_0^{ij}(\Omega_1^6, \Omega_3^1) = 0$	0	$\frac{425-290-0}{4-0} = \frac{135}{4} = 33.75$	33.75	33.75	
			1	$\Delta_1^{11}(\Omega_1^6, \Omega_3^1) = 40$	1	$\frac{135-40}{4-1} = \frac{95}{3} = 31\frac{2}{3}$			
			2	$\Delta_2^{22}(\Omega_1^6, \Omega_3^1) = 45$	2	$\frac{95-45\cdot 2}{3-2} = 5$			
		Ω_3^2	0	$\Delta_0^{ij}(\Omega_1^6, \Omega_3^2) = 0$	0	$\frac{380-290-0}{4-0} = \frac{90}{4} = 22.5$	25		
			1	$\Delta_1^{23}(\Omega_1^6, \Omega_3^2) = 15$	1	$\frac{90-15}{4-1} = \frac{75}{3} = 25$			
			2	$\Delta_2^{22}(\Omega_1^6, \Omega_3^2) = 45$	2	$\frac{75-45\cdot 2}{3-2} < 0$			
G_4	Ω_1^6	Ω_4^1	0	$\Delta_0^{ij}(\Omega_1^6, \Omega_4^1) = 0$	0	$\frac{435-290-0}{4-0} = \frac{145}{4} = 36.25$	40	40	40
			1	$\Delta_1^{21}(\Omega_1^6, \Omega_4^1) = 25$	1	$\frac{145-25}{4-1} = \frac{120}{3} = 40$			
			2	$\Delta_2^{11}(\Omega_1^6, \Omega_4^1) = 40$	2	$\frac{120-40\cdot 2}{3-2} = 40$			

defined by formula (3.9) (see page 64) for each operation $O_{ij} \in N(\Omega_k^u, \Omega_1^v) = \{\cup_{\mu \in \Omega_k^u \cup \Omega_1^v}[\mu] : n_{ij}(\Omega_k^u) \neq n_{ij}(\Omega_1^v)\}$. The order of these values is defined by (3.10). The corresponding values $N_\beta(\Delta)$ are given in column 6. Column 8 contains the value $r_{\Omega_1^v, \Omega_k^u}$ which is equal to the maximum of the values given in column 7 for $\beta = 0, 1, \ldots, |N(\Omega_1^v, \Omega_k^u)| - 1$ (see formula (3.11)). The values

$$\overline{r}_{k1}^B = \min_{\Omega_1^v \in \Omega_{1k}^*} \max_{\substack{u \in \{1,2,\ldots,\omega_k^T\}, \\ \sum_{\nu \in \Omega_k^u} l^p(\nu) \ge L_1^p}} r_{\Omega_1^v, \Omega_k^u}$$

calculated according to (3.17) are given in column 10. As follows from Theorem 3.3, the last step is to take the minimum value in column 10: $\overline{\varrho}_1^{\Lambda(G)}(p \in T) = \min\{\overline{r}_{k1}^B,\ k = 2, 3, 4\} = \overline{r}_{31} = 18.75$.

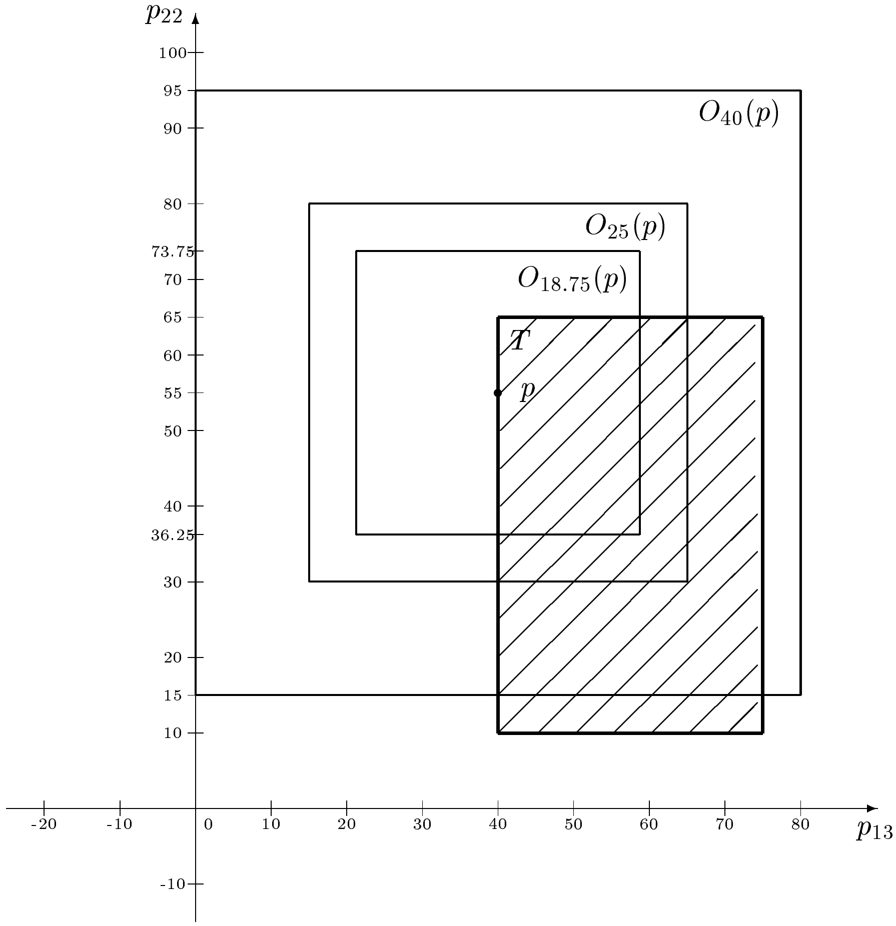


Figure 3.4: *Projections of the stability balls with the center* $p = (75, 50, 40, 60, 55, 30)$ *on the plane for problem* $\mathcal{J}3/n\!=\!2, a_i \le p_i \le b_i / \sum \mathcal{C}_i$

Table 3.8: *Optimal digraphs for problem* $\mathcal{J}3/n\!=\!2, a_i \le p_i \le b_i / \sum \mathcal{C}_i$ *with different initial vectors* $p \in T$

Initial vector $p^j \in T$	Objective function values $\sum \mathcal{C}_i$	Optimal digraph G_s	$\overline{\varrho}_s^{\Lambda(G)}(p^j \in T)$	Competitive digraph of G_s
1	2	3	4	5
$p^1 = (75, 95, 40, 60, 10, 30)$	$L_2^{p^1} = 365, L_3^{p^1} = 380, L_1^{p^1} = 410,$ $L_4^{p^1} = 480, L_5^{p^1} = 550$	G_2	$\overline{r}_{32}^{\Lambda(G)} = 3.75$	G_3
$p^2 = (80, 95, 40, 55, 10, 35)$	$L_3^{p^2} = 380, L_2^{p^2} = 385, L_1^{p^2} = 425,$ $L_4^{p^2} = 490, L_5^{p^2} = 555$	G_3	$\overline{r}_{23}^{\Lambda(G)} = 1.25$	G_2
$p^3 = (35, 35, 50, 85, 10, 30)$	$L_4^{p^3} = 315, L_2^{p^3} = 320, L_1^{p^3} = 335,$ $L_3^{p^3} = 340, L_5^{p^3} = 410$	G_4	$\overline{r}_{14}^{\Lambda(G)} = 1.25$	G_1

As follows from Remark 3.4, we also construct an increasing sequence of relative stability radii $\overline{\varrho}_1^{\Lambda(G)}(p \in T) = \overline{r}_{31} = 18.75$, $\overline{\varrho}_1^{\Lambda(G)\setminus\{G_3\}}(p \in T) = \overline{r}_{21} = 25$, $\overline{\varrho}_1^{\Lambda(G)\setminus\{G_3,G_2\}}(p \in T) = \overline{r}_{41} = 40$, $\overline{\varrho}_1^{\Lambda(G)\setminus\{G_3,G_2,G_4\}}(p \in T) = \infty$ (see Figure 3.4) and a sequence of nested sets of competitive digraphs G_k of digraph G_1: $\Gamma_1 = \{G_3\}$, $\Gamma_2 = \{G_2\}$, $\Gamma_3 = \{G_4\}$, for which dominance relation $G_1 \preceq_T G_k$ does not hold. We draw the projections of the stability balls in Figure 3.4 for the same components p_{13} and p_{22} of the vector p as for problem $\mathcal{J}3/n = 2, a_i \le p_i \le b_i/\mathcal{C}_{max}$ from Example 2.1 in Section 2.1 (see Figure 2.5 and Figure 2.6). From Theorem 3.1 and Remark 3.4, it follows that the set $\Lambda^*(G) = \{G_1\} \cup \{\cup_{i=1}^3 \Gamma_i\} = \{G_1\} \cup \{G_k : G_1 \not\preceq_T G_k\} = \{G_1, G_2, G_3, G_4\}$ is a solution of problem $\mathcal{J}3/n = 2, a_i \le p_i \le b_i/\sum \mathcal{C}_i$. Moreover, this solution is minimal since for each digraph $G_k \in \Lambda^*(G)$, there exists a feasible vector for which this digraph is the unique optimal one (see Table 3.8).

Next, we introduce a bound for the stability radii $\overline{\varrho}_s^B(p \in T)$, which is analogous to the bound for $\widehat{\varrho}_s^B(p \in T)$ (see Section 2.4). This bound can restrict the calculation of the relative stability radius.

3.3 Redundant Digraphs for Calculating $\overline{\varrho}_s^B(p \in T)$

To calculate the relative stability radius $\overline{\varrho}_s^B(p \in T)$ of the optimal digraph G_s, we have to use the formulas (3.11) and (3.18) from Theorem 3.3. More exactly, one must compare each set $\Omega_s^v, v = 1, 2, \ldots, \omega_s^T$, of representatives of the family of sets $(H_s^i)_{J_i \in J}$, with the sets $\Omega_k^u, u = 1, 2, \ldots, \omega_k^T$, of representatives of the family of sets $(H_k^i)_{J_i \in J}$ of each digraph $G_k \in B \subseteq \Lambda(G),\ k = 1, 2, \ldots, |B|,\ k \ne s$. The following bound, in which $\overline{r}_{ks}^B$ is defined by formula (3.17), restricts the number of feasible digraphs G_k with which a comparison of the optimal digraph G_s has to be done during the calculation of the relative stability radius $\overline{\varrho}_s^B(p \in T)$.

Lemma 3.4 *If $\overline{\varrho}_s^B(p \in T) < \infty$ and there exists a digraph $G_k \in B$ such that*

$$\overline{r}_{ks}^B \le \frac{L_t^p - L_s^p}{nq - n} \quad \text{for some } t \text{ with } G_t \in B, \tag{3.26}$$

then it is not necessary to consider digraph G_t during the calculation of $\overline{\varrho}_s^B(p \in T)$.

PROOF. Let us compare the optimal digraph G_s with a feasible digraph $G_t,\ t \ne k$. Digraph $G_t, t \ne s$, is a competitive digraph for G_s if we can construct a vector $\overline{x} \in T$ that satisfies the condition 1' at page 63, i.e. equality (3.2) holds: $L_s^{\overline{x}} = L_t^{\overline{x}}$. Moreover, for any given real $\epsilon > 0$, which may be as small as desired, there must exist a vector $\overline{p}^\epsilon \in T$ such that $d(\overline{x}, \overline{p}^\epsilon) = \epsilon$ and inequality (3.3) $L_s^{\overline{p}^\epsilon} > L_t^{\overline{p}^\epsilon}$ is satisfied for digraph G_t (see condition 2'). More precisely, we must construct a vector $\overline{x}$ of the form $\overline{x} = p(\overline{r}_{ts}^B) = (p_{11}(\overline{r}_{ts}^B), p_{12}(\overline{r}_{ts}^B), \ldots, p_{nn_n}(\overline{r}_{ts}^B))$ with the components $p_{ij}(\overline{r}_{ts}^B)$ from the set $\{p_{ij}, p_{ij} + \min\{\overline{r}_{ts}^B, b_{ij} - p_{ij}\}, p_{ij} - \min\{\overline{r}_{ts}^B, p_{ij} - a_{ij}\}\}$ according to formula (3.8). Due to condition 3' (see page 63), the distance $d(p, \overline{x}) = d(p, p(\overline{r}_{ts}^B)) = \overline{r}_{ts}^B$ must achieve minimal value among the distances between the vector p and the other vectors in the polytope T which satisfy both conditions 1' and 2'.

Suppose that the conditions of Lemma 3.4 are satisfied, i.e. inequality (3.26) holds, and the vector $\overline{x} = p(\overline{r}_{ks}^B)$ satisfies both above conditions. We can show that the distance $d(p, p(\overline{r}_{ts}^B))$ cannot become less than the distance $d(p, p(\overline{r}_{ks}^B))$. Next, we show that inequality $\overline{r}_{ks}^B \le \overline{r}_{ts}^B$ follows from condition (3.26). We have:

$$\begin{aligned} \overline{r}_{ks}^B \quad &\le \quad \frac{L_t^p - L_s^p}{nq - n} = \\ &= \quad \frac{\sum_{\nu \in \Omega_t^{u^*}} l^p(\nu) - \sum_{\mu \in \Omega_s^{v^*}} l^p(\mu)}{n(q-1)} = \overline{r}', \end{aligned}$$

where $\Omega_t^{u^*}$, $u^* \in \{1,2,\dots,\omega_t^T\}$, and $\Omega_s^{v^*}$, $v^* \in \{1,2,\dots,\omega_s^T\}$, are critical sets for the digraphs G_t and G_s, respectively. Since $\sum_{O_{ij}\in Q} |n_{ij}(\Omega_k^{u^*}) - n_{ij}(\Omega_s^{v^*})| < n$, we get the following inequalities:

$$\begin{aligned}
\overline{r}' &< \frac{\sum_{\nu\in\Omega_t^{u^*}} l^p(\nu) - \sum_{\mu\in\Omega_s^{v^*}} l^p(\mu)}{(\sum_{O_{ij}\in Q} |n_{ij}(\Omega_k^{u^*}) - n_{ij}(\Omega_s^{v^*})|)(q-1)} \leq \\
&\leq \max_{u\in\{1,2,\dots,\omega_t^T\}} \frac{\sum_{\nu\in\Omega_t^{u}} l^p(\nu) - \sum_{\mu\in\Omega_s^{v^*}} l^p(\mu)}{\sum_{O_{ij}\in Q} |n_{ij}(\Omega_t^{u}) - n_{ij}(\Omega_s^{v^*})|} \leq \\
&\leq \min_{\Omega_s^v\in\Omega_{s,t}} \max_{u\in\{1,2,\dots,\omega_t^T\}} \frac{\sum_{\nu\in\Omega_t^{u}} l^p(\nu) - \sum_{\mu\in\Omega_s^{v}} l^p(\mu)}{\sum_{O_{ij}\in Q} |n_{ij}(\Omega_t^{u}) - n_{ij}(\Omega_s^{v})|} \leq \overline{r}_{ts}^B.
\end{aligned}$$

Thus, the value $\overline{r}_{ts}^B$ cannot become less than $\overline{r}_{ks}^B$ and therefore digraph G_t need not to be considered during the calculation of the relative stability radius $\overline{\varrho}_s^B(p\in T)$.

◇

The above lemma directly implies the following assertion.

Corollary 3.7 *Let the set $\Lambda(G) = \{G_s = G_{i_1}, G_{i_2}, \dots, G_{i_\lambda}\}$ be sorted in non-decreasing order $G_{i_1}, G_{i_2}\dots, G_{i_\lambda}$ of the objective function values $L_{i_1}^p \leq L_{i_2}^p \leq \dots \leq L_{i_\lambda}^p$. If for the currently compared digraph G_{i_k} from the set $B \subseteq \Lambda(G) = \{G_s = G_{i_1}, G_{i_2}, \dots, G_{i_k}, \dots, G_{i_t}, \dots, G_{i_\lambda}\}$ the inequality*

$$\overline{r}_{i_k s}^B \leq \frac{L_{i_t}^p - L_{i_1}^p}{nq - n} \tag{3.27}$$

holds for digraph $G_{i_t} \in B \subseteq \Lambda(G)$ with $L_{i_k}^p \leq L_{i_t}^p$, then it is possible to exclude the digraphs $G_{i_t}, G_{i_{t+1}}, \dots, G_{i_\lambda}$ from further considerations during the calculation of $\overline{\varrho}_s^B(p\in T)$.

Proof. Since the set $B \subseteq \Lambda(G)$ is sorted in non-decreasing order of the objective function values and inequality (3.27) holds for digraph G_{i_t}, inequality

$$\overline{r}_{i_k s}^B \leq \frac{L_{i_j}^p - L_{i_1}^p}{nq - n}$$

holds for each digraph $G_{i_j}, j = t+1, t+2, \dots, |B|$, and due to Lemma 3.4, these digraphs need not to be considered during the calculation of the relative stability radius.

◇

3.4 Algorithms for Problems $\mathcal{J}/a_i \leq p_i \leq b_i/\Phi$, $\Phi \in \{\mathcal{C}_{max}, \sum\mathcal{C}_i\}$

In this section, we focus on criterion $\sum\mathcal{C}_i$. So, using the above mathematical background, we propose first Algorithm $SOL\_\sum\mathcal{C}_i$ for finding a solution $\Lambda^*(G) \subseteq \Lambda(G)$ with 'relatively small' cardinality. As the input data for Algorithm $SOL\_\sum\mathcal{C}_i$, a set of schedules $B \subseteq \Lambda(G)$, which is a solution of problem $\mathcal{J}/a_i \leq p_i \leq b_i/\sum\mathcal{C}_i$, and a vector $p \in T$ of the processing times are used. This algorithm generates a covering of the polytope T (see Theorem 3.1) by nested closed balls $O_r(p)$ with the common center $p \in T$ and different radii r which are relative stability radii $\overline{\varrho}_s^B(p \in T)$ of the same digraph G_s but for different nested sets B.

Let the set $B \subseteq \Lambda(G)$ be a given solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ (in the worst case, the whole set $\Lambda(G)$ of digraphs may be used as such an *input set* B). We also fix a vector $p \in T$ of the processing times and number the digraphs of the set $B = \{G_1, G_2, \dots, G_{|B|}\}$ in non-decreasing order of the values of the objective function. An 'expected' vector of the processing times (or a vector which has been considered in a previous calculation or some other suitable vector from the polytope T) may be used as the *input vector* p in the following algorithm (in square brackets we give the changes of this algorithm in the case of criterion $\mathcal{C}_{max}$).

Algorithm $SOL\_\sum \mathcal{C}_i$ [**Algorithm** $SOL\_\mathcal{C}_{max}$]

Input: A fixed vector $p = (p_{11}, p_{12}, \dots, p_{nn_n}) \in T$, a set $B = \{G_1, G_2, \dots, G_{|B|}\}$ such that $L_1^p \le L_2^p \le \dots \le L_{|B|}^p$ for criterion $\sum \mathcal{C}_i$ [such that $l_1^p \le l_2^p \le \dots \le l_{|B|}^p$ for criterion $\mathcal{C}_{max}$].

Output: The relative stability radius $\overline{\varrho}_1^B(p \in T)$ [$\widehat{\varrho}_1^B(p \in T)$] of the optimal digraph G_1 and a solution $\Lambda^*(G)$.

Step 1: Set $k = 2$ and $\Lambda = \emptyset$;

Step 2: for digraph $G_k \in B$, test dominance relation $G_1 \preceq_T G_k$ using Lemma 3.1 with the objective function $\Phi_k^p = L_k^p$ [$\Phi_k^p = l_k^p$];

Step 3: **IF** $G_1 \not\preceq_T G_k$ **THEN** calculate $\overline{r}_{k1}^B$ [$\widehat{r}_{k1}^B$] using formulas (3.11) and (3.17) [formula (2.25)] for the input vector p;
ELSE GOTO *Step 5*;

Step 4: set $\Lambda := \Lambda \cup \{G_k\}$;

Step 5: set $k := k + 1$;
IF $k \le |B|$ **THEN GOTO** *Step 2*;
ELSE using Theorem 3.3 [Theorem 2.3 and Remark 3.1] calculate
$\overline{\varrho}_1^B(p \in T) = \min\{\overline{r}_{k1}^B : G_1 \not\preceq_T G_k\}$ [$\widehat{\varrho}_1^B(p \in T) = \min\{\widehat{r}_{k1}^B : G_1 \not\preceq_T G_k\}$]
and set $\Lambda^*(G) = \Lambda \cup \{G_1\}$
stop.

It is easy to see that the set $\Lambda^*(G) = \{G_{i_1=1}, G_{i_2}, \dots, G_{i_{|\Lambda^*(G)|}}\}$, $i_1 < i_2 < \dots < i_{|\Lambda^*(G)|}$, generated by Algorithm $SOL\_\sum \mathcal{C}_i$ is a solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$. Indeed, the set $\Lambda^*(G)$ is a subset of the set B which is assumed to be a solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ and the set $\Lambda^*(G)$ includes digraph G_1 and also each digraph G_k, $2 \le k \le |B|$, provided that dominance relation $G_1 \preceq_T G_k$ does not hold.

Along with a solution $\Lambda^*(G)$, Algorithm $SOL\_\sum \mathcal{C}_i$ calculates the value $\overline{r}_{k1}^B$ for each digraph $G_k \in B$ such that dominance relation $G_1 \preceq_T G_k$ does not hold (see Step 3). The value $\overline{r}_{k1}^B$ denotes the largest distance $d(p, p')$ such that inequality $L_1^{p'} > L_k^{p'}$ is guaranteed for each vector $p' \in T$ of the processing times. Therefore, dominance relation $G_1 \preceq_{T^*} G_k$

holds for each polytope $T^* = T \cap O_r(p)$ if $r \leq \overline{r}^B_{k1}$, and we have $G_1 \not\preceq_{T^*} G_k$ for the polytope $T^* = T \cap O_r(p)$ if $r > \overline{r}^B_{k1}$. Let us put the digraphs in the set $\Lambda^*(G)$ in non-decreasing order of the values $\overline{r}^B_{k1}$: $\Lambda^*(G) = \{G_{j_1=1}, G_{j_2}, \dots, G_{j_{|\Lambda^*(G)|}}\}$, where $\overline{r}^B_{j_2 1} \leq \overline{r}^B_{j_3 1} \leq \dots \leq \overline{r}^B_{j_{|\Lambda^*(G)|}1}$. Due to Theorem 3.3, it follows that $\overline{\varrho}^B_1(p \in T) = \overline{r}^B_{j_2 1}$. Similarly, for the set $B \setminus \{G_{j_2}\}$ we have the equality $\overline{\varrho}_1^{B\setminus\{G_{j_2}\}}(p \in T) = \overline{r}^B_{j_3 1}$, and in general, we have the equality $\overline{\varrho}_1^{B\setminus\{\cup^l_{k=2} G_{j_k}\}}(p \in T) = \overline{r}^B_{j_{l+1}1}$, where $1 < l < |\Lambda^*(G)|$. These values $\overline{r}^B_{k1}$ will be used in Algorithm $MINSOL\_\sum\mathcal{C}_i$ which follows. Moreover, they may be used in a realization of the best schedule. Indeed, to realize a solution $\Lambda^*(G)$ (when values $\overline{r}^B_{k1}$ are known), we can start with digraph G_1 which is optimal (or one of the optimal digraphs) for the 'expected' vector $p \in T$ of the processing times. If we will get additional information about the error r of the processing times p_{ij}, we can use r for a suitable modification of the schedule which is currently realized. To this end, we select $\overline{r}^B_{j_l 1}$ such that inequalities $\overline{r}^B_{j_l 1} < r \leq \overline{r}^B_{j_{l+1}1}$ hold, and we can find a better digraph in the set $\cup^l_{u=1} G_{i_u}$ which may be realized further instead of the initial digraph G_1. It is practically important that, if the possible error of the given processing times is no more than r, we have the guarantee that the set $\cup^l_{u=1} G_{i_u}$ contains at least one optimal digraph.

Note that solution $\Lambda^*(G)$ generated by Algorithm $SOL\_\sum\mathcal{C}_i$ may be not minimal. To exclude redundant digraphs, we can test the dominance relation $\preceq_T$ between the digraphs from the set $\Lambda^*(G) \setminus \{G_{i_1=1}\}$ which may be done as follows. First, we exclude all digraphs G_{i_k}, $2 < k \leq |\Lambda^*(G)|$, from the set $\Lambda^*(G)$ for which dominance relation $G_{i_2} \preceq_T G_{i_k}$ holds. To this end, we repeat Algorithm $SOL\_\sum\mathcal{C}_i$ with the set $\Lambda^*(G) \setminus \{G_{i_1=1}\}$ being used instead of set B. Then, similarly, we can exclude all digraphs from the solution which are dominated by digraph G_{i_3} and so on. After no more than $|\Lambda^*(G)| - 2$ repetitions of Algorithm $SOL\_\sum\mathcal{C}_i$ we can remove all redundant digraphs (or an essential part of the redundant digraphs) from the set $\Lambda^*(G)$, and as a result we often get a minimal solution $\Lambda^T(G)$.

Next, we give a formal algorithm for finding a minimal solution on the basis of the above repetitions of Algorithm $SOL\_\sum\mathcal{C}_i$ (see Step 3) and the verification of the strong dominance relation (see Step 5). We set $\Lambda' = \Lambda^*(G) \setminus \{G_{i_1=1}\} = \{G_{i_2}, G_{i_3}, \dots, G_{i_{|\Lambda^*(G)|}}\}$, where $\Lambda^*(G)$ is obtained by Algorithm $SOL\_\sum\mathcal{C}_i$ provided that inequalities $L^p_2 \leq L^p_3 \leq \dots \leq L^p_{|\Lambda^*(G)|}$ hold.

Algorithm $MINSOL\_\sum\mathcal{C}_i$ [**Algorithm** $MINSOL\_\mathcal{C}_{max}$]

Input: A set $\Lambda' = \Lambda^*(G) \setminus \{G_{i_1=1}\} = \{G_{i_2}, G_{i_3}, \dots, G_{i_{|\Lambda^*(G)|}}\}$.
Output: A minimal solution $\Lambda^T(G)$.

Step 1: Set $\Lambda^T(G) = \{G_{i_1=1}\}$;

Step 2: set $B = \Lambda'$ and change the subscripts of the digraphs as follows: $G_u := G_{i_{u+1}}$, $1 \leq u < |\Lambda'| - 1$, i.e. in the following *Steps 3* and *4* the ordered set $(G_{i_2}, G_{i_3}, \dots, G_{|\Lambda'|+1})$ will be referred to as the ordered set $(G_1, G_2, \dots, G_{|\Lambda'|})$;

Step 3: perform Algorithm $SOL\_\sum\mathcal{C}_i$ [Algorithm $SOL\_\mathcal{C}_{max}$] with the input set $B = \{G_1, G_2, \dots, G_{|B|}\}$ defined in *Step 2* and with the same input vector p;

Step 4: set $\Lambda' := \Lambda' \setminus \{G_1\}$ and $\Lambda^T(G) := \Lambda^T(G) \cup \{G_1\}$;
IF $|\Lambda'| \geq 2$ **THEN GOTO** *Step 2*;
ELSE GOTO *Step 5*;

Step 5: **FOR** each digraph $G_s \in \Lambda^T(G)$ **DO**
begin
calculate the vector $p^{(s)} \in T$ such that the strong dominance relation $G_s \prec_{p^{(s)}} G_k$ holds for each digraph $G_k \in \Lambda^T(G) \setminus \{G_s\}$;
IF there does not exist such a vector $p^{(s)} \in T$
THEN set $\Lambda^T(G) := \Lambda^T(G) \setminus \{G_s\}$
end stop.

Obviously, solution $\Lambda^T(G)$ generated by Algorithm $MINSOL\_\sum\mathcal{C}_i$ satisfies the conditions of Theorem 3.2 and hence this solution is minimal. However, Step 5 may be rather complicated, at least it needs to be discussed in more detail. As the desired vector $p^{(s)}$ for digraph G_s, we can test the vector $p_{ij}(r)$ calculated by formula (3.8) in Algorithm $SOL\_\sum\mathcal{C}_i$, where $r = \overline{r}^B_{ks} + \epsilon$ with ϵ being a small positive real number. This vector will be either sufficient for Step 5 or not. In the latter case, i.e. when for the vector $p^{(s)}$ the strong dominance relation $G_s \prec_{p^{(s)}} G_k$ does not hold for at least one digraph $G_k \in \Lambda^T(G) \setminus \{G_s\}$, the realization of Step 5 in Algorithm $MINSOL\_\sum\mathcal{C}_i$ may be more sophisticated.

So, in our experiments we test only Algorithm $MINSOL^*\_\sum\mathcal{C}_i$ (Algorithm $MINSOL^*\_\mathcal{C}_{max}$) which consists of Steps 1 – 4 of the above Algorithm $MINSOL\_\sum\mathcal{C}_i$ (Algorithm $MINSOL\_\mathcal{C}_{max}$). If for the solution $\Lambda^T(G) = \Lambda^*(G)$ generated by Algorithm $MINSOL^*\_\sum\mathcal{C}_i$ the inequality $|\Lambda^*(G)| \leq 2$ holds, then set $\Lambda^*(G)$ obviously satisfies the conditions of Theorem 3.2 and therefore this solution is minimal. If $|\Lambda^*(G)| > 2$, solution $\Lambda^*(G)$ may be not minimal. Indeed, even if $\Lambda^*(G) = \{G_1, G_2, G_3\}$, Algorithm $MINSOL^*\_\sum\mathcal{C}_i$ only guarantees that no digraph from the set $\Lambda^*(G)$ dominates another digraph from the set $\Lambda^*(G)$. However, it might be that two digraphs 'jointly dominate' the remaining one which is not recognized by Algorithm $MINSOL^*\_\sum\mathcal{C}_i$. Nevertheless, Algorithm $MINSOL^*\_\sum\mathcal{C}_i$ often constructs a minimal solution even if $|\Lambda^T(G)| > 2$. Indeed, it is easy to see that, if a schedule is the unique optimal schedule in the interior of its stability region, then dominance relation $\preceq_D$ implies the strong dominance relation $\prec_D$ (except points at the boundary of the stability region, where an optimal schedule usually is not unique). Fortunately, as it was shown in [SSW97] by computational experiments, a mean flow time optimal schedule is uniquely determined for most job shop problems provided that the processing times are non-negative real numbers (not necessarily integers as it is often assumed in classical scheduling theory), and thus, due to the test of the dominance relation $\preceq_D$, Algorithm $MINSOL^*\_\sum\mathcal{C}_i$ usually constructs a minimal solution.

Next, we present three algorithms for constructing a solution B (for any regular criterion Φ) used as input set in Algorithm $SOL\_\sum\mathcal{C}_i$ (Algorithm $SOL\_\mathcal{C}_{max}$). The first one (called Algorithm $EXPL$) is based on an explicit enumeration of all semiactive schedules for the case of a classical job shop problem. The other two algorithms (called $B\&B1$ and $B\&B2$) are of the branch-and-bound type and may be used for the job shop problem $\mathcal{J}/a_i \leq p_i \leq b_i/\Phi$ with uncertain numerical input data and any regular criterion.

Algorithm $EXPL$

Input: A polytope T, a weighted mixed graph $G(p) = (Q(p), A, E), p \in T$.
Output: An optimal digraph $G_s(p)$, a set $B = \{G_1^T, G_2^T, \dots, G_{|B|}^T\} \subseteq \Lambda(G)$.

Step 1: Generate all feasible digraphs $\Lambda(G) = \{G_1, G_2, \dots, G_\lambda\}$ by an explicit enumeration of the permutations of the operations Q_k for $k = 1, 2, \dots, m$, and by testing whether the generated digraph has a circuit;

Step 2: calculate the values Φ_k^a, Φ_k^b and Φ_k^p for each digraph G_k, $k = 1, 2, \dots, \lambda$, and transform each digraph G_k into digraph G_k^T;

Step 3: find a digraph G_s^T such that $\Phi_s^b = \min\{\Phi_k^b \ : \ G_k \in \Lambda(G)\}$;

Step 4: set $B = \{G_k^T \ : \ G_k \in \Lambda(G), \ \Phi_s^b > \Phi_k^a\}$ **stop**.

As follows from Lemma 3.1, the set $\Lambda(G) \setminus B$ contains only digraphs G_k such that dominance relation $G_s^T \preceq_T G_k^T$ holds (see Step 4 of Algorithm $EXPL$), and Algorithm $EXPL$ excludes only such digraphs from the set $\Lambda(G)$. To present branch-and-bound algorithms, we need the following preliminary arguments from [LSSW98].

Both branch-and-bound algorithms realize an implicit enumeration scheme which may be represented by a branching tree. Each vertex of this tree is a mixed graph $G_{(s)} = (Q, A_{(s)}, E_{(s)})$ with $A \subseteq A_{(s)}$ and $E_{(s)} \subseteq E$. The root of the tree is a mixed graph $G = G_{(1)}$, and a pair $G_{(s)}$ and $G_{(k)}$ is connected by the arc $(G_{(s)}, G_{(k)})$ if and only if the mixed graph $G_{(k)}$ is obtained directly from the mixed graph $G_{(s)}$ by orientating one edge. In both branch-and-bound algorithms under consideration, an edge is oriented only if it is a conflict one, i.e. when both orientations of this edge imply a conflict with previously calculated earliest start times. Next, we give a formal definition of a conflict edge. For a mixed graph $G_{(s)} = (Q, A_{(s)}, E_{(s)})$ with $[O_{ij}, O_{uv}] \in E_{(s)}$, let us define the following three digraphs:

$$
\begin{aligned}
G_{s^0} &= (Q, A_{(s)}, \emptyset), \\
G_{s'} &= (Q, A_{(s)} \cup \{(O_{ij}, O_{uv})\}, \emptyset) \text{ and} \\
G_{s''} &= (Q, A_{(s)} \cup \{(O_{uv}, O_{ij})\}, \emptyset).
\end{aligned}
$$

Definition 3.3 *An edge $[O_{ij}, O_{uv}] \in E_{(s)}$ of the mixed graph $G_{(s)}$ is called a conflict edge, if there exists a vector $p \in T$ such that the following inequalities (3.28) and (3.29) hold:*

$$l_{s^0}^p(O_{uv}) < l_{s'}^p(O_{uv}), \tag{3.28}$$

$$l_{s^0}^p(O_{ij}) < l_{s''}^p(O_{ij}). \tag{3.29}$$

Obviously, if inequalities (3.28) and (3.29) hold, then each orientation of the edge $[O_{ij}, O_{uv}]$ implies an increase of the value $l_{s'}^p(O_{uv})$ or the value $l_{s''}^p(O_{ij})$. To verify whether an edge is a conflict one, we can use the following necessary conditions.

Lemma 3.5 *An edge $[O_{ij}, O_{uv}] \in E_{(s)}$ is not a conflict edge if one of the following inequalities (3.30) or (3.31) holds:*

$$l^a_{s^0}(O_{uv}) \ge l^b_{s^0}(O_{ij}) + b_{ij}, \tag{3.30}$$

$$l^a_{s^0}(O_{ij}) \ge l^b_{s^0}(O_{uv}) + b_{uv}. \tag{3.31}$$

PROOF. It is easy to see that inequality (3.28) may hold only if the maximal path ending in vertex O_{uv} includes the arc (O_{ij}, O_{uv}), i.e. if

$$l^p_{s'}(O_{uv}) = l^p_{s^0}(O_{ij}) + p_{ij}. \tag{3.32}$$

Similarly, inequality (3.29) may hold only if

$$l^p_{s''}(O_{ij}) = l^p_{s^0}(O_{uv}) + p_{uv}. \tag{3.33}$$

First, suppose that inequality (3.30) holds. For any vector $p \in T$, we have $l^p_{s^0}(O_{uv}) \ge l^a_{s^0}(O_{uv}) \ge l^b_{s^0}(O_{ij}) + b_{ij} \ge l^p_{s^0}(O_{ij}) + p_{ij}$. Taking into account (3.32), we conclude that inequality $l^p_{s^0}(O_{uv}) \ge l^p_{s'}(O_{uv})$ holds which means that edge $[O_{ij}, O_{uv}]$ is not a conflict one.

Now, suppose that inequality (3.31) holds. For any vector $p \in T$, we have $l^p_{s^0}(O_{ij}) \ge l^a_{s^0}(O_{ij}) \ge l^b_{s^0}(O_{uv}) + b_{uv} \ge l^p_{s^0}(O_{uv}) + p_{uv}$. Taking into account (3.33), we conclude that inequality $l^p_{s^0}(O_{ij}) \ge l^p_{s''}(O_{ij})$ holds which means that edge $[O_{ij}, O_{uv}]$ is not a conflict one.

$\diamond$

For each edge $[O_{ij}, O_{uv}] \in E_{(s)}$, one can calculate a *conflictness measure* as follows:

$$\min\{\max\{0,\ l^p_{s^0}(O_{ij}) + p_{ij} - \bar{l}^p_{s^0}(O_{uv})\},\ \max\{0,\ l^p_{s^0}(O_{uv}) + p_{uv} - \bar{l}^p_{s^0}(O_{ij})\}\},$$

where $\bar{l}^p_{s^0}(O_{ij})$ denotes the *latest start time* of operation O_{ij}, i.e. the difference between the weight of the critical path μ in digraph G_{s^0} and the maximal weight of the path in G_{s^0} starting from vertex O_{ij}:

$$\bar{l}^p_s(O_{ij}) = l^p(\mu) - \sum_{O_{uv} \in [\nu]} p_{uv},$$

where path ν has the maximal weight among all paths in digraph G_{s^0} starting from O_{ij} and ending in vertex O_{ln_l}, $J_l \in J$. The conflictness measure gives the smallest possible increase of the earliest start time of the operation due to the orientation of this edge (e.g. for a non-conflict edge this measure is equal to zero). So, in order to branch a set $\Lambda(G_{(s)})$ into two subsets $\Lambda(G_{(s')})$ and $\Lambda(G_{(s'')})$, where

$$G_{(s')} = (Q,\ A_{(s)} \cup \{(O_{ij}, O_{uv})\},\ E_{(s)} \setminus \{[O_{ij}, O_{uv}]\}) \text{ and}$$
$$G_{(s'')} = (Q,\ A_{(s)} \cup \{[O_{uv}, O_{ij}]\},\ E_{(s)} \setminus \{[O_{ij}, O_{uv}]\}),$$

we select the edge $[O_{ij}, O_{uv}]$ which has the largest value of the conflictness measure. We use the following lower bound in both branch-and-bound algorithms. For any digraph $G_t = (Q, A_{(s)} \cup A_t, \emptyset) \in \Lambda(G_{(s)})$, the bound

$$\sum_{i=1}^{n} l^p_t(O_{in_i}) \ge \sum_{i=1}^{n} l^p_{s^0}(O_{in_i}) \tag{3.34}$$

is valid since the set of arcs in the digraph $G_{s^0} = (Q, A_{(s)}, \emptyset)$ is a subset of the arcs in digraph G_t. Note that, if digraph $G_{(s)}$ has no conflict edge, there exists a digraph $G_t \in \Lambda(G_{(s)})$ such that condition (3.34) is realized as equality. To construct such a digraph, we have to replace each remaining edge $[O_{ij}, O_{uv}] \in E_s$ by the arc (O_{ij}, O_{uv}) if inequality (3.30) holds, or by the arc (O_{uv}, O_{ij}) if inequality (3.31) holds. Obviously, for each p_{ij} and p_{uv} with $a_{ij} \leq p_{ij} \leq b_{ij}$ and $a_{uv} \leq p_{uv} \leq b_{uv}$, all operations in the resulting digraph will have the same earliest start times as in the digraph G_{s^0}. We use the latter as a stopping rule for branching the set $\Lambda(G_{(s)})$. Next, we present an algorithm for constructing a set of k schedules which are the best for the input vector $p \in T$ of the processing times and which will be used as the input set B in Algorithm $SOL\_\sum\mathcal{C}_i$ or Algorithm $SOL\_\mathcal{C}_{max}$ depending on the chosen objective function values $\Phi^p_s = L^p_s$ and $\Phi^p_s = l^p_s$, respectively.

Algorithm $B\&B1$

Input: A polytope T, a weighted mixed graph $G(p) = (Q(p), A, E), p \in T$, a number k of the best generated digraphs.
Output: An optimal digraph $G_s(p)$, a set $B = \{G_1, G_2, \ldots, G_k\} \subseteq \Lambda(G)$.

Step 1: Set $X = \{G\} := \{G_{(1)}\},\ Y = \emptyset$ and $\Phi = \infty$;

Step 2: **IF** $X = \emptyset$ **THEN GOTO** *Step 8*;
ELSE select a mixed graph $G_{(s)} \in X$ with the smallest value $\Phi^p_{s^0}$ and set $X := X \setminus \{G_{(s)}\}$;

Step 3: **IF** the mixed graph $G_{(s)}$ has no conflict edge **THEN GOTO** *Step 6*;

Step 4: select a conflict edge $[O_{ij}, O_{uv}] \in E_{(s)}$ with the largest conflictness measure;

Step 5: **IF** $\Phi^p_{s'} < \Phi$ **THEN** set $X := X \cup \{G_{(s')}\}$;
IF $\Phi^p_{s''} < \Phi$ **THEN** set $X := X \cup \{G_{(s'')}\}$; **GOTO** *Step 2*;

Step 6: **IF** $|Y| < k$ **THEN** set $Y := Y \cup \{G_{(s)}\}$; **GOTO** *Step 2*;
ELSE IF $\Phi^p_{s^0} < \Phi$ (where $\Phi = \Phi^p_t$) **THEN** set $Y := Y \cup \{G_{(s)}\} \setminus \{G_{(t)}\}$;

Step 7: calculate $\Phi = \max\{\Phi^p_t \ : \ G_{(t)} \in Y\}$; **GOTO** *Step 2*;

Step 8: construct the set $\Lambda(G_{(t)})$ for each mixed graph $G_{(t)} \in Y$;

Step 9: select a subset B of k best digraphs from the set $\cup_{G_{(t)} \in Y} \Lambda(G_{(t)})$;

Step 10: calculate $\Phi^* = \min\{\Phi^b_s \ : \ G_s \in B\}$ and set $B := B \setminus \{G_t \ : \ \Phi^a_t \geq \Phi^*\}$
stop.

In Algorithm $B\&B1$, the lower bound for the objective function is calculated in Step 7, branching is realized in Step 5, and the stopping rule of branching is realized in Step 3. Step 6 has a special form in order to construct the k best schedules (instead of only one optimal schedule). Steps 8 and 9 are also necessary only if $k > 1$. Indeed, if $k = 1$, then it is sufficient to consider only one best schedule from the set $\Lambda(G_{(s)})$, and for any mixed graph $G_{(s)} = (Q, A_{(s)}, E_{(s)})$, the set $\Lambda(G_{(s)})$ has at least one best schedule $G_u \in \Lambda(G_{(s)})$ for which Φ^p_u reaches the minimal possible value Φ^p_s, where $G_{s^0} = (Q, A_{(s)}, \emptyset)$ (condition (3.34) turns into an equality). But, if $k > 1$, we have to generate also other schedules from the set $\Lambda(G_{(s)})$. Unfortunately, we cannot use Algorithm $EXPL$ for a fast generation of the set $\Lambda(G_{(s)})$ because the edges of the set $E \setminus E_{(s)}$ are already oriented. Step 8 realizes a procedure based on the sequential orientation of non-conflict edges, which is essentially slower than the permutation enumeration used in Algorithm $EXPL$.

Using sufficiency of Lemma 3.1, Algorithm $B\&B2$ aims to construct a set of schedules which necessarily dominate all other schedules from the set $\Lambda(G)$ in the polytope T. Steps 1-5 and Steps 8-10 in Algorithm $B\&B2$ are similar to those in Algorithm $B\&B1$. So, in the following we describe only Steps 6 and 7 of Algorithm $B\&B2$, which are different from those in Algorithm $B\&B1$.

Algorithm $B\&B2$ (specific part)

Step 6: **IF** $\Phi^a_{s^0} \le \Phi$ **THEN** set $Y := Y \cup \{G_{(s)}\}$;

Step 7: calculate $\Phi = \min\{\Phi^b_{(t)} \ : \ G_{(t)} \in Y\}$; **GOTO** *Step 2.*

In Section 4.4, we present computational results for randomly generated classical job shop problems solved by the above algorithms coded in Fortran-77.

Example 3.1 (continued). As it was noted, the solution $\Lambda^*(G)$ and the minimal solution $\Lambda^T(G)$ of the scheduling problem with uncertain processing times may be not unique. From Remark 2.5 it follows that fixing the vector $p \in T$ and the choice of an optimal digraph $G_s(p)$ have a large influence on the resulting solution for criterion $\mathcal{C}_{max}$. For the job shop problem $\mathcal{J}3/n=3, a_i \le p_i \le b_i/\sum \mathcal{C}_i$ from Example 3.1, we find a solution set $\Lambda^*(G)$ with different initial real vectors $p \in T$ (see column 1 in Table 3.9) the components of which are taken from the closed intervals $[a_{ij}, b_{ij}]$ (vectors a and b are given in Table 3.1). The three algorithms $EXPL$, $B\&B1$ and $B\&B2$ construct the same set $B = \{G_1, G_2, \dots, G_{12}\}$ for Example 3.1 as it was constructed with the initial vector p^0 (see page 59), but the digraphs from the set B form another order according to non-decreasing mean flow time objective function values with different feasible vectors (see column 3 in Table 3.9). We calculate the following sums of job completion times with the initial vectors $p^1, p^2, \dots, p^9$ from Table 3.9 (to avoid a confusion, we leave the same subscript of the digraphs indicating the location according to non-decreasing values of the function $\sum \mathcal{C}_i$ calculated with the vector p^0 (see page 59)):
$L_5^{p^1} = 482, L_2^{p^1} = 486, L_1^{p^1} = 486, L_9^{p^1} = 512, L_3^{p^1} = 516, L_4^{p^1} = 536, L_7^{p^1} = 566, L_6^{p^1} = 596, L_8^{p^1} = 636, L_{12}^{p^1} = 666, L_{11}^{p^1} = 676, L_{10}^{p^1} = 686;$
$L_2^{p^2} = 450, L_1^{p^2} = 470, L_5^{p^2} = 470, L_9^{p^2} = 500, L_3^{p^2} = 500, L_4^{p^2} = 520, L_7^{p^2} = 550, L_6^{p^2} = 580, L_8^{p^2} = 580, L_{10}^{p^2} = 630, L_{12}^{p^2} = 650, L_{11}^{p^2} = 660;$
$L_2^{p^3} = 455, L_1^{p^3} = 460, L_5^{p^3} = 505, L_3^{p^3} = 510, L_4^{p^3} = 515, L_7^{p^3} = 520, L_8^{p^3} = 520, L_9^{p^3} = 555, L_6^{p^3} = 570, L_{10}^{p^3} = 575, L_{11}^{p^3} = 635, L_{12}^{p^3} = 645;$
$L_1^{p^4} = 365, L_2^{p^4} = 370, L_3^{p^4} = 395, L_4^{p^4} = 415, L_5^{p^4} = 420, L_6^{p^4} = 435, L_8^{p^4} = 435, L_7^{p^4} =$

Table 3.9: *Solution of problem $\mathcal{J}3/n\!=\!3, a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$ for different initial vectors $p \in T$*

Initial vector $p^j \in T$	G_s	Set B	$G_k,\ G_s \not\leq_T G_k$	$\overline{r}^B_{ks}$
1	2	3	4	5
$p^1 = (60, 20, 46, 30, 70, 80, 50, 30)$	G_5	$B = \{G_5, G_2, G_1, G_9, G_3, G_4, G_7, G_6, G_8, G_{12}, G_{11}, G_{10}\}$	G_1	$\overline{\varrho}^B_5(p^1 \in T) = \overline{r}^B_{15} = 0.5$
			G_2	$\overline{r}^B_{25} = 0.6667$
			G_3	$\overline{r}^B_{35} = 5.6667$
			G_4	$\overline{r}^B_{45} = 7.7143$
			G_6	$\overline{r}^B_{65} = 12.3333$
			G_8	$\overline{r}^B_{85} = 12.8333$
			G_7	$\overline{r}^B_{75} = 14$
			G_{10}	$\overline{r}^B_{10,5} = 19$
	Solution: $\Lambda^*(G) = \{G_1, G_2, G_3, G_4, G_5, G_6, G_8, G_7, G_{10}\}$			
	Minimal solution: $\Lambda^T(G) = \{G_1, G_2, G_5\}$			
$p^2 = (60, 20, 50, 30, 50, 80, 50, 30)$	G_2	$B = \{G_2, G_1, G_5, G_9, G_3, G_4, G_7, G_6, G_8, G_{10}, G_{12}, G_{11}\}$	G_1	$\overline{\varrho}^B_2(p^2 \in T) = \overline{r}^B_{12} = 3.3333$
			G_5	$\overline{r}^B_{52} = 15$
			G_4	$\overline{r}^B_{42} = 16.6667$
			G_8	$\overline{r}^B_{82} = 16.6667$
			G_7	$\overline{r}^B_{72} = 20$
	Solution: $\Lambda^*(G) = \{G_1, G_2, G_4, G_5, G_7, G_8\}$			
	Minimal solution: $\Lambda^T(G) = \{G_1, G_2, G_5\}$			
$p^3 = (80, 20, 50, 10, 65, 60, 45, 35)$	G_2	$B = \{G_2, G_1, G_5, G_3, G_4, G_7, G_8, G_9, G_6, G_{10}, G_{11}, G_{12}\}$	G_1	$\overline{\varrho}^B_2(p^3 \in T) = \overline{r}^B_{12} = 1$
			G_8	$\overline{r}^B_{82} = 11$
			G_4	$\overline{r}^B_{42} = 15$
			G_5	$\overline{r}^B_{52} = 18.3333$
			G_7	$\overline{r}^B_{72} = 20$
	Solution: $\Lambda^*(G) = \{G_1, G_2, G_4, G_5, G_7, G_8\}$			
	Minimal solution: $\Lambda^T(G) = \{G_1, G_2, G_5\}$			
$p^4 = (60, 20, 45, 10, 50, 60, 30, 30)$	G_1	$B = \{G_1, G_2, G_3, G_4, G_5, G_6, G_8, G_7, G_9, G_{10}, G_{11}, G_{12}\}$	G_2	$\overline{\varrho}^B_1(p^4 \in T) = \overline{r}^B_{21} = 1.25$
			G_5	$\overline{r}^B_{51} = 18.3333$
	Solution: $\Lambda^*(G) = \{G_1, G_2, G_5\}$			
	Minimal solution: $\Lambda^T(G) = \{G_1, G_2, G_5\}$			
$p^5 = (70, 30, 52.5, 20, 60, 70, 40, 35)$	G_1	$B = \{G_1, G_2, G_3, G_4, G_5, G_6, G_7, G_8, G_9, G_{10}, G_{11}, G_{12}\}$	G_2	$\overline{\varrho}^B_1(p^5 \in T) = \overline{r}^B_{21} = 2.5$
			G_5	$\overline{r}^B_{51} = 9.2857$
	Solution: $\Lambda^*(G) = \{G_1, G_2, G_5\}$			
	Minimal solution: $\Lambda^T(G) = \{G_1, G_2, G_5\}$			
$p^6 = (80, 40, 60, 30, 70, 80, 50, 40)$	G_1	$B = \{G_1, G_2, G_4, G_5, G_3, G_6, G_7, G_8, G_9, G_{11}, G_{10}, G_{12}\}$	G_2	$\overline{\varrho}^B_1(p^6 \in T) = \overline{r}^B_{21} = 8$
			G_5	$\overline{r}^B_{51} = 18.75$
	Solution: $\Lambda^*(G) = \{G_1, G_2, G_5\}$			
	Minimal solution: $\Lambda^T(G) = \{G_1, G_2, G_5\}$			
$p^7 = (80, 40, 60, 30, 65, 60, 30, 35)$	G_1	$B = \{G_1, G_4, G_7, G_2, G_5, G_8, G_3, G_6, G_{10}, G_9, G_{11}, G_{12}\}$	G_2	$\overline{\varrho}^B_1(p^7 \in T) = \overline{r}^B_{21} = 12$
			G_5	$\overline{r}^B_{51} = 19.2857$
	Solution: $\Lambda^*(G) = \{G_1, G_2, G_5\}$			
	Minimal solution: $\Lambda^T(G) = \{G_1, G_2, G_5\}$			

Table 3.9 (continuation): *Solution of problem $\mathcal{J}3/n=3, a_i \le p_i \le b_i/\sum \mathcal{C}_i$ for different initial vectors $p \in T$*

Initial vector $p^j \in T$	G_s	Set B	$G_k,\ G_s \not\preceq_T G_k$	$\overline{r}^B_{ks}$
1	2	3	4	5
$p^8=(60,20,49,$ $30,69,80,50,40)$	G_2	$B=\{G_2,G_5,G_1,G_9,G_3,G_4,$ $G_7,G_6,G_8,G_{12},G_{10},G_{11}\}$	G_5	$\overline{\varrho}^B_2(p^8 \in T) = \overline{r}^B_{52} = 0$
			G_1	$\overline{r}^B_{12} = 0.1667$
			G_4	$\overline{r}^B_{42} = 15$
			G_8	$\overline{r}^B_{82} = 17.0909$
			G_7	$\overline{r}^B_{72} = 20$
	Solution: $\Lambda^*(G) = \{G_1,G_2,G_4,G_5,G_7,G_8\}$ Minimal solution: $\Lambda^T(G) = \{G_1,G_2,G_5\}$			
	G_5	$B=\{G_5,G_2,G_1,G_9,G_3,G_4,$ $G_7,G_6,G_8,G_{12},G_{10},G_{11}\}$	G_2	$\overline{\varrho}^B_5(p^8 \in T) = \overline{r}^B_{25} = 0$
			G_1	$\overline{r}^B_{15} = 0.125$
			G_3	$\overline{r}^B_{35} = 5.1667$
			G_4	$\overline{r}^B_{45} = 7.6250$
			G_6	$\overline{r}^B_{65} = 12$
			G_8	$\overline{r}^B_{85} = 12.5455$
			G_7	$\overline{r}^B_{75} = 14$
			G_{10}	$\overline{r}^B_{10,5} = 18.8$
	Solution: $\Lambda^*(G) = \{G_1,G_2,G_3,G_4,G_5,G_6,G_7,G_8,G_{10}\}$ Minimal solution: $\Lambda^T(G) = \{G_1,G_2,G_5\}$			
$p^9=(60,20,50,$ $30,70,80,50,30)$	G_1	$B=\{G_1,G_2,G_5,G_9,G_3,G_4,$ $G_7,G_6,G_8,G_{12},G_{11},G_{10}\}$	G_2	$\overline{\varrho}^B_1(p^9 \in T) = \overline{r}^B_{21} = 0$
			G_5	$\overline{r}^B_{52} = 0$
	Solution: $\Lambda^*(G) = \{G_1,G_2,G_5\}$ Minimal solution: $\Lambda^T(G) = \{G_1,G_2,G_5\}$			
	G_2	$B=\{G_2,G_1,G_5,G_9,G_3,G_4,$ $G_7,G_6,G_8,G_{12},G_{11},G_{10}\}$	G_1	$\overline{\varrho}^B_2(p^9 \in T) = \overline{r}^B_{12} = 0$
			G_5	$\overline{r}^B_{52} = 0$
			G_4	$\overline{r}^B_{42} = 15$
			G_8	$\overline{r}^B_{82} = 17.2727$
			G_7	$\overline{r}^B_{72} = 20$
	Solution: $\Lambda^*(G) = \{G_1,G_2,G_4,G_5,G_7,G_8\}$ Minimal solution: $\Lambda^T(G) = \{G_1,G_2,G_5\}$			
	G_5	$B=\{G_5,G_1,G_2,G_9,G_3,G_4,$ $G_7,G_6,G_8,G_{12},G_{11},G_{10}\}$	G_1	$\overline{\varrho}^B_5(p^9 \in T) = \overline{r}^B_{15} = 0$
			G_2	$\overline{r}^B_{25} = 0$
			G_9	$\overline{r}^B_{95} = 5$
			G_4	$\overline{r}^B_{45} = 7.1429$
			G_6	$\overline{r}^B_{65} = 12$
			G_8	$\overline{r}^B_{85} = 12.7273$
			G_7	$\overline{r}^B_{75} = 14$
			G_{10}	$\overline{r}^B_{10,5} = 19$
	Solution: $\Lambda^*(G) = \{G_1,G_2,G_4,G_5,G_6,G_7,G_8,G_9,G_{10}\}$ Minimal solution: $\Lambda^T(G) = \{G_1,G_2,G_5\}$			

$445, L^{p^4}_9 = 450, L^{p^4}_{10} = 485, L^{p^4}_{11} = 495, L^{p^4}_{12} = 505;$
$L^{p^5}_1 = 438, L^{p^5}_2 = 460, L^{p^5}_3 = 497.5, L^{p^5}_4 = 502.5, L^{p^5}_5 = 510, L^{p^5}_6 = 532.5, L^{p^5}_7 = 538.5, L^{p^5}_8 = 547.5, L^{p^5}_9 = 560, L^{p^5}_{10} = 612.5, L^{p^5}_{11} = 617.5, L^{p^5}_{12} = 627.5;$
$L^{p^6}_1 = 510, L^{p^6}_2 = 550, L^{p^6}_4 = 590, L^{p^6}_5 = 600, L^{p^6}_3 = 600, L^{p^6}_6 = 630, L^{p^6}_7 = 630, L^{p^6}_8 =$

660, $L_9^{p^6} = 670, L_{11}^{p^6} = 740, L_{10}^{p^6} = 740, L_{12}^{p^6} = 750$;
$L_1^{p^7} = 460, L_4^{p^7} = 515, L_7^{p^7} = 520, L_2^{p^7} = 520, L_5^{p^7} = 550, L_8^{p^7} = 560, L_3^{p^7} = 580, L_6^{p^7} = 585, L_{10}^{p^7} = 635, L_9^{p^7} = 640, L_{11}^{p^7} = 660, L_{12}^{p^7} = 670$;
$L_2^{p^8} = 497, L_5^{p^8} = 497, L_1^{p^8} = 498, L_9^{p^8} = 527, L_3^{p^8} = 528, L_4^{p^8} = 558, L_7^{p^8} = 578, L_6^{p^8} = 618, L_8^{p^8} = 646, L_{12}^{p^8} = 698, L_{10}^{p^8} = 706, L_{11}^{p^8} = 708$;
$L_1^{p^9} = 490, L_2^{p^9} = 490, L_5^{p^9} = 490, L_9^{p^9} = 520, L_3^{p^9} = 520, L_4^{p^9} = 540, L_7^{p^9} = 570, L_6^{p^9} = 600, L_8^{p^9} = 640, L_{12}^{p^9} = 670, L_{11}^{p^9} = 680, L_{10}^{p^9} = 690$.

First, we construct a solution $\Lambda^*(G)$ by Algorithm $SOL\_\sum\mathcal{C}_i$, and then a minimal solution $\Lambda^T(G)$ by Algorithm $MINSOL^*\_\sum\mathcal{C}_i$. In column 2 of Table 3.9, we give the chosen optimal digraph $G_s(p^j)$ for the fixed vector p^j. The set $B := B \setminus \{G_s\}$ ordered according to non-decreasing values $L_u^{p^j}, u \in \{1, 2, \ldots, |B|\}$, is presented in column 4. For digraph $G_k \in B$, we test the dominance relation $G_s \preceq_T G_k$ using Lemma 3.1 with the objective function values $\Phi_s^p = L_s^p$ (see Step 2 of Algorithm $SOL\_\sum\mathcal{C}_i$). For all digraphs G_k with $G_s \not\preceq_T G_k$ (see column 4), we calculate the value $\overline{r}_{ks}^B$ using formulas (3.11) and (3.17) from Theorem 3.3 for the input vector p^j. Column 5 presents a non-decreasing order of the values $\overline{r}_{ks}^B$ calculated according to (3.17).

Due to Theorem 3.3, it follows that the minimal value of $\overline{r}_{ks}^B$ is equal to the relative stability radius $\overline{\varrho}_s^B(p^j \in T)$. An optimal digraph G_s and all digraphs G_k, for which the dominance relation $G_s \preceq_T G_k$ does not hold, form the solution $\Lambda^*(G)$ of the scheduling problem $\mathcal{J}3/n{=}3, a_i{\le}p_i{\le}b_i/\sum\mathcal{C}_i$.

As we see, a choice of the initial vector $p \in T$ gives different solution sets. The best choice of such a feasible vector is still an open question. We fix, for example, the vector p^4 (p^6) equal to the given lower bound a (upper bound b, respectively) of the feasible polytope T (see Table 3.1), and the vector p^5 with components $p_{ij} = \frac{1}{2}(b_{ij} - a_{ij})$. Such a choice gives the following solution set $\Lambda^*(G) = \Lambda^T(G) = \{G_1, G_2, G_5\}$. Note that there is no minimal solution set with a smaller cardinality than $\Lambda^T(G) = \{G_1, G_2, G_5\}$. Moreover, each of the digraphs G_1, G_2 and G_5 is the unique optimal one for some vector $p \in T$, i.e., for example, the following strong dominance relations hold:

$G_1 \prec_{p^0} G_k, \quad G_k \in \Lambda(G) \setminus \{G_1\}$,
$G_2 \prec_{p^2} G_k, \quad G_k \in \Lambda(G) \setminus \{G_2\}$, and
$G_5 \prec_{p^1} G_k, \quad G_k \in \Lambda(G) \setminus \{G_5\}$ (see column 3 of Table 3.9).

It means that there is no proper subset of the set $\{G_1, G_2, G_5\}$ which is a solution of problem $\mathcal{J}3/n{=}3, a_i{\le}p_i{\le}b_i/\sum\mathcal{C}_i$ and so the solution $\Lambda^*(G) = \{G_1, G_2, G_5\}$ is minimal in the sense of inclusion and in the sense of cardinality equal to 3. As we see, our developed Algorithm $SOL\_\sum\mathcal{C}_i$ may construct some redundant schedules, which are not necessarily in a minimal solution set $\Lambda^T(G)$ of problem $\mathcal{J}/a_i{\le}p_i{\le}b_i/\sum\mathcal{C}_i$.

As it was noted, for the scheduling problem with the makespan criterion (see Remark 2.5) not only fixing the initial vector $p \in T$ has a large influence on the resulting solution, but also the choice of an optimal digraph G_s for the further calculations, if it is not uniquely determined. For the vectors p^8 and p^9, the optimal schedule is not unique. For example, there are two optimal digraphs $G_2(p^8)$ and $G_5(p^8)$ for the vector p^8, therefore we run Algorithm $SOL\_\sum\mathcal{C}_i$ twice. First, we order digraphs in the set B as follows $\{G_2, G_5, G_1, G_9, G_3, G_4, G_7, G_6, G_8, G_{12}, G_{10}, G_{11}\}$ and we make all calculations according to Algorithm $SOL\_\sum\mathcal{C}_i$ for the first digraph G_2 in the set B. Secondly, we order digraphs in the set B as follows $\{G_5, G_2, G_1, G_9, G_3, G_4, G_7, G_6, G_8, G_{12}, G_{10}, G_{11}\}$ and we make all calculations for the first digraph G_5. Thus in the first case, solution $\Lambda^*(G)$ consists of six schedules since there are five digraphs G_k for which the dominance relation $G_2 \preceq_T G_k$ does not hold. In the second case, the solution $\Lambda^*(G)$ consists of nine schedules since there are eight digraphs G_k, $G_5 \not\preceq_T G_k$. Since there are three optimal digraphs for the vector p^9 at all, the corresponding cardinalities of the obtained solutions $\Lambda^*(G)$ are 3, 6 and 9, respectively (see Table 3.9).

As we see from Table 3.9, the covering of the polytope T by the minimal number of stability balls (cardinality-minimal covering) is an interesting question. The cardinality-minimal covering seems to be a more difficult problem than an inclusion-minimal covering. However, this dissertation deals only with the investigation of inclusion-minimal coverings. At least we do not know a practicable algorithm for constructing a cardinality-minimal covering of polytope T.

3.5 Resume and Notations

In Section 3.1, we have defined a solution of job shop problems with uncertain processing times. The network presentation of the structural input data (precedence and capacity constraints) and a minimal solution have been discussed in Section 3.1, where the decision process is presented as the construction of a set of schedules (digraphs) which dominate other schedules. To solve problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$, we developed an approach for calculating the relative stability radius. Theorem 3.3 generalized the results from [BSW96], where the stability radius $\overline{\varrho}_s^B(p \in T)$ was investigated for the special case when $B = \Lambda(G)$ and the whole space R^q_+ being used instead of the polytope T. Theorem 3.5 (Theorem 3.4) provides necessary and sufficient conditions for a zero (for an infinitely large, respectively) relative stability radius.

In Sections 2.4 and 3.3, upper bounds for the relative stability radii $\widehat{\varrho}_s^B(p \in T)$ and $\overline{\varrho}_s^B(p \in T)$ have been used to restrict the number of digraphs compared with an optimal digraph for calculating the relative stability radius. These bounds have been derived for $\widehat{\varrho}_s(p)$ and $\overline{\varrho}_s(p)$ in [SSW97] and will be used in Chapter 4.

Note that in this dissertation the term 'time' is used in three different senses: namely, as the time for processing an operation, as the time for decision-making, and as the time for running an algorithm. We hope that these different uses of the same word do not cause any confusion.

The main notations used for the job shop problem are summarized in Table 3.10, which follows.

Table 3.10: *Notations for the job shop problem*

Symbols	Description
O_{ij}	Operation of job J_i at the technological stage $j \in \{1, 2, \dots, n_i\}$
O_{in_i}	Last operation of job $J_i,\ 1 \le i \le n$
Q	Set of all operations: $Q = \cup_{v=1}^{m} Q_v = \{Q_{ij}\ :\ J_i \in J, j = 1, 2, \dots, n_i\}$
q	Number of operations: $q = \vert Q\vert = \sum_{i=1}^{n} n_i = \sum_{k=1}^{m} \vert Q_k\vert$
$M_{k_{ij}}$	Machine on which operation O_{ij} has to be processed
s_{ij}	Start time of operation O_{ij}
c_{ij}	Completion time of operation O_{ij}
p_{ij}	Processing time of operation O_{ij}
a_{ij}	Given lower bound for the processing time of operation O_{ij}
b_{ij}	Given upper bound for the processing time of operation O_{ij}
$l_k^p(O_{ij})$	Earliest start time of operation O_{ij} in digraph G_k
L_k^p	Sum of job completion times for schedule $G_k(p)$ with processing times p: $\Phi_p^k = L_k^p = \sum_{i=1}^{n} \max_{\nu \in H_k^i} l^p(\nu)$ (critical sum of weights)
Ω_k^u	Set of representatives of the family of sets $(H_k^i)_{J_i \in J}$
ω_k	Number of different sets of representatives for digraph $G_k :\ \omega_k = \prod_{i=1}^{n} \vert H_k^i\vert$
Ω_k	Set of all sets of representatives for digraph G_k: $\{\Omega_k^u :\ u = 1, 2, \dots, \omega_k\}$
ω_k^T	Number of different sets of representatives for digraph $G_k^T :\ \omega_k^T = \prod_{i=1}^{n} \vert H_k^i(T)\vert$
$n_{ij}(\Omega_k^u)$	Number of copies of vertex O_{ij} contained in the multiset $\{[\nu] : \nu \in \Omega_k^u\}$
$\Omega_k^{u^*}$	Critical set of digraph $G_k(p) \in \Lambda(G),\ u^* \in \{1, 2, \dots, \omega_k\}$
$\Omega_k(p)$	Set of all critical sets $\Omega_k^{u^*}$ of digraph $G_k \in \Lambda(G)$ with processing times $p \in R_+^q$
Ω_{sk}	Set of all sets of representatives $\Omega_s^v, v \in \{1, 2, \dots, \omega_k\}$, with the following property: There does not exist a set Ω_k^u with $n_{ij}(\Omega_s^v) \le n_{ij}(\Omega_k^u)$ for each $i = 1, 2, \dots, n$ and each $j = 1, 2, \dots, n_i$
Ω_{sk}^*	Set of all sets of representatives $\Omega_s^v,\ v \in \{1, 2, \dots, \omega_s^T\}$, such that for the vector $p^* \in T$ defined by formula (3.15) inequality $\sum_{\mu \in \Omega_s^v} l^{p^*}(\mu) > \sum_{\nu \in \Omega_k^u} l^{p^*}(\nu)$ holds for each set $\Omega_k^u,\ u \in \{1, 2, \dots, \omega_k^T\}$

Chapter 4

Computational Results

In Sections 4.1 - 4.3 of this chapter, we present computational results for randomly generated job shop problems $\mathcal{J}//\mathcal{C}_{max}$ and $\mathcal{J}//\sum\mathcal{C}_i$. In Section 4.1, we develop algorithms for calculating the stability radii $\widehat{\varrho}_s(p)$ and $\overline{\varrho}_s(p)$ on the basis of the formulas from [BSW96, Sot91, SWW98] (see Chapter 1). Section 4.2 investigates the influence of errors and possible changes of the processing times on the property of a schedule to be optimal. To this end, extensive numerical experiments with randomly generated job shop problems, which satisfy Assumptions 1 and 2, are performed and discussed. Computational results for randomly generated job shop problems $\mathcal{J}m/n{=}k, a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ and $\mathcal{J}m/n = k, a_i \leq p_i \leq b_i/\sum\mathcal{C}_i$ with uncertain numerical input data by testing exact and heuristic algorithms derived in Section 3.4 are discussed in Section 4.4 and Section 4.5. All algorithms were coded in Fortran-77, tested on a PC and the computational results below were published in [LSSW98, SSW97, Sotskova99a].

4.1 Calculation of the Stability Radius

This section is devoted to the calculation of the stability radius of an optimal schedule for a job shop problem, when the objective is to minimize mean or maximum flow times. The used approach may be regarded as an *a posteriori* analysis. We investigate the influence of errors and possible changes of the processing times on the property of a schedule to be optimal. To this end, extensive numerical experiments with randomly generated job shop problems are performed. Due to the developed software, we have the possibility to compare the values of the stability radii, the numbers of optimal schedules and some other 'numbers' for the two criteria $\mathcal{C}_{max}$ and $\sum\mathcal{C}_i$. The main question we try to answer is how large the stability radius is, on average, for randomly generated job shop problems.

The formulas for calculating the stability radii $\widehat{\varrho}_s(p)$ and $\overline{\varrho}_s(p)$ of an optimal digraph $G_s(p)$, derived in [BSW96, Sot91, SWW98] (see Chapter 1), were coded in Fortran-77. Due to these formulas (1.9), (1.10) and (1.14), (1.15), the calculation of the stability radii based on a direct comparison of the paths of an optimal digraph G_s and of each feasible digraph $G_k \in \Lambda(G), k \neq s$, for C_{max} and subsets of paths of G_s and of $G_k \in \Lambda(G), k \neq s$, for $\sum C_i$ is very complicated and time-consuming (even for the small Example 1.1 it is only for the makespan criterion possible to do this calculation 'by hand' without a computer). Nevertheless, such an 'unpractical' calculation for sample problems allows to derive some properties of the job shop problems, which may be used in practically efficient methods for determining lower and/or upper bounds for the stability radii. Computational results for randomly generated job shop problems are presented in this chapter. The stability radii have been calculated for more than 10,000 randomly generated job shop problems.

Next, we present the formal algorithm for calculating the stability radius $\widehat{\varrho}_s(p)$ on the basis of the coded formulas (1.9) and (1.10). We calculate the set of stability radii $\widehat{\mathcal{R}} = \{\widehat{\varrho}_1(p), \widehat{\varrho}_2(p), \dots, \widehat{\varrho}_{opt}(p)\}$ for the set of all optimal digraphs $G_1(p), G_2(p), \dots, G_{opt}(p)$ from the set $\Lambda(G)$ generated from a weighted mixed graph (Q, A, E). Here *opt* indicates the number of optimal schedules.

Algorithm *RAD_*$\widehat{\varrho}_s(p)$

Input: A weighted mixed graph $G(p) = (Q, A, E)$ with a vector $p \in R^q_+$ of processing times.
Output: The set $\widehat{\mathcal{R}}$ of the stability radii for all optimal digraphs.

Step 1: Construct the set of all feasible digraphs
$\Lambda(G) = \{G_1(p), G_2(p), \dots, G_{opt}(p), \dots, G_\lambda(p)\}$ generated from a weighted mixed graph $G(p) = (Q, A, E)$ and numbered in non-decreasing order of their makespans: $l^p_1 = l^p_2 = \dots = l^p_{opt} < l^p_{opt+1} \le l^p_{opt+2} \le \dots \le l^p_\lambda$;
set $\widehat{\mathcal{R}} = \emptyset$;

IF *opt* $= 1$ **THEN** $s = 1$ **GOTO** *Step 4*;

Step 2: **FOR** $s = 1$ **TO** *opt* **DO**

begin

Step 3: **IF** there exists a path $\mu^* \in H_s(p)$ such that for some digraph $G_k(p), k \neq s, k \le opt$ (i.e. $l^p_s = l^p_k$), there does not exist any path $\nu^* \in H_k(p)$ with $[\mu^*] \subseteq [\nu^*]$;
THEN set $\widehat{\varrho}_s(p) = 0$ and $\widehat{\mathcal{R}} := \widehat{\mathcal{R}} \cup \{\widehat{\varrho}_s(p)\}$;
IF $s < opt$ **THEN GOTO** *Step 2* **ELSE stop**;
ELSE

Step 4: set $\widehat{\varrho}_s(p) := \infty$;
IF the conditions of Theorem 1.3 hold for the digraph $G_s(p)$
THEN $\widehat{\mathcal{R}} := \widehat{\mathcal{R}} \cup \{\widehat{\varrho}_s(p)\}$;
IF $s < opt$ **THEN GOTO** *Step 2* **ELSE stop**;
ELSE

Step 5: **FOR** $k = 1, k \neq s$ **TO** λ **DO**

begin

Step 6: construct the set $H_{sk} = \{\mu \in H_s :$ There is no path $\nu \in H_k$ such that $[\mu] \subseteq [\nu]\}$;
IF $H_{sk} = \emptyset$

Step 7: **IF** $k = \lambda$
IF $H_{st} = \emptyset$ for each digraph $G_t(p), t \neq s, t \in \{1, 2, \dots, \lambda\}$
THEN $\widehat{\mathcal{R}} := \widehat{\mathcal{R}} \cup \{\widehat{\varrho}_s(p)\}$;
IF $s < opt$ **THEN GOTO** *Step 2* **ELSE stop**;
ELSE

ELSE GOTO *Step 5*;
ELSE
set $r_k = 0$;
Step 8: **FOR** each $\mu \in H_{sk}$ **DO**

begin
Step 9: **FOR** each $\nu \in H_k : \ l^p(\nu) \geq l^p_s$ **DO**

begin
Step 10: set $r_\beta = 0$;
construct a sequence $(p^{\nu\mu}_{(0)}, p^{\nu\mu}_{(1)}, \dots, p^{\nu\mu}_{(w_{\nu\mu})})$, where $p^{\nu\mu}_{(0)} = 0$ and $(p^{\nu\mu}_{(1)}, p^{\nu\mu}_{(2)}, \dots, p^{\nu\mu}_{(w_{\nu\mu})})$ is a non-decreasing sequence of the processing times of the operations from the set $[\nu]\backslash[\mu]$ with $w_{\nu\mu} = |[\nu]\backslash[\mu]|$;

FOR $\beta = 0$ **TO** $\omega_{\nu\mu}$ **DO**
begin
$r_\beta = \max\left\{r_\beta, \frac{l^p(\nu) - l^p(\mu) - \sum_{\alpha=0}^{\beta} p^{\nu\mu}_{(\alpha)}}{|[\mu]+[\nu]| - \beta}\right\}$
end
end
set $r_k := \max\{r_k, r_\beta\}$;
end
Step 11: set $\widehat{\varrho}_s(p) := \min\{\widehat{\varrho}_s(p), r_k\}$;
Step 12: **FOR** $k := k+1$ **TO** $\lambda + 1$ **Do**
begin
IF $\widehat{\varrho}_s(p) > \frac{l^p_k - l^p_s}{q}$ **THEN GOTO** *Step 5*
end

set $\widehat{\mathcal{R}} := \widehat{\mathcal{R}} \cup \{\widehat{\varrho}_s(p)\}$;
IF $s < opt$ **THEN GOTO** *Step 2* **ELSE stop**
end
end stop.

At the worst, the calculation of $\widehat{\varrho}_s(p)$ (in just the same way, as the calculation of $\overline{\varrho}_s(p)$) implies not only to have an optimal digraph $G_s(p)$, which already is an NP-hard problem, but to construct all feasible digraphs $G_1(p), G_2(p), \dots, G_\lambda(p)$ (see Step 1) and for each of them, which has to be compared with the optimal digraph, Algorithm $RAD\_\widehat{\varrho}_s(p)$ finds all dominant paths (see Step 6) introduced in Definition 1.2. We can avoid such a time-consuming comparison in the two following cases.

First, if there are two or more optimal digraphs it is possible that the stability radius of one of them or the radii of both are equal to zero (see Theorem 1.1). In Step 3, we check such a condition for $\widehat{\varrho}_s(p) = 0$. Second, there are two cases of an infinitely large stability radius $\widehat{\varrho}_s(p) = \infty$. One of them follows from the graph construction: There is identified a problem class whose optimal solutions are implied only by the given structural input data and even independently from the numerical input data (see Theorem 1.3). Thus, the necessary and sufficient conditions of Theorem 1.3 for an infinitely large stability radius $\mathcal{J}//\mathcal{C}_{max}$ can be verified in polynomial time $O(q^2)$ in Step 4 (q is the number of operations, $q = |Q|$). The second condition for an infinitely large stability radius follows directly from Theorem 1.2 and it is checked in Step 7. More exactly, from Theorem 1.2 and the definition of the set H_{sk} it follows that, if $H_{sk} = \emptyset$ for each feasible digraph $G_k(p), k \neq s$, then $\widehat{\varrho}_s(p) = \infty$.

From Step 7 to Step 11, we calculate the value $\widehat{r}_{ks}$ according to formula (1.10). In Algorithm $RAD\_\widehat{\varrho}_s(p)$, the value $\widehat{r}_{ks} := \widehat{\varrho}_s(p) := \min\{\widehat{\varrho}_s(p), r_k\}$ is finally defined in Step 11. To restrict the number of digraphs G_k with which an optimal digraph has to be compared, Algorithm $RAD\_\widehat{\varrho}_s(p)$ uses the bounds (2.34) in Step 12.

Using Algorithm $RAD\_\widehat{\varrho}_s(p)$, we construct a set $\widehat{\mathcal{R}} = \{\widehat{\varrho}_1(p), \widehat{\varrho}_2(p), \ldots, \widehat{\varrho}_{opt}(p)\}$ of the relative stability radii. As it follows from Remark 2.4, this algorithm is more effective than Algorithm $SOL\_\mathcal{C}_{max}(2)$. So, if $\widehat{\mathcal{R}}$ is not a single-element set, then a decision-maker can use one of the optimal digraphs $G_s(p), s = 1, 2, \ldots, opt$, which is more stable, i.e. a schedule with the largest value of the stability radius $\widehat{\varrho}_1(p) \in \widehat{\mathcal{R}}$.

Next, we present the formal algorithm for the calculation of the stability radii $\overline{\varrho}_s(p)$, which uses the formulas (1.14) and (1.15) derived for the job shop problem.

Algorithm $RAD\_\overline{\varrho}_s(p)$

Input: A weighted mixed graph $G(p) = (Q, A, E)$ with a vector $p \in R^q_+$ of processing times.

Output: The set $\overline{\mathcal{R}}$ of the stability radii for all optimal digraphs.

Step 1: Construct the set of all feasible digraphs $\Lambda(G) = \{G_1(p), G_2(p), \ldots, G_{opt}(p), \ldots, G_\lambda(p)\}$ generated from a weighted mixed graph (Q, A, E) and numbered in non-decreasing order of the mean flow time objective function values: $L^p_1 = L^p_2 = \ldots = L^p_{opt} < L^p_{opt+1} \leq L^p_{opt+2} \leq \ldots \leq L^p_\lambda$;
set $\overline{\mathcal{R}} = \emptyset$.

IF $opt = 1$ **THEN** set $s = 1$ **GOTO** *Step 4*;

Step 2: **FOR** $s = 1$ **TO** opt **DO**

begin

Step 3: **IF** there exists a set $\Omega^{v^*}_s \in \Omega_s(p)$ such that for any set $\Omega^{u^*}_k \in \Omega_k(p)$, there exists an operation $O_{ij} \in Q$ such that condition $n_{ij}(\Omega^{v^*}_s) \geq n_{ij}(\Omega^{u^*}_k)$ (or condition $n_{ij}(\Omega^{v^*}_s) \leq n_{ij}(\Omega^{u^*}_k)$) holds and this inequality has the sign $>$ (or $<$) for at least one set $\Omega^{u^0}_k \in \Omega_k(p)$
THEN set $\overline{\varrho}_s(p) = 0$ and $\overline{\mathcal{R}} := \overline{\mathcal{R}} \cup \{\overline{\varrho}_s(p)\}$;
IF $s < opt$ **THEN GOTO** *Step 2*; **ELSE stop**;
ELSE

Step 4: set $\overline{\varrho}_s(p) := \infty$;

Step 5: **FOR** $k = 1, k \neq s$ **TO** λ **DO**

begin

Step 6: construct the set $\Omega_{sk} = \{\Omega^v_s$: There does not exist a set Ω^u_k such

that $n_{ij}(\Omega_s^v) \leq n_{ij}(\Omega_k^u)$ for each $i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, n_i\}$;
set $r_k = 0$;

Step 7: **FOR** $v = 1$ **TO** ω_s **DO**

begin

Step 8: **FOR** each $\Omega_k^u \in \Omega_k, u = 1, 2, \ldots, \omega_k$, with $\sum_{\nu \in \Omega_k^u} l^p(\nu) \geq L_s^p$ **DO**

begin

Step 9: set $r_\beta = 0$;
order the set of operations Q in the following way:
$O_{ij_{(1)}}, O_{ij_{(2)}}, \ldots, O_{ij_{(m)}}, O_{ij_{(m+1)}}, \ldots, O_{ij_{(q)}}$, where for all $\alpha = 1, 2, \ldots, m$ inequality $n_{ij_{(\alpha)}}(\Omega_k^u) \leq n_{ij_{(\alpha)}}(\Omega_s^v)$ holds and for each $\alpha \in \{m+1, m+2, \ldots, q\}$ the inequalities $n_{ij_\alpha}(\Omega_k^u) > n_{ij_\alpha}(\Omega_s^v)$ and $p_{ij_{(m+1)}} \geq p_{ij_{(m+2)}} \geq \ldots \geq p_{ij_{(q)}}$ have to be satisfied;
FOR $\beta = 0$ **TO** $q - m$ **DO**

begin

$$r_\beta = \max\left\{r_\beta, \frac{\sum_{\alpha=1}^{m+\beta} p_{ij_{(\alpha)}}(n_{ij_{(\alpha)}}(\Omega_k^u) - n_{ij_{(\alpha)}}(\Omega_s^v))}{\sum_{\alpha=1}^{m+\beta} |n_{ij_{(\alpha)}}(\Omega_k^u) - n_{ij_{(\alpha)}}(\Omega_s^v)|}\right\}$$

end

end

set $r_k := \max\{r_k, r_\beta\}$;

end

Step 10: set $\overline{\varrho}_s(p) := \min\{\overline{\varrho}_s(p), r_k\}$;

Step 11: **FOR** $k := k + 1$ **TO** $\lambda + 1$ **DO**

begin

IF $\overline{\varrho}_s(p) > \frac{L_k^p - L_s^p}{nq - n}$ **THEN GOTO** *Step 5*

end

$\overline{\mathcal{R}} := \overline{\mathcal{R}} \cup \{\overline{\varrho}_s(p)\}$;
IF $s < opt$ **THEN GOTO** *Step 2*; **ELSE stop**

end

end stop.

If there exist at least two optimal schedules, i.e. if *opt* > 1, we verify in Step 3 the condition for a zero stability radius on the basis of Theorem 1.6. In Step 4, we set $\overline{\varrho}_s(p) := \infty$ (note that $\overline{\varrho}_s(p) < \infty$ due to Theorem 1.7 and Remark 1.1). Theorem 1.5 is used for the calculation of the stability radius $\overline{\varrho}_s(p)$, $0 < \overline{\varrho}_s(p) < \infty$, for each optimal digraph $G_s, s = 1, 2, \ldots, opt$, (see Steps 6 - 10). In Step 11, we can reduce the set of digraphs in our considerations due to Lemma 3.4.

Both above formal algorithms were coded in Fortran-77. So, for a small problem size the program starts with generating all feasible digraphs and for each of them, which has to be compared with the optimal digraph, it finds dominant paths (see Definition 1.2). Then formulas (1.9) and (1.10) from Section 1.2 are used for calculating $\widehat{\varrho}_s(p)$ and formulas (1.14) and (1.15) from Section 1.4 are used for calculating $\overline{\varrho}_s(p)$. To restrict the number of digraphs G_k with which an optimal digraph has to be compared, we use the bound

(2.34) (see Chapter 2) for the makespan criterion and the bound (3.26) (see Chapter 3) for the mean flow time criterion.

Note that the software developed is rather general. In principle, it allows to calculate the exact or approximate values of $\widehat{\varrho}_s(p)$ and $\overline{\varrho}_s(p)$ for most scheduling problems (since there exists a possibility to represent them as extremal problems on a mixed graph, see Section 1.1). The only 'theoretical' requirement for such problems is the prohibition of preemptions of operations (see Assumption 2). However, in the simulation study we are forced to take into account also 'practical' requirements: The running time and the memory of the computers. Remind that the most critical parameter of the problem under consideration is the number of edges in the mixed graph G because the whole number of feasible (without a circuit) and infeasible (with circuit) digraphs generated by G is equal to $2^{|E|}$. Moreover, for each feasible digraph G_k, we have to find all dominant paths for C_{max} and (what is essentially larger) all subsets of the set of dominant paths for $\sum C_i$.

4.2 Experimental Design and Results

In this section, computations were restricted to job shop problems. We considered three different levels of the simulation study in dependence on running time and memory limits. The stability region of the optimal digraph G_s (the whole set of non-negative q-dimensional vectors, for which G_s is optimal) is a closed cone [TSS94, p. 326]. Indeed, if G_s is optimal for the vector $p \in R^q_+$ of the processing times, it remains optimal for the processing times $\alpha p_{11}, \alpha p_{12}, \ldots, \alpha p_{nn_n}$ with any real $\alpha > 0$ (obviously, the stability radius is the largest radius of a stability ball, which is fully contained in the stability region). So, when considering the influence of 'load leveling' factors (numbers and distributions of operations per machines and per jobs) to the stability radius, we consider the same range of variations of the processing times for the problems of the first level: The processing times of the operations are uniformly distributed real numbers (with four digits after the decimal point) between the same bounds 10 and 100.

First, we generated small instances with 12 operations in each case, for which the exact values of the stability radii $\widehat{\varrho}_s(p)$ and $\overline{\varrho}_s(p)$ may be calculated on a PC 386 usually within some seconds using only internal memory of the computer. For each combination of the number of jobs from 3 to 7 and of the number of machines from 4 to 8, we randomly generated and solved 50 instances. Moreover, at the first level simulation includes four different types of problems in dependence on the distribution of the number of operations to the machines (evenly or randomly) and the operations, distributed to the same machine, to the jobs (evenly or randomly). Thus, we consider at the first level problems of the four types

EE (evenly, evenly),
ER (evenly, randomly),
RE (randomly, evenly) and
RR (randomly, randomly).

At the first level, we calculated the stability radii for 5000 job shop problems ($4 \cdot 5 \cdot 5 \cdot 50 =$ 4 (types) $\cdot$ 5 (combinations of the number of jobs n) $\cdot$ 5 (combinations of the number of machines m) $\cdot$ 50 (randomly generated instances in each series)) with 12 operations in each instance. Note that, if there were two or more optimal schedules for a sample problem, we calculated the stability radius for each of them.

After solving the above problems (without using external memory on a hard disk), we considered series of instances for each combination of the number of jobs from 8 to 10 and of the number of machines from 4 to 8, and for each combination of the number of jobs from 3 to 10 and $m = 3$. The number of operations in each instance was equal to 12. Since the number of edges in the mixed graph exceeded 20 (and so the number of generated feasible and infeasible digraphs exceeded $2^{20} = 1,048,576$), we had to use external memory on a hard disk for such instances and the running time for some of them achieved one or even two hours on a PC 486. So, we were forced to restrict the number of considered instances in the most difficult series for such combinations of the numbers of jobs and the numbers of machines to 10.

On the basis of the obtained information within the first level of experiments (for the instances with 12 operations), we designed the second and third ones. First, we decided to consider only instances generated for an evenly distributed number of operations to the machines and evenly distributed operations on the same machine to the jobs (i.e. type EE). At the second level, we calculated the exact values of the stability radii for job shop problems with 16 and 20 operations, considering 10 instances in each series while considering the influence of 'load leveling' factors. Note that for some of the instances at the second level, the CPU time of a Pentium PC exceeded 10 hours.

Along with 'load leveling' factors, other ones also influence the complexity and stability of scheduling problems, e.g. the variability of $p_{ij}, O_{ij} \in Q$, across the entire shop and the variability of the average processing time from job to job or from machine to machine are also important factors of the complexity of shop scheduling problems (remind the famous job shop problem with 10 jobs and 10 machines given in [FT63], which was so difficult to attain due to a special processing time variability). Therefore, at the second level we also investigated the influence of the latter factors for random modifications of the processing times of the job shop problem with the same mixed graph G. More precisely, for the same randomly generated mixed graph G (see Chapter 1) at the second level of the simulation study, we considered six different ranges of variations of the given processing times, namely: [1, 10], [1, 100], [1, 1000], [10, 100], [10, 1000] and [100, 1000]. Obviously, intervals [10, 100] and [100, 1000] may be obtained from the interval [1, 10] after multiplying with 10 and 100, respectively. However, the number of optimal schedules, and the number of problems with a zero value of stability radii may be different for these three intervals, since we consider all real numbers with fixed number of decimal places. Due to the same reason, we consider the intervals [1, 100] and [10, 1000]. For the above segments, we calculated $\widehat{\varrho}_s(p)$ and $\overline{\varrho}_s(p)$ for each optimal schedule s in series with 50 instances. Moreover, we investigated instances in which different jobs had different ranges of variations of the given processing times. At the third level, we considered a well-known job shop test problem with 6 jobs and 6 machines from [FT63] with different ranges of variations of the given processing times across the entire shop and across different jobs.

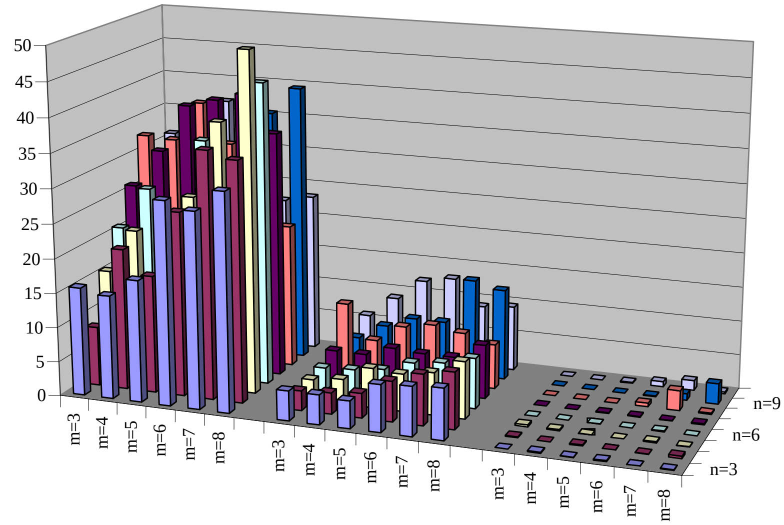
In Figures 4.1 and 4.2, we present the maximal, average and minimal values of the stability radii for each combination of the number of jobs n and the number of machines m, considered at the first level when the number of operations are evenly distributed to the machines and the operations on a machine are evenly distributed to the jobs (type EE). While the processing times are real numbers between 10 and 100, the stability radii are approximately between 0.001 and 50 for $\mathcal{C}_{max}$ and between 0.001 and 35 for $\sum \mathcal{C}_i$. Similar data for the other three types of distributing the operations are given in Figures 4.3 and 4.4 (types ER, RE and RR). The largest value of $\widehat{\varrho}_s(p)$ was about 90, and the largest value of $\overline{\varrho}_s(p)$ was about 70. For all types EE, ER, RE and RR, the average value of $\widehat{\varrho}_s(p)$ was larger than that of $\overline{\varrho}_s(p)$. An obvious conclusion from these diagrams is that an optimal makespan schedule (Figures 4.1 and 4.3) is more stable than an optimal mean flow time schedule (Figures 4.3 and 4.4). An important issue from Figures 4.1 - 4.4 is also that for each series of instances the smallest value of $\widehat{\varrho}_s(p)$ and $\overline{\varrho}_s(p)$ is greater than zero.

Table 4.1: *Randomly generated problems*

n x m	RADIUS / p_{MAX}			RADIUS / p_{AVE}			$\frac{100\gamma}{\lambda}$	NOS		DIFF	
	MIN	AVE	MAX	MIN	AVE	MAX		AVE	MAX	AVE	MAX
1	2	3	4	5	6	7	8	9	10	11	12
Maximum flow time											
6 x 6	0.01	0.62	4.26	0.02	1.09	7.40	-	21.50	78	1.60	3.83
7 x 7	0.07	1.76	11.16	0.12	3.45	23.59	-	15.60	43	2.71	9.72
8 x 8	0.07	3.43	12.66	0.13	6.00	17.80	-	17.00	70	4.91	12.20
9 x 9	0.00	3.97	11.52	0.00	6.91	22.14	4.43	28.90	144	6.07	11.38
10 x 10	0.18	3.33	21.90	0.32	5.97	41.38	1.54	12.40	48	5.68	18.84
Mean flow time											
6 x 6	0.33	1.28	5.10	0.59	2.25	8.67	2.27	1.10	2	0.00	0.00
7 x 7	0.23	1.33	6.57	0.40	2.32	11.19	2.42	1.20	2	0.00	0.00
8 x 8	0.26	1.86	6.54	0.51	3.28	11.85	0.03	1.20	2	0.00	0.00
9 x 9	0.60	2.20	4.41	1.10	3.84	8.22	0.10	1.10	2	0.00	0.00
10 x 10	0.46	3.83	8.05	0.75	6.79	13.69	0.57	1.00	1	0.00	0.00

The results for the sample problems of the second level for 'load leveling' factors are presented in Table 4.1, where the minimal (MIN), average (AVE) and maximal (MAX) values of the stability radius divided by the maximal processing times (p_{MAX}) are given in columns 2, 3, and 4, and similar values divided by the average processing times (p_{AVE}) are given in columns 5, 6, and 7. During our experiments, we also determined the largest number γ of competitive digraphs in the sequence $(G_{i_1}, G_{i_2}, \ldots, G_{i_\gamma}, \ldots, G_{i_{\lambda'}}, \ldots, G_{i_\lambda})$ (where the digraphs are ordered according to non-decreasing objective function values) and the number λ' of the digraph, which was the last considered one in this sequence, while calculating the exact value of the stability radius. Column 8 contains the average values of the percentage of digraphs, which may be a competitive digraph for the optimal one $(100t/\lambda)$. For the set of instances presented in Table 4.1 with the mean flow time criterion, these values are bounded by 2.42%. When minimizing the makespan, these values are larger, but the latter results are mostly due to the large numbers of optimal makespan schedules (the average and maximal numbers NOS of optimal semiactive schedules for an instance are given in columns 9 and 10, respectively). Note that for some 6 x 6 instances (i.e. those with 6 jobs and 6 machines), 7 x 7 instances and 8 x 8 instances, the number λ of all semiactive schedules was not calculated in our experiments, and therefore the values of $100\gamma/\lambda$ are not presented for these series. If there is more than one optimal schedule, we calculate the differences of their stability radii. The average and maximal values of these differences (DIFF) are presented in columns 11 and 12, respectively. We can also note that for the mean flow time criterion, an optimal schedule is usually uniquely determined, and even if there are two optimal mean flow time schedules, they have often the same stability radius. Consequently, for the mean flow time criterion we have not much need to look for an optimal schedule with the largest stability radius.

Next, we present the randomly generated mixed graph G for the job shop problem $\mathcal{J}6/n = 4/\Phi$ with 4 jobs and 6 machines, which is used for the simulation study of

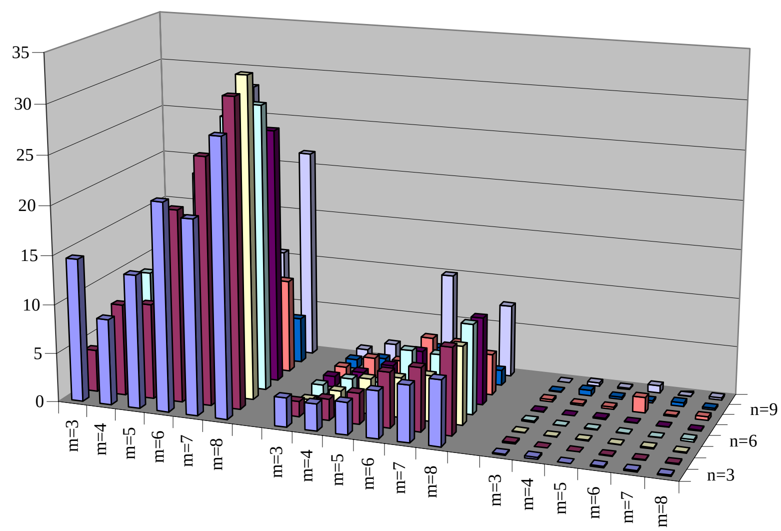


	m=3	m=4	m=5	m=6	m=7	m=8
n=3	15,71	15	17,71	29,48	28,4	31,56
n=4	8,6	20,49	17,04	26,74	35,86	34,85
n=5	15,65	22,04	21,67	27,76	38,84	49,27
n=6	20,96	27,11	21,63	34,9	30,88	43,82
n=7	26,16	31,69	38,68	39,84	40,99	35,63
n=8	32,69	32,4	38,18	32,47	14	20,91
n=9	23,07	22,07	22,33	22,31	36,44	40,39
n=10	31,13	28,14	36,7	28,14	22,31	23,19

	m=3	m=4	m=5	m=6	m=7	m=8
n=3	4,33	4,3	4	6,83	7,12	7,43
n=4	2,84	3,11	3,58	5,86	7,38	8,2
n=5	3,02	3,61	5,72	5,41	6,15	8,27
n=6	3,34	3,58	4,13	5,62	6,13	7,32
n=7	4,47	4,44	5,88	5,57	5,57	7,83
n=8	10,2	5,18	7,71	8,45	7,74	6,53
n=9	3,74	6,01	7,59	7,56	14,25	13,27
n=10	5,84	8,94	12,02	12,83	9,08	9,49

	m=3	m=4	m=5	m=6	m=7	m=8
n=3	0,01	0,14	0,03	0,15	0,01	0,08
n=4	0,14	0,01	0,16	0,03	0,06	0,34
n=5	0,3	0,18	0,01	0	0,18	0,04
n=6	0	0,01	0,03	0	0,12	0
n=7	0	0	0,08	0,11	0,05	0,15
n=8	0	0	0,01	0,53	2,89	0,17
n=9	0	0	0	0,12	0,97	2,98
n=10	0	0	0,14	0,75	1,44	0,29

Figure 4.1: *Maximal, average and minimal values of $\widehat{\varrho}_s(p)$ for the problems of type EE*

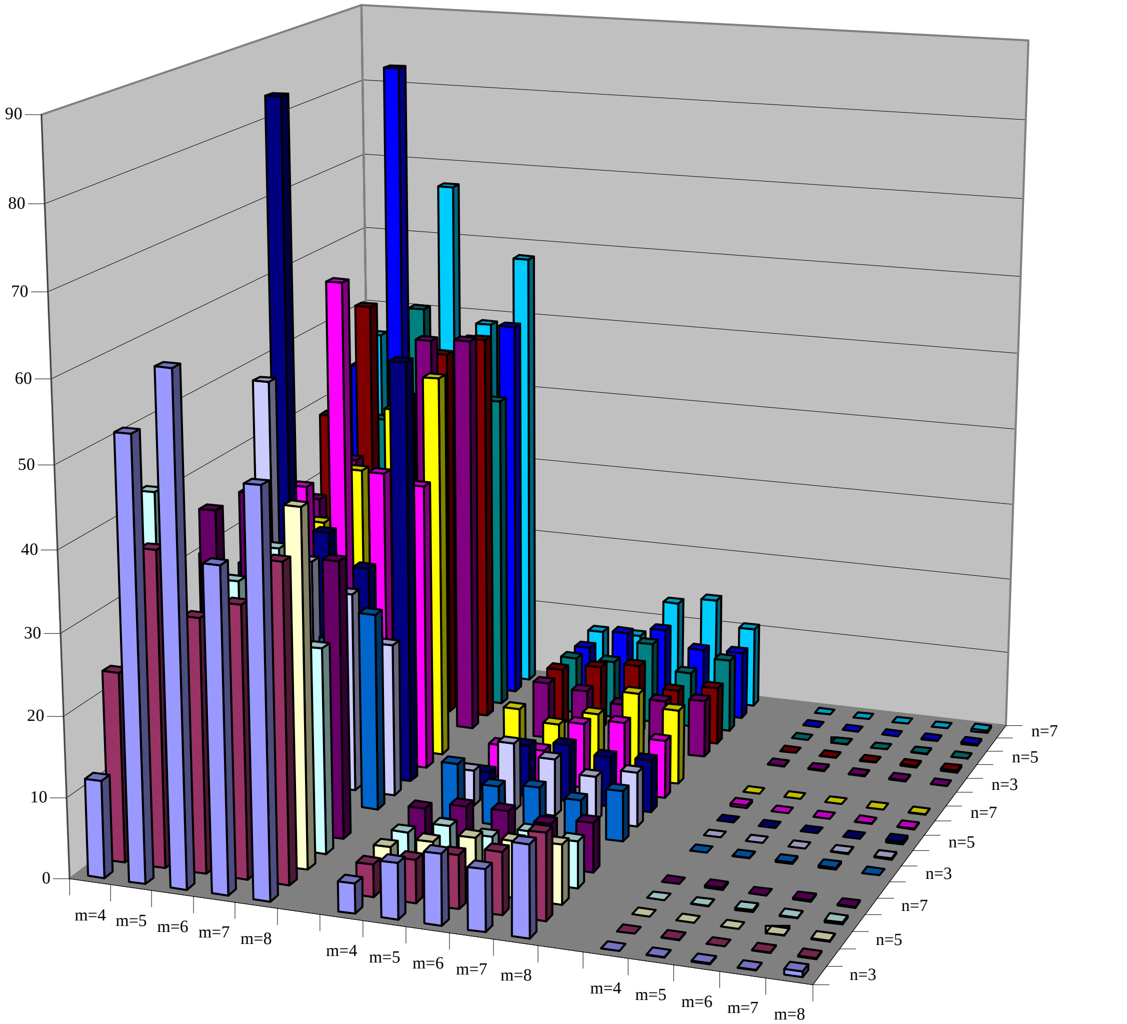


	m=3	m=4	m=5	m=6	m=7	m=8
n=3	14,57	8,76	13,58	21,15	19,77	28,04
n=4	4,3	9,35	9,71	19,61	25,17	31,23
n=5	2,27	5,29	12,33	15,13	16,32	32,7
n=6	3,27	10,93	7,16	21,72	27,74	29,16
n=7	2,68	8,59	16,15	20,28	20,88	25,96
n=8	2,82	5,32	5,57	13,32	18,89	9,57
n=9	2,57	4,13	10,46	6,52	2,97	4,67
n=10	4,27	3,97	2,5	28,14	10,67	21,55

	m=3	m=4	m=5	m=6	m=7	m=8
n=3	2,95	2,73	3,26	4,78	5,74	6,63
n=4	1,55	2,17	3,16	5,62	6,45	8,83
n=5	0,69	1,95	3,59	4,04	4,63	7,95
n=6	1,22	2,2	2,22	5,86	5,77	9,21
n=7	1,11	1,89	2,96	4,8	5,08	8,92
n=8	1,11	2,41	2,47	5,23	5,06	4,19
n=9	0,93	1,4	2,39	3,24	1,53	1,57
n=10	1,05	1,96	1,24	10,01	3,48	7,43

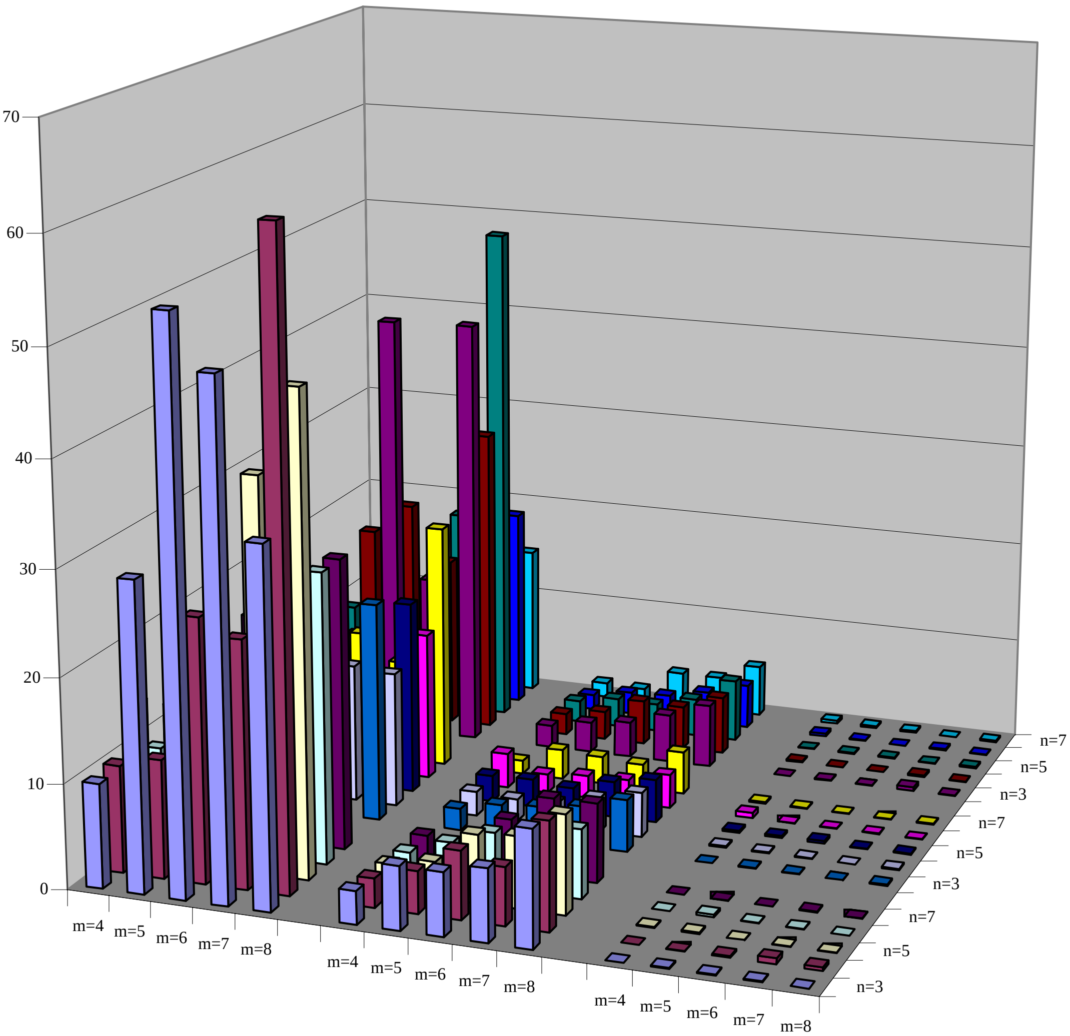
	m=3	m=4	m=5	m=6	m=7	m=8
n=3	0,09	0,2	0	0,13	0,15	0,13
n=4	0,14	0	0,02	0,05	0,07	0,09
n=5	0,07	0,04	0,08	0,04	0,13	0,09
n=6	0,15	0,04	0,09	0,05	0,04	0,23
n=7	0,07	0,01	0,14	0,06	0,03	0,08
n=8	0,22	0,16	0,21	1,58	0,12	0,36
n=9	0,07	0,56	0,2	0,21	0,39	0,19
n=10	0,01	0,32	0,14	0,75	0,03	0,24

Figure 4.2: *Maximal, average and minimal values of $\overline{\varrho}_s(p)$ for the problems of type EE*



	m=4	m=5	m=6	m=7	m=8		m=4	m=5	m=6	m=7	m=8		m=4	m=5	m=6	m=7	m=8
n=3	12,02	54,07	62,11	39,75	49,58		3,67	6,86	8,68	7,55	11,22		0,05	0,07	0,08	0,1	0,66
n=4	23,51	38,96	31,35	33,52	39,21		4,01	5,3	6,58	7,67	10,67		0	0,07	0	0,05	0,12
n=5	20,85	20,68	28,29	29,73	44,23		4,2	5,56	6,76	6,97	7,33		0,02	0,03	0,03	0,01	0,14
n=6	42,53	26,58	32,69	37,25	25,61		4,1	5,58	4,95	6,25	5,8		0	0,03	0,26	0,07	0,25
n=7	41,6	39,35	41,96	34,49	34,72		5,21	6,17	6,25	5,36	6,15		0	0,23	0,01	0,19	0,09
n=3	30,34	29,67	23,96	21,33	24,99		7,19	4,98	5,49	4,57	6,38		0	0,04	0,17	0,23	0,01
n=4	22,34	51,36	29,06	25,46	19,47		4,52	8,68	7,33	5,73	6,89		0	0	0	0,11	0,18
n=5	23,97	85,56	31,31	27,22	53,93		2,45	6,62	7,35	6,46	6,65		0	0	0	0,01	0,31
n=6	19,45	35,31	61,9	38,04	36,88		4,25	4,23	8,35	9,1	7,41		0,34	0	0	0,07	0,12
n=7	25,1	29,22	36,58	44,9	49,43		7,32	5,89	7,9	11,22	9,64		0	0	0	0	0,05
n=3	28,97	34,73	29,5	51,46	51,8		7,35	6,91	5,68	6,8	7,45		0,05	0,15	0,1	0,14	0,02
n=4	38,89	53,89	42,31	48,45	50,81		7,54	8,49	9,29	6,51	7,48		0	0,03	0,08	0,18	0,25
n=5	31,22	37,56	52,86	37,46	41,39		7,47	7,57	10,59	7,23	9,57		0	0,05	0	0,09	0,06
n=6	43,12	83,36	43,61	47,98	50,2		7,29	9,92	10,92	8,76	8,95		0	0	0	0,08	0,23
n=7	46,22	47,29	67,11	48,91	58,15		7,93	7,93	13,04	14	10,63		0	0	0	0	0,3

Figure 4.3: *Maximal, average and minimal values of* $\widehat{\varrho}_s(p)$ *for the problems of types ER, RE and RR*



	m=4	m=5	m=6	m=7	m=8		m=4	m=5	m=6	m=7	m=8		m=4	m=5	m=6	m=7	m=8
n=3	9,98	29,41	53,81	48,64	33,93		3,15	6,06	6,03	6,94	11,11		0,01	0,11	0,09	0,11	0,03
n=4	10,27	11,33	25,18	23,58	61,41		2,8	4,06	6,51	5,53	10,35		0,02	0,03	0,17	0,54	0,32
n=5	14,85	15,13	24,29	37,5	45,89		2,64	3,4	6,52	6,85	9,49		0,08	0,15	0,04	0,01	0,03
n=6	9,18	16,54	19,5	13,04	27,82		2,25	3,74	5,2	4,8	6,63		0,02	0,24	0,03	0,06	0,01
n=7	11	12,42	21,58	36,63	27,88		2,39	1,13	5,01	7,57	7,62		0,02	0,01	0,01	0,11	0,03
n=3	10,67	12,73	14,42	17,59	21,14		2,12	2,97	3,32	3,87	5,06		0	0,08	0,06	0,03	0,13
n=4	6,99	9,11	11,18	13,41	13,09		2,34	2,14	2,32	3,3	4,32		0,15	0,09	0,07	0,01	0,12
n=5	11,21	14,45	8,78	11,67	18,76		2,56	2,67	2,35	3,48	4,18		0,21	0,21	0,27	0,03	0,11
n=6	5,29	5,15	8,77	9,51	14,44		3,4	1,82	2,15	2,25	3,27		0,51	0,02	0,03	0,06	0,04
n=7	3,17	10,25	12,48	9,94	23,97		1,3	2,9	2,7	2,43	4,15		0,14	0,09	0,06	0,04	0,11
n=3	7,54	9,4	42,06	16	42,27		2,21	3	3,49	4,75	6,2		0,01	0,07	0,05	0,38	0,08
n=4	9,47	19,1	22,12	16,73	30,19		2,11	2,8	4,39	4,27	5,71		0,12	0,01	0,01	0,17	0,07
n=5	9,45	9,11	11,83	20,48	49,67		2,15	2,83	2,74	3,75	6,13		0,06	0,1	0,11	0,05	0,17
n=6	4,4	8,5	7,52	14,2	19,77		1,49	2,2	2,37	3,18	4,39		0,26	0,07	0	0,13	0,07
n=7	3,86	2,87	9,56	14,75	14,68		1,46	1,31	3,45	3,45	5,07		0,3	0,21	0,17	0,03	0,16

Figure 4.4: *Maximal, average and minimal values of $\overline{\varrho}_s(p)$ for the problems of types ER, RE and RR*

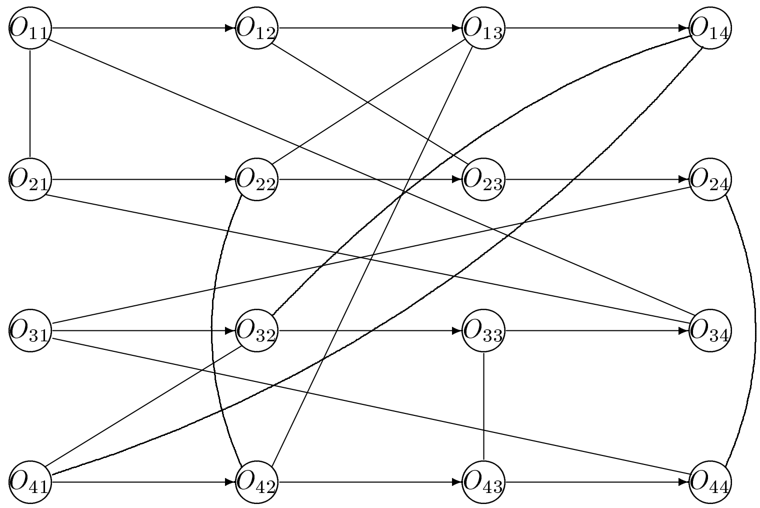


Figure 4.5: *Randomly generated mixed graph for problem $\mathcal{J}6/n=4/\Phi$*

the influence of the variability of the processing times: $Q = \{O_{11}, O_{12}, \ldots, O_{44}\}$; $E = \{[O_{11}, O_{21}], [O_{11}, O_{34}], [O_{21}, O_{34}];\ [O_{12}, O_{23}];\ [O_{13}, O_{22}],\ [O_{13}, O_{42}], [O_{22}, O_{42}];\ [O_{14}, O_{32}], [O_{14}, O_{41}], [O_{32}, O_{41}];\ [O_{24}, O_{31}],\ [O_{24}, O_{44}], [O_{31}, O_{44}];\ [O_{33}, O_{43}]\}$ (see Figure 4.5). Computational results for this mixed graph are given in Table 4.2 and Table 4.3. Table 4.2 presents the computational results for different ranges of the processing times for the same mixed graph G, which is described above. Note also that both criteria $\mathcal{C}_{max}$ and $\sum \mathcal{C}_i$ are considered for the same 50 examples for which the obtained results are presented in the corresponding rows of Table 4.2 (row i for $\mathcal{C}_{max}$ corresponds to row $i+6$ for $\sum \mathcal{C}_i$).

Table 4.2: *Problem $\mathcal{J}6/n=4/\Phi$, $\Phi \in \{\mathcal{C}_{max}, \sum \mathcal{C}_i\}$, with different ranges of variations of p_{ij}*

Bounds for p_{ij}		RADIUS/p_{MAX}			RADIUS/p_{AVE}			$\frac{100\gamma}{\lambda}$	$\frac{100\lambda'}{\lambda}$	NOS		NMO	DIFF	
LB	UB	MIN	AVE	MAX	MIN	AVE	MAX			AVE	MAX		AVE	MAX
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Maximum flow time														
1	10	0.07	2.37	8.48	0.13	4.40	17.15	0.48	2.93	2.80	12	34	0.15	0.75
1	100	0.02	2.31	12.11	0.04	4.91	27.38	0.71	5.02	3.58	21	34	2.35	10.53
1	1000	0.00	3.63	13.83	0.00	8.33	36.71	0.73	8.66	4.26	36	32	30.93	132.28
10	100	0.13	2.52	10.78	0.26	4.64	21.18	0.50	4.71	2.74	12	31	2.36	9.53
10	1000	0.01	3.08	13.06	0.01	6.23	31.36	0.62	8.07	4.60	30	41	26.90	123.57
100	1000	0.04	2.88	11.89	0.07	5.20	24.85	0.53	4.18	2.30	12	25	20.63	108.90
Mean flow time														
1	10	0.06	2.56	10.17	0.11	4.76	17.89	0.30	3.87	1.02	2	1	0.13	0.13
1	100	0.07	2.47	9.90	0.13	5.06	20.54	0.34	4.44	1.02	2	1	0.00	0.00
1	1000	0.03	2.07	10.21	0.05	3.95	16.96	0.30	3.40	1.04	2	2	1.80	3.59
10	100	0.12	2.41	7.30	0.21	4.51	15.63	0.30	3.42	1.00	1	0	-	-
10	1000	0.17	2.26	8.67	0.37	4.52	16.29	0.32	3.40	1.12	2	6	6.18	22.48
100	1000	0.05	2.53	11.43	0.11	4.48	19.78	0.33	4.22	1.04	2	2	16.37	32.75

Table 4.3 presents the computational results for different ranges of the values p_{ij} of the operations of different jobs. Along with the columns defined for Table 4.1, we also present

the percentage of considered digraphs while calculating the exact value of the stability radius (column 10 in Table 4.2 and column 16 in Table 4.3) and the number NMO of problems with two or more optimal schedules (column 13 in Table 4.2 and column 19 in Table 4.3).

For the problems considered at the second level, the 'superiority' of the stability radius for the makespan criterion is lost in most cases. At least the minimal values of $\overline{\varrho}_s(p)$ became larger than those of $\widehat{\varrho}_s(p)$. Of course, the large number of optimal makespan schedules has influenced this relation essentially, but even on average, we cannot find a large superiority of the stability radius of one criterion over the other (for the considered classes of randomly generated job shop problems).

Next, we discuss some questions on the basis of our experimental calculation of the stability radii of the optimal schedules for small randomly generated job shop problems.

How often is the stability radius equal to zero? In the experiments at the first and the second levels, we obtained only once a stability radius equal to zero for criterion $\mathcal{C}_{max}$ and never for criterion $\sum \mathcal{C}_i$ although it takes not much effort to construct such an example by hand (see Theorem 1.1 for $\mathcal{C}_{max}$ and Theorem 1.6 for $\sum \mathcal{C}_i$). So, in principle, to find an optimal schedule for almost all problems generated in our experiments has sense. On the other hand, in many series there are instances with very small values of the stability radius (even less than 0.001). So, if for such an instance the precision of the processing times is not sufficiently high, we have no guarantee that the (a priori) constructed optimal schedule will be indeed the best one in its practical realization.

May the stability radius be infinitely large? From theoretical results it follows that for any given n and m, there exist job shop problems with an optimal makespan schedule s, which remains optimal for any feasible variation of the processing times, i.e. $\widehat{\varrho}_s(p) = \infty$ (see Theorem 1.2 and Theorem 1.3). In particular, an easily verifiable characterization of such a schedule has been derived for criterion $\mathcal{C}_{max}$ (see Theorem 1.3). In contrast, it was shown that for mean flow time, we have $\overline{\varrho}_s(p) \leq \max_{O_{ij} \in Q}\{p_{ij}\}$ for any job shop problem (see Theorem 1.7). Although in [KSW95] a practical example of an infinitely large stability radius was presented (for a traffic-light on the intersection of two roads), nevertheless such a shop appears to be rather artificial for large numbers of jobs and machines. Surprisingly, in our randomly generated job shop problems with the makespan criterion an infinitely large stability radius was obtained not seldom, at least essentially more often than a zero stability radius (of course, we did not include infinite stability radii while calculating the average and maximal values of $\widehat{\varrho}_s(p)$). So, our experiments indicate that the results derived in [KSW95, Sot91] will have not only theoretical significance.

How much 'best' schedules do we need to consider? As already mentioned, we also determined the number γ of competitive digraphs and the number λ' of considered digraphs, while calculating the exact value of the stability radius. For the problems of the first level, the diagrams for the percentage of the numbers γ and for the percentage of the numbers λ' for the problems of type EE are presented in Figure 4.6 (Figure 4.7) for criterion $\mathcal{C}_{max}$ (for criterion $\sum \mathcal{C}_i$, respectively). In the front part of the diagrams in Figures 4.6 and 4.7, the minimal, average and maximal values of the percentages $100\gamma/\lambda$ are presented, while in the background of these diagrams the minimal, average and maximal values of the percentages $100\lambda'/\lambda$ are presented. As it follows from Figures 4.6 and 4.7, the value $100\gamma/\lambda$ may be smaller than 1 % and it is not greater than 73 % for $\mathcal{C}_{max}$ and not greater than 56 % for $\sum \mathcal{C}_i$.

Table 4.3: *Problem $\mathcal{J}6/n=4/\Phi$, $\Phi \in \{\mathcal{C}_{max}, \sum \mathcal{C}_i\}$, with variability of p_{ij} from job to job*

Lower bound (LB_i) and upper bound (UB_i) for the processing times of job J_i								RADIUS/p_{MAX}			RADIUS/p_{AVE}			$\frac{100\gamma}{\lambda}$	$\frac{100\lambda'}{\lambda}$	NOS		NMO	DIFF	
LB_1	UB_1	LB_2	UB_2	LB_3	UB_3	LB_4	UB_4	MIN	AVE	MAX	MIN	AVE	MAX			AVE	MAX		AVE	MAX
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Maximum flow time																				
10.	20.	30.	40.	70.	80.	90.	100.	0.01	0.87	3.81	0.01	1.59	6.92	3.83	9.15	29.90	54	50	0.48	3.24
10.	20.	40.	50.	60.	70.	90.	100.	0.00	0.98	4.37	0.01	1.79	8.06	1.10	5.86	7.64	16	50	0.57	3.54
10.	30.	30.	50.	60.	80.	80.	100.	0.01	1.41	6.62	0.02	2.56	12.01	1.70	7.95	15.48	32	50	1.28	5.05
10.	40.	30.	60.	50.	80.	10.	100.	0.00	1.66	8.56	0.01	2.99	15.59	1.21	6.97	7.68	22	50	1.40	6.92
10.	60.	30.	60.	50.	80.	50.	100.	0.06	1.84	8.03	0.09	3.32	15.27	0.84	4.25	5.54	24	49	1.00	7.08
10.	60.	10.	60.	50.	100.	50.	100.	0.00	3.30	12.05	0.01	6.02	22.56	1.30	12.52	11.98	24	50	3.77	10.57
10.	40.	40.	70.	10.	100.	10.	100.	0.01	4.28	14.81	0.01	8.31	31.23	0.65	18.38	15.92	58	50	6.26	14.01
10.	30.	10.	50.	60.	80.	60.	100.	0.01	1.78	9.82	0.02	3.20	18.20	1.24	6.81	10.62	32	50	1.58	9.31
10.	20.	30.	50.	60.	80.	80.	100.	0.00	1.03	4.69	0.00	1.87	8.85	2.36	9.54	19.38	54	50	1.10	4.13
10.	25.	25.	75.	25.	75.	50.	100.	0.09	2.73	14.34	0.16	5.30	25.56	0.81	7.93	3.62	12	42	2.03	14.07
Mean flow time																				
10.	20.	30.	40.	70.	80.	90.	100.	0.08	1.08	2.97	0.14	1.95	5.36	0.27	1.56	1.00	1	0	-	-
10.	20.	40.	50.	60.	70.	90.	100.	0.04	1.21	2.82	0.07	2.21	5.22	0.26	1.28	1.00	1	0	-	-
10.	30.	30.	50.	60.	80.	80.	100.	0.02	1.15	4.11	0.04	2.08	7.33	0.31	1.64	1.00	1	0	-	-
10.	40.	30.	60.	50.	80.	70.	100.	0.09	1.54	4.85	0.15	2.78	8.02	0.31	1.78	1.00	1	0	-	-
10.	60.	30.	60.	50.	80.	50.	100.	0.05	1.87	6.14	0.09	3.35	10.85	0.35	2.39	1.00	1	0	-	-
10.	60.	10.	60.	50.	100.	50.	100.	0.01	1.45	4.41	0.01	2.62	7.53	0.32	2.02	1.04	3	1	0.00	0.00
10.	40.	40.	70.	70.	100.	10.	100.	0.07	2.32	7.83	0.12	4.23	13.62	0.33	3.40	1.04	2	2	0.00	0.00
10.	30.	10.	50.	60.	80.	60.	100.	0.04	1.31	4.66	0.08	2.34	7.97	0.33	1.51	1.00	1	0	-	-
10.	20.	30.	50.	60.	80.	80.	100.	0.01	0.94	3.26	0.01	1.70	5.95	0.25	1.14	1.00	1	0	-	-
10.	25.	25.	75.	25.	75.	50.	100.	1.01	2.18	5.51	0.02	4.17	9.92	0.31	3.27	1.00	1	0	-	-

It should be noted that for the case of a large number of machines and a small number of operations (at the first level when $q = 12$), there often exist only a few feasible semi-active schedules which make the relative values of γ and λ' rather large. Moreover, for criterion $\mathcal{C}_{max}$, we have a relatively large number of optimal schedules which also enlarges the relative values of γ and λ'. Thus, calculating the exact value of the stability radius on the basis of bounds (2.34) and (3.26) may require to consider the whole set $\Lambda(G)$ of digraphs for the problems considered at the first level of our simulation study.

From Tables 4.2 and 4.3, it follows that the competitive digraphs are within 3.83 % of the whole set of feasible digraphs for criterion $\mathcal{C}_{max}$ and within 0.35 % for criterion $\sum \mathcal{C}_i$, and the percentage of digraphs which have been considered while calculating the stability radius is no more than 18.38 % for criterion $\mathcal{C}_{max}$ and no more than 4.44 % for criterion $\sum \mathcal{C}_i$. So, it is not necessary to construct the whole set of feasible digraphs for calculating the stability radius of an optimal digraph for these types of problems.

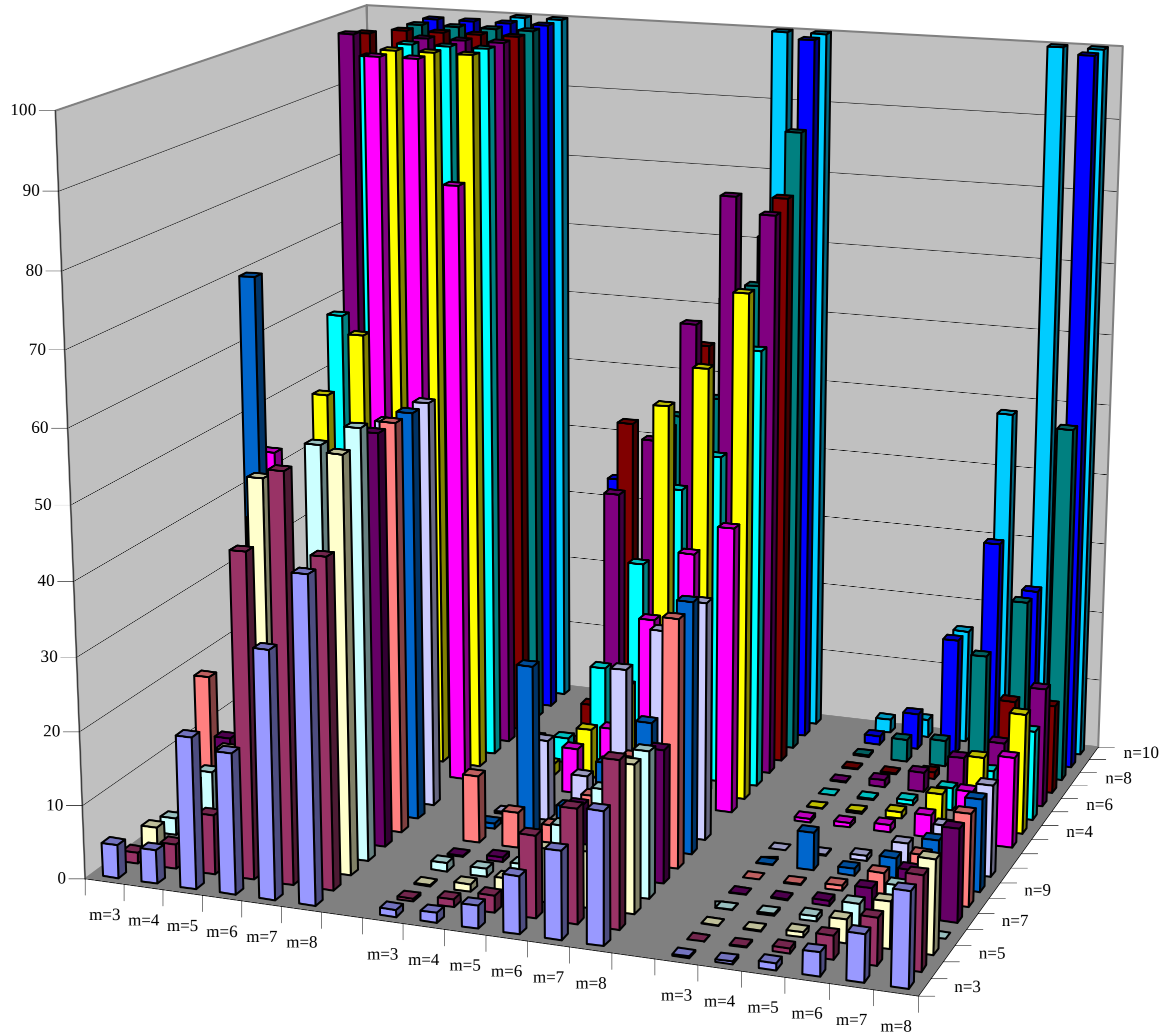
After studying the obtained results at the first and second levels of our experiments, we enlarged the size of problems, which are still suitable for calculating the exact value of the stability radius (or at least its upper bound). For calculating the stability radius for instances of larger size, we constructed for each of them only the k best schedules (with $k = 100$ in most cases) by a direct enumeration of the whole set of feasible digraphs. Then, considering only these k best digraphs, we intended to calculate the stability radius of an optimal digraph (or optimal digraphs). If this process has stopped before the whole k best digraphs were compared with the optimal one, we have obtained the exact value of the stability radius due to the bounds (2.34) or (3.26), otherwise we have obtained at least an upper bound for the stability radius. Moreover, to shorten the running time we used the branch-and-bound method for calculating the k best digraphs.

How can one combine this approach with the branch-and-bound method? The following approach to stability analysis for scheduling problems seems to be practically efficient. Using a branch-and-bound method (e.g. [BJS94, CP89]), one can construct not only one optimal but the k best schedules. In particular, in our computational study we used a branch-and-bound algorithm with the conflict resolution strategy. Due to an implicit enumeration of the feasible mixed graphs $G_{(s)}(p) = (Q, A_{(s)}, E_{(s)})$, we construct the k best ones and calculate the exact value or an upper bound for the stability radius of an optimal schedule in the same manner as described in the above paragraph "How much 'best' schedules do we need to consider?". Note that, while an explicit enumeration of the digraphs $G_1(p), G_2(p), \dots, G_\lambda(p)$ gives the exact value of $\widehat{\varrho}_s(p)$ for $|E| \leq 30$, the branch-and-bound algorithm gives the possibility to calculate $\widehat{\varrho}_s(p)$ for $|E| \leq 100$ (often within the same CPU time).

In particular, at the third level of the experiments we considered the well-known classical job shop problem from [FT63] with 6 jobs and 6 machines. The assignment of the operations $Q = \{O_{11}, O_{12}, \dots, O_{66}\}$ to the set of machines $M = \{M_1, M_2, \dots, M_6\}$ is as follows:

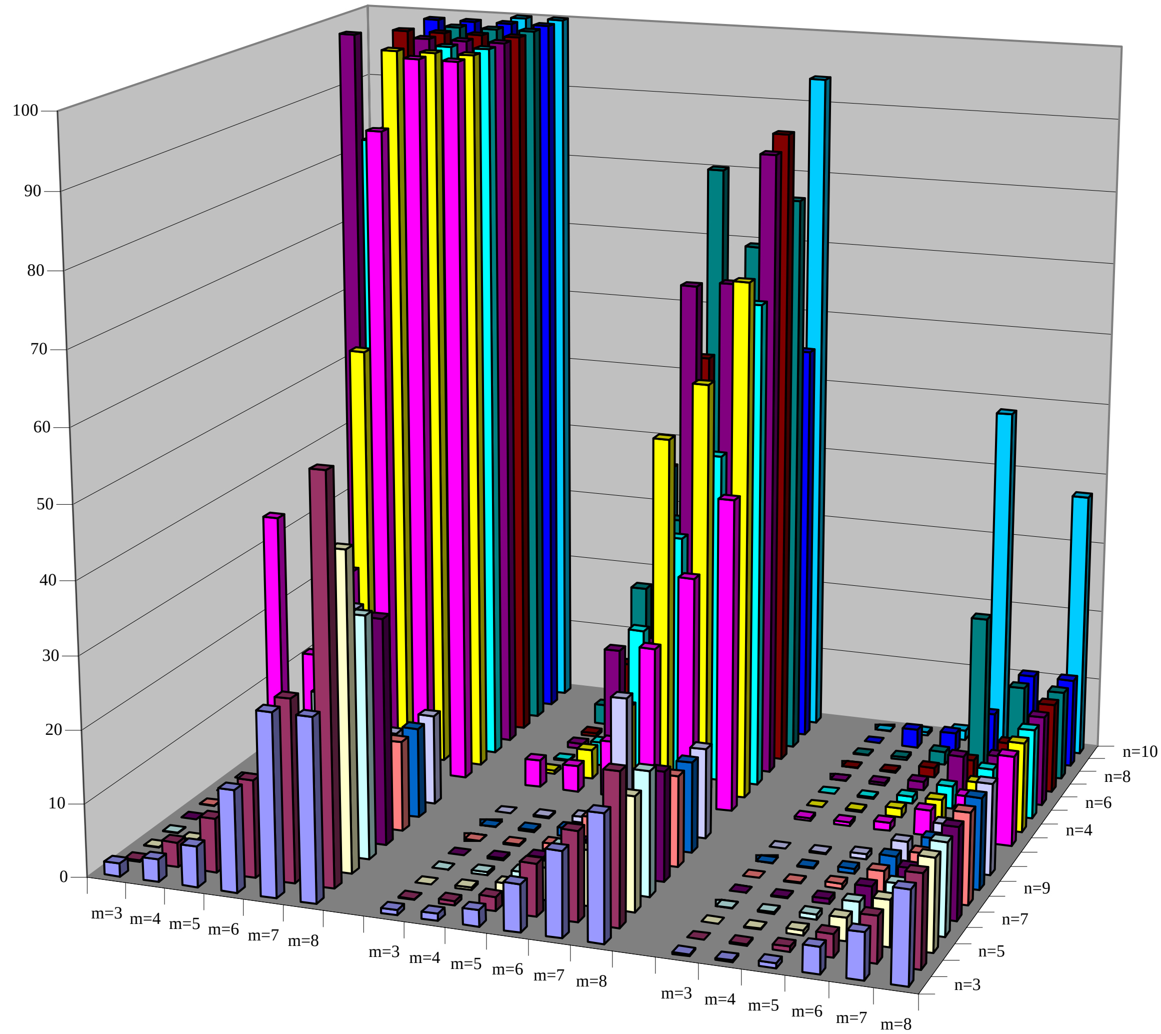
$$
\begin{aligned}
Q_1 &= \{O_{12}, O_{25}, O_{34}, O_{42}, O_{55}, O_{64}\},\\
Q_2 &= \{O_{13}, O_{21}, O_{35}, O_{41}, O_{52}, O_{61}\},\\
Q_3 &= \{O_{11}, O_{22}, O_{31}, O_{43}, O_{51}, O_{66}\},\\
Q_4 &= \{O_{14}, O_{26}, O_{32}, O_{44}, O_{56}, O_{62}\},\\
Q_5 &= \{O_{16}, O_{23}, O_{36}, O_{45}, O_{53}, O_{65}\},\\
Q_6 &= \{O_{15}, O_{24}, O_{33}, O_{46}, O_{54}, O_{63}\}.
\end{aligned}
$$

For this problem, each job has to be processed on each machine exactly once and hence we have $q = 6 \text{ x } 6 = 36$ and $|E| = 6 \text{ x } \binom{6}{2} = 90$. By the branch-and-bound algorithm



	m=3	m=4	m=5	m=6	m=7	m=8		m=3	m=4	m=5	m=6	m=7	m=8		m=3	m=4	m=5	m=6	m=7	m=8
n=3	4,55	4,55	20,5	19,05	33,33	43,75		0,96	1,33	3,05	7,71	11,76	17,66		0,23	0,35	0,93	3,13	6,25	12,5
n=4	1,5	3,36	8,05	44	55	44,44		0,39	1,03	2,33	11,05	15,33	22,4		0,03	0,15	0,69	3,13	6,25	12,5
n=5	3	9,9	15,03	52,08	15,63	56,25		0,16	1,03	2,49	7,24	7,57	19,9		0,04	0,15	0,69	3,13	6,25	12,5
n=6	2,3	9,5	11,11	27,78	55,56	58,33		1,15	1,11	2,41	8,41	13,97	19,78		0,03	0,25	0,69	3,13	6,25	0
n=7	2,63	12,35	43,21	35,94	40,63	56,25		0,13	0,64	2,67	9,21	11,15	18,1		0,03	0,15	0,69	3,13	6,25	12,5
n=8	18,56	9,76	14,24	27,08	31,25	56,25		9,28	4,88	3,78	8,56	14,6	33,92		0,03	0,15	0,69	3,13	6,25	12,5
n=9	6,83	72,73	17,01	26,56	53,13	56,25		0,71	23,31	4,42	11,23	17,47	34,55		0,02	5,17	0,93	3,13	6,25	12,5
n=10	2,8	22,48	25,35	26,56	53,13	56,25		0,3	11,16	6,85	22,41	28,36	32,81		0,02	0,15	0,69	3,13	6,25	12,5
n=3	44,16	27,27	37,76	100	100	83,33		7,26	6,35	9,88	25,95	35,74	39,85		0,49	0,65	1,04	3,13	7,14	12,5
n=4	8,8	51,47	60,37	100	100	100		1,67	7,28	14,22	54,26	59,87	70,53		0,09	0,28	0,93	4,17	10	16,67
n=5	19	61,52	98,04	100	100	100		3,58	14,5	30,07	41,17	46,32	61,5		0,04	0,22	0,69	3,13	6,25	12,5
n=6	3,24	100	94,44	100	100	100		1,19	38	46,19	63,03	81,06	78,83		0,11	1,01	2,78	5,56	8,33	16,67
n=7	24,99	99,31	100	100	100	100		4,91	46,78	46,55	58,75	62,13	80,13		0,14	0,15	0,93	3,13	12,5	12,5
n=8	22,71	67,28	100	100	100	100		7,22	24,74	46,92	49,95	66,46	88,19		0,24	3,19	3,7	16,67	25	50
n=9	87,09	72,73	100	100	100	100		35,3	23,31	47,5	62,97	72,19	100		1,24	5,17	16,67	31,25	25	100
n=10	23,47	77,78	66,67	50	100	100		11,48	24,83	45,44	49,69	100	100		1,94	2,55	16,32	48,44	100	100

Figure 4.6: *Percentage of the number γ (and λ') of competitive (considered) digraphs for the problems of type EE for $\widehat{\varrho}_s(p)$*



	m=3	m=4	m=5	m=6	m=7	m=8		m=3	m=4	m=5	m=6	m=7	m=8		m=3	m=4	m=5	m=6	m=7	m=8
n=3	1,77	3,07	5,6	13,89	25	25		0,68	0,96	2,21	6,41	11,53	17,19		0,23	0,25	0,69	3,57	6,25	12,5
n=4	0,23	3,38	7,41	13,33	25	55,56		0,11	0,59	1,9	7,1	12,19	20,75		0,03	0,23	0,69	3,13	6,25	12,5
n=5	0,08	1,79	8,73	10,94	15,63	43,75		0,05	0,37	1,55	4,43	7,57	15,5		0,02	0,15	0,69	3,13	6,25	12,5
n=6	0,1	0,88	2,47	11,11	22,22	33,33		0,04	0,41	1,11	5,43	9,73	16,96		0,02	0,25	0,69	3,13	6,25	12,5
n=7	0,03	0,82	2,31	9,38	15,63	31,25		0,02	0,28	1,18	3,92	7,31	15		0,02	0,15	0,69	3,13	6,25	12,5
n=8	0,08	0,31	1,74	12,5	12,5	12,5		0,04	0,22	0,97	5,42	7,08	12,5		0,03	0,15	0,69	3,13	6,25	12,5
n=9	0,06	0,63	2,78	4,69	6,25	12,5		0,04	0,28	1,12	3,29	6,25	12,5		0,02	0,15	0,69	3,13	6,25	12,5
n=10	0,04	0,51	1,04	26,56	9,38	12,5		0,02	0,24	0,82	18,26	6,56	12,5		0,02	0,15	0,69	3,13	6,25	12,5
n=3	34,63	15,34	28	90	100	100		3,82	3,68	7,81	21,73	32,21	43,66		0,36	0,51	1,14	3,57	6,25	12,5
n=4	2,16	22,27	58,02	100	100	100		0,49	4,12	11,01	49,57	57,61	72		0,03	0,23	1,22	3,13	6,25	12,5
n=5	0,84	15,24	86,51	93,75	100	100		0,19	3,32	20,35	34,19	46,32	67,75		0,05	0,21	0,98	3,13	6,25	12,5
n=6	2,51	100	87,04	100	100	100		0,71	15,18	16,84	68,29	69,01	87,08		0,04	0,38	1,23	5,56	6,25	12,5
n=7	0,95	64,75	100	100	100	100		0,48	11,44	27,42	57	62	88,88		0,04	0,21	1,39	3,13	6,25	12,5
n=8	9,57	58,13	74,54	100	100	100		2,87	21,09	31,69	82,29	71,88	78,75		0,11	0,39	1,85	21,88	12,5	12,5
n=9	8,51	81,19	100	100	100	100		2,29	18,45	31,34	58,44	59,06	56,25		0,05	2,65	2,78	6,25	12,5	12,5
n=10	9,99	72,61	54,86	50	100	100		2,99	36,02	29,87	49,69	80,63	93,75		0,02	0,46	1,39	48,44	6,25	37,5

Figure 4.7: *Percentage of the number γ (and λ') of competitive (considered) digraphs for the problems of type EE for $\overline{\varrho}_s(p)$*

we constructed $k = 150$ best schedules: 22 of them are optimal with $\mathcal{C}_{max} = 55$ and 54 other schedules have a makespan value equal to 56, and at least 74 schedules have a makespan value equal to 57. We calculated an upper bound for $\widehat{\varrho}_s(p)$ for each optimal makespan schedule. It turned out that 14 of them have a zero stability radius and the other 8 optimal schedules have an upper bound for $\widehat{\varrho}_s(p)$ equal to 0.08333. The existence of unstable optimal schedules for this test problem is implied mainly by the fact that its processing times are integers from 1 to 10.

We also randomly generated 50 instances with 6 jobs, 6 machines and 36 operations. Again each job has to be processed on each machine exactly once (i.e. we considered classical job shop problems), but in contrast to the problem from [FT63], the processing times were uniformly distributed real numbers between 1 and 10. For each generated problem with 36 operations, we constructed 50 best schedules (for the makespan criterion) on the basis of the branch-and-bound algorithm and calculated upper bounds for $\widehat{\varrho}_s(p)$ for each optimal makespan schedule which was constructed. Note that 45 of these instances had more than one optimal makespan schedule and among them, 7 instances had 50 or even more optimal makespan schedules. The average value of the stability radius $\widehat{\varrho}_s(p)$ was equal to 0.12939, and for all calculated optimal makespan schedules s the following bounds were satisfied: $0.001 \leq \widehat{\varrho}_s(p) \leq 0.87455$. We calculated also the differences between $\widehat{\varrho}_s(p)$ for different optimal makespan schedules $s \in S$ of the same instance (if this instance had two or more optimal makespan schedules). The maximum of this difference was equal to 0.84636, the average difference was 0.11709 and some optimal makespan schedules had the same stability radius. Among the 50 instances, there was no optimal schedule with a zero stability radius.

Table 4.4: *Test problem $\mathcal{J}6/n\!=\!6/\mathcal{C}_{max}$ with variability of p_{ij}*

Bounds for p_{ij}		RADIUS/p_{MAX}			RADIUS/p_{AVE}			NOS		NMO	DIFF	
LB	UB	MIN	AVE	MAX	MIN	AVE	MAX	AVE	MAX		AVE	MAX
1	2	3	4	5	6	7	8	9	10	11	12	13
Common bounds for p_{ij} for different jobs												
1.	10.	0.0067	0.1843	0.8744	0.0138	0.3374	1.5340	13.5	52	8	0.0106	0.0393
1.	100.	0.0077	0.3265	1.2092	0.0163	0.6158	2.0630	26.7*	100*	10	0.1705	1.1278
1.	1000.	0.0749	0.6679	2.4344	0.1639	1.4461	5.1843	31.5	90	10	6.3276	22.6764
10.	100.	0.0440	0.7733	3.9813	0.0820	1.4277	7.4289	17.5*	100*	9	0.5540	3.6507
10.	1000.	0.0070	0.4436	1.7260	0.0116	0.8298	3.2587	27.4*	100*	10	3.5290	10.9608
100.	1000.	0.0308	0.5694	1.9779	0.0564	1.0109	3.8182	17.0	54	10	7.2888	18.3886
Different bounds for p_{ij} for different jobs												
LB_i^1	UB_i^1	0.0000	0.6429	3.9997	0.0000	1.1009	7.2044	41.2*	100*	10	0.9175	3.1636
LB_i^2	UB_i^2	0.0216	0.5046	1.3379	0.0383	0.8487	2.3764	5.2	12	9	0.0000	0.0000
LB_i^3	UB_i^3	0.0000	1.0051	4.2719	0.0000	1.7247	7.4870	74.6*	100*	10	1.7654	4.1433
LB_i^4	UB_i^4	0.0031	0.9716	9.2608	0.0053	1.7292	16.4208	82.2*	100*	10	2.1819	8.6476

To investigate the influence of the variability of the processing times p_{ij} on the stability radius, we considered again the test problem with 6 jobs and 6 machines given in [FT63], but with different distributions of the processing times to the operations. More precisely, the mixed graph $G = (Q, A, E)$ was defined in accordance with [FT63], but the processing times were randomly generated real numbers with the same lower and upper bounds for all jobs (see rows 1 - 6 in Table 4.4) and with different lower and upper bounds for different jobs (rows 7 - 10 in Table 4.4). Each row in Table 4.4 presents the results obtained for a series of 10 instances. For each instance, we calculated the stability radius using 100 best schedules generated by the branch-and-bound algorithm.

For row 7 in Table 4.4, the lower bound LB_i^1 and the upper bound UB_i^1 for job J_i are as follows:

$$\begin{array}{ll} LB_1^1 = 10, & UB_1^1 = 40; \\ LB_2^1 = 20, & UB_2^1 = 50; \\ LB_3^1 = 30, & UB_3^1 = 60; \\ LB_4^1 = 50, & UB_4^1 = 80; \\ LB_5^1 = 60, & UB_5^1 = 90; \\ LB_6^1 = 70, & UB_6^1 = 100. \end{array}$$

For row 8 these bounds are:

$$\begin{array}{ll} LB_1^2 = 10, & UB_1^2 = 60; \\ LB_2^2 = 30, & UB_2^2 = 60; \\ LB_3^2 = 40, & UB_3^2 = 60; \\ LB_4^2 = 50, & UB_4^2 = 70; \\ LB_5^2 = 50, & UB_5^2 = 80; \\ LB_6^2 = 50, & UB_6^2 = 100. \end{array}$$

For row 9 these bounds are:

$$\begin{array}{ll} LB_1^3 = 10, & UB_1^3 = 40; \\ LB_2^3 = 20, & UB_2^3 = 50; \\ LB_3^3 = 40, & UB_3^3 = 70; \\ LB_4^3 = 60, & UB_4^3 = 90; \\ LB_5^3 = 70, & UB_5^3 = 100; \\ LB_6^3 = 10, & UB_6^3 = 100. \end{array}$$

For row 10 these bounds are:

$$\begin{array}{ll} LB_1^4 = 10, & UB_1^4 = 30; \\ LB_2^4 = 20, & UB_2^4 = 40; \\ LB_3^4 = 30, & UB_3^4 = 50; \\ LB_4^4 = 60, & UB_4^4 = 80; \\ LB_5^4 = 70, & UB_5^4 = 90; \\ LB_6^4 = 80, & UB_6^4 = 100. \end{array}$$

In Table 4.4, we marked the series of instances, for which the number of optimal schedules is larger than 100 by an asterisk. Since we calculated only 100 best schedules for each instance, we had not the exact number of optimal semiactive makespan schedules.

Unfortunately, the developed software did not allow us to find $\overline{\varrho}_s(p)$ for most of the above instances with 36 operations and 90 edges since the calculation of the stability radius for the mean flow time criterion is essentially more time-consuming than for the makespan.

How to use this approach for problems of practical size? For large instances, for which a direct enumeration of all feasible digraphs was practically impossible, we constructed only a subset of feasible digraphs, selected then the best digraph G_s among them and calculated an upper bound for the 'stability radius' of G_s by a comparison with all other digraphs that have been constructed. This variant of the implementation of the software may be useful for some practical problems. Indeed, in reality OR workers have at most one or only a few feasible schedules (usually without an exact information about their quality). In the case when a set of feasible schedules is known, we can investigate the stability radius of the best of them in comparison with the others at hand.

Even if we have not the possibility to find an optimal schedule by a branch-and-bound method and only an approximate schedule (with information about its quality) or a heuristic schedule has been constructed, we can investigate the 'stability radius' of this schedule in comparison with the other $k-1$ schedules that have been constructed.

4.3 Remarks on the Stability Radii

The main issue from our experiments is that an optimal schedule is usually stable: Its stability radius is not equal to zero and so there exists a ball with the center p of the processing times in the space R^q_+ of input data, within which the schedule remains optimal. Thus, such a radius may be useful as a measure of the stability of an optimal schedule. Moreover, on the basis of the above computational experiments (though limited problem sizes), one can make the conclusion that an optimal schedule for criterion $\mathcal{C}_{max}$ is often more stable than an optimal schedule for criterion $\sum \mathcal{C}_i$ when the size of the problem is small.

Moreover, our approach gives not only the exact value or a bound for the stability radius but also competitive schedules (competitive digraphs) which along with an optimal schedule have to be considered as candidates for the practical realization, when the stability radius or its upper bound is less than the possible error of the processing times known in advance.

Note that the problem of calculating the stability radius of the digraph $G_s(p)$ is NP-hard even provided that an optimal schedule s is known. It is even NP-hard to find the 'tolerances' of a single processing time p_{ij}, which do not violate the optimality of the optimal digraph. The latter result follows from [RC95] since the problem considered in that paper may be presented as a special case of the job shop problem.

Another insight is that an optimal mean flow time schedule is usually uniquely determined, while two or more optimal makespan schedules are very usual (at least in our simulation study). So, in the latter case it makes sense to look for an optimal makespan schedule with the largest value of the stability radius (the difference of the stability radii for different optimal schedules of the same problem may be very large for the makespan criterion). Such a schedule has a better chance to be makespan optimal in its practical realization. However, this is not valid for the mean flow time criterion, for which one can be satisfied by the first constructed optimal schedule because even if there are two or more optimal mean flow time schedules, they usually have the same value (or close values) of the stability radii.

Moreover, there exist shops for which we can look for an optimal makespan schedule with an infinitely large stability radius. In particular, if one can influence the properties of the shop (i.e. technological routes of the jobs, the number of used machines and the distribution of the operations to the machines, etc.), one can design a shop that has an optimal makespan schedule with an infinitely large stability radius (see Theorem 1.3). In this case the variations of the processing times have no influence on such a schedule to be optimal. For some scheduling problems, such a property may be practically important.

Since a zero stability radius of the optimal schedule s is rather seldom, there exists an $\epsilon > 0$ such that s will remain optimal for any variations $p_{ij} \pm \epsilon$ of the processing times. In particular, this is true for almost all problems with the mean flow time criterion, which were considered in our experiments, since for these problems an optimal schedule is often uniquely determined, and as a result, it has a strictly positive stability radius. On the other hand, it has practical sense to make the error in the determination of the processing times as small as possible in order to guarantee the real optimality of a schedule at hand: Almost in all series there were schedules with very small (but strictly positive) values of the stability radii.

After the analysis of the influence of possible changes of the given processing times of the operations, i.e. the largest quantity of independent variations (stability radius) within which an optimal schedule of problems $\mathcal{J}//\mathcal{C}_{max}$ and $\mathcal{J}//\sum \mathcal{C}_i$ remains optimal, we make experimental investigations of job shop problems with *uncertain* processing times $\mathcal{J}/a_i \le p_i \le b_i/\mathcal{C}_{max}$ and $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$, which satisfy Assumptions 1, 2 and 3 (see Introduction). These computational results are described in Section 4.4.

4.4 Problems $\mathcal{J}/a_i \le p_i \le b_i/\Phi$, $\Phi \in \{\mathcal{C}_{max}, \Sigma\, \mathcal{C}_i\}$

The algorithms derived in Section 3.4 were coded in Fortran-77 and were tested on a PC 486 (120 MHz) for the exact solution and on a PC 486 (50 MHz) for the heuristic solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ and on a PC 486 (133 MHz) for the exact and the heuristic solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\mathcal{C}_{max}$. Here the term 'exact solution' is used for indicating a set $\Lambda^*(G)$ which satisfies Definition 3.1 in contrast to the 'heuristic solution' indicating a set $\Lambda \subset \Lambda(G)$ which generally may not contain an optimal schedule for each vector $p \in T$.

The experimental design was as follows. First, we considered series of instances of problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$ with small n and m for which an exact solution and the exact minimal solution may be calculated within one hour on a PC 486 (120 MHz). After finding upper bounds for such n and m, we started experiments with medium size problems in order to find at least their heuristic solution. From this moment, the experiments were continued simultaneously on both computers in order to find upper bounds on n and m for a 'good' heuristic solution on a PC 486 (50 MHz) (see Table 4.9), and to increase the problem size for the exact solution on a PC 486 (120 MHz) (see Table 4.7).

Table 4.5: *Types of problems considered in the experiments*

	Types of problems	Errors of the processing times	Types of problems	
Exact solutions:	A	5%, 10%, 15%, 20%		
sets $B, \Lambda^*(G)$	B	2%, 6%, 8%, 10%	B	Heuristic
and $\Lambda^T(G)$	C	1%, 3%, 5%, 7%	C	solution:
		1%, 2%, 3%, 4%	D	set B
		0.1%, 0.2%, 0.3%, 0.4%	E	

We tested the algorithms for the makespan criterion from Section 3.4 with all corresponding changes for criterion $\mathcal{C}_{max}$ for the same randomly generated test problems.

Table 4.6: *The minimal lower and maximal upper bounds of processing times*

Errors	Lower bound	Upper bound	The actual processing time p^*_{ij}
20%	$(1-0.2)p_{ij}$	$(1+0.2)p_{ij}$	$0.8p_{ij} \leq p^*_{ij} \leq 1.2p_{ij}$
15%	$(1-0.15)p_{ij}$	$(1+0.15)p_{ij}$	$0.85p_{ij} \leq p^*_{ij} \leq 1.15p_{ij}$
10%	$(1-0.1)p_{ij}$	$(1+0.1)p_{ij}$	$0.9p_{ij} \leq p^*_{ij} \leq 1.1p_{ij}$
8%	$(1-0.08)p_{ij}$	$(1+0.08)p_{ij}$	$0.92p_{ij} \leq p^*_{ij} \leq 1.08p_{ij}$
7%	$(1-0.07)p_{ij}$	$(1+0.07)p_{ij}$	$0.93p_{ij} \leq p^*_{ij} \leq 1.07p_{ij}$
6%	$(1-0.06)p_{ij}$	$(1+0.06)p_{ij}$	$0.94p_{ij} \leq p^*_{ij} \leq 1.06p_{ij}$
5%	$(1-0.05)p_{ij}$	$(1+0.05)p_{ij}$	$0.95p_{ij} \leq p^*_{ij} \leq 1.05p_{ij}$
4%	$(1-0.04)p_{ij}$	$(1+0.04)p_{ij}$	$0.96p_{ij} \leq p^*_{ij} \leq 1.04p_{ij}$
3%	$(1-0.03)p_{ij}$	$(1+0.03)p_{ij}$	$0.97p_{ij} \leq p^*_{ij} \leq 1.03p_{ij}$
2%	$(1-0.02)p_{ij}$	$(1+0.02)p_{ij}$	$0.98p_{ij} \leq p^*_{ij} \leq 1.02p_{ij}$
1%	$(1-0.01)p_{ij}$	$(1+0.01)p_{ij}$	$0.99p_{ij} \leq p^*_{ij} \leq 1.01p_{ij}$
0.4%	$(1-0.004)p_{ij}$	$(1+0.004)p_{ij}$	$0.996p_{ij} \leq p^*_{ij} \leq 1.004p_{ij}$
0.3%	$(1-0.003)p_{ij}$	$(1+0.003)p_{ij}$	$0.997p_{ij} \leq p^*_{ij} \leq 1.003p_{ij}$
0.2%	$(1-0.002)p_{ij}$	$(1+0.002)p_{ij}$	$0.998p_{ij} \leq p^*_{ij} \leq 1.002p_{ij}$
0.1%	$(1-0.001)p_{ij}$	$(1+0.001)p_{ij}$	$0.999p_{ij} \leq p^*_{ij} \leq 1.001p_{ij}$

Heuristic solutions of problem $\mathcal{J}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ are presented in Table 4.10 and exact solutions are given in Table 4.8.

For criterion $\Phi = \sum \mathcal{C}_i$ (Tables 4.8 and 4.10 for criterion $\Phi = \mathcal{C}_{max}$), both Tables 4.7 and 4.9 present computational results for classical job shop problems only (see Table 1.1 at page 17). So, each randomly generated instance $\mathcal{J}m/n{=}k, a_i \leq p_i \leq b_i/\Phi$ has $|Q| = mn$ operations and the corresponding mixed graph G has $(m-1)n$ arcs and $\binom{n}{2}m$ edges (note that the latter parameter has the most influence on the running times of our algorithms). For more than 700 classical job shop problems with different combinations of $n \leq 10$ and $m \leq 8$, we calculated the average number of all feasible schedules λ, the average cardinality $|B|$ of the set B, the average cardinality $|\Lambda^*(G)|$ of the set $\Lambda^*(G)$, and the average cardinality $|\Lambda^T(G)|$ of the set $\Lambda^T(G)$ for both criteria $\sum \mathcal{C}_i$ and $\mathcal{C}_{max}$.

For each combination of n and m under consideration, three types of series (called A, B and C) of instances were considered for the case of an exact solution (see Table 4.7 and Table 4.8). Each series consists of 10 instances with randomly generated technological routes. The expected processing times, which form the *input vector* p, are real numbers uniformly distributed in the segment $[10, 100]$. In each instance of types A, B and C, all operations are partitioned into four approximately equal parts with different maximal errors of the processing times (see Table 4.5). For an instance of type C, these errors are 1%, 3%, 5% and 7%, for an instance of type B, errors are 2%, 6%, 8% and 10%, and for an instance of type A, errors are 5%, 10%, 15% and 20% (see Table 4.5). In particular, the operations of the fourth part of an instance of type A have the most uncertain processing times: If the *input* (expected) processing time is supposed to be equal to p_{ij}, then the lower bound for the actual processing time is equal to $(1-0.2)p_{ij}$ and the upper bound is equal to $(1+0.2)p_{ij}$ (see Table 4.6). On the other hand, the operations of the first part of an instance of type C have the processing times with the smallest error: If the *input* processing time is supposed to be equal to p_{ij}, then the lower

Table 4.7: *Exact solutions of randomly generated problems* $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$

$n \times m$ type λ	λ' λ^* λ^T	CPU time for Scheme I	CPU time for Scheme II	CPU time for Scheme III	λ' λ^* λ^T	CPU time for Scheme IV	λ' λ^* λ^T	CPU time for Scheme V
1	2	3	4	5	6	7	8	9
3×3	5.2	.6	.7	.6	4.8	.4	4.8	.2
C	1.9	1.0	1.1	1.0	1.9	.8	1.9	.6
91.2	1.9	1.1	1.3	1.1	1.9	.9	1.9	.7
3×3	9.3	.7	.8	.5	6.1	.2	6.6	.2
B	2.9	1.2	1.3	1.0	2.7	.6	2.7	.6
90.7	2.6	1.4	1.5	1.2	2.4	.9	2.4	.9
3×3	16.5	.7	.8	.9	10.6	.3	11.3	.3
A	3.3	1.5	1.7	1.8	2.9	1.0	3.0	1.0
77.4	3.3	1.7	1.9	2.0	2.9	1.4	3.0	1.4
3×4	3.4	1.9	1.6	1.1	3.0	.8	3.0	.3
C	1.7	2.2	1.9	1.8	1.5	1.0	1.5	.6
261.9	1.7	2.3	2.0	1.5	1.5	1.1	1.5	.6
3×4	15.1	2.4	2.6	2.5	9.5	.9	10.0	.7
B	3.0	5.9	5.9	5.9	2.5	3.2	2.5	3.0
300.8	2.7	6.0	6.0	6.1	2.4	3.5	2.4	3.3
3×4	32.5	2.7	3.2	3.4	15.2	.8	16.8	.9
A	5.1	12.0	12.1	12.3	3.8	5.6	3.9	6.4
276.8	4.1	14.1	14.1	14.3	3.1	7.5	3.2	8.3
3×5	4.7	8.3	6.3	5.8	4.1	1.4	4.2	1.0
C	1.5	8.6	6.6	6.2	1.5	1.8	1.5	1.4
604.8	1.4	8.7	6.8	6.3	1.5	1.9	1.5	1.4
3×5	12.9	10.5	11.6	10.6	9.5	2.2	10.2	1.1
B	3.4	12.8	13.7	12.8	3.3	3.8	3.3	2.9
894.1	3.1	13.0	14.0	13.0	3.0	4.3	3.0	3.3
3×5	77.6	12.9	20.8	21.3	30.5	2.4	33.0	2.8
A	11.9	63.4	68.9	70.1	8.7	19.4	9.0	27.1
896.7	10.8	96.5	100.6	102.2	7.7	40.5	8.0	52.8
3×6	9.3	48.8	51.4	34.2	7.1	2.5	7.3	1.5
C	3.5	49.2	52.0	34.8	2.8	2.9	2.8	2.0
1555.9	2.7	49.5	52.8	35.6	2.1	3.2	2.1	2.3
3×6	21.0	49.1	66.5	58.9	13.9	3.6	14.1	2.3
B	4.6	71.9	88.2	80.6	4.5	19.5	4.5	18.3
1760.9	4.2	77.2	93.2	85.7	4.1	24.7	4.1	23.6
3×6	65.3	48.3	119.2	139.5	19.2	2.4	23.4	2.7
A	7.6	198.3	262.0	282.7	4.8	51.8	6.0	54.1
1559.0	7.0	476.0	526.2	548.3	4.5	111.5	5.6	289.4
3×7	5.4	307.6	343.0	310.0	4.1	5.4	4.4	1.6
C	1.5	308.3	343.7	310.7	1.5	5.9	1.5	2.2
4611.1	1.5	308.4	343.9	310.8	1.5	6.0	1.5	2.3
3×7	38.6	313.8	751.0	769.7	21.2	7.3	23.0	7.2
B	6.6	363.8	797.2	817.0	4.9	39.9	5.1	43.3
4805.1	5.7	371.1	804.1	824.0	4.3	45.3	4.5	49.3
3×7	156.0	279.7	1319.0	1274.7	21.5	2.4	27.0	3.7
A	19.2	923.4	1934.1	1897.3	9.2	91.4	10.2	108.6
2742.8	17.9	1032.7	2038.5	2003.1	8.2	123.0	9.2	144.8

Table 4.7 (continuation): *Exact solutions of randomly generated problems* $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$

$n \times m$ type λ	λ' λ^* λ^T	CPU time for Scheme I	CPU time for Scheme II	CPU time for Scheme III	λ' λ^* λ^T	CPU time for Scheme IV	λ' λ^* λ^T	CPU time for Scheme V
1	2	3	4	5	6	7	8	9
3×8	20.4	2338.2	2106.5	1935.0	15.7	19.7	15.8	7.1
C	4.7	2348.2	2115.2	1938.5	4.0	28.1	4.0	15.5
21923.1	4.6	2354.6	2121.7	1943.6	3.9	33.2	3.9	20.6
$3 \times 8^*$	28.7	2060.8	4192.0	4018.8	26.0	9.9	27.0	7.1
B	7.4	2172.5	4282.2	4107.4	7.2	57.5	7.2	55.4
8810.2	7.3	2196.4	4302.5	4120.9	6.3	80.3	6.3	78.2
$3 \times 8^*$	141.5	2054.5	4119.2	3742.9	56.3	9.0	59.2	12.7
A	17.3	3999.9	5938.1	5556.0	19.1	1477.5	21.0	1463.0
8059.0	16.3	4501.5	6039.7	5811.8	17.6	1952.4	19.6	1963.1
4×3	13.3	19.2	14.2	9.5	11.4	6.1	11.7	2.3
C	3.2	20.3	15.4	10.7	3.2	7.3	3.2	3.5
2906.6	2.8	21.0	17.4	12.7	2.8	8.1	2.8	4.3
4×3	38.9	17.5	27.8	28.8	29.9	5.4	30.7	3.4
B	6.3	41.4	50.8	51.9	5.5	24.5	5.5	22.6
2217.4	5.5	45.4	54.8	55.8	4.8	28.3	4.8	26.4
4×3	290.9	28.6	70.7	69.0	110.1	5.9	115.2	8.1
A	31.0	414.0	440.4	442.0	22.5	288.4	22.8	293.9
2990.2	27.7	688.6	699.1	703.9	19.6	501.9	19.9	509.8
4×4	34.2	303.1	867.3	852.4	23.9	15.8	24.1	5.9
C	7.5	328.5	891.5	876.5	6.5	33.4	6.5	23.6
17159.1	6.3	330.5	893.0	878.0	5.5	35.1	5.5	25.3
$4 \times 4^*$	88.3	308.2	1501.2	1354.7	52.2	14.4	52.9	9.0
B	16.1	1293.8	2444.9	2297.2	13.5	782.8	13.5	780.5
17767.7	14.5	1574.0	2771.7	2580.5	12.0	999.6	12.0	997.6
4×4	477.7	319.4	2682.9	2505.1	131.7	15.4	132.0	23.8
A	30.8	3070.1	5355.7	5180.0	24.8	2680.7	24.8	2639.4
16142.6	30.1	3466.2	5771.4	5594.7	24.0	2905.8	24.0	2879.2

bound is equal to $(1 - 0.01)p_{ij}$ and the upper bound is equal to $(1 + 0.01)p_{ij}$. Table 4.7 for $\sum \mathcal{C}_i$ (Table 4.8 for $\mathcal{C}_{max}$) presents the results for the following three computational schemes, in which Algorithms $EXPL$, $B\&B1$ and $B\&B2$ are used with $\Phi^p_s = L^p_s$ (with $\Phi^p_s = l^p_s$, respectively) for the mean flow time criterion (for the makespan criterion).

Scheme I: Algorithm $EXPL \rightarrow$
Algorithm $SOL\_\sum \mathcal{C}_i$ (Algorithm $SOL\_\mathcal{C}_{max}$) $\rightarrow$
Algorithm $MINSOL^*\_\sum \mathcal{C}_i$ (Algorithm $MINSOL^*\_\mathcal{C}_{max}$)

Scheme II: Algorithm $B\&B1 \rightarrow$
Algorithm $SOL\_\sum \mathcal{C}_i$ (Algorithm $SOL\_\mathcal{C}_{max}$) $\rightarrow$
Algorithm $MINSOL^*\_\sum \mathcal{C}_i$ (Algorithm $MINSOL^*\_\mathcal{C}_{max}$)

Scheme III: Algorithm $B\&B2 \rightarrow$
Algorithm $SOL\_\sum \mathcal{C}_i$ (Algorithm $SOL\_\mathcal{C}_{max}$) $\rightarrow$
Algorithm $MINSOL^*\_\sum \mathcal{C}_i$ Algorithm $MINSOL^*\_\mathcal{C}_{max}$

Each of these schemes constructs first a solution B, then a solution $\Lambda^*(G)$ by Algorithm $SOL\_\sum \mathcal{C}_i$ ($SOL\_\mathcal{C}_{max}$) and finally a minimal solution $\Lambda^T(G)$ by Algorithm

Table 4.8: *Exact solutions of randomly generated problems* $\mathcal{J}/a_i \le p_i \le b_i/C_{max}$

$n \times m$ type λ	λ' λ^* λ^T	CPU time for Scheme I	CPU time for Scheme II	CPU time for Scheme III	λ' λ^* λ^T	CPU time for Scheme IV	λ' λ^* λ^T	CPU time for Scheme V
1	2	3	4	5	6	7	8	9
3×3	4.2	.17	.31	.15	3.4	.22	3.4	.10
C	1.3	.17	.31	.16	1.3	.26	1.3	.15
91.2	1.1	.17	.31	.16	1.2	.26	1.3	.16
3×3	16.0	.18	.28	.28	7.4	.14	7.8	.14
B	2.6	.18	.29	.30	2.0	.19	2.0	.19
90.7	2.4	.20	.30	.30	1.7	.20	1.6	.19
3×3	12.8	.14	.31	.30	8.3	.19	8.5	.17
A	2.2	.15	.32	.31	2.2	.24	2.2	.23
77.4	2.0	.15	.32	.31	2.0	.25	2.0	.23
3×4	7.0	.76	.86	.53	4.0	.48	4.3	.22
C	1.5	.77	.86	.54	1.5	.52	1.5	.25
261.9	1.3	.77	.86	.54	1.3	.53	1.3	.25
3×4	18.1	.85	1.0	.92	10.9	.55	11.0	.42
B	3.0	.90	1.1	.95	2.5	.61	2.7	.49
300.8	2.2	.90	1.1	.96	2.0	.64	2.0	.51
3×4	36.4	.80	1.22	1.14	13.6	.46	15.0	.45
A	7.1	.85	1.27	1.19	5.1	.54	5.5	.55
276.8	5.8	.89	1.29	1.21	4.2	.57	4.5	.59
3×5	8.0	4.05	2.87	2.24	5.4	.87	5.5	.40
C	2.2	4.06	2.88	2.25	1.9	.93	1.9	.45
604.8	1.6	4.07	2.89	2.25	1.7	.94	1.7	.45
3×5	11.8	4.79	4.48	3.22	7.2	1.33	7.6	.44
B	3.1	4.83	4.50	3.23	2.4	1.41	2.4	.52
894.1	2.3	4.83	4.51	3.23	2.1	1.42	2.1	.52
3×5	103.7	5.25	9.19	8.80	24.9	1.43	29.2	1.32
A	17.8	5.48	9.41	9.02	8.2	1.61	8.5	1.52
896.7	13.8	5.62	9.48	9.08	6.6	1.70	6.8	1.63
3×6	7.0	25.98	20.83	11.42	4.1	1.49	4.7	.57
C	2.4	25.99	20.83	11.43	2.1	1.54	2.3	.62
1555.9	2.2	25.99	20.83	11.43	1.9	1.54	2.1	.63
3×6	21.6	25.73	22.48	19.36	11.5	2.15	12.2	1.08
B	4.3	25.79	22.54	19.42	3.9	2.26	3.9	1.23
1760.9	3.5	25.83	22.57	19.43	3.1	2.31	3.1	1.29
3×6	67.8	25.37	54.80	54.62	18.1	1.48	19.5	1.30
A	12.5	25.64	55.06	54.89	7.2	1.68	8.0	1.51
1559.0	8.8	25.78	55.18	55.01	4.8	1.83	5.4	1.68
3×7	7.5	169.47	114.97	99.24	4.0	3.18	4.1	.60
C	1.7	169.51	114.99	99.26	1.4	3.26	1.4	.68
4611.1	1.5	169.51	115.00	99.27	1.3	3.28	1.3	.70
3×7	42.4	180.42	264.59	261.54	17.8	3.93	19.7	2.29
B	7.1	180.65	264.80	261.74	4.4	4.17	5.3	2.55
4805.1	5.7	180.71	269.88	261.78	3.9	4.29	4.4	2.69
3×7	90.4	152.22	604.40	523.97	16.6	1.39	19.2	1.66
A	17.7	152.76	604.89	524.45	7.6	1.59	8.6	1.91
2742.8	13.2	153.00	605.10	524.66	5.6	1.72	6.3	2.06

Table 4.8 (continuation): *Exact solutions of randomly generated problems $\mathcal{J}/a_i \le p_i \le b_i/\mathcal{C}_{max}$*

$n \times m$ type λ	λ' λ^* λ^T	CPU time for Scheme I	CPU time for Scheme II	CPU time for Scheme III	λ' λ^* λ^T	CPU time for Scheme IV	λ' λ^* λ^T	CPU time for Scheme V
1	2	3	4	5	6	7	8	9
3×8	19.3	1297.43	841.55	795.93	11.3	11.39	11.2	2.66
C	5.2	1297.54	841.65	796.02	3.9	11.53	4.3	2.83
21923.1	4.6	1297.58	841.68	796.05	3.4	11.60	3.6	2.92
$3 \times 8^*$	32.9	1190.78	1959.74	1938.03	13.5	5.74	14.4	2.10
B	7.2	1191.12	1960.05	1938.35	4.6	6.01	4.9	2.37
8960.8	4.4	1191.19	1960.11	1938.39	3.2	6.18	3.2	2.55
$3 \times 8^*$	160.3	1161.51	3411.75	3022.93	40.9	5.21	44.8	6.73
A	23.6	1164.83	3414.66	3026.13	12.2	6.39	13.1	7.99
8296.1	19.0	1166.78	3415.74	3027.31	9.9	7.69	10.2	9.42
4×3	18.1	8.47	6.35	3.93	12.1	3.47	12.4	1.11
C	3.3	8.55	6.41	3.99	3.1	3.56	3.1	1.20
2906.6	2.1	8.56	6.41	3.99	2.4	3.61	2.1	1.24
4×3	40.9	7.35	10.01	8.67	27.7	3.19	29.3	1.29
B	7.2	7.48	10.10	8.77	5.0	3.34	7.0	1.45
2217.4	4.2	7.50	10.12	8.78	2.0	3.39	3.8	1.53
4×3	286.4	8.94	23.07	21.36	92.3	3.50	96.3	3.64
A	24.8	9.61	23.64	21.95	17.2	4.08	16.3	4.25
2990.2	21.3	9.74	23.75	22.03	14.1	4.45	13.3	4.61
4×4	41.8	164.20	199.87	201.76	19.9	7.84	19.9	2.53
C	6.4	164.41	200.05	201.95	3.8	7.99	3.8	2.67
17159.1	2.4	164.43	200.05	201.96	2.4	7.04	2.4	2.71
$4 \times 4^*$	79.0	169.08	199.25	190.39	27.1	7.52	27.3	2.47
B	14.7	169.58	199.69	190.85	7.2	7.79	6.9	2.73
17763.3	9.5	169.62	199.72	190.88	4.5	7.94	4.4	2.86
4×4	434.9	164.40	729.63	638.22	104.2	8.27	112.8	9.52
A	43.5	165.76	730.90	639.48	20.5	9.34	25.7	10.74
16142.6	34.8	166.43	731.42	640.01	15.6	10.17	20.0	11.70

$MINSOL^*\_\sum\mathcal{C}_i$ ($MINSOL^*\_\mathcal{C}_{max}$). In Table 4.7 (Table 4.8), λ denotes the average number of schedules (third row in column 1), λ' the average cardinality of the set B, λ^* the average cardinality of the set $\Lambda^*(G)$, and λ^T the average cardinality of the set $\Lambda^T(G)$ (first, second and third rows in column 2, respectively). Of course, for each instance $\lambda, \lambda', \lambda^*$ and λ^T are integers, but their average values are real numbers given in Table 4.7, Table 4.8 and the tables below with one decimal place.

The application of Scheme I to Example 3.1 of problem $\mathcal{J}3/n=3, a_i \le p_i \le b_i/\sum\mathcal{C}_i$ described in Section 3.1, gives the following sets of schedules. Firstly, Algorithm $EXPL$ constructs the set $\Lambda(G) = \{G_1, G_2, \dots, G_{22}\}$ of all schedules and set $B \subset \Lambda(G)$ with $|B| = 12$ (see Step 4, where sufficiency from Lemma 3.1 is used). Then using solution B, Algorithm $SOL\_\sum\mathcal{C}_i$ constructs the set $\Lambda^*(G) = \{G_1, G_2, G_5\}$ which is also a solution. Finally, Algorithm $MINSOL\_\sum\mathcal{C}_i$ shows that solution $\Lambda^*(G)$ is minimal, i.e. $\Lambda^T(G) = \Lambda^*(G)$. Thus, for Example 3.1 we have $\lambda = 22, \lambda' = 12$ and $\lambda^* = \lambda^T = 3$.

The average CPU time (in seconds) for constructing set B, set $\Lambda^*(G)$ and set $\Lambda^T(G)$ (first, second and third rows) are presented in columns 3, 4 and 5 for Schemes I, II and III, respectively. As follows from Table 4.7 and Table 4.8, in most cases for both criteria $\sum\mathcal{C}_i$ and $\mathcal{C}_{max}$, Scheme III based on Algorithm B&B2 is the best for the problems of type C, while Scheme I based on Algorithm $EXPL$ is the best for the problems of types

A and B. As it was mentioned, Steps 8 and 9 for the branch-and-bound algorithm are not so fast as Step 1 of Algorithm $EXPL$. Moreover, due to a large uncertainty of the input vector p for problems A and B, Algorithms $B\&B1$ and $B\&B2$ have to construct a lot of intermediate digraphs $G_{(t)}$ in the branching tree which are not in the set $\Lambda(G)$. Unfortunately, the exact minimal solution was obtained within 1.5 hours by the worst of the Schemes I, II or III only for some combinations of n and m with $n \le 4$ and $m \le 8$ and the exact solution was not obtained by Scheme I for some combinations of n and m for the reason 'not enough memory' or 'limit of time' (such series are marked in the first column of Table 4.7 and Table 4.8 by an asterisk).

To solve problems with larger size, we were forced to consider restricted variants of the branch-and-bound algorithms: Algorithm $B\&B1^*$ (Algorithm $B\&B2^*$) denotes Algorithm $B\&B1$ (Algorithm $B\&B2$, respectively) without Steps 8 and 9. In general, such modifications do not guarantee to obtain a solution B, but they are essentially faster. Fortunately, for almost all problems presented in Table 4.7 and Table 4.8, the restricted variants of the branch-and-bound algorithms still give a solution, i.e. for each $p \in T$ the set B constructed contains an optimal schedule. The main reason for this computational result is that Steps 8 and 9 often generate only schedules which are dominated by other ones. Therefore, it is possible to exclude these schedules due to Theorem 3.1. Columns 6 – 9 of Table 4.7 (of Table 4.8) present computational results on a PC 486 (120 MHz) (on a PC 486 (133 MHz), respectively) for the following two computational schemes.

Scheme IV: Algorithm $B\&B1^*$ with $\Phi^p_s = L^p_s$ (with $\Phi^p_s = l^p_s$) $\rightarrow$
Algorithm $SOL\_\sum\mathcal{C}_i$ (Algorithm $SOL\_\mathcal{C}_{max}$) $\rightarrow$
Algorithm $MINSOL^*\_\sum\mathcal{C}_i$ (Algorithm $MINSOL^*\_\mathcal{C}_{max}$)

Scheme V: Algorithm $B\&B2^*$ with $\Phi^p_s = L^p_s$ (with $\Phi^p_s = l^p_s$) $\rightarrow$
Algorithm $SOL\_\sum\mathcal{C}_i$ (Algorithm $SOL\_\mathcal{C}_{max}$) $\rightarrow$
Algorithm $MINSOL^*\_\sum\mathcal{C}_i$ (Algorithm $MINSOL^*\_\mathcal{C}_{max}$)

More precisely, column 6 presents the average approximate values λ' (first row), λ^* (second row) and λ^T (third row) calculated by Algorithm $B\&B1^*$, and column 7 presents the average running times for constructing approximations of the sets $B, \Lambda^*(G)$ and $\Lambda^T(G)$ by Algorithm $B\&B1^*$. Similarly, column 8 presents the average approximate values λ', λ^* and λ^T calculated by Algorithm $B\&B1^*$, and column 9 presents the average running times for constructing approximations of the sets $B, \Lambda^*(G)$ and $\Lambda^T(G)$ by Algorithm $B\&B2^*$. From Table 4.7 for criterion $\sum\mathcal{C}_i$ and Table 4.8 for criterion $\mathcal{C}_{max}$, it follows that Algorithm $B\&B2^*$ (with the corresponding criterion) in Scheme V is often faster than Algorithm $B\&B1^*$ in Scheme IV: Only for some series of type A, Algorithm $B\&B1^*$ is, on average, faster than Algorithm $B\&B2^*$. Note also that Algorithm $B\&B2^*$ gives more often an exact solution than Algorithm $B\&B1^*$.

As it follows from Table 4.7 and Table 4.8, even the heuristic Schemes IV and V require rather large running times. So, for larger problem sizes, we used only Algorithms $B\&B1^*$ and $B\&B2^*$ for constructing the sets B heuristically, i.e. without a guarantee that the constructed set B is indeed a solution. Obviously, the cardinality of a solution increases not only with increasing the size of the problem (which in turn increases the running time), but also with increasing the uncertainty of the numerical input data. Therefore, to reduce the cardinality of a solution, we consider along with instances of types A, B, and C also problems of the following two types D and E with smaller errors of the given processing times, namely: The problems of type D with the errors of the processing times

Table 4.9: *Heuristic solutions of randomly generated problems* $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$

n x m; type		k	$B\&B1^*$				$B\&B2^*$			
			λ'			CPU	λ'			CPU
			MIN	AVE	MAX	time	MIN	AVE	MAX	time
1		2	3	4	5	6	7	8	9	10
4 x 4;	B	150	18	52.3	102	34.3	18	52.9	103	21.5
4 x 4;	C	150	4	22.4	53	35.8	4	22.6	53	13.0
4 x 4;	D	150	1	11.1	39	47.1	1	12.2	50	12.2
4 x 5;	C	150	5	35.8	140	102.9	5	37.0	145	41.3
4 x 5;	D	150	2	8.6	18	72.8	2	8.6	18	18.8
4 x 6;	C	150	15	33.8	78	148.5	15	35.6	78	63.1
4 x 6;	D	150	1	7.2	12	100.5	1	7.3	12	22.3
4 x 7;	C	150	6	52.0	134	170.6	5	54.6	136	98.9
4 x 7;	D	150	3	12.8	29	177.3	3	12.9	29	56.4
4 x 8;	C	150	15	54.4	120	416.4	14	58.7	122	292.0
4 x 8;	D	150	7	27.5	57	287.8	7	27.8	55	134.2
4 x 9;	C	150	6	78.0	150	495.1	8	80.3	150	335.6
4 x 9;	D	150	3	22.9	56	458.4	4	22.9	54	156.1
4 x 10;	C	150	25	86.6	150	682.9	24	87.8	150	852.5
4 x 10;	D	150	3	28.5	70	707.9	3	29.1	66	362.5
5 x 3;	C	150	19	62.0	146	85.9	19	62.8	147	65.2
5 x 3;	D	150	2	38.6	150	95.1	2	38.5	150	51.9
5 x 4;	C	150	11	63.1	150	191.8	11	64.3	150	154.0
5 x 4;	D	150	2	23.2	50	182.6	2	23.4	52	106.7
5 x 5;	C	150	63	114.5	150	500.5	62	116.2	150	854.6
5 x 5;	D	150	11	36.9	133	499.0	11	37.4	139	291.1
5 x 5;	E	100	1	1.7	4	366.0	1	1.7	4	86.6
5 x 6;	C	150	15	81.4	150	862.3	16	82.7	150	1220.3
5 x 6;	D	150	7	49.0	89	761.5	7	48.6	88	493.6
5 x 7;	D	150	9	47.9	150	1390.3	9	48.9	150	1642.0
5 x 7;	E	50	1	2.6	7	539.3	1	2.6	7	214.7
5 x 8;	D	100	18	78.5	100	1803.5	18	80.5	100	2446.7
5 x 8;	E	50	1	3.2	6	1054.5	1	3.2	6	328.1
5 x 9;	E	50	1	2.5	6	1531.3	1	2.5	6	653.4
5 x 10;	E	50	1	2.5	5	2071.7	1	2.5	5	617.9
6 x 3;	D	150	19	101.3	150	538.4	19	100.3	150	621.4
6 x 3;	E	50	1	4.2	18	456.8	1	4.2	18	309.8
6 x 4;	D	150	20	99.9	150	1197.8	18	81.3	150	1858.1
6 x 4;	E	100	1	2.3	6	936.7	1	2.3	6	403.6
6 x 5;	D	100	6	90.1	100	1671.0	6	88.1	100	3022.7
6 x 5;	E	50	1	2.8	8	1382.4	1	2.8	8	724.1
6 x 6;	C	50	50	50	50	2389.6	50	50	50	7350.4
6 x 6;	D	50	15	46.5	50	1997.6	15	46.5	50	5252.0
6 x 6;	E	50	1	4.1	12	1997.6	1	3.5	12	1226.2
7 x 3;	D	150	42	122.5	150	1311.9	76	131.8	150	2302.3
7 x 4;	E	100	1	7.1	20	2204.5	1	7.0	24	3608.4
7 x 5;	E	50	1	8.4	39	3074.2	2	15.7	50	6139.9
8 x 3;	E	50	1	4.5	9	1781.5	1	5.1	11	3103.3
9 x 2;	E	100	1	14.1	100	1297.3	1	14.9	100	1958.7
10 x 2;	E	50	2	14.1	50	1651.6	2	9.3	50	2781.4

equal to 1 %, 2 %, 3 % and 4 %, and problems of type E with the errors of the processing times equal to 0.1 %, 0.2 %, 0.3 % and 0.4 % (see Table 4.5).

Table 4.10: *Heuristic solutions of randomly generated problems $\mathcal{J}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$*

n x m; type		$B\&B1^*$				$B\&B2^*$			
		λ'			CPU	λ'			CPU
		MIN	AVE	MAX	time	MIN	AVE	MAX	time
1		2	3	4	5	6	7	8	9
4 x 4;	B	5	27.2	51	7.29	5	27.3	50	2.46
4 x 4;	C	2	19.9	62	7.59	2	19.9	64	2.51
4 x 4;	D	4	11.8	28	9.70	4	12.5	35	1.81
4 x 5;	C	7	32.1	69	15.97	7	33.1	74	6.58
4 x 5;	D	1	5.2	19	14.10	1	5.4	20	1.56
4 x 6;	C	6	28.0	79	27.53	6	29.7	82	8.69
4 x 6;	D	2	10.6	45	21.91	2	10.7	44	3.82
4 x 7;	C	6	38.7	118	31.93	5	41.3	129	16.43
4 x 7;	D	1	12.4	45	35.68	1	13.4	45	7.85
4 x 8;	C	14	60.6	143	63.53	18	66.0	134	38.91
4 x 8;	D	3	18.3	45	51.04	3	18.5	45	13.59
4 x 9;	C	8	55.1	145	76.46	8	60.6	150	42.13
4 x 9;	D	1	24.5	101	74.88	1	25.8	109	25.55
4 x 10;	C	16	74.9	150	100.86	17	82.3	150	78.79
4 x 10;	D	5	29.6	81	95.54	5	31.8	86	51.35
5 x 3;	C	22	101.0	150	13.06	22	100.4	150	11.64
5 x 3;	D	19	93.1	150	13.15	19	93.3	150	12.18
5 x 4;	C	17	86.1	150	26.86	17	88.4	150	21.81
5 x 4;	D	17	42.1	102	29.31	17	44.1	102	15.63
5 x 5;	B	57	133.1	150	46.29	60	136.4	150	128.36
5 x 5;	C	86	142.2	150	55.33	88	143.3	150	125.10
5 x 5;	D	13	59.7	150	55.67	13	60.7	150	31.70
5 x 5;	E	1	14.6	84	55.43	1	15.1	89	12.20
5 x 6;	C	19	91.3	150	83.44	23	95.9	150	94.73
5 x 6;	D	7	49.8	104	88.35	7	54.0	131	52.46
5 x 7;	D	35	103.9	150	148.23	35	104.6	150	364.76
5 x 7;	E	1	8.2	24	112.03	1	8.2	24	21.18
5 x 8;	D	22	92.1	150	211.49	21	91.9	150	331.82
5 x 8;	E	2	11.3	58	179.53	2	10.9	58	49.70
5 x 9;	E	1	8.9	57	233.89	1	9.1	57	49.22
5 x 10;	E	1	5.9	21	309.72	1	5.9	21	53.87
6 x 3;	D	25	132.7	150	57.70	25	137.5	150	141.25
6 x 3;	E	5	102.9	150	65.60	8	119.7	150	160.52
6 x 4;	D	78	139.7	150	130.70	86	140.5	150	327.15
6 x 4;	E	2	59.2	150	100.75	2	59.6	150	62.80
6 x 5;	D	109	135.8	150	155.49	112	139.3	150	319.53
6 x 5;	E	1	37.0	143	157.26	1	37.0	143	125.21
6 x 6;	C	150	150	150	271.68	150	150	150	1136.78
6 x 6;	D	25	110.0	150	255.49	42	126.3	150	550.78
6 x 6;	E	1	17.6	43	255.48	1	17.7	43	133.24
7 x 3;	D	150	150	150	169.66	110	146.0	150	414.17
7 x 4;	E	4	87.1	150	268.30	12	96.2	150	505.38
7 x 5;	E	5	87.1	150	342.51	2	124.7	150	1057.61
8 x 3;	E	2	134.4	150	234.02	36	138.6	150	585.55
9 x 2;	E	150	150	150	157.68	150	150	150	416.59
10 x 2;	E	150	150	150	209.69	150	150	150	606.73

Heuristic solutions are represented in Table 4.9 for problems $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$ and in Table 4.10 for problems $\mathcal{J}/a_i \le p_i \le b_i/\mathcal{C}_{max}$ with the same structural and numerical input data. Next, we describe the design of Table 4.9. The values of k used for criterion $\sum \mathcal{C}_i$ are given in column 2. Columns 3 - 6 (and columns 7 - 10) present computational results for Algorithm $B\&B1^*$ (and Algorithm $B\&B2^*$, respectively). Column 3 (column 7) gives the minimal value of the cardinality λ' of the set B constructed, column 4 (column 8) the average value of λ', and column 5 (column 9) the maximal value of λ'. The average CPU times are given in column 6 for Algorithm $B\&B1^*$ and in column 10 for Algorithm $B\&B2^*$. Table 4.10 has a similar design with the exception of the column with the values of k used. For criterion $\mathcal{C}_{max}$, we set $k = 150$ for all computational results presented for problems $\mathcal{J}/a_i \le p_i \le b_i/\mathcal{C}_{max}$. As follows from Table 4.9 and Table 4.10, if the problems have small size, Algorithm $B\&B2^*$ is preferable to Algorithm $B\&B1^*$ (in both running time and the quality of the solution constructed). However, if the number of potentially optimal schedules is large (due to a large problem size or due to a large uncertainty of the numerical input data), then Algorithm $B\&B1^*$ has a smaller running time, however the quality of the solution constructed by Algorithm $B\&B2^*$ remains still better. Moreover, the value of k has a large influence on the quality and the running time of Algorithm $B\&B1^*$ in contrast to Algorithm $B\&B2^*$ which is independent of k. In principle, we use the parameter k in Algorithm $B\&B2^*$ mainly to have the same conditions for the comparison with Algorithm $B\&B1^*$.

4.5 Remarks on the Scheduling Problems with Uncertainty

On the basis of the characterizations of a solution and a minimal solution derived in Section 3.2, we have developed an explicit enumeration scheme for the flow shop and for the classical job shop problems and branch-and-bound schemes for the general job shop problem. Instead of Algorithm $B\&B1$, one can use any known branch-and-bound method developed for problems $\mathcal{J}//\Phi$ (with fixed numerical input data) after a simple modification with the aim to construct the k best schedules instead of only an optimal one. However, the question which still remains open, is how to choose k to have a guarantee that the k best schedules contain an exact solution of problem $\mathcal{J}/a_i \le p_i \le b_i/\Phi$. To answer this question, we have available only experimental results given in Section 4.2 for calculating the stability radius of an optimal schedule for randomly generated job shop problems with $n \le 10$, $m \le 7$ and small numbers of conflict edges in the mixed graph G. In particular, in all experiments presented in Table 4.7 and Table 4.8, we used $k = 150$ which was sufficient to obtain an exact solution for almost all problems considered. For the computational results presented in Table 4.9 for criterion $\sum \mathcal{C}_i$, we used $k = 150, k = 100$ and $k = 50$ depending on the problem size and on the uncertainty of the input vector of the processing times (which is defined by the problem type A, B, C, D or E). As follows from Table 4.9 and Table 4.10 (in which $k = 150$), these values of k were not sufficient for some instances in the sense that the cardinality of the set of schedules which may be optimal for some vectors $p \in T$ was larger than the value of k used. E.g., for each instance of type C with $n = m = 6$, the number of such schedules was larger than 50 (although $k = 50$ was used).

Algorithm $B\&B2$ constructs the set of all 'potentially optimal schedules' for problem $\mathcal{J}/a_i \le p_i \le b_i/\sum \mathcal{C}_i$: If a schedule may be optimal for some feasible vector of the processing times, it has to be contained in the set constructed by Algorithm $B\&B2$. Thus, Algorithm

$B\&B2$ guarantees an exact solution after a complete realization. Moreover, its running time was often less than that of Algorithm $B\&B1$. The 'heuristic' Algorithm $B\&B2^*$ was also often preferable in comparison with Algorithm $B\&B1^*$ in both running times and in the numbers of instances for which a better solution was obtained.

The comparison with Algorithm $EXPL$ may be as follows. On the one hand, Algorithm $EXPL$ takes often smaller running times for problems of types B and C. On the other hand, it is suitable only for a classical job shop problem (and for a flow shop problem as its special case), while Algorithms $B\&B1$ and $B\&B2$ are suitable for the general case of a job shop problem. Moreover, Algorithm $EXPL$ may be realized efficiently only in the exact version in contrast to Algorithms $B\&B1$ and $B\&B2$ which have efficient heuristic versions as well (namely Algorithms $B\&B1^*$ and $B\&B2^*$). Note also that some advantage of Algorithm $EXPL$ is based on the fast generation of all digraphs $G_s^T,\ G_s \in \Lambda(G)$, which have only dominant paths (see Steps 1 and 2). The dominance relation $\preceq_T$ between digraphs is tested in Step 4 of Algorithm $EXPL$. In contrast to Algorithm $EXPL$, both above branch-and-bound algorithms (and their four heuristic versions) first test dominance relation $\preceq_T$ (see Step 5) and then use a more time-consuming procedure (but which is suitable for the general case of a job shop problem) for excluding redundant paths from the digraph $G_s \in \Lambda(G)$.

It should be noted that the software developed allows to solve within one hour problems $\mathcal{J}/a_i \le p_i \le b_i / \sum \mathcal{C}_i$ exactly with n x $m \le 25$ on a PC 486 (120 MHz) and with n x $m \le 50$ heuristically on a PC 486 (50 MHz). The cardinality of the set $\Lambda^T(G)$ (and as a result the running time of the above algorithms) grows very quickly with increasing the problem size or/and the size of the polytope T. So, the software developed may be practically efficient only if at least one of the above sizes is sufficiently small.

The developed approach seems to be useful for a preliminary analysis of a scheduling environment with uncertain numerical data and fixed structural data. After calculating a solution $\Lambda^*(G)$ or (what is better but more time-consuming) a minimal solution $\Lambda^T(G)$, a decision-maker may quickly choose the best schedule if additional information on the numerical data will be available at the stage of the realization of a schedule. To this end, it is desirable to construct at the stage of scheduling a small number of schedules which may be considered as possible candidates for a realization. Note that, for criterion $\sum \mathcal{C}_i$, the average value of $\lambda^T = |\Lambda^T(G)|$ (see column 2) was equal to 8 for the instances presented in Table 4.7, and for the instances presented in Table 4.9, the average value of $\lambda' = |B|$ (see columns 4 and 8) was equal to 38. Unfortunately, in the latter case for 12 % of the instances under consideration, the number of potentially optimal schedules exceeded the value of k used.

As it was noted in [Alt00a], "typically an optimal solution for a model of the problem situation is generated (often by a computer), and this solution is translated back to a solution for the original real world problem". Usually, real world problems are of a large size and with a large uncertainty of the numerical input data. Since algorithms for constructing k best solutions do not work in such cases, Althöfer [Alt00b] presents an approach where 'true alternatives' are generated one after the other by introducing penalties. The development of such algorithms for scheduling problems may be a direction of some future work.

Next, we discuss how it is possible to use the results of this dissertation for the following two-stage practical processes. We propose to consider two sequential stages. At the first stage (*scheduling problem*), a set of *potentially* optimal schedules has to be constructed under the conditions of uncertain numerical input data. In other words, problem $\mathcal{G}/a_i \le$

$p_i \leq b_i/\Phi$ has to be solved, i.e. a minimal solution $\Lambda^T(G)$ has to be found. Each schedule from the set $\Lambda^T(G)$ is a *potentially* optimal schedule, and at the first stage, a decision-maker does not know which schedule from the set $\Lambda^T(G)$ will be the best in reality.

At the second stage (*control problem*), it is required to choose an optimal schedule from the set $\Lambda^T(G)$ and to realize it taking into account the additional information about the processing times of the operations. By a solution of the *control problem*, we mean a digraph $G_s \in \Lambda^T(G)$, which is optimal for the vector $p^* = (p_1^*, p_2^*, \ldots, p_q^*) \in T$ of the processing times: $\Phi_s^{p^*} = \min\{\Phi_k^{p^*} \ : \ G_k \in \Lambda^T(G)\}$, where p_i^* denotes the actual processing time of operation $i \in Q$. In the worst case, p_i^* may become known only after the completion of operation $i \in Q$. The digraph $G_s \in \Lambda^T(G)$, for which equality $\Phi_s^{p^*} = \min\{\Phi_k^{p^*} \ : \ G_k \in \Lambda^T(G)\}$ holds, is *definitely* optimal.

The problem of the first stage (*scheduling problem*) and the problem of the second stage (*control problem*) are distinguished by the time which is acceptable for the decision-making: To solve the *scheduling problem*, it is possible to use essentially more time than for the *control problem*, which has to be solved in a very short time as in on-line scheduling (see [CV97, HZ97, San95, Sei98]).

In contrast to the *scheduling problem*, which necessarily has a solution if digraph $(Q, A, \emptyset)$ does not contain a circuit, a solution (i.e. a definitely optimal schedule) of the *control problem* may not exist from some instant (i.e. an optimal continuation of a partial schedule which was realized may not exist after previous false decisions).

Some more details about the above two-stage processes are given in [ST98, Sotskova00]. In particular, sufficient conditions have been derived for some cases when *control problem* has a solution.

Conclusions

In spite of a large number of papers and books published about optimal sequencing and scheduling, the utilization of numerous results of the scheduling theory in most production environments is far from the desired volume. One of the reasons for the gap between scheduling theory and practice is connected with the usual assumption that the processing times of the jobs are known exactly before scheduling (for deterministic models) or that they are random values with known probability distributions (for stochastic models).

In the first part of these Conclusions, we summarize what we have learned from studying the scheduling paradigm when the processing times of the operations (and/or other numerical input data) are uncertain before scheduling. In the second part, we outline some topics for future research which follow directly from the above results. The common notations used in this dissertation are combined in Table 4.11, see also special notations for the general shop (Table 2.7 at page 53) and special notations for the job shop (Table 3.10 at page 94).

1. In this dissertation, a model of more realistic scheduling scenarios was considered. It was assumed that in the practical realization of a schedule, the processing time of an operation may take any real value between lower and upper bounds (within the polytope T), which are given before applying a scheduling procedure, 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 a stability analysis of an optimal schedule with respect to the perturbations of the processing times (a survey of the main results on stability of an optimal schedule was given in Chapter 1).

In Chapter 2 and Chapter 3, we introduced the notion of the relative stability radius of an optimal schedule s as the maximal value of the radius of a stability ball (in the space of real vectors of the processing times) within which schedule s remains the best among the given set B of schedules (see Definition 2.2 at page 34 for the maximum flow time criterion and Definition 3.2 at page 63 for the mean flow time criterion). The relativity is considered with respect to the polytope T of 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.

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 (see Table 2.1 at page 27). The most results obtained in this dissertation (see Chapters 2 and 3) are formulated in terms of paths in the digraphs G_s.

In Chapter 2, we focused on dominance relations between feasible schedules taking into account the given polytope T (Section 2.2). 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 (Theorem 2.2 at page 39). Under such conditions, schedule s remains optimal (has the minimal length) 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 (Theorem 2.1 at page 36). 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 are based on a comparison of an optimal schedule s with other schedules from the set B (Theorem 2.3 at page 41), and we show how it is possible to restrict the number of schedules from the set B examined for such a calculation of the relative stability radius (Lemma 2.3 at page 50). 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 (Corollary 2.5 at page 51).

In Chapter 3, analogous results were obtained for the mean flow time criterion, and the focus was on the dominance relations between feasible schedules taking into account the given criterion (Definition 3.1 at page 56). Formulas for calculating the exact value of the relative stability radius were given in Theorem 3.3 at page 66. A possibility to restrict the number of schedules under consideration has been discussed in Section 3.3. We established necessary and sufficient conditions for an infinitely large relative stability radius of an optimal schedule for the mean flow time criterion (Theorem 3.4 at page 67) and necessary and sufficient conditions for a zero relative stability radius of an optimal schedule (Theorem 3.5 at page 68).

Using these results, we developed several exact and heuristic algorithms for constructing a solution and a minimal solution (see Definition 2.1 at page 27) of a scheduling problem with uncertain processing times. The developed software was tested on randomly generated job shop problems, and the computational results were discussed in Chapter 4. For the maximum and mean flow time criteria, we calculated the stability radii of the optimal schedules for more than 10,000 randomly generated instances. For a randomly generated uncertain scheduling problems with the same criteria, we constructed solutions and minimal solutions as well. In the experiments both the numbers of jobs and machines were restricted by 10. The most critical parameter for the running time of the programs was the number of edges E in the mixed graph $G = (Q, A, E)$.

2. In conclusion, we present some topics for future research. We can note that the most part of this dissertation (Chapters 2, 3 and 4) is devoted to the *scheduling problem* (with uncertain numerical input data) for the criteria $\mathcal{C}_{max}$ and $\sum \mathcal{C}_i$. Other regular criteria for the *scheduling problem* may be the subject of further research.

In Section 4.5, we introduced the *control problem* (when uncertain numerical data are realized), which follows after the *scheduling problem*. For the *control problem*, only very preliminary results are known, however the *control problem* seems to be very important for practice. The *control problem* (in the setting presented in Section 4.5) may be a subject for future research.

The next important direction for further research is to construct more efficient algorithms for the *scheduling problem*, in particular, to use this approach for *scheduling problems* whose deterministic versions have polynomial algorithms for constructing optimal schedules.

Another direction of future research may be connected with the consideration of a minimal solution of an uncertain scheduling problem with respect to the cardinality of the solution obtained (the minimal solution $\Lambda^T(G)$ was determined with respect to inclusion).

After carrying out the computational experiments with the calculation of the stability radii of an optimal schedule (see Chapter 4), we can select the following topics for future research. For practical aims, it is useful to develop further a branch-and-bound algorithm for constructing the k best schedules (instead of one, which is usually constructed) and to combine such a calculation with a stability analysis on the basis of the results discussed in Chapters 1 and 4.

Another possible topic is to improve the bounds (2.34) at page 50 and (3.26) at page 81 in order to restrict the number of digraphs G_s, with which an optimal digraph has to be compared, while calculating its stability radius.

A more complex question is to find simpler (practical) formulas for calculating the stability radius or at least lower and/or upper bounds for it (without considering the paths of the digraph G_s).

If the calculation of a bound for the stability radius will be simplified considerably, it seems to be useful to calculate this bound within a branch-and-bound framework. One can obtain a new type of stopping rule (or some other advantages) due to a possible connection between the stability radius of the best constructed schedule and its proximity to the optimum.

Sections 4.1 and 4.2 were basically devoted to an overall enumeration scheme for calculating the stability radii, and an implicit enumeration scheme was only used *before* performing the stability analysis (i.e. for calculating optimal and near optimal schedules). The application of the stability analysis *within* an implicit enumeration framework should have practical utility and it may be a topic for future research, too.

The scheduling problem with uncertainty remains an interesting and challenging subject for the studies, which may combine some theoretical results with practical problems.

Table 4.11: *Common notations*

Symbols	Description
$\mathcal{J}$	Symbol for job shop (in the three-field notation)
$\mathcal{F}$	Symbol for flow shop
$\mathcal{O}$	Symbol for open shop
$\mathcal{G}$	Symbol for general shop
M	Set of machines: $M = \{M_1, M_2, \dots, M_m\}$
J	Set of jobs: $J = \{J_1, J_2, \dots, J_n\}$
$Q^{(k)}$	Set of operations of a job $J_k, 1 \leq k \leq n$
Q_k	Set of operations being processed on machine $M_k, 1 \leq k \leq m$, where $Q = \cup_{k=1}^{m} Q_k$ and $Q_k \cap Q_l = \emptyset$ if $k \neq l$
C_i	Completion time of job J_i
$\Phi(C_1, C_2, \dots, C_n)$	Given objective function
Φ_k^p	Value of the objective function calculated for schedule $G_k \in \Lambda(G)$ with fixed processing times $p \in R_+^q$
$\mathcal{C}_{max} = \max_{i=1}^{n} C_i$	Criterion of minimizing the maximum flow time (makespan)
$\sum \mathcal{C}_i = \sum_{i=1}^{n} C_i$	Criterion of minimizing the mean flow time (sum of job completion times)
R^q	Space of q-dimensional real vectors
R_+^q	Space of non-negative q-dimensional real vectors
$d(p, p')$	Distance between the vectors $p \in R^q$ and $p' \in R^q$
T	Polytope in the space R_+^q of feasible vectors of processing times
$G = (Q, A, E)$	Mixed graph which defines the structural input data
$G_k = (Q, A \cup E_k, \emptyset)$	Acyclic digraph (schedule) generated from the mixed graph G
E_k	Signature of schedule G_k
$G(p) = (Q(p), A, E)$	Weighted mixed graph
$G_k(p) = (Q(p), A \cup E_k, \emptyset)$	Acyclic weighted digraph
G_k^T	Minimal subgraph of G_k containing all dominant paths with respect to the polytope T
$\Lambda(G) = \{G_1, G_2, \dots, G_\lambda\}$	Set of all feasible (acyclic) digraphs generated from the mixed graph G
$S = \{1, 2, \dots, \lambda\}$	Set of all semiactive schedules
λ	Number of semiactive schedules
$\Lambda^*(G) \subseteq \Lambda(G)$	Solution of the scheduling problem with uncertain processing times
$\Lambda^T(G) \subseteq \Lambda^*(G)$	Minimal solution of the scheduling problem with uncertain processing times
$S^\Phi(p)$	Set of all optimal semiactive schedules with respect to criterion Φ
Γ	Set of competitive digraphs
γ	Number of competitive digraphs
$O_\varrho(p)$	Stability ball of the optimal schedule G_s with the radius ϱ and the center p
$\varrho_s(p)$	Stability radius of the optimal schedule G_s for an arbitrarily given regular criterion
$\widehat{\varrho}_s(p)$	Stability radius of the optimal schedule G_s for the makespan criterion
$\overline{\varrho}_s(p)$	Stability radius of the optimal schedule G_s for the mean flow time criterion
$\widehat{\varrho}_s^B(p \in T)$	Relative stability radius of the schedule G_s with respect to the polytope T for the makespan criterion
$\overline{\varrho}_s^B(p \in T)$	Relative stability radius of the schedule G_s with respect to the polytope T for the mean flow time criterion
$[\mu]$	Set of vertices (operations) which are contained in path μ
$l^p(\mu)$	Weight of path μ in the digraph with processing times $p \in R_+^q$
$\tilde{H}_k^i$	Set of paths in digraph G_k ending in the last operation of job J_i
H_k^i	Set of all dominant paths in $\tilde{H}_k^i$
$H_k^i(T)$	Subset of all dominant paths of set H_k^i with respect to the polytope T
$G_s \preceq_D G_k$	Dominance relation implying $\Phi_s^p \leq \Phi_k^p$ for each vector $p \in D$
$G_s \prec_D G_k$	Strong dominance relation implying $\Phi_s^p < \Phi_k^p$ for each vector $p \in D$

Index

Bibliography

[AG99] M.S. Akturk and E. Gorgulu. Match-up scheduling under a machine breakdown. *J. Oper. Res. Soc.*, 112:81–97, 1999.

[AGA98] A. Allahverdi, J.N.D. Gupta, and T. Aldowaisan. A review of scheduling research involving setup considerations. Department of Mechanical and Industrial Engineering Kuwait University, Safat, Kuwait, 1998.

[All95] A. Allahverdi. Two-stage production scheduling with separated setup times and stochastic breakdowns. *J. Oper. Res. Soc.*, 46:896–904, 1995.

[All97] A. Allahverdi. Scheduling in stochastic flowshop with independent setup, processing and removal times. *Comput. Oper. Res.*, 24:955–960, 1997.

[Alt00a] I. Althöfer. Decision support systems with multiple choice structure. In I. Althöfer, N. Cai, and et al., editors, *Numbers, Information and Complexity*, volume 16, pages 72–108. Kluwer, Amsterdam, 2000.

[Alt00b] I. Althöfer. K-alternative algorithms for sum type problems. In S. Dempe, H. Schreier, and I. Schiermeyer, editors, *14. Workshop on Discrete Optimization*, pages 5 – 8, Holzhau, Erzgebirge - Germany, May 23 -26, 2000. Technische Universität Bergakademie Freiberg.

[AM98] A. Allahverdi and J. Mittenthal. Dual criteria scheduling on a two machine flow shop subject to random breakdowns. *Internat. Transactions in Oper. Res.*, 5:317–324, 1998.

[BBMN91] J.C. Bean, J.R. Birge, J. Mittenehal, and C.E. Noon. Match-up scheduling with multiple resources, release dates and disruption. *Oper. Res.*, 39:470–483, 1991.

[BDP96] J. Blazewicz, W. Domschke, and E. Pesch. The job shop scheduling problem: Conventional and new solution techniques. *European J. Oper. Res.*, 93:1–33, 1996.

[BESW93] J. Blazewicz, K. Ecker, G. Schmidt, and J. Weglarz. *Scheduling in Computer and Manufacturing Systems.* Springer, Berlin, 1993.

[BHTW99a] H. Bräsel, M. Harboth, T. Tautenhahn, and P. Willenius. On the hardness of the classical job shop problem. *Ann. Oper. Res.*, 92:265–279, 1999.

[BHTW99b] H. Bräsel, M. Harboth, T. Tautenhahn, and P. Willenius. On the set of solutions of the open shop problem. *Ann. Oper. Res.*, 92:241–263, 1999.

[BJS94] P. Brucker, B. Jurisch, and B. Sievers. A fast branch and bound algorithm for the job shop scheduling problem. *Discrete Appl. Math.*, 49:107–127, 1994.

[BK96a] H. Bräsel and M. Kleinau. New steps in the amazing world of sequences and schedules. *Math. Methods Oper. Res.*, 43:195–214, 1996.

[BK96b] P. Brucker and A. Krämer. Polynomial algorithms for resource-constrained and multiprocessor task scheduling problems. *European J. Oper. Res.*, 90:214–226, 1996.

[BSW96] H. Bräsel, Yu.N. Sotskov, and F. Werner. Stability of a schedule minimizing mean flow time. *Math. Comput. Modelling*, 24:39–53, 1996.

[CCLe95] P. Chretienne, E.G. Coffman, J.K. Lenstra, and Z. Liu (editors). *Scheduling Theory and its Applications*. John Wiley & Sons, 1995.

[CDL94] G. Chrysslouris, K. Dicke, and M. Lee. An approach to real-time flexible scheduling. *Internat. J. Flexible Manufacturing Systems*, 6:235–253, 1994.

[CG00] C. Chu and V. Gordon. TWK due date determination and scheduling model: NP-hardness and polynomially solvable case. In J.-M. Proth and V. Tanaev, editors, *Proceedings of the International Workshop on Discrete Optimization Methods an Scheduling and Computer-aided Design*, pages 99–101, Minsk, Belarus, September 5-6, 2000.

[CP89] J. Carlier and E. Pinson. An algorithm for solving the job shop problem. *Management Sci.*, 35:164–176, 1989.

[CS98a] T.C.E. Cheng and N. Shakhlevich. Proportionate flow shop with controllable processing times. Working Paper No. 13/97-8, The Hong Kong Polytechnic University, 1998.

[CS98b] T.C.E. Cheng and N. Shakhlevich. Single machine scheduling of unit-time jobs with controllable release dates. Working Paper No. 01/98-9, The Hong Kong Polytechnic University, 1998.

[CT96] X. Cai and F.S. Tu. Scheduling jobs with random processing times on a single machine subject to stochastic breakdowns to minimize early-tardy penalties. *Naval Res. Logist.*, 43:1127–1146, 1996.

[CTU96] S. Chand, R. Traub, and R. Uzsoy. Single-machine scheduling with dynamic arrivals: Decomposition results and an improved algorithm. *Naval Res. Logist.*, 43:709–716, 1996.

[CV97] B. Chen and A.P.A. Vestjens. Scheduling on identical machines: How good is LPT in an on-line setting? *Oper. Res. Lett.*, 21:165–169, 1997.

[DHM96] R.L. Daniels, B.J. Hoopes, and J.B. Mazzola. Scheduling parallel manufacturing cells with resource flexibility. *Management Sci.*, 42:1260–1276, 1996.

[Dij59] E.W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[DJ88] W.J. Davis and A.T. Jones. A real-time production scheduler for a stochastic manufacturing environment. *Internat. J. Production Res.*, 1(2):101–112, 1988.

[DK95] R.L. Daniels and P. Kouvelis. Robust scheduling to hedge against processing time uncertainty in single stage production. *Management Sci.*, 41:363–376, 1995.

[DL82] V. Dumitru and F. Luban. Membership functions, some mathematical programming models and production scheduling. *Fuzzy Sets and Systems*, 8:19–33, 1982.

[FT63] H. Fisher and M.L. Thompson. Probabilistic learning combinations of local job-shop scheduling rules. In J.F. Muth and G.L. Thompson, editors, *Industrial Scheduling*, pages 225–251. Prentice-Hall, Englewood Cliffs, 1963.

[Gan19] H.L. Gantt. Efficiency and democracy. *Trans. Amer. Soc. Mech. Engrg.*, 40:799–808, 1919.

[GG94] B. Grabot and L. Geneste. Dispatching rules in scheduling: A fuzzy approach. *Internat. J. Production Res.*, 32(4):903–915, 1994.

[GL80] E.N. Gordeev and V.K. Leontev. Stability in bottleneck problems. *U.S.S.R. Comput. Maths. Math. Phys*, 20(4):275–280, 1980.

[GL85] E.N. Gordeev and V.K. Leontev. The complexity of the tabulation of trajectory problems. *U.S.S.R. Comput. Maths. Math. Phys*, 25(4):199–201, 1985.

[GLS83] E.N. Gordeev, V.K. Leontev, and I.Ch. Sigal. Computational algorithms for finding stability radius in choice problems. *U.S.S.R. Comput. Maths. Math. Phys.*, 23(4):128–132, 1983.

[Gor89] E.N. Gordeev. Solution stability of the shortest path problem. *Discrete Math.*, 1(3):45–56, 1989. (In Russian).

[Gra81] S.C. Graves. A review of production scheduling. *Oper. Res.*, 29(4):646–675, 1981.

[Gre97] H.J. Greenberg. An annotated bibliography for post-solution analysis in mixed integer programming and combinatorial optimization. University of Colorado at Denver, Mathematics Department, 1997.

[HZ97] O. Holthaus and H. Ziegler. Look ahead job demanding for improving job shop perfomance. *OR Spektrum*, 19:23–29, 1997.

[IMN87] H. Ishii, T. Masuda, and T. Nishida. Two machine mixed shop scheduling problems with controllable machine speeds. *Discrete Appl. Math.*, 17:29–38, 1987.

[IN86] H. Ishii and T. Nishida. Two machine open shop scheduling problem with controllable machine speeds. *J. Oper. Res. Soc. Japan*, 29:123–131, 1986.

[IT95] H. Ishii and M. Tada. Single machine scheduling problem with fuzzy precedence relation. *European J. Oper. Res.*, 87:284–288, 1995.

[Jan88] A. Janiak. General flow-shop scheduling with resource constraints. *Internat. J. Production Res.*, 26:125–138, 1988.

[JM99] A.S. Jain and S. Meeran. Deterministic job-shop scheduling: Past, present and future. *European J. Oper. Res.*, 113:390–434, 1999.

[KDV00] P. Kouvelis, R.L. Daniels, and G. Vairaktarakis. Robust scheduling of a two-machine flow shop with uncertain processing times. *IIE Transactions*, 32:421–432, 2000.

[KK00] G. Kyparisis and C. Koulamas. Open shop scheduling with makespan and total completion time criteria. *Comput. Oper. Res.*, 27:15–27, 2000.

[KKG92] P. Kouvelis, A.A. Kurawarwala, and G.J. Gutierrez. Algorithms for robust single- and multiple-period layout planning for manufacturing systems. *European J. Oper. Res.*, 63:287–303, 1992.

[KRKvHW94] A.W.H. Kolen, A.H.G. Rinnooy Kan, C.P.M. van Hoesel, and A.P.M. Wagelmans. Sensitivity analysis of list scheduling algorithms. *Discrete Appl. Math.*, 55:145–162, 1994.

[KSW95] S.A. Kravchenko, Yu.N. Sotskov, and F. Werner. Optimal schedules with infinitely large stability radius. *Optimization*, 33:271–280, 1995.

[KW96] M. Kuroda and Z. Wang. Fuzzy job shop scheduling. *Internat. J. Production Economics*, 44:45–51, 1996.

[LC95] W. Li and J. Cao. Stochastic scheduling on a single machine subject to multiple breakdowns according to different probabilities. *Oper. Res. Lett.*, 18:81–91, 1995.

[Leo75] V.K. Leontev. The stability of the traveling salesman problem. *U.S.S.R. Comput. Math. and Math. Phys.*, 15(5):199–213, 1975.

[Leo76] V.K. Leontev. Stability in combinatorial choice problems. *Soviet Math. Dokl.*, 17:635–638, 1976.

[Lib91] M. Libura. Sensitivity analysis for minimum Hamiltonian path and traveling salesman problems. *Discrete Appl. Math.*, 30:197–211, 1991.

[Lib99] M. Libura. On accuracy of solutions for discrete optimization problems with perturbed coefficients of the objective function. *Ann. Oper. Res.*, 86:53–62, 1999.

[LLLH00] H. Li, Z. Li, L.X. Li, and B. Hu. A production rescheduling expert simulation system. *European J. Oper. Res.*, 124:283–293, 2000.

[LLRKS93] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. Sequencing and scheduling: Algorithms and complexity. In S.S. Graves, A.H.G. Rinnooy Kan, and P. Zipkin, editors, *Handbooks in Operations Research and Management Science*, volume 4, pages 445–522. Logistics of Production and Inventory, North-Holland, New York, 1993.

[LS99] T.-C. Lai and Yu.N. Sotskov. Sequencing with uncertain numerical data for makespan minimization. *J. Oper. Res. Soc.*, 50:230–243, 1999.

[LSSW97] T.-C. Lai, Yu.N. Sotskov, N.Y. Sotskova, and F. Werner. **Optimal makespan scheduling with given bounds of processing times**. *Math. Comput. Modelling*, 26(3):67–86, 1997.

[LSSW98] T.-C. Lai, Yu.N. Sotskov, N.Y. Sotskova, and F. Werner. **Mean flow time minimization with uncertain processing times**. Otto-von-Guericke-Universität, FMA, Preprint No. 15/98, Magdeburg, Germany, 1998.

[LvdPSvdV96] M. Libura, E.S. van der Poort, G. Sierksma, and J.A.A. van der Veen. Sensitivity analysis based on k-best solutions of the traveling salesman problems. Technical Report 96A14, Research Institute Systems, Organizations and Management, University of Groningen, The Netherlands, Groningen, 1996.

[LWS94] V.J. Leon, S.D. Wu, and R.H. Storer. Robustness measures and robust scheduling for job shops. *IIE Transactions*, 26:32–43, 1994.

[Mel78] O. I. Melnikov. Optimal schedule stability for the Bellman-Johnson problem. *Vesti Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk.*, 6:99–101, 1978. (In Russian).

[ML93] B.L. MacCarthy and J. Liu. Addressing the gap in scheduling reserach: A review of optimization and heuristic methods in production research. *Internat. J. Production Res.*, 31:59–79, 1993.

[MVZ95] J.M. Mulvey, R.J. Vanderbei, and S.A. Zenios. Robust optimization of large-scale systems. *Oper. Res.*, 43:264–281, 1995.

[OD99] E.C. Özelkan and L. Duckstein. Optimal fuzzy counterparts of scheduling rules. *European J. Oper. Res.*, 113:593–609, 1999.

[PD91] H. Parunak and Van Dyke. Characterizing the manufacturing scheduling problem. *J. Manufacturing Systems*, 10(3):241–259, 1991.

[Pin95a] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1995.

[Pin95b] E. Pinson. The job shop scheduling problem: A concise survey and some recent development. In P. Chretienne, E.G. Coffman, J.K. Lenstra, and Z. Liu, editors, *Scheduling Theory and its Applications*, pages 277–293. John Wiley & Sons, 1995.

[PL94] N.M. Paz and W. Leigh. Maintance scheduling: Issues, results, and research needs. *Internat. J. Oper. and Production Management*, 47:47–69, 1994.

[PQ78] J. Picard and M. Queyranne. The time-dependent traveling salesman problem and its application to the tardiness problem in one machine scheduling. *Oper. Res.*, 26:86–110, 1978.

[PZ76] C.J. Piper and A.A. Zoltners. Some easy postoptimality analysis for zero-one programming. *Management Sci.*, 22(7):759–765, 1976.

[RC95] R. Ramaswamy and N. Chakravarti. Complexity of determining exact tolerances for min-sum and min-max combinatorial optimization problems. Technical Report WPS-247/95, Indian Institute of Management, Calcutta, 1995.

[RL87] J.J. Rosenblatt and H.L. Lee. A robustness approach to facilities design. *Internat. J. Production Res.*, 25:479–486, 1987.

[RR96] D.J. Robb and T.R. Rohleder. An evaluation of scheduling heuristics for dynamic single-processor scheduling with early/tardy costs. *Naval Res. Logist.*, 43:349–364, 1996.

[RS64] B. Roy and B. Sussmann. Les problèmes d'ordonnancement avec contraintes disjonctives. Note DS No. 9 bis, SEMA, Montrouge, 1964.

[San95] E. Sanlaville. Nearly on line scheduling of preemptive independent tasks. *Discrete Appl. Math.*, 57:229–241, 1995.

[Sch00] G. Schmidt. Scheduling with limited machine availability. *European J. Oper. Res.*, 121:1–15, 2000.

[Sei98] S.S. Seiden. Randomized online interval scheduling. *Oper. Res. Lett.*, 22:171–177, 1998.

[Sga98] J. Sgall. On-line scheduling. In A. Fiat and G.J. Woeginger, editors, *On-Line Algorithms, Lecture Notes in Computer Science*. Springer, Berlin, 1998.

[SH99] R. Slowinski and M. Hapke. *Scheduling Under Fuzziness*. Physica-Verlag, Heidelberg, New York, 1999.

[SJ97] C. Stein and W. Joel. On the existence of schedules that are near-optimal for both makespan and total weighted completion time. *Oper. Res. Lett.*, 21:115–122, 1997.

[SK94] E. Szelke and R.M. Kerr. Knowledge-based reactive scheduling. *Production Planning and Control*, 5(2):124–145, 1994.

[SK00] M. Sakawa and R. Kubota. Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. *European J. Oper. Res.*, 120:393–407, 2000.

[SLG95] Yu.N. Sotskov, V.N. Leontev, and E.N. Gordeev. Some concepts of stability analysis in combinatorial optimization. *Discrete Appl. Math.*, 58:169–190, 1995.

[SOM+90] S.F. Smith, P.S. Ow, N. Muscettola, J. Potvin, and D.C. Matthy. An integrated framework for generating and revising factory schedules. *J. Oper. Res. Soc.*, 41(6):539–552, 1990.

[Sot91] Yu.N. Sotskov. Stability of an optimal schedule. *European J. Oper. Res.*, 55:91–102, 1991.

[Sotskova99a] N.Y. Sotskova. **Optimal makespan schedules for a job-shop with uncertain processing times**. In M.F. Carvalho and F.M. Müller, editors, *Proceedings of the 15th International Conference on CAD/CAM, Robotics & Factories of the Future CARS & FOF99*, volume 1, pages MW4:7 – MW4:12, Águas de Lindóia - Brazil, August 18-20, 1999. Technological Center for Informatics Foundation Automation Institute, Campinas-SP-Brazil.

[Sotskova99b] N.Y. Sotskova. **Optimization in a job shop with given bounds of processing times**. In J. Soldek and J. Pejas, editors, *Proceedings of the Sixth International Conference on Advanced Computer Systems ACS'99*, pages 462–467, Szczecin - Poland, November 18-19, 1999. Technical University of Szczecin.

[Sotskova99c] N.Y. Sotskova. **Solution of an uncertain job-shop problem with regular criterion**. In *Book of Abstracts. Symposium on Operations Research SOR'99*, page 103, Magdeburg - Germany, September 1 - 3, 1999. Otto-von-Guericke-Universität Magdeburg.

[Sotskova00] N.Y. Sotskova. **Two-stage scheduling optimization of requirements maintenance with different routes**. In *Proceedings of the Eighth Belarussian Mathematical Conference*, page 24, Minsk, Belarus, June 19-24, 2000. (In Russian).

[SSW97] Yu.N. Sotskov, N.Y. Sotskova, and F. Werner. **Stability of an optimal schedule in a job shop**. *OMEGA - Internat. J. Management Sci.*, 25(4):397–414, 1997.

[ST98] N.Y. Sotskova and V.S. Tanaev. **About the realization of an optimal schedule with operation processing times under conditions of uncertainty**. *Dokl. Natsionalnoi Akad. Nauk Belarusi*, 42(5):8–12, 1998. (In Russian).

[Str95] V.A. Strusevich. Two machine flow shop scheduling problem with no wait in process: Controllable machine speeds. *Discrete Appl. Math.*, 59:75–86, 1995.

[Str99] V.A. Strusevich. A heuristic for the two-machine open-shop scheduling problem with transportation times. *Discrete Appl. Math.*, 93:287–304, 1999.

[STW98] Yu.N. Sotskov, V.S. Tanaev, and F. Werner. Stability radius of an optimal schedule: A survey and recent developments. In G. Yu, editor, *Industrial Applications of Combinatorial Optimization*, volume 16, pages 72–108. Kluwer Academic Publishers, Boston, MA, 1998.

[Sus72] B. Sussmann. Scheduling problems with interval disjunctions. *J. Oper. Res.*, 16:165–178, 1972.

[SW80] D.R. Shier and G. Witzgall. Arc tolerances in shortest path and network flow problems. *Networks*, 10:277–291, 1980.

[SW00] N.Y. Sotskova and F. Werner. **Dominance relations on the set of schedules for uncertain job-shop**. In J.-M. Proth and V. Tanaev, editors, *Proceedings of the International Workshop on Discrete Optimization Methods an Scheduling and Computer-aided Design*, pages 99–101, Minsk, Belarus, September 5-6, 2000.

[SWW98] Yu.N. Sotskov, A.P.M. Wagelmans, and F. Werner. On the calculation of the stability radius of an optimal or an approximate schedule. *Ann. Oper. Res.*, 83:213–252, 1998.

[Tar82] R.E. Tarjan. Sensitivity analysis of minimum spanning trees and shortest path trees. *Inform. Process. Lett.*, 14:30–33, 1982.

[Tri94] M.A Trick. Scheduling multiple variable-speed machines. *Oper. Res.*, 42:234–248, 1994.

[TSS94] V.S. Tanaev, Yu.N. Sotskov, and V.A. Strusevich. *Scheduling Theory: Multi-Stage Systems*. Kluwer Academic Publishers, The Netherlands, Dordrecht, 1994.

[vdP97] E.S. van der Poort. *Aspects of Sensitivity Analysis for the Traveling Salesman Problem*. PhD thesis, Graduate School / Research Institute Systems, Organizations and Management, University of Groningen, The Netherlands, Groningen, 1997.

[Wag95] H.M. Wagner. Global sensitivity analysis. *Oper. Res.*, 43:948–969, 1995.

[WBS99] S.D. Wu, E.-S. Byeon, and R.H. Storer. A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. *Oper. Res.*, 47(1):113–124, 1999.

[WJ88] G.R. Wilson and H.K. Jain. An approach to postoptimality and sensitivity analysis of zero-one goal programs. *Naval Res. Logist.*, 35:73–84, 1988.

Zusammenfassung

Scheduling ist ein bedeutender Bestandteil des Decision-Making in Unternehmen vieler Industriezweige. Jedoch ist die Nutzung verschiedener Resultate der klassischen Scheduling-Theorie bei praktischen Problemstellungen aus der Industrie bisher sehr begrenzt. Insbesondere schränkt die in der Scheduling-Theorie übliche Voraussetzung, da die Bearbeitungszeiten der Operationen vor Nutzung eines Scheduling-Algorithmus bekannt sind, die praktische Anwendung ein. Diese Dissertation ist ein Versuch, einige theoretische Resultate für breitere Anwendungen nutzbar zu machen.

Neben den üblichen Voraussetzungen, da die strukturellen Daten fixiert sind und da zu jeder Zeit eine Maschine nur einen Auftrag bearbeiten kann und jeder Auftrag auf höchstens einer Maschine bearbeitet werden kann, wird in dieser Dissertation die folgende Annahme getroffen: Die tatsächliche Bearbeitungszeit p_i einer Operation i kann jeden reellen Wert zwischen einer gegebenen unteren Schranke a_i und einer gegebenen oberen Schranke b_i annehmen. Ein Job-Shop Problem mit derartigen Unsicherheiten in den Eingangsdaten wird im Fall der Minimierung der Gesamtbearbeitungszeit (Makespan) mit $\mathcal{J}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ bezeichnet. Gilt für jede Operation $i \in Q = \{1, 2, \ldots, q\}$ eines Problems $\mathcal{J}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ die Beziehung $a_i = b_i$, so liegt ein deterministisches Job-Shop Problem vor, andernfalls ein stochastisches Problem ohne a priori Information über die Verteilungen der Zufallsgröen der Bearbeitungszeiten.

Für Problem $\mathcal{J}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ mu nicht notwendig ein eindeutig bestimmter Plan existieren, der für alle möglichen Realisierungen der Bearbeitungszeiten optimal bleibt. Daher mu für ein Problem mit unsicheren Eingangsdaten eine Menge semiaktiver Pläne konstruiert werden, die die Menge der restlichen Pläne für das gegebene Zielkriterium dominiert. Um eine solche Menge von Plänen zu bestimmen, wird eine Stabilitätsanalyse eines optimalen Planes bezüglich der möglichen Störungen der Bearbeitungszeiten durchgeführt. Es wird der Begriff des relativen Stabilitätsradius eines optimalen Plans eingeführt. Dabei wird die Relativität bezüglich des Polytops $T = \{p' = (p'_1, p'_2, \ldots, p'_q) : a_i \leq p'_i \leq b_i, i \in Q\}$ der zulässigen Vektoren p' der Bearbeitungszeiten und bezüglich der Teilmenge B von semiaktiven Plänen, für die die Überlegenheit eines verfügbaren Planes garantiert werden mu, betrachtet. In der Arbeit wird das gemischte (disjunktive) Graphenmodell benutzt, welches für den gesamten Scheduling Proze von dem gemischten Ausgangsgraphen bis zum Graphen G_s, der einen vollständigen semiaktiven Plan s repräsentiert, geeignet ist.

Die formale Definition des relativen Stabilitätsradius ist wie folgt. Sei $O_\varrho(p)$ eine abgeschlossene Kugel im Raum R^q der q-dimensionalen reellen Vektoren. Angenommen, der Plan $s \in B$ habe für jeden Vektor $p' \in O_\varrho(p) \cup T$ von Bearbeitungszeiten eine minimale Gesamtbearbeitungszeit (Makespan). Dann wird der maximale Wert des Radius ϱ einer solchen Kugel als relativer Stabilitätsradius bezeichnet und mit $\varrho_s^B(p \in T)$ abgekürzt. In der Dissertation werden notwendige und hinreichende Bedingungen für den Fall $\varrho_s^B(p \in$

$T) = 0$ und für den Fall eines unendlich groen Wertes von $\varrho_s^B(p \in T)$ abgeleitet.

Die abgeleitete Formel für die Berechnung des exakten Wertes von $\varrho_s^B(p \in T)$ basiert of dem Vergleich des Planes s mit anderen zulässigen Plänen von der Menge B, und es wird gezeigt, wie man die Anzahl der zu betrachtenden Digraphen der Menge B bei diesem Vergleich reduzieren kann. hnliche Resultate wie zuvor für das Makespan-Kriterium beschrieben wurden für den Fall der Minimierung der Summe der Bearbeitungsendtermine der Aufträge (Mean Flow Time) erhalten. Insbesondere wurden notwendige und hinreichende Bedingungen für einen unendlich groen Wert von $\varrho_s^B(p \in T)$ im Fall des Mean-Flow-Time Kriteriums abgeleitet.

Auf der Grundlage dieser Resultate wurden exakte und heuristische Algorithmen zur Lösung der Job-Shop Probleme $\mathcal{J}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ und $\mathcal{J}/a_i \leq p_i \leq b_i/\sum \mathcal{C}_i$ und des General-Shop Problems $\mathcal{G}/a_i \leq p_i \leq b_i/\mathcal{C}_{max}$ abgeleitet. Die entwickelte Software wurde an zufällig erzeugten Job-Shop Problemen getestet.

Curriculum Vitae

Nadezhda Sotskova was born in May 21, 1975, in Minsk (Belarus). From 1982 to 1992 she studied in the Secondary School in Minsk (mathematical class). From 1992 to 1997 she studied mathematics at the Belarussian State University (Department of Applied Mathematics and Computer Science). She graduated in June 1997, on the subject of economical cybernetics, and got qualification of a mathematician-economist and qualification of a teacher of mathematics and computer science. The title of her Master's thesis was "The experimental investigations of the stability radius of the optimal graphs (schedules)". In January 1998 she started as a PhD student at the University of Magdeburg (Department of Mathematics, Institute of Mathematical Optimization).