Mixed Integer Programming Approach to Multiprocessor Job Scheduling with Setup Times

Eremeev A.V. and Kovalenko Yu.V.

Sobolev Institute of Mathematics,
4, Akad. Koptyug avenue, 630090, Novosibirsk, Russia.
Emails: eremeev@ofim.oscsbras.ru, julia.kovalenko.ya@yandex.ru

Abstract

Multiprocessor jobs require more than one processor at the same moment of time. We consider two basic variants of scheduling multiprocessor jobs with various regular criteria. In the first variant, for each job the number of required processors is given and fixed, and the job can be processed on any subset of parallel processors of this size. In the second variant, the subset of dedicated processors required by a job is given and fixed. A sequence dependent setup time is needed between different jobs. We formulate mixed integer linear programming models based on a continuous time representation for the NP-hard scheduling problems under consideration. Using these models, we identify new polynomially solvable cases.

Keywords: multiprocessor job, setup time, integer linear programming, polynomial solvability

1 Introduction

We consider the multiprocessor scheduling problem, where a set of k jobs $\mathcal{J} = \{1, \dots, k\}$ has to be executed by m processors such that each processor can work on at most one job at a time, and each job must be processed simultaneously by several processors. Let $\mathcal{M} = \{1, \dots, m\}$ and let p_j denote the processing time of job j for each $j \in \mathcal{J}$. Dedicated and parallel variants of the problem are studied here. In the first variant, there is a size $size_j$ associated with each job $j \in \mathcal{J}$ indicating that the job can be processed on any subset of parallel processors of the given size. In the second variant, each job $j \in \mathcal{J}$ requires a simultaneous use of a pre-specified subset (mode) fix_j of dedicated processors. Following the traditional definitions in scheduling theory [1] we consider rigid jobs in the first variant and single mode multiprocessor jobs in the second variant.

A sequence dependent setup time is required to switch a processor from one job to another. For the parallel model of the problem let $s_{jj'}$ be the non-negative setup time from job j to job j', where $j,\ j' \in \mathcal{J}$. For the dedicated variant of the problem let $s^l_{jj'}$ denote the setup time from job j to job j' on processor l, $l \in \mathcal{M}$, $j,\ j' \in F_l$, where $F_l = \{j \in \mathcal{J} :\ l \in fix_j\}$ is the set of jobs that use processor $l \in \mathcal{M}$.

Following the traditional thee-field notation for scheduling problems [1], we denote preemptive version of the problem with single mode multiprocessor jobs (rigid jobs) as $P|fix_j, pmtn, s^l_{jj'}|\gamma$ ($P|size_j, pmtn, s_{jj'}|\gamma$) and non-preemptive version of the problem is denoted by $P|fix_j, s^l_{jj'}|\gamma$ ($P|size_j, s_{jj'}|\gamma$). Here γ specifies an objective function. It is assumed that the subset of processors used by a rigid job can be changed at runtime in preemptive scheduling.

We consider four widely used objective functions to be minimized: the makespan, $C_{\max} = \max_{j\in\mathcal{J}} C_j$, the maximum lateness, $L_{\max} = \max_{j\in\mathcal{J}}(C_j - d_j)$, the sum of completion times, $C_{\sum} = \sum_{j\in\mathcal{J}} C_j$, the sum of latenesses, $L_{\sum} = \sum_{j\in\mathcal{J}}(C_j - d_j)$, where C_j denote the completion time of job $j \in \mathcal{J}$, and d_j is the due date of job $j \in \mathcal{J}$, i.e. the time by which job j should be completed.

In practice one often may assume that the setup times satisfy the triangle inequality:

$$s^l_{j'',j'} \le s^l_{j'',j} + s^l_{j,j'},\ j,\ j',\ j'' \in F_l,\ l \in \mathcal{M} \text{ (for dedicated processors)}, \tag{1}$$

$$s_{j'',j'} \le s_{j'',j} + s_{j,j'},\ j,\ j',\ j'' \in \mathcal{J} \text{ (for parallel processors)}. \tag{2}$$

We denote this special case by placing Δ in front of $s^l_{jj'}$ (or $s_{jj'}$) in the second field of the three-field notation. In the special case where the number of machines $m = 1$ is not a part of the input, there is no difference between rigid jobs and single mode multiprocessor jobs and the problem notation simplifies to $1|pmtn, \Delta s_{jj'}|\gamma$ and $1|\Delta s_{jj'}|\gamma$ for preemptive and non-preemptive jobs respectively.

All mentioned above problems $P|fix_j, pmtn, \Delta s^l_{jj'}|\gamma$, $P|fix_j, \Delta s^l_{jj'}|\gamma$, $P|size_j, pmtn, \Delta s_{jj'}|\gamma$ and $P|size_j, \Delta s_{jj'}|\gamma$ with $\gamma \in \{C_{\max}, L_{\max}, C_{\sum}, L_{\sum}\}$ are NP hard even in the single-machine case as implied by the following proposition.

Proposition 1 *Problems* $1|pmtn, \Delta s_{jj'}|\gamma$ *and* $1|\Delta s_{jj'}|\gamma$ *with* $\gamma \in \{C_{\max}, L_{\max}, C_{\sum}, L_{\sum}\}$ *are strongly NP-hard.*

Proposition 1 may be attributed to the “folklore”. However for the sake of completeness we provide its proof in the appendix.

In [2], the problem of scheduling multiprocessor jobs with sequence-dependent setup times was combined with a lot-sizing problem where it is required to produce a set of the products in demanded volumes. In this setting, the multiprocessor jobs were called *multi-machine technologies*, each technology engaging a number of machines simultaneously to produce a batch of some product. In [2], this problem with $C_{\max}$ criterion was shown to be hard to approximate and new NP-hard and polynomially solvable special cases were identified.

A MIP model was proposed by M.A. Shaik et al. in [10] for a problem of scheduling multi-machine technologies with sequence-dependent setup times for continuous production plants. Besides that, a decomposition method was developed in [10] to solve real-life problems from chemical industry where straightforward application of the MIP model was impractical.

A survey of results on multiprocessor jobs scheduling in the case of zero setup times is provided by M. Drozdowski in [1]. It is known that in this case problem

$P|size_j, pmtn|C_{\max}$ with the number of processors bounded above by a constant (denoted $Pm|size_j, pmtn|C_{\max}$) and problem $P|fix_j, pmtn|C_{\max}$ with only two-processor jobs are both polynomially solvable. This result is based on the fact that the considered special cases may be treated as linear programming (LP) problems using the so-called *configurations* introduced by K. Jansen and L. Porkolab [7, 8] and resembling *patterns* proposed by L.V. Kantorovich and V.A. Zalgaller for the one-dimensional cutting-stock problem in the middle of 20-th century (see e.g. [9]). In [7, 8] a configuration is defined as a set of jobs which may be processed simultaneously. The total number of configurations is $O(k^m)$ and the resulting LP problems contain $O(k^m)$ variables, each one representing the time of using the corresponding configuration in a schedule, and $O(k)$ constraints. These problems may be solved in polynomial time by considering the dual LP problems and applying the ellipsoid method of M. Grötschel, L. Lovász and A. Schrijver. [4].

Unfortunately the configurations-based approach can not be extended to the case of sequence-dependent setup times because in the general case evaluation of objective function requires not only the durations but also the sequence of jobs on each machine. In the present paper, we develop the MIP models for multiprocessor jobs scheduling with sequence-dependent setup times using the notion of *event points*, which was originally proposed in the context of single-processor jobs by M.G. Ierapetritou and C.A. Floudas in [6]. In the case of multiprocessor jobs, an event point, as well as a configuration, corresponds to some set of compatible jobs, but in contrast to the set of jobs of a configuration (which is defined a priori) the set of jobs of an event point is defined by the values of Boolean variables of this event point. The presence of Boolean variables allows to account for the sequence dependent setup times. Besides that, unlike the jobs of a configuration, the jobs of an event point may have different starting and completion times.

The MIP models proposed on the basis of event points are described in Section 2. Using these models, the new polynomially solvable cases with sequence-dependent setup times are identified in Section 3. The concluding remarks are provided in Section 4.

2 Mixed Integer Linear Programming Models

2.1 Single Mode Multiprocessor Jobs

Let us define the notion of *event points* analogously to [2, 6]. By event point we will mean a subset of variables in mixed integer linear programming (MIP) model, which characterize a selection of a certain set of jobs and their starting and completion times. In one event point each processor may be utilized in at most one job. The set of all event points will be denoted by $N = \{1, \dots, n_{\max}\}$, where the parameter $n_{\max}$ is chosen sufficiently large on the basis of a prior estimates or preliminary experiments.

The structure of the schedule is defined by the Boolean variables w_{jn} such that $w_{jn} = 1$ if job j is executed in event point n, and $w_{jn} = 0$ otherwise. In case job j is executed in event point n, the staring time and the completion time of job j in this event point are given by the real-valued variables T^{st}_{jn} and T^{f}_{jn} accordingly.

Let H be an upper bound on the schedule length,

$$H := \sum_{j\in\mathcal{J}} p_j + (k-1)\cdot \max_{l\in\mathcal{M},\ j\neq j'\in F_l}\{s^l_{jj'}\}.$$

Then the set of feasible solutions for problem $P|fix_j, pmtn, s^l_{jj'}|\gamma$ is defined as follows

$$\sum_{j\in F_l} w_{jn} \leq 1,\ l\in\mathcal{M},\ n\in N, \tag{3}$$

$$T^f_{jn} \geq T^{st}_{jn},\ j\in\mathcal{J},\ n\in N, \tag{4}$$

$$T^{st}_{jn} \geq T^f_{j'n'} + s^l_{j'j} - H(2 - w_{jn} - w_{j'n'} + \sum_{\tilde{j}\in F_l}\sum_{n'<\tilde{n}<n} w_{\tilde{j}\tilde{n}}), \tag{5}$$

$$l\in\mathcal{M},\ j,j'\in F_l,\ n,n'\in N,\ n\neq 1, n'<n,$$

$$T^f_{jn} - T^{st}_{jn} \leq w_{jn}\cdot p_j,\ j\in\mathcal{J},\ n\in N, \tag{6}$$

$$\sum_{n\in N}\frac{T^f_{jn} - T^{st}_{jn}}{p_j} \geq 1,\ j\in\mathcal{J}, \tag{7}$$

$$w_{jn}\in\{0,1\},\ T^{st}_{jn}\geq 0,\ j\in\mathcal{J},\ n\in N. \tag{8}$$

Constraint (3) implies that in any event point on processor l at most one job may be executed. Constraint (5) indicates that the starting time of job j on processor l should not be less than the completion time of a preceding job on the same processor, plus the setup time. Constraint (4) guarantees that all jobs may be performed only for non-negative time. If a job j is not executed in the event point n (i.e. $w_{jn} = 0$) then its duration should be zero – this is ensured by inequality (6). Constraint (7) implies that each job $j\in\mathcal{J}$ is entirely executed. Constraints (8) give the area where the variables are defined.

The set of feasible solutions for problem $P|fix_j, s^l_{jj'}|\gamma$ may be obtained from (3)–(8) by adding the inequality

$$\sum_{n\in N} w_{jn} \leq 1,\ j\in\mathcal{J}, \tag{9}$$

which ensures each job is executed without preemptions.

The optimization criteria for presented models are formulated in the following form.

1. Makespan

$$C_{\max}\to\min,$$

$$C_{\max} \geq T^f_{jn},\ j\in\mathcal{J},\ n\in N.$$

2. Sum of completion times

$$\sum_{j\in\mathcal{J}} T^f_j \to\min,$$

$$T^f_j \geq T^f_{jn},\ j\in\mathcal{J},\ n\in N.$$

3. Maximum lateness

$$L_{\max}\to\min,$$

$$L_{\max} \geq T^f_{jn} - d_j,\ j\in\mathcal{J},\ n\in N.$$

4. Sum of latenesses

$$\sum_{j\in\mathcal{J}} L_j \to \min,$$

$$L_j \geq T^f_{jn} - d_j,\ j \in \mathcal{J},\ n \in N.$$

2.2 Rigid Jobs

MIP models for problems with rigid jobs are constructed based on the same principles as the previous models. However, in this case, if only the jobs are allocated in the event points, then a problem of assignment of the jobs to processors arises. Mainly, this assignment is needed to calculate the setup times between jobs on processors. The following proposition shows that such assignment problems with criteria $C_{\max}$ and $L_{\max}$ are NP-hard.

Proposition 2 *Suppose a family of subsets of jobs $\{\mathcal{J}_1, \ldots, \mathcal{J}_{n_{\max}}\}$ is given as a part of the problem input. Then problems $P2|size_j, pmtn, s_{jj'}|\gamma$ and $P2|size_j, s_{jj'}|\gamma$ with $\gamma \in \{C_{\max}, L_{\max}\}$, under additional constraint that n-th job processed on each machine belongs to $\mathcal{J}_n$, $n = 1, \ldots, n_{\max}$, is NP-hard.*

Proof. The hardness of considered problems can be shown by a polynomial reduction of Ordered Partition problem, which is known to be NP-complete [3]. Ordered Partition problem is formulated as follows: Let an ordered set $A = \{a_1, a_2, \ldots, a_{2k_0}\}$ be given. A positive integer e_i is associated with each element $a_i \in A,\ i = 1, \ldots, k$, such that $\sum\limits_{a_i\in A} e_i = 2E$. Ordered Partition problem asks if there exists a partition of A into two subsets A_1 and A_2 such that $\sum\limits_{a_i\in A_1} e_i = \sum\limits_{a_i\in A_2} e_i = E$, $|A_1| = |A_2| = k_0$ and set A_1 includes exactly one element from each pair $a_{2i-1},\ a_{2i},\ i = 1, \ldots, k_0$.

For brevity we will denote $P|size_j, pmtn, s_{jj'}|C_{\max}$ and $P|size_j, s_{jj'}|C_{\max}$ problems under additional constraint that n-th job processed on each machine belongs to $\mathcal{J}_n,\ n = 1, \ldots, n_{\max}$, by P1 and P2 respectively.

We reduce an Ordered Partition instance to instances of problems P1 and P2 as follows. Put the number of jobs $k := 2k_0$; the number of processors $m := 2$; $size_j := 1$, $p_j := e_j$ and $s_{jj'} := 0$ for all $j \neq j' \in \mathcal{J}$. Besides that we define a family of subsets of jobs assuming $n_{\max} := k_0$ and $\mathcal{J}_n := \{2n-1, 2n\}$ for $n = 1, \ldots, k_0$.

Consider the decision versions of problems P1 and P2 which ask if there is a schedule with makespan $C_{\max} \leq K$ for a given K.

Note that in the instances of P1 and P2 defined above, each job j belongs only to one subset $\mathcal{J}_n$ and $\sum_{j\in\mathcal{J}} p_j = 2E$. Hence, in a schedule with $C_{\max} \leq K$, assuming $K := E$, all jobs are executed without preemptions on one of the two processors, because overall available time on two processors does not exceed $2E$. Therefore, a positive answer to an instance of decision problem P1 or P2 implies a positive answer to the Ordered Partition problem and vice versa.

In the case of criterion $L_{\max}$, the statement of the proposition holds because criteria $L_{\max}$ and $C_{\max}$ are equivalent when $d_j = 0$ for all $j \in \mathcal{J}$. ∎

In view of Proposition 2, in the case of rigid jobs we formulate our MIP models in such a way that both jobs and processors, on which they are executed, are assigned in the event points.

The structure of the schedule is also defined by the Boolean variables w_{jn} and the real-valued variables T_{jn}^{st} and T_{jn}^{f}, which have the same meaning as in the case of single mode multiprocessor jobs. Moreover, we include additional Boolean variables z_{jln} such that $z_{jln} = 1$ if job j is executed in event point n and uses processor l, and $z_{jln} = 0$ otherwise.

Let H be an upper bound on schedule length. It suffices to put

$$H = \sum_{j \in \mathcal{J}} p_j + (k-1) \cdot \max_{j \neq j'} \{s_{jj'}\}.$$

Based on the above remarks and variables, the set of feasible solutions for problem $P|size_j, pmtn, s_{jj'}|\gamma$ is defined by the following constraints:

$$\sum_{j \in \mathcal{J}} z_{jln} \leq 1,\ l \in \mathcal{M},\ n \in N, \tag{10}$$

$$\sum_{l \in \mathcal{M}} z_{jln} = size_j \cdot w_{jn},\ j \in \mathcal{J},\ n \in N, \tag{11}$$

$$T_{jn}^{f} \geq T_{jn}^{st},\ j \in \mathcal{J},\ n \in N, \tag{12}$$

$$T_{jn}^{st} \geq T_{j'n'}^{f} + s_{j'j} - H(2 - z_{jln} - z_{j'ln'} + \sum_{\tilde{j} \in \mathcal{J}} \sum_{n' < \tilde{n} < n} z_{\tilde{j}l\tilde{n}}), \tag{13}$$

$$l \in \mathcal{M},\ j \neq \tilde{j} \in \mathcal{J},\ n, n' \in N,\ n \neq 1, n' < n,$$

$$T_{jn}^{f} - T_{jn}^{st} \leq w_{jn} \cdot p_j,\ j \in \mathcal{J},\ n \in N, \tag{14}$$

$$\sum_{n \in N} \frac{T_{jn}^{f} - T_{jn}^{st}}{p_j} \geq 1,\ j \in \mathcal{J}, \tag{15}$$

$$z_{jln} \in \{0,1\},\ w_{jn} \in \{0,1\},\ T_{jn}^{st} \geq 0,\ j \in \mathcal{J},\ l \in \mathcal{M},\ n \in N. \tag{16}$$

Constraints (10), (12)–(15) have the same interpretation as in the model for problem $P|fix_j, pmtn, s_{jj'}^{l}|\gamma$. Constraint (11) guarantees that job j uses exactly $size_j$ processors if it is executed in the event point n (i.e. $w_{jn} = 1$).

The set of feasible solutions for problem $P|size_j, s_{jj'}|\gamma$ may be obtained from (10)–(16) by adding inequality (9). The optimization criteria are modeled as in the case of dedicated processors.

3 Polynomially Solvable Cases

New polynomially solvable special cases with non-zero setup times are found using proposed MIP models, under assumption that the number of jobs is bounded by a constant. An instance of a multiprocessor job scheduling problem is reduced to a number of instances of a linear programming problem, obtained from the MIP model assigning some fixed values to Boolean variables.

3.1 Single Mode Multiprocessor Jobs

In order to find an optimal solution to $P|fix_j, s^l_{jj'}|\gamma$ using model (3)–(9), it is sufficient to assume $n_{\max} = k$ because the preemptions are not allowed. Denote $\mathcal{P}_{\mathit{fix}}$ the linear programming problem obtained by fixing all Boolean variables (w_{jn}) in model (3)–(9) supplemented by a linear programming formulation of optimization criterion γ. Here and below by *fixing* of the variables we assume assignment of some fixed values to them (which turns these variables into parameters). Problem $\mathcal{P}_{\mathit{fix}}$ with $n_{\max} = k$ involves a polynomially bounded number of variables and constraints, which means it is polynomially solvable (see e.g. [4]).

Let τ_{fix} be an upper bound on the time complexity of solving problem $\mathcal{P}_{\mathit{fix}}$. The problem $P|fix_j, s^l_{jj'}|\gamma$, where the number of jobs is bounded from above by a constant, we denote by $P|fix_j, s^l_{jj'}, k = \text{const}|\gamma$. This problem reduces to $(n_{\max})^k$ problems of $\mathcal{P}_{\mathit{fix}}$ type with $n_{\max} = k$. Therefore the following theorem holds.

Theorem 1 *Problem $P|fix_j, s^l_{jj'}, k = \text{const}|\gamma$, $\gamma \in \{C_{\max}, C_{\sum}, L_{\max}, L_{\sum}\}$, is polynomially solvable within $O(\tau_{\mathit{fix}} \cdot k^k)$ time.*

To find an optimal solution to $P|fix_j, pmtn, \Delta s^l_{jj'}|\gamma$ problem, it suffices to set $n_{\max} = k^m$ in model (3)–(8). Indeed, the number of different sets of jobs that may be executed simultaneously does not exceed k^m. Besides that, there exists an optimal solution to problem $P|fix_j, pmtn, \Delta s^l_{jj'}|\gamma$ where each of the above mentioned sets of jobs is executed simultaneously at most once. This fact follows by the lot shifting technique which is applicable here since the setup times obey the triangle inequality (see e.g. [11]).

Let $\mathcal{P}'_{\mathit{fix}}$ denote the linear programming problem obtained by fixing all Boolean variables (w_{jn}) in MIP model (3)–(8) supplemented by a linear programming formulation of optimization criterion γ. A problem $\mathcal{P}'_{\mathit{fix}}$ with $n_{\max} = k^m$ and the number of processors bounded above by a constant is polynomially solvable. Let τ'_{fix} denote an upper bound of the time complexity of solving $\mathcal{P}'_{\mathit{fix}}$. The problem $P|fix_j, pmtn, \Delta s^l_{jj'}|\gamma$, where the numbers of processors and jobs are bounded by a constant is denoted by $Pm|fix_j, pmtn, \Delta s^l_{jj'}, k = \text{const}|\gamma$ in what follows. This problem reduces to $2^{kn_{\max}}$ problems of $\mathcal{P}'_{\mathit{fix}}$ type, where $n_{\max} = k^m$. So the following result holds.

Theorem 2 *Problem $Pm|fix_j, pmtn, \Delta s^l_{jj'}, k = \text{const}|\gamma$, $\gamma \in \{C_{\max}, C_{\sum}, L_{\max}, L_{\sum}\}$, is polynomially solvable within $O\left(\tau'_{\mathit{fix}} \cdot 2^{k^{m+1}}\right)$ time.*

In some works it is assumed that a job has a number of alternative modes, where each processing mode is specified by a subset of processors and the execution time of the job on that particular processor set. Such jobs are called *multimode multiprocessor* jobs [1]. Our MIP models and polynomially solvable cases for single mode multiprocessor jobs may be extended to the scheduling problem with multimode multiprocessor jobs and various regular criteria, which can be formulated in terms of linear programming.

3.2 Rigid Jobs

In order to find an optimal solution to $P|size_j, s_{jj'}|\gamma$ using model (9)–(16), it is sufficient to set $n_{\max} = k$ because the preemptions are not allowed. Denote by $\mathcal{P}_{size}$ the linear programming problem obtained by fixing all Boolean variables (w_{jn}) and (z_{jln}) in model (9)–(16) with a linear formulation of optimization criterion γ. Problem $\mathcal{P}_{size}$ with $n_{\max} = k$ involves $O(k^2)$ variables and $O(k^4 m)$ constraints, then it is pseudopolynomially solvable, since m is the numerical parameter of the problem. Let τ_{size} be an upper bound on the time complexity of solving problem $\mathcal{P}_{size}$.

The problem $P|size_j, s_{jj'}|\gamma$, where the number of jobs is bounded by a constant from above, is denoted by $P|size_j, s_{jj'}, k = \text{const}|\gamma$. This problem reduces to $O\left(n_{\max}^k \prod_{j=1}^{k} C_m^{size_j}\right)$ problems of $\mathcal{P}_{size}$ type with $n_{\max} = k$. Problem $P|size_j, s_{jj'}, k = \text{const}|\gamma$ is polynomially solvable in $O\left(\tau_{size} \cdot k^k \prod_{j=1}^{k} C_m^{size_j}\right)$ time, if $m \leq \sum_{j=1}^{k} size_j$ and sizes $size_j$ are bounded by a constant for all $j \in \mathcal{J}$, and the problem is trivial, if $m > \sum_{j=1}^{k} size_j$. In the latter case, all jobs start at time moment $t = 0$ in the early schedule. Therefore the following theorem holds.

Theorem 3 *Problem $P|size_j, s_{jj'}, k = \text{const}|\gamma$, $\gamma \in \{C_{\max}, C_{\sum}, L_{\max}, L_{\sum}\}$, is polynomially solvable, when parameters $size_j$ are bounded by a constant for all $j \in \mathcal{J}$*

To find an optimal solution to $P|size_j, pmtn, \Delta s_{jj'}|\gamma$ problem, it suffices to set $n_{\max} = k^m$ in model (10)–(16) as in the case of dedicated processors (see section 3.1). Let $\mathcal{P}'_{size}$ be the linear programming problem obtained by fixing all Boolean variables (w_{jn}) and (z_{jln}) in MIP model (10)–(16) supplemented by a linear formulation of optimization criterion γ. Denote by τ'_{size} an upper bound of the time complexity of solving $\mathcal{P}'_{size}$. The problem $P|size_j, pmtn, \Delta s_{jj'}|\gamma$, where the numbers of processors and jobs are bounded by a constant, we denote by $Pm|size_j, pmtn, \Delta s_{jj'}, k = \text{const}|\gamma$. This problem reduces to $2^{kmn_{\max}}$ problems of $\mathcal{P}'_{size}$ type, where $n_{\max} = k^m$. Thus, we have

Theorem 4 *Problem $Pm|size_j, pmtn, \Delta s_{jj'}, k = \text{const}|\gamma$, $\gamma \in \{C_{\max}, C_{\sum}, L_{\max}, L_{\sum}\}$, is polynomially solvable within $O\left(\tau'_{size} \cdot 2^{mk^{m+1}}\right)$ time.*

Let us assume that there is a set of usable processor numbers for each job $j \in \mathcal{J}$. Then the jobs are called *moldable* jobs [1], if the number of required processors is chosen before starting a job and is not changed until the job termination. Jobs are called *malleable* [1], if the number of processors can be changed at runtime. The MIP models and polynomially solvable cases presented above for the case of rigid jobs may be generalized to the scheduling problems with moldable and malleable jobs.

4 Conclusions

The problem of multiprocessor job scheduling is studied in parallel and dedicated versions. MIP models are formulated for both versions of the problem using the event-points approach and continuous time representation. New polynomially solvable special cases of the problem are found using the MIP models, under assumption that the number of jobs is bounded by a constant.

Presented models and polynomially solvable cases are extended to the other (more general) scheduling problems with moldable jobs, malleable jobs, multimode multiprocessor jobs and various regular criteria, which can be formulated in linear form.

Acknowledgements

This research is supported by the Russian Science Foundation grant 15-11-10009. The authors are grateful to M.Y. Kovalyov for helpful remarks.

References

[1] Drozdowski M.: Scheduling for Parallel Processing. Springer-Verlag, London (2009)

[2] Eremeev, A.V., Kovalenko J.V.: On multi-product lot-sizing and scheduling with multi-machine technologies. Operations Research Proceedings 2014. Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), (September 2-5, 2014), 301–306, Springer, Berlin (2016)

[3] Garey, M.R., Johnson, D.S.: Computers and intractability. A guide to the theory of NP-completeness. W.H. Freeman and Company, San Francisco (1979)

[4] Grötschel M., Lovász L., Schrijver A.: Geometric Algorithms and Combinatorial Optimizations, 2nd corrected edition. Springer-Verlag (1993)

[5] Itai A., Papadimitriou C. H., Szwarcfiter J. L.: Hamilton paths in grid graphs. SIAM J. Comput. **11** (4) 676–686 (1982)

[6] Ierapetritou, M.G., Floudas, C.A.: Effective continuous-time formulation for short-term scheduling: I. Multipurpose batch process. Ind. Eng. Chem. Res. **37** 4341–4359 (1998)

[7] Jansen, K., Porkolab, L.: Preemptive parallel task scheduling in $O(n)+poly(m)$ time. D.T. Lee and S.H. Teng, editors, Proc. of ISAAC 2000. LNCS, Vol. 1969, 398–409 (2000)

[8] Jansen, K., Porkolab, L.: Preemptive scheduling with dedicated processors: applications of fractional graph coloring. Journ. Scheduling **7** 35–48 (2004)

[9] Mukhacheva, E.A. and Mukhacheva, A.S.: L.V. Kantorovich and cutting-packing problems: New approaches to combinatorial problems of linear cutting and rectangular packing. Journal of Mathematical Sciences **133** (4) 1504–1512 (2006)

[10] Shaik, M.A., Floudas, C.A., Kallrath, J., Pitz, H.J.: Production scheduling of a large-scale industrial continuous plant: short-term and medium-term scheduling. Comp. Chem. Engng. **33**, 670–686 (2009)

[11] Tanaev, V.S., Kovalyov, M.Y., Shafransky, Y.M.: Scheduling Theory. Group Technologies. Minsk, Institute of Technical Cybernetics NAN of Belarus (1998) (In Russian)

Appendix: The Proof of Proposition 1

Proposition 1 *Problems* $1|pmtn, \Delta s_{jj'}|\gamma$ *and* $1|\Delta s_{jj'}|\gamma$ *with* $\gamma \in \{C_{\max}, L_{\max}, C_{\sum}, L_{\sum}\}$ *are strongly NP-hard.*

Proof. We will consider only $C_{\sum}$ criterion since the problems with other three criteria are treated analogously.

In [5] it is proven that recognition of grid graphs with a Hamiltonian path (the Hamilton Path Problem) is NP-complete. Recall that a graph $G' = (V', E')$ with vertex set V' and edge set E' is called a *grid* graph, if its vertices are the integer vectors $v = (x_v, y_v) \in \mathbf{Z}^2$ on plane, i.e., $V' \subset \mathbf{Z}^2$, and a pair of vertices is connected by an edge iff the Euclidean distance between them is equal to 1. Here and below, $\mathbf{Z}$ denotes the set of integer numbers. We can assume that graph G' is connected since otherwise Hamiltonian path does not exist and this can be recognized in polynomial time.

Let us first reduce Hamilton Path problem to $1|\Delta s_{jj'}|C_{\sum}$, assuming that jobs correspond to vertices and the setup times are equal to Euclidean distances between the integer points where the corresponding vertices are located. All processing times $p_j = 1$.

Then minimal setup times are equal to one. The earliest starting times of the jobs are $1, 3, 5, 7, \ldots, 2k-1$, where k is the number of jobs.

In the recognition version of $1|\Delta s_{jj'}|C_{\sum}$ it is required to answer the question: Is there a schedule with $C_{\sum}$ value not greater than a given value K?

Let us put $K := (1 + 3 + 5 + 7 + .. + 2k - 1) = k^2$.

On one hand, if a schedule with the value of $C_{\sum}$ at most K exists, then all setups of this schedule are equal to 1 and graph G' contains a Hamilton path. On the other hand, if graph G' contains a Hamilton path then ordering the jobs in the sequence of vertices of this path we obtain a schedule with the value of $C_{\sum} = k^2$.

This reduction is computable in polynomial time and all input parameters of $1|\Delta s_{jj'}|C_{\sum}$ instance are upper bonded by k, so we conclude that $1|\Delta s_{jj'}|C_{\sum}$ problem is strongly NP-hard.

In the case of $1|pmtn, \Delta s_{jj'}|C_{\sum}$ problem we construct the same reduction. Note that in a schedule with $C_{\sum} \leq k^2$ the preemptions are impossible. Indeed, suppose that job j' is the first job that has a preemption and all n preceding jobs have no preemptions. Let job j' be executed for a units of time and then preemption took place and let j'' be the first job, which finishes after this preemption.

In case $j'' \neq j'$, job j'' ends at time $t \geq 2(n+1) - 1 + a$. In this case even if all jobs after j'' finish in the earliest possible times $2(n+2) - 1,\ 2(n+3) - 1, \ldots, 2k - 1$, then still $C_{\sum} > k^2$.

In case $j'' = j'$, job j' ends at time $t \geq 2(n+1) - 1 + b$, where b is the total preemption time of job j'. Thus by the same reasoning as in the previous case, we conclude that $C_{\sum} > k^2$. ∎