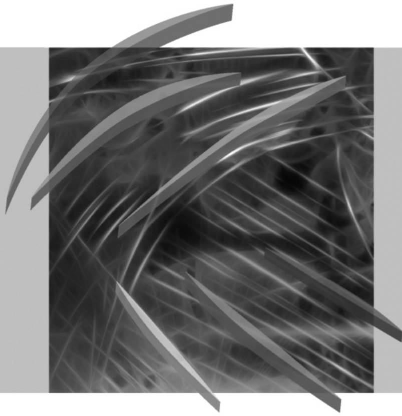
**PRACE NAUKOWE**
Uniwersytetu Ekonomicznego we Wrocławiu
**RESEARCH PAPERS**
of Wrocław University of Economics

**206**

# Advanced Information Technologies for Management – AITM 2011

## Intelligent Technologies and Applications



edited by
**Jerzy Korczak, Helena Dudycz, Mirosław Dyczkowski**

Publishing House of Wrocław University of Economics
Wrocław 2011

Reviewers: Frederic Andres, Witold Chmielarz, Jacek Cypryjański, Beata Czarnacka-Chrobot,
Bernard F. Kubiak, Halina Kwaśnicka, Antoni Ligęza, Anna Ławrynowicz,
Mikołaj Morzy, Stanisław Stanek, Ewa Ziemba

Copy-editing: Agnieszka Flasińska

Layout: Barbara Łopusiewicz

Proof-reading: Marcin Orszulak

Typesetting: Adam Dębski

Cover design: Beata Dębska

This publication is available at www.ibuk.pl

Abstracts of published papers are available in the international database
The Central European Journal of Social Sciences and Humanities http://cejsh.icm.edu.pl
and in The Central and Eastern European Online Library www.ceeol.com

Information on submitting and reviewing papers is available on the Publishing House's website
www.wydawnictwo.ue.wroc.pl



**ISSN 1899-3192**
**ISBN 978-83-7695-182-9**

The original version: printed
Printing: Printing House TOTEM

# Contents

**Preface** ........................................................................................ 9

**Witold Abramowicz, Jakub Dzikowski, Agata Filipowska, Monika Kaczmarek, Szymon Łazaruk,** Towards the Semantic Web's application for preparation of reviews – requirements and architecture for the needs of incentive-based semantic content creation ........................................ 11

**Frederic Andres, Rajkumar Kannan,** Collective intelligence in financial knowledge management, Challenges in the information explosion era .... 22

**Edyta Brzychczy, Karol Tajduś,** Designing a knowledge base for an advisory system supporting mining works planning in hard coal mines .. 34

**Helena Dudycz,** Research on usability of visualization in searching economic information in topic maps based application for return on investment indicator ........................................................................................ 45

**Dorota Dżega, Wiesław Pietruszkiewicz,** AI-supported management of distributed processes: An investigation of learning process ................. 59

**Krzysztof Kania,** Knowledge-based system for business-ICT alignment..... 68

**Agnieszka Konys,** Ontologies supporting the process of selection and evaluation of COTS software components ............................................. 81

**Jerzy Leyk,** Frame technology applied in the domain of IT processes job control ........................................................................................ 96

**Anna Ławrynowicz,** Planning and scheduling in industrial cluster with combination of expert system and genetic algorithm............................... 108

**Krzysztof Michalak, Jerzy Korczak,** Evolutionary graph mining in suspicious transaction detection ........................................................... 120

**Celina M. Olszak, Ewa Ziemba,** The determinants of knowledge-based economy development – the fundamental assumptions ............................ 130

**Mieczysław L. Owoc, Paweł Weichbroth,** A framework for Web Usage Mining based on Multi-Agent and Expert System An application to Web Server log files........................................................................................ 139

**Kazimierz Perechuda, Elżbieta Nawrocka, Wojciech Idzikowski,** E-organizer as the modern dedicated coaching tool supporting knowledge diffusion in the beauty services sector ..................................................... 152

**Witold Rekuć, Leopold Szczurowski,** A case for using patterns to identify business processes in a company............................................................ 164

**Radosław Rudek,** Single-processor scheduling problems with both learning and aging effects ........................................................................ 173

**Jadwiga Sobieska-Karpińska, Marcin Hernes,** Multiattribute functional dependencies in Decision Support Systems ............................................. 183

**Zbigniew Twardowski, Jolanta Wartini-Twardowska, Stanisław Stanek,** A Decision Support System based on the DDMCC paradigm for strategic management of capital groups ........ 192
**Ewa Ziemba, Celina M. Olszak,** The determinants of knowledge-based economy development – ICT use in the Silesian enterprises ........ 204
**Paweł Ziemba, Mateusz Piwowarski,** Feature selection methods in data mining techniques ........ 213

## Streszczenia

**Witold Abramowicz, Jakub Dzikowski, Agata Filipowska, Monika Kaczmarek, Szymon Łazaruk,** Wykorzystanie mechanizmów sieci semantycznej do przygotowania i publikacji recenzji – wymagania i architektura aplikacji ........ 21
**Frederic Andres, Rajkumar Kannan,** Inteligencja społeczności w finansowych systemach zarządzania wiedzą: wyzwania w dobie eksplozji informacji........ 33
**Edyta Brzychczy, Karol Tajduś,** Projektowanie bazy wiedzy na potrzeby systemu doradczego wspomagającego planowanie robót górniczych w kopalniach węgla kamiennego ........ 44
**Helena Dudycz,** Badanie użyteczności wizualizacji w wyszukiwaniu informacji ekonomicznej w aplikacji mapy pojęć do analizy wskaźnika zwrotu z inwestycji ........ 56
**Dorota Dżega, Wiesław Pietruszkiewicz,** Wsparcie zarządzania procesami rozproszonymi sztuczną inteligencją: analiza procesu zdalnego nauczania ........ 67
**Krzysztof Kania,** Oparty na wiedzy system dopasowania biznes-IT ........ 80
**Agnieszka Konys,** Ontologie wspomagające proces doboru i oceny składników oprogramowania COTS ........ 95
**Jerzy Leyk,** Technologia ramek zastosowana do sterowania procesami wykonawczymi IT ........ 107
**Anna Ławrynowicz,** Planowanie i harmonogramowanie w klastrze przemysłowym z kombinacją systemu eksperckiego i algorytmu genetycznego .. 119
**Krzysztof Michalak, Jerzy Korczak,** Ewolucyjne drążenie grafów w wykrywaniu podejrzanych transakcji........ 129
**Celina M. Olszak, Ewa Ziemba,** Determinanty rozwoju gospodarki opartej na wiedzy – podstawowe założenia........ 138
**Mieczysław L. Owoc, Paweł Weichbroth,** Architektura wieloagentowego systemu ekspertowego w analizie użytkowania zasobów internetowych: zastosowanie do plików loga serwera WWW ........ 151

**Kazimierz Perechuda, Elżbieta Nawrocka, Wojciech Idzikowski,** E-organizer jako nowoczesne narzędzie coachingu dedykowanego wspierającego dyfuzję wiedzy w sektorze usług kosmetycznych ...................... 163
**Witold Rekuć, Leopold Szczurowski,** Przypadek zastosowania wzorców do identyfikacji procesów biznesowych w przedsiębiorstwie .................. 172
**Radosław Rudek,** Jednoprocesorowe problemy harmonogramowania z efektem uczenia i zużycia ..................................................................... 181
**Jadwiga Sobieska-Karpińska, Marcin Hernes,** Wieloatrybutowe zależności funkcyjne w systemach wspomagania decyzji .................................... 191
**Zbigniew Twardowski, Jolanta Wartini-Twardowska, Stanisław Stanek,** System wspomagania decyzji oparty na paradygmacie DDMCC dla strategicznego zarządzania grupami kapitałowymi ....................................... 203
**Ewa Ziemba, Celina M. Olszak,** Determinanty rozwoju gospodarki opartej na wiedzy – wykorzystanie ICT w śląskich przedsiębiorstwach ............... 212
**Paweł Ziemba, Mateusz Piwowarski,** Metody selekcji cech w technikach *data mining* ......................................................................................... 223

**PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU nr 206**
**RESEARCH PAPERS OF WROCŁAW UNIVERSITY OF ECONOMICS**

Advanced Information Technologies for Management – AITM 2011 ISSN 1899-3192
Intelligent Technologies and Applications

**Radosław Rudek***
Wrocław University of Economics, Wrocław, Poland

# SINGLE-PROCESSOR SCHEDULING PROBLEMS WITH BOTH LEARNING AND AGING EFFECTS

**Abstract:** In this paper, we analyse single processor scheduling problems with both learning and aging effects to minimize the following criteria: the maximum completion time with release dates, the maximum lateness and the number of late jobs. The learning effect can be understood as a process of an acquiring experience that leads to increasing efficiency of a processor, which results in decreasing processing times of jobs. The opposite phenomenon called the aging effect decreases the efficiency of a processor. The measurable result of this effect is that the time required to process a single item decreases as more units are processed. We prove properties of the investigated problems and on their basis we provide optimal polynomial time algorithms for some cases.

**Keywords:** scheduling, learning effect, aging effect, polynomial-time algorithm.

## 1. Introduction

In this paper, we analyse scheduling problems, where processing times of jobs (e.g., products) vary due to learning and aging effects of a processor. Namely, the learning effect can be understood as a process of an acquiring experience that leads to increasing efficiency of a processor (e.g., a human worker, an algorithm, etc.) that results in decreasing processing times of jobs. The opposite phenomenon called the aging effect decreases the efficiency of a processor (e.g., a human worker, a chemical cleaning bath, drills or knives of a lath machine). The measurable result of this effect is that the time required to process a single item decreases as more units are processed.

The learning effect has a significant impact on productivity in manufacturing systems specialized in Hi-Tech electronic equipment [Adler, Clark 1991], memory chips and circuit boards [Webb 1994], electronic guidance systems [Kerzner 1998] and in many others (e.g. [Carlson, Rowe 1976; Cochran 1960; Holzer, Riahi-Belkaoui 1986; Jaber, Bonney 1999; Lien, Rasch 2001; Wright 1936; Yelle 1979]). Therefore, it is not surprising that scheduling problems with the learning effect have at-

* e-mail: radoslaw.rudek@ue.wroc.pl.

tracted particular attention in research society. For a survey see [Biskup 2008] and [Janiak, Rudek 2009].

On the other hand, the aging effect also can be found in many manufacturing and industrial systems in which, for instance, tiredness of human workers (e.g. [Dababneh, Swanson, Shell 2001; Eilon 1964]), decreasing concentration of active chemical cleaning substances [Mandich 2003] or tool wear of lathe machines (e.g. [Stanford, Lister 2004]) affects the production output. Therefore, it is worth taking this phenomenon into consideration during production planning and scheduling (see [Janiak, Rudek 2010; Rudek, Rudek 2012]).

Nevertheless, the learning and aging effects were usually analysed separately. There are only few papers where it is pointed out that the proposed algorithms can be used for problems with the learning or aging effects or both of them (e.g., [Mosheiov Sarig 2008; Rudek 2011]). Note that for instance the learning effect in case of human workers is long lasting process, whereas the aging effect is related with shifts. Therefore, taking into consideration both learning and aging effects during processing repetitive tasks is weakly justified. However, in this paper, we focus on problems where human worker is learning, where the machine, which is operated by a human worker, is deteriorating (aging). Therefore, the worker is learning and the processing time of operating the machine decreases, however, the machine deteriorates (drilling or lathe machine) that causes increasing the job processing times. In this case both phenomena are long lasting.

For the practical reasons production scheduling usually is focused on the minimization of the following time-objectives: the maximum completion time, the maximum lateness, the sum of the job completion times (the total completion time) and the number of late jobs. Therefore, in this paper, we analyse single processor scheduling problems with both learning and aging effects to minimize the presented optimization criteria.

The remainder of this paper is organized as follows. The next section contains problem formulation. In Section 3 properties of the analysed problems and resulting optimal polynomial time solution algorithms for some of their special cases are presented. The last section concludes the paper.

## 2. Problem formulation

There are given a single processor (e.g., a human worker and a machine) and $J = \{1, ..., j, ..., n\}$ denote the set of $n$ jobs (e.g., raw materials, semi-finished products) that have to be processed by the processor (e.g., to machine or to assemble a final product). By the practical reason it is assumed the processor performs jobs without preemptions, otherwise for instance a renewed calibration of a lathe machine is required or even a semi-finished product can be damaged. Furthermore, there are no precedence constraints between jobs, e.g., the order of machining raw materials on a lathe or processing semi-finished product can be performed in an arbitrary order.

Each job $j$ is available for processing at its release date $r_j$ and has to be processed before its due date $d_j$. Due to the learning and aging effects the processing time of job $j$ processed as the $v$-th in a schedule is defined as follows:

$$p_j(v)=p_j+a_jv-b_jv, \tag{1}$$

where $p_j$ is the normal processing time of job $j$ that is defined as its processing time if the processor is not affected by learning nor aging, $a_j$ is the aging ratio that describes increasing the job processing time, whereas $b_j$ is the learning ratio that describes decreasing of the job processing time. Note that $p_j + a_jv - b_jv > 0$ for $j = 1, ..., n$ and $v = 1, ..., n$, since the processing time cannot be negative.

Let $\pi = \langle\pi(1), \pi(2), ..., \pi(i), ..., \pi(n)\rangle$ denote the sequence/schedule of jobs (i.e., permutation of the elements of the set $J$), where $\pi(i)$ is the index of a job processed in position $i$ in this sequence. By $\Pi$ we will denote the set of all such permutations. For the given sequence (permutation) $\pi \in \Pi$, we can easily determine the completion time $C_{\pi(i)}$ of a job placed in the $i$-th position in $\pi$ from the following formulae:

$$C_{\pi(i)}= \max\{r_{\pi(i)}, C_{\pi(i-1)}\}+p_{\pi(i)}(i) = \max\{r_{\pi(i)}, C_{\pi(i-1)}\}+ p_{\pi(i)}+ a_{\pi(i)}i - b_{\pi(i)}i; \tag{2}$$

where $C_{\pi(0)} = 0$ and a lateness of this job:

$$L_{\pi(i)} = C_{\pi(i)} - d_{\pi(i)}. \tag{3}$$

On this basis, we formulate the minimization objectives that are functions of the schedule $\pi$, i.e., the maximum completion time $C_{\max}(\pi)$, the total completion time $TCT(\pi)$, the maximum lateness $L_{\max}(\pi)$, the number of late jobs $\sum U_j(\pi)$:

$$C_{\max}(\pi) = C_{\pi(n)}, \tag{4}$$

$$TCT(\pi) = \sum_{i=1}^{n} C_{\pi(i)}, \tag{5}$$

$$L_{\max}(\pi) = \max_{i=1,...,n} \{L_{\pi(i)}\}, \tag{6}$$

$$\Sigma U_j(\pi) = \sum_{i=1}^{n} U_{\pi(i)}, \tag{7}$$

where $U_{\pi(i)} = \begin{cases} 0, & C_{\pi(i)} \le d_{\pi(i)} \\ 1, & C_{\pi(i)} > d_{\pi(i)} \end{cases}$.

The objective is to find such a schedule (i.e., sequence) $\pi$ of jobs performed by the processor that minimizes one of the following (4)–(7). Formally the optimal schedule $\pi^* \in \Pi$ for each of the considered minimization objectives is defined as follows:

$$\pi^* = \arg\min_{\pi \in \Pi} \{C_{\pi(n)}\}. \tag{8}$$

$$\pi^* = \arg\min_{\pi \in \Pi} \{TCT(\pi)\}. \tag{9}$$

$$\pi^* = \arg\min_{\pi \in \Pi} \{L_{\max}(\pi)\}. \tag{10}$$

$$\pi^* = \arg\min_{\pi \in \Pi} \{\Sigma U_j(\pi)\}. \tag{11}$$

According to the three field notation scheme the analysed problems will be denoted as follows: $1|r_j, p_j(v) = p_j + a_j v - b_j v|C$, where $C \in \{C_{\max}, TCT, L_{\max}, \Sigma U_j\}$ depending on the criterion objectives. If $r_j = 0$, then it is omitted in the used notation. Furthermore, if $r_j < r_k$ or $d_j < d_k$ implies $(a_j - b_j) < (a_k - b_k)$, then we say that job parameters are agreeable and it will be denoted by *agr* in the three field notation scheme.

## 3. Properties

In this section, we provide properties of the analysed problems together with the following optimal polynomial time algorithms. The results are based on the observations of properties of the problems with the learning effect or with the aging effect only, which were presented in published papers. In the further part of this paper, for convenience of mathematical considerations, we will use the following term $\Delta_j = a_j - b_j$ for $j = 1, ..., n$.

**Property 1.** The problem $1 | p_j(v) = p_j + a_j v - b_j v | C_{\max}$ can be optimally solved in time $O(n\log n)$ by scheduling jobs according to the non-increasing order of $\Delta_j = a_j - b_j$.
**Proof.** The proof can be done using job interchanging technique. □

Note that the problem $1 | p_j(v) | C_{\max}$, where $p_j(v)$ is an arbitrary positive function that describes the processing time of job $j$, can be solved optimally in time $O(n^3)$ by expressing it as the assignment problem (see [Bachman, Janiak 2004]).

**Property 2.** The problem $1 | p_j(v) = p_j + a_j v - b_j v | TCT$ can be optimally solved in time $O(n\log n)$ by scheduling jobs according to the non-increasing order of $\Delta_j = a_j - b_j$.
**Proof.** The proof can be done using job interchanging technique. □

Note that the problems $1 | p_j(v) | C_{\max}$ and $1 | p_j(v) | TCT$, where $p_j(v)$ is an arbitrary positive function that describes the processing time of job $j$, can be solved optimally in time $O(n^3)$ by expressing them as the assignment problem (see [Bachman, Janiak 2004]).

Further, we will analyse the makespan minimization problem, where jobs can have distinct release dates $r_j$. Thus, it has to be decided: process the current job or wait for the next one and process the current later. This simple problem becomes strongly NP-hard if the processing times of jobs are not constant values, but vary due to learning or aging.

**Corollary 1.** The problem $1 | r_j, p_j(v) = p_j + a_j v - b_j v | C_{\max}$ is strongly NP--hard.
**Proof.** It follows from the NP-hardness of the problem $1 | r_j, p_j(v) = p_j - b_j v | C_{\max}$ (see [Bachman, Janiak 2004]). □

Note that Bachman and Janiak [2004] claimed that they proved also strong NP-hardness of the problem $1 | r_j, p_j(v) = p_j v^a \; | C_{\max}$ (where $\alpha$ is the exponential learning ratio common for all jobs). However, they proved only NP-hardness (not strong NP-hardness) since the transformation applied in the proof was polynomial not pseudopolynomial (as [Bachman, Janiak 2004] erroneously thought). This serious misconception was proved by the author of this paper in [Rudek 2012]. However, this problem becomes polynomially solvable under special conditions.

**Property 3.** The problem $1 | r_j, p_j(v) = p_j + av - bv | C_{\max}$ can be optimally solved in time $O(n\log n)$ by scheduling jobs according to the non-decreasing order of release dates $r_j$.
**Proof.** The proof can be done using job interchanging technique. □

**Property 4.** The problem $1 | agr, r_j, p_j(v) = p_j + a_j v - b_j v | C_{\max}$ can be optimally solved in time $O(n\log n)$ by scheduling jobs according to the non-increasing order of $(a_j - b_j)$ or according to the non-decreasing order of release dates $r_j$.
**Proof.** The proof can be done using job interchanging technique. □

Next, we will focus on the problem, where jobs have due-dates $d_j$ (when they have to be completed) and the objective is to minimize the maximum lateness. This polynomially solvable problem is strongly NP-hard if learning or aging effects are taken into consideration.

**Corollary 2.** The problem $1 | p_j(v) = p_j + a_j v - b_j v | L_{\max}$ is strongly NP-hard.
**Proof.** It follows from the NP-hardness of the problem $1 | r_j, p_j(v) = p_j + a_j v | L_{\max}$ (see [Bachman, Janiak 2004]). □

Nevertheless, we will show the special cases of this problem are still polynomially solvable.

**Property 5.** The problem can be optimally solved in time $O(n\log n)$ by scheduling jobs according to the non-decreasing order of release dates $d_j$.
**Proof.** Although the proof is simple and is based on the well known job interchanging technique, in this case we provide it formally. Assume that there is given an optimal permutation $\pi$ that does not comply with the thesis of this property. Therefore,

for this permutation there exists a pair of jobs $\pi(i)$ and $\pi(i+1)$, where $r_{\pi(i)} > r_{\pi(i+1)}$. Assume there is given a permutation $\pi'$, which has been obtained from $\pi$ by interchanging jobs from positions $i$ and $i+1$. Observe that completion times and lateness of jobs scheduled in positions 1, ..., $i-1$ and $i+2$, ..., $n$ are the same in both permutations. Therefore, we have to analyse lateness of jobs scheduled in positions $i$ and $i+1$ in both permutations:

$$L_{\pi(i)} = C_{\pi(i-1)} + p_{\pi(i)} + (a-b)i - d_{\pi(i)},$$

$$L_{\pi(i+1)} = C_{\pi(i-1)} + p_{\pi(i)} + (a-b)i + p_{\pi(i+1)} + (a-b)(i+1) - d_{\pi(i+1)},$$

$$L_{\pi'(i)} = C_{\pi(i-1)} + p_{\pi(i+1)} + (a-b)i - d_{\pi(i+1)},$$

$$L_{\pi'(i+1)} = C_{\pi(i-1)} + p_{\pi(i)} + (a-b)i + p_{\pi(i+1)} + (a-b)(i+1) - d_{\pi(i)}.$$

Since $d_{\pi(i)} > d_{\pi(i+1)}$, then $L_{\pi(i)} > L_{\pi(i+1)}$. Note also that $L_{\pi(i+1)} > \max\{L_{\pi(i)}, L_{\pi(i+1)}\}$, therefore, the permutation $\pi$ cannot be optimal and the non-decreasing order of $d_{\mathrm{j}}$ gives the optimal solution to the considered problem. □

**Property 6.** The problem $1 | agr, p_j(v) = p_j + a_j v - b_j v | L_{\max}$ can be optimally solved in time $O(n\log n)$ by scheduling jobs according to the non-increasing order of $(a_j - b_j)$ or according to the non-decreasing order of release dates $d_j$.
**Proof.** The proof can be done using job interchanging technique in the similar manner as the proof to the previous property. □

**Corollary 3.** The problem $1 | p_j(v) = p_j + a_j v - b_j v | \Sigma U_j$ is strongly NP-hard.
**Proof.** It follows from the NP-hardness of the problem $1 | r_j, p_j(v) = p_j + a_j v - b_j v | C_{\max}$ (see Property 6). □

The single processor scheduling problem with the minimization of the number of late jobs, $1||\Sigma U_j$, can be solved optimally by the well known Moore's algorithm [Moore 1968]. Here, we will prove that this algorithm is still optimal for the considered problem $1 | r_j, p_j(v) = p_j + av - bv | \Sigma U_j$, i.e., if jobs similarly deteriorate a machine ($a_j = a$) and they have similar impact on learning of a human worker ($b_j = b$).

**Moore's Algorithm**

STEP 1: Schedule the jobs in non-decreasing order of their due dates (EDD)
STEP 2: If no jobs in the sequence are late go to STEP 7
STEP 3: Find the first late job, denote this job by $\alpha$
STEP 4: Find a job $\beta$ such that $p_\beta = \max_{i=1,\ldots,\alpha}\{p_i\}$
STEP 5: Remove $\beta$ from the schedule and process it after all the jobs that are not late have been processed
STEP 6: Go to STEP 2
STEP 7: The schedule is optimal

**Property 7.** The problem $1|p_j(v)=p_j+av-bv|\Sigma U_j$ can be optimally solved in time $O(n\log n)$ by by Moore's algorithm.

**Proof.** The proof will be done using the inductive method in the similar manner as in [Sturm 1970]. Based on Property 2, we can note that there exists a schedule for $1|p_j(v)=p_j+av-bv|\Sigma U_j$ having no late jobs if and only if the schedule of jobs according to the non-decreasing order of their due dates (EDD) has no late jobs. On this basis, we will consider only EDD sequences. To simplify the proof, assume that such a sequence is 1, 2, ..., $n$ (if it is not a case we can renumber the jobs).

Assume that using Moore's algorithm (MA) we determine a subset $B = \{\beta_1, ..., \beta_q\}$ of $q$ late jobs. Suppose also that it is possible to choose from the set $J = \{1, ..., n\}$ a subset $\Gamma = \{\gamma_1, ..., \gamma_{q-1}\}$ of $q-1$ jobs, such that the remaining $n-q+1$ jobs $J\backslash\Gamma$ are not late. Thus, for all $i = 1, ..., n$ the following inequality must hold:

$$d_i \ge \sum_{j=1}^{i} p_j + (a-b)j - \sum_{\gamma_j \in \Gamma_i} p_{\gamma_j} - \sum_{j=1}^{|\Gamma_i|}(a-b)(i-j+1), \tag{13}$$

where $\Gamma_i = \{\gamma_j : \gamma_j \le i, \gamma_j \in \Gamma\}$ and $|\Gamma_i|$ is the cardinality of $\Gamma_i$. Without loss of generality we can also assume $\forall(i,j)\ \beta_i \ne \gamma_j$.

Using MA, we find the first late job $\alpha_1$, i.e., that satisfies $d_{\alpha_1} < \sum_{j=1}^{\alpha_1} p_j + (a-b)j$ and $d_i < \sum_{j=1}^{i} p_j + (a-b)j$ for $i = 1, ..., \alpha_1 - 1$. From the definition of $\Gamma$ follows that there is at least one job $\gamma_j \in \Gamma_{\alpha 1}$, i.e., inequality (13) must hold. Let us choose an element $\delta_1$ from $\Gamma_{\alpha 1}$ with $p_{\delta 1} = \max\{p_{\gamma i} : \gamma_i \in \Gamma_{\alpha 1}\}$. On the other hand, MA chooses job $\beta_1$ that satisfies:

$$d_{\alpha_1} \ge d_{\alpha_1 - 1} \ge \sum_{j=1}^{\alpha_1 - 1} p_j + (a-b)j = \sum_{j=1}^{\alpha_1} p_j + (a-b)j - p_{\alpha_1}$$
$$-(a-b)\alpha_1 \ge \sum_{j=1}^{\alpha_1} p_j + (a-b)j - p_{\beta_1} - (a-b)\alpha_1.$$

Observe that if $\alpha_1 \ne \beta_1$ then job $\alpha_1$ is no longer late, since job $\beta_1$ is skipped. It is easy to notice that $a_{\delta 1} \le a_{\beta 1}$. Thus, there must be at least one job in $\Gamma_{\alpha 1}$ and $\delta_1 \in \Gamma_{\alpha 1}$.

Suppose now that there is at least $l$ ($l < q$) jobs in $\Gamma_{\alpha l}$ and we are able to choose among them $l$ elements $\delta_i$ such that $a_{\delta i} \le a_{\beta i}$ for $i = 1, ..., l$.

Using MA we find job $\alpha_{l+1}$ (i.e., the first late job after $l$ jobs are skipped) that satisfies:

$$d_{\alpha_{l+1}} < \sum_{j=1}^{\alpha_{l+1}} p_j + (a-b)j - \sum_{j=1}^{l} p_{\beta_j} - \sum_{j=1}^{l}(a-b)(\alpha_{l+1} - j + 1)$$
$$\le \sum_{j=1}^{\alpha_{l+1}} p_j + (a-b)j - \sum_{j=1}^{l} p_{\delta_j} - \sum_{j=1}^{l}(a-b)(\alpha_{l+1} - j + 1).$$

From (13) follows that there must be at least $l+1$ jobs in $\Gamma_{\alpha(l+1)}$ to satisfy:

$$d_{\alpha_{l+1}} \geq \sum_{j=1}^{\alpha_{l+1}} p_j + (a-b)j - \sum_{\gamma_j \in \Gamma_{\alpha_{l+1}}} p_{\gamma_j} - \sum_{j=1}^{|\Gamma_{\alpha_{l+1}}|} (a-b)(\alpha_{l+1} - j + 1).$$

and $\delta_i \in \Gamma_{\alpha(l+1)}$, $i = 1, ..., l$. Thus, we find the $(l+1)$-th element with $p_{\delta(l+1)} = \max\{p_{\gamma i}: \gamma_i \in \Gamma_{\alpha(l+1)} \backslash \{\delta_i\}, j = 1, ..., l\}$. On the other hand, MA finds $\beta_{l+1}$ (i.e., the $(l+1)$-th late job) with $a_{\beta(l+1)} = \{a_{\beta i}: i = 1, ..., \alpha_{(l+1)}, i \neq \beta_1, ..., \beta_l\}$ and it is easy to notice that $a_{\delta(l+1)} \leq a_{\beta(l+1)}$. Therefore, there must be at least $l+1$ jobs in $\Gamma_{\alpha(l+1)}$ and among them $l+1$ jobs $\delta_i$ with $p_{\delta i} \leq p_{\beta i}$ for $i = 1, ..., l+1$. Concluding in the same way, we can show that when MA finds job $\alpha_q$ then there must be at least $q$ jobs in $\Gamma_{\alpha q}$ and it contradicts the assumption $|\Gamma| = q - 1$. Thus MA finds the minimum number of late jobs for the considered scheduling problem. □

## 4. Conclusions

In this paper, we analysed single processor scheduling problems with both learning and aging effects to minimize the following criteria: the maximum completion time with release dates, the maximum lateness and the number of late jobs. We proved properties of the investigated problems and on their basis, we provided optimal polynomial time algorithms for some of their special cases. The applied methodology to prove the properties is intuitive and it is similar as for the problems with one effect only, i.e., the learning effect or the aging effect.

Our future work will focus on the construction of efficient exact and approximation algorithms (heuristics that will be based on the properties provided in this paper and metaheuristics) for the NP-hard cases.

## References

Adler P.S., Clark K.B. (1991), Behind the learning curve: A sketch of the learning process, *Management Science*, Vol. 37, pp. 267–281.

Bachman A., Janiak A. (2004), Scheduling jobs with position dependent processing times, *Journal of the Operational Research Society*, Vol. 55, pp. 257–264.

Biskup D. (2008), A state-of-the-art review on scheduling with learning effects, *European Journal of Operational Research*, Vol. 188, pp. 315–329.

Carlson J.G., Rowe R.G. (1976), How much does forgetting cost?, *Industrial Engineering*, Vol. 8, pp. 40–47.

Cochran E.B. (1960), New concepts of the learning curve, *The Journal of Industrial Engineering*, Vol. 11, pp. 317–327.

Dababneh A.J., Swanson N., Shell R.L. (2001), Impact of added rest breaks on the productivity and well being of workers, *Ergonomics*, Vol. 44, pp. 164–174.

Eilon S. (1964), On a mechanistic approach to fatigue and rest periods, *International Journal of Production Research*, Vol. 3, pp. 327–332.
Holzer H.P., Riahi-Belkaoui A. (1986), *The Learning Curve: A Management Accounting Tool*, Quorum Books, Westport, CT.
Jaber Y.M., Bonney M. (1999), The economic manufacture/order quantity (EMQ/EOQ) and the learning curve: Past, present, and future, *International Journal of Production Economics*, Vol. 59, pp. 93–102.
Janiak A., Rudek R. (2009), Experience based approach to scheduling problems with the learning effect, *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, Vol. 39, pp. 344–357.
Janiak A., Rudek R. (2010), Scheduling jobs under an aging effect, *Journal of the Operational Research Society*, Vol. 61, pp. 1041–1048.
Kerzner H. (1998), *Project Management: A System Approach to Planning, Scheduling, and Controlling*, John Wiley & Sons New York.
Lien T.K., Rasch F.O. (2001), Hybrid automatic-manual assembly systems, *Annals of the CIRP*, Vol. 50, pp. 21–24.
Mandich N.V. (2003), Overview of surface preparation of metals prior to finishing: Part 2, *Metal Finishing*, Vol. 101, pp. 33–58.
Moore J.M. (1968), An *n* jobs, one machine sequencing algorithm for minimizing the number of late jobs, *Management Science*, Vol. 15, pp. 102–109.
Mosheiov G., Sarig A. (2008), A due-window assignment problem with position-dependent processing times, *Journal of the Operational Research Society*, Vol. 59, pp. 997–1003.
Rudek R. (2011), Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect, *Computers & Industrial Engineering*, Vol. 61, pp. 20–31.
Rudek A., Rudek R. (2012), A note on optimization in deteriorating systems using scheduling problems with the aging effect and resource allocation models, *Computers & Mathematics with Applications*, Vol. 218, pp. 6498–6510.
Rudek R. (2012), A note on proving the strong NP-hardness of a scheduling problem with position dependent job processing times, *Optimization Letters* [forthcoming].
Stanford M., Lister P.M. (2004), Investigation into the relationship between tool-wear and cutting environments when turning EN32 steel, *Industrial Lubrication and Tribology*, Vol. 56, pp. 114–121.
Sturm L.B.J.M. (1970), A simple optimality proof of Moore's sequencing algorithm, *Management Science*, Vol. 17, pp. 116–118.
Webb G.K. (1994) Integrated circuit (IC) pricing, *High Technology Management Research*, Vol. 5, pp. 247–260.
Wright T.P. (1936), Factors affecting the cost of airplanes, *Journal of Aeronautical Sciences*, Vol. 3, pp. 122–128.
Yelle L.E. (1979), The learning curve: Historical review and comprehensive study, *Decision Science*, Vol. 10, pp. 302–328.

## JEDNOPROCESOROWE PROBLEMY HARMONOGRAMOWANIA Z EFEKTEM UCZENIA I ZUŻYCIA

**Streszczenie:** W artykule analizowane są jednoprocesorowe problemy harmonogramowania z efektem uczenia i zużycia (starzenia) przy następujących kryteriach minimalizacji: długość uszeregowania zadań z terminami ich dostępności, suma czasów zakończenia wykonywania zadań, maksymalna nieterminowość zadań oraz liczba opóźnionych zadań. Efekt uczenia jest rozumiany jako proces nabywania doświadczenia przez procesor, który prowadzi do skrócenia czasów wykonywania kolejnych zadań. Natomiast efekt zużycia (starzenia) powoduje obniżenie efektywności procesora. Mierzalnym rezultatem jest wydłużenie czasów wykonywania zadań. W pracy wykazano szereg własności badanych problemów, które pozwalają na konstrukcję wielomianowych optymalnych algorytmów rozwiązania dla szczególnych przypadków tychże problemów.

**Słowa kluczowe:** harmonogramowanie, efekt uczenia, efekt starzenia, wielomianowe optymalne algorytmy.