Approximation Schemes for Scheduling on a Single Machine Subject to Cumulative Deterioration and Maintenance

Hans Kellerer

Institut für Statistik und Operations Research, Universität Graz,
Universitätsstraße 15, A-8010, Graz, Austria
e-mail: hans.kellerer@uni-graz.at

Kabir Rustogi and Vitaly A. Strusevich

School of Computing and Mathematical Sciences, University of Greenwich,
Old Royal Naval College, Park Row, Greenwich, London SE10 9LS, U.K.
e-mail: {K.Rustogi,V.Strusevich}@greenwich.ac.uk

Abstract

We consider a scheduling problem on a single machine to minimize the makespan. The processing conditions are subject to cumulative deterioration, but can be restored by a single maintenance. We link the problem to the Subset-sum problem (if the duration of maintenance is constant) and to the Half-product problem (if the duration of maintenance depends on its start time). For both versions of the problem, we adapt the existing fully polynomial-time approximation schemes to our problems by handling the additive constants.

Keywords: Single machine; Deterioration; Maintenance; Approximation scheme; Subset-sum problem; Half-Product Problem

1 Introduction

In recent scheduling research, there has been a considerable interest in models in which the processing conditions are subject to changes that affect the actual processing times of the jobs. There are two opposite effects that are normally studied: deterioration and learning. Under *deterioration*, the later a job starts, the more time is required to process it, which is attributed to the fact that the processing conditions get worse (e.g., an operator gets tired, a machine-tool loses its quality, etc.) On the other hand, a *learning* effect leads to the decreasing of the actual processing times for the jobs that are scheduled later (e.g., an operator gets experience in performing jobs that require similar skills). See a survey by Biskup (2008) and a monograph by Gawiejnowicz (2008) for the results, motivations and practical applications in this area.

In this paper, we focus on single machine scheduling problems with a particular deterioration effect. The jobs of set $N = \{1, 2, \ldots, n\}$ have to be processed on a single machine. Each job $j \in N$ is associated with an integer p_j that is called its "normal" processing time. A meaningful interpretation of p_j is that it defines the processing duration of job j, provided that the machine is in the perfect condition. In scheduling literature, three basic types of deterioration effects can be found; see Gordon et al. (2008) where all three types are studied in one paper. Informally, they can be stated as follows:

- *Positional* Deterioration: the actual processing time of job j depends on p_j and on the position of the job in the sequence;

- *Time-Dependent* Deterioration: the actual processing time of job j depends on the start time of the job;

- *Cumulative* Deterioration: the actual processing time of job j depends on p_j and on the sum of normal processing times of all jobs sequenced earlier.

Notice that numerous studies on scheduling with deterioration/learning suffer from a common drawback: unless additional conditions are imposed the actual processing times either eventually go to zero (in the case of learning) or grow uncontrollably (in the case of deterioration). One way of handling this drawback is to look at more realistic truncated models, see, e.g., Wu et al. (2011) for truncated cumulative learning, and Kovalyov and Kubiak (1998) for truncated time deterioration. In truncated models, the actual processing time cannot become either less or larger than a given value.

In the case of deterioration, there is another approach to preventing actual processing times from becoming unacceptably large. This approach is more natural and relevant to the practical needs: the processing conditions can be restored, either fully or partly, by running *machine maintenance*. In this case, a special *maintenance period* (MP) is inserted into a schedule, and during each MP no processing is done on the machine.

Although scheduling problems with machine maintenance have been an object of extensive studies, quite often it appears that the introduced periods are maintenance periods only by name, not by nature. This, for example, happens when a fixed machine non-availability interval is called an MP, however such a model does not address the issues of machine deterioration and restoration. Partly free from this drawback are models that follow Lee and Leon (2001), in which an MP is treated as a *rate-modifying* activity.

The papers in which scheduling of maintenance activities is combined with consideration of various deterioration effects have started to appear only very recently, see Kuo and Yang (2008), Zhao and Tang (2010), Yang and Yang (2010) for initial work on single machine scheduling with maintenance and positional deterioration, and Lodree and Geiger (2010) for a study on single machine scheduling with maintenance and time-dependent deterioration.

This paper addresses the problem of minimizing the makespan on a single machine that is subject to cumulative deterioration and the decision-maker decides when to schedule a single maintenance period that completely restores the processing conditions. The duration of the MP is either a constant or is described by a linear function of its start time.

Notice that the problems with multiple maintenance periods and positional deterioration are polynomially solvable even in the most general settings, see Rustogi and Strusevich (2011). In the case of the cumulative deterioration of a fairly simple structure with only a single MP, the problem under consideration is NP-hard, and therefore we concentrate on the design of approximation schemes.

Recall that for a problem of minimizing a function $G(\mathbf{x})$, where $\mathbf{x}$ is a collection of decision variables, a polynomial-time algorithm that finds a feasible solution $\mathbf{x}^H$ such that $G(\mathbf{x}^H)$ is at most $\rho \geq 1$ times the optimal value $G(\mathbf{x}^*)$ is called a $\rho-$*approximation* algorithm; the value of ρ is called a *worst-case ratio bound*. A family of $\rho-$approximation algorithms is called a *fully polynomial-time approximation scheme (FPTAS)* if $\rho = 1 + \varepsilon$ for any $\varepsilon > 0$ and the running time is polynomial with respect to both the length of the problem input and $1/\varepsilon$.

The approach that we pursue in this study is based on linking the corresponding scheduling problem to problems of Boolean programming. In particular, for the problem with a constant MP, we show that the variable part of the objective function is related to the Subset-sum problem, a version of the linear knapsack problem; see Kellerer et al. (2004). On the other hand, if the duration of the MP depends linearly on its start time, we establish its link to a problem of quadratic Boolean programming, known as the Half-product problem, see Badics and Boros (1998) and Kellerer and Strusevich (2011). Although each of the mentioned Boolean programming problems admits an FPTAS, a challenge remains to adapt such an FPTAS to handling the original objective function. The latter task is not straightforward due to the presence of an additive constant of the sign that is opposite to the sign of the variable part of the function.

To illustrate this, consider a function of the form

$$F(\mathbf{x}) = G(\mathbf{x}) + K,$$

where $G(\mathbf{x})$ represents a variable part of the overall function $F(\mathbf{x})$ to be minimized, and K is a constant. If $\mathbf{x}^*$ minimizes the function $G(\mathbf{x})$, it will obviously minimize the function $F(\mathbf{x})$ as well. Suppose that for minimizing function $G(\mathbf{x})$ an FPTAS is available that delivers a solution $\mathbf{x}^H$, such that $G(\mathbf{x}^H) - G(\mathbf{x}^*) \leq \varepsilon |G(\mathbf{x}^*)|$.

For $\mathbf{x}^H$ to be accepted as an $\varepsilon-$approximate solution for minimizing the function $F(\mathbf{x})$, we must establish the inequality

$$F(\mathbf{x}^H) \leq (1 + \varepsilon) F(\mathbf{x}^*). \tag{1}$$

For a solution $\mathbf{x}^H$ found by an FPTAS for minimizing $G(\mathbf{x})$, we will have

$$F(\mathbf{x}^H) = G(\mathbf{x}^H) + K \leq G(\mathbf{x}^*) + \varepsilon |G(\mathbf{x}^*)| + K = F(\mathbf{x}^*) + \varepsilon |G(\mathbf{x}^*)|.$$

This leads to two cases.

Case 1: For $G(\mathbf{x}^*) \geq 0$, we have $F(\mathbf{x}^H) \leq F(\mathbf{x}^*) + \varepsilon G(\mathbf{x}^*) = (1 + \varepsilon) F(\mathbf{x}^*) - \varepsilon K$. If $K \geq 0$, the inequality (1) holds; however, if $K < 0$, there is no evidence that (1) will hold, and further analysis must be performed. We face the latter situation when studying the problem with a constant time MP in Section 3.

Case 2: For $G(\mathbf{x}^*) < 0$, we have $F(\mathbf{x}^H) \leq F(\mathbf{x}^*) - \varepsilon G(\mathbf{x}^*) \leq (1 + \varepsilon) F(\mathbf{x}^*) + \varepsilon K$. If $K \leq 0$, the inequality (1) holds; however, if $K > 0$, there is no evidence that (1) will hold, and further analysis must be performed. We face the latter situation when studying the problem with a start-time dependent MP in Section 4.

The remainder of this paper is organized as follows. Section 2 formally describes the problems under consideration; we also show that the problem with a constant MP is NP-hard. In Section 3, we show how an FPTAS by Kellerer et al. (2003) developed for the Subset-sum problem can be adapted to the scheduling problem with a constant MP. In Section 4, we show how an FPTAS by Erel and Ghosh (2008) developed for the Half-product problem can be adapted to the scheduling problem with a MP of a variable duration. Some concluding remarks can be found in Section 5.

2 Preliminaries

In this section, we give formal statements of the problems under consideration and establish their computational complexity.

The jobs of set $N = \{1, 2, \ldots, n\}$ have to be processed on a single machine. Each job $j \in N$ is associated with an integer p_j that is called its "normal" processing time. The machine is subject to deterioration. A maintenance period has to be run exactly once during the planning period and it will restore the machine conditions completely, i.e., after the MP the machine is as good as new.

Throughout this paper, for a non-empty subset N' define $p(N') := \sum_{j \in N'} p_j$; additionally define $p(\varnothing) := 0$. In a similar sense, we write $e(N')$, $q(N')$, etc.

Let the jobs be sequenced in accordance with a permutation π, so that $\pi(k)$ is the job sequenced in position k. In the case of no machine maintenance, one of the most common models for cumulative deterioration is as follows, see, e.g., Gordon et al. (2008): The actual processing time of a job j that is sequenced in position r of a permutation π is given by

$$
p_j(r) = p_j \left(1 + \sum_{k=1}^{r-1} p_{\pi(k)} \right)^Z,
\tag{2}
$$

where Z is a given non-negative constant which is common for all jobs. This definition is very similar to the one given by Kuo and Yang (2006a, 2006b) who have initiated a study on cumulative effects in the case of learning, i.e., they assume that $Z < 0$. Wu et al. (2011) list about a dozen of models with cumulative learning, in which $p_j(r)$ is expressed in different ways in terms of $\sum_{k=1}^{r-1} p_{\pi(k)}$. Models with precedence constraints and cumulative deterioration effects as given by (2), for $Z = 1$ and $Z = 2$, are studied by Gordon et al. (2008). As a rule, for the problems with cumulative deterioration and no machine maintenance, polynomial-time algorithms are derived.

In this paper, we focus on the models with a specific cumulative deterioration effect, such that the actual processing time of a job j that is sequenced in position r, $1 \leq r \leq n$ of a permutation π is given by

$$
p_j(r) = p_j \left(A_j + B \sum_{k=1}^{r-1} p_{\pi(k)} \right),
\tag{3}
$$

where A_j, $j \in N$, and B are positive constants. Comparing the above model with (2), we assume that $Z = 1$, while on the other hand, we extend the model (2) by introducing the additional coefficients A_j and B. These coefficients allow us to handle other variations of the problem, without the need to alter our methodology in any way. In a special case of our model, with $A_j = 1$, $j \in N$, and $B = p(N)^{-1}$, the effect (3) becomes equivalent to the model introduced by Koulamas and Kyparisis (2007).

In this paper, we distinguish between two versions of the maintenance periods:

(i) Constant maintenance: the duration of the MP is β time units, where $\beta > 0$.

(ii) Start-time dependent maintenance: the duration of the MP is $\alpha\tau + \beta$ time units, provided that the MP starts at time τ; here $\alpha > 0$ and $\beta \geq 0$.

For the latter type of maintenance, the later a machine is sent for maintenance, the longer it takes to restore it to an acceptable condition. This type of maintenance has been introduced by Kubzin and Strusevich (2005, 2006), and in combination with positional deterioration has recently been studied by Mosheiov and Sidney (2010), Rustogi and Strusevich (2011) and Yang and Yang (2010).

Given a schedule S, let $C_{\max}(S)$ denote the makespan, i.e., the maximum completion time. Extending standard notation for scheduling problems, we denote the problem of minimizing the makespan by $1\,|Cumu, MP(\alpha)|\,C_{\max}$, provided that the machine is subject to cumulative deterioration and the duration of the MP is equal to $\alpha\tau + \beta$. The version with constant maintenance is denoted by $1\,|Cumu, MP(0)|\,C_{\max}$.

An instance of problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$ is defined by the sequences p_j and A_j, $j \in N$, and numbers B and β, which are arbitrary positive integers. However, for $\alpha > 0$, we assume that α is bounded from above by a constant. This assumption is well justified by the fact that the duration of an MP is at least α times longer than the preceding period during which the machine was used. Without the made assumption, maintaining the machine would take considerably longer than the total processing time before the maintenance, which is hardly realistic.

In a schedule with a single MP the jobs are split into two groups: group 1 consists of the jobs scheduled before the maintenance and group 2 contains all other jobs. For problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$, consider a schedule S with two groups. Let N_i be the set of jobs in group i and $|N_i| = n_i$, for $i \in \{1, 2\}$. Let $\pi = (\pi(1), \ldots, \pi(n_1))$ and $\sigma = (\sigma(1), \ldots, \sigma(n_2))$ denote a sequence of jobs of set N_1 and N_2, respectively. In accordance with (3), the makespan of schedule S is given by

$$
\begin{aligned}
C_{\max}(S) \;=\; & p_{\pi(1)}A_{\pi(1)} + \sum_{r=2}^{n_1} p_{\pi(r)}\left(A_{\pi(r)} + B\sum_{k=1}^{r-1} p_{\pi(k)}\right) \\
& + \alpha\left(p_{\pi(1)}A_{\pi(1)} + \sum_{r=2}^{n_1} p_{\pi(r)}\left(A_{\pi(r)} + B\sum_{k=1}^{r-1} p_{\pi(k)}\right)\right) + \beta \\
& + p_{\sigma(1)}A_{\sigma(1)} + \sum_{r=2}^{n_2} p_{\sigma(r)}\left(A_{\sigma(r)} + B\sum_{k=1}^{r-1} p_{\sigma(k)}\right).
\end{aligned}
$$

The total processing time of the jobs in the first group can be computed as

$$
\begin{aligned}
p_{\pi(1)}A_{\pi(1)} + \sum_{r=2}^{n_1} p_{\pi(r)}\left(A_{\pi(r)} + B\sum_{k=1}^{r-1} p_{\pi(k)}\right) &= \sum_{r=1}^{n_1} p_{\pi(r)}A_{\pi(r)} + B\sum_{1 \le k < r \le n_1} p_{\pi(k)}p_{\pi(r)} \\
&= \sum_{r=1}^{n_1} p_{\pi(r)}A_{\pi(r)} + \frac{B}{2}\left(p(N_1)^2 - \sum_{r=1}^{n_1} p_{\pi(r)}^2\right).
\end{aligned}
$$

Similarly, the total processing time of the jobs in the second group can be computed as

$$
p_{\sigma(1)}A_{\sigma(1)} + \sum_{r=2}^{n_2} p_{\sigma(r)}\left(A_{\sigma(r)} + B\sum_{k=1}^{r-1} p_{\sigma(k)}\right) = \sum_{r=1}^{n_2} p_{\sigma(r)}A_{\sigma(r)} + \frac{B}{2}\left(p(N_2)^2 - \sum_{r=1}^{n_2} p_{\sigma(r)}^2\right).
$$

Define

$$
q_j = p_j A_j, \ j = 1, 2, \ldots, n,
$$

so that

$$q(N_1) := \sum_{r=1}^{n_1} p_{\pi(r)} A_{\pi(r)}, \ q(N_2) := \sum_{r=1}^{n_2} p_{\sigma(r)} A_{\sigma(r)}, \ q(N) := q(N_1) + q(N_2).$$

Thus, the makespan can be written as

$$C_{\max}(S) = q(N_1) + \frac{B}{2}\left(p(N_1)^2 - \sum_{r=1}^{n_1} p_{\pi(r)}^2\right) + \alpha\left(q(N_1) + \frac{B}{2}\left(p(N_1)^2 - \sum_{r=1}^{n_1} p_{\pi(r)}^2\right)\right) + \beta$$
$$+ q(N_2) + \frac{B}{2}\left(p(N_2)^2 - \sum_{r=1}^{n_2} p_{\sigma(r)}^2\right),$$

which implies

$$C_{\max}(S) = q(N) + \frac{B}{2}\left(p(N_1)^2 + p(N_2)^2 - \sum_{j \in N} p_j^2\right) + \alpha\left(q(N_1) + \frac{B}{2}\left(p(N_1)^2 - \sum_{j \in N_1} p_j^2\right)\right) + \beta \tag{4}$$

for problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$ and

$$C_{\max}(S) = q(N) + \frac{B}{2}\left(p(N_1)^2 + p(N_2)^2\right) - \frac{B}{2}\sum_{j \in N} p_j^2 + \beta \tag{5}$$

for problem $1\,|Cumu, MP(0)|\,C_{\max}$.

Notice that (4) and (5) demonstrate that for problems $1\,|Cumu, MP(\alpha)|\,C_{\max}$ and $1\,|Cumu, MP(0)|\,C_{\max}$ respectively, the order of jobs in each group does not affect the makespan. This complies with Gordon et al. (2008), where the makespan has been shown to be sequence independent for the single machine problem with the deterioration effect (2), with $Z = 1$, and no maintenance period. Thus, the main issue in solving problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$, including its simpler version $1\,|Cumu, MP(0)|\,C_{\max}$, is to find an appropriate partition of the jobs into two groups.

In Section 3, problem $1\,|Cumu, MP(0)|\,C_{\max}$ is proved NP-hard in the ordinary sense. It is clear that problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$ is no easier than problem $1\,|Cumu, MP(0)|\,C_{\max}$. Thus, the best possible approximation result that can be derived for either problem is an FPTAS. In the subsequent sections we develop such approximation schemes.

3 Constant Maintenance: FPTAS by Subset-Sum

In this section, we consider problem $1\,|Cumu, MP(0)|\,C_{\max}$. It follows from (5) that to achieve the minimum makespan we only need to partition the set N of jobs into two subsets, N_1 and N_2 in such a way that the value $p(N_1)^2 + p(N_2)^2$ is as small as possible.

First, we show that the smallest value of $p(N_1)^2 + p(N_2)^2$ can be achieved if the values $p(N_1)$ and $p(N_2)$ are as close as possible. The latter problem can be formulated as the

well-known *Subset-sum* problem:

$$\begin{array}{ll} \max & \sum_{j \in N} p_j x_j \\ \text{subject to} & \sum_{j \in N} p_j x_j \leq \Delta \\ & x_j \in \{0,1\}, \ j \in N, \end{array} \tag{6}$$

where $\Delta := p(N)/2$. In scheduling terms, this problem is interpreted as problem $P2 \,||\, C_{\max}$ of minimizing the makespan on two parallel identical with no preemption allowed. Jurisch et al. (1997) show that the latter problem reduces to maximizing the product $\left(\sum_{j \in N} p_j x_j \right) \left(\sum_{j \in N} p_j (1 - x_j) \right)$ for $x_j \in \{0,1\}, \ j \in N$.

For problem $1 \,|\, Cumu, MP(0) \,|\, C_{\max}$, let $x_j = 1$ if job j is assigned to set N_1; otherwise, define $x_j = 0$. The problem reduces to minimizing $p(N_1)^2 + p(N_2)^2 = \left(\sum_{j \in N} p_j x_j \right)^2 + \left(\sum_{j \in N} p_j (1 - x_j) \right)^2 = p(N) - 2 \left(\sum_{j \in N} p_j x_j \right) \left(\sum_{j \in N} p_j (1 - x_j) \right)$, i.e., to maximizing $\left(\sum_{j \in N} p_j x_j \right) \left(\sum_{j \in N} p_j (1 - x_j) \right)$, as in problem $P2 \,||\, C_{\max}$.

From this we immediately derive that problems $1 \,|\, Cumu, MP(0) \,|\, C_{\max}$ and $P2 \,||\, C_{\max}$ are essentially equivalent, and the following statements hold.

Proposition 1 *Problem $1 \,|\, Cumu, MP(0) \,|\, C_{\max}$ is NP-hard in the ordinary sense.*

Proposition 2 *Suppose that $x_j^* \in \{0,1\}, \ j \in N$, are the optimal values of the decision variables for the problem (6). Define $N_1^* := \left\{ j \in N | x_j^* = 1 \right\}$ and $N_2^* = N \backslash N_1^*$. Then for problem $1 \,|\, Cumu, MP(0) \,|\, C_{\max}$ there exists an optimal schedule S^* in which the jobs of set N_1^* are scheduled in one group and the jobs of set N_2^* are scheduled in the other group.*

Corollary 1 *For a schedule S^* that is optimal for problem $1 \,|\, Cumu, MP(0) \,|\, C_{\max}$ the following lower bound*

$$C_{\max}(S^*) \geq q(N) + B\Delta^2 - \frac{B}{2} \sum_{j \in N} p_j^2 + \beta \tag{7}$$

holds.

To see this, observe that for any partition of set N into subsets N_1 and N_2 the inequality $p(N_1)^2 + p(N_2)^2 \geq 2\Delta^2$ holds. Indeed, if for some non-negative δ we have that $p(N_1) = \Delta - \delta$ and $p(N_2) = \Delta + \delta$, then $p(N_1)^2 + p(N_2)^2 = 2\Delta^2 + 2\delta^2 \geq 2\Delta^2$.

Our further consideration is based on the following statement; see Kellerer et al. (2003) and Lemma 4.6.1 in Kellerer et al. (2004).

Theorem 1 *Consider the Subset-sum problem of the form*

$$\begin{array}{ll} \max & \sum_{j \in N} p_j x_j \\ \text{subject to} & \sum_{j \in N} p_j x_j \leq c \\ & x_j \in \{0,1\}, \ j \in N, \end{array} \tag{8}$$

This problem admits an FPTAS that for a given positive ε, either finds an optimal solution $x_j^* \in \{0,1\}$, $j \in N$, such that

$$\sum_{j \in N} p_j x_j^* < (1-\varepsilon)c$$

or finds an approximate solution $x_j^\varepsilon \in \{0,1\}$, $j \in N$, such that

$$(1-\varepsilon)c \le \sum_{j \in N} p_j x_j^\varepsilon \le c.$$

Such an FPTAS requires no more than $O\left(\min\left\{n/\varepsilon, n + \frac{1}{\varepsilon^2}\log\left(\frac{1}{\varepsilon}\right)\right\}\right)$ time.

The algorithm below assigns the jobs to groups in accordance with the above mentioned FPTAS, applied to problem (6) with $c = \Delta = p(N)/2$.

Algorithm Eps1

    INPUT: An instance of problem $1\,|Cumu, MP(0)|\,C_{\max}$ and an $\varepsilon > 0$

    OUTPUT: A schedule S^ε such that $C_{\max}(S^\varepsilon) \le (1+\varepsilon)\,C_{\max}(S^*)$

Step 1. For a given $\varepsilon > 0$ define $\varepsilon_0 := \frac{\varepsilon}{\varepsilon+1}$.

Step 2. With the defined ε_0, run an FPTAS for problem (6) to find the values $x_j^\varepsilon \in \{0,1\}$, $j \in N$. Define $N_1^\varepsilon := \left\{j \in N | x_j^\varepsilon = 1\right\}$ and $N_2^\varepsilon := N \backslash N_1^\varepsilon$.

Step 3. Output schedule S^ε for the original problem $1\,|Cumu, MP(0)|\,C_{\max}$, in which the jobs of set N_1^ε are assigned to one group and sequenced before the maintenance and the jobs of set N_2^ε are assigned to the other group to be scheduled after the maintenance. Stop.

Recall that the makespan as given in (5), consists of a variable part and a constant. Due to Proposition 2 and Theorem 1, the variable part can be minimized by means of an FPTAS. However, as discussed in Section 1, a direct application of that FPTAS does not necessarily result into an FPTAS for the original problem, since (5) contains a constant $q(N) + \beta - \frac{B}{2}\sum_{j \in N} p_j^2$, which can be negative. Below we prove that Algorithm Eps1 gives an appropriate treatment to the negative constant, and therefore allows us to adapt the existing FPTAS to deliver an $\varepsilon-$approximate solution for minimizing the overall original objective function.

Theorem 2 *Algorithm Eps1 is an FPTAS for problem $1\,|Cumu, MP(0)|\,C_{\max}$ that runs in $O\left(\min\left\{n/\varepsilon, n + \left(1 + \frac{1}{\varepsilon}\right)^2\log\left(1 + \frac{1}{\varepsilon}\right)\right\}\right)$ time.*

Proof: Using an FPTAS by Kellerer et al. (2003) from Theorem 1 with $\varepsilon_0 := \frac{\varepsilon}{\varepsilon+1}$, we observe that $O(n/\varepsilon_0) = O\left(n\frac{\varepsilon+1}{\varepsilon}\right) = O(n/\varepsilon)$ and $\frac{1}{\varepsilon_0^2}\log\left(\frac{1}{\varepsilon_0}\right) = \left(1 + \frac{1}{\varepsilon}\right)^2\log\left(1 + \frac{1}{\varepsilon}\right)$, so that the required running time is achieved. To complete the proof, we need to prove that $C_{\max}(S^\varepsilon) \le (1+\varepsilon)\,C_{\max}(S^*)$.

Due to Theorem 1, we only need to consider the case that the FPTAS in Step 2 does not find an optimal solution to problem (6); otherwise schedule S^ε is optimal. Below we

only look at the instances of problem $1\,|Cumu, MP(0)|\,C_{\max}$ for which $p_j \leq \Delta$, $j \in N$, since otherwise an optimal solution can be obtained by scheduling the largest job in one group and the remaining jobs in the other.

We assume that there exists a δ, $\delta \leq \varepsilon_0$, such that $(1 - \varepsilon_0)\Delta \leq p(N_1^\varepsilon) = \Delta\,(1 - \delta) < \Delta$ and $p(N_2^\varepsilon) = (1 + \delta)\,\Delta$. Applying (5) and (7), we have that

$$
\begin{aligned}
C_{\max}(S^\varepsilon) &= q(N) + \frac{B}{2}\left(p(N_1^\varepsilon)^2 + p(N_2^\varepsilon)^2\right) + \beta - \frac{B}{2}\sum_{j \in N} p_j^2 \\
&= q(N) + B\left(\Delta^2 + \delta^2\Delta^2\right) + \beta - \frac{B}{2}\sum_{j \in N} p_j^2 \leq C_{\max}(S^*) + B\delta^2\Delta^2.
\end{aligned}
$$

Below we demonstrate that $B\delta(1 - \delta)\Delta^2$ is a lower bound on the optimal makespan $C_{\max}(S^*)$. Consider the problem

$$
\begin{aligned}
\max \quad & \sum_{j \in N} p_j^2 \\
\text{subject to} \quad & \sum_{j \in N_1^\varepsilon} p_j = (1 - \delta)\,\Delta \\
& \sum_{j \in N_2^\varepsilon} p_j = (1 + \delta)\,\Delta \\
& 0 \leq p_j \leq \Delta, \; j \in N.
\end{aligned}
\tag{9}
$$

This problem is related to one of the basic problems of submodular optimization, a so-called resource allocation problem with a convex separable objective function; see Hochbaum and Hong (1995) and Katoh and Ibaraki (1998). The problem is known to be solvable by the greedy algorithm, which in the case under consideration, scans the values p_j in any order and gives each of them the largest possible value. In our case, the greedy algorithm will find an optimal solution to (9) in which one of the p_j's is equal to Δ, one to $(1 - \delta)\,\Delta$ and one to $\delta\Delta$, while all others are equal to zero. Thus,

$$
\sum_{j \in N} p_j^2 \leq (1 - \delta)^2\,\Delta^2 + \Delta^2 + (\delta\Delta)^2 = 2\Delta^2(1 + \delta^2 - \delta)
$$

provides an upper bound on the sum of squares of the processing times for all instances of the problem for which Step 2 of Algorithm Eps1 delivers $p(N_1^\varepsilon) = \Delta\,(1 - \delta)$ and $p(N_2^\varepsilon) = (1 + \delta)\,\Delta$, including the instance under consideration.

Substituting this into (7) we derive a lower bound

$$
C_{\max}(S^*) \geq q(N) + B\left(\delta - \delta^2\right)\Delta^2 + \beta \geq B\delta\,(1 - \delta)\,\Delta^2.
$$

This lower bound implies that

$$
C_{\max}(S^\varepsilon) \leq C_{\max}(S^*) + B\delta^2\Delta^2 \leq \left(1 + \frac{\delta}{1 - \delta}\right)C_{\max}(S^*).
$$

Since $\frac{\delta}{1-\delta}$ increases, we have that

$$
C_{\max}(S^\varepsilon) \leq \left(1 + \frac{\varepsilon_0}{1 - \varepsilon_0}\right)C_{\max}(S^*).
$$

Thus, to obtain an FPTAS for our problem with the accuracy ε, we need to use the FPTAS for problem (6) with $\varepsilon_0 = \frac{\varepsilon}{\varepsilon+1}$. $\blacksquare$

4 Start Time Dependent Maintenance: an FPTAS by Half-Product

In this section, we show that problem $1\,|Cumu, MP\,(\alpha)|\,C_{\max}$ can be formulated in terms of quadratic Boolean programming. We discuss an opportunity that this reformulation offers regarding the design of an FPTAS for the problem under consideration.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ denote a vector with n $0-1$ components. Introduce the function

$$H\left(\mathbf{x}\right) = \sum_{1 \le i < j \le n} a_i b_j x_i x_j - \sum_{j=1}^{n} h_j x_j, \tag{10}$$

where for each j, $1 \le j \le n$, the coefficients a_j and b_j are non-negative integers, while h_j is an integer that can be either negative or positive. Problems of quadratic Boolean programming similar to (10) were introduced in 1990s as mathematical models for various scheduling problems by Kubiak (1995) and Jurisch et al. (1997). The function $H\left(\mathbf{x}\right)$ is called a *Half-product* since its quadratic part consists of roughly half of the terms of the product $\left(\sum_{j=1}^{n} a_j x_j\right)\left(\sum_{j=1}^{n} b_j x_j\right)$. This function and the term "Half-product" were introduced by Badics and Boros (1998), who considered the problem of minimizing the function H with respect to Boolean decision variables with no additional constraints. Notice that we are only interested in the instances of the problem for which the optimal value of the function is strictly negative; otherwise, setting all decision variables to zero solves the problem. The problem of minimizing function $H\left(\mathbf{x}\right)$ of the form (10) is called the *Half-product problem*. The first FPTAS for the Half-Product problem that requires strongly polynomial time is due to Erel and Ghosh (2008), the running time is $O(n^2/\varepsilon)$.

Given problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$, introduce a Boolean variable x_j in such a way that

$$x_j = \begin{cases} 1, & \text{if job } j \text{ is scheduled in the first group} \\ 0, & \text{otherwise} \end{cases}$$

for each job j, $1 \le j \le n$. Taking the jobs in any order, i.e., in the order of their numbering, if job j is scheduled in the first group then it completes at time

$$C_j = p_j x_j \left(A_j + B \sum_{i=1}^{j-1} p_i x_i \right),$$

so that the MP starts at time $\sum_{j=1}^{n} p_j x_j \left(A_j + B \sum_{i=1}^{j-1} p_i x_i \right)$. If job j is scheduled in the second group, then

$$\begin{aligned} C_j &= \sum_{j=1}^{n} p_j x_j \left(A_j + B \sum_{i=1}^{j-1} p_i x_i \right) + \alpha \left(\sum_{j=1}^{n} p_j x_j \left(A_j + B \sum_{i=1}^{j-1} p_i x_i \right) \right) + \beta \\ &\quad + \sum_{j=1}^{n} p_j (1 - x_j) \left(A_j + B \sum_{i=1}^{j-1} p_i (1 - x_i) \right). \end{aligned}$$

This implies that in order to solve problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$, we need to minimize the function

$$F_\alpha\left(\mathbf{x}\right) = (\alpha + 1) \left(\sum_{j=1}^{n} p_j x_j \left(A_j + B \sum_{i=1}^{j-1} p_i x_i \right) \right)$$

$$+ \sum_{j=1}^{n} p_j(1 - x_j) \left(A_j + B \sum_{i=1}^{j-1} p_i(1 - x_i) \right) + \beta,$$

which can be rewritten as

$$F_\alpha\left(\mathbf{x}\right) \;=\; (\alpha + 1) \left(B \sum_{1 \le i < j \le n} p_i p_j x_i x_j + \sum_{j=1}^{n} p_j A_j x_j \right) \tag{11}$$

$$+ B \sum_{1 \le i < j \le n} p_i p_j (1 - x_i)(1 - x_j) + \sum_{j=1}^{n} p_j A_j (1 - x_j) + \beta.$$

The lemma below links the function $F_\alpha\left(\mathbf{x}\right)$ to the Half-product problem.

Lemma 1 *Function $F_\alpha\left(\mathbf{x}\right)$ can be represented as $F_\alpha\left(\mathbf{x}\right) \;=\; H\left(\mathbf{x}\right) + K$, where $H(\mathbf{x})$ is the half-product function of the form (10), with $a_i := (\alpha + 2) B p_i, b_j := p_j$ and $h_j := B\left(p_j p(N) - p_j^2\right) + \alpha p_j A_j$, $j \in N$, and the constant K is defined as*

$$K := \beta + q(N) + \sum_{1 \le i < j \le n} p_i p_j,$$

where $q(N) = \sum_{j=1}^{n} p_j A_j$.

Proof: It follows that

$$\sum_{1 \le i < j \le n} p_i p_j (1 - x_i)(1 - x_j) = \sum_{1 \le i < j \le n} p_i p_j x_i x_j + \sum_{1 \le i < j \le n} p_i p_j$$

$$- \sum_{j=1}^{n} p_j x_j \left(\sum_{i=1}^{j-1} p_i \right) - \sum_{j=1}^{n} p_j \left(\sum_{i=1}^{j-1} p_i x_i \right).$$

Notice that

$$\sum_{j=1}^{n} p_j \left(\sum_{i=1}^{j-1} p_i x_i \right) = \sum_{j=1}^{n} p_j x_j \left(\sum_{i=j+1}^{n} p_i \right)$$

so that

$$\sum_{1 \le i < j \le n} p_i p_j (1 - x_i)(1 - x_j) = \sum_{1 \le i < j \le n} p_i p_j x_i x_j + \sum_{1 \le i < j \le n} p_i p_j - \sum_{j=1}^{n} \left(p_j p(N) - p_j^2 \right) x_j.$$

Thus, (11) becomes

$$F_\alpha\left(\mathbf{x}\right) \;=\; (\alpha + 2) B \sum_{1 \le i < j \le n} p_i p_j x_i x_j - \sum_{j=1}^{n} \left(B p_j p(N) - B p_j^2 + \alpha p_j A_j \right) x_j$$

$$+ \left(\beta + \sum_{j=1}^{n} p_j A_j + B \sum_{1 \le i < j \le n} p_i p_j \right),$$

11

which proves the lemma. ∎

Consider the problem of minimizing the function $F(\mathbf{x}) = H(\mathbf{x}) + K$, where $H(\mathbf{x})$ is a Half-product function of the form (10), and K is a constant. It is known that an FPTAS for minimizing the function $H(\mathbf{x})$ does not necessarily behave as an FPTAS for minimizing the function $F(\mathbf{x})$. This is due to the fact the optimal value of $H(\mathbf{x})$ is negative; see Erel and Ghosh (2008) and Kellerer and Strusevich (2011) for discussion and examples. Suppose that a lower bound LB and an upper bound UB on the optimal value of the function $F(\mathbf{x})$ are available, i.e., $LB \leq F(\mathbf{x}^*) \leq UB$. Erel and Ghosh (2008) adopt their FPTAS for minimizing the function $H(\mathbf{x})$ to minimizing the function $F(\mathbf{x})$. They develop an algorithm that delivers a solution $\mathbf{x}^0$ such that $F(\mathbf{x}^0) - LB \leq \varepsilon LB$ in $O(\gamma n^2/\varepsilon)$ time, where $\gamma \geq UB/LB$. We refer to this version of the scheme as $\gamma-$FPTAS.

The makespan $C_{\max}(S)$ associated with a partition of the jobs $N = N_1 \cup N_2$ into two groups will be denoted by $F_\alpha(N_1, N_2)$ and defined by (4); for $\alpha = 0$ the makespan will be denoted by $F_0(N_1, N_2)$ and defined by (5).

Below we describe how to adapt the $\gamma-$FPTAS for solving problem $1 \,|Cumu, MP(\alpha)|\, C_{\max}$.

Algorithm Eps2

INPUT: An instance of problem $1 \,|Cumu, MP(\alpha)|\, C_{\max}$ with α bounded by a constant and an $\varepsilon > 0$

OUTPUT: A schedule S^ε such that $C_{\max}(S^\varepsilon) \leq (1 + \varepsilon)\, C_{\max}(S^*)$

Step 1. Given an instance for problem $1 \,|Cumu, MP(\alpha)|\, C_{\max}$, take an arbitrary positive ε' and run Algorithm Eps1 with $\varepsilon = \varepsilon'$, applied to the counterpart of the original problem with constant maintenance ($\alpha = 0$). Let $N_1^{\varepsilon'}$ and $N_2^{\varepsilon'}$ be the groups found by Algorithm Eps1. Compute $F_0(N_1^{\varepsilon'}, N_2^{\varepsilon'})$ by (5) with $N_1 = N_1^{\varepsilon'}$ and $N_2 = N_2^{\varepsilon'}$.

Step 2. Define $UB := \left(1 + \frac{\alpha}{2}\right) F_0(N_1^{\varepsilon'}, N_2^{\varepsilon'})$, $\gamma := \left(1 + \frac{\alpha}{2}\right)(1 + \varepsilon')$. Take a small positive ε and run the $\gamma-$FPTAS by Erel and Ghosh (2008). With the found values $x_j^\varepsilon \in \{0, 1\}$, $j \in N$, define $N_1^\varepsilon := \left\{j \in N | x_j^\varepsilon = 1\right\}$ and $N_2^\varepsilon = N \backslash N_1^\varepsilon$. If

$$q(N_1^\varepsilon) + \frac{B}{2} p(N_1^\varepsilon)^2 - \frac{B}{2} \sum_{j \in N_1^\varepsilon} p_j^2 > q(N_2^\varepsilon) + \frac{B}{2} p(N_2^\varepsilon)^2 - \frac{B}{2} \sum_{j \in N_2^\varepsilon} p_j^2,$$

swap N_1^ε and N_2^ε.

Step 3. Output schedule S^ε for the original problem $1 \,|Cumu, MP(\alpha)|\, C_{\max}$, in which the jobs of set N_1^ε are assigned to one group and sequenced before the maintenance and the jobs of set N_2^ε are assigned to the other group to be scheduled after the maintenance. Stop.

Theorem 3 *Algorithm Eps2 is an FPTAS for problem $1 \,|Cumu, MP(\alpha)|\, C_{\max}$ that runs in $O(n^2/\varepsilon)$ time.*

Proof: It follows from (4) and (5) that

$$F_\alpha(N_1, N_2) = F_0(N_1, N_2) + \alpha \left(q(N_1) + \frac{B}{2} p(N_1)^2 - \frac{B}{2} \sum_{j \in N_1} p_j^2 \right).$$

12

Besides, for the purpose of finding the best schedule for problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$ defined by a partition $N = N_1 \cup N_2$ we may assume that

$$q(N_1) + \frac{B}{2}p(N_1)^2 - \frac{B}{2}\sum_{j \in N_1} p_j^2 \leq q(N_2) + \frac{B}{2}p(N_2)^2 - \frac{B}{2}\sum_{j \in N_2} p_j^2,$$

otherwise, we will swap the groups scheduled before and after the maintenance. This implies that

$$q(N_1) + \frac{B}{2}p(N_1)^2 - \frac{B}{2}\sum_{j \in N_1} p_j^2 \leq \frac{1}{2}F_0(N_1, N_2),$$

and therefore

$$F_\alpha(N_1, N_2) \leq \left(1 + \frac{\alpha}{2}\right) F_0(N_1, N_2). \tag{12}$$

Let S_α^* denote a schedule that is optimal for problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$. That schedule is defined by a partition of the set N of jobs into two subsets, which we denote by $N_1^*(\alpha)$ and $N_1^*(\alpha)$. In particular, $N_1^*(0)$ and $N_1^*(0)$ define an optimal schedule for problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$. Let also $N_1^{\varepsilon'}$ and $N_2^{\varepsilon'}$ be the sets that are found in Step 1 of Algorithm Eps2. Due to (12) we have

$$F_\alpha(N_1^*(\alpha), N_2^*(\alpha)) \leq F_\alpha(N_1^{\varepsilon'}, N_2^{\varepsilon'}) \leq \left(1 + \frac{\alpha}{2}\right) F_0(N_1^{\varepsilon'}, N_2^{\varepsilon'}).$$

On the other hand,

$$F_\alpha(N_1^*(\alpha), N_2^*(\alpha)) \geq F_0(N_1^*(0), N_2^*(0)) \geq \frac{F_0(N_1^{\varepsilon'}, N_2^{\varepsilon'})}{(1 + \varepsilon')}.$$

Thus, for the optimal makespan in problem $1\,|Cumu, MP(\alpha)|\,C_{\max}$, we deduce that $\frac{F_0(N_1^{\varepsilon'}, N_2^{\varepsilon'})}{(1+\varepsilon')}$ is a lower bound, while $\left(1 + \frac{\alpha}{2}\right) F_0(N_1^{\varepsilon'}, N_2^{\varepsilon'})$ is an upper bound, and therefore the values of UB and γ in Step 2 are correct. The overall running time of Algorithm Eps2 is determined by the time complexity of Step 2. According to Erel and Ghosh (2008), the γ−FPTAS requires $O(\gamma n^2/\varepsilon)$, which in our case becomes $O(n^2/\varepsilon)$, since γ only depends on a given α bounded by a constant and on a chosen constant ε'. Algorithm Eps2 will deliver a solution of the required accuracy, i.e., $C_{\max}(S^\varepsilon)/C_{\max}(S^*) \leq 1 + \varepsilon$. ∎

5 Conclusion

The paper makes a contribution to the fast developing area of scheduling with rate-modifying maintenance activities, which allow us to restore the conditions of the machine that may get worse during the processing. Mathematically, the considered models are linked to linear and quadratic problems of Boolean programming that admit an FPTAS. Our main technical task has been to adapt the known FPTASs to our problems, which is not straightforward due to the opposite signs of the variable and constant parts of the objective function.

The next step in studying the models with cumulative deterioration could be a search for approximation algorithms or schemes that would allow us to handle multiple maintenance periods.

References

Badics, T., & Boros, E. (1998). Minimization of half-products. *Mathematics of Operations Research*, 33, 649–660.

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

Erel, E., & Ghosh, J.B. (2008). FPTAS for half-products minimization with scheduling applications. *Discrete Applied Mathematics*, 156, 3046–3056.

Gawiejnowicz, S. (2008). *Time-Dependent Scheduling*. Berlin: Springer.

Gordon, V.S., Potts, C.N., Strusevich, V.A., Whitehead, J.D. (2008). Single machine scheduling models with deterioration and learning: Handling precedence constraints via priority generation. *Journal of Scheduling*, 11, 357–370.

Hochbaum, D.S., & Hong, S.-P. (1995). About strongly polynomial time algorithms for quadratic optimization over submodular constraints. *Mathematical Programming*, 69, 269–309.

Jurisch, B., Kubiak. W., Józefowska, J. (1997). Algorithms for minclique scheduling problems. *Discrete Applied Mathematics*, 72, 115–139.

Katoh, N. & Ibaraki, T. (1998). Resource allocation problems. In D.-Z. Du and P.M. Pardalos (Eds.) *Handbook of Combinatorial Optimization*, Vol. 2 (pp. 159–260), Dordrecht: Kluwer.

Kellerer, H., Mansini, R., Pferschy, U., Speranza, M.G. (2003). An efficient fully polynomial approximation scheme for the Subset-Sum Problem. *Journal of Computer and System Sciences*, 66, 349–370.

Kellerer, H., Pferschy, U., Pisinger, D. (2004). *Knapsack Problems*. Berlin: Springer.

Kellerer, H., & Strusevich, V.A. (2010a). Fully polynomial approximation schemes for a symmetric quadratic knapsack problem and its scheduling applications. *Algorithmica*, 57, 769–795.

Kellerer, H., & Strusevich, V.A. (2010b). Minimizing total weighted earliness-tardiness on a single machine around a small common due date: An FPTAS using quadratic knapsack. *International Journal of Foundations of Computer Science*, 21, 357–383.

Kellerer, H., & Strusevich, V.A. (2011). The symmetric quadratic knapsack problem: approximation and scheduling applications. Submitted.

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

Kovalyov, M.Y., & Kubiak, W. (1998). A fully polynomial approximation scheme for minimizing makespan of deteriorating jobs. *Journal of Heuristics*, 3, 287–297.

Kubiak, W. (1995) New results on the completion time variance minimization. *Discrete Applied Mathematics,* 58, 157–168

Kubzin, M.A., & Strusevich, V.A. (2005). Two-machine flow shop no-wait scheduling with machine maintenance. *4OR*, 3, 303-313.

Kubzin, M.A., & Strusevich, V.A. (2006). Planning machine maintenance in two-machine shop scheduling. *Operations Research*, 54, 789–800.

Kuo, W.-H., & Yang, D.-L. (2006a). Minimizing the makespan in a single machine scheduling problem with a time-based learning effect. *Information Processing Letters*, 97, 64–67.

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

Kuo, W.-H., & Yang, D.-L. (2008). Minimizing the makespan in a single-machine scheduling problem with the cyclic process of an aging effect. *Journal of the Operational Research Society*, 59, 416–420.

Lee, C.-Y., & Leon, V.J. (2001). Machine scheduling with a rate-modifying activity. *European Journal of Operational Research*, 128, 119–128.

Lodree Jr., E. J., & Geiger, C.D. (2010). A note on the optimal sequence position for a rate-modifying activity under simple linear deterioration. *European Journal of Operational Research*, 201, 644–648.

Mosheiov, G., & Sidney, J.B. (2010). Scheduling a deteriorating maintenance activity on a single machine. *Journal of the Operational Research Society*, 61, 882–887.

Rustogi, K., & Strusevich, V.A. (2012). Single machine scheduling with general positional deterioration and rate-modifying maintenance. *Omega*, doi: 10.1016/j.omega.2011.12.007.

Wu, C.-C., Yin, Y, Cheng, S.-R. (2011). Some single-machine scheduling problems with a truncation learning effect. *Computers and Industrial Engineering*, 60, 790–795.

Yang, S.-J., & Yang, D.-L. (2010). Minimizing the makespan single-machine scheduling with aging effects and variable maintenance activities. *Omega*, 38, 528–533.

Zhao, C.-L., & Tang, H.-Y. (2010). Single machine scheduling with general job-dependent aging effect and maintenance activities to minimize makespan. *Applied Mathematical Modelling*, 34, 837–841.