Decision Making in Manufacturing and Services
Vol. 5 • 2011 • No. 1–2 • pp. 37–48

Scheduling Jobs with Linear Model of Simultaneous Ageing and Learning Effects

Adam Janiak[⋆], Maciej Lichtenstein[⋆†], Agata Rusoń[⋆]

Abstract. In the paper, we introduce some new scheduling model in which learning and aging effects are both considered simultaneously. In this model the actual processing time of the jobs depends only on its position in a schedule and can be described by the piecewise linear function. For single-processor problem with introduced model, we show that the problem of minimizing the makespan criterion for independent jobs with release dates is strongly NP-hard, but some special cases of this problem are polynomially solvable. Based on those special cases, we propose 4 heuristic algorithms and we experimentally examine their usefulness for solving the general problem.

Keywords: sequencing, single machine, learning effect, ageing effect, computational complexity

Mathematics Subject Classification: 90B35, 68Q17, 11Y16

Revised: 25 August 2011

1. INTRODUCTION

Classical scheduling models assume that the processing time of a job is a given, fixed value. However, in models of many real life systems we cannot make this assumption. Thus the processing time of a job (or other job parameters) is treated as a variable dependent on the schedule itself. The scheduling models that assume variable jobs processing times include among others: resource dependent processing times (Shabtay and Steiner, 2007; Janiak *et al.*, 2007), deteriorating environments (Cheng *et al.*, 2004; Bachman *et al.*, 2002), etc. Recently, in the scientific literature many papers regarding the learning effect (Biskup, 2008; Janiak and Śnieżyk, 2004), and the ageing effect (Yang and Yang, 2010; Janiak and Rudek, 2010) were published, which also assume that the value of processing time of a job is a variable dependent on the schedule.

† Corresponding author
⋆ Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Z. Janiszewskiego 11/17, 50 372 Wrocław, Poland. E-mail:{adam.janiak, maciej.lichtenstein, agata.ruson}@pwr.wroc.pl

In this paper we introduce some new scheduling model which takes into account both learning, and ageing effect simultaneously. In this model, the actual processing time of a job depends on its position in the schedule and is described by piecewise linear function. This function is dependent on the position of a job in the sequence of jobs and describes three phases of processor efficiency: the learning phase in which processing time of a job decreases, maturity phase in which processing time of a job is constant, and the ageing phase, in which job processing time increases.

On the basis of the introduced model we formulate the single processor scheduling problem with given job release dates and the makespan (the schedule length) minimization objective. We show that considered problem is in general strongly NP-hard, but we identify several special cases of this problem which can be polynomially solved. Based on these special cases we construct 4 heuristic algorithms for solving general case of the problem and verify their efficiency by experimental analysis.

The remainder of the paper is organized as follows. In the next section we shortly present state of research in the domain of scheduling problems with both learning, and ageing effect. For the complete and up-to-date presentation of the results in the area of scheduling with learning effect and ageing effect we refer the reader to Janiak *et al.* (2011). In section 3 we formulate precisely mentioned above model of job processing time and the considered makespan minimization problems. In Section 4 we present some properties of the problem and describe its special, polynomially solvable cases. Section 5 is devoted to the proposed heuristic algorithms together with experiments testing their efficiency. Finally, in Section 6 we conclude the paper pointing out some directions of future research in this subject.

2. LITERATURE REVIEW

The learning and ageing phenomenons were introduced into the scheduling area in the beginning of the XXI century and since then attracted many researchers all over the world. Both learning effect and ageing effect were initially considered separately and recently there are many attempts to combine both this effect into a single model. In what follows, we presents some basic and most important results available in the scheduling literature separately for the learning effect models, ageing effect models, and the models that combine learning and ageing effects. The reader interested in the deeper look into the results presented in the literature in this matter, is referred to the paper by Janiak *et al.* (2011).

The learning effect itself (not in the scheduling area) was firstly observed by Wright (1936) while performing research on the workers efficiency in airplanes factory. He proposed the exponential shape of the learning curve (the curve that describes the increase of worker efficiency over the number of tasks he performed). Another research introduced many other shapes of the learning curve such as the S-shaped curve by Jordan (1965) and Carlson and Rowe (1976). The detailed description and interpretation of the learning curves can be found in (Jaber and Bonney, 1999; Badiru, 1992; Dutton and Thomas, 1984).

The first results regarding the learning effect in scheduling area are due to Biskup (1999) and Cheng and Wang (2000). In (Biskup, 1999) author used the model introduced by Wright (1936), i.e. the exponential, non-increasing job processing time function dependent on its position in the schedule. The properties developed by Biskup where then used by Mosheiov in (Mosheiov, 2001b; Mosheiov and Sidney, 2003; Mosheiov, 2001a). Independently, Cheng and Wang (2000) the linear model of the learning curve was introduced. Future work with linear model of learning curve was conducted in (Bachman and Janiak, 2004). Beside model in which processing time of a job is dependent on its position in the sequence, the model with the processing time dependent on the time that passed since start of schedule to the beginning of job processing were introduced in (Kuo and Yang, 2006). Future work in this area include (Biskup, 2008; Yin *et al.*, 2009). Unfortunately, in the mentioned papers there a not convincing, reasonable real-life examples of application of such scheduling problems with learning effects.

The ageing effect is a very similar phenomenon to the ageing effect, however, the function that describes the job processing time is non-decreasing, i.e. the processing of a takes longer if it is processed later in the sequence. This phenomenon models the degeneration of the processor with the number of jobs it processed or over time. The exponential ageing effect in the scheduling area was introduced by Mosheiov (2001a). Then Janiak and Śnieżyk (2004) and Bachman and Janiak (2004) the linear model of ageing was introduced. More complex model were considered by Wang *et al.* (2009), Cheng *et al.* (2008), and Janiak and Śnieżyk (2005a). Similarly, to the learning effects the time dependent models were considered in (Janiak and Śnieżyk, 2005a; Gawiejnowicz and Kononov, 2010). Again, in all those cited papers the convincing practical application of considered scheduling problems is not presented which put in question the motivation of the research on the ageing effect in the scheduling area.

The model that include both learning and ageing effects simultaneously in a single model was introduced in by Lee (2004). He considered model in which processing time of a job is dependent on its position in the sequence (the learning effect), however, deteriorates with its starting time (ageing effect). The research on similar models (combining the learning effect and deteriorating jobs) were continued by Wang (2007), Yang and Kuo (2009), and Toksari *et al.* (2010). Another kind of model that combines the learning and ageing effects was introduced by Sun (2009). In this model the actual processing time of a job increases with its position in the sequence (the ageing effect), but decreases with the total processing time of jobs already processed (the learning effect). The research on this type of models was continued in (Cheng *et al.*, 2010; Wang and Liu, 2009).

In this section we briefly presented some basic results available in the scheduling literature that deals with the learning and aging phenomenons. The reader interested in more detailed survey is again referred to the paper by Janiak *et al.* (2011) which is complete and up-to-date.

The models available in the literature that combine the both learning and ageing effects in a single model are very non-realistic (even bizarre) and again lack the convincing real-life system of their application. Moreover, the interpretation of these models is somehow difficult since they are extensions of complex models of learning and ageing effects. Thus, in this paper we introduce a new model of simultaneous learning and ageing that is easy to interpret which can be viewed as a basic linear model and we hope to find for it use in scheduling problems some convincing real-life applications in computer engineering, or technical, or economical environment.

3. PROBLEM FORMULATION

Before we formulate the problem that is considered in the paper, first we introduce the learning-aging model described by the piecewise linear curve, which is depicted in Figure 1.

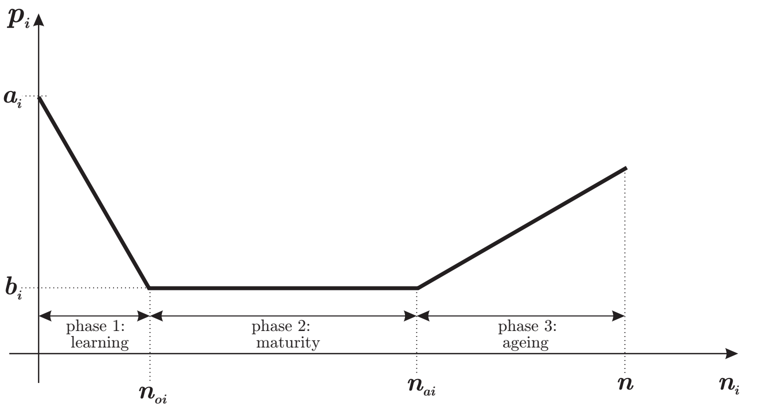

Fig. 1. *The learning–ageing curve.*

In the considered model, the processing time of a job (say i) which is executed by the machine on the position n_i in the sequence of n jobs is given by the following function:

$$p_i(n_i) = a_i - v_i \min\{n_i, n_{oi}\} + w_i \max\{0, n_i - n_{ai}\}, \tag{1}$$

where:

- $a_i > 0$ – the initial processing time of job i,
- $v_i \geqslant 0$ – the learning ratio of job i,
- $w_i \geqslant 0$ – the ageing ratio of job i,
- $n_{oi} \geqslant 1$ – the learning threshold – the number of the last position at which the learning effect can be observed; the end of the learning phase and the beginning of the maturity phase,
- $n_{ai} \leqslant n$ – the ageing threshold – the number of the first position at which the aging effect can be observed; the end of the maturity phase and the beginning of the aging phase.

In the sequence of n jobs every job can be scheduled at one of n positions, so the processing time can have n (potentially) different values. On the other hand, the processing time of a job can be viewed as a function of the number of jobs processed before this job, which is actually equal to the job position decremented by 1. Looking at the Figure 1, we have to remember that the domain of the function is discrete, not continuous, as the figure could suggest.

On the basis of introduced above model (1) we formulate the following scheduling problem. There is a given set $J = \{1, ..., n\}$ of n non-preemptive jobs to be processed on a single processor. Every job $i \in J$ is defined by its release date $r_i \geqslant 0$, i.e. the time at which the job is available for processing, and the parameters of its processing time $p_i(n_i)$ given in (1), i.e. values a_i, v_i, w_i, n_{oi}, n_{ai}. All the parameters have to take values which ensure that for every position n_i the processing time has a non-negative value. The objective of the problem is to find a sequence of jobs (i.e. the permutation of the set J) such that the schedule length (makespan) $C_{max} = \max_{i \in J}\{C_i\}$ is minimized, where C_i is the completion time of job i. For convenience, by S_i we denote the starting time of job i. It is clear that $C_i = S_i + p_i(n_i)$.

As mentioned, the sequence of job processing can be defined by a permutation $\pi = (\pi(1), ..., \pi(n))$ of the set J, where $\pi(k)$ denotes the job scheduled as kth in this sequence. For a given permutation π the completion time of job scheduled at position k can be calculated using the following formulae:

$$C_{\pi(k)} = \max\{C_{\pi(k-1)}, r_{\pi(k)}\} + p_{\pi(k)}(k) \tag{2}$$

for $k = 1, ..., n$, where $\pi(0) = 0$ and $C_0 = 0$.

In the following sections, the defined above problem will be denoted as PLA.

4. PROBLEM PROPERTIES

Bachman and Janiak (2004) showed that the similar problem with job processing time given as the linear function of job position $p_i(k) = a_i - b_i k$, which models the learning effect only, is NP-hard in the strong sense. Similarly Janiak and Śnieżyk (2005b) showed that the problem with processing times given as $p_i(k) = a_i + b_i k$ is also NP-hard in strong sense. From this facts, since both these problem are the special cases of PLA, it follow that PLA is also NP-hard in the strong sense. Thus, we can formulate the following corollary.

Corollary 1 *The problem PLA is NP-hard in strong sense.*

On the other hand, it can be showed that another special case of PLA denoted here as $PLA0$, in which $r_i = 0$ for all $i \in J$ can be solved in polynomial time.

Proposition 1 *The problem $PLA0$ can be solved in polynomial time $O(n^3)$.*

Proof. Let x_{ij} are binary variables that indicate that if $x_{ij} = 1$ then job i is scheduled at jth position and $x_{ij} = 0$ otherwise. In this situation we can formulate the problem $PLA0$ as follows.

Minimize:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(a_i - v_i \min\{j, n_{oi}\} + w_i \max\{0, j - n_{ai}\}x_{ij},$$

subject to:

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, ..., n,$$

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, ..., n,$$

$$x_{ij} \in \{0, 1\}.$$

As it can be seen the above formulation is an assignment problem instance which can be solved in $O(n^3)$ time (e.g. Papadimitrou and Steiglitz, 1982; Ji *et al.*, 1997). ∎

Let I denote some instance of the problem PLA and I_0 be the modification of I such that $r_i = 0$ for all $i \in J$. It is clear that I_0 is an instance of $PLA0$. Let π_0 denote the optimal solution of I_0, obtained according to Property 1. If we apply the solution π_0 to the original instance I we will have some feasible but not necessarily optimal solution. However, solution π_0 is optimal for PLA if the following property holds.

Proposition 2 *If $S_i \geqslant r_i$ for all $i \in J$ in solution π_0, then π_0 is the optimal solution to PLA.*

Proof. It is easy to see, that in such case, the release dates of jobs don't have impact on the makespan value. So the obtained solution is the optimal one. ∎

Finally, the following two properties, can be proved by simple job interchange argument.

Proposition 3 *For the problem PLA, if $v_i = v$, $n_{oi} = n_o$, $w_i = w$, and $n_{ai} = na$ for all $i \in J$ then the optimal solution to the problem can be obtained by ERD rule, i.e., by sorting jobs according the non-decreasing values of their release dates ($r_i \nearrow$).*

Proof. Assume that the permutation π is optimal and doesn't meet the statement of the property. Thus, there exists at least one pair of jobs $\pi(k)$ and $\pi(k+1)$ in π such that

$$r_{\pi(k)} > r_{\pi(k+1)}. \tag{3}$$

Let π' be a permutation constructed form π such that jobs $\pi(k)$ and $\pi(k+1)$ are interchanged. We will show that the completion time of the $k+1$th job in π' is less (or at most equal) to the $k+1$th job in π, i.e. $C_{\pi'(k+1)} \leqslant C_{\pi(k+1)}$. This observation leads to the conclusion that $C_{max}(\pi) \geqslant C_{max}(\pi')$ and π cannot be the optimal permuation.

Taking into account the formulae (2) and the fact that $C_{\pi(i)} = C_{\pi'(i)}$, for $i = 1, ..., k-1$ we can write the following equations:

$$\begin{aligned}
&C_{\pi(k)} = \max\{C_{\pi(k-1)}, r_{\pi(k)}\} + a_{\pi(k)} - v\min\{k, n_0\} + w\max\{0, k - n_a\},\\
&C_{\pi(k+1)} = \max\{C_{\pi(k)}, r_{\pi(k+1)}\} + a_{\pi(k+1)} - v\min\{k+1, n_0\} + w\max\{0, k+1 - n_a\},\\
&C_{\pi'(k)} = \max\{C_{\pi(k-1)}, r_{\pi(k+1)}\} + a_{\pi(k+1)} - v\min\{k, n_0\} + w\max\{0, k - n_a\},\\
&C_{\pi'(k+1)} = \max\{C_{\pi'(k)}, r_{\pi(k)}\} + a_{\pi(k)} - v\min\{k+1, n_0\} + w\max\{0, k+1 - n_a\}.
\end{aligned} \tag{4}$$

We will show that the value $\Delta = C_{\pi(k+1)} - C_{\pi'(k+1)}$ is always non-negative.

Following (4) we have:

$$\begin{aligned}
&\Delta = C_{\pi(k+1)} - C_{\pi'(k+1)} =\\
&\max\{C_{\pi(k)}, r_{\pi(k+1)}\} + a_{\pi(k+1)} - v\min\{k+1, n_0\} + w\max\{0, k+1 - n_a\} -\\
&\max\{C_{\pi'(k)}, r_{\pi(k)}\} - a_{\pi(k)} + v\min\{k+1, n_0\} - w\max\{0, k+1 - n_a\} =\\
&\max\{C_{\pi(k)}, r_{\pi(k+1)}\} + a_{\pi(k+1)} - \max\{C_{\pi'(k)}, r_{\pi(k)}\} - a_{\pi(k)}.
\end{aligned}$$

Noticing that $C_{\pi(k-1)} < C_{\pi(k)}$ and $C_{\pi'(k-1)} < C_{\pi'(k)}$, and taking into account (3) we have to consider the following 4 exhaustive cases:

1) $r_{\pi(k+1)} < r_{\pi(k)} \leqslant C_{\pi(k+1)}$
We have

$$\begin{aligned}&\Delta = \max\{C_{\pi(k)}, r_{\pi(k+1)}\} + a_{\pi(k+1)} - \max\{C_{\pi'(k)}, r_{\pi(k)}\} - a_{\pi(k)} =\\ &C_{\pi(k)} + a_{\pi(k+1)} - C_{\pi'(k)} - a_{\pi(k)} =\\ &C_{\pi(k-1)} + a_{\pi(k)} - v\min\{k, n_0\} + w\max\{0, k-n_a\} + a_{\pi(k+1)} -\\ &C_{\pi(k-1)} - a_{\pi(k+1)} + v\min\{k, n_0\} - w\max\{0, k-n_a\} - a_{\pi(k)} = 0.\end{aligned}$$

2) $r_{\pi(k+1)} \leqslant C_{\pi(k-1)} < r_{\pi(k)} \leqslant C_{\pi'(k)}$
We have

$$\begin{aligned}&\Delta = \max\{C_{\pi(k)}, r_{\pi(k+1)}\} + a_{\pi(k+1)} - \max\{C_{\pi'(k)}, r_{\pi(k)}\} - a_{\pi(k)} =\\ &C_{\pi(k)} + a_{\pi(k+1)} - C_{\pi'(k)} - a_{\pi(k)} =\\ &r_{\pi(k)} + a_{\pi(k)} - v\min\{k, n_0\} + w\max\{0, k-n_a\} + a_{\pi(k+1)} -\\ &C_{\pi(k-1)} - a_{\pi(k+1)} + v\min\{k, n_0\} - w\max\{0, k-n_a\} - a_{\pi(k)} =\\ &r_{\pi(k)} - C_{\pi(k-1)} > 0.\end{aligned}$$

3) $r_{\pi(k+1)} \leqslant C_{\pi(k-1)} < C_{\pi'(k)} \leqslant r_{\pi(k)}$
We have

$$\begin{aligned}&\Delta = \max\{C_{\pi(k)}, r_{\pi(k+1)}\} + a_{\pi(k+1)} - \max\{C_{\pi'(k)}, r_{\pi(k)}\} - a_{\pi(k)} =\\ &C_{\pi(k)} + a_{\pi(k+1)} - r_{\pi(k)} - a_{\pi(k)} =\\ &r_{\pi(k)} + a_{\pi(k)} - v\min\{k, n_0\} + w\max\{0, k-n_a\} + a_{\pi(k+1)} - r_{\pi(k)} - a_{\pi(k)} =\\ &a_{\pi(k+1)} - v\min\{k, n_0\} + w\max\{0, k-n_a\} = p_{\pi(k+1)} > 0.\end{aligned}$$

4) $C_{\pi(k-1)} < r_{\pi(k+1)} < r_{\pi(k)} \leqslant C_{\pi'(k)}$
We have

$$\begin{aligned}&\Delta = \max\{C_{\pi(k)}, r_{\pi(k+1)}\} + a_{\pi(k+1)} - \max\{C_{\pi'(k)}, r_{\pi(k)}\} - a_{\pi(k)} =\\ &C_{\pi(k)} + a_{\pi(k+1)} - C_{\pi'(k)} - a_{\pi(k)} =\\ &r_{\pi(k)} + a_{\pi(k)} - v\min\{k, n_0\} + w\max\{0, k-n_a\} + a_{\pi(k+1)} -\\ &r_{\pi(k+1)} - a_{\pi(k+1)} + v\min\{k, n_0\} - w\max\{0, k-n_a\} + a_{\pi(k)} =\\ &r_{\pi(k)} - r_{\pi(k+1)} > 0.\end{aligned}$$

We have shown, that in every case the value of Δ is non-negative, which completes the proof. ■

Proposition 4 *For the problem PLA, if $v_i = v$, $n_{oi} = n_o$, $w_i = w$, and $r_i = 0$ for all $i \in J$ then the optimal solution to the problem can be obtained by sorting jobs according the non-decreasing values of their parameters n_{ai} ($n_{ai} \nearrow$).*

Proof. The proof is very similar to the proof of Property 3 and therefore will be omitted. ■

5. HEURISTIC ALGORITHMS

The considered problem is strongly NP-hard so it is highly unlikely to find an algorithm which can solve this problem in polynomial time. Therefore we propose a few efficient heuristics to find an optimal solution. To solve the problem we constructed four algorithms based on Properties 1–4. These algorithms are denoted respectively as $r_i \nearrow$, $n_{ai} \nearrow$, $ASSIGNMENT$, and NEH.

In the $r_i \nearrow$ algorithm ties are broken according to the non-decreasing values of the parameter $v_i n_{oi} - w_i n_{ai}$. In the $n_{ai} \nearrow$ algorithm ties are broken according to the non-increasing values of the product $w_i n_{ai}$. The NEH algorithm is a direct adaptation of the insertion procedure proposed by Nawaz *et al.* (1983).

All tests were made on the computer with Intel Core^{TM} 2 Duo 3.00GHz, 2GB RAM and Windows 7. We constructed four sets of instances with different parameters of jobs. For every set, we described the interval of numbers for every parameter of job in the instance. Parameters for every particular job in the instance are uniformly distributed from those interval.

Test set 1 (small r_i values compared with a_i, small v_i and w_i values compared with a_i)

$$r_i \in [10, 50n], a_i \in [50, 100], v_i \in [0.1, 1], w_i \in [0.1, 1], n_{oi} \in [0, n/2-1], n_{ai} \in [n/2, n-1]$$

where n is a number of jobs.

Test set 2 (large r_i values compared with a_i, small v_i and w_i values compared with a_i)

$$r_i \in [10, 100n], a_i \in [50, 100], v_i \in [0.1, 1], w_i \in [0.1, 1], n_{oi} \in [0, n/2-1], n_{ai} \in [n/2, n-1]$$

where n is a number of jobs.

Test set 3 (small r_i values compared with a_i, large v_i and w_i values compared with a_i)

$$r_i \in [10, 8n], a_i \in [5, 10], v_i \in [0.1, 1], w_i \in [0.1, 1], n_{oi} \in [0, n/2-1], n_{ai} \in [n/2, n-1]$$

where n is a number of jobs.

Test set 4 (large r_i values compared with a_i, large v_i and w_i values compared with a_i)

$$r_i \in [10, 40n], a_i \in [5, 10], v_i \in [0.1, 1], w_i \in [0.1, 1], n_{oi} \in [0, n/2-1], n_{ai} \in [n/2, n-1]$$

where n is a number of jobs.

To compare the efficiency of the algorithms, we define the relative gap to the best (optimal) objective value as:

$$\frac{C^A_{max} - C^{OPT}_{max}}{C^{OPT}_{max}} \times [100\%],$$

where C^A_{max} is a criterion value found by an algorithm $A \in \{r_i \nearrow, n_{ai} \nearrow, ASSIGNMENT, NEH\}$, for a given instance, and C^{OPT}_{max} is an optimal value of the criterion for an instance with $n \leqslant 9$, which was generated by an exhaustive search method, and the best found value found by all the algorithms for $n > 9$.

Table 1. *The efficiency of the algorithms for the instances of the Test set 1.*

n	$r_i \nearrow$	$n_{oi} \nearrow$	*ASSIGN*	*NEH*
9	0.2	30.6	1.6	1.2
50	0.2	51.8	1.0	6.8
100	0.3	59.9	0.5	9.9
average	0.2	47.4	1.0	6.0

Table 2. *The efficiency of the algorithms for the instances of the Test set 2.*

n	$r_i \nearrow$	$n_{oi} \nearrow$	*ASSIGN*	*NEH*
9	0.0	35.9	5.9	2.2
50	0.0	49.3	10.8	8.5
100	0.0	48.6	11.9	8.6
average	0.0	44.6	9.5	6.4

Table 3. *The efficiency of the algorithms for the instances of the Test set 3.*

n	$r_i \nearrow$	$n_{oi} \nearrow$	*ASSIGN*	*NEH*
9	0.4	37.5	5.0	2.7
50	0.0	44.5	9.8	20.4
100	0.1	48.1	11.0	31.5
average	0.2	43.4	8.6	18.2

Table 4. *The efficiency of the algorithms for the instances of the Test set 4.*

n	$r_i \nearrow$	$n_{oi} \nearrow$	*ASSIGN*	*NEH*
9	0.0	8.4	1.5	0.1
50	0.0	8.0	2.5	1.9
100	0.0	10.1	3.2	4.5
average	0.0	8.8	2.4	2.2

Table 5. *The average efficiency of the algorithms for the instances of all the Test sets.*

n	$r_i \nearrow$	$n_{oi} \nearrow$	*ASSIGN*	*NEH*
average	0.1	36.1	5.4	8.2

The results of tests for every set are presented in Tables 1 to 4. For every set, algorithm and values of $n \in \{9, 50, 100\}$, we calculated a performance ratio. Last row of the table is an average performance ratio over the test set for every algorithm. In Table 5, we can find the values of total performance ratio, which is an average value of average performance ratios over all test sets.

According to our expectations, the efficiency of algorithms which are based on the release date values is the best. The rest of the algorithms are based on different parameters of jobs and the release dates do not influence on the order of jobs in a solution. If there is a job with large value of release date and the algorithm schedule this job on the first position, the solution generated by this algorithm can be far from an optimal one. The efficiency of those algorithms depends on the relation between parameters of the job, but it increases if r_i values are small in comparison with a_i.

6. CONCLUSIONS

In this paper we introduced a new model of scheduling problems, which takes into account the learning and ageing effects simultaneously. In this model the processing time of a job is described by piecewise linear function of its position in the jobs sequence and consists of three phases of the processor efficiency: the learning phase, the maturity phase, and the ageing phase. On the basis of this model we formulated the single machine makespan minimization problem and showed its computational complexity (strong NP-hardness) as well as its special cases that can be solved in polynomial time.

On the basis on showed problem properties we proposed 4 heuristic algorithms to solve general case of the problem. The efficiency of proposed algorithms is verified by experimental analysis and shows their practical applicability in the environments that accepts relative error about 10 percent to the optimal solution.

The model introduced in this paper can be extended to other machine environments (such as parallel machines, shop problems), as well as introduce more complex (nonlinear) shapes of the learning-ageing functions. However, it would be sense to consider (from practical point of view) such kind of problems if some reasonable real-life examples of applications of the considered models are found.

ACKNOWLEDGMENTS

The work of this paper was partially supported by the project number N N519 315235 of Ministry of Science and Higher Education of Republic of Poland.

REFERENCES

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

Bachman, A., Janiak, A. and Kovalyov, M. (2002). Minimizing the total weighted completion time of deteriorating jobs. *Information Processing Letters* **81**(2), 81–84.

Badiru, A., 1992. Computational survey of univariate and multivariate learning curve models. *IEEE Transactions on Engineering Management* **39**(2), 176–188.

Biskup, D., 1999. Single-machine scheduling with learning considerations. *European Journal of Operational Research* **115**, 173–178.

Biskup, D., 2008. A state-of-the-art review on scheduling with learning effects. *European Journal of Operational Research* **188**(2), 315–329.

Carlson, J., Rowe, A., 1976. How much does forgetting cost. *Industrial Engineering* **8**(9), 40–47.

Cheng, T., Ding, Q. and Lin, B., 2004. A concise survey of scheduling with time-dependent processing times. *European Journal of Operational Research* **152**(1), 1–13.

Cheng, T., Lee, W. and Wu, C., 2010. Scheduling problems with deteriorating jobs and learning effects including proportional setup times. *Computers & Industrial Engineering* **58**(2), 326–331.

Cheng, T. and Wang, G., 2000. Single machine scheduling with learning effect considerations. *Annals of Operations Research* **98**(1), 273–290.

Cheng, T., Wu, C. and Lee, W., 2008. Some scheduling problems with deteriorating jobs and learning effects. *Computers & Industrial Engineering* **54**(4), 972–982.

Dutton, J. and Thomas, A., 1984. Treating progress functions as a managerial opportunity. *Academy of Management Review* pp. 235–247.

Gawiejnowicz, S. and Kononov, A., 2010. Complexity and approximability of scheduling resumable proportionally deteriorating jobs. *European Journal of Operational Research* **200**(1), 305–308.

Jaber, M., Bonney, M., 1999. The economic manufacture/order quantity (emq/eoq) and the learning curve: past, present, and future. *International journal of production economics* **59**(1–3), 93–102.

Janiak, A., Janiak, W. and Lichtenstein, M., 2007. Resource management in machine scheduling problems: A survey. *Decision Making in Manufacturing and Services* **1**, 59–89.

Janiak, A., Krysiak, T. and Trela, R., 2011. Scheduling problems with learning and ageing effects: A survey. *Decision Making in Manufacturing and Services* **5**, to appear.

Janiak, A., Śnieżyk, A., 2004. Scheduling with position dependent processing times. In Z. Bubnicki, O. Hryniewicz and J. Węglarz (eds), *Badania operacyjne i systemowe 2004. Zastosowania*, Akademicka Oficyna Wydawnica EXIT, Warszawa, Poland, pp. 177–188.

Janiak, A., Śnieżyk, A., 2005a. Single processor scheduling with the linear aging efect, ready times and the makespan criterion. *Proceedings of 11th IEEE International Conference on Methods and Models in Automation and Robotics*, pp. 25–28.

Janiak, A., Śnieżyk, A., 2005b. Single processors scheduling with the linear aging effect, ready times and makespan criterion. *Proceedings of 11th IEEE International Conference on Methods and Models in Automation and Robotics*, pp. 1073–1078.

Janiak, A., Rudek, R., 2010. Scheduling jobs under an aging effect. *Journal of the Operational Research Society* **61**(6), 1041–1048.

Ji, P., Lee, W. B. and Li, H., 1997. A new algorithm for the assignment problem: An alternative to the hungarian method. *Computers Operational Research* **24**(11), 1017–1023.

Jordan, R., 1965. *How to use the learning curve*. Materials Management Institute.

Kuo, W., Yang, D., 2006. Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect. *European Journal of Operational Research* **174**(2), 1184–1190.

Lee, W., 2004. A note on deteriorating jobs and learning in single-machine scheduling problems. *International Journal of Business* **3**(1), 83–89.

Mosheiov, G., 2001a. Parallel machine scheduling with a learning effect. *Journal of the Operational Research Society* **52**(10), 1165–1169.

Mosheiov, G., 2001b. Scheduling problems with a learning effect. *European Journal of Operational Research* **132**(3), 687–693.

Mosheiov, G., Sidney, J., 2003. Scheduling with general job-dependent learning curves. *European Journal of Operational Research* **147**(3), 665–670.

Nawaz, M., Enscore, E. E. and Ham, I., 1983. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, *Omega* **11**(1), 91–95.

Papadimitrou, C., Steiglitz, K., 1982. *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ.

Shabtay, D., Steiner, G., 2007. A survey of scheduling with controllable processing times. *Discrete Applied Mathematics* **155**(13), 1643–1666.

Sun, L., 2009. Single-machine scheduling problems with deteriorating jobs and learning effects. *Computers & Industrial Engineering* **57**(3), 843–846.

Toksari, M., Isleyen, S., Guner, E. and Baykoç, Ö., 2010. Assembly line balancing problem with deterioration tasks and learning effect. *Expert Systems with Applications* **37**(2), 1223–1228.

Wang, J., 2007. Single-machine scheduling problems with the effects of learning and deterioration. *Omega* **35**(4), 397–402.

Wang, J., Huang, X., Wang, X., Yin, N. and Wang, L., 2009. Learning effect and deteriorating jobs in the single machine scheduling problems. *Applied Mathematical Modelling* **33**(10), 3848–3853.

Wang, J. and Liu, L., 2009. Two-machine flow shop problem with effects of deterioration and learning. *Computers & Industrial Engineering* **57**(3), 1114–1121.

Wright, T., 1936. Factors affecting the cost of airplanes. *Journal of Aeronautical Science* **3(4)**, 122–128.

Yang, D. and Kuo, W., 2009. Single-machine scheduling with both deterioration and learning effects. *Annals of Operations Research* **172**(1), 315–327.

Yang, S. and Yang, D., 2010. Single-machine scheduling problems with aging/deteriorating effect under an optional maintenance activity consideration. *INFOR: Information Systems and Operational Research* **48**(3), 171–179.

Yin, Y., Xu, D., Sun, K. and Li, H., 2009. Some scheduling problems with general position-dependent and time-dependent learning effects. *Information Sciences* **179**(14), 2416–2425.