Structural Properties of Preemptive Schedules

Ph. Baptiste* J. Carlier† A. Kononov‡ M. Queyranne§
S. Sevastyanov¶ M. Sviridenko‖

Abstract

Consider some *preemptive* scheduling problem, *i.e.*, a problem where we can interrupt operations and resume them later without any penalties. What can we say about the optimal schedule? When does the optimal schedule exist? When does it have a limited (polynomial) number of interruptions? Do they occur at integer time points? Such theoretical questions are also of practical interest since structural properties can be used to reduce the search space in a practical scheduling application. In this paper we answer these questions for a large variety of scheduling models, including parallel machine scheduling, shop scheduling and resource constrained project scheduling.

1 Introduction

Motivation. Consider some *preemptive* scheduling problem, *i.e.*, a problem where we can interrupt operations and resume them later without any penalties. What can we say about the optimal schedule for such a problem? When does it exist? When does it admit a finite (polynomial) number of interruptions? When can we reduce all possible interruptions to only those happening at integral (or rational) points in time? Such questions are investigated in this paper for a large variety of scheduling models and objective functions. It can be shown that in some cases we cannot guarantee even the existence of an optimal schedule even if the set of feasible schedules is nonempty. As a simple example, let us consider a single-machine instance with one job having unit processing time $p_1 = 1$; the penalty function $f_1(C_1)$ depending on the completion time C_1 of job J_1 is defined as:

$$f_1(x) = \begin{cases} x, & \text{for } x \leq 1; \\ x-1, & \text{for } x > 1. \end{cases}$$

*CNRS, Heudiasyc, Univ. de Tech. de Compiègne & École Polytechnique; Email: baptiste@utc.fr
†CNRS, Heudiasyc, Univ. de Tech. de Compiègne; Email: carlier@utc.fr
‡Sobolev Institute of Mathematics, Novosibirsk, Russia; Email: alvenko@math.nsc.ru
§Faculty of Commerce and Business Administration, University of British Columbia, Vancouver, B.C., Canada V6T 1Z2; Email: Maurice.Queyranne@commerce.ubc.ca; Supported by a research grant from the Natural Sciences and Research Council of Canada
¶Sobolev Institute of Mathematics, Novosibirsk, Russia; Email: seva@math.nsc.ru; Supported by the Russian Foundation for Basic Research (grant no. 05-01-00960)
‖IBM T.J. Watson Research Center, Yorktown Heights, USA; Email: sviri@us.ibm.com

The infimum of the penalty function over all feasible schedules is equal to 0, but it can not be attained by any feasible schedule — neither non-preemptive, nor preemptive.

As the reader might conclude from this example, the trouble with the above example is an objective function that is not nondecreasing. Consider another scheduling problem that can be shortly written as $Q2|p_j = 1, pmtn|\sum f_j(C_j)$ in standard scheduling notation [11]. In this problem we are given two jobs J_1 and J_2 with unit processing times that should be scheduled preemptively on two uniform machines M_1 and M_2 having speeds $s_1 = 1$ and $s_2 = 2$. The goal is to minimize an additive function $F(C_1, C_2) = f_1(C_1) + f_2(C_2)$, where

$$f_1(x) = 2x; \;\; f_2(x) = \begin{cases} x, & \text{for } x < 3/4; \\ x + 1, & \text{for } x \geq 3/4. \end{cases}$$

As one can see, the objective function is now nondecreasing, and the infimum of its values over all feasible preemptive schedules is equal to 7/4. Yet it still cannot be attained.

This time our failure in getting the optimum may be associated with the objective function being not continuous. Still it would be interesting to get an answer to the question: what are the minimal conditions that should be imposed on the objective function of a scheduling problem with preemptions allowed, so as to guarantee the existence of at least one optimal solution? Of course, it is hardly possible to derive such minimal conditions that would fit in any case, to ALL scheduling problems. However, as we prove in Section 2.4, some minimal universal requirements to the objective functions exist that for "almost all" scheduling problems can guarantee the existence of their optimal solutions. To be exact, it suffices for the objective function to be nondecreasing and continuous from the left.

In other situations the optimal value of an objective function can be only attained by schedules with an infinite number of interruptions, although there are feasible schedules without interruptions. An instance of such a problem is presented on page 10.

At the same time, we show that in many situations it is possible to guarantee the existence of an optimal schedule that satisfies some nice properties. For example, the classical preemptive job shop scheduling problem with integral data and the minimum makespan objective has an optimal schedule where all interruptions occur at integral dates; moreover, preemption only occurs when some operations complete their processing. On the other hand, for a natural extension of this problem, namely, the preemptive *dag shop* scheduling problem (where the precedence constraints between operations of the same job need not form a chain) these properties do not hold (an example is given in Section 4.2 and is pictured in Figure 1).

Our Results. We first obtain some general results for preemptive scheduling problems concerning the existence of an optimal schedule, and the existence of an optimal schedule with a finite number of interruptions. Such results are established for a wide class of preemptive scheduling models including most classical, as well as non-traditional machine scheduling models and project scheduling models with constrained resources and a large variety of objective functions including all classical ones. We next show that certain classical shop and parallel machine scheduling problems with integral data satisfy the following *Strong Structural* property: there exists an optimal preemptive schedule where all interruptions occur at *integral* dates. We establish this property for the preemptive versions of the parallel machine scheduling with chain precedence constraints and release dates, for the job shop and two-machine dag shop scheduling problems under a general classes of objective functions.

We also show that for the preemptive three-machine dag shop scheduling problem with two operations per job and the minimum makespan objective this property does not hold. Besides that, for the job shop problem we prove the existence of an optimal schedule in which each preemption of an operation occurs only when another operation is completed.

A straightforward consequence of our results is the so-called preemption redundancy for a wide class of scheduling problems with unit processing times. This preemption redundancy implies that all such preemptive scheduling problems are equivalent to their non-preemptive counterparts from complexity viewpoints. This equivalence provides us with new and sometimes simpler proofs of existing complexity results. E.g., our results imply NP-hardness of $Pm|p_j = 1, chains, pmtn|\sum w_j C_j$ and $Pm|p_j = 1, chains, pmtn|\sum U_j$ previously published in [1, 5, 19]. Our results also imply the existence of polynomial time algorithm for $P|p_j = 1, r_j, pmtn|\sum w_j T_j$ and NP-hardness for $Pm|p_j = 1, chains, pmtn|\sum T_j$ for any m, — both problems were open before.

Finally, for the most general scheduling model and quite general objective functions we prove two *Rational Structure Theorems* (Theorems 5.1 and 5.2) which state that for any instance of the problem there exists an optimal schedule with the following properties:
(1) the total number of interruptions grows polynomially with the number of operations and the number of fixed dates;
(2) all operation start times and completion times and all interruptions occur at integer multiples of a rational number $\delta > 0$ with size polynomially bounded in the input size;
(3) the optimal value of the objective function is an integer multiple of δ, where the size of the integer multiplier is also polynomially bounded in the input size.

An important consequence of those Rational Structure Theorems is that the decision version of preemptive scheduling problems under consideration are in the class $\mathcal{NP}$. Indeed, for any feasible instance and a specified target objective value, there exists a concise certificate of polynomial size which is provided by a feasible schedule satisfying properties (1)–(3).

Related Results. There are few systematic studies of such structural questions in the preemptive scheduling literature, and most known results follow from either (i) the fact that there is no advantage to preemption [2, 3], or (ii) the existence or properties of polynomial time algorithms. Results following from (i) are clearly the strongest type of structural results one could hope for for scheduling problems. We refer to the standard scheduling literature (e.g., [11]) for many such classical results; another extensive reference is the book by Tanaev, Gordon and Shafransky [18].

Structural results following from (ii), the existence or properties of polynomial time algorithms, have been obtained mostly for parallel machine and open shop problems. We use the standard three-field notation [11] to describe such scheduling problems. McNaughton [12] constructs an optimal schedule with at most $m-1$ interruptions for the problem $P|pmtn|C_{\max}$ on m identical parallel machines and makespan objective. Gonzalez and Sahni [7] construct an optimum schedule with at most $2(m-1)$ interruptions for the uniform parallel machine version $Q|pmtn|C_{\max}$ of this problem. The bounds of McNaughton and of Gonzalez and Sahni on the number of interruptions (and preemption dates) are tight. Labetoulle et al. [9] prove that the natural greedy algorithm for the problem $Q|r_j, pmtn|\sum C_j$ with m machines and n jobs finds an optimal solution with at most $2n - m$ interruptions. For the unrelated parallel machine problem $R|pmtn|C_{\max}$, Lawler et al. [11] state that a procedure of Lawler and Labetoulle [10] can be modified to yield an optimal schedule with no more than $7m^2/2$ interruptions. Turning now to open shop problems, Gonzalez and Sahni [6] construct an optimal schedule for the problem $O|pmtn|C_{max}$ with m machines, n jobs and ξ operations, which has at most $\xi + n + m$ preemption dates. Du and Leung [4] proved the corresponding result for $O2|pmtn|\sum_j C_j$. Little attention seems to have been given in the literature to structural properties for other preemptive shop scheduling problems.

Some general structural results were obtained by Sauer and Stone [16] (see also [15]). They proved that for the parallel machine scheduling problem with n jobs, precedence constraints, unit processing times and the minimum makespan objective there is an optimal preemptive schedule with at most $n-1$ preemption dates. They also proved that the length of the time interval between two such consecutive points is equal to a rational number and gave a lower bound on its value. We extend the technique used in [16] to prove our general Rational Structure Theorem for a much more general class of scheduling problems and objective functions.

Paper Outline. In Section 2 we give definitions of basic notions, formulate the most general problem and prove most general results about the existence of an optimal schedule for a given preemptive problem and the finiteness of the number of interruptions in an optimal schedule. In Sections 3 and 4 some strong structural properties are shown for parallel machine and shop scheduling problems respectively. Section 5 is dedicated to weaker but more general Rational Structure Theorems. Finally, in Section 6 we discuss our results and propose some open questions.

2 Definitions and general results

In this section we first introduce some necessary notions and give some definitions. The most general problem **GP** will be formulated, for which the *Finiteness Theorem* will be proved. According to this theorem, every time when an optimal schedule exists, there exists such a schedule with a finite (and polynomially bounded in the input size) number of interruptions. The *Optimum Existence Theorem* establishes the existence of an optimal schedule for a given instance of the **GP** problem with an arbitrary *regular* objective function, provided that the set of feasible schedules is nonempty.

2.1 Definitions

For two vectors $x' = (x'_1, \ldots, x'_n)$ and $x'' = (x''_1, \ldots, x''_n)$ we write $x' \leq x''$, if the inequality $x'_i \leq x''_i$ holds for each component i.

Definition 1 We say that a function $F(x)$ $(x \in \mathbb{R}^n)$ defined on a domain $D \subseteq \mathbb{R}^n$ is *nondecreasing* if $F(x') \leq F(x'')$ holds for any pair of vectors $x', x'' \in D$ such that $x' \leq x''$.

Definition 2 We say that a function $F(x)$ $(x \in \mathbb{R}^n)$ defined on a domain $D \subseteq \mathbb{R}^n$ is *continuous from the left*, if for any point $x \in D$ and any $\varepsilon > 0$ there exists a number $\delta > 0$ such that the inequality $|F(x) - F(x')| < \varepsilon$ holds for every $x' \in D$ such that $x' \leq x$ and $x_i - x'_i \leq \delta$, $\forall\ i = 1, \ldots, n$.

Definition 3 A real valued function $F(x)$ ($x \in \mathbb{R}^n$) is called *regular*, if it is nondecreasing and continuous from the left.

Definition 4 We say that a function $F(x)$ ($x \in \mathbb{R}^n$) is *integer-concave*, if the inequality

$$F(\lambda x' + (1-\lambda)x'') \geq \lambda F(x') + (1-\lambda)F(x'')$$

holds for any $\lambda \in [0,1]$ and any $x' = (x'_1, \dots, x'_n)$, $x'' = (x''_1, \dots, x''_n)$ such that $x'_i, x''_i \in [t_i, t_i + 1]$ for some integer t_i, $i = 1, \dots, n$.

For instance, tardiness is a classical example of a penalty function (of either operation or job completion times) which is integer-concave for integral due dates, but not concave. Consequently, the total weighted (but not the maximum!) tardiness is integer-concave. Of course, any linear, e.g. the total weighted completion time, or just concave function is integer-concave, as well.

On the other hand, such a classical objective function as the makespan is not integer-concave. Indeed, let us consider the example with 2 jobs and $F(C_1, C_2) = \max\{C_1, C_2\} \doteq C_{\max}$, where C_i denotes the completion time of job i. Let $x' = (C'_1, C'_2) = (1, 0)$; $x'' = (C''_1, C''_2) = (0, 1)$. Then for the point $x = \frac{1}{2}x' + \frac{1}{2}x''$ we have $F(x) = \max\{0.5, 0.5\} = 0.5$, whereas $\frac{1}{2}F(x') + \frac{1}{2}F(x'') = 1$, and the required inequality is failed.

Definition 5 Let $D \subseteq \mathbb{R}^n$ be a convex domain in $\mathbb{R}^n$. We say that a function $F(x)$ defined on $x \in D$ is *quasiconcave*, if for any $x', x'' \in D$ and any $\lambda \in [0,1]$ the inequality $F(\lambda x' + (1 - \lambda)x'') \geq \min\{F(x'), F(x'')\}$ holds.

Simple examples of such functions are:

1. concave functions;
2. any nondecreasing function $F(x)$ of a one-dimensional variable x.

The latter example admits extensions to the n-dimensional case, yet not that trivial.

2.2 Problem formulation

Let $\mathcal{O} = \{o_1, \dots, o_\xi\}$ be a finite set of *operations* to be processed. Each operation $o_j \in \mathcal{O}$ has a specified *processing requirement* $p_j \geq 0$ and may be processed in a number of different *modes* (specifically: on different machines, with different speeds, consuming different amounts of resources, *etc.*). Let $\mathcal{M}(o_j)$ be the finite set of all possible modes for operation $o_j \in \mathcal{O}$. With each mode $\mu \in \mathcal{M}(o_j)$ a *speed* $s_\mu > 0$ is associated such that it takes p_j/s_μ time units if operation o_j is processed entirely using mode μ. We assume that an operation may use several modes in parallel (e.g., different machines, resources, tools, etc.). Furthermore, for the sake of generality we assume that the modes belonging to different operations should be treated, in general, as different. This enables us to model a much wider variety of compatibility constraints, taking into account the whole specificity of each operation (see the examples on page 8 for illustration).

Let $\mathcal{M} = \cup_{o_j \in \mathcal{O}} \mathcal{M}(o_j)$ be the set of all modes. As follows from the above assumption, all mode sets $\mathcal{M}(o_j)$ are disjoint, i.e. $\mathcal{M}(o_i) \cap \mathcal{M}(o_j) = \varnothing$ for $o_i \neq o_j$, which enables us, given a mode $\mu \in \mathcal{M}$, to identify uniquely, to which operation $\omega(\mu) \in \mathcal{O}$ it belongs.

We next assume that a finite sequence $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_D$ of *fixed dates* τ_k (independent of a schedule) is defined such that the processing of all given operations must occur during the time interval $[\tau_1, \tau_D]$. Thus, τ_1 may be the earliest release date of an operation or a job, or the earliest availability date of a resource; in many situations we may have $\tau_1 = 0$. The last date τ_D may be any finite upper bound on the completion time of the last operation in any "reasonable" (or optimal) schedule; such an upper bound is usually easy to derive from the problem data. A few typical examples of fixed dates can be found on page 8. The situation $\tau_k = \tau_{k+1}$ may occur, when we have got a zero length operation whose release date and deadline coincide.

We consider three types of constraints: *compatibility constraints* between modes, *precedence constraints* between operations, and a *full execution requirement.*

Compatibility constraints are defined for each time interval $[\tau_k, \tau_{k+1}]$ $(k = 1, \ldots, D-1)$ by associated families $\mathcal{P}(k) \subseteq 2^{\mathcal{M}}$ of feasible subsets of modes $P \subseteq \mathcal{M}$ (*feasible patterns*) that may be simultaneously used at any time $t \in [\tau_k, \tau_{k+1}]$. In our model they play a role of the main components of any feasible schedule. In most situations, $\mathcal{P}(k)$ are closed under inclusion, that is, $P \subset P'$ and $P' \in \mathcal{P}(k)$ imply $P \in \mathcal{P}(k)$ — unless technological constraints require the simultaneous processing of certain modes. Similarly, we allow $\varnothing \in \mathcal{P}(k)$, whereby all operations and resources may be *idle.*

Now we need to introduce the notion of a *slice*, as an elementary item of any feasible schedule.

Definition 6 A partial schedule $\hat{S}$ is called a *slice*, if it is defined in a time interval $[t', t''] \subseteq [\tau_1, \tau_D]$ so that only one (maybe, empty) pattern P is used throughout this interval. In this case a slice is called a *P-slice.* In other words, each slice can be specified by (and will be associated with) a triplet $(t', t''; P)$. A slice $(t', t''; P)$ is called *feasible*, if $[t', t''] \subseteq [\tau_k, \tau_{k+1}]$, $P \in \mathcal{P}(k)$ for some $k \in \{1, \ldots, D-1\}$. A positive length slice will be simply called a *positive slice.*

Definition 7 Let $\mathcal{I} = \{1, 2, \ldots\}$ be a countable (finite or infinite) *set of indices.* A family of feasible slices $\{\hat{S}_i = (t'_i, t''_i; P_i) \,|\, i \in \mathcal{I}\}$ is called a *full schedule* for a given instance of problem **GP**, if:
(a) each point $t \in [\tau_1, \tau_D]$ is covered by at least one slice;
(b) any two intervals $[t'_i, t''_i], [t'_j, t''_j]$ $(i, j \in \mathcal{I}; i \neq j)$ may overlap in at most one point which can only be a boundary point for both intervals.
The end-points t''_i of slices will be further referred to as *changeover dates.*

It follows from (b) that the slices follow each other, in the sense that for any two slices $\hat{S}_i$ and $\hat{S}_j$ of a full schedule S we can determine which slice goes first. Namely, we set $\hat{S}_i \preceq \hat{S}_j$, if $t''_i \leq t'_j$. (For zero-length slices $\hat{S}_i$ and $\hat{S}_j$ assigned to the same time point t we have simultaneously $\hat{S}_i \preceq \hat{S}_j$ and $\hat{S}_j \preceq \hat{S}_i$.) However, this does not mean that we can always renumber the slices $\{\hat{S}_i \,|\, i \in \mathcal{I}\}$ so as to have $\hat{S}_i \preceq \hat{S}_j$ for any $i < j$. For instance, such a numbering is impossible, if there is a point $t \in [\tau_1, \tau_D)$ such that each of its neighborhoods contains an infinite number of slices.

Full execution requirement. Suppose, we are given a full preemptive schedule $S = \{(t'_i, t''_i; P_i) \,|\, i \in \mathcal{I}\}$, and let $s_{ji} \doteq \sum_{\mu \in P_i \cap \mathcal{M}(o_j)} s_\mu$ stand for the total speed of processing an operation o_j in a pattern P_i. Then we should have

$$\sum_{i \in \mathcal{I}} s_{ji}(t''_i - t'_i) = p_j, \ \forall \ o_j \in \mathcal{O}.$$

Furthermore, for any operation o_j with $p_j = 0$ there must be at least one slice $\hat{S}_i \in S$ such that $P_i \cap \mathcal{M}(o_j) \neq \varnothing$.

Definition 8 The *completion* and the *starting time* of an operation $o_j \in \mathcal{O}$ in a preemptive schedule $S = \{(t'_i, t''_i; P_i) \,|\, i \in \mathcal{I}\}$ are defined as $C_j = \sup\{t''_i \,|\, i \in \mathcal{I},\ P_i \cap \mathcal{M}(o_j) \neq \varnothing\}$ and $S_j = \inf\{t'_i \,|\, i \in \mathcal{I},\ P_i \cap \mathcal{M}(o_j) \neq \varnothing\}$ respectively.

Precedence constraints on the set of operations $\mathcal{O}$ can be specified by a directed weighted graph $G = (\mathcal{O}, U)$ whose vertices are operations of the given instance, and to each arc $(o_i, o_j) \in U$ a weight $p_{ij} \in \mathbb{R}$ is assigned (which may be negative). Each arc $(o_i, o_j) \in U$ imposes the constraint

$$C_i + p_{ij} \leq S_j. \tag{1}$$

When all weights $p_{ij} \geq 0$, they are usually called *precedence delays* in the scheduling literature. Yet we will also consider the cases that allow negative p_{ij}.

Let now $F(x_1, \dots, x_\xi)$ be a function defined in every point of the ξ-dimensional domain $[\tau_1, \tau_D]^\xi$. The goal of the **GP**-problem consists in constructing a full schedule S that is feasible with respect to the three above constraints and minimizing the function $F(C_1, \dots, C_\xi)$ of the operation completion times. If $F(C_1, \dots, C_\xi)$ is a regular function, then we say that problem **GP** has a *regular criterion*.

Summing up the above said, the **GP**-problem is the problem of minimizing a function $F(C_1, \dots, C_\xi)$ of the operation completion times over all preemptive schedules $S = \{(t'_i, t''_i; P_i) \,|\, i \in \mathcal{I}\}$ satisfying the full execution requirement, some compatibility constraints and some precedence constraints.

We should make here an important note on the usage of zero-length slices in schedules.

Note 2.1 In our model we admit existence of zero-length operations. This enables one to model more general situations in a uniform style, and also provides convenient tool to specify some complicated constraints via an introduction of a few zero-length (dummy) operations. As a price for this convenience, we have to admit the existence of zero-length slices, as well as possible existence in schedule S of several such slices $(\hat{S}_i, \hat{S}_j, \dots)$ assigned to the same point in time $(t''_i = t'_i = t''_j = t'_j = \dots)$. The latter assumption is a natural extension of the agreement commonly used in machine scheduling that several zero-length operations can be processed **on the same machine at the same point in time**. (Despite the admitted constraint that **at most one operation** is allowed to be processed on each machine at any point in time.) Furthermore, once preemption is allowed, a zero-length slice need not contain a zero-length operation, but instead may consist of zero-length pieces of operations.

This enables us to place a zero-length slice with a given operation o (*i.e.*, with a pattern P containing a mode $\mu \in \mathcal{M}(o)$) at any point t where the pattern P is feasible. Especially, such a trick can help, when we try to attain a desirable completion time of an operation provided that some capacity constraints do not prevent this.

Finally, we give a strict definition of the notion of *interruption*.

Definition 9 We say that an operation $o_j \in \mathcal{O}$ has an *interruption* at time t in a preemptive schedule $S = \{(t'_i, t''_i; P_i) \mid i \in \mathcal{I}\}$, if

(a) t is the endpoint of a slice where the operation is processed; formally, there exists $i \in \mathcal{I}$ such that $t''_i = t$ and $P_i \cap \mathcal{M}(o_j) \neq \varnothing$;

(b) arbitrarily close to t, there must be a slice where the operation is processed in another set of modes or is not processed at all. Formally, for any $\varepsilon > 0$ there exists an index $i' \in \mathcal{I}$ such that $t'_{i'} \in [t, t+\varepsilon]$ and $P_{i'} \cap \mathcal{M}(o_j) \neq P_i \cap \mathcal{M}(o_j)$;

(c) the operation o_j is again processed at a later moment; formally, there exists $i'' \in \mathcal{I}$ such that $t''_{i''} > t$ and $P_{i''} \cap \mathcal{M}(o_j) \neq \varnothing$.

Now let us formulate few examples of special scheduling problems covered by the **GP**-problem. The most typical examples of fixed dates are job release dates and deadlines, as well as changeover points of the objective functions (e.g. job due dates) and changeover points of the resource availability functions.

The compatibility constraints may prevent a simultaneous processing of:

(i) two different modes μ, μ' for the same operation, if $\{\mu, \mu'\} \not\subseteq P$ for all $P \in \mathcal{P}$;

(ii) two operations o and o' that belong to the same *job* J_j, as in classical shop scheduling models (see, e.g., [11]), if $\mathcal{M}(o) \cap P \neq \varnothing$ implies $\mathcal{M}(o') \cap P = \varnothing$;

(iii) similarly, of two operations on the same *machine* or *processor* with unit capacity;

(iv) more generally, of any subset Q of modes which would require a larger amount of a *resource* than is available at any time $t \in (\tau_k, \tau_{k+1}]$, if $Q \not\subseteq P$ for all $P \in \mathcal{P}(k)$;

(v) finally, assuming mode sets of different operations to be disjoint allows us to model situations of *personal incompatibility.* Let us consider the following example: operations 1,2,3 mean passing an examination by students A,B,C, while there are two alternative examiners: D and E, i.e. "machines" on which those operations can be processed. Let modes 1 and 2 correspond to passing the exam by student A (to examiners D and E respectively); similarly, modes 3,4 correspond to student B, and modes 5,6 — to student C. Then we may have the situation that each examiner can accept the exam of any two students simultaneously, except the case that the students are A and B and the examiner is E because of their incompatible personal relations. In other words, the combination of modes $\{2, 4\}$ is forbidden. Evidently, it would be hard to reflect this constraint if we treated the processing of ANY operation on the same machine as the same mode and therefore, we could not distinguish between the modes 2,4,6.

Furthermore, letting the family $\mathcal{P}(k)$ of feasible patterns vary over time allows us to model:

(vi) *mode date constraints*, whereby mode $\mu \in \mathcal{M}$ cannot be used before a given *mode release date* r_μ, if $\mu \in P \in \mathcal{P}(k)$ implies $k \geq \rho(\mu)$, where $\tau_{\rho(\mu)} = r_\mu$; or after its *mode deadline* $\bar{d}_\mu$, if $\mu \in P \in \mathcal{P}(k)$ implies $k < \delta(\mu)$, where $\tau_{\delta(\mu)} = \bar{d}_\mu$;

(vii) similarly, an *availability calendar* for a resource, whereby the processing in any mode using the resource may be restricted to one or several given time intervals;

(viii) more generally, the restriction of any given subset of modes to a finite number of given time intervals.

Note that the mode date constraints (vi) can also be used to model *job* release dates and deadlines; resource availability calendars (vii) can be used to model planned equipment upgrades and maintenance, varying (but piecewise constant) speeds and yield, and certain forms of *nonrenewable* (or *consumable*) *resources.* Some examples of classic scheduling problems that fall into this framework are the precedence constrained preemptive job shop problem $J|prec,\ pmtn|F$, and the preemptive scheduling problem $R|r_j, \bar{d}_j,\ prec,\ pmtn|F$ on unrelated parallel machines subject to release dates, deadlines and precedence constraints.

Next, precedence constraints of the form (1) enable us to model many interesting constraints on the order of processing the operations. For instance,

(ix) to make two operations start (alternatively, be completed) at the same point in time in **any** feasible schedule; to do this for operations o_i, o_j, we add to graph G two arcs: (o_i, o_j) and (o_j, o_i); if we now put $p_{ij} = -p_i,\ p_{ji} = -p_j$, then the operations will always start simultaneously; alternatively, if we put $p_{ij} = -p_j,\ p_{ji} = -p_i$, then the operations will finish at the same time in any feasible schedule;

(x) to make several operations follow in any feasible schedule in a common bunch, so that no operation of the bunch outruns other operations of the bunch too much, as well as no operation falls behind too much; for instance, operations o_1, o_2, o_3 have to be processed on the same machine in a bunch; to achieve this, we add to graph G an arc (o_i, o_j) of length $p_{ij} = -p_1 - p_2 - p_3$ for each pair $i, j \in \{1, 2, 3\},\ i \neq j$; then, as one can easily check, in any feasible schedule these three operations will be processed successively, without an intermediate delay of the machine;

(xi) to make an operation o_i be processed within an interval of a given length ℓ_i (which however is not fixed in time), we add to graph G a loop (o_i, o_i) of length $p_{ii} = -\ell_i$; in particular, to prevent any delay in processing this operation, we should put $\ell_i = p_i$ (although, this cannot prevent an interruption of the operation by a zero-length slice).

Of course, such constraints cannot be realized without using cycles and arcs of negative weight in graph G.

2.3 Finiteness of the number of interruptions

Lemma 2.2 *If for a given instance of the* **GP**-*problem there exists a feasible schedule* S, *then there exists a feasible schedule* S' *with the same values of the operation completion times* *and at most* $(2\xi + D - 1)H + 2\xi$ *slices, where* ξ *is the number of operations,* D *is the number of fixed dates,* H *is the number of feasible patterns.*

Proof. Suppose, we are given a feasible schedule S, and let S_j, C_j be the starting and the completion times of operation o_j in schedule S. We define a set of distinguished dates $T = \{S_j, C_j \,|\, o_j \in \mathcal{O}\} \cup \{\tau_k \,|\, k = 1, \ldots, D\} \doteq \{\tau'_k \,|\, k = 1, \ldots, K\}$, with $\tau'_1 < \tau'_2 < \cdots < \tau'_K$. It is clear that each slice $\hat{S}_i \in S$ is entirely located within some interval $I_k = [\tau'_k, \tau'_{k+1}]$ for $k = 1, \ldots, K-1$.

Let $\mathcal{S}_k$ be the set of slices located within the interval I_k in schedule S. For each $k = 1, \ldots, K-1$ we do the following.

1. For each operation o_j with $C_j = \tau'_{k+1}$ we choose a pattern P'_j used by the operation o_j at time τ'_{k+1} and form an additional zero-length P'_j-slice $\hat{S}'_j$ and assign it to time τ'_{k+1}. Analogously, for each operation o_j with $S_j = \tau'_k$ we choose a pattern P''_j used by the operation o_j at time τ'_k and form an additional zero-length P''_j-slice $\hat{S}''_j$ and assign it to time τ'_k. Let $\mathcal{S}'_k$ be the set of such slices. We also include into the set $\mathcal{S}'_k$ all zero-length slices $\hat{S} \in \mathcal{S}_k$ containing operations o_j with starting times $S_j = \tau'_{k+1}$ or completion times $C_j = \tau'_k$.
2. For each pattern P feasible in the whole interval I_k glue together all P-slices $\hat{S}_i \in \mathcal{S}_k \backslash \mathcal{S}'_k$. Sequence the resulting aggregated slices within interval I_k in an arbitrary order.

It can be seen that the above transformation of schedule S (into a schedule S') does not change the completion and starting times of operations because of dummy zero-length slices. Therefore, it does not violate precedence and compatibility constraints and preserves the value of the objective function.

The total number of zero-length slices in schedule S' containing at least one completion or starting time of an operation is upper bounded by 2ξ. Then we may claim that for each k all remaining slices located within the interval $[\tau'_k, \tau'_{k+1}]$ in schedule S' use pairwise different patterns. Therefore, each interval $[\tau'_k, \tau'_{k+1}]$ contains at most H such slices. This gives in total (over all intervals) at most $(K-1)H \leq (2\xi + D - 1)H$ slices, which results in the desired bound. □

As a straightforward corollary, we can formulate the following

Theorem 2.3 *If for a given instance of the* **GP***-problem there exists an optimal schedule, then there exists such a schedule with a finite number of interruptions consisting of at most* $(2\xi + D - 1)H + 2\xi$ *slices, where* ξ *is the number of operations,* D *is the number of fixed dates,* H *is the number of feasible patterns.*

It should be noted that the crucial factor providing the finiteness of the number of slices is the finiteness of the number of fixed dates and the number of operations. In the following instance with an infinite number of fixed dates, any optimal schedule has to have an infinite number of slices, despite the fact that a feasible schedule exists having no preemptions at all.

Consider the problem with the objective to minimize the completion time of a single operation of unit length. The operation consumes a resource available within the time intervals $\{[5/2^i, 6/2^i] \,|\, i = 0, 1, \dots\}$. Although scheduling the operation in the interval $[5, 6]$ provides its processing without any interruption, to attain the optimal value $C_{\max} = 3$, we have to make an infinite number of interruptions of the operation.

2.4 Theorem on the existence of an optimal schedule

Theorem 2.4 *For any instance of the* **GP***-problem with a nonempty set of feasible schedules and* K *regular criteria* $F_1(C), \dots, F_K(C)$ *each depending on the vector* $C = (C_1, \dots, C_\xi)$ *of operation completion times, there exists a feasible schedule* S *that lexicographically minimizes the vector-function* $(F_1(C(S)), \dots, F_K(C(S)))$.

Proof. The statement of the theorem will be proved by induction on the number of regular criteria. The basis of induction follows from the existence of a feasible schedule since we can always define an additional objective function $F_0(C(S)) \equiv \text{const}$ (which is, evidently, regular). Thus, it remains to prove the induction step only. Keeping in mind that the completion time C_j of an operation o_j depends on a given schedule S, each function $F_i(C_1(S), \dots, C_\xi(S))$ can be viewed as some function $\tilde{F}_i(S)$ of a schedule.

So, let us assume that for some $n < K$ the set $\mathcal{S}$ of feasible schedules that lexicographically minimize the vector-function $(\tilde{F}_1(S), \dots, \tilde{F}_n(S))$ is nonempty (basis of induction). Since each function $\tilde{F}_i$, $i = 1, \dots, n$, has identical values $\tilde{F}_i(S)$ for all $S \in \mathcal{S}$, this single value can be denoted by F_i^*. Our goal is to prove that there exists a schedule $S \in \mathcal{S}$ at which the value $\inf\{\tilde{F}_{n+1}(S) \,|\, S \in \mathcal{S}\}$ is attained (induction step).

Suppose to the contrary that there is no schedule $S \in \mathcal{S}$ which minimizes the value of $\tilde{F}_{n+1}(S)$ on the set of schedules $S \in \mathcal{S}$. Since the function $\tilde{F}_{n+1}(S)$ is bounded from below (e.g., by the value $F_{n+1}(\tau_1, \dots, \tau_1)$ — because of the function $F_{n+1}(C)$ being nondecreasing), the set of its values $\{\tilde{F}_{n+1}(S) \,|\, S \in \mathcal{S}\}$ has a tight lower bound denoted by F_{n+1}^*. Since there is no schedule $S \in \mathcal{S}$ for which $\tilde{F}_{n+1}(S) = F_{n+1}^*$, by the definition of a tight lower bound there exists an infinite sequence $S_1, S_2, \dots$ of schedules $S_i \in \mathcal{S}$ such that the sequence of values $\tilde{F}_{n+1}(S_1), \tilde{F}_{n+1}(S_2), \dots$ converges to F_{n+1}^*.

By Lemma 2.2, with each schedule S_j $(j = 1, 2, \dots)$ we can associate a feasible schedule S_j' with at most $T \doteq (2\xi + D - 1)H + 2\xi$ slices and the same values of the operation completion times. The latter implies that $S_j' \in \mathcal{S}$, $\forall\, j$, and $\tilde{F}_{n+1}(S_j') \to F_{n+1}^*$ for $j \to \infty$.

Now we define a subset $\mathcal{S}' \subseteq \mathcal{S}$ of schedules with the following properties:

a) each schedule $S' \in \mathcal{S}'$ consists of at most T slices $\{\hat{S}_\nu\}$;

b) each slice is entirely scheduled within one of the time intervals $[\tau_k, \tau_{k+1}]$.

Evidently, all schedules $\{S_j'\}$ found above belong to $\mathcal{S}'$.

Let us consider an arbitrary schedule $S' \in \mathcal{S}'$ consisting of $\theta(S') \le T$ slices $\{\hat{S}_\nu = (t_\nu', t_\nu''; P_\nu) \,|\, \nu = 1, \dots, \theta(S')\}$. Since S' is a full schedule (see Definition 7), the finite set of its slices can be sequenced in such an order $\sigma(S') = (\hat{S}_1, \dots, \hat{S}_{\theta(S')})$ that

1. $t_{\nu-1}'' = t_\nu'$, $\forall\, \nu = 2, \dots, \theta(S')$; $\quad t_1' = \tau_1$, $t_{\theta(S')}'' = \tau_D$;

2. there are indices $\{\theta_k \,|\, k = 1, \dots, D\}$ such that

$$0 = \theta_1 \le \theta_2 \le \dots \le \theta_D = \theta(S'), \tag{2}$$

and $[t'_\nu, t''_\nu] \subseteq [\tau_k, \tau_{k+1}],\ P_\nu \in \mathcal{P}(k)$ holds for all $\nu = \theta_k + 1, \dots, \theta_{k+1};\ k = 1, \dots, D-1$.

Let $\pi(S') = (P_1, \dots, P_{\theta(S')})$ be the sequence of patterns used in slices $\hat{S}_1, \dots, \hat{S}_{\theta(S')}$ in this order, $L(S') = (l_1, \dots, l_{\theta(S')})$ be the vector of lengths of slices $\{\hat{S}_\nu\}$; $\Theta(S') = (\theta_1, \dots, \theta_D) \in \mathbb{Z}^D$. The pair $\mathfrak{A}(S') = (\pi(S'), \Theta(S'))$ and the triplet $\mathfrak{B}(S') = (\pi(S'), \Theta(S'), L(S'))$ will be called a *short* and a *full configuration* of schedule S' respectively.

It is clear that any schedule $S' \in \mathcal{S}'$ is completely defined by its full configuration $\mathfrak{B}(S')$ whose components are constrained by the following relations:

$$\sum_{\nu=\theta_k+1}^{\theta_{k+1}} l_\nu = \tau_{k+1} - \tau_k, \quad k = 1, \dots, D-1; \tag{3}$$

$$\sum_{\nu=1}^{\theta(S')} s_{j\nu} l_\nu = p_j, \quad \forall\ o_j \in \mathcal{O}, \tag{4}$$

where $s_{j\nu} = \sum_{\mu \in P_\nu \cap \mathcal{M}(o_j)} s_\mu$ is the total speed of processing the operation o_j in pattern P_ν. Here relations (3) prescribe the constraints for each fixed date τ_k be a boundary point of some slice $\hat{S}_\nu \in S'$, while relations (4) control the requirement that each operation o_j must be completed. Furthermore, let $s(o_j)$ and $c(o_j)$ be the indices of the first and the last slice in sequence $\pi(S')$, where the operation o_j is processed. (Formally, $s(o_j) = \min\{\nu \,|\, P_\nu \cap \mathcal{M}(o_j) \neq \varnothing\}$, $c(o_j) = \max\{\nu \,|\, P_\nu \cap \mathcal{M}(o_j) \neq \varnothing\}$, which shows that both parameters $s(o_j)$ and $c(o_j)$ entirely depend on the component $\pi(S') = (P_1, \dots, P_{\theta(S')})$.) Then for any arc $(o_i, o_j) \in U$ of graph G, the full configuration $\mathfrak{B}(S')$ must meet the requirement

$$\sum_{\nu=1}^{c(o_i)} l_\nu - \sum_{\nu=1}^{s(o_j)-1} l_\nu \le -p_{ij}. \tag{5}$$

Conversely, it can be seen that each triplet (π, Θ, L), $\pi = (P_1, \dots, P_\theta)$, $\theta \le T$, $\Theta = (\theta_1, \dots, \theta_D) \in \mathbb{Z}^D$, $L = (l_1, \dots, l_\theta) \in [0, \tau_D - \tau_1]^\theta \doteq \mathcal{L}$ satisfying the above requirements uniquely defines a feasible schedule $S' \in \mathcal{S}'$.

Note that $\pi(S')$ may take at most H^T different values (over all schedules $S' \in \mathcal{S}'$), as well as $\Theta(S')$ may take no more than T^D values. Therefore, the short configuration $\mathfrak{A}(S')$ may take at most a finite number of values. Since the sequence of schedules $\alpha = S'_1, S'_2, \dots$ is infinite, α contains an infinite subsequence $\beta = S''_1, S''_2, \dots$ such that all schedules $S''_i \in \beta$ have the same short configuration $\mathfrak{A}(S''_i) = (\pi, \Theta)$ and only differ in vectors $L(S''_i)$.

Since all vectors $L(S''_i)$ belong to the compact domain $\mathcal{L} \subset \mathbb{R}^\theta$, we can find an infinite subsequence $\gamma = S'''_1, S'''_2, \dots$ of the sequence β such that the sequence of vectors $L(S'''_i)$ converges to a certain point $L^* = (l^*_1, \dots, l^*_\theta) \in \mathcal{L}$. It can be easily checked that vector L^* meets all relations (3)–(5). Therefore, the triplet (π, Θ, L^*) specifies a feasible schedule S^*. Let us show now that

$$S^* \in \mathcal{S}' \quad \text{and} \quad \tilde{F}_{n+1}(S^*) = F^*_{n+1}. \tag{6}$$

Since the completion time of an operation o_j in each schedule $S_i''' \in \gamma$ can be computed by the formula

$$C_j(S_i''') = \sum_{\nu=1}^{c(o_j)} l_\nu(S_i''')$$

(where $c(o_j)$ is the same for all schedules $S_i''' \in \gamma$ and for schedule S^*), we have

$$\lim_{i\to\infty} C_j(S_i''') = \sum_{\nu=1}^{c(o_j)} \lim_{i\to\infty} l_\nu(S_i''') = \sum_{\nu=1}^{c(o_j)} l_\nu^* = C_j(S^*),\ \forall\ j = 1, \dots, \xi,$$

in other words,

$$C(S_i''') \to C(S^*),\ \text{as } i \to \infty. \tag{7}$$

Now, for each schedule S_i''' we define a vector $\tilde{C}(S_i''') = (\tilde{C}_1(S_i'''), \dots, \tilde{C}_\xi(S_i'''))$, where $\tilde{C}_j(S_i''') = \min\{C_j(S_i'''), C_j(S^*)\}$. Using (7), we have

(a) $\tilde{C}_j(S_i''') \le C_j(S^*)$, for every j;

(b) $\lim_{i\to\infty} \tilde{C}(S_i''') = C(S^*)$;

(c) $\tilde{C}_j(S_i''') \le C_j(S_i''')$, for every j.

Recalling that all functions $F_\mu(C)$ are continuous from the left, from (a) and (b) we derive

$$\lim_{i\to\infty} F_\mu(\tilde{C}(S_i''')) = F_\mu(C(S^*)),\ \forall\ \mu. \tag{8}$$

On the other hand, from (c) and the nondecreasing of $F_\mu(C)$ we obtain $F_\mu(\tilde{C}(S_i''')) \le F_\mu(C(S_i'''))$, and so,

$$\lim_{i\to\infty} F_\mu(\tilde{C}(S_i''')) \le \lim_{i\to\infty} F_\mu(C(S_i''')) = F_\mu^*,\ \mu = 1, \dots, n+1. \tag{9}$$

It follows from (8) and (9) that $\tilde{F}_\mu(S^*) \doteq F(C(S^*)) \le F_\mu^*$ holds for any $\mu = 1, \dots, n+1$. Since schedule S^* is feasible, by the definitions of values F_μ^* strict inequalities $\tilde{F}_\mu(S^*) < F_\mu^*$ are impossible, which implies (6) and completes the proof of the induction condition. □

Note 2.5 In the proof of the above theorem we used that the objective functions are nondecreasing and continuous from the left. As the examples presented in the introduction show, none of those properties can be neglected. Indeed, in the first example the function is continuous from the left, but is not nondecreasing, while in the second one we have an inverse situation: the function is nondecreasing, but is not continuous from the left. In both situations even for the simplest scheduling models we cannot guarantee the existence of an optimal solution, despite the fact that the set of feasible solutions is nonempty.

3 Integer Interruptions in Preemptive Parallel Machine Scheduling with Integer-Concave Objective Functions

Consider the situation when n jobs $J_1, \dots, J_n$ with integer processing times $p_1, \dots, p_n$ and release dates $r_1, \dots, r_n$ have to be scheduled on m parallel identical machines preemptively. Following the notation of Graham *et al.* [8], this scheduling problem is denoted as $P|pmtn, r_j|F$.

The following theorem includes some of the results from [2] and [3] as special cases. In particular, our theorem implies the preemption redundancy for $P|p_j = 1, pmtn, r_j|\sum w_j U_j$ and $P|p_j = 1, pmtn, r_j|\sum w_j T_j$ thereby resolving an open question of [3].

Theorem 3.1 *For any instance of problem $P|pmtn, r_j|F(C_1, \dots, C_n)$ with integer processing times and release dates and with the objective to minimize a regular integer-concave function $F(C_1, \dots, C_n)$, there exists an optimal schedule where all interruptions occur at integral dates.*

Proof. Clearly, for any instance of the above problem the set of its feasible schedules is nonempty, and once the objective function is regular, there always exists an optimal schedule by Theorem 2.4. Assume that we are given an instance of the above problem and its optimal schedule S. Let $C_i(S)$ denote the completion time of job J_i $(i = 1, \dots, n)$ in that schedule and $C_{\max}(S) = \max_{i=1,\dots,n} C_i(S)$. Let $\mathcal{T} = \{t \in \mathbb{Z} \,|\, \min_i r_i \le t < C_{\max}(S)\}$ be the set of time units occupied by the schedule S. We define a bipartite graph $G = (\{J_1, \dots, J_n\}, \mathcal{T}; E)$ such that for any job J_i and a time point $t \in \mathcal{T}$, (J_i, t) belongs to E if and only if $r_i \le t < C_i(S)$. Every edge $e \in E$ has a unit *capacity*. We say that a flow $x = \{x_{it} \,|\, (J_i, t) \in E\}$ in network G is *feasible* if

$$\sum_t x_{it} = p_i, \ \forall\, i = 1, \dots, n; \tag{10}$$

$$\sum_i x_{it} \le m, \ \forall\, t \in \mathcal{T}; \tag{11}$$

$$0 \le x_{it} \le 1, \ \forall\, i, t. \tag{12}$$

The set of feasible flows is denoted by X.

Let $x_{it}(S)$ be the volume of job J_i scheduled in S within the integral time interval $[t, t+1]$. It is clear that the set of values $x(S) = \{x_{it}(S) \,|\, (J_i, t) \in E\}$ meets requirements (10)–(12), and therefore, the flow $x(S)$ determined by schedule S belongs to X.

Next, for any $i = 1, \dots, n$ we define an integral function t_i of $x \in X$:

$$t_i(x) \doteq \max\{t \,|\, x_{it} > 0\}.$$

(The function is well-defined, since we may assume, w.l.o.g., all p_i being positive.) Let

$$\bar{C}_i(x) \doteq t_i(x) + x_{it_i(x)}, \ \bar{C}(x) = (\bar{C}_1(x), \dots, \bar{C}_n(x)).$$

Evidently, we have

$$\bar{C}_i(x(S)) \le C_i(S). \tag{13}$$

Now we define a cost function $\tilde{F}$ on the set X of feasible flows:

$$\tilde{F}(x) \doteq F(\bar{C}(x)).$$

It follows from (13) and nondecreasing of F that

$$\tilde{F}(x(S)) \le F(C(S)). \tag{14}$$

Let us now prove that function $\tilde{F}(x)$ is concave on the set X of feasible flows. Consider $x', x'' \in X,\ \lambda \in (0,1),\ x = \lambda x' + (1-\lambda)x''$. First let us show that

$$t_i(x) = \max\{t_i(x'), t_i(x'')\}. \tag{15}$$

Indeed, since both coefficients λ and $(1-\lambda)$ are strictly positive, the value of $x_{it} = \lambda x'_{it} + (1-\lambda)x''_{it}$ is positive, while at least one of two components x'_{it} and x''_{it} is positive. On the other hand, if both components x'_{it} and x''_{it} are zero, then $x_{it} = 0$ as well, which implies (15).

Let $\tilde{C}_i(x') \doteq \max\{\bar{C}_i(x'), t_i(x'')\}$ and $\tilde{C}_i(x'') \doteq \max\{t_i(x'), \bar{C}_i(x'')\}$. Using (15), we now prove that

$$\tilde{C}_i(x'), \tilde{C}_i(x'') \in [t_i(x), t_i(x)+1]. \tag{16}$$

and

$$\bar{C}_i(x) = \lambda\tilde{C}_i(x') + (1-\lambda)\tilde{C}_i(x''). \tag{17}$$

Indeed, in the case $t_i(x') > t_i(x'')$ we have $\bar{C}_i(x'') \le t_i(x'') + 1 \le t_i(x') = t_i(x)$, hence, $\tilde{C}_i(x'') = t_i(x')$, whereas $\tilde{C}_i(x') = \bar{C}_i(x')$ (which provides (16)). Note also that $x''_{it_i(x)} = 0$. Therefore,

$$\begin{aligned} \lambda\tilde{C}_i(x') + (1-\lambda)\tilde{C}_i(x'') &= \lambda\bar{C}_i(x') + (1-\lambda)t_i(x') = t_i(x') + \lambda x'_{it_i(x')} \\ &= t_i(x) + \lambda x'_{it_i(x)} + (1-\lambda)x''_{it_i(x)} = t_i(x) + x_{it_i(x)} = \bar{C}_i(x). \end{aligned}$$

In the case $t_i(x') < t_i(x'')$ the proof of (16),(17) is similar. Finally, if $t_i(x') = t_i(x'') = t_i(x)$, then we have

$$\tilde{C}_i(x') = \bar{C}_i(x') \in [t_i(x), t_i(x)+1] \quad \text{and} \quad \tilde{C}_i(x'') = \bar{C}_i(x'') \in [t_i(x), t_i(x)+1].$$

So, we have (16). Using these relations, we obtain (17):

$$\begin{aligned} \lambda\tilde{C}_i(x') + (1-\lambda)\tilde{C}_i(x'') &= \lambda\bar{C}_i(x') + (1-\lambda)\bar{C}_i(x'') = t_i(x) + \lambda x'_{it_i(x)} + (1-\lambda)x''_{it_i(x)} \\ &= t_i(x) + x_{it_i(x)} = \bar{C}_i(x). \end{aligned}$$

Now, we can derive

$$\begin{aligned} \tilde{F}(x) &= F(\bar{C}(x)) \overset{(17)}{=} F(\lambda\tilde{C}(x') + (1-\lambda)\tilde{C}(x'')) \overset{(16)\text{ and i.c.}}{\ge} \lambda F(\tilde{C}(x')) + (1-\lambda)F(\tilde{C}(x'')) \\ &\overset{\text{n.d.}}{\ge} \lambda F(\bar{C}(x')) + (1-\lambda)F(\bar{C}(x'')) = \lambda\tilde{F}(x') + (1-\lambda)\tilde{F}(x''), \end{aligned}$$

which implies that function $\tilde{F}(x)$ is concave in X. (Here “i.c.” means *integer-concaveness* and “n.d.” means *nondecreasing*.)

Let $\tilde{x}$ be the flow minimizing $\tilde{F}(x)$ over all $x \in X$. Due to (14), we have

$$\tilde{F}(\tilde{x}) \le \tilde{F}(x(S)) \le F(C(S)). \tag{18}$$

Since the minimum of every concave function over a polytop is achieved on a vertex of the polytop, and since the transportation polytop (10)–(12) is integral, $\tilde{x}$ is an integer flow. Therefore, we can map the flow to a new schedule $\tilde{S}$, in which all interruptions occur at integral time points. The integrality of flow $\tilde{x}$ implies $\bar{C}_i(\tilde{x}) = C_i(\tilde{S})$ for every i, and therefore,

$$\tilde{F}(\tilde{x}) = F(\bar{C}_1(\tilde{x}), \ldots, \bar{C}_n(\tilde{x})) = F(C_1(\tilde{S}), \ldots, C_n(\tilde{S})) = F(C(\tilde{S})).$$

From (18) we have $F(C(\tilde{S})) \leq F(C(S))$, and therefore, schedule $\tilde{S}$ is also optimal. □

A natural attempt to generalize the above theorem is to add precedence constraints. Unfortunately, Baptiste and Timkovsky [2] gave examples showing that preemptions are not redundant for $P2|p_j = 1, intree, pmtn|\sum C_j$ and $P2|p_j = 1, outtree, pmtn|\sum w_j C_j$. This means that the most we could hope for is to add chain precedence constraints, i.e., the directed graph corresponding to these precedence relations must be a collection of chains.

Theorem 3.2 *For any instance of problem $P|r_j, chains, pmtn|F(C_1, \ldots, C_n)$ with integer processing times and release dates and with the objective to minimize a regular integer-concave function $F(C_1, \ldots, C_n)$, there exists an optimal schedule where all interruptions occur at integral dates.*

Proof. The general idea for the proof is very similar to the one of Theorem 3.1. The only difference is in item (3) below.

1. Given an optimal preemptive schedule S, we construct a network G and a flow in this network. Show that the flow is feasible for this network.

2. On the set of feasible flows of maximum capacity in network G we find the optimal network minimizing the objective function. Since the polyhedron corresponding to the set of maximum flows in network G is integral and the optimal solution for a regular integer-concave objective function is attained at a node of the polytop, this implies the existence of an integral optimal solution (a flow in network G). This flow corresponds to an integral schedule S^* in which all interruptions occur at integral dates and which minimizes our objective function.

3. We finally proof that schedule S^* meets precedence constraints, and so, it is feasible and optimal.

Instead of presenting the complete proof, we just mention the points in which the proof differs from that of Theorem 3.1. First of all, network $G = (V, E)$ is slightly different from the one defined in the proof of Theorem 3.1. We define here the vertex set as $V = \{s\} \cup \{\tau\} \cup \{J_1, \ldots, J_n\} \cup \Lambda \cup \mathcal{T}$, where the set Λ contains a vertex v_{ij} for every pair of jobs J_i and J_j such that J_i precedes J_j in the precedence relations and both jobs share the same unit time interval $[t, t+1]$ for some $t \in \mathcal{T}$. The edge set E is defined as follows.

1. There are n edges of capacities p_j $(j = 1, \ldots, n)$ connecting source s with jobs $\{J_1, \ldots, J_n\}$.

2. There are $|\mathcal{T}|$ edges of capacity m connecting vertices from the set $\mathcal{T}$ with the sink τ.

3. If J_i precedes J_j and for some $t \in \mathcal{T}$ we have $t < C_i(S) \leq \Gamma_j(S) < t+1$ (where $\Gamma_j(S)$ is the starting time of job J_j in schedule S), then we define three edges (J_i, v_{ij}), (J_j, v_{ij}) and (v_{ij}, t) of unit capacities. In such a case we say that J_i and J_j compete for the same time interval $[t, t+1]$. Every job J_j has at most two time intervals where it competes with other jobs.

4. Let $I_j \subseteq \mathcal{T}$ be the set of "noncompetitive" time units for job J_j, i.e., $\lfloor \Gamma_j(S) \rfloor \leq t < C_j(S)$ and job J_j does not compete with any other job in the time interval $[t, t+1]$ for $t \in I_j$. For any time unit $t \in I_j$ we define an edge (J_j, t) of unit capacity.

The intuition behind the above definition is that we define nearly the same graph, with the only difference: if two jobs related by a precedence constraint are using the same time interval $[t, t+1]$ in a preemptive schedule, then we would like to create some capacity constraints preventing the concurrent unit assignment of such jobs to the same time unit t.

Given an optimal preemptive schedule S, we define the flow variables $\{x_{ij}\}$ on edges of graph G as follows. We send p_i units of flow from the source s into job vertex J_i for $i = 1, \ldots, n$. After that we split the flow according to schedule S. If $x_{it}(S) > 0$, i.e., there is a nonzero amount of job J_i processed in the time interval $[t, t+1]$, we send $x_{it}(S)$ units of flow from vertex J_i to vertex $t \in \mathcal{T}$ if there is a direct edge (J_i, t). Otherwise, there exists either a successor or predecessor J_j of J_i processed in time unit t in schedule S. In this case we send $x_{it}(S)$ units of flow to the intermediate vertex $v_{ij} \in \Lambda$. All other flow variables are defined by the flow conservation constraint. This s-to-t flow is feasible since it satisfies the flow conservation and capacity constraints.

The proof that the flow defined above is feasible for network G is straightforward, and the arguments to prove item (2) are the same as those used for Theorem 3.1. So, it only remains to proof that any integral flow in network G corresponds to a feasible schedule which meets precedence constraints.

Indeed, if J_i precedes J_j, then all the time units $t \in \mathcal{T}$ to which there is a positive flow from vertex J_i are located before the time units to which there may exist a positive flow from vertex J_j (due to the properly defined set of edges of graph G). The only intersection of these two sets of "time" vertices is the "boundary" time unit t^* to which we have an edge from vertex v_{ij}. But due to the integrality of the flow only one of two jobs $\{J_i, J_j\}$ may produce a positive (in fact, unit) flow from vertex J_ν ($\nu \in \{i, j\}$) to vertex t^*. Thus, all time units to which job J_i has a nonnegative flow in network G are located strictly before the time units to which job J_j has a nonnegative flows, which means that precedence constraints are satisfied.

The remainder of the proof is similar. □

4 Shop Scheduling

In this section, we will show first that in some special cases of the **GP**-problem (such as shop scheduling problems) the number of slices that is guaranteed for an optimal schedule by Theorem 2.3 can be significantly decreased. We will consider the classical shop scheduling problems where all machines are fully available during the scheduling horizon.

In a preemptive shop scheduling problem we are given a set of jobs $\mathcal{J} = \{J_1, \ldots, J_n\}$, a set of machines $\mathcal{M} = \{M_1, \ldots, M_m\}$, and a set of operations $\mathcal{O} = \{o_1, \ldots, o_\xi\}$. Each operation $o_k \in \mathcal{O}$ belongs to a specific job $J(o_k) \in \mathcal{J}$ and must be processed on a specific machine $\mathcal{M}(o_k) \in \mathcal{M}$ for a given amount of time p_k which is a non-negative integer. At any date, at most one operation can be processed on each machine and at most one operation of each job can be processed. Any operation can be preempted at any date and resumed later without any penalty.

Shop scheduling problems are further classified based on *ordering restrictions* for the operations of a job. Ordering restrictions represent a special case of precedence constraints between operations mentioned in the formulation of the **GP**-problem, when the weights p_{ij} of all arcs $(o_i, o_j) \in U$ are equal to zero. The existence of such an arc $(o_i, o_j) \in U$ in graph G just means that the operation o_j may start only after the operation o_i is completed. In an *open shop*, the operations of each job may be processed in any order. In a *job shop*, the operations of each job must be processed in a given linear order specific to that job. In a *dag shop*, the operations of each job J_j must be processed subject to a given partial order defined by a directed acyclic graph (or *dag*) G_j specific to that job. Clearly, dag shop scheduling includes both open shop and job shop scheduling as special cases.

By definition, all interruptions occur at completion times of slices. Since in classical shop scheduling models, at most m operations can be processed (and therefore, can be interrupted!) in each slice, the number of interruptions can be bounded above by the amount $\theta m - \xi$, where θ is the number of slices. The amount ξ is subtracted in this bound, because from among θm possible pairs $\langle$a slice $\hat{S}_j$, a machine $M_i\rangle$, one should remove all pairs corresponding to the completion of an operation on a machine. In fact, this upper bound on the number of interruptions may be far from being tight. At least, it should be reduced by the number of idle time intervals on all machines within the interval $[0, C_{\max}(S)]$.

Furthermore, it is clear that the number of interruptions in a feasible schedule S is finite (polynomial) **if and only if** the number of slices is finite (polynomial). Next, let

$$l_{\max} \doteq \max_{M_i \in \mathcal{M}} \sum_{\{o_k \in \mathcal{O} \mid \mathcal{M}(o_k) = M_i\}} p_k \quad \text{and} \quad l_{\max}^{\mathcal{J}} \doteq \max_{J_j \in \mathcal{J}} \sum_{\{o_k \in \mathcal{O} \mid J(o_k) = J_j\}} p_k$$

stand for the *maximum machine load* and the *maximum job length*, respectively. Evidently, the makespan of any feasible schedule for a given instance of the preemptive shop scheduling problem formulated above cannot be less than the amount $\bar{C} \doteq \max\{l_{\max}, l_{\max}^{\mathcal{J}}\}$. The following result was established in [6] for the open shop problem (which, as was noted above, is a special case of the dag shop problem).

Theorem 4.1 (Gonzalez and Sahni [6]) *For any instance of the $O|pmtn|C_{\max}$ problem with ξ operations, n jobs and m machines ($m \leq n$), there exists an optimal schedule S with at most $\xi + n + m$ slices and the makespan $C_{\max}(S) = \bar{C}$.*

Note 4.2 *A slight modification of the Gonzalez and Sahni algorithm enables one to reduce the number of slices down to $\xi + m$.*

Although, the theorem was proved by Gonzalez and Sahni for the classical version of the open shop problem, in which each job may have at most one operation per machine, the result can be easily extended to the case of the generalized open shop problem $\tilde{O}|pmtn|C_{\max}$, where the above restriction is omitted.

Theorem 4.3 *For any instance of the $\tilde{O}|pmtn|C_{\max}$ problem with ξ operations, n jobs and m machines ($m \le n$), there exists an optimal schedule S with at most $\xi + m$ slices and the makespan $C_{\max}(S) = \bar{C}$.*

Proof. Given an instance $\tilde{I}$ of the $\tilde{O}|pmtn|C_{\max}$ problem, let us glue together all operations of each job J_j being processed on the same machine M_i. The resulting "big" operations form an instance I of the classical open shop problem. Let ξ' be the number of "big" operations (clearly, $\xi' \le \xi$). Applying to I Note 4.2, we obtain an optimal schedule S with at most $\xi' + m$ slices. Let $\mathcal{T}$ be the set of completion times of those slices. Clearly, $|\mathcal{T}| \le \xi' + m$.

Now we define an optimal schedule $\tilde{S}$ for the instance $\tilde{I}$ by choosing an arbitrary order for the original ("small") operations within each "big" operation. Consider a "big" operation $o \in I$ consisting of k "small" operations $\tilde{o}_1, \dots, \tilde{o}_k$ being processed in $\tilde{S}$ in this order. Since the completion time of the operation $\tilde{o}_k$ coincides with the completion time of the "big" operation o, it belongs to $\mathcal{T}$. The completion times of the other $k-1$ "small" sub-operations of the operation o may generate new completion times of slices in schedule $\tilde{S}$, not belonging to $\mathcal{T}$, therefore, the total number of such new completion times is no greater than $\xi - \xi'$. It can be easily seen that no other completion times of slices may occur in schedule $\tilde{S}$, and so, the total number of such dates in schedule $\tilde{S}$ is no greater than $\xi + m$. □

In what follows, $C_j(S)$ will stand for the completion time of the operation $o_j \in \mathcal{O}$ in schedule S.

Lemma 4.4 *For any instance of the preemptive dag shop problem with a set of operations $\{o_1, \dots, o_\xi\}$ and any feasible schedule S there exists a feasible schedule S' with at most $\xi(\xi + 1)/2 + m\xi$ slices and such that $C_j(S') \le C_j(S),\ j = 1, \dots, \xi$.*

Proof. For simplicity, we may assume w.l.o.g. that the instance under consideration contains no zero-length operations. (Adding one such operation increases the needed number of slices by at most one.) Suppose, we are given a feasible schedule S for the given instance, and let $C'_0 = 0 < C'_1 < \dots < C'_{\xi'}$ represent the sequence of all pairwise different operation completion times in schedule S (so, we have $\xi' \le \xi$). We split the schedule S at points $\{C'_\tau\}$. Let $\mathcal{O}_{k\tau}$ denote the set of all pieces of an operation o_k scheduled in S within the time interval $I_\tau \doteq [C'_{\tau-1}, C'_\tau]$, $\mathcal{O}_\tau = \cup_{k=1}^t \mathcal{O}_{k\tau}$, and let $p_{k\tau}$ be the total length of all pieces $o \in \mathcal{O}_{k\tau}$. Clearly, the feasibility of schedule S implies

$$\sum_{\{o_k \in \mathcal{O} \,|\, \mathcal{M}(o_k) = M_i\}} p_{k\tau} \le C'_\tau - C'_{\tau-1},\ M_i \in \mathcal{M},\ \tau = 1, \dots, \xi'; \qquad (19)$$

$$\sum_{\{o_k \in \mathcal{O} \,|\, J(o_k) = J_j\}} p_{k\tau} \le C'_\tau - C'_{\tau-1},\ J_j \in \mathcal{J},\ \tau = 1, \dots, \xi'. \qquad (20)$$

Since we cannot have any precedence constraints between operations processed in the same interval I_τ, pieces from the set $\mathcal{O}_\tau$ form an instance N_τ of the $\tilde{O}|pmtn|C_{\max}$ problem in which the set of pieces $\mathcal{O}_{k\tau}$ can be treated as a set of pieces of an operation $o_{k\tau}$. By (19) and (20), each machine load and each job length (and therefore, the parameter $\bar{C}(N_\tau)$) in the instance N_τ can be bounded from above by the length of the interval I_τ. Therefore, applying Theorem 4.3, we can find a preemptive schedule S_τ for the instance N_τ with length $C_{\max}(S_\tau) \le C'_\tau - C'_{\tau-1}$ and having no more than $\xi - \tau + 1 + m$ slices (because at least $\tau - 1$ operations have completed prior to the interval I_τ). Concatenating the schedules $S_1, \dots, S_{\xi'}$ (in this order), we get a feasible preemptive schedule S' for the original problem with at most $\xi(\xi+1)/2 + m\xi$ slices. Clearly, the described transformation of schedule S increases no operation completion times. □

The following result is a straightforward corollary of Lemma 4.4.

Lemma 4.5 *If there exists an optimal schedule for an instance of the preemptive dag shop problem with ξ operations, m machines and a nondecreasing objective function $F(C_1, \dots, C_\xi)$ of the operation completion times then there exists such a schedule with at most $\xi(\xi+1)/2 + m\xi$ slices.* □

Since there are no reasons for the set of feasible schedules to be empty in our case (e.g., no deadlines are put to the operations), for the dag shop problem with an arbitrary regular criterion $F(C_1, \dots, C_\xi)$, we can derive (by means of Theorem 2.4) the following

Theorem 4.6 *For any instance of the preemptive dag shop problem with ξ operations, m machines and a regular criterion $F(C_1, \dots, C_\xi) \to \min$, there exists an optimal schedule with at most $\xi(\xi+1)/2 + m\xi$ slices.* □

Another sufficient condition for an optimal schedule to exist is provided by

Lemma 4.7 *Suppose, we are given a function $F(x_1, \dots, x_\xi)$ that may take only a finite number of values. Then for any instance of the preemptive dag shop scheduling problem with the objective to minimize the function $F(C_1, \dots, C_\xi)$ of the operation completion times, there exists an optimal schedule.* □

The proof is trivial. The following result is a corollary of Lemmas 4.5 and 4.7.

Theorem 4.8 *Suppose, we are given a nondecreasing function $F(x_1, \dots, x_\xi)$ that may take only a finite number of values. Then for any instance of the preemptive dag shop problem with ξ operations, m machines and the objective to minimize the function $F(C_1, \dots, C_\xi)$ of the operation completion times, there exists an optimal schedule with at most $\xi(\xi+1)/2 + m\xi$ slices.* □

It can be easily checked that all classical objective functions meet the conditions put on the function F in Theorem 4.6. Moreover, the objective function weighted number of late jobs $\sum w_j U_j$ meets both the conditions of Lemma 4.6 (in particular, it is continuous from the left) and the conditions of Theorem 4.8 (because it may take only a finite number of values).

Next, we introduce a notion of an *early preemptive schedule* playing an important role in our further analysis and derive a few simple properties of such schedules.

Definition 10 A feasible schedule for a preemptive scheduling problem is called an *active preemptive schedule*, if it contains a finite number of slices, and no positive length piece of an operation can be moved to an earlier idle time interval without violating the feasibility of the schedule.

Definition 11 A piece of an operation in a preemptive schedule S is called a *whole piece* (or *w-piece*, for short), if it is a maximal (by inclusion) continuously processed piece of the operation.

Let S be an active preemptive schedule. It can be shown that for each w-piece o' of an operation there exists a *critical chain* $Ch(o')$ in S, *i.e.*, a chain $o'_1 \to o'_2 \to \cdots \to o'_k$ consisting of a finite number of consecutive w-pieces of operations and such that

1. it starts at time zero and finishes by the w-piece $o'_k = o'$;
2. the start time of each w-piece coincides with the completion time of the previous w-piece;
3. every two consecutive w-pieces either belong to the same job or are processed by the same machine.

Lemma 4.9 *If in an active preemptive schedule S of the dag shop problem there exists an operation o_i with nonintegral completion time $C(o_i, S)$, then there exists a fractional w-piece o' in schedule S such that*

(a) $C(o', S) < C(o_i, S)$*;*

(b) $C(o', S)$ *is nonintegral;*

(c) there is no operation o_j such that $C(o', S) = C(o_j, S)$.

Proof. Let o_i be the operation with the least nonintegral completion time $C(o_i, S)$, and let $Ch(o'_i)$ be a critical chain for the last w-piece o'_i of the operation o_i. If w-piece o'_i is integral, then there exists a fractional w-piece o' in the chain $Ch(o'_i)$ with completion time $C(o', S) < C(o_i, S)$. Alternatively, if o'_i is fractional, then there exists another fractional w-piece o' of the operation o_i with $C(o', S) < C(o_i, S)$. In both cases we have property (a). Let now o'' be the fractional w-piece with the least completion time $C(o'', S)$ over all fractional w-pieces in schedule S. Clearly, $C(o'', S) \le C(o', S) < C(o_i, S)$. Moreover $C(o'', S)$ is nonintegral since otherwise, in the chain $Ch(o'')$ there would be another fractional w-piece o''' with $C(o''', S) < C(o'', S)$, which contradicts the choice of o''. So, we have (b). Finally, (c) follows from (a), (b), and the choice of operation o_i. □

Since the w-piece o' from the formulation of Lemma 4.9 cannot be the last piece of the operation o_k to which o' belongs (because of (c)), it follows that $C(o', S)$ is a *preemption date* (a date when the operation o_k is interrupted in schedule S). So, we obtain

Corollary 4.10 *If the completion time $C(o_i, S)$ of an operation o_i in an active preemptive schedule S of the dag shop problem is nonintegral, then there exists a nonintegral preemption date $t' < C(o_i, S)$ in schedule S at which no operation completes its processing.* □

4.1 Structural properties of the preemptive job shop problem

Theorem 4.11 *For any instance of the preemptive job shop problem with integer processing times and a regular criterion* $F(C_1, \ldots, C_\xi) \to \min$*, there exists an optimal schedule* S *satisfying the following properties:*

(a) S *is an active preemptive schedule;*

(b) all changeover dates in S *are integral;*

(c) the set of changeover dates coincides with the set of completion times of the operations.

Proof. Let S' be a feasible schedule lexicographically minimizing the vector-function $(F(C_1, \ldots, C_\xi), \sum_{k=1}^{\xi} C_k)$. The existence of such a schedule follows from the Theorem 2.4 since both functions are regular and the set of feasible schedules is nonempty. Due to Lemma 4.4, there exists another feasible schedule S with a finite number of slices and such that $C_j(S) \leq C_j(S')$, $j = 1, \ldots, t$. Since both functions are nondecreasing, schedule S also minimizes the vector-function $(F(C_1, \ldots, C_\xi), \sum_{k=1}^{\xi} C_k)$. It is also clear that S is an active preemptive schedule. Otherwise, we could obtain a feasible schedule S'' with a strictly smaller value of the function $\sum_{k=1}^{\xi} C_k$ (and a non-greater value of the function F) by moving an ending positive-length piece of some operation to an earlier idle time interval which contradicts the optimality of schedule S.

Since the objective function F is nondecreasing in operation completion times, we may assume, w.l.o.g., that each zero-length operation is processed in schedule S exactly at the completion time of the previous operation of the same job. Furthermore, we may assume that there are no *proper* zero-length w-pieces in schedule S, i.e. all zero-length w-pieces correspond to zero-length operations in the instance.

Next we get rid of "bad preemption dates" that is preemption dates at which no operation is completed. Suppose in schedule S there exist such "bad" interruptions that happen at "bad preemption dates", and let t_1 be the earliest such date. Since there are no proper zero-length w-pieces, $t_1 > 0$. Suppose that it happens with a w-piece o_1' of an operation o_1 of job J_1 on machine M_1, and let I' be the time interval between the completion time of the w-piece o_1' and the starting time of the next w-piece o_1'' of the operation o_1. Since we deal with the job shop problem, i.e. we have a linear order of processing the operations for each job, no other piece of an operation belonging to the same job J_1 can be processed in the time interval I'. Since S is an active schedule, there can be no idle time on machine M_1 in the time interval I'. Therefore, there exists a positive-length w-piece of an operation o_2' being processed on machine M_1 right after the w-piece o_1'. Clearly, the w-pieces o_1' and o_2' must belong to different jobs, say J_1 and J_2.

Let $t_0 < t_1$ be the latest operation completion time prior to t_1. If there are no such operations then $t_0 \doteq 0$. Notice that there are no changeover dates in the time interval (t_0, t_1). We now describe a transformation of schedule S to another feasible schedule S' with a strictly less value of the vector $(F(C_1, \ldots, C_\xi), \sum_{k=1}^{\xi} C_k)$. Our actions will depend on which of the two operations o_1 or o_2 is completed first in the schedule S.

In the case when $C(o_2, S) < C(o_1, S)$, we consider the union of the subintervals where o_1 or o_2 are processed in the time interval $[t_0, C(o_2, S))$. We now process o_2 first in these subintervals as early as possible, and after that we process o_1 in the remaining time. It can be observed that no piece of jobs J_1 and J_2 can be processed in the time interval $(t_0, C(o_2, S))$ on machines $M_i \in \mathcal{M} \setminus \{M_1\}$, and therefore the new schedule is feasible. As a result of the transformation, we have $C(o_2, S') < C(o_2, S)$, and $C(o, S') = C(o, S)$ for all other operations $o \in \mathcal{O}$. The schedule S' is also optimal since the the objective function F is nondecreasing. Moreover, the value of the function $\sum C_k$ is strictly decreased. — A contradiction.

The symmetric case $C(o_1, S) < C(o_2, S)$ can be considered in a similar way, leading to a new schedule S' with better completion time for operation o_1 and leaving all other completion times unchanged. Thus, we may conclude that schedule S cannot have "bad interruptions". Next, since S is an active preemptive schedule, the starting time of each w-piece of an operation (except pieces contained in the first slice) coincides with a completion time of another w-piece. But, as we have already established, each w-piece completes at a "good" time equal to the completion time of some operation. So, the set of all changeover dates in schedule S coincides with the set of completion times of operations.

Finally, property (b) follows from the properties (a) and (c) and Corollary 4.10. □

Note 4.12 *It follows from the property (c) of the above theorem that for the job shop problem with ξ operations and an arbitrary regular criterion, there exists an optimal schedule with at most ξ slices.*

4.2 Structural properties of the preemptive dag shop problem

Unfortunately, we were unable to prove strong structural properties for the preemptive two-machine dag shop scheduling problem with a regular criterion. Below we define a more restrictive class of objective functions which still includes many interesting and classical criteria. Let $\mathcal{B} \subseteq 2^{\mathcal{O}}$ be a subset of the set of operation sets, $N \doteq |\mathcal{B}|$. For a given schedule S and a set $B \in \mathcal{B}$, let $C(B) = \max_{o_j \in B} C_j(S)$ stand for the completion time of set $B \in \mathcal{B}$. Let $C = (C(B_1), \ldots, C(B_N))$ denote the vector of the operation set completion times, where $B_1, \ldots, B_N$ are the sets from $\mathcal{B}$ numbered in an arbitrary order. We will consider regular quasiconcave objective functions $F(C)$ of set completion times. As an example of such functions, we mention here the *total weighted operation set completion time* $\sum_{B \in \mathcal{B}} w_B C(B)$ studied in [13, 14]. A more general example of the function is $\sum_{B \in \mathcal{B}} w_B C(B)^{\lambda_B}$, where $\lambda_B \in [0, 1]$ for each $B \in \mathcal{B}$. Note also that our objective function with set completion times models bipartite precedence constraints for a certain special case of $1|prec|\sum_j w_j C_j$ that is equivalent to the general case of this problem [20].

Theorem 4.13 *For every instance of the preemptive two-machine dag shop problem with ξ operations and a regular quasiconcave objective function of the operation set completion times $F(C)$ there exists an optimal schedule in which:*

1. *there are at most ξ slices and at most ξ interruptions;*

2. *if all processing times of operations are integer, then all interruptions occur at integral dates.*

Proof. By Theorem 4.6, there exists at least one optimal schedule with a finite number of slices. Let $\tilde{S}$ be an optimal schedule with the minimum number of slices. Let $\mathcal{O}_1$ and $\mathcal{O}_2$ be the sets of operations that have to be processed on machines M_1 and M_2 respectively. We also define two dummy operations $\tilde{o}_1$ and $\tilde{o}_2$ assumed to be processed on machines M_1 and M_2 (respectively) every time that the corresponding machine is idle; $\mathcal{O}'_\nu \doteq \mathcal{O}_\nu \cup \{\tilde{o}_\nu\}$ $(\nu = 1, 2)$. We now define a bipartite multigraph $\tilde{G} = (\mathcal{O}'_1, \mathcal{O}'_2; E)$ by establishing a one-to-one correspondence between the set of slices of schedule $\tilde{S}$ and the set of edges E of graph $\tilde{G}$. Namely, each edge $e = (o_1, o_2) \in E$ corresponds to a slice S_τ in which operation $o_\nu \in \mathcal{O}'_\nu$ $(\nu = 1, 2)$ is processed on machine M_ν. For each edge $e \in E$, we define the weight w_e equal to the length of the corresponding slice.

First of all, it can be easily checked that graph $\tilde{G}$ has no multiple edges. Indeed, if there is more than one slice in $\tilde{S}$ corresponding to the same pair of operations $\{o_1, o_2\}$, we glue these slices together accumulating them at the last such slice, which, clearly, preserves the feasibility of the schedule and does not increase completion times. As a result of this gluing procedure, we obtain a new optimal schedule S' with strictly fewer number of slices, which contradicts the choice of schedule $\tilde{S}$.

Next, we may assume that each zero-length operation o_i is presented in schedule $\tilde{S}$ in a single slice of zero length. Otherwise, we could decrease the number of slices containing the operation o_i (without increasing the multiplicity of other zero-length operations) by assigning the operation o_i to its earliest slice only. This, clearly, decreases its completion time, and thereby, does not violate the optimality of the schedule. Unfortunately, we cannot get rid of such zero-length operations completely, because they may significantly affect the value of the objective function, and so, the positions of such operations in the desired schedule have to be determined simultaneously with the positions of other operations. Therefore, there is exactly one edge incident to a vertex of graph $\tilde{G}$ corresponding to a zero length operation, which implies that such vertices cannot lie on cycles of graph G (*Note 1*).

Finally, it can be easily observed (*Note 2*) that there may be no zero-length slices in schedule $\tilde{S}$ except those containing zero-length operations. (Otherwise, such a slice could be eliminated, which would generate an optimal schedule with fewer number of slices and would contradict the choice of schedule $\tilde{S}$.) Therefore, due to these two notes, we may conclude that no cycle in $\tilde{G}$ may contain edges corresponding to zero-length slices.

We now show that $\tilde{G}$ contains no cycles. Assume to the contrary that there is a cycle $\mathcal{C}$ in $\tilde{G}$. We may also assume that $\mathcal{C}$ is a simple cycle. Since $\tilde{G}$ is a bipartite graph, cycle $\mathcal{C}$ can be represented as a union of two edge disjoint matchings: $\boldsymbol{M_1}$ and $\boldsymbol{M_2}$. For any $\varepsilon \in \mathbb{R}$ and any edge $e \in \mathcal{C}$ we can define a new weight $w_e(\varepsilon)$:

$$w_e(\varepsilon) = \begin{cases} w_e + \varepsilon, & \text{for } e \in \boldsymbol{M_1}, \\ w_e - \varepsilon, & \text{for } e \in \boldsymbol{M_2}. \end{cases}$$

Let $\varepsilon_1 = \min_{e \in \boldsymbol{M_1}} w_e$ and $\varepsilon_2 = \min_{e \in \boldsymbol{M_2}} w_e$. Since all weights w_e are positive due to the above conclusion about zero-length slices, we have $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$. It is clear that for every $\varepsilon \in [-\varepsilon_1, \varepsilon_2]$ the new weights $w_e(\varepsilon)$ of edges $e \in \tilde{G}$ are nonnegative. Let us show that these new weights correspond to a feasible schedule S_ε for the original instance.

Firstly, it should be noticed that each vertex $o_i \in \mathcal{C}$ is incident to a single edge $e_1 \in \boldsymbol{M_1}$ and a single edge $e_2 \in \boldsymbol{M_2}$. Therefore, subtracting the amount ε from the weight of edge e_2 and adding it to the weight of another edge e_1 incident to the same vertex o_i corresponds to decreasing the processing time of the operation o_i in one slice and increasing it in another slice by the same amount. As a result, the overall processing time of each operation remains unchanged. Secondly, the relative positions of operations with respect to each other also remain unchanged, which implies that no precedence and no compatibility constraints can be violated under the described transformation of weights. So, each schedule S_ε for $\varepsilon \in [-\varepsilon_1, \varepsilon_2]$ is feasible.

Now let us analyze how the set completion times and the value of the objective function change under the above transformation of the schedule. Let $B \in \mathcal{B}$ be one of the "control sets" of operations, and let $\hat{S}_\tau$ be the last slice containing a piece of an operation from B. Under the transformation of weights $w_e(\varepsilon)$ (and the corresponding transformation of slices), the set completion time $C(B, S_\varepsilon)$ may be changed as a result of changing the length of some slices preceding the slice $\hat{S}_\tau$. Indeed, let $n_1(S_\varepsilon)$ and $n_2(S_\varepsilon)$ be the numbers of slices preceding $\hat{S}_\tau$ in schedule S_ε (including, may be, the slice $\hat{S}_\tau$ itself) and corresponding to edges from $\boldsymbol{M_1}$ and $\boldsymbol{M_2}$ respectively. Then the completion time of the set B in schedule S_ε can be calculated as $C(B, S_\varepsilon) = C(B, \tilde{S}) + \alpha(B)\varepsilon$, where $\alpha(B) = n_1(S_\varepsilon) - n_2(S_\varepsilon)$, and so, each function $C(B, S_\varepsilon)$ is a linear function of ε.

For the extremum values $\varepsilon = -\varepsilon_1$ and $\varepsilon = \varepsilon_2$, some of those slices may become of zero length, but we still keep them in schedule S_ε since we don't want to change the values of $n_1(S_\varepsilon)$ and $n_2(S_\varepsilon)$. Thus, while varying ε within the interval $[-\varepsilon_1, \varepsilon_2]$, all coefficients $\alpha(B)$ $(B \in \mathcal{B})$ do not change their value. Let $\alpha = (\alpha(B_1), \ldots, \alpha(B_N))$. Since function F is quasiconcave, its minimum over all $\varepsilon \in [-\varepsilon_1, \varepsilon_2]$ is attained at one of the endpoints: either $-\varepsilon_1$, or ε_2. W.l.o.g., we may assume that it is $-\varepsilon_1$. Then $F(C - \alpha\varepsilon_1) \leq F(C)$, which means that schedule $S_{-\varepsilon_1}$ is also optimal. Furthermore, one can observe that at least one slice (corresponding to an edge from $\boldsymbol{M_1}$) becomes of zero length, and therefore, can be deleted from the schedule, which contradicts the choice of schedule $\tilde{S}$ (see *Note 2*). Thus, we have proved that there cannot be any cycles in graph $\tilde{G}$.

Since graph $\tilde{G}$ contains $\xi + 2$ vertices and has no cycles, it contains no more than $\xi + 1$ edges. One of those edges is definitely the edge $(\tilde{o}_1, \tilde{o}_2)$ between two dummy operations (because the infinite length of the corresponding slice cannot be decreased down to zero). Thus, there are no more than ξ "real" slices in schedule $\tilde{S}$ (and so, no more than $\xi - 2$ preemption dates, because there cannot be interruptions at the end of the last but one slice, as well as at the end of the last one). Using the upper bound $\theta m - \xi$ on the number of interruptions (from page 18), we derive that schedule $\tilde{S}$ contains no more than ξ interruptions.

Finally, we show that schedule $\tilde{S}$ cannot contain interruptions that occur at nonintegral points in time, since otherwise there must exist a cycle in $\tilde{G}$. Let E' denote the set of edges corresponding to slices with nonintegral length. Suppose that $E' \neq \varnothing$. Since the processing times of all operations are integral, each vertex $o_i \in \mathcal{O}$ incident to an edge $e \in E'$ is incident to at least one more edge $e' \in E'$. Therefore, the set of edges E' either contains a cycle, or contains a path starting at one dummy operation $(\tilde{o}_1)$ and ending at another dummy operation $(\tilde{o}_2)$. In the latter case we also obtain a cycle in $\tilde{G}$, because two dummy operations are connected in $\tilde{G}$ by the edge $(\tilde{o}_1, \tilde{o}_2) \in E$. □

Theorem 4.14 *For every instance of the preemptive dag shop problem with two jobs, ξ operations with integer processing times and a regular quasiconcave objective function of the operation set completion times $F(C)$ there exists an optimal schedule in which:*

1. *there are at most ξ slices and at most ξ interruptions;*

2. *all interruptions occur at integral dates.*

We skip the proof of this theorem since it is almost identical to the previous one. The main observation we use is that there are at most two operations processed at any point in time in any feasible schedule. Therefore, we can build a similar graph G with operations of the first job on one side and operations of the second job on another side, and the rest of the argument is the same.

We now construct an instance of the dag shop scheduling problem with three machines (M_1, M_2, M_3), three jobs (J_1, J_2, J_3) and two operations per job such that each optimal schedule unavoidably contains interruptions at nonintegral points in time.

Job J_1 has two operations: the first operation must be processed on the first machine and has unit processing time, the second operation must be processed on the second machine and has processing time 2. This job is of the job shop type, *i.e.*, the second operation must be processed strictly after the first one.

Job J_2 is also of the job shop type and has two operations. The first operation must be processed on the first machine and the second operation must be processed on the third machine. Both operations have unit length.

Job J_3 is of the open shop type, *i.e.*, there are no precedence relations between two operations of this job. One operation must be processed on the second machine and has unit processing time, while the other operation must be processed on the third machine and has processing time 2.

The objective function is the makespan, *i.e.*, the maximum operation completion time. We claim that 3.5 is a lower bound on the length of the optimal schedule. Indeed, in every feasible schedule after time 1 job J_1 has at least two units of processing, job J_2 — at least one unit of processing, and job J_3 — at least two units of processing on the second and third machines. Therefore, the second and the third machines must process at least five units after time 1. So, $3.5 = 1 + 5/2$ is a lower bound for the makespan. A feasible schedule with makespan equal to this lower bound is depicted on Figure 1. It is also clear that there is no feasible schedule for this problem instance with makespan 3.5 without fractional interruptions.

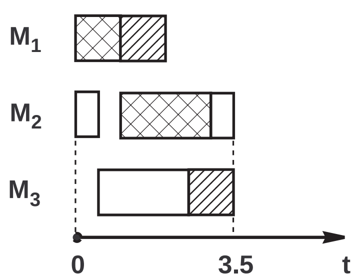

Figure 1: A dag shop instance with non-integral optimal interruptions.

Note 4.15 Almost all results proved in section 4 (except Theorem 4.11) can be easily extended to a more general problem with precedence and compatibility constraints specified on the set of all operations (a kind of a project scheduling problem).

5 Rational Structure Theorem

In this section we again consider the most general **GP**-problem, yet with more special objective functions. However, these special functions enable us to derive the *Rational Structure Theorem* for the **GP**-problem with almost all classic objective functions.

To simplify notation, we will further assume that $\tau_1 = 0$.

Let N piece-wise linear functions $f_i(x)$ $(i = 1, \dots, N)$ be defined for all $x \in [0, \tau_D]$ as

$$f_i(x) = \begin{cases} \alpha_{i0}, & \text{for } x = 0, \\ \alpha_{ik} + \beta_{ik}x, & \text{for } x \in (\tau_k, \tau_{k+1}],\ k = 1, \dots, D-1. \end{cases} \tag{21}$$

We assume all functions $f_i(x)$ to be nondecreasing, which is guaranteed by the relations:

$$\beta_{ik} \geq 0,\ \forall\ i, k; \quad \alpha_{i1} \geq \alpha_{i0},\ \forall\ i; \quad \alpha_{ik} - \alpha_{i,k+1} \leq (\beta_{i,k+1} - \beta_{i,k})\tau_{k+1},\ \forall\ i, k.$$

In our first theorem, we consider *additive piece-wise linear nondecreasing* (*APLN* — for short) objective functions $F(C) = F(C(B_1), \dots, C(B_N))$ depending on the vector C of the operation set completion times as

$$F(C) \doteq \sum_{i=1}^{N} f_i(C(B_i)). \tag{22}$$

As was already mentioned in the Introduction the following Rational Structure Theorem is a generalization of the results from [15, 16] for parallel machine scheduling with precedence constraints.

Theorem 5.1 (Rational Structure Theorem) *Suppose, we consider a* **GP***-problem with ξ operations, D fixed dates, and an APLN objective function $F(C(B_1), \dots, C(B_N))$ of operation set completion times, defined by (21) and (22). If for a given instance of this problem there exist feasible schedules, then there exists an optimal schedule with the following properties:*

1. *There are at most $\xi^2 + \xi + D - 1$ positive slices. If all $p_{ij} = 0$ in constraints (1), i.e., there are no precedence delays, then there are at most $\xi + D - 1$ positive slices.*

2. *If all processing times, mode speeds, and fixed dates are integers, then there exists a rational number δ with polynomial encoding length such that each changeover date g_ν is an integer multiple of δ.*

3. *If, in addition to the above conditions, all coefficients $\{\alpha_{ik}, \beta_{ik}\}$ of the objective function (21),(22) are integers, then the optimal value of the objective function is an integer multiple of δ.*

Proof. Suppose, we are given an instance of the **GP**-problem, and the set of its feasible schedules is nonempty. It is clear that our APLN objective function, being considered as a function of operation completion times, is regular. Hence, by Theorem 2.4, it has an optimal schedule S_0, and due to Theorem 2.3, we may assume that this schedule has at most $\theta(S_0) \leq (2\xi + D - 1)H + 2\xi$ slices. Let $\mathfrak{A}(S_0) = (\pi(S_0), \Theta(S_0))$ and $\mathfrak{B}(S_0) = (\pi(S_0), \Theta(S_0), L(S_0))$

be its short and full configurations defined on page 12, where $L(S_0) = (l_1(S_0), \dots, l_{\theta(S_0)}(S_0))$ is the vector of lengths of slices of schedule S_0 in the order of their implementation. Due to the feasibility of schedule S_0, configuration $\mathfrak{B}(S_0)$ meets relations (2)–(5).

Let $\mathcal{S}' = \{S'\}$ be the set of feasible schedules with the short configuration $\mathfrak{A}(S') = \mathfrak{A}(S_0)$ and such that $l_\nu(S') = 0$ for any ν such that $l_\nu(S_0) = 0$. We know that $\mathcal{S}' \neq \varnothing$.

Note that for any fixed and feasible short configuration, all parameters in relations (3), (4), (5), except variables $\{l_\nu\}$, become fixed. Furthermore, for all $S' \in \mathcal{S}'$ each operation set completion time $C(B_i, S')$ remains within a fixed interval $[\tau_k, \tau_{k+1}]$. More precisely, if $C(B_i, S_0) = \tau_k$ for some k, then $C(B_i, S') = \tau_k$ as well, and if $C(B_i, S_0) \in (\tau_k, \tau_{k+1})$, then $C(B_i, S') \in [\tau_k, \tau_{k+1}]$. To our purposes we can derive that for the operation set B_i ($i = 1, \dots, N$), in any schedule $S' \in \mathcal{S}'$ we may use either the same penalty function $\alpha_{ik} + \beta_{ik} C(B_i, S')$ (as used for B_i in schedule S_0), or the value $\alpha_{i,k-1} + \beta_{i,k-1}\tau_k$ of the penalty function used in the previous interval $(\tau_{k-1}, \tau_k]$ (with a possible jumping the penalty down), — in the case when $C(B_i, S')$ decreases down to τ_k.

Let us consider the optimization problem $\mathbf{P}$ with variables $\{l_\nu\}$ corresponding only to positive values $l_\nu(S_0)$, and with linear constraints (3), (4), (5). The objective is to minimize the function $F'(C(S')) \doteq \sum_{i=1}^{N} f_i'(C(B_i, S'))$ over all schedules $S' \in \mathcal{S}'$, where each function $f_i'(x)$ is determined by the value of the completion time of set B_i in schedule S_0:

$$f_i'(x) = \alpha_{ik(i)} + \beta_{ik(i)} x, \text{for } C(B_i, S_0) \in (\tau_{k(i)}, \tau_{k(i)+1}].$$

If $C(B_i, S_0) = 0$ we assume $k(i) = 0$ and $\beta_{i0} = 0$.

As was noted above, in the case that $C(B_i, S') = \tau_k$, while $C(B_i, S_0) \in (\tau_k, \tau_{k+1}]$, we may accidentally have

$$f_i(C(B_i, S')) < f_i'(C(B_i, S')) \tag{23}$$

that means that we use a different penalty function for the set B_i in problem $\mathbf{P}$. In general, we have

$$F(C(S')) \le F'(C(S')), \ \forall\, S' \in \mathcal{S}'. \tag{24}$$

Since the short configuration is fixed for $S' \in \mathcal{S}'$, for each set B_i we have also fixed the number η_i of the last slice $\hat{S}_\nu$, where an operation $o_j \in B_i$ is presented. So, we have $C(B_i, S') = \sum_{\nu=1}^{\eta_i} l_\nu(S'), \ \forall\, S' \in \mathcal{S}'$, and hence,

$$\tilde{F}(L) \doteq F'(C(S')) = \sum_{i=1}^{N} \alpha_{ik(i)} + \sum_{i=1}^{N} \beta_{ik(i)} \sum_{\nu=1}^{\eta_i} l_\nu(S') = \alpha_0 + \sum_{\nu} \beta_\nu l_\nu(S'), \tag{25}$$

where $\beta_\nu = \sum_{\{i \,|\, \eta_i \ge \nu\}} \beta_{ik(i)}$, and α_0 is a constant. Thus, problem $\mathbf{P}$ is a linear programming problem.

It is known that there exists a *basic optimal solution* $\{l_\nu^*\}$ of the above LP program [17], in which the number (θ^*) of positive values of variables l_ν is upper bounded by the rank $r(A)$ of the matrix A of coefficients in the left-hand side of relations (3), (4), (5). The rank $r(A)$ is upper bounded by the number $m + \xi + D - 1$ of constraints (3), (4), (5) where m is the number of arcs in graph G, or the number of conditions (5). Since graph G admits counter arcs (o_i, o_j), (o_j, o_i) for any two operations $\{o_i, o_j\}$, as well as loops (o_i, o_i), m may be as large as ξ^2. If all $p_{ij} = 0$ then we can drop all constraints (5) since in this case all precedence constraints can be enforced by the order between slices and therefore $r(A)$ can be upper bounded by $\xi + D - 1$.

Let $L^* = (l_1^*, \dots, l_{\theta(S_0)}^*)$ be the complete vector of values of variables $\{l_\nu \mid \nu = 1, \dots, \theta(S_0)\}$, including zero values $l_\nu(S_0)$ omitted in the above LP program. Once the triplet $(\pi(S_0), \Theta(S_0), L^*)$ meets all requirements (2)–(5), it specifies a feasible schedule S^* with no greater value of the objective function: $F'(C(S^*)) \le F'(C(S_0))$. Hence, by means of (24), we obtain

$$F(C(S^*)) \le F'(C(S^*)) \le F'(C(S_0)) = F(C(S_0)), \tag{26}$$

which means that schedule S^* is also optimal. It also implies that inequality (23) is impossible for any set B_i and schedule $S' = S^*$. The number of positive slices of schedule S^* is equal to $\theta^* \le \xi^2 + \xi + D - 1$ (or $\xi + D - 1$ if all $p_{ij} = 0$).

The second and third statements in the theorem follow from the standard bounds on the encoding length of optimal basic solutions of linear programs with integer coefficients [17].

□

It can be easily seen that various total weighted penalty functions such as *total weighted lateness*, *tardiness*, *flow time*, and even discontinuous *weighted number of late jobs* $\sum w_j B_j$ can be included into the APLN objective function framework defined by (21)–(22). Furthermore, *makespan* $C_{\max}$ is a trivial special case of the function $F(C)$, when there is a single set B_1, and $f_1(x) = x$. At the same time, objective functions such as the maximum lateness and the maximum tardiness cannot be directly represented by APLN functions. However, we can show that the **GP**-problem with an objective function of the max-type also satisfies the properties of the Rational Structure Theorem.

Indeed, consider the **GP**-problem with the objective function

$$F(C(S')) = \max f_i(C(B_i, S')), \tag{27}$$

where functions f_i $(i = 1, \dots, N)$ are still defined by (21). Assuming that an optimal schedule S_0 exists, the optimum F^* is attained at some set B_{i^*} with the completion time $C(B_{i^*}, S_0)$ within an interval $(\tau_{k^*}, \tau_{k^*+1}]$. Let us consider now problem **P** with constraints (3), (4), (5) where all parameters, except variables $\{l_\nu\}$, are specified by the short configuration $\mathfrak{A}(S_0)$ of schedule S_0 and with the objective to minimize the function $\alpha_{i^*k^*} + \beta_{i^*k^*} C(B_{i^*}, S')$. To guarantee that the minimum of this function provides the optimum for the original **GP**-problem, it suffices to add $N - 1$ constraints of the type:

$$\alpha_{ik(i)} + \beta_{ik(i)} C(B_i, S') \le F^*, \ i \ne i^*, \tag{28}$$

where $k(i)$ for each set B_i is determined by the relation $C(B_i, S_0) \in (\tau_{k(i)}, \tau_{k(i)+1}]$. Clearly, the set $\mathcal{S}'$ of schedules $\{S'\}$ satisfying constraints (3), (4), (5), (28) is nonempty because $S_0 \in \mathcal{S}'$. Using the expressions $C(B_i, S') = \sum_{\nu=1}^{\eta_i} l_\nu(S')$ for fixed $\{\eta_i\}$, we can represent constraints (28) as linear inequalities in terms of variables $\{l_\nu\}$. Therefore, problem **P** is a linear program, and its optimum provides the optimum for the original **GP**-problem. Once we add $N - 1$ linear constraints to our LP problem, the number of positive slices in the optimal schedule may increase by this amount. As a result of all these arguments, we can formulate the following

Theorem 5.2 (Rational Structure Theorem II) *Suppose, we consider a* **GP***-problem with ξ operations, D fixed dates, and an objective function $F(C(B_1), \ldots, C(B_N))$ of the operation set completion times, defined by (21) and (27). If for a given instance of this problem there exist feasible schedules, then there exists an optimal schedule with the following properties:*

1. *There are at most $\xi^2 + \xi + D + N - 2$ positive slices. If all $p_{ij} = 0$, then there are at most $\xi + D + N - 2$ positive slices.*

2. *If all input data are integers, then there exists a rational number δ with polynomially bounded encoding length such that each changeover date g_ν is an integer multiple of δ and the optimal value of the objective function is an integer multiple of δ.*

6 Concluding remarks

In section 5 for the **GP**-problem and a wide variety of objective functions we established the existence of optimal schedules with some nice properties provided that for the given instance there exist feasible schedules. The presented proofs (especially, that of the first Rational Structure Theorem) are constructive. Indeed, to find the desired optimal schedule, it suffices to enumerate various short configurations, and for each configuration to solve the corresponding linear program. Although, it is quite clear that the number of short configurations is huge, the results obtained at least demonstrate that the complexity of the problem under consideration is no more than that huge number.

For the second Rational Structure Theorem we presented a shorter but somewhat less constructive proof (for it assumes the value of the optimum to be known). However, it is clear that a similar constructive way of proof is also possible.

For the two-machine (two-job) dag shop scheduling problem, we have not succeeded either to prove, or disprove that there always exists an optimal schedule in which the set of changeover dates coincides with the set of completion times of operations. This "exercise" is left to the reader. The same question remains open for the m-machine open shop problem.

References

[1] P. Baptiste, P. Brucker, S. Knust and V. Timkovsky, Ten notes on equal-processing-time scheduling: at the frontiers of solvability in polynomial time. 4OR 2 (2004), 111–127.

[2] Ph. Baptiste and V. Timkovsky (2001). On Preemption Redundancy in Scheduling Unit Processing Time Jobs on Two Parallel Machines. *Operations Research Letters* **28**, 205–212.

[3] P. Brucker, S. Heitmann and J. Hurink. How Useful are Preemptive Schedules? Oper. Res. Lett. 31 (2003), no. 2, 129–136.

[4] J. Du and J. Y. T. Leung (1993). Minimizing mean flow time in two-machine open shops and flow shops. *J. Algorithms* **14**, 24–44.

[5] J. Du, Y. Leung and G. Young, Scheduling chain-structured tasks to minimize makespan and mean flow time. Inform. and Comput. 92 (1991), 219–236.

[6] T. Gonzalez and S. Sahni (1976). Open Shop Scheduling to Minimize Finish Time. *Journal of the ACM* **23**, 665–679.

[7] T. Gonzalez and S. Sahni (1978). Preemptive Scheduling of Uniform Processor Systems. *Journal of the ACM* **25**, 92–101.

[8] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics* **5**, 287–326.

[9] J. Labetoulle, E.L. Lawler, J.K. Lenstra and A. H. G. Rinnooy Kan (1984). Preemptive scheduling of uniform machines subject to release dates. In: W. R. Pulleyblank (Ed.), *Progress in Combinatorial Optimization*, Academic Press, 245–261.

[10] E.L. Lawler and J. Labetoulle (1978). On Preemptive Scheduling of Unrelated Parallel Processors by Linear Programming. *Journal of the ACM* **25**, 612–619.

[11] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys (1993). Sequencing and Scheduling: Algorithms and Complexity. In: S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin (Eds.), *Logistics of Production and Inventory*, Handbooks in Operations Research and Management Science 4, North–Holland, 445–522.

[12] R. McNaughton (1959). Scheduling with deadlines and loss functions. *Management Science* **6**, 1–12.

[13] M. Queyranne and M. Sviridenko (2002). Approximation Algorithms for Shop Scheduling Problems with Minsum Objective. *Journal of Scheduling* **5**, 287–305.

[14] M. Queyranne and M. Sviridenko (2002). A $(2+\varepsilon)$-Approximation Algorithm for Generalized Preemptive Open Shop Problem with Minsum Objective. Journal of Algorithms **45**, 202–212. Preliminary version appeared in: K. Aardal and B. Gerards (Eds.), *Integer Programming and Combinatorial Optimization (Proceedings of the 8th International IPCO Conference)*, Lecture Notes in Computer Science 2081, Springer, 361–369.

[15] N. Sauer and M. Stone, Preemptive scheduling. Algorithms and order (Ottawa, ON, 1987), 307–323, Kluwer Acad. Publ., Dordrecht, 1989.

[16] N. Sauer and M. Stone (1987). Rational preemptive scheduling. *Order* **4** (2), 195–206.

[17] A. Schrijver (1986). *Theory of Linear and Integer Programming*, Wiley.

[18] V.S. Tanaev, V.S. Gordon and Y. M. Shafransky (1994). *Scheduling theory. Single-stage systems*, Kluwer.

[19] V. Timkovsky, Identical parallel machines vs. unit-time shops and preemptions vs. chains in scheduling complexity. European J. Oper. Res. 149 (2003), 355–376.

[20] G. Woeginger, On the approximability of average completion time scheduling under precedence constraints. Discrete Appl. Math. 131 (2003), no. 1, 237–252.