Decision Making in Manufacturing and Services
Vol. 5 • 2011 • No. 1–2 • pp. 19–36

Scheduling Problems with Learning and Ageing Effects: A Survey

Adam Janiak*, Tomasz Krysiak*, Radosław Trela*

Abstract. In recent years, many papers concerning scheduling problems with simultaneous learning and ageing effects were published. In this paper, the state of the art of research concerning these problems is presented. In order to facilitate understanding this subject, the scheduling problems where these effects occur separately, are firstly explained. Then, the papers devoted to scheduling problems combining the effects of learning and ageing are discussed. Particular attention was paid on practical applications of the considered scheduling problems. After thorough analysis it turned out that both scheduling problems with learning effect, and with ageing effect, as well as, in particular, the problems with models merging learning and ageing effects do not have any reasonable real-life applications. This is because the learning and ageing effects are in general long time horizon phenomena observed in repetitive systems and scheduling theory concerns either with repetitive short-horizon planning problems or single long-horizon projects. Therefore, there is no sense to continue research considering these scheduling problems from practical (computer engineering, automatic control, technical and economical) point of view, unless such reasonable real-life example appears.

Keywords: scheduling, learning effect, ageing effect, real-life applications

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

Revised: 25 August 2011

1. INTRODUCTION

In the last decade of 20th century two phenomena, called a learning effect and an ageing effect, were brought into the scheduling field. Problems with these effects can be perceived as problems with changeable efficiency of a processor. The learning effect (without scheduling process) may be observed, e.g., during organizing of the workplace, minimizing the number of tasks necessary to obtain the final product, etc. A processor improves then its efficiency, i.e., it needs less time to accomplish the task.

* Institute of Computer Engineering, Wrocław University of Technology, Control and Robotics, Poland. E-mail: {adam.janiak, tomasz.krysiak, radoslaw.trela}@pwr.wroc.pl

The opposite effect, observed (without scheduling process) during processing of tasks which reduce the efficiency of a processor (e.g., blunting of a blade, wear of drill bits, tiredness of a doctor), is called an ageing effect.

This paper is a survey of scheduling problems where both effects exist simultaneously. In order to facilitate understanding them, the scheduling problems where these effects occur separately, are firstly explained. Therefore, in Section 2, a learning effect is presented and the most important scheduling models covering it are shown. Section 3 concerns an ageing effect. The state of the art of problems combining these phenomena is presented in Section 4. Finally, all the presented results are summarized in Section 5.

2. SCHEDULING PROBLEMS WITH A LEARNING EFFECT

A learning effect, i.e. a phenomenon of improving efficiency of a processor, was firstly observed and described by Wright (1936). Into scheduling theory it was brought by Biskup (1999) as a dependency of job processing times on the number of already processed jobs (i.e., on the position of the job in a sequence). However, the learning effect in the context of scheduling was defined already in the 80s and 90s of 20th century (see e.g., Meilijson and Tamir (1984), Dondeti and Mohanty (1998), where the processing times were dependent on the total processing time of already processed jobs rather than their number). The obtained results in this domain were summarized recently by Biskup (2008). Anyway, the main idea of scheduling problems with the learning effect is that a processor gains an experience while executing jobs. This experience improves efficiency of the processor and as a result it helps to shorten the processing times of the jobs (i.e., the more experience the processor gains, the more the processing time of the next job is shortened).

Relation between the job processing time and the experience of the processor is called a learning curve. The following three learning curves were investigated so far: Wright's curve (Figure 1), S-shaped curve (Figure 2) and stepwise curve (Figure 3).

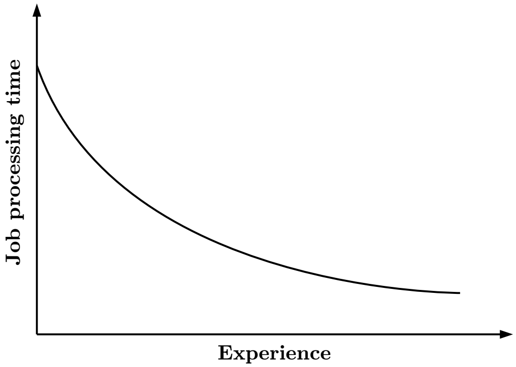

Fig. 1. *Wright's curve.*

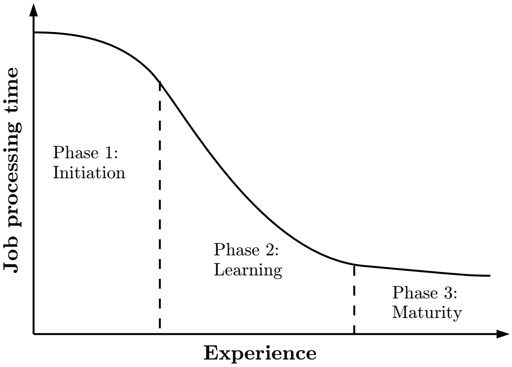

Fig. 2. *S-shaped curve.*

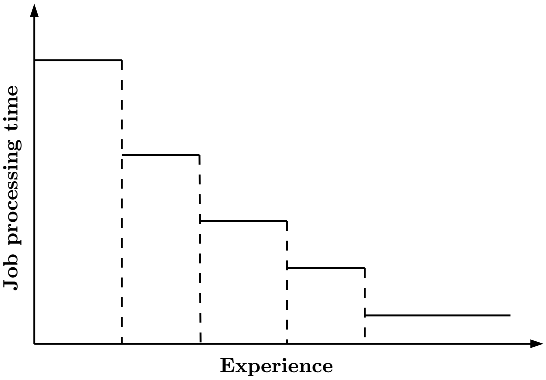

Fig. 3. *Stepwise curve.*

In the Wright's curve it is assumed that first unit of the experience gives the most shortening of the job processing time and each next unit experience gives less shortening of this time because of a saturation or weariness (thus it looks like convex decreasing exponential function). The S-shaped curve describes the learning effect more precisely, since it includes three main phases of this phenomenon, i.e., initiation, learning and maturity. In the first phase, the processor does not acquire the experience in its maximal capabilities (e.g. because of organizing of a workplace) – thus the function decreases slowly. The processor is the most efficient in acquiring the experience in the next phase – just after the initiation – then the function decreases the fastest. Finally, in the last phase, the processor possesses almost the whole possible experience and thus each next unit experience gives less shortening of the job processing time (similarly like in the Wright's curve). As a result, we obtain a function which resembles the letter "S". The stepwise curve describes the learning phenomenon in discrete way and one can obtain different shapes by matching its parameters (value and length of each step).

Three models of gaining experience were formulated in the analyzed papers: the model dependent on a number of processed jobs, on a sum of normal processing times of finished jobs and general model where each job can provide different experience to the processor.

Now some selected, most important models of processing time covering learning effect will be shown.

Biskup (1999) introduces the model in which processing time of a job depends on position of the job in a sequence (in other words it depends on the number of jobs already processed) and which realizes Write's curve:

$$p_{j[r]} = p_j r^{-\alpha}. \tag{1}$$

In the above equation $p_{j[r]}$ denotes an actual processing time of job j sequenced in rth position, p_j is a normal processing time (without learning) of job j and $\alpha > 0$ is a learning index.

Generalization of this model is considered by Mosheiov and Sidney (2003). They assume that each job can have its own learning index α_j:

$$p_{j[r]} = p_j r^{-\alpha_j}. \tag{2}$$

Janiak and Rudek (2005) find out that existing models of learning are rather crude simplification of complex industrial problems. Therefore, they consider scheduling problems with more general, experience based processing time:

$$p_{j[r]} = p_j (\sum_{l=1}^{r} e_{[l]})^{-\alpha_j}, \tag{3}$$

where $\sum_{l=1}^{r} e_{[l]}$ is an experience of a processor at the moment when processing of the job scheduled in rth position is started (this experience is gained by processing the jobs from positions $1, \ldots, r$). Note that the above model is generalization of the model of Mosheiov and Sidney (2003), i.e., it is based on the Write's curve and it is assumed that each job can provide different amount of experience, e_j, to the processor (unlike the above mentioned position-based models, where each job provides an amount of experience equal to 1).

The same authors discover that experience gaining is often a continuous process (Janiak and Rudek, 2007). They introduce a general model of experience obtained by the processor at the beginning of processing of a job sequenced in rth position, $E_\beta(r)$. In this model, it is assumed that a part of the experience from a job sequenced in rth position is gained at the beginning of its processing and the remaining part after the completion of this job:

$$E_\beta(r) \triangleq E(r-1) + \beta_{[r]} e_{[r]} = \sum_{l=1}^{r-1} e_{[l]} + \beta_{[r]} e_{[r]},$$

where $E(r-1)$ is the total experience gained by the processor during processing the jobs from the positions $1,\dots,r-1$ and $\beta_{[r]} \in [0,1]$ is a parameter which determines the part of experience $e_{[r]}$ provided to the processor at the beginning of processing of job sequenced in rth position. Based on the above model of the experience, the authors provide a general model of the job processing time with the learning effect and the S-shaped learning curve:

$$p_{j[r]} = p_j - b_j \left(\min\{E_\beta(r), g_j\}\right)^{\alpha_j} , \tag{4}$$

where g_j denotes a learning threshold of job j and α_j is a learning index of this job. In Janiak and Rudek (2008) these authors introduce a stepwise function of processing time including the general model of experience $E_\beta(r)$:

$$p_{j[r]} = \begin{cases} p_{j,1},\ 0 \leqslant E_\beta(r) < g_{j,1} \\ p_{j,2},\ g_{j,1} \leqslant E_\beta(r) < g_{j,2} \\ \vdots \\ p_{j,k},\ g_{j,k-1} \leqslant E_\beta(r) \end{cases} . \tag{5}$$

The authors analyze this model, since it is not restricted to any particular shape and thus it can fit every possible learning function by matching its parameters (value, $p_{j,q}$, and length, $[g_{j,q-1}, g_{j,q}]$, of each step $q = 1,\dots,k$).

The authors of the above mentioned papers describe there only short and general examples of learning phenomenon (without scheduling process) in practical situations, mainly in assembly environment where workers improve their skills by repeating similar operations (i.e., they learn). These examples seem to be intuitive, however there is not presented full and exhaustive real-life example (considering also scheduling process), with all necessary details, where scheduling process, specific environment of processors, model of job processing times, etc., would be noticeable explicitly.

Another research direction in the area of scheduling with the learning effect is to analyze the problems with the job processing times dependent on time spent by processor on executing earlier jobs rather than only on number of these jobs. As an example of this phenomenon, an optimization of surface inspection of steel plates or bars in a foundry is given. There is a set of steel plates or bars that need to be checked if they do not have any surface deffects. Time required to perform the inspection of one element depends on its area (plates) or length (bars). Authors claim that a person who makes inspection learns more if more time spends on checking. In scheduling context a set of elements to be checked is a set of jobs and a person that perform this quality control may be viewed as a processor. Unfortunately, again the example is superficial and unconvincing (especially the learning process).

In the considered area, Kuo and Yang (2006) propose the following model:

$$p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^{-\alpha_j} , \tag{6}$$

where $p_{[l]}$ denotes normal processing time of job scheduled in lth position and $\alpha_j > 0$ is a learning index of job j.

Similar model is investigated also by Koulamas and Kyparisis (2007):

$$p_{j[r]} = p_j \left(1 - \sum_{l=1}^{r-1} \frac{p_{[l]}}{\sum_{i=1}^{n} p_i}\right)^{\alpha}. \tag{7}$$

Finally, it is worth mentioning that the concept of learning under scheduling context was mentioned some years earlier (Meilijson and Tamir (1984), Dondeti and Mohanty (1998)), however, these authors do not consider any specified model of job processing times. They provide only some preliminary general considerations.

To sum up practical motivation of the analyzed research, there should be emphasized that the presented examples of practical application of scheduling problems with the learning effect are too brief. They usually contain only a description of the learning effect or (rarely) only scheduling process, but do not contain a meaningful combination of both these processes. As long as full and exhaustive real-life example is not given, there is no sense to do research (from practical point of view) in the area of scheduling with the learning effect. Recently, there were published two papers (Rudek (2011), Rudek and Wielgus (2010)) examining even the learning effect in a multiprocessor flow shop environment without any real-life practical justification.

3. SCHEDULING PROBLEMS UNDER AN AGEING EFFECT

An ageing effect is a phenomenon of reduction the processor efficiency as a result of its fatigue (wear). Thus, the processing times of the subsequent jobs executed by the processor increase. Job processing time may be an (increasing) function of: job starting time (deteriorating jobs), position or sum of normal processing times of jobs already processed. Moreover, there was considered a general model, in which processing of one job may cause different wear of the processor than processing of another job. Analogous to the learning effect, several types of relation between job processing time and wear of a processor were considered: exponential, linear, piecewise linear, exponential with ageing threshold and stepwise.

There are much less papers on the scheduling problems with the ageing effect than the ones with the learning effect – see, e.g., Yang and Yang (2010) for a survey. Let us show some selected, most important results in this area.

A model with linear curve of wear is investigated by Bachman and Janiak (2004):

$$p_{j[r]} = p_j + \alpha_j r, \tag{8}$$

where $p_{j[r]}$ denotes actual processing time of the job j scheduled in position r, p_j is normal processing time of this job and $\alpha_j \geqslant 0$ is an ageing index. The authors describe short and general practical example of the phenomenon of ageing in the context of decreasing sharpness of drills, however they do not explain scheduling process, model of job processing times, etc.

Kunnathur and Gupta (1990) introduce a piecewise linear model and processing time dependent on starting time:

$$p_j(t) = p_j + \alpha_j \max\{t - g_j, 0\}, \tag{9}$$

where t is starting time of job j, α_j is its ageing index and g_j denotes ageing threshold dependent on this job.

Analogous model but with processing time dependent on position, r, of the job is formulated by Janiak and Śnieżyk (2005):

$$p_{j[r]} = p_j + \alpha_j \max\{r - g_j, 0\}\,. \tag{10}$$

The authors present an interesting example of practical application of the analyzed problem – an optimization of data transmission in a client-server type computer network. The problem is as follows. There are a server and a set of clients. The server connection has a limited bandwidth. When each client connects to the server, it gets some available part of the bandwith. In response, the server must send a large amount of initialization data (e.g. configuration data, program code) and keep the connection open. Note that every client requires different initialization data and a different amount of work from server to prepare data. When the number of established connections increases, resources (processor time, network bandwidth, etc.) become scarce. As a result, the time required to open new connection, prepare and send initialization data increases. What is important, is the fact that client requests occur in different moments. The objective is to determine such a sequence of processing of clients' requests that minimizes the time required to open and initialize all connections. This problem can be considered as a scheduling problem with an ageing effect, ready times and a makespan criterion. Although the example is interesting, there should be explained more precisely the details of network communication (especially in the context of scheduling clients' requests and ageing effect in initializing of new connections), since in the present form the described problem seems to be immaterial.

Janiak and Rudek (2010) introduce a general model of wear of the processor, $W(r-1) = \sum_{j=1}^{r-1} w_j$, caused by processing the jobs from the positions $1, \ldots, r-1$ of a sequence. Obviously, each job causes wear of the processor in different extent, which is described just by the parameter w_j. Based on the described general wear model, the authors investigate exponential curve with an ageing threshold:

$$p_{j[r]} = p_j + \alpha_j \left(\max\{W(r-1) - g_j, 0\}\right)^{\beta_j}\,. \tag{11}$$

In the above equation all indices are non-negative rational numbers and $W(r-1)$ is wear of the processor after completion of $r-1$ jobs. As a practical application for this model, a production with usage of Computer Numerical Control (CNC) drilling or cutting machines is shown. This example is based on that presented by Bachman and Janiak (2004) and again it is short and vague. Indeed, the both examples are unconvincing and do not include the scheduling process.

Similarly as for the scheduling problems with the learning effect, the presented practical examples of scheduling problems with the ageing effect are vague or incomplete. Especially, the process of scheduling is not explained sufficiently in the presented examples. Therefore, the conducted research should be much more justified from a practical point of view. Unless complete and convincing real-life example (in which the both scheduling and ageing processes would be shown) is not given, there is no use to consider such problems from practical point of view.

4. THE STATE OF THE ART OF SCHEDULING PROBLEMS WITH SIMULTANEOUS LEARNING AND AGEING EFFECTS

As it can be seen in the previous sections, effects of learning and ageing are known and have been investigated for several decades, however to the scheduling theory they were brought in the last decade of the 20th century. Since 2009 the papers covering scheduling problems with simultaneous learning and ageing are regularly published, although already in 2004 and 2007 two first papers in this domain have appeared. In this section, we give overview of just such problems. In the first subsection, we analyze the problems in which the learning effect is characterized by dependency of job processing times on position and the ageing effect is characterized by dependency on job starting times (i.e., there are deteriorating jobs). Subsection 4.2 contains problems with the same type of the learning effect as in previous section (i.e., dependency on position) and the ageing effect is characterized by dependency on sum of normal processing times of already finished jobs. In Subsection 4.3 we present the model of jobs processing times which does not depend on job position at all. Instead, it depends non-increasingly on the sum of normal processing times of finished jobs (the learning effect) and non-decreasingly on the completion time of last completed job (the ageing effect).

4.1. TIME AND POSITION DEPENDENT MODELS

As it was mentioned above, the first paper considering a scheduling problem with simultaneous learning and ageing effects is Lee (2004). He considers a model of job processing times dependent on starting times and positions (learning effect + deteriorating jobs) of the following form:

$$p_{j[r]} = \alpha_j t r^a, \tag{12}$$

where t is the job starting time, r is position of the job in a sequence, $a \leqslant 0$ is a learning index and α_j is a deterioration rate. The author shows that the single processor problems with criteria $C_{\max}$, $\sum C_j$ and $\sum L_j$ is polynomially solvable by SDR (the smallest deteriorating rate) principle. He proves also that the corresponding problems with a slightly different model: $p_{j[r]} = (p_0 + \alpha_j t)\, r^a$ (where p_0 is common basic processing time) are no longer optimally solvable by SDR principle.

No practical application is presented for the considered problem by Lee (2004). However, research on this kind of scheduling problems is continued by Wang (2007), where a short vague example of simultaneous existing of learning and deteriorarting effects is provided. In this description it is explained that workers learn by repeating some tasks, but they forget the acquired abilities while they are moved to another kind of tasks. Unfortunately, this description does not reflect the considered problem (in particular scheduling process and the considered model of processing times). In fact, Wang considers a slightly more general model of job processing times than Lee, namely:

$$p_{j[r]} = p_j \left(\alpha(t) + \beta r^a\right), \tag{13}$$

where p_j is the normal processing time, $\alpha(t)$ is an increasing function, β is a constant and $a \leqslant 0$ is the learning index. The author proves that the single processor problems with $C_{\max}$, $\sum C_j$, $\sum C_j^2$ criteria and some special cases with $\sum w_j C_j$ and $L_{\max}$ criteria are polynomially solvable.

Similar model is investigated by Yang and Kuo (2009),Wang and Guo (2010), Toksarı *et al.* (2010):

$$p_{j[r]} = (p_j + \alpha t)\, r^a, \tag{14}$$

where $a \leqslant 0$ and $\alpha \geqslant 0$ are learning and ageing indices, respectively. In the paper of Yang and Kuo (2009), single processor problems are considered. It is shown that minimization of $C_{\max}$, $\sum C_j$, $\sum C_j^k$ and some special cases of minimization of $\sum w_j C_j$, $L_{\max}$, $TADC$[1], $ETCP$[2] are polynomially solvable. In the paper of Wang and Guo (2010) it is proven that the single processor due-date assignment problem with effects of learning and ageing is still polynomially solvable ($O(n \log n)$). The same – short and vague – description of practical application as in Wang (2007) is given in the both above mentioned papers. What is more, Wang and Guo (2010) give another example (independent of the previous one) describing due-date assignment (but without the learning and ageing effects!).

Instead, simply assembly line balancing problem (SALBP) is investigated by Toksarı *et al.* (2010). Authors show that SALBP with this model of job processing time can be solved optimally in polynomial time, using mixed nonlinear integer programming. Moreover, for bigger instances of the problem a heuristic algorithm is constructed. The authors give a short practical example (and references to several other papers) of SALBP problem only with the learning effect (without ageing or job deterioration effect).

Yang and Kuo (2010) formulate the following model of processing time:

$$p_{j[r]} = p_j r^{b_j} + \alpha t, \tag{15}$$

where $b_j \leqslant 0$ denotes a learning index of job j and $\alpha > 0$ is an ageing index. In this paper the special case with learning index common for all jobs ($b_j = b$) is investigated as well. Some single processor scheduling problems and flow shop problems are taken into consideration. It is proven that single processor problems and some flow shop problems with criteria $C_{\max}$, $\sum C_j$ and $TADC$ are polynomially solvable. However, authors do not show practical applications for their research.

The models, in which learning effect and ageing effect multiply together, are also considered in the scientific literature. Namely, Wang *et al.* (2009b) formulate job processing time as:

$$p_{j[r]} = \alpha_j (b + ct) r^a, \tag{16}$$

[1] $TADC = \sum_{i=1}^{n} \sum_{j=1}^{n} |C_i - C_j|$ — total absolute differences in completion times

[2] $ETCP = \sum_{i=1}^{n} (\alpha E_i + \beta T_i + \gamma d)$ — sum of earliness, tardiness and common due-date penalty (α, β, γ — some coefficients)

where α_j is an ageing index of job j, $a \leqslant 0$ is a learning index and b, c, d are some constant parameters. They also consider setup times:

$$s_{[r]} = d\sum_{i=1}^{r-1} p^*_{[i]},$$

where $p^*_{[i]}$ denotes actual processing time of job scheduled in position i. They prove that single processor problems with such model of processing time and setups and criteria $C_{\max}$, $\sum C_j$ and $\sum C_j^\delta$ are polynomially solvable. Moreover, some special cases with criteria $L_{\max}$, $\sum U_j$, $\sum w_j C_j$ are proven to be solvable in polynomial time. Unfortunately, an example of a real life system with learning effect, ageing effect and setups is not given.

In this group of scheduling problems, there are such that include arbitrary function of starting time (Wang, 2009):

$$p_{j[r]} = p_j \alpha(t)\left(M + (1-M)\, r^a\right), \tag{17}$$

and

$$p_{j[r]} = p_j\left(\alpha(t) + M + (1-M)\, r^a\right), \tag{18}$$

where $\alpha(t)$ in the above equations is an increasing ageing function ($\alpha(0) \geqslant 0$) and $M \in [0,1]$ is a constant. It is shown that single processor problems with job processing time expressed by one of the above equations and criteria $C_{\max}$, $\sum C_j$ and $\sum C_j^2$ are polynomially solvable. Some special cases of minimization of $\sum w_j C_j$ and $L_{\max}$ are also shown to be polynomially solvable. However, explanation for such model of processing time is not sufficient (especially from practical point of view, since exactly the same example as in Wang (2007) is given).

Finally, the problems, in which ageing index depended on jobs and on processors, are analyzed (Wang and Liu, 2009), i.e. the processing time is expressed as:

$$p_{ij[r]} = \alpha_{ij} t r^a, \tag{19}$$

where $\alpha_{ij} > 0$ is an ageing index of job j processed on processor i and $a \leqslant 0$ is a learning index. Starting time of job must be a positive number ($t > 0$). In this paper some single processor polynomial solvable problems are provided. For two machine flow shop problem, domination rules and a lower bound are shown and they are later used in a branch and bound algorithm. What is more, some heuristic algorithms are constructed. Unfortunately, a real life system that can be modelled as a flow shop problem with effects of learning and ageing is not shown.

Different model – in which not power but exponential function of position is given – is introduced by Wang *et al.* (2009a):

$$p_{j[r]} = p_j\left(\alpha a^{r-1} + \beta\right)(bt + c). \tag{20}$$

In the above equation $a \in (0,1]$ is a learning index, $\alpha \geqslant 0$ and $\beta \geqslant 0$ are learning parameters and $b \geqslant 0$, $c \geqslant 0$ are ageing parameters. It is proven that problems with criteria $C_{\max}$, $\sum C_j$, $\sum C_j^\delta$ and some special cases with criteria $\sum w_j C_j$ and $L_{\max}$ are solvable in polynomial time. As a practical motivation of the considered model, the authors give the same short and vague example as in Wang (2007, 2009), Yang and Kuo (2009), Wang and Guo (2010). Moreover, they provide another short example (preheating of the ingot in steel production), but only to justify consideration of coefficient of starting time proportional to the normal processing time. Thus, the authors again do not provide exhaustive (i.e., including dependency of the job processing times on their starting times and positions in a sequence) and convincing practical example of the analyzed scheduling model.

4.2. TOTAL NORMAL PROCESSING TIME AND POSITION DEPENDENT MODELS

Another group of scheduling problems with learning and ageing effects are such that the processing times depend on sum of normal processing times of already finished jobs (not the starting times as in previous models) and on positions.

Cheng *et al.* (2010) introduce the following model of processing time:

$$p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} \log p_{[l]}\right)^a r^b, \tag{21}$$

where $p_{j[r]}$ is actual processing time of job j scheduled in rth position, p_j is normal processing time of this job, $p_{[l]}$ denotes normal processing time of job scheduled in lth position, $a < 0$ and $b > 0$ are learning and ageing indices, respectively. Logarithm in the above equation prevents dramatic decline of the processing time when number of jobs increases. Authors investigate single processor problems where setup before processing job j scheduled in rth position is described as follows:

$$s_{j[r]} = \gamma \sum_{l=1}^{r-1} p_{[l]},$$

where γ is a normalizing parameter ($0 < \gamma < 1$). It is proven that problems with criteria $C_{\max}$, $\sum C_j$, $\sum C_j^2$ are polynomially solvable. Moreover, some special, polynomially solvable cases of minimization of $\sum T_i$ and $L_{\max}$ are shown. As a practical application of the above mentioned model of setups, the manufacturing of integrated circuit boards is given. However, this example does not include such complicated model of job processing times as presented above.

Sun's paper (Sun, 2009) concerns a slightly different model of processing time:

$$p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^a r^b. \tag{22}$$

In the above equation $a \geqslant 1$ is an ageing index and $b < 0$ is a learning index. Contrary to the previous model, wear of the processor is a function of sum of normal processing times and learning is a function of position. It is shown that single processor problems of minimization $C_{\max}$, $\sum C_j$, $\sum C_j^2$ and special cases of minimization $\sum w_j C_j$ and $L_{\max}$ are polynomially solvable.

A similar model is introduced by Huang *et al.* (2010). In their model processing time is not a power function but exponential:

$$p_{j[r]} = p_j \left(1 + \sum_{i=1}^{r-1} p_{[l]}\right)^{a} \alpha^{r-1}, \tag{23}$$

where $a > 1$ and $\alpha \in (0, 1]$ are ageing and learning indices, respectively. In the paper some exact polynomial time algorithms for single processor problems with criteria $C_{\max}$, $\sum C_j$ are developed. Authors also construct polynomial time algorithms for special cases of minimization of $L_{\max}$ and $\sum w_j C_j$.

No detailed examples of practical applications of the models analyzed by Sun and by Huang *et al.* are provided in the above mentioned papers. There are only short vague sentences (similar in the both papers) that the phenomena of deteriorating jobs (i.e., starting time dependent processing times) and learning effect exist simultaneously in many real-life situations. Then few vague examples are provided (ignoring for instance the scheduling process). In our opinion, such sentences are not sufficient motivation (from practical point of view) to lead research on these – quite complicated – models.

Problems with the job processing times dependent on the sum of normal processing times already finished jobs and position of the job are also considered by Wang *et al.* (2010). They suggest the following model of processing time:

$$p_{j[r]} = p_j \left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l=1}^{n} p_l}\right)^{a_1} r^{a_2}, \tag{24}$$

where a_1 and a_2 are learning or ageing indices ($a_1 < 0$, $a_2 > 0$ — ageing, $a_1 > 0$, $a_2 < 0$ — learning) and p_0 is constant.

It is proven that special cases of single processor problems with this model of processing time and criteria $C_{\max}$, $\sum C_j$ and their weighted sum are polynomially solvable. For other special cases it is shown that optimal schedule is V-shaped with respect to job normal processing times. Unfortunately, authors again do not give an example of real life system that can be modelled in such way.

4.3. NON-POSITION DEPENDENT MODELS

A model proposed by Toksarı and Güner (2010) differs from all the above mentioned, since it is the only model which does not depend on job position in a sequence. Instead, the learning effect reveals by dependency on the total normal processing time (as ageing effect in some previously described models) and the ageing effect reveals by dependency on completion time of last completed job (which is similar to starting time dependency occurring in deteriorating jobs). The featured model has the following form:

$$p_{j[r]} = \left[p_j + \left(\alpha + {C_{[r-1]}}^{b}\right)\right] \left(1 + \sum_{k=1}^{r-1} p_{[k]}\right)^{a}. \tag{25}$$

In this model, $C_{[r-1]}$ is ompletion time of job scheduling in position $r-1$, $a \leqslant 0$ is a learning index, $\alpha > 0$ and $b > 0$ are deterioration indices. Also a special case of the model with $b = 1$ is considered. In this paper earliness/tardiness (ET) scheduling problem on a parallel processor environment with common due-date is considered. It is shown that the optimal solution is V-shaped under certain agreeable conditions. Also an exact algorithm solving the problem is constructed. Unfortunately, practical applications for the research are not shown.

From analysis given in this section follows that authors introduce new, quite complicated models of job processing times including simultaneously the learning and ageing effects. However, introducing them to scheduling theory, the authors do not give sensible justification including complete and convincing real-life application example, consisting simultaneously scheduling process together with learning and ageing effects. Therefore, there is no use to do research on them from practical point of view until such an appropriate example is given.

5. CONCLUSIONS

In this paper, the state of the art of scheduling problems with simultaneous effects of learning and ageing was shown. Such problems are an emerging concept in scheduling theory. The papers covering them have been published since 2009 (to the best of our knowledge, only two papers with preliminary results in this domain have appeared in 2004 and 2007).

Mainly single processor problems were investigated so far. Authors focused on polynomial solvable cases. Complexity of many problems, especially with criteria $L_{\max}$, $\sum w_j C_j$, $\sum T_j$, is still open. A list of the considered problems in the described area is presented in Table 1.

Since the effects of learning and ageing were previously considered separately in the context of job scheduling, we gave also brief overview of these results. The following additional observation arises from the above analysis. Namely, authors consider scheduling problems with new models of job processing times (including only learning effect, or only ageing effect, or both learning and ageing effects), but they do not provide appropriate – full, sensible and detailed – practical example of the analyzed problems. We did not find any complete example: they were too vague, unrealistic, did not show scheduling process, or simply did not reflect the mathematical complexity of analyzed model. This situation could be explained intuitively as follows.

The learning and ageing effects are in general long time horizon phenomena observed in repetitive systems and scheduling theory concerns either with repetitive short-horizon planning problems or single long-horizon projects. Therefore, from computer engineering, automatic control, technical and economical point of view there is no sense to consider these scheduling problems, unless an appropriate reasonable practical example is presented.

Table 1. *Results obtained for scheduling problems with simultaneous effects of learning and ageing.*

Problem	Algorithm	Source
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} \log p_{[l]}\right)^{a_1} r^{a_2}, s_{psd}\vert C_{\max}$	SPT	(Cheng *et al.*, 2010)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} \log p_{[l]}\right)^{a_1} r^{a_2}, s_{psd}\vert \sum C_j$	SPT	(Cheng *et al.*, 2010)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} \log p_{[l]}\right)^{a_1} r^{a_2}, s_{psd}\vert \sum C_j^2$	SPT	(Cheng *et al.*, 2010)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} \log p_{[l]}\right)^{a_1} r^{a_2}, s_{psd}, d_i \leqslant d_j, p_i \leqslant p_j\vert \sum T_i$	EDD	(Cheng *et al.*, 2010)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} \log p_{[l]}\right)^{a_1} r^{a_2}, s_{psd}, d_i \leqslant d_j, p_i \leqslant p_j\vert L_{\max}$	EDD	(Cheng *et al.*, 2010)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^{a} \alpha^{r-1}\vert C_{\max}$	SPT	(Huang *et al.*, 2010)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^{a} \alpha^{r-1}\vert \sum C_j^{\delta}$	SPT	(Huang *et al.*, 2010)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^{a} \alpha^{r-1}, p_i > p_j, w_i \leqslant w_j\vert C_{\max}$	WSPT	(Huang *et al.*, 2010)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^{a} \alpha^{r-1}, d_i > d_j, p_i > p_j\vert L_{\max}$	EDD	(Huang *et al.*, 2010)
$1\vert p_{j[r]} = p_j r^b + \alpha t\vert C_{\max}$	SPT	(Yang and Kuo, 2010)
$1\vert p_{j[r]} = p_j r^b + \alpha t\vert TC$	SPT	(Yang and Kuo, 2010)
$1\vert p_{j[r]} = p_j r^b + \alpha t\vert TADC$	other $O(n \log n)$	(Yang and Kuo, 2010)
$1\vert p_{j[r]} = p_i r^b + \alpha t\vert \{C_{\max}, TC, TADC\}$	other $O(n^2)$	(Yang and Kuo, 2010)
$Fm\vert p_{ji[r]} = p_{ji} r^b \alpha t, M_1 \prec M_2 \prec \ldots \prec M_m\vert \{C_{\max}, TC\}$	other $O(n \log n)$	(Yang and Kuo, 2010)
$F2\vert p_{ji[r]} = p_{ji} r^b \alpha t, p_{j1} = p_1, p_{12} \leqslant p_{22} \leqslant \ldots \leqslant p_{n2}\vert \{C_{\max}, TC\}$	SPT	(Yang and Kuo, 2010)
$F2\vert p_{ji[r]} = p_{ji} r^b \alpha t, p_{j2} = p_2, p_{11} \leqslant p_{21} \leqslant \ldots \leqslant p_{n1}\vert \{C_{\max}, TC\}$	SPT	(Yang and Kuo, 2010)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^{a} r^b\vert C_{\max}$	SPT	(Sun, 2009)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^{a} r^b\vert \sum C_j$	SPT	(Sun, 2009)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^{a} r^b\vert \sum C_j^2$	SPT	(Sun, 2009)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^{a} r^b, p_j \leqslant p_l, w_j \geqslant w_l\vert \sum w_j C_j$	WSPT	(Sun, 2009)
$1\vert p_{j[r]} = p_j \left(1 + \sum_{l=1}^{r-1} p_{[l]}\right)^{a} r^b, p_j \leqslant p_l, d_j \leqslant d_l\vert L_{\max}$	SPT	(Sun, 2009)

Table 1. *(continued)*

Problem	Algorithm	Source
$1 \mid p_{j[r]} = \alpha(t)\left(M + (1-M)\,r^a\right), \alpha \geqslant 0 \mid C_{\max}$	SPT	(Wang, 2009)
$1 \mid p_{j[r]} = \left(\alpha(t) + M + (1-M)\,r^a\right), \alpha \geqslant 0 \mid C_{\max}$	SPT	(Wang, 2009)
$1 \mid p_{j[r]} = \alpha(t)\left(M + (1-M)\,r^a\right), \alpha \geqslant 0 \mid \sum C_j$	SPT	(Wang, 2009)
$1 \mid p_{j[r]} = \left(\alpha(t) + M + (1-M)\,r^a\right), \alpha \geqslant 0 \mid \sum C_j$	SPT	(Wang, 2009)
$1 \mid p_{j[r]} = \alpha(t)\left(M + (1-M)\,r^a\right), \alpha \geqslant 0 \mid \sum C_j^2$	SPT	(Wang, 2009)
$1 \mid p_{j[r]} = \left(\alpha(t) + M + (1-M)\,r^a\right), \alpha \geqslant 0 \mid \sum C_j^2$	SPT	(Wang, 2009)
$1 \mid p_{j[r]} = \alpha(t)\left(M + (1-M)\,r^a\right), \alpha \geqslant 0, p_j < p_k, w_j \geqslant w_k \mid \sum w_j C_j$	WSPT	(Wang, 2009)
$1 \mid p_{j[r]} = \left(\alpha(t) + M + (1-M)\,r^a\right), \alpha \geqslant 0, p_j < p_k, w_j \geqslant w_k \mid \sum w_j C_j$	WSPT	(Wang, 2009)
$1 \mid p_{j[r]} = \alpha(t)\left(M + (1-M)\,r^a\right), \alpha \geqslant 0, p_j = p \mid \sum w_j C_j$	WSPT	(Wang, 2009)
$1 \mid p_{j[r]} = \left(\alpha(t) + M + (1-M)\,r^a\right), \alpha \geqslant 0, p_j = p \mid \sum w_j C_j$	WSPT	(Wang, 2009)
$1 \mid p_{j[r]} = \alpha(t)\left(M + (1-M)\,r^a\right), \alpha \geqslant 0, p_j < p_k, d_j \leqslant d_k \mid L_{\max}$	EDD	(Wang, 2009)
$1 \mid p_{j[r]} = \left(\alpha(t) + M + (1-M)\,r^a\right), \alpha \geqslant 0, p_j < p_k, d_j \leqslant d_k \mid L_{\max}$	EDD	(Wang, 2009)
$1 \mid p_{j[r]} = \alpha(t)\left(M + (1-M)\,r^a\right), \alpha \geqslant 0, p_j = p \mid L_{\max}$	EDD	(Wang, 2009)
$1 \mid p_{j[r]} = \left(\alpha(t) + M + (1-M)\,r^a\right), \alpha \geqslant 0, p_j = p \mid L_{\max}$	EDD	(Wang, 2009)
$1 \mid p_{j[r]} = \alpha_j t r^a \mid C_{\max}$	SDR	(Wang and Liu, 2009)
$F2 \mid p_{j[r]} = \alpha_j t r^a, \alpha_{2j} = \alpha_2 \mid \sum C_j$	SDR	(Wang and Liu, 2009)
$F2 \mid p_{j[r]} = \alpha_j t r^a, \alpha_{1j} = \alpha_{2j} \mid \sum C_j$	SDR	(Wang and Liu, 2009)
$F2 \mid p_{j[r]} = \alpha_j t r^a \mid \sum C_j$	B&B	(Wang and Liu, 2009)
$F2 \mid p_{j[r]} = \alpha_j t r^a \mid \sum C_j$	HA	(Wang and Liu, 2009)
$1 \mid p_{j[r]} = p_j \left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l-1}^{r-1} p_l}\right)^{a_1} r^{a_2}, a_1 \geqslant 1, a_2 \leqslant 0 \mid C_{\max}$	SPT	(Wang *et al.*, 2010)
$1 \mid p_{j[r]} = p_j \left(\frac{p_0 + \sum_{l=1}^{r-1} p_{[l]}}{p_0 + \sum_{l-1}^{r-1} p_l}\right)^{a_1} r^{a_2}, a_1 \geqslant 1, a_2 \leqslant 0 \mid \sum C_j$	SPT	(Wang *et al.*, 2010)

Table 1. *Results obtained for scheduling problems with simultaneous effects of learning and ageing (continued).*

$1\vert p_{j[r]} = (p_i + \alpha t)\, r^a \vert C_{\max}$	SPT	(Yang and Kuo, 2009)
$1\vert p_{j[r]} = (p_i + \alpha t)\, r^a \vert \sum C_j$	SPT	(Yang and Kuo, 2009)
$1\vert p_{j[r]} = (p_i + \alpha t)\, r^a, p_i \leqslant p_j, w_i \geqslant w_j, w_i p_i \geqslant w_j p_j \vert \sum w_j C_j$	WSPT	(Yang and Kuo, 2009)
$1\vert p_{j[r]} = (p_i + \alpha t)\, r^a \vert \sum C_j^k$	SPT	(Yang and Kuo, 2009)
$1\vert p_{j[r]} = (p_i + \alpha t)\, r^a, p_i \leqslant p_j, d_i \leqslant d_j \vert L_{\max}$	EDD	(Yang and Kuo, 2009)
$1\vert p_{j[r]} = p_j \left(\alpha a^{r-1} + \beta\right)(bt + c) \vert C_{\max}$	SPT	(Wang *et al.*, 2009a)
$1\vert p_{j[r]} = p_j \left(\alpha a^{r-1} + \beta\right)(bt + c) \vert \sum C_j$	SPT	(Wang *et al.*, 2009a)
$1\vert p_{j[r]} = p_j \left(\alpha a^{r-1} + \beta\right)(bt + c) \vert \sum C_j^\delta$	SPT	(Wang *et al.*, 2009a)
$1\vert p_{j[r]} = p_j \left(\alpha a^{r-1} + \beta\right)(bt + c), p_j < p_k, w_j \geqslant w_k \vert \sum w_j C_j$	WSPT	(Wang *et al.*, 2009a)
$1\vert p_{j[r]} = p_j \left(\alpha a^{r-1} + \beta\right)(bt + c), p_j < p_k, d_j \leqslant d_k \vert L_{\max}$	EDD	(Wang *et al.*, 2009a)
$1\vert p_{j[r]} = \alpha_j (b + ct) r^a, s_{psd} \vert C_{\max}$	SDR	(Wang *et al.*, 2009b)
$1\vert p_{j[r]} = \alpha_j (b + ct) r^a, s_{psd} \vert \sum C_j$	SDR	(Wang *et al.*, 2009b)
$1\vert p_{j[r]} = \alpha_j (b + ct) r^a, s_{psd} \vert \sum C_j^\delta$	SDR	(Wang *et al.*, 2009b)
$1\vert p_{j[r]} = \alpha_j (b + ct) r^a, s_{psd}, \alpha_j \leqslant \alpha_k, w_j \geqslant w_k \vert \sum w_j C_j$	WSDR	(Wang *et al.*, 2009b)
$1\vert p_{j[r]} = \alpha_j (b + ct) r^a, s_{psd}, \alpha_j \leqslant \alpha_k, d_j \leqslant d_k \vert L_{\max}$	EDD	(Wang *et al.*, 2009b)
$1\vert p_{j[r]} = \alpha_j (b + ct) r^a, s_{psd}, \alpha_j \leqslant \alpha_k, d_j \leqslant d_k \vert \sum U_j$	Moore	(Wang *et al.*, 2009b)
$1\vert p_{j[r]} = \alpha_j (b + ct) r^a, s_{psd}, \alpha_j = \alpha \vert \sum U_j$	EDD	(Wang *et al.*, 2009b)
$1\vert p_{j[r]} = \alpha_j (b + ct) r^a, s_{psd}, d_j = d \vert \sum U_j$	SDR	(Wang *et al.*, 2009b)

SPT — shortest processing time,
WSPT — weighted shortest processing time,
EDD — earliest due date,
SDR — smallest deteriorating rate,
WSDR — weighted smallest deteriorating rate,
B&B — branch and bound,
HA — heuristic algorithm,
Moore — Moore's algorithm,
other — other algorithm.

REFERENCES

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

Biskup, D., 1999. Single-machine scheduling with learning considerations. *European Journal of Operational Research*, **115**(1), 173–178.

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

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

Dondeti, V. R., Mohanty, B. B., 1998. Impact of learning and fatigue factors on single machine scheduling with penalties for tardy jobs. *European Journal of Operational Research*, **105**, 509–524.

Huang, X., Wang, J.-B., Wang, L.-Y., Gao, W.-J., Wang, X.-R., 2010. Single machine scheduling with time-dependent deterioration and exponential learning effect. *Computers & Industrial Engineering*, **58**(1), 58–63.

Janiak, A., Rudek, R., 2005. On a general model of the learning effect. In: *Proceedings of the 11th IEEE International Conference on Methods and Models in Automation and Robotics*. Międzyzdroje, Poland, 29 August – 1 September 2005, 1115–1120.

Janiak, A., Rudek, R., 2007. The learning effect: Getting to the core of the problem. *Information Processing Letters*, **103**(5), 183–187.

Janiak, A., Rudek, R., 2008. A new approach to the learning effect: Beyond the learning curve restrictions. *Computers & Operations Research*, **35**(11), 3727–3736.

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

Janiak, A., Śnieżyk, A., 2005. Minimizing the maximum lateness of jobs with the aging effect. In: *Proceedings of the 11th IEEE International Conference on Methods and Models in Automation and Robotics*. Międzyzdroje, Poland, 29 August – 1 September 2005, 1067–1072.

Koulamas, C., Kyparisis, G. J., 2007. Single-machine and two-machine flowshop scheduling with general learning functions. *European Journal of Operational Research*, **178**(2), 402–407.

Kunnathur, A. S., Gupta, S. K., 1990. Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem. *European Journal of Operational Research*, **47**(1), 56–64.

Kuo, W.-H., Yang, D.-L., 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.-C., 2004. A note on deteriorating jobs and learning in single-machine scheduling problems. *International Journal of Business and Economics*, **3**(1), 83–89.

Meilijson, I., Tamir, A., 1984. Minimizing flow time on parallel identical processors with variable unit processing time. *Operations Research*, **32**(2), 440–448.

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

Rudek, R., 2011. Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect. *Computers & Industrial Engineering*, **61**, 20–31.

Rudek, R., Wielgus, A., 2010. Solution algorithms for some scheduling problems with learning considerations. In: A.Grzech, P.Świątek and K.Brzostowski (Eds.) *Applications of Systems Science*, pp. 33–42. Academic Publishing House EXIT, Warsaw.

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

Toksarı M. D., Güner, E., 2010. The common due-date early/tardy scheduling problem on a parallelmachine under the effects of time-dependent learning and linear and nonlinear deterioration. *Expert Systems with Applications*, **37**(1), 92–112.

Toksarı, M. D., İşleyen, S. K., Güner, E., Baykoç, Ö. F., 2010. Assembly line balancing problem with deterioration tasks and learning effect. *Expert Systems with Applications*, **37**(2), 1223–1228.

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

Wang, J.-B., 2009. Single-machine scheduling with learning effect and deteriorating jobs. *Computers & Industrial Engineering*, **57**(4), 1452–1456.

Wang, J.-B., Guo, Q., 2010. A due-date assignment problem with learning effect and deteriorating jobs. *Applied Mathematical Modelling*, **34**(2), 309–313.

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

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

Wang, J.-B., Jiang, Y., Wang, G., 2009b. Single-machine scheduling with past-sequence-dependent setup times and effects of deterioration and learning. *The International Journal of Advanced Manufacturing Technology*, **41**, 1221–1226.

Wang, J.-B., Wang, D., Zhang, G.-D., 2010. Single-machine scheduling problems with both deteriorating jobs and learning effects. *Applied Mathematical Modelling*, **34**(10), 2831–2839.

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

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

Yang, D.-L., Kuo, W.-H., 2010. Some scheduling problems with deteriorating jobs and learning effects. *Computers & Industrial Engineering*, **58**(1), 25–28.

Yang, S.-J., Yang, D.-L., 2010. Single-machine scheduling problems with aging/deteriorating effect under an optional maintenance activity consideration. *INFOR*, **48**(3), 171–179.