Decision Making in Manufacturing and Services
Vol. 1 • 2007 • No. 1–2 • pp. 59–89

Resource Management in Machine Scheduling Problems: A Survey

Adam Janiak*, Władysław Janiak**, Maciej Lichtenstein*

Abstract. The paper is a survey devoted to job scheduling problems with resource allocation. We present the results available in the scientific literature for commonly used models of job processing times and job release dates, i.e., the models in which the job processing time or the job release date is given as a linear or convex function dependent on the amount of the additional resource allotted to the job. The scheduling models with resource dependent processing times or resource dependent release dates extend the classical scheduling models to reflect more precisely scheduling problems that appear in real life. Thus, in this paper we present the computational complexity results and solution algorithms that have been developed for this kind of problems.

Keywords: scheduling, resource allocation, resource dependent processing times, resource dependent release dates.

Mathematics Subject Classification: 90B35, 90B30, 90B50.

Received/Revised: 16 April 2007/05 July 2007

1. INTRODUCTION

In the modern, rapidly technologically developing manufacturing process, one of the most important tasks that managers are faced with is to determine the production schedules. This task (in general) is to distribute available resources (machines, manpower, energy, money) in such a way, that the total production costs will be minimized while the productivity, and the quality of products will be maximized.

The importance of the optimal production scheduling follows from the increasing maintenance costs of production infrastructure, which forces the managers to utilize it optimally. Thus, the field of scheduling theory, is the area of expertise that should be familiar to many managers that are faced to the problems of production planning. The scheduling theory already attracted particular attention of the researchers for several

* Institute of Computer Engineering, Wrocław University of Technology, Control and Robotics, Poland. E-mail: {adam.janiak,maciej.lichtenstein}@pwr.wroc.pl.
** A student of the Faculty of Computer Science and Management, Wrocław University of Technology, Poland. E-mail: wladyslaw.janiak@interia.pl.

decades, and there is a significant development of models and algorithms that can be viewed as a tools ready for optimal scheduling in various production environments.

The classical scheduling theory deals with the models in which it is assumed that all problem parameters, such as job processing times, job release dates, etc. are constant given values. These models, however, do not reflect precisely many real life production systems in which problems parameters are variables dependent on some additional resources. In early 80's these classical models were extended by Janiak [41] and independently by Vickson [77], [78] to models with resource dependent processing times. Similarly, in 1989 Janiak introduced the problems with resource dependent release dates [31].

Generally, in scheduling problems with additional resource allocation some problem parameters, e.g.: processing times, release dates, setup times, are given as functions dependent of additional resources.

Those resources may be continuously divisible (e.g. energy, financial outlay, etc.) or discretely divisible (e.g. manpower, tools, memory, etc.). Thus, in this kind of scheduling problems during the optimization process, beside determining of the order of job processing, the additional resource allocation vector must be found, which usually makes problems harder to solve.

The necessity of modeling many real-life production and computer systems, in which resource allocation occurs, forced researchers to develop new models and solution algorithms for scheduling problems with resource dependent parameters.

In this paper we survey the results available in the literature regarding scheduling problems with resource dependent processing times and scheduling problems with resource dependent release dates. This survey do not cover the model of the resource amount vs. processing rate where the processing rate (not the processing time) of a job is a function of the continuous resources [80], [51], [52] and models in which, the job requires the resource to be processed and its processing time is not dependent on its amount [2]. Also the project scheduling do not fit in the scope of this paper, since they are different from the machine scheduling problems.

The paper is organized as follows. In the next section we define precisely considered problems and introduce some basic definitions and notations. Section 3 is devoted to single processor scheduling problems with job processing times linearly dependent on continuous resources. Section 4 deals with scheduling problems with resource dependent processing times in multiprocessor environments such as parallel processors, flow shops, job shops, etc. Scheduling problems with resource dependent release dates are surveyed in Section 5. Sections 6 and 7 are devoted to scheduling problems with processing times and/or release dates described as a convex functions of the additional resources, and linear functions dependent on discrete resources, respectively. Some conclusions and directions for future research are given in Section 8.

2. PROBLEMS DEFINITION AND FORMULATION

Let $J = \{1, \ldots, j, \ldots, n\}$ denote the set of n independent jobs to be processed on a single processor, or on the set of m processors $M = \{1, \ldots, i, \ldots, m\}$. For each job

$j \in J$, the following parameters may be defined: p_{ij} – the processing time on processor i (index i is omitted for single processor problems), r_j – the release date, d_j – the due date, $\bar{d}_j$ – the deadline. It may be also assumed that there exist some precedence constraints between the jobs, which is denoted by *prec* in the problem notation.

The processing time as well as the release date are functions of allocated resources u_{ij} or u_j, which is denoted by $p_{ij}(u_{ij})$ (index i is omitted for single processor problems) and $r_j(u_j)$, respectively. Resources may be constrained locally, i.e. $\alpha_{ij} \leq u_{ij} \leq \beta_{ij}$ as well as globally, i.e. $\Sigma_{j\in J, i\in M} u_{ij} \leq R$, which is simply denoted by $\Sigma u_{ij} \leq R$, where α_{ij} and β_{ij} are given bounds. Moreover, the resources can be continuously or discretely divisible. In the case of continuously divisible resources the amount u_j of the resource assigned to the job j, (or its release date), can be any value within the range $[\alpha_j, \beta_j]$. In the case of discretely divisible resources, the amount of the resource u_j, can be one of the finite number of values within the same range. Usually, it is assumed in the literature that the amount of the resource can have one of the values from the set $\{\alpha_j, \beta_j\}$. In the case of discretely divisible resources, we use the separate upperscript d to indicate this fact, i.e., u_j is denoted by u_j^d.

The functions of allocated resources that appear in the literature are linear or convex. In the linear case, the job processing times or release dates are given by the following formulae:

$$f_j(u_j) = b_j - a_j u_j, \tag{1}$$

where f_j is the job processing time (p_j or p_{ij}) or the release date (r_j); b_j is the maximum value of the job processing time or release date and a_j is the resource consumption ratio.

In the convex case, the job processing times or release dates are given by the following formulae:

$$f_j(u_j) = (a_j/u_j)^k, \tag{2}$$

where f_j is the job processing time (p_j or p_{ij}) or the release date (r_j); a_j and k are the resource consumption parameters.

The objective is to find the sequence of jobs (if is not given) and the resource allocation (the amounts of resource assigned to each job or to each release date) that minimizes the given objective function value.

In considered problems the schedule cost can be measured by two objective functions: (classical) time criterion (TC) and resource consumption penalty (RC).

The time criterion is one of the commonly used schedule cost measures such as:

— $C_{max} = \max_{j\in J}\{C_j\}$ – the maximum completion time, makespan,
— $L_{max} = \max_{j\in J}\{L_j\} = \max_{j\in J}\{d_j - C_j\}$ – the maximum lateness,
— $T_{max} = \max_{j\in J}\{T_j\} = \max_{j\in J}\{0, d_j - C_j\}$ – the maximum tardiness,
— $E_{max} = \max_{j\in J}\{E_j\} = \max_{j\in J}\{0, C_j - d_j\}$ – the maximum earlines,
— $c_{max} = \max_{j\in J}\{c_j(C_j)\}$, where $c_j(t)$ is nondecreasing penalty function of completing job j at time t – maximum penalty cost,
— $\Sigma w_j C_j = \sum_{j\in J} w_j C_j$ – the total weighted completion time,
— $\Sigma w_j T_j = \sum_{j\in J} w_j T_j$ – the total weighted tardiness, etc,

where C_j denotes the completion time of job j.

The resource consumption penalty is the total resource consumption Σu_j, Σu_{jk} or total weighted resource consumption $\Sigma v_j u_j$ or $\Sigma v_{jk} u_{jk}$, where where v_j and v_{jk} are constant given weights.

The following four optimization problems are considered: (1) minimization of TC under a given constraint on RC; (2) minimization of RC under a given constraint on TC; (3) minimization of $TC + RC$ and (4) double-criteria approach $TC \wedge RC$ in which the set of Pareto optimal solutions has to be found. It is easy to notice, that NP-hardness of the problem with models (1)–(3) lead to NP-hardness of the same problem with model (4). Similarly, polynomial solvability of (4) leads to polynomial solvability of (1)–(3).

In the following we survey the results present in the literature for the problems with resource dependent job processing times and resource dependent release dates. The notation of scheduling problems used in this paper is compliant with the three-field notation introduced in [17] and extended in [39].

3. SINGLE PROCESSOR PROBLEMS WITH LINEAR MODELS OF JOB PROCESSING TIMES

In this section we consider the single processor problems with job processing times given as a linear functions dependent on some additional resources, i.e. job processing times are given by formulae (1). The results are summarized in Table 1.

Table 1. Computational complexity of single processor problems with linear models of processing times

Problem	Complexity	Reference
$1\|prec, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|C_{max}$	$O(n \log n)$	[20]
$1\|prec, p_j = b_j - a_j u_j, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[32]
$1\|prec, p_j = b_j - a_j u_j\|C_{max} \wedge \Sigma u_j$	$O(n \log n)$	[32]
$1\|prec, r_j, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|C_{max}$	$O(n^2)$	[20]
$1\|prec, r_j, p_j = b_j - a_j u_j, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n^2)$	[32]
$1\|prec, r_j, p_j = b_j - a_j u_j\|C_{max} \wedge \Sigma u_j$	$O(n^2)$	[32]
$1\|prec, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|L_{max}$	$O(n^2)$	[20]
$1\|prec, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}, C_j \leq d_j\|C_{max}$	$O(n^2)$	[32]
$1\|prec, p_j = b_j - a_j u_j, L_{max} \leq \hat{L}\|\Sigma u_j$	$O(n^2)$	[32]
$1\|prec, p_j = b_j - a_j u_j\|L_{max} \wedge \Sigma u_j$	$O(n^2)$	[32]
$1\|prec, r_j, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|L_{max}$	strongly NP-hard	[20]
$1\|prec, r_j, p_j = b_j - a_j u_j, L_{max} \leq \hat{L}\|\Sigma u_j$	strongly NP-hard	[32]
$1\|prec, r_j, p_j = b_j - a_j u_j\|L_{max} \wedge \Sigma u_j$	strongly NP-hard	[32]
$1\|r_j, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|L_{max}$	strongly NP-hard	[32]
$1\|r_j, p_j = b_j - a_j u_j, L_{max} \leq \hat{L}\|\Sigma u_j$	strongly NP-hard	[32]

Table 1 (continued)

Problem	Complexity	Reference
$1\|r_j, p_j = b_j - a_j u_j\|L_{max} \wedge \Sigma u_j$	strongly NP-hard	[32]
$1\|prec, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|T_{max}$	$O(n^2)$	[20]
$1\|prec, p_j = b_j - a_j u_j, T_{max} \leq \hat{T}\|\Sigma u_j$	$O(n^2)$	[32]
$1\|prec, p_j = b_j - a_j u_j\|T_{max} \wedge \Sigma u_j$	$O(n^2)$	[32]
$1\|prec, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|c_{max}$	$O(n^2)$	[22]
$1\|p_j = b_j - a_j u_j, c_{max} \leq \hat{c}\|\Sigma u_j$	$O(n^2)$	[75], [22], [39]
$1\|p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|c_{max}$	class P	[39]
$1\|p_j = b_j - a_j u_j\|c_{max} \wedge \Sigma u_j$	open	[39]
$1\|p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|\Sigma w_j C_j$	open	[39]
$1\|prec\cdot, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|\Sigma w_j C_j$	$O(n \log n)$	[26]
$1\|r_j, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|\Sigma C_j$	strongly NP-hard	[29]
$1\|prec, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|\Sigma C_j$	NP-hard	[29]
$1\|p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|\Sigma T_j$	NP-hard	[29]
$1\|r_j, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|\Sigma T_j$	strongly NP-hard	[29]
$1\|p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|\Sigma w_j T_j$	strongly NP-hard	[29]
$1\|p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|\Sigma c_j$	strongly NP-hard	[29]
$1\|p_j = b_j - a_j u_j, C_j \leq d_j\|\Sigma v_j u_j$	$O(n \log n)$	[42]
$1\|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j\|T_{max} + \Sigma v_j u_j$	$O(n^2)$	[78]
$1\|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j\|L_{max} + \Sigma v_j u_j$	$O(n^2)$	[67]
$1\|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, L_{max} \leq \hat{L}\|\Sigma v_j u_j$	$O(n^2)$	[67]
$1\|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, \Sigma v_j u_j \leq \hat{R}\|L_{max}$	$O(n^2)$	[67]
$1\|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j\|L_{max} \wedge \Sigma v_j u_j$	$O(n^2)$	[67]
$1\|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j\|C_{max} + \Sigma v_j u_j$	$O(n^2)$	[67]
$1\|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, C_{max} \leq \hat{C}\|\Sigma v_j u_j$	$O(n^2)$	[67]
$1\|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, \Sigma v_j u_j \leq \hat{R}\|C_{max}$	$O(n^2)$	[67]
$1\|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j\|C_{max} \wedge \Sigma v_j u_j$	$O(n^2)$	[67]
$1\|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, q_j\|C_{max} + \Sigma v_j u_j$	strongly NP-hard	[58]
$1\|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j\|L_{max} + \Sigma v_j u_j$	strongly NP-hard	[58]
$1\|p_j = b_j - a_j u_j, 0 \leq u_j \leq \beta_j, d_j = d\|\|\Sigma U_j \wedge \Sigma u_j$	NP-hard	[6]
$1\|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, d_j = d\|\Sigma U_j \wedge \Sigma v_j u_j$	NP-hard	[7]
$1\|s, p_j = b_j - u_j, \alpha_j \leq u_j \leq \beta_j, \tilde{d}\|\Sigma v_j u_j$	NP-hard	[11]
$1\|r_j, s_j = b'_j - a'_j u'_j, p_j = b_j - a_j u_j, 0 \leq u'_j \leq \beta'_j, 0 \leq u_j \leq \beta_j, \Sigma v_j \leq \hat{R}_1, \Sigma u_j \leq \hat{R}_2\|F$ where $F \in \{C_{max}, L_{max}, T_{max}, \Sigma w_j C_j, \Sigma \alpha_j E_j + \Sigma \gamma_j T_j\}$ and the job sequence is fixed	$O(n^2)$	[45]

Problems:

a) $1|prec, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|C_{max}$,
b) $1|prec, p_j = b_j - a_j u_j, C_{max} \leq \hat{C}|\Sigma u_j$,
c) $1|prec, p_j = b_j - a_j u_j|C_{max} \wedge \Sigma u_j$.

In problems a) – c), the criterion function value does not depend on the sequence of jobs. The optimal solution may be obtained in $O(n \log n)$ steps by allocation of resources to any permutation of jobs feasible with respect to precedence constraints. In problem a), the maximal possible amount of resources is allocated to jobs starting from the job with maximal value of a_j till resource depletion [22]. Similarly, for problem b), resources are allocated to jobs starting from the job with maximal value of a_j and allocation is finished when $C_{max} \leq \hat{C}$ holds [32]. For the problem c), the set of Pareto-optimal solutions may be calculated using an algorithm described in [32]. The algorithm allocates resources in a special manner to jobs with maximal value of a_j. The set of Parteo-optimal solutions forms a piece-wise linear decreasing convex curve.

Problems:

a) $1|prec, r_j, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|C_{max}$,
b) $1|prec, r_j, p_j = b_j - a_j u_j, C_{max} \leq \hat{C}|\Sigma u_j$,
c) $1|prec, r_j, p_j = b_j - a_j u_j|C_{max} \wedge \Sigma u_j$.

The optimal solutions for above mentioned problems may be obtained in $O(n^2)$ steps by an algorithm analogous to the algorithm for classical $1|prec, r_j|C_{max}$ problem [19]. Jobs with not scheduled predecessors are scheduled according to increasing values of r_j from the first position in constructed schedule. Resources are allocated to shorten the critical path using as little as possible of resource. An exact algorithms of resource allocation are described in [20] for problem a) and in [39] for b) and c).

Problem $1|prec, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|L_{max}$ – The optimal solution may be obtained in $O(n^2)$ steps by an algorithm analogous to the algorithm for classical $1|prec|L_{max}$ problem [19]. Jobs with not scheduled successors are scheduled according to decreasing d_j from the last position in constructed schedule. Resources are allocated to shorten the critical path using at least as possible of the resource [20].

Problems $1|prec, p_j = b_j - a_j u_j, L_{max} \leq \hat{L}|\Sigma u_j$ and $1|prec, p_j = b_j - a_j u_j|L_{max} \wedge \Sigma u_j$ are symmetrical to $1|prec, r_j, p_j = b_j - a_j u_j, C_{max} \leq \hat{C}|\Sigma u_j$ and $1|prec, r_j, p_j = b_j - a_j u_j|C_{max} \wedge \Sigma u_j$, respectively, thus very similar algorithms may be constructed to solve them [39].

Problems:

a) $1|prec, r_j, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|L_{max}$,
b) $1|prec, r_j, p_j = b_j - a_j u_j, L_{max} \leq \hat{L}|\Sigma u_j$,
c) $1|prec, r_j, p_j = b_j - a_j u_j|L_{max} \wedge \Sigma u_j$.

It was shown in [32] that all of the above problems are strongly NP-hard. For a given sequence of jobs, problems a) and b) reduce to a linear programming task, and problem c) to a parametric linear programming task [39], and thus may be solved by polynomial algorithms [55], [53] or more effective well known simplex algorithm which is non polynomial. Practically, the above algorithms are not computationally effective. It was proven that any algorithm based on greedy approach cannot be constructed for problems with a given job sequence [39]. Modified algorithm for time/cost trade-off curve may be used. An adaptation was shown in [32] and pseudopolynomial algorithm was constructed. Heuristic algorithms for problems a) b) and c) are presented in [32]. Numerical experiments show that average values of upper bound of relative error are about 15% and running times are less than a few seconds. In [23], a branch and bound algorithm was constructed for problem a) and may be used for problems with up to 100 jobs.

Problem $1|r_j, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|L_{max}$ is strongly NP-hard [32]. The algorithms based on artificial neural networks (ANN), genetic search (GS) and tabu-search (TS) are presented in [39]. The best solutions are delivered by the ANN algorithm.

Problem $1|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, q_j|C_{max}$ was considered in [82] and an approximation algorithm, a lower bound of criterion function value and the worst-case analysis for the proposed algorithm are presented.

Problems:

a) $1|prec, r_j, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|T_{max}$,
b) $1|prec, r_j, p_j = b_j - a_j u_j, T_{max} \leq \hat{T}|\Sigma u_j$,
c) $1|prec, r_j, p_j = b_j - a_j u_j|T_{max} \wedge \Sigma u_j$.

The optimal solution for any problem with resource allocation and criterion L_{max} is optimal for the same problem with criterion T_{max}. Hence, problem a) may be solved by the algorithm for mentioned above problem $1|prec, r_j, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|L_{max}$. A special algorithm was constructed for problem a) in [25]. Similarly, problems b) and c) may be solved by algorithms for analogical problems with criterion L_{max} [39].

Problems

a) $1|prec, p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|c_{max}$,
b) $1|prec, p_j = b_j - a_j u_j, c_{max} \leq \hat{c}|\Sigma u_j$,
c) $1|prec, p_j = b_j - a_j u_j|c_{max} \wedge \Sigma u_j$.

First of all it is easy to notice that, in problem a) with equal resource consumption ratios ($a_j = a$), i.e. for problem $1|prec, p_j = b_j - a u_j, \Sigma u_j \leq \hat{R}|c_{max}$, if a job sequence is given, the optimal resource allocation may be obtained by the maximal possible allocation of resources to consecutive jobs, starting from the first, till the resource depletion (the complexity $O(n)$).

In [22], for the general case of a) (with different values of a_j), an optimal algorithm with the complexity $O(n^2)$ was presented. The algorithm finds the optimal sequence of jobs and the optimal resource allocation. In [75], an algorithm was presented for some problem similar to $1|p_j = b_j - a_j u_j, c_{max} \leq \hat{c}|\Sigma u_j$. Based on this result, an algorithm with the complexity $O(n^2)$ may be constructed for problem $1|p_j =$

$b_j - a_j u_j, c_{max} \leq \hat{c}|\Sigma u_j$ [39]. This algorithm cannot be adopted to solve problem $1|p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|c_{max}$, however, this problem can be solved in polynomial time [39].

Problem $1|p_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|\Sigma w_j C_j$ – For a given job sequence, the optimal resource allocation may be found in $O(n \log n)$ steps, [39]. If job sequence is not given, then the computational complexity of the problem is an open question [39]. In [26], the problem with a specific dominance relation (denoted by "$< \cdot$") was considered. The dominance relation "$< \cdot$" is defined as follows: job j dominates i if $b_i \leq b_j$ and $a_i \geq a_j$ and $\beta_i \geq \beta_j$ and $w_i \geq w_j$. In such case, in optimal solution, job j is processed before job i. If relation "$< \cdot$" holds for every pair of jobs then it is possible to find the optimal solution in $O(n \log n)$ steps by sorting jobs according to "$< \cdot$" and assigning the resource in maximum possible amount from the first job till its depletion.

Problem $1|p_j = b_j - a_j u_j, C_j \leq d_j|\Sigma v_j u_j$ – In [42] and [39], it was shown that the solution for the problem may be found in $O(n \log n)$ steps. Jobs are scheduled according to increasing critical paths and resource allocation is done by a generalization of Moore's algorithm [63].

Problem $1|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|\Sigma w_j C_j + \Sigma v_j u_j$ is considered in [77]. Two important properties for the problem are presented: 1) each job is shortened maximally ($u_j = \beta_j$) or is not shorten at all ($u_j = 0$); 2) the SWPT rule (*Shortest Weighted Processing Time First* [73]) holds for the optimal sequence of jobs. Generally, there are 2^n cases of possible choices of u_j that have to be checked to find the optimal solution. Some tests are proposed to reduce the space where the optimal solution may be found. A special function for bounds calculation for branch and bound method and efficient heuristic algorithm are presented. Numerical experiments show that the heuristic algorithm is efficient for problems with size from 10 to 100 jobs and delivered solutions are optimal or near to optimal. The problem with $w_j = w$ may be formulated as an assignment problem [78]. The general version (with arbitrary values of w_j) has been proved to be NP-hard [43].

Problem $1|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|T_{max} + \Sigma v_j u_j$ is presented in [78]. The optimal sequence of jobs may be found by scheduling the jobs according to nondecreasing due dates. The optimal algorithm with complexity $O(n \log n)$ is presented.

Problem $1|p_j = b_j - a_j u_j|T_{\max} \wedge \Sigma v_j u_j$ was considered in [79]. The greedy algorithm with the complexity $O(n^2)$ was presented.

Problems:

a) $1|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|L_{max} + \Sigma v_j u_j$,
b) $1|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, L_{max} \leq \hat{L}|\Sigma v_j u_j$,
c) $1|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, \Sigma v_j u_j \leq \hat{R}|L_{max}$,
d) $1|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|L_{max} \wedge \Sigma v_j u_j$,
e) $1|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|C_{max} + \Sigma v_j u_j$,
f) $1|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, C_{max} \leq \hat{C}|\Sigma v_j u_j$,
g) $1|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, \Sigma v_j u_j \leq \hat{R}|C_{max}$,
h) $1|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|C_{max} \wedge \Sigma v_j u_j$

were considered in [67]. An algorithm with complexity $O(n^2)$ for problem $1|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|C_{max} \wedge \Sigma v_j u_j$ is presented and equivalence of this problem to problem $1|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|L_{max} \wedge \Sigma v_j u_j$ is shown. These two problems are the most general ones among those presented above, so their solution algorithms may be applied to solve the remaining ones.

Problem $1|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|L_{max} + \Sigma v_j u_j$ is strongly NP-hard which follows from its classical counterpart (with fixed job processing times) [58]. Similarly, the problem $1|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, q_j|C_{max} + \Sigma v_j u_j$ is strongly NP-hard. An approximation algorithm with the worst case performance ratio equal to 2 was presented in [67].

In [81] and [65], approximation algorithms with worst case performance ratio equal to $\rho + 1/2$ and $\rho + 1/3$ are presented, where ρ denotes worst case performance ratio for a the problem with fixed given job processing times.

Problem $1|r_j, p_j = b_j - u_j, 0 \leq u_j \leq \beta_j, d_j = d|\Sigma(vE_j + wT_j + yu_j)) \wedge \min d$, where $\min d$ indicates that the due date value has to be optimally selected during the optimization process, was considered in [69]. Some properties of the problem are presented and its formulation as an assignment problem.

Problem $1|p_j = b_j - a_j u_j, 0 \leq u_j \leq \beta_j|\Sigma U_j \wedge \Sigma u_j$ is considered in [6]. It is proved that the problem is NP-hard for the case where $d_j = d$. Some properties and a dynamic programming algorithm is presented for the general case and for the case with common due date, $d_j = d$.

Problem $1|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|\Sigma w_j U_j \wedge \Sigma v_j u_j$ is considered in [7]. It is proved that the case with common due date, $d_j = d$, and common weight, $w_j = w$, is NP-hard. A dynamic programming algorithm, a fully polynomial approximation scheme and some approximation algorithms are proposed.

Problem $1|p_j = b_j - u_j, 0 \leq u_j \leq \beta_j|\Sigma(vE_j + wT_j + xd_j + y_j u_j)$ is considered in [12]. The problem was formulated as a polynomially solvable assignment problem.

The paper [10] is devoted to some bicriterion scheduling problems in which the processing time of each job is a linear decreasing function of the amount of a common discrete resource allocated to the a job. The quality of a solution is measured by two independent criteria F_1 and F_2. The first criterion is the maximal or total (weighted) resource consumption and the second criterion is a regular scheduling criterion depending on the job completion times. Both criteria have to be minimized. Algorithms for solving and general schemes for the construction of the Pareto set and the Pareto ϵ-approximation are presented.

Problem with setup times $1|s, p_j = b_j - a_j u_j, \alpha_j \leq u_j \leq \beta_j, \tilde{d}|\Sigma v_j u_j$ is studied in [11]. The authors prove that problem is NP-hard even if $a_j = 1$, i.e. for $1|s, p_j = b_j - u_j, \alpha_j \leq u_j \leq \beta_j, \tilde{d}|\Sigma v_j u_j$. A dynamic programming algorithm and a fully polynomial approximation scheme are presented.

Some properties of approximation schemes are presented in [56]. The properties improve quality and efficiency of calculation of upper and lower bounds of optimal criterion function value. A special procedure based on approximation scheme is presented. That procedure may be used to many single and multi processor scheduling problems.

4. MULTI PROCESSOR SCHEDULING PROBLEMS WITH LINEAR MODELS OF JOB PROCESSING TIMES

4.1. PARALLEL PROCESSORS

At first, we present some general result derived for parallel machine problems. The authors of [45] showed that for every scheduling criterion $F \in \{C_{max}, L_{max}, T_{max}, \Sigma w_j C_j, \Sigma\alpha_j E_j + \Sigma\gamma_j T_j\}$, the resource allocation problem with respect to a given job schedule in parallel processor problem $P|p_j = b_j - a_j u_j|F$ can be formulated as the linear programming task and thus solved in polynomial time. Notice, that the last of the criterion mentioned above, $\Sigma\alpha_j E_j + \Sigma\gamma_j T_j$, is not regular and not linear with respect to job completion times, so the result is not trivial. Moreover they propose some algorithm with $O(n^2)$ complexity for optimal resource distribution in $P|p_j = b_j - a_j u_j|C_{max}$.

The problem $Pm|pmtn, p_j = b_j - u_j|C_{\max} \wedge \Sigma v_j u_j$ was considered in [68]. The authors propose optimal algorithm that requires $O(n^2)$ steps, which proves that also $Pm|pmtn, p_j = b_j - u_j, \Sigma v_j u_j \leq \hat{U}|C_{\max}$, $Pm|pmtn, p_j = b_j - u_j, C_{\max} \leq \hat{C}|\Sigma v_j u_j$, and $Pm|pmtn, p_j = b_j - u_j|C_{\max} + \Sigma v_j u_j$ are polynomially solvable in $O(n^2)$ time. However, an algorithm with the computational complexity $O(n)$ was proposed in [49] for the problem $Pm|pmtn, p_j = b_j - a_j u_j, C_{\max} \leq \widehat{C}|\Sigma v_j u_j$. In [49], the problems without preemption were also considered. The polynomial time approximation schemes (PTAS) were proposed for the following problems: $Pm|p_j = b_j - a_j u_j, C_{\max} \leq \widehat{C}|\Sigma v_j u_j$, $Pm||p_j = b_j - a_j u_j, \Sigma v_j u_j \leq \widehat{U}|C_{\max}$, $Pm|p_j = b_j - a_j u_j|C_{\max} + \Sigma v_j u_j$.

The problems $Pm|p_j = b_j - a_j u_j|\Sigma w_j C_j + \Sigma v_j u_j$ and $Pm|p_j = b_j - a_j u_j|\Sigma w_j U_j + \Sigma v_j u_j$ were considered in [3]. Since both problems are NP-hard, a column generation branch and bound algorithm was proposed.

In [5], it was shown that problems $Pm|p_{ij} = a_{ij} - u_{ij}, 0 \leq u_{ij} \leq \beta_{ij}|\Sigma f_{ij} u_{ij} + \Sigma C_j$ and $Pm|p_{ij} = a_{ij} - u_{ij}, 0 \leq u_{ij} \leq \beta_{ij}, d \geq \Sigma a_{ij}|\Sigma f_{ij}(u_{ij}) + \Sigma(vE_j + wT_j)$ where $f(\cdot)$ is a convex, nondecreasing function, are equivalent to the assignment problem and may be solved by an algorithm with complexity $O(n^3 m + \log(nm))$. However, if m is a part of the input data all above mentioned problems become NP-hard.

The problems with the release dates $Pm|r_j, p_j = b_j - a_j u_j, \Sigma v_j u_j \leq \widehat{U}|\Sigma C_j$ and $Pm|pmtn, r_j, p_j = b_j - a_j u_j, \Sigma v_j u_j \leq \widehat{U}|\Sigma C_j$ were considered in [62]. Authors showed that the first problem is NP-hard and propose a $3 - (2/m)$-approximation algorithm. For the second problem, a linear programming formulation is presented which solves the problem optimally in polynomial time.

The problems $Qm|pmtn, p_j = b_j - u_j, C_{\max} \leq \widehat{C}|\Sigma v_j u_j$ and $Qm|pmtn, p_j = b_j - u_j|C_{\max} \wedge \Sigma v_j u_j$ were analyzed in [68]. An optimal algorithm that runs in $O(n \max(m, \log n))$ time and an ϵ-approximation scheme ware proposed for the first and the second problem, respectively. However, the computational complexity of the second problem is unknown.

The problem $Rm|p_j = b_j - u_j|\Sigma C_j + \Sigma v_j u_j$ was considered in [1]. The authors showed that the problem may be formulated as an assignment problem and is solvable in $O(n^{3m} + n^2 m \log(nm))$ time. For problem $Rm|p_j = b_j - u_j|\Sigma w_j C_j + \Sigma v_j u_j$,

a 3/2-approximation algorithm was proposed in [83]. In [74], [72] approximation algorithms were proposed for the problem $Rm|p_j = b_j - a_j u_j|C_{\max} + \Sigma v_j u_j$.

4.2. FLOW SHOP

We start our considerations from problems with C_{max} criterion function.

Recall that the classical two stage flow shop problem with C_{max} criterion without additional resource is solvable by Johnson's algorithm in $O(nlogn)$ time, [50]. The case with three stages is strongly NP-hard [15]. So, the problems with resource allocation are at least as difficult as their classical counterparts, which is shown in Table 2.

Table 2 Complexity results for multi processor problems

Problem	Complexity	Reference
$P\|r_j, s_j = b'_j - a'_j v_j, p_j = b_j - a_j u_j, 0 \le v_j \le \beta'_j, 0 \le u_j \le \beta_j, \Sigma v_j \le \hat{R}_1, \Sigma u_j \le \hat{R}_2\|F$ where $F \in \{C_{max}, L_{max}, T_{max}, \Sigma w_j C_j, \Sigma \alpha_j E_j + \Sigma \gamma_j T_j\}$ and the job sequence is fixed	solvable as Linear Programming task	[45]
$Pm\|pmtn, p_j = b_j - u_j\|C_{\max} \wedge \Sigma v_j u_j$	$O(n^2)$	[68]
$Pm\|pmtn, p_j = b_j - a_j u_j, C_{\max} \le \widehat{C}\|\Sigma v_j u_j$	$O(n)$	[49]
$Pm\|p_j = b_j - a_j u_j, C_{\max} \le \widehat{C}\|\Sigma v_j u_j$	PTAS	[49]
$Pm\|\|p_j = b_j - a_j u_j, \Sigma v_j u_j \le \widehat{U}\|C_{\max}$	PTAS	[49]
$Pm\|p_j = b_j - a_j u_j\|C_{\max} + \Sigma v_j u_j$	PTAS	[49]
$Pm\|r_j, p_j = b_j - a_j u_j, \Sigma v_j u_j \le \widehat{U}\|\Sigma C_j$	NP-hard	[62]
$Pm\|pmtn, r_j, p_j = b_j - a_j u_j, \Sigma v_j u_j \le \widehat{U}\|\Sigma C_j$	Lin. Prog.	[62]
$Pm\|p_j = b_j - a_j u_j\|\Sigma w_j C_j + \Sigma v_j u_j$	NP-hard	[3]
$Pm\|p_j = b_j - a_j u_j\|\Sigma w_j U_j + \Sigma v_j u_j$	NP-hard	[3]
$Pm\|p_{ij} = a_{ij} - u_{ij}, 0 \le u_{ij} \le \beta_{ij}\|\Sigma f_{ij} u_{ij} + \Sigma C_j$ and $f()$-is convex	$O(n^3 m + n^2 m \log(nm))$	[6]
$Pm\|p_{ij} = a_{ij} - u_{ij}, 0 \le u_{ij} \le \beta_{ij}, d \ge \Sigma a_{ij}\|\Sigma f_{ij}(u_{ij}) + \Sigma(vE_j + wT_j)$ and $f()$ is convex	$O(n^3 m + n^2 m \log(nm))$	[6]
$Qm\|pmtn, p_j = b_j - u_j, C_{\max} \le \widehat{C}\|\Sigma v_j u_j$	$O(n \max(m, \log n))$	[68]
$Qm\|pmtn, p_j = b_j - u_j\|C_{\max} \wedge \Sigma v_j u_j$	Open	[68]
$Rm\|p_j = b_j - u_j\|\Sigma C_j + \Sigma v_j u_j$	$O(n^{3m} + n^2 m \log(nm))$	[1]
$Rm\|p_j = b_j - a_j u_j\|C_{\max} + \Sigma v_j u_j$		[74], [72]
$F2\|p_{j1} = b_{j1} - a_1 u_{j1}, p_{j2} = b_{j2}, \Sigma u_j \le \hat{R}\|C_{max}$	NP-hard	[32], [37], [39]
$F2\|p_{j1} = b_{j1} - a_1 u_{j1}, p_{j2} = b_2, \Sigma u_j \le \hat{R}\|C_{max}$	$O(n^2)$	[32], [37], [39]
$F2\|p_{j1} = b_{j1} - a_1 u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_{j2}, \Sigma u_j \le \hat{R}\|C_{max}$	$O(n \log n)$	[32], [37], [39]
$F2\|p_{j1} = b_{j1} - a_1 u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_2, \Sigma u_j \le \hat{R}\|C_{max}$	$O(n \log n)$	[32], [37], [39]
$F2\|p_{j1} = b_{j1} - a_1 u_{j1}, p_{j2} = b_{j2}, C_{max} \le \hat{C}\|\Sigma u_j$	NP-hard	[30], [39]
$F2\|p_{j1} = b_{j1} - a_1 u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_{j2}, C_{max} \le \hat{C}\|\Sigma u_j$	$O(n \log n)$	[30], [39]

Table 2 (continued)

Problem	Complexity	Reference
$F2\|p_{j1} = b_1 - a_1u_{j1}, p_{j2} = b_2, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[30], [39]
$F2\|p_{j1} = b_1 - a_1u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_{j2}, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[30], [39]
$F2\|p_{j1} = b_1 - a_1u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_2, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[30], [39]
$F2\|p_{j1} = b_1 - a_1u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_2 - a_1u_{j2}, \beta_{j2} = \beta_2, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n)$	[30], [39]
$F2\|p_{j1} = b_{j1} - a_1u_{j1}, p_{j2} = b_{j2}\|C_{max} \wedge \Sigma u_j$	at least as hard as NP-hard $F2\|p_{j1} = b_{j1} - a_1u_{j1}, p_{j2} = b_{j2}, C_{max} \leq \hat{C}\|\Sigma u_j$	[30], [39]
$F2\|p_{j1} = b_1 - a_1u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_2\|C_{max} \wedge \Sigma u_j$	$O(n \log n)$	[30], [39]
$F2\|p_{j1} = b_{j1} - a_1u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_2\|C_{max} \wedge \Sigma u_j$	$O(n \log n)$	[30], [39]
$F2\|p_{j1} = b_1 - a_1u_{j1}, p_{j2} = b_2\|C_{max} \wedge \Sigma u_j$	$O(n \log n)$	[30], [39]
$F2\|p_{j1} = b_1 - a_1u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_{j2}\|C_{max} \wedge \Sigma u_j$	$O(n \log n)$	[30], [39]
$F2\|p_{j1} = b_1 - a_1u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_2 - a_1u_{j2}, \beta_{j2} = \beta_2\|C_{max} \wedge \Sigma u_j$	$O(n)$	[30], [39]
$F2\|p_{1j} = a_j - x_j, p_{2j} = a_j - y_j, 0 \leq x_j \leq u_j, 0 \leq y_j \leq v_j\|C_{max} + \Sigma(c_jx_j + d_jy_j)$	NP-hard	[66]
$Fm\|p_{ij} = \overline{p}_j - u_j\|C_{\max} \wedge \Sigma v_ju_j$	$O(n \log n)$	[13]
$FP\|prec, p_j = f_j^w(u_j), \Sigma u_j \leq \hat{R}\|C_{max}$	for $m \geq 3$ at least as hard as strongly NP-hard $FP3\|\|C_{max}$	[30], [39]
$FP\|prec, p_j = f_j^w(u_j), C_{max} \leq \hat{C}\|\Sigma u_j$	as above	[30], [39]
$FP\|prec, p_j = f_j^w(u_j)\|C_{max} \wedge \Sigma u_j$	as above	[30], [39]
$FP\|p_{jv} = b - au_{jv}, \beta_{jv} = \beta, j = 1, \ldots, n; v = 1, \ldots, m\|C_{max} \wedge \Sigma u_j$	$O(nm)$	[39]
$FP\|p_{ij} = a_{ij} - x_{ij}, 0 \leq x_{ij} \leq u_{ij}\|C_{max} + \Sigma c_{ij}x_{ij}$	NP-hard	[64]
$F\|prec, p_j = f_j^w(u_j), \Sigma u_j \leq \hat{R}\|C_{max}$	for m=2 at least as hard as NP-hard $F2\|p_{j1} = b_{j1} - a_1u_{j1}, p_{j2} = b_{j2}, \Sigma u_j \leq \hat{R}\|C_{max}$; for $m \geq 3$ at least as hard as strongly NP-hard $FP3\|\|C_{max}$	[27], [39]
$F\|prec, p_j = f_j^w(u_j), C_{max} \leq \hat{C}\|\Sigma u_j$	as above	[28], [39]
$F\|prec, p_j = f_j^w(u_j)\|C_{max} \wedge \Sigma u_j$	as above	[32], [39]

Table 2 (continued)

Problem	Complexity	Reference
$J\|prec, p_j = f_j^w(u_j), \Sigma u_j \leq \hat{R}\|C_{max}$	for $m \geq 2$ at least as hard as strongly NP-hard $J2\|p_j \in 1,2\|C_{max}$	[21], [39]
$J\|prec, p_j = f_j^w(u_j), C_{max} \leq \hat{C}\|\Sigma u_j$	for $m \geq 2$ at least as hard as NP-hard $F2\|p_{j1} = b_{j1} - a_1 u_{j1}, p_{j2} = b_{j2}, C_{max} \leq \hat{C}\|\Sigma u_j$	[47], [39]
$J\|prec, p_j = f_j^w(u_j)\|C_{max} \wedge \Sigma u_j$	as above	[47], [39]

Some properties for non-permutational flow shop problem are presented in [27]. The first result is that for problem $F|prec, p_j = f_j(u_j), \Sigma u_j \leq \hat{R}|C_{max}$ where $f_j(\cdot)$ is a non-increasing, convex function there exists some optimal solution where the sequence of jobs is the same on the first two processors and the sequence of jobs is the same on the last two processors. The second result states that if all of the jobs are available at the same time and the criterion function is regular and the processing times of jobs are non-increasing convex functions of the resource, then there exists an optimal solution such that the sequence of jobs is the same on the first and the second processor. The examples of regular criterion functions are C_{max} and Σu_j. From the above results we may deduce that there exist an optimal permutation schedule,(i.e., job sequence is the same on each processor) for two and three stage flow shop with makespan criterion[39].

Problem $F2|p_{j1} = b_{j1} - a_1 u_{j1}, p_{j2} = b_{j2}, \Sigma u_j \leq \hat{R}|C_{max}$ is NP-hard (see Table 2). Hence, more general problem $F2|p_j = b_j - a u_j, \Sigma u_j \leq \hat{R}|C_{max}$ is also NP-hard. However, there are many cases solvable in polynomial time. The optimal resource allocation for a given job sequence may be obtained by algorithm with the complexity $O(n)$ for problem $F2|p_{j1} = b_{j1} - a_1 u_{j1}, p_{j2} = b_{j2}, \Sigma u_j \leq \hat{R}|C_{max}$. Optimal polynomial algorithm with the complexity $O(n^2)$ for problem $F2|p_{j1} = b_{j1} - a_1 u_{j1}, p_{j2} = b_{j2}, \Sigma u_j \leq \hat{R}|C_{max}$ was presented in [32],[37],[39]. In these papers, an algorithms with the complexity $O(n \log n)$ were presented for problems $F2|p_{j1} = b_{j1} - a_1 u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_{j2}, \Sigma u_j \leq \hat{R}|C_{max}$ and $F2|p_{j1} = b_{j1} - a_1 u_{j1}, \beta_{j1} = \beta_1, p_{j2} = b_2, \Sigma u_j \leq \hat{R}|C_{max}$.

For more general NP-hard problem $F2|p_j = b_j - a u_j, \Sigma u_j \leq \hat{R}|C_{max}$ there exists an algorithm of optimal resource allocation with respect to a given sequence that runs in $O(n^3)$ time. Eight different 2-approximation algorithms were presented in [32] and this estimation is tight. Author also presents some numerical experiments and proposes a very fast branch and bound algorithm.

Flow shop problems with more than 2 stages are NP-hard as mentioned in the beginning of this paragraph. Hence, the existence of polynomial algorithms for these problem is highly unlikely. Approximation algorithms, optimal enumerative algo-

rithms, and algorithms for resource allocation for a given job sequence are mostly constructed. Many elimination properties which allow to bound the solution space for the search of the optimal solution are presented in [32],[39],[9]. These properties allow to construct more efficient approximation and optimal non-polynomial algorithms. A branch and bound algorithm which can solve the instances with up to 100 jobs and some approximation algorithms are presented in [32],[9]. Genetic algorithms are presented in [40],[46],[39]. Resource allocation algorithms for a given permutation are considered in [32]. For linear resource functions $f(\cdot)$, the problem may be transformed into the network programming problem. Modified pseudopolynomial algorithm from [18] is used to solve the problem. For more general problem, where $f(\cdot)$ is convex, the optimal solution may be obtained by convex programming algorithms.

Non-permutation flow shop is considered in [27]. An approximation algorithm and a branch and bound algorithm are proposed. The author also presets some specific problem properties on the basis of the graph representation of the schedule.

In the sequel we present the results obtained for the problems with Σu_j criterion under a given constraint on the makespan value.

Recall that for the two stage flow shop problems with regular criterion we can restrict the search of an optimal solution to permutation schedules only.

Problem $F2|p_{j1} = b_{j1} - a_1 u_{j1}, p_{j2} = b_{j2}, C_{max} \leq \hat{C}|\Sigma u_j$ (or more generally, $F2|p_j = b_j - a u_j, C_{max} \leq \hat{C}|\Sigma u_j$) is NP-hard as shown in [31]. Some special cases are solvable in polynomial time. An optimal resource allocation for a given job sequence may be obtained in $O(n)$ time by the algorithm proposed in[39]. In [30] it was shown that problem $F2|p_{j1} = b_{j1} - a_1 u_{j1}, p_{j2} = b_{j2}, C_{max} \leq \hat{C}|\Sigma u_j$ may be solved in $O(n \log n)$ time if one of the following relations hold:

1) $\beta_{j1} = \beta_1$, $p_{j2} = b_{j2}$;
2) $b_{j1} = b_1$, $p_{j2} = b_{j2}$;
3) $b_{j1} = b_1$, $\beta_{j1} = \beta_1$.

The special cases of the general two processor flow shop problem, in which the following relations hold 1) $b_{j1} = b_1$, $\beta_{j1} = \beta_1$, $\beta_{j2} = 0$, $p_{j2} = b_{j2}$; 2) $b_{j1} = b_1$, $b_{j2} = b_2$, $a_{j1} = a_1$, $a_{j2} = a_2$, $\beta_{j1} = \beta_1$, $\beta_{j2} = \beta_2$ may be resolved in time $O(n \log n)$ and $O(n)$, respectively. For problem $F2|p_j = b_j - a u_j, C_{max} \leq \hat{C}|\Sigma u_j$ eight heuristic algorithms are proposed in [32] and experimentally analyzed, however, worst-case performance of these algorithms is still an open question [39].

The classical permutation flow shop problems with three or more processors are strongly NP-hard [15]. Thus, the case with processing times linearly dependent on the resources are also strongly NP-hard. However, in [39] the author proposes an $O(nm)$ time optimal algorithm of resource allocation with respect to a given job permutation, where m is the number of processing stages. Also heuristic algorithm is presented for the general permutation problem with linear models of processing times in [28].

In what follows, we present the results obtained for the two-criteria flow shop problems, i.e., with $C_{max} \wedge u_j$ criterion.

Problem $F2|p_j = b_j - a u_j|C_{max} \wedge u_j$ is at least as difficult as $F2|p_j = b_j - a u_j, C_{max} \leq \hat{C}|\Sigma u_j$ which is NP-hard [32]. Similarly to the problems with C_{max} and

$\sum u_j$ criteria, in two-criteria approach, the search may be restricted to the permutation schedules only.

In [32] it was shown that the following special cases may be solved in polynomial time:

1) $a_{j1} = a_1$, $\beta_{j1} = \beta_1$, $\beta_{j2} = 0$, $p_{j2} = p_2$;
2) $a_{j1} = a_1$, $\beta_{j2} = 0$, $b_{j1} = b_1$, $p_{j2} = p_2$;
3) $a_{j1} = a_1$, $\beta_{j2} = 0$, $b_{j1} = b_1$, $\beta_{j1} = \beta_1$;
4) $\beta_{j2} = 0$, $b_{j1} = b_1$, $\beta_{j1} = \beta_1$, $p_{j2} = p_2$;
5) $b_{j1} = b_1$, $b_{j2} = b_2$, $a_{j1} = a_1$, $a_{j2} = a_2$, $\beta_{j1} = \beta_1$, $\beta_{j2} = \beta_2$.

The proportionate flow shop problem $Fm|p_{ij} = p_j - u_j|C_{\max} \wedge \Sigma v_j u_j$ was considered in [13]. The authors showed that this problem is solvable in polynomial time and propose an algorithm which requires $O(n \log n)$ time to find optimal solution. Recall that in proportionate flow shop problems, each job has the same processing time at each stage.

Problem $F2|p_{1j} = a_j - u_j, p_{2j} = a_j - u'_j, 0 \leq u_j \leq \beta_j, 0 \leq u'_j \leq \beta'_j|C_{max} + \Sigma(v_j u_j + v'_j u'_j)$ was considered in [66]. The authors prove that the problem is NP-hard and propose some heuristic algorithms.

In [64], problem $FP|p_{ij} = a_{ij} - u_{ij}, 0 \leq u_{ij} \leq \beta_{ij}|C_{max} + \Sigma v_{ij} u_{ij}$ was considered. Author proves that the problem is NP-hard and propose some approximation algorithms. To the best of our knowledge, the only polynomially solvable case of the hybrid flow shop problem is $FP|p_{ji} = b - a u_{ji}, \beta_{ji} = \beta, j = 1, \ldots, n, i = 1, \ldots, m|C_{max} \wedge \Sigma u_j$, considered in [39]. The author proposed an optimal algorithm with complexity equal to $O(nm)$. Notice, that in this problem any permutation of jobs is optimal, because the resource functions are identical for each job.

4.3. JOB SHOP

The job shop scheduling problems are at lest as difficult as flow shop problem because the flow shop is a special case of the job shop.

Problem $J|prec, p_j = f_j^w(u_j), \Sigma u_j \leq \hat{R}|C_{max}$, where $f_j(\cdot)$ are convex, non-increasing functions have been studied in [21] where the author proposes a representation of the schedule by a disjunctive graph. Some properties for this graph may be found in [32], [47], [39]. These properties are used in construction of branch and bound algorithms proposed in [21], [16], [32], [47]. The most efficient is the algorithm presented in [47]. The experimental comparison of above algorithms is presented in [32] and [47].

The hybrid job shop problem with resource dependent job processing time is considered in [28]. Some properties of the problem are proved and these properties are used to construct an algorithm solving this problem. The algorithm uses the disjunctive graph schedule representation and branch and bound technique.

The polynomial time approximation schemes (PTAS) for job shop scheduling problems with makespan plus total resource consumption criterion, are presented in [48]. The authors, however, assume that the number of processors and the number of operations per job are the fixed given values.

5. SINGLE PROCESSOR PROBLEMS WITH THE RESOURCE DEPENDENT RELEASE DATES

Scheduling problems with resource dependent release dates were introduced by Janiak in late 80's. Problem $1|prec, r_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|C_{max}$ was considered in [31], [34] and [39]. If resource allocation is given then the problem may be reduced to the classical $1|prec, r_j|C_{max}$ problem which is solvable in $O(n \log n)$ time by the modified Jackson's algorithm [19]. On the other hand, if the sequence of jobs is given then the optimal resource allocation may be found in $O(n^2)$ time by the algorithm proposed in [31]. Nevertheless, it was shown in [34] that the problem is NP-hard. Some special cases of the problem were considered in [31] and [39]. If we assume that $b_j = b$ for $j \in J$ then the optimal resource allocation for fixed job sequence can be obtained in $O(n)$ time [31]. Problem $1|prec, r_j = b - au_j, \beta_j = \beta, p_j = p, \Sigma u_j \leq \hat{R}|C_{max}$ may be solved optimally by an algorithm which runs in $O(n)$ time [39]. For the general problem $1|prec, r_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|C_{max}$ two approximation 2-algorithms are proposed in [39] and the estimation of the upper bound of relative error for both of them is tight.

Problem $1|prec, r_j = b_j - a_j u_j, C_{max} \leq \hat{C}|\Sigma u_j$ is NP-hard even if the functions of the release dates are identical for all the jobs [35]. An approximation algorithm for this problem is proposed in [39]. However, the worst case performance for this algorithm is still an open question. The special case: $1|prec, r_j = b - au_j, \beta_j = \beta, p_j = p, C_{max} \leq \hat{C}|\Sigma u_j$ may be solved in time $O(n)$ [35].

Problem $1|prec, r_j = b_j - a_j u_j|C_{max} \wedge \Sigma u_j$ is NP-hard [36] and the only case which may be solved in polynomial time is $1|prec, r_j = b - au_j, \beta_j = \beta, p_j = p|C_{max} \wedge \Sigma u_j$ [36].

Problem $1|r_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|C_{max}$ is strongly NP-hard [31],[38]. However, some of its special cases may be solved in polynomial time by the algorithm based on a sorting procedure:

a) $b_j = b$, $a_j = a$, $\beta_j = \beta - p_j$ $\searrow$,
b) $b_j = b$, $\beta_j = \beta$, $p_j = p - a_j$ $\searrow$,
c) $b_j = b$, $a_j = a$, $p_j = p - \beta_j$ $\searrow$,
d) $b_j = b$, $a_j = a$, $p_j = p - b_j$ $\searrow$,

where the term x_j $\searrow$ means that the optimal sequence of jobs can be obtained by ordering the jobs according to non-increasing values of the parameter $x_j \in \{p_j, a_j, \beta_j, b_j\}$.

Since the general problem $1|r_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|C_{max}$ is NP-hard, some general approximation scheme is proposed in [32] for this problem. The scheme is as follows:

1) construct the sequence of the jobs according to one of the 26 rules presented in [32] - the complexity $O(n \log n)$,
2) allocate the resource optimally by the algorithm for $1|prec, r_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|C_{max}$ - the complexity $O(n^2)$.

Notice that the complexity of each algorithm constructed according to presented above scheme equals $O(n^2)$. In [32], the author proves that every algorithm based on presented above scheme is a 2-approximation algorithm and this estimation is tight. Numerical experiments, however, show that solutions delivered by the algorithms are significantly better than the worst case performance ratio.

Problem $1|r_j = b_j - a_j u_j, C_{max} \leq \hat{C}|\Sigma u_j$ is strongly NP-hard [33]. However, some cases mentioned bellow below may be solved in time $O(n \log n)$ by a simple sorting algorithm:

a) $b_j = b$, $a_j = a$, $\beta_j = \beta - p_j$ $\searrow$,
b) $b_j = b$, $\beta_j = \beta$, $p_j = p - a_j$ $\searrow$,
c) $b_j = b$, $a_j = a$, $p_j = p - \beta_j$ $\searrow$,
d) $b_j = b$, $\beta_j = \beta$, $p_j = p - b_j$ $\searrow$.

The optimal resource allocation may be obtained in $O(n)$ steps by the algorithm proposed for $1|prec, r_j = b_j - a_j u_j, C_{max} \leq \hat{C}|\Sigma u_j$. Some heuristic algorithms for the general case of the problem are proposed in [33], but their worst case performance is an open question.

Problem $1|r_j = b_j - a_j u_j|C_{max} \wedge \Sigma u_j$ is strongly NP-hard [33]. However, some cases mentioned below can be solved in time $O(n \log n)$ by sorting-type algorithms:

a) $b_j = b$, $a_j = a$, $\beta_j = \beta - p_j$ $\searrow$,
b) $b_j = b$, $\beta_j = \beta$, $p_j = p - a_j$ $\searrow$,
c) $b_j = b$, $a_j = a$, $p_j = p - \beta_j$ $\searrow$,
d) $b_j = b$, $a_j = a$, $p_j = p - b_j$ $\searrow$,

and the application of the resource allocation algorithm. The resource allocation is reduced to constructing a time-resource trade-off curve. The curve is convex, decreasing, piecewise linear and may be determined in $O(n)$ time for cases a), b), c) and $O(n \log n)$ time for the case d) [39].

Problems:

a) $1|r_j = b_j - u_j, \Sigma u_j \leq \hat{R}|C_{max}$,
b) $1|r_j = b_j - u_j, C_{max} \leq \hat{C}|\Sigma u_j$,
c) $1|r_j = b_j - u_j|C_{max} \wedge \Sigma u_j$,

were considered in [67]. The authors prove that all of them are strongly NP-hard. For problem a), the family of ten 2-approximation algorithms based on same scheme is presented. The scheme is similar to the scheme for $1|r_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}|C_{max}$ mentioned above. In step 2, the optimal resource allocation is done by solving an appropriate linear programming task. The author proves that the worst case performance ratio 2 is the tight estimation.

There are some results in the literature for more general function for the release dates: $r_j = f(u_j)$, $\alpha \leq u_j \leq \beta$ for $j \in J$ where $f(j)$ is a continuous, strictly decreasing and nonnegative function.

Problem $1|prec, r_j = f(u_j), \Sigma u_j \leq \hat{R}|C_{max}$ was considered in [39]. The author proved that the problem is NP-hard and developed optimal resource allocation algorithm which runs in $O(\max\{g(n), n\})$ steps, where $O(g(n))$ is the time required

to calculate the values of f,and the values of f^{-1} [24]. The author also showed that problem $1|prec, r_j = f(u_j), p_j = p, \Sigma u_j \leq \hat{R}|C_{max}$ may be solved optimally by taking any feasible sequence of jobs and calculating optimal resource allocation in the same way as for the previous problem [39]. In [24], the author showed that problem $1|r_j = f(u_j), \Sigma u_j \leq \hat{R}|C_{max}$ may be solved by sequencing the jobs according to nondecreasing values of p_j and calculating the optimal resource allocation by the algorithm with the complexity $O(\max\{g(n), n \log n\})$, where $g(n)$ is defined in the same way as previously.

Problem $1|prec, r_j = f(u_j), C_{max} \leq \hat{C}|\Sigma u_j$ is NP-hard [39]. Similar to the previous group of problems, for a given sequence of jobs, the optimal resource allocation can be obtained in $O(\max\{g(n), n\})$ time, where $g(n)$ is defined in the same way as previously [8]. Similarly, problem $1|prec, r_j = f(u_j), p_j = p, C_{max} \leq \hat{C}|\Sigma u_j$ may be solved by taking any feasible sequence of jobs and perform resource allocation [39].

Analogously problem $1|r_j = f(u_j), C_{max} \leq \hat{C}|\Sigma u_j$ may be solved by sequencing the jobs in nondecreasing order of p_j and next by allocating the resource optimally in $O(\max\{g(n), n\})$ time [8].

Problem $1|prec, r_j = f(u_j)|C_{max} \wedge \Sigma u_j$ is NP-hard [39]. However, problem $1|prec, r_j = f(u_j)|C_{max} \wedge \Sigma u_j$ is easy to solve. The optimal sequence of jobs can be obtained by sorting them according to the nondecreasing values of p_j and then, the problem is reduced to calculate the time-cost trade-off curve. This may be calculated in $O(\max\{g(n), n \log n\})$ where $g(n)$ is defined as above [32]. Problem $1|prec, r_j = f(u_j), p_j = p|C_{max} \wedge \Sigma u_j$ may be solved in the analogous way [39].

Problem $1|r_j = f(u_j), \Sigma C_j \leq \hat{C}|\Sigma u_j$ is NP-hard [60]. The case with linear resource function, i.e., $1|r_j = b - au_j, \Sigma C_j \leq \hat{C}|\Sigma u_j$ may be solved by an algorithm proposed in [60] that requires $O(n \log n)$ time. The extension of the results obtained in [60] are presented in [76], where some properties are proved for the convex decreasing resource function.

The general case of the problems with $\Sigma w_j C_j$ criterion with nonlinear models of the release dates are NP-hard or strongly NP-hard as it is presented in Table 3. Some special cases, however, may be solved in polynomial time.

Problems $1|r_j = b - a(\delta_j u_j), 1/\delta_j = w_j, \Sigma u_j \leq \hat{R}|\Sigma w_j C_j$, $1|r_j = b - a(\delta_j u_j), 1/\delta_j = w_j, \Sigma w_j C_j \leq \hat{C}|\Sigma u_j$, and $1|r_j = b - a(\delta_j u_j), 1/\delta_j = w_j|\Sigma w_j C_j \wedge \Sigma u_j$, may be solved optimally in $O(n \log n)$ steps [39]. The algorithms use the SWPT rule, i.e., the optimal sequence of jobs may be obtained by sorting them according to the nondecreasing values of p_j/w_j.

The optimal algorithms with complexity $O(n \log n)$ are presented in [59] for the following problems:
$1|r_j = f_j(u_j), f_j^{-1}(u_j) = p_j f_j^{-1}(u_j)|C_{max} \leq \hat{C}|\Sigma u_j$,
$1|r_j = f_j(u_j), f_j^{-1}(u_j) = p_j f_j^{-1}(u_j)|\Sigma C_j \leq \hat{C}|\Sigma u_j$, and
$1|r_j = f_j(u_j), f_j^{-1}(u_j) = p_j f_j^{-1}(u_j)|v\Sigma u_j + wC_{max}$.

Problem $1|r_j = f_j(u_j)|v\Sigma u_j + w\Sigma C_j$ is shown in [59] to be NP-hard.

Problems $1|r_j = f_j(u_j), d_j = d|\Sigma u_j + \Sigma w T_j$ and $1|r_j = f_j(u_j), d_j = d, f_j^{-1}(u_j) = p_j f_j^{-1}(u_j)|w_j = kp_j|\Sigma u_j + \Sigma w T_j$ are NP-hard [61].

Some properties of the optimal solution and some dynamic programming algorithms are presented in [61].

The results presented in this section are summarized in Table 3.

Table 3. The complexity of problems with resource dependent release dates

Problem	Complexity	Reference
$1\|r_j = b_j - a_j u_j, \Sigma u_j \leq \hat{R}\|C_{max}$	strongly NP-hard	[31], [38]
$1\|r_j = b_j - a_j u_j, C_{max} \leq \hat{C}\|\Sigma u_j$	strongly NP-hard	[33]
$1\|r_j = b_j - a_j u_j\|C_{max} \wedge \Sigma u_j$	strongly NP-hard	[33]
$1\|r_j = b_j - a u_j, \Sigma u_j \leq \hat{R}\|C_{max}$	NP-hard	[31], [38]
$1\|r_j = b_j - a u_j, C_{max} \leq \hat{C}\|\Sigma u_j$	NP-hard	[33]
$1\|r_j = b_j - a u_j\|C_{max} \wedge \Sigma u_j$	NP-hard	[33]
$1\|r_j = b - a u_j, \beta_j = \beta, \Sigma u_j \leq \hat{R}\|C_{max}$	$O(n \log n)$	[31], [38]
$1\|r_j = b - a u_j, \beta_j = \beta, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[33]
$1\|r_j = b - a u_j, \beta_j = \beta\|C_{max} \wedge \Sigma u_j$	$O(n \log n)$	[39]
$1\|r_j = b - a_j u_j, \beta_j = \beta, p_j = p, \Sigma u_j \leq \hat{R}\|C_{max}$	$O(n \log n)$	[31], [38]
$1\|r_j = b - a_j u_j, \beta_j = \beta, p_j = p, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[33]
$1\|r_j = b - a_j u_j, \beta_j = \beta, p_j = p\|C_{max} \wedge \Sigma u_j$	$O(n \log n)$	[39]
$1\|r_j = b - a u_j, p_j = p, \Sigma u_j \leq \hat{R}\|C_{max}$	$O(n \log n)$	[31], [38]
$1\|r_j = b - a u_j, p_j = p, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[33]
$1\|r_j = b - a u_j, p_j = p\|C_{max} \wedge \Sigma u_j$	$O(n \log n)$	[39]
$1\|r_j = b - a u_j, \beta_j = \beta, p_j = p, \Sigma u_j \leq \hat{R}\|C_{max}$	$O(n \log n)$	[31], [38]
$1\|r_j = b - a u_j, \beta_j = \beta, p_j = p, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[33]
$1\|r_j = b - a u_j, \beta_j = \beta, p_j = p\|C_{max} \wedge \Sigma u_j$	$O(n \log n)$	[39]
$1\|prec, r_j = b - a u_j, \beta_j = \beta, \Sigma u_j \leq \hat{R}\|C_{max}$	NP-hard	[34]
$1\|prec, r_j = b - a u_j, \beta_j = \beta, C_{max} \leq \hat{C}\|\Sigma u_j$	NP-hard	[35]
$1\|prec, r_j = b - a u_j, \beta_j = \beta\|C_{max} \wedge \Sigma u_j$	NP-hard	[39]
$1\|prec, r_j = b - a u_j, \beta_j = \beta, p_j = p, \Sigma u_j \leq \hat{R}\|C_{max}$	$O(n)$	[39]
$1\|prec, r_j = b - a u_j, \beta_j = \beta, p_j = p, C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n)$	[35]
$1\|prec, r_j = b - a u_j, \beta_j = \beta, p_j = p\|C_{max} \wedge \Sigma u_j$	$O(n)$	[36]
$1\|r_j = b_j - u_j, \Sigma u_j \leq \hat{R}\|C_{max}$	strongly NP-hard	[67]
$1\|r_j = b_j - u_j, C_{max} \leq \hat{C}\|\Sigma u_j$	strongly NP-hard	[67]
$1\|r_j = b_j - u_j\|C_{max} \wedge \Sigma u_j$	strongly NP-hard	[67]

Table 3 (continued)

Problem	Complexity	Reference
$1\|r_j = f(u_j), \Sigma u_j \leq \hat{R}\|C_{max}$	$O(\max\{g(n), n\log n\})$ where $O(g(n))$ is the complexity of calculation of $f^{-1}()$	[24]
$1\|r_j = f(u_j), C_{max} \leq \hat{C}\|\Sigma u_j$	as above	[8]
$1\|r_j = f(u_j)\|C_{max} \wedge \Sigma u_j$	as above	[32]
$1\|r_j = f(u_j), p_j = p, \Sigma u_j \leq \hat{R}\|C_{max}$	as above	[39]
$1\|r_j = f(u_j), p_j = p, C_{max} \leq \hat{C}\|\Sigma u_j$	as above	[39]
$1\|r_j = f(u_j), p_j = p\|C_{max} \wedge \Sigma u_j$	as above	[39]
$1\|r_j = g(u_j), \Sigma u_j \leq \hat{R}\|\Sigma C_j$	NP-hard	[39]
$1\|r_j = g(u_j), \Sigma C_j \leq \hat{C}\|\Sigma u_j$	NP-hard	[60], [39]
$1\|r_j = g(u_j)\|\Sigma C_j \wedge \Sigma u_j$	NP-hard	[39]
$1\|r_j = g(\delta_j u_j), \Sigma u_j \leq \hat{R}\|\Sigma w_j C_j$	strongly NP-hard	[44]
$1\|r_j = g(\delta_j u_j), \Sigma w_j C_j \leq \hat{C}\|\Sigma u_j$	strongly NP-hard	[39]
$1\|r_j = g(\delta_j u_j)\|\Sigma w_j C_j \wedge \Sigma u_j$	strongly NP-hard	[39]
$1\|r_j = b - au_j, \Sigma C_j \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[60]
$1\|r_j = b - a(\delta_j u_j), 1/\delta_j = w_j, \Sigma u_j \leq \hat{R}\|\Sigma w_j C_j$	$O(n \log n)$	[39]
$1\|r_j = b - a(\delta_j u_j), 1/\delta_j = w_j, \Sigma w_j C_j \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[39]
$1\|r_j = b - a(\delta_j u_j), 1/\delta_j = w_j\|\Sigma w_j C_j \wedge \Sigma u_j$	$O(n \log n)$	[39]
$1\|r_j = f_j(u_j), d_j = d\|\Sigma u_j + \Sigma w T_j$	NP-hard	[61]
$1\|r_j = f_j(u_j), d_j = d, f_j^{-1}(u_j) = p_j f_j^{-1}(u_j), w_j = kp_j\|\Sigma u_j + \Sigma w T_j$	NP-hard	[61]
$1\|r_j = f_j(u_j), f_j^{-1}(u_j) = p_j f_j^{-1}(u_j), C_{max} \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[59]
$1\|r_j = f_j(u_j), f_j^{-1}(u_j) = p_j f_j^{-1}(u_j), \Sigma C_j \leq \hat{C}\|\Sigma u_j$	$O(n \log n)$	[59]
$1\|r_j = f_j(u_j), f_j^{-1}(u_j) = p_j f_j^{-1}(u_j)\|v\Sigma u_j + wC_{max}$	$O(n \log n)$	[59]
$1\|r_j = f_j(u_j)\|v\Sigma u_j + w\Sigma C_j$	NP-hard	[59]
$1\|u_j = f_j(r_j), \Sigma C_j \leq \hat{C}\|\Sigma f(r_j)$ and $f()$ is decreasing and convex	NP-hard	[76]

6. SCHEDULING PROBLEMS WITH CONVEX FUNCTION OF JOB PROCESSING TIMES

In this section we present the results for the problems with job processing times given as a convex functions, i.e., the processing time, p_j, of job j is given by the following formulae:

$$p_j = \left(\frac{a_j}{u_j}\right)^k,$$

where a_j and k are the resource consumption parameters.

Problem $1|p_j = (a_j/u_j)^k, \Sigma u_j \leq U|L_{\max}$ was considered in [70]. In this problem, the optimal sequence of jobs does not depend on the resource allocation and may be found by applying the EDD rule [12], i.e., by sorting the jobs in nondecreasing order of d_j. Then the optimal resource allocation may be found by the algorithm which requires $O(n^3)$ time proposed in [70]. It is easy to notice that in problem $1|p_j = b_j/u_j|C_{\max} + \Sigma v_j u_j$, the optimal sequence of jobs also does not depend on the resource allocation. Hence, the algorithm proposed in [70] can be applied to solve the makespan minimization problem.

In [70] authors consider also two-resource allocation problem $1|p_j = \max\{(a_{1j}/u_{1j})^k, (a_{2j}/u_{2j})^k\}, \Sigma u_{1j} \leq U_1, \Sigma u_{2j} \leq U_2|L_{\max}$. The algorithm with the complexity $O(n^3 \log 1/\epsilon)$, where ϵ is a given accuracy, was proposed.

Problem $1|p_j = b_j + a_j/u_j, \Sigma u_j \leq U|\Sigma C_j$ was considered in [57]. In general the problem is NP-hard. However, some special cases of the problem are solvable in $O(n \log n)$ time, namely the case with parameter a_j identical for all the jobs ($a_j = a$) and the case with parameter b_j identical for all the jobs ($b_j = b$).

Problem $1|p_j = (a_j/u_j)^k, \Sigma u_j \leq U|\Sigma w_j C_j$ was considered in [71]. The problem was reduced to discrete optimization problem, however, the computational complexity of the problem remains an open question. A dynamic programming algorithm and approximation algorithms were proposed.

Five polynomially solvable special cases of the problem were identified:

1) $w_j = w$,
2) $a_j = a$,
3) $w_j = \alpha a_j$,
4) $a_i \leq a_j \wedge w_i \geq w_j$ for each $i \neq j$
5) $(a_i^{k/(k+1)}/w_i^{k/(k+1)}) \leq (a_j^{k/(k+1)}/w_j^{k/(k+1)}) \wedge a_i \geq a_j$ for each $i \neq j$.

All these cases are solvable in $O(n \log n)$ time.

Problem $1|r_j = (b_j/v_j)^k, p_j = (a_j/u_j)^k|C_{\max} \wedge \Sigma u_j + \Sigma v_j$ was considered in [54]. The optimal resource allocation as a function of the job sequence can be determined in a linear time, and the optimal job sequence is independent of the total resources being used. Thereby, the problem may be reduced to the sequencing one. However, the computational complexity of presented problem is open. A dynamic programming algorithm and approximation algorithms were proposed.

Some special cases of the presented problem are solvable in $O(n \log n)$ time:

1) $b_j = b$,
2) $a_j = a$,
3) $a_j = \alpha b_j$,
4) $b_i \leq b_j \wedge a_i \geq a_j$ for each $i \neq j$
5) $a_i \leq a_j \wedge b_i/a_i \geq b_j/a_j$ for each $i \neq j$.

We summarize the results presented in this section in Table 4.

Table 4. The computational complexity of the problems with convex function of job processing times

Problem	Complexity	Reference
$1\|p_j = (a_j/u_j)^k, \Sigma u_j \leq U\|L_{\max}$	$O(n \log n)$	[70]
$1\|p_j = \max\{(a_{1j}/u_{1j})^k, (a_{2j}/u_{2j})^k\}, \Sigma u_{1j} \leq U_1, \Sigma u_{2j} \leq U_2\|L_{\max}$	$O(n^3 \log 1/\epsilon)$	[70]
$1\|p_j = b_j + a_j/u_j, \Sigma u_j \leq U\|\Sigma C_j$	NP-hard	[57]
$1\|p_j = b + a_j/u_j, \Sigma u_j \leq U\|\Sigma C_j$	$O(n \log n)$	[57]
$1\|p_j = b_j + a/u_j, \Sigma u_j \leq U\|\Sigma C_j$	$O(n \log n)$	[57]
$1\|p_j = (a_j/u_j)^k, \Sigma u_j \leq U\|\Sigma w_j C_j$	Open	[71]
$1\|p_j = (a_j/u_j)^k, \Sigma u_j \leq U\|\Sigma C_j$	$O(n \log n)$	[71]
$1\|p_j = (a/u_j)^k, \Sigma u_j \leq U\|\Sigma w_j C_j$	$O(n \log n)$	[71]
$1\|p_j = (a_j/u_j)^k, w_j = \alpha a_j, \Sigma u_j \leq U\|\Sigma w_j C_j$	$O(n \log n)$	[71]
$1\|p_j = (a_j/u_j)^k, \Sigma u_j \leq U\|\Sigma w_j C_j$ and $b_i \leq b_j \wedge a_i \geq a_j$ for each $i \neq j$	$O(n \log n)$	[71]
$1\|p_j = (a_j/u_j)^k, \Sigma u_j \leq U\|\Sigma w_j C_j$ and $(a_i^{k/(k+1)}/w_i^{k/(k+1)}) \leq (a_j^{k/(k+1)}/w_j^{k/(k+1)}) \wedge a_i \geq a_j$ for each $i \neq j$	$O(n \log n)$	[71]
$1\|r_j = (b_j/v_j)^k, p_j = (a_j/u_j)^k\|C_{\max} \wedge \Sigma u_j + \Sigma v_j$	Open	[54]
$1\|r_j = (b/v_j)^k, p_j = (a_j/u_j)^k\|C_{\max} \wedge \Sigma u_j + \Sigma v_j$	$O(n \log n)$	[54]
$1\|r_j = (b_j/v_j)^k, p_j = (a/u_j)^k\|C_{\max} \wedge \Sigma u_j + \Sigma v_j$	$O(n \log n)$	[54]
$1\|r_j = (b_j/v_j)^k, p_j = (a_j/u_j)^k, a_j = \alpha b_j\|C_{\max} \wedge \Sigma u_j + \Sigma v_j$	$O(n \log n)$	[54]
$1\|r_j = (b_j/v_j)^k, p_j = (a_j/u_j)^k\|C_{\max} \wedge \Sigma u_j + \Sigma v_j$ and $b_i \leq b_j \wedge a_i \geq a_j$ for each $i \neq j$	$O(n \log n)$	[54]
$1\|r_j = (b_j/v_j)^k, p_j = (a_j/u_j)^k\|C_{\max} \wedge \Sigma u_j + \Sigma v_j$ and $a_i \leq a_j \wedge b_i/a_i \geq b_j/a_j$ for each $i \neq j$	$O(n \log n)$	[54]

7. SCHEDULING PROBLEMS WITH DISCRETE RESOURCES

The problems with discrete resources were widely studied in [10] and [39]. The results presented in [39] complement the results presented in [10]. Thus, the results from both above mentioned publications will be presented jointly.

We consider the following problems:

$1|p_j^d = b_j - a_j u_j^d, F_1 \leq \hat{U}|F_2$,
$1|p_j^d = b_j - a_j u_j^d, F_2 \leq \hat{C}|F_1$,
$1|p_j^d = b_j - a_j u_j^d|F_1 \wedge F_2$

where F_1 is the resource penalty function, i.e., $F_1 \in \{g_{\max}, \Sigma u_j, \Sigma v_j u_j\}$, and F_2 is the time criterion function, i.e., $F_2 \in \{C_{\max}, c_{\max}, \Sigma U_j, \Sigma w_j U_j, \Sigma C_j, \Sigma w_j C_j\}$, where

$g_{max} = \max_{j \in J}\{g_j(u_j)\}$ and g_j is some general nondecreasing resource consumption cost function.

In [10] and [39] the following problems were classified as NP-hard:

$1|p_j^d = b_j - a_j u_j^d, g_{\max} \le K|\Sigma w_j U_j,$
$1|p_j^d = b - a_j u_j^d, \beta_j = 1, \Sigma v_j u_j \le K|C_{\max},$
$1|p_j^d = b - a_j u_j^d, \beta_j = 1, C_{\max} \le K|\Sigma v_j u_j,$
$1|p_j^d = b - a_j u_j^d, \beta_j = 1|\Sigma v_j u_j \wedge C_{\max},$
$1|p_j^d = b_j - a_j u_j^d, \beta_j = 1, \Sigma v_j u_j \le K|\Sigma w_j C_j,$
$1|p_j^d = b_j - a_j u_j^d, \beta_j = 1, \Sigma w_j C_j \le K|\Sigma v_j u_j,$
$1|p_j^d = b_j - a u_j^d, d_j = d, \Sigma U_j \le K|\Sigma u_j,$
$1|p_j^d = b_j - a u_j^d, d_j = d, \Sigma u_j \le K|\Sigma U_j,$
$1|p_j^d = b_j - a u_j^d, d_j = d|\Sigma U_j \wedge \Sigma u_j,$
$1|p_j^d = b_j - a_j u_j^d, \beta_j = 1, \Sigma v_j u_j \le K|\Sigma w_j C_j,$
$1|p_j^d = b_j - a_j u_j^d, \beta_j = 1, \Sigma w_j C_j \le K|\Sigma v_j u_j,$
$1|p_j^d = b_j - a_j u_j^d, \beta_j = 1|\Sigma w_j C_j \wedge \Sigma v_j u_j.$

The following problems can be solved in $O(n \log n)$ time:

$1|p_j^d = b_j - a_j u_j^d, g_{\max} \le K|F_2$, where $F_2 \in \{c_{max}, \Sigma U_j, \Sigma w_j C_j\}$,
$1|p_j^d = b_j - a u_j^d, c_{\max} \le K|\Sigma v_j u_j,$
$1|p_j^d = b_j - a_j u_j^d, c_{\max} \le K|\Sigma u_j,$
$1|p_j^d = b - a u_j^d, d_j = d, \Sigma u_j \le K|\Sigma C_j,$
$1|p_j^d = b_j - a u_j^d, C_j \le \overline{d}_j|\Sigma v_j u_j.$

In the publication mentioned above the general schemes for the construction of the Pareto set and the Pareto set ϵ-approximation are presented. The dynamic programming algorithms and approximation algorithms are proposed for the selected problems.

Problem with the deadlines $1|p_j^d = b_j - a_j u_j^d, C_j \le \overline{d}_j|\Sigma v_j u_j$ was considered in [39]. The problem was proved to be NP-hard even for the case $b_j = b, \overline{d}_j = d, \beta_j = 1$ for $j = 1, 2, \ldots, n$. Some approximation algorithm and a fully polynomial approximation scheme were presented. Some cases of the general problem were proved to be solvable in $O(n \log n)$ time by the mentioned above approximation algorithm for the general problem, namely: $1|p_j^d = b_j - a u_j^d, C_j \le \overline{d}_j|\Sigma v_j u_j$ and $1|p_j^d = b_j - a_j u_j^d, C_j \le \overline{d}_j|\Sigma u_j$.

The problems with the general resource consumption functions g_j were considered in [4]. Problems $1|p_j^d = b_j - a_j u_j^d|\Sigma C_j + \Sigma g_j(u_j)$ and $1|p_j^d = b_j - a_j u_j^d, d_j = d > D|\rho\Sigma E_j + \sigma\Sigma T_j + \Sigma g_j(u_j)$ were proved to be solvable in $O(n^3)$ steps by reducing them to an assignment problem. Problems $1|r_j, p_j^d = b_j - a_j u_j^d|C_{\max} + \Sigma g_j(u_j)$, $1|p_j^d = b_j - a_j u_j^d|T_{\max} + \Sigma g_j(u_j)$, $1|p_j^d = b_j - a_j u_j^d, d_j = d|T_{\max} + \Sigma g_j(u_j)$, $1|p_j^d = b_j - a_j u_j^d|\Sigma w_j U_j + \Sigma g_j(u_j)$, $1|p_j^d = b_j - a_j u_j^d, d_j = d|\Sigma w_j U_j + \Sigma g_j(u_j)$ were proved to be NP-hard and some optimal pseudopolynomial time algorithms based on the dynamic programming were proposed. This implies that the mentioned above problems are not strongly NP-hard.

The problems $Pm|p_j^d = b_j - a_j u_j^d|\Sigma w_j C_j + \Sigma v_j u_j$ and $Pm|p_j^d = b_j - a_j u_j^d|\Sigma w_j U_j + \Sigma v_j u_j$ were considered in [3]. Since both problems are NP-hard, a column generation branch and bound algorithm was proposed.

The results presented in this section are summarized in Table 5.

Table 5. The computational complexity of the problems with discrete resources

Problem	Complexity	Reference
$1\|p_j^d = b_j - a_j u_j^d, g_{\max} \le K\|F_2$ where $F_2 \in \{c_{max}, \Sigma U_j, \Sigma w_j C_j\}$	$O(n \log n)$	[10], [39]
$1\|p_j^d = b_j - a_j u_j^d, F_2 \le K\|g_{\max}$ where $F_2 \in \{c_{max}, \Sigma U_j, \Sigma w_j C_j\}$	$O(n \log n \log(\max\{g_j(\beta_j)\}))$	[10], [39]
$1\|p_j^d = b_j - a_j u_j^d\|F_2 \wedge g_{\max}$ where $F_2 \in \{c_{max}, \Sigma U_j, \Sigma w_j C_j\}$	$O(\|P\|n(\log n + \log(\max\{g_j(\beta_j)\})))$ where $\|P\|$ is the cardinality of the Pareto set	[39]
$1\|p_j^d = b_j - a_j u_j^d, g_{\max} \le K\|\Sigma w_j U_j$	NP-hard	[14]
$1\|p_j^d = b - a_j u_j^d, \beta_j = 1, \Sigma v_j u_j \le K\|C_{\max}$	NP-hard	[10]
$1\|p_j^d = b - a_j u_j^d, \beta_j = 1, C_{\max} \le K\|\Sigma v_j u_j$	NP-hard	[10]
$1\|p_j^d = b - a_j u_j^d, \beta_j = 1\|\Sigma v_j u_j \wedge C_{\max}$	NP-hard	[39]
$1\|p_j^d = b_j - a u_j^d, c_{\max} \le K\|\Sigma v_j u_j$	$O(n \log n)$	[10]
$1\|p_j^d = b_j - a u_j^d, \Sigma v_j u_j \le K\|c_{\max}$	$O(n \log n \log(\max\{c_j(\Sigma_{i=1}^n b_i)\}))$	[10]
$1\|p_j^d = b_j - a u_j^d\|c_{\max} \wedge \Sigma v_j u_j$	$O(\|P\|n(\log n + \log(\max\{c_j(\Sigma_{j=1}^n) b_j\})))$ where $\|P\|$ is the cardinality of the Pareto set	[39]
$1\|p_j^d = b_j - a_j u_j^d, c_{\max} \le K\|\Sigma u_j$	$O(n \log n)$	[10]
$1\|p_j^d = b_j - a_j u_j^d, \Sigma u_j \le K\|c_{\max}$	$O(n \log n \log(\max\{c_j(\Sigma_{i=1}^n b_i)\}))$	[10]
$1\|p_j^d = b_j - a_j u_j^d\|c_{\max} \wedge \Sigma u_j$	$O(\|P\|n(\log n + \log(\max\{c_j(\Sigma_{j=1}^n) b_j\})))$ where $\|P\|$ is the cardinality of the Pareto set	[39]
$1\|p_j^d = b_j - a_j u_j^d, \beta_j = 1, \Sigma v_j u_j \le K\|\Sigma w_j C_j$	NP-hard	[39]
$1\|p_j^d = b_j - a_j u_j^d, \beta_j = 1, \Sigma w_j C_j \le K\|\Sigma v_j u_j$	NP-hard	[39], [10]
$1\|p_j^d = b_j - a u_j^d, d_j = d, \Sigma U_j \le K\|\Sigma u_j$	NP-hard	[6]
$1\|p_j^d = b_j - a u_j^d, d_j = d, \Sigma u_j \le K\|\Sigma U_j$	NP-hard	[10]
$1\|p_j^d = b_j - a u_j^d, d_j = d\|\Sigma U_j \wedge \Sigma u_j$	NP-hard	[39]
$1\|p_j^d = b_j - a_j u_j^d, \Sigma v_j u_j \le K\|\Sigma w_j U_j$	$O(nK(\Sigma w_j)\Sigma\beta_j)$	[39]
$1\|p_j^d = b_j - a_j u_j^d, \Sigma w_j U_j \le K\|\Sigma v_j u_j$	$O(nK(\Sigma w_j)\Sigma\beta_j)$	[39]
$1\|p_j^d = b_j - a_j u_j^d\|\Sigma v_j u_j \wedge \Sigma w_j U_j$	$O(\|P\|n(\Sigma w_j)\Sigma(w_j + v_j\beta_j)\Sigma\beta_j)$	[39]
$1\|p_j^d = b - a u_j^d, \Sigma u_j \le K\|\Sigma C_j$	$O(n \log n)$	[10]
$1\|p_j^d = b - a u_j^d, \Sigma C_j \le K\|\Sigma u_j$	$O(n \log n \log(\Sigma\beta_j))$	[10]

Table 5 (continued)

Problem	Complexity	Reference
$1\|p_j^d = b - au_j^d\|\Sigma C_j \wedge \Sigma u_j$	$O(\|P\|n(\log n + \log(\Sigma\beta_j)))$ where $\|P\|$ is the cardinality of the Pareto set	[39]
$1\|p_j^d = b_j - a_j u_j^d, \beta_j = 1, \Sigma v_j u_j \leq K\|\Sigma w_j C_j$	NP-hard	[39]
$1\|p_j^d = b_j - a_j u_j^d, \beta_j = 1, \Sigma w_j C_j \leq K\|\Sigma v_j u_j$	NP-hard	[39]
$1\|p_j^d = b_j - a_j u_j^d, \beta_j = 1\|\Sigma w_j C_j \wedge \Sigma v_j u_j$	NP-hard	[39]
$1\|p_j^d = b_j - a_j u_j^d, \beta_j = 1, \overline{d}_j = \overline{d}, C_j \leq \overline{d}\|\Sigma v_j u_j$	NP-hard	[39]
$1\|p_j^d = b_j - au_j^d, C_j \leq \overline{d}_j\|\Sigma v_j u_j$	$O(nlogn)$	[39]
$1\|p_j^d = b_j - a_j u_j^d, C_j \leq \overline{d}_j\|\Sigma u_j$	$O(nlogn)$	[39]
$1\|p_j^d = b_j - a_j u_j^d\|\Sigma C_j + \Sigma g_j(u_j)$	$O(n^3)$	[4]
$1\|p_j^d = b_j - a_j u_j^d$, $d_j = d > D\|\rho\Sigma E_j + \sigma\Sigma T_j + \Sigma g_j(u_j)$	$O(n^3)$	[4]
$1\|r_j, p_j^d = b_j - a_j u_j^d\|C_{\max} + \Sigma g_j(u_j)$	NP-hard	[4]
$1\|p_j^d = b_j - a_j u_j^d\|T_{\max} + \Sigma g_j(u_j)$	NP-hard	[4]
$1\|p_j^d = b_j - a_j u_j^d, d_j = d\|T_{\max} + \Sigma g_j(u_j)$	NP-hard	[4]
$1\|p_j^d = b_j - a_j u_j^d\|\Sigma w_j U_j + \Sigma g_j(u_j)$	NP-hard	[4]
$1\|p_j^d = b_j - a_j u_j^d, d_j = d\|\Sigma w_j U_j + \Sigma g_j(u_j)$	NP-hard	[4]
$Pm\|p_j^d = b_j - a_j u_j^d\|\Sigma w_j C_j + \Sigma v_j u_j$	NP-hard	[3]
$Pm\|p_j^d = b_j - a_j u_j^d\|\Sigma w_j U_j + \Sigma v_j u_j$	NP-hard	[3]

8. CONCLUSIONS

Scheduling problems with additional resource allocation are quite new part of the scheduling theory, however, high practical significance of these problems attracts many researchers. In this paper we surveyed the results obtained in this area in available literature. It can be observed that the computational complexity status have been derived for most problems, but there are still some interesting open problems with respect to NP-hardness or strong NP-hardness.

The presented results indicate that there are some practical problems which may be resolved in the polynomial time. However, most of these problems, especially multi processor ones, are NP-hard or strongly NP-hard. In our opinion computationally effective algorithms for these problems should be developed. Future research should focus on problems with complex processor environment such as flow shop and job shop. There is a broad area for using specialized algorithms and techniques as tabu search, genetic algorithms, simulated annealing or algorithms which use artificial intelligence methods. The ability to allocate resources for a given job processing order in computationally effective manner may be very useful tool for mentioned above algorithms, and such tools should be also developed.

REFERENCES

1. B. Alidaee, A. Ahmadian, *Two parallel machine sequencing problems involving controllable job processing times*, European Journal of Operational Research, 70, (1993), 335–341.

2. J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, J. Weglarz, *Scheduling Computer and Manufacturing Processes*, Springer, Berlin, 2001.

3. Z. Chen, *Simultaneous job scheduling and resource allocation on parallel machines*, Annals of Operations Research, 129, (2004), 135–153.

4. Z. Chen, Q. Lu, G. Tang, *Single machine scheduling with discretely controllable processing times*, Operations Research Letters, 21, (1997), 69–76.

5. T. Cheng, Z. Chen, C.-L. Li, *Parrallel-machine scheduling with controllable processing times*, IIE Transactions, 28.

6. T. Cheng, Z.-L. Chen, C.-L. Li, *Single-machine scheduling with trade-off between number of tardy jobs and resource allocation*, Operations Research Letters, 16, (1996), 237–242.

7. T. Cheng, Z.-L. Chen, C.-L. Li, B.-T. Lin, *Scheduling to minimize the total compression and late costs*, Naval Research Logistics, 45, (1998), 67–82.

8. T. Cheng, A. Janiak, *Resource optimal control in some single-machine scheduling problems*, IEEE Transation on Automatic Control, 39 (6), (1994), 1243–1246.

9. T. Cheng, A. Janiak, *A permutation flow-shop scheduling problem with convex models of operation processing times*, Annals of Operations Research, 96, (2000), 39–60.

10. T. Cheng, A. Janiak, M. Kovalyov, *Bicriterion single machine scheduling with resource depandaent processing times*, SIAM Journal of Optimization, 8, (1998), 617–630.

11. T. Cheng, M. Kovalyov, *Single machine batch scheduling with deadlines and resource dependent processing times*, Operations Research Letters, 17, (1995), 243–248.

12. T. Cheng, C. Oguz, X. Qi, *Due-date assignment and single machine scheduling with compressible processing times*, International Journal of Production Economics, 43, (1996), 107–113.

13. T. Cheng, N. Shakhlevich, *Proportionate flow shop with controllable processing times*, Journal of Scheduling, 2, (1999), 253–265.

14. M. Garey, D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completness*, W.H. Freeman, San Francisco, 1979.

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

16. J. Grabowski, A. Janiak, *job-shop scheduling with resource-time models of operations*, European Journal of Operational Research, 28, (1987), 58–73.

17. R. Graham, E. Lawler, J. Lenstra, A. Rinnoy Kan, *Optimization and approximation in deterministic sequencing and scheduling theory: a survey*, Annals of Discrete Mathematics, 5, (1979), 287–326.

18. H. Hamacher, S. Tufekci, *Algebraic flows and time-cost tradeoff problems*, Annals of Discrete Mathematics, 19, (1984), 165–182.

19. J. Jackson, *Scheduling a production line to minimize maximum tardiness*, Management Science Research Project Research report 43, University of California, Los Angeles, CA., 1955.

20. A. Janiak, *Job scheduling problems on single machine with resource allocation(in polish)*, Zeszyty Naukowe Politechniki Slaskiej seria Automatyka, z. 84, 894, (1986), 81–92.

21. A. Janiak, *Job-shop scheduling with resource constraints*, in G. Mesnard, editor, *Proceedings of the International AMSE Conference: Modeling and Simulation*, 1986, 97–110.

22. A. Janiak, *On a single machine sequencing to minimize the maximum job cost subject to resource and precedence constraints*, Archiwum Automatyki i Telemetryki, t. 31, z. 4, (1986), 415–417.

23. A. Janiak, *One-machine scheduling problems with resiurce constraints*, Springer-Verlag, volume 84 of *Lecture Notes in Control and Information Science*, chapter System Modeling and Optimization, 1986.

24. A. Janiak, *Time-optimal control in a single machine problem with resource constrains*, Automatica, 22 (6), (1986), 745–747.

25. A. Janiak, *Minimization of the maximum tardiness in one-machine scheduling problem subject to precedence and resource constraints*, International Journal of System Analysis - Modeling - Simulation, 4 (6), (1987), 549–556.

26. A. Janiak, *One-machine scheduling with allocation of continuosly-divisable resource and with no precedence constraints*, Kybernetika, 23 (4), (1987), 289–293.

27. A. Janiak, *General flow-shop scheduling with resource constraints*, International Journal of Production Research, 26 (6), (1988), 1089–1103.

28. A. Janiak, *Minimization of the total resource consumption in permutation flow-shop sequencing subject to a given makespan*, Hournal of Modeling Simulation and Control (AMSE Press), 13 (2), (1988), 1–11.

29. A. Janiak, *Single machine sequencing with linear models of jobs subject to precedence constraints*, Archiwum Automatyki i Telemetryki, 33 (2), (1988), 203–210.

30. A. Janiak, *Minimalization of the total resource consumption under a given deadline in two-processor flow-shop scheduling problem*, Information Processing Letters, 32 (2), (1989), 101–112.

31. A. Janiak, *Minimization of the blooming mill standstills - mathematical model - suboptimals algorithms*, Mechanika AGH, 8 (2), (1989), 37–49.

32. A. Janiak, *Exact and approximate algorithms of job sequencing and resource allocation in discrete manufacturing pocesses (in polish)*, Monografie, Wydawnictwo Politechniki Wroclawskiej, Prace Naukowe Instytutu Cybernetyki Technicznej Politechniki Wroclawskiej edition, 1991.

33. A. Janiak, *Single machine scheduling problem with a common deadline and resource dependent release dates*, European Journal of Operational Research, 53 (3), (1991), 317–325.

34. A. Janiak, *Single machine scheduling problem with precedence and resource costraints (in polih)*, Zeszyty Naukowe Politechniki Slaskiej, seria Automatyka, z. 116, 31–47.

35. A. Janiak, *Analysis of computational complexity of single machine scheduling problems with release dates dependent on resources(in polish)*, Zeszyty Naukowe Politechniki Slaskiej, seria Automatyka, z. 117, 69–84.

36. A. Janiak, *Computational complexity analysis of single machine scheduling problems with job release dates dependent on resources*, in *Proceedings of the Symposium on Operations Research (SOR96), Braunschweig, September 3-6*, Springer-Verlag, 1997, 203–207.

37. A. Janiak, *Minimization of the makespan in a two-machine problem under given resource constraints*, European Journal of Operational Research, 107, (1998), 325–337.

38. A. Janiak, *Single machine sequencing with linear models of release dates*, Naval Research Logistics, 45, (1998), 99–113.

39. A. Janiak, *Selected problems and algorithms for task schedulig and Resource Allocation (in polish)*, Problemy Wspolczesnej Nauki, Teoria i Zastosowania - Informatyka, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1999.

40. A. Janiak, K. Chudzik, *Classical genetic approach to some flow type manufacturing problem*, in K. R. Domek S., Emirsajlow A., editor, *Proceedings of the Fourth International Symposium on Methods and Models in Automation and Robotics, Miedzyzdroje Poland, 26-29 August,*, 1997, vol. 3, pp. 1077–1082.

41. A. Janiak, J. Grabowski, *Optimization problems of sceduling with resource allocation in manufacturing processes (in russian)*, in *Proceedings of the VII Polish-Bulgarian Symposium, Warsaw*, 1980, 129–138.

42. A. Janiak, M. Kovalyov, *Single machine scheduling subject to deadlines and resource dependent processing times*, European Journal of Operational Research, 94, (1996), 284–291.

43. A. Janiak, M. Kovalyov, W. Kubiak, F. Werner, *Positive half-products and scheduling with controllable processing times*, European Journal of Operational Research, 165, (2005), 416–422.

44. A. Janiak, C.-L. Li, *Scheduling to minimize the total wighted completion time with a constraint on the release time resource consumption*, Mathematics of Computation Modelling, 20 (2), (1994), 53–58.

45. A. Janiak, M. Lichtenstein, *Optimal resource distribution in scheduling problems with resource dependent setup and processing times*, in *Proceedings of the 10th IEEE International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland, 30 August - 2 September, vol. 2*, 2004.

46. A. Janiak, M.-C. Portmann, *Genetic algorithm for the permutation flow-shop scheduling problem with linear models of operations*, Annals of Operation Research, 83, (1998), 95–114.

47. A. Janiak, T. Szkodny, *Job-shop scheduling with convex models of operations*, Mathematics of Computation Modelling, 20 (2), (1994), 59–68.

48. K. Jansen, M. Mastrolilli, R. Solis-Oba, *Approximation schemes for job shop scheduling problems with controllable processing times*, European Journal of Operational Research, 167, (2005), 297–319.

49. K. Jansena, M. Mastrolilli, *Approximation schemes for parallel machine scheduling problems with controllable processing times*, Computers & Operations Research, 31, (2004), 1565–1581.

50. S. Johnson, *Optimal two- and three-stage production schedules with setup times times included*, Naval Research Logistics Quarterly, 1, (1954), 61–68.

51. J. Jozefowska, J. Weglarz, *Discrete-continuous scheduling problems – mean completion time results*, European Journal of Operational Research, 94/2, (1996), 302–310.

52. J. Jozefowska, J. Weglarz, *On a methodology for discrete-continuous scheduling problems*, European Journal of Operational Research, 107/2, (1998), 338–353.

53. N. Karmarkar, *A new polynomial - time algorithm for linear programming*, Combinatorica, 4 (4), (1984), 373–395.

54. M. Kaspi, D. Shabtay, *A bicriterion approach to time/cost trade-offs in scheduling with convex resource-dependent job processing times and release dates*, Computers & Operations Research, 33, (2006), 3015–3033.

55. L. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Mathematical Doklady, 5 (20), (1979), 191–194.

56. M. Kovalyov, *Improving the cpmplexities of approximation algorithms for opimization problems*, Operations Research Letters, 17, (1995), 85–87.

57. C.-Y. Lee, L. Lei, *Multiple-project scheduling with controllable project duration and hard resource constraint: Some solvable cases*, Annals of Operations Research, 102, (2001), 287–307.

58. J. Lenstra, A. Rinnoy Kan, P. Brucker, *Complexity of machine cheduling problems*, Annals of Discrete Mathematics, 1, (1977), 343–362.

59. C.-L. Li, *Scheduling with resource-dependent release dates - a comparison of two different resource consumption functions*, Naval Research Logistics, 41, (1994), 807–819.

60. C.-L. Li, *Scheduling to minimize the total resource consumption with a constraint on the sum of completion times*, European Journal of Operational Research, 80, (1995), 381–388.

61. C.-L. Li, E. Sewell, T. Cheng, *Scheduling to minimize release-time resource consumption and tardiness penalties*, Naval Research Logistics, 42, (1995), 949–966.

62. M. Mastrolilli, *Notes on max flow time minimization with controllable processing times*, Computing, 71, (2003), 375–386.

63. J. Moore, *An n job, one machine sequencing algorithm for minimizing the number of late jobs*, Management Science, 15, (1968), 102–109.

64. E. Nowicki, *An approximation algorithm for the m-machine permutation flow shop scheduling problem with controllable processing times*, European Journal of Operational Research, 70, (1993), 342–349.

65. E. Nowicki, *An approximation algorithm for a single-machine scheduling problem with release times, delivery times and controllable processing times*, European Journal of Operational Research, 72, (1994), 74–81.

66. E. Nowicki, S. Zdrzalka, *A two machine flow shop scheduling problem with controllable job processing times*, European Journal of Operational Research, 34, (1988), 208–220.

67. E. Nowicki, S. Zdrzalka, *A survey of results for sequencing problems wwith controllable processing times*, Discrete Applied Mathematics, 26, (1990), 271–287.

68. E. Nowicki, S. Zdrzalka, *A bicriterion approach to preemptive scheduling of parallel machines with controllable job processing times*, Discrete Applied Mathematics, 63, (1995), 237–256.

69. S. Panwalkar, R. Rajagopalan, *Single-machine sequencing with controllable processing times*, European Journal of Operational Research, 59, (1992), 298–302.

70. D. Shabtay, *Single and a two-resource allocation algorithms for minimizing the maximal lateness in a single machine-scheduling problem*, Computers and Operations Research, 31 (8), (2004), 1303–1315.

71. D. Shabtay, M. Kaspi, *Minimizing the total weighted flow time in a single machine with controllable processing times*, Computers and Operations Research, 31 (13), (2004), 2279–2289.

72. D. Shmoys, E. Tardos, *An approximation algorithm for the generalized assignment problem*, Mathematical Programming, 62, (1993), 461–474.

73. W. Smith, *Various optimizers for single-stage production*, Naval Researcg Logistics Quartery, 3, (1956), 59–66.

74. M. Trick, *Scheduling multiple variable-speed machines*, Operations Research, 42 (2), (1994), 234–248.

75. A. Tuzikow, *O dwuchkriterjalnej zadace teorii raspisanij z ucatom izmienienija dlitelnostej obsluzivanija*, Zurnal Vycislitelnoj Matematyki i Matematiceskoj Fizyki, 10, (1984), 1585–1590.

76. S. Vasiliev, B. Foote, *On minimizing resource consumption with constraints on the makespan and the total completion time*, Europenan Journal of Operational Research, 96, (1997), 612–621.

77. R. Vickson, *Choosing the job sequence and processing times to minimize total processing plus flow cost of a single machine*, Operations Research, 28 (5), (1980), 1155–1167.

78. R. Vickson, *Two single machine sequencing problems involving controllable job processing times*, AIIE Transactions, 12 (3), (1980), 258–262.

79. L. V. Wassenhove, K. Baker, *A bicriterion approach to time/cost trade-offs in sequencing*, European Journal of Operational Research, 11, (1982), 48–54.

80. J. Weglarz, *Multiprocessor scheduling with memory allocation – a deterministic approach*, IEEE Trans. Computers, C-29/8, (1980), 703–710.

81. S. Zdrzalka, *Scheduling jobs on a single machine with release dates, due dates and controllable processing times: worst-case analysis*, Operations Research Letters, 10, (1991), 519–523.

82. S. Zdrzalka, *An approximation algorithm for a single-machine scheduling prolem with release times, delivery times and controllable processing times*, European Journal of Operational Research, 72, (1994), 74–81.

83. F. Zhang, G. Tang, Z.-L. Chen, *A 3/2-approximation algorithm for parallel machine scheduling with controllable processing times*, Operations Research Letters, 29, (2001), 41–47.