Author’s Accepted Manuscript

www.elsevier.com/locate/omega

On single-machine scheduling with workload-dependent maintenance duration

Wenchang Luo, Feng Liu

PII: S0305-0483(16)30335-8
DOI: http://dx.doi.org/10.1016/j.omega.2016.06.008
Reference: OME1686

To appear in: *Omega*

Received date: 19 November 2015
Revised date: 2 April 2016
Accepted date: 20 June 2016

Cite this article as: Wenchang Luo and Feng Liu, On single-machine scheduling with workload-dependent maintenance duration, *Omega*, http://dx.doi.org/10.1016/j.omega.2016.06.008

On single-machine scheduling with workload-dependent maintenance duration

Wenchang Luo

Faculty of Science, Ningbo University, Ningbo 315211, PR China

Feng Liu *

School of Management Science and Engineering,

Dongbei University of Finance and Economics, Dalian 116025, PR China

Abstract

In this paper, we consider a single-machine scheduling problem with workload-dependent maintenance duration. The objective is to minimize the total weighted completion time. For the case where the maintenance duration is an arbitrarily non-decreasing function on the workload, we propose a $(2 + \varepsilon)$-approximation algorithm and a fully polynomial time approximation scheme, which extends the previous results presented by Xu et al. [Xu, D., Wan, L., Liu, A., & Yang, D-L. (2015). Single machine total completion time scheduling problem with workload-dependent maintenance duration. Omega, 52, 101-106]

Keywords. Scheduling; Maintenance; Workload; Approximation scheme

1 Introduction

Scheduling with machine maintenance has been extensively investigated in recent years. According to the maintenance duration, research literatures in this area can be classified into two classes, i.e., the fixed maintenance duration and the variable maintenance duration. For the fixed maintenance duration, it is assumed that the duration of a maintenance activity is a fixed time length. There are so many research articles which contribute this topic. We refer the readers to the latest survey paper [8]. For the variable maintenance duration, Kubzin and Strusevich [3] were the first pioneers that considered the scheduling problems with variable maintenance duration. They investigated the makespan minimization problem in a two-machine flow shop and a two-machine open shop. They showed that the open shop problem is polynomially solvable for quite general functions defining the maintenance duration while the flow shop problem is binary NP-hard and pseudo-polynomially solvable by dynamic programming. Furthermore, they presented a fully polynomial time approximation

*Corresponding author. liufengapollo@163.com.

scheme (FPTAS) and a fast 3/2-approximation algorithm for this problem. Xu, Sun, and Li [9] investigated the parallel machine scheduling problem with almost periodic maintenance and non-preemptive jobs to minimize makespan. Xu, Yin, and Li [10] considered two scheduling problems with machine maintenance under the assumption that the maintenance duration is an increasing linear function of the total processing time of the jobs that are processed after the machine's last maintenance activity. The first problem concerns parallel-machine scheduling to minimize the completion time of the last finished maintenance, where the length of the time interval between any two consecutive maintenance activities is between two given positive numbers. The second problem deals with single-machine scheduling to minimize the completion time of the last finished job, where the length of the time interval between any two consecutive maintenance activities is fixed. They proposed two approximation algorithms for the considered problems and analyzed their performances. Bock, Briskorn and Horbach [1] studied a single-machine scheduling problem that integrated machine deterioration, where the current maintenance state of the machine is determined by a maintenance level which drops by a certain, possibly job-dependent, amount when jobs are processed. A maintenance level of less than zero is associated with the machine's breakdown and is therefore forbidden. Consequently, maintenance activities that raise the maintenance level again may become necessary and have to be scheduled additionally. Two general types of maintenance activities are introduced. In the full maintenance case, maintenance activities are always executed until the machine has reached the maximum maintenance level. In contrast to this, the schedule in the partial maintenance case has to additionally determine the duration of maintenance activities. By combining both cases with regular objective functions such as minimization of maximum tardiness, minimization of the sum of completion times, or minimization of the number of tardy jobs, they analyzed the computational complexity of general and some specific cases. Luo, Cheng and Ji [7] addressed the problem of scheduling a maintenance activity and jobs on a single machine, where the maintenance activity must start before a given deadline and the maintenance duration increases with its starting time. They provided polynomial time algorithms to solve the problems to minimize the makespan, sum of completion times, maximum lateness, and number of tardy jobs.

In the very recent, Xu et al. [11] introduced a single-machine scheduling problem with workload-dependent maintenance duration. The objective is to minimize the total completion time. For the case where the derivation of the maintenance duration function is greater than or equal to 1, a polynomial time optimal algorithm was proposed. For the case where the derivation of the maintenance duration function is less than 1, a polynomial time approximation scheme was presented.

In this paper, we extend the problem proposed by Xu et al. [11]. We consider a more general objective function, i.e., the total weighted completion time. For the workload-dependent maintenance duration, we only assume that it is a nonnegative and non-decreasing function on the workload but can be computed in polynomial time. Compared to the assumption in [11], we do not require any derivation information and our assumption is more general. The formal problem statement can be described as follows.

Problem statement: Given a set of non-preemptive jobs $\mathcal{J} = \{J_1, J_2, \cdots, J_n\}$ to be processed on a single machine, where the machine is subject to a maintenance activity. All the jobs are available at time 0. For job J_i, the processing time is p_i and the weight is w_i, $i = 1, 2, \cdots, n$. For the maintenance activity, its starting time is S, which is known and prefixed. However, its duration D is a nonnegative and non-decreasing function $f(\cdot)$ on the machine's workload l before the maintenance activity, i.e., $D = f(l)$, where l is the sum of processing times of jobs scheduled before the maintenance activity and function $f(\cdot)$ can be computed in polynomial time. Let T_f denote the time for computing function $f(\cdot)$. For a given schedule, let C_i denote the completion time of job J_i, $i = 1, 2, \cdots, n$. The task is to find a schedule to minimize the total weighted completion time ($\sum_i w_i C_i$). Adopting the well-known three-field notation scheme proposed in [2] and following the similar notations as [11], we name this problem as $1, h_1, wldmt||\sum_i w_i C_i$, where "$h_1$" denotes that there is a maintenance activity on the machine (a hole in the planning horizon) and "$wldmt$" denotes that the maintenance duration is workload-dependent. Clearly, the $1, h_1, wldmt||\sum_i w_i C_i$ problem is NP-hard, since the special case $1, h_1, wldmt||\sum_i C_i$ is shown as NP-hard in Xu et al. [11].

2 A (2+ε)-approximation algorithm

To solve the $1, h_1, wldmt||\sum_i w_i C_i$ problem , we first split the interval $[0, S]$ into some subintervals, then in each subinterval we solve a minimization Knapsack problem to determine the jobs that scheduled before the maintenance activity and naturally we obtain the job sequences before and after the maintenance activity according to the weighted shortest processing time first (WSPT) rule. Finally, we choose the best schedule as our output solution.

The formal algorithm description is below.

Algorithm H_1:

Step 1: Given an $\varepsilon > 0$ and let $s_0 = \min_{i=1,2,\cdots,n} p_i$. Compute a series of

$$s_i = \max_{s_{i-1} \le t \le S} \{t | f(t) \le f(s_0)(1+\varepsilon)^i\}$$

by using binary search, $i = 1, 2, \cdots, r-1$, where $r = \lfloor \log_{1+\varepsilon} f(S) - \log_{1+\varepsilon} f(s_0) \rfloor + 1$. Split the interval $[0, S]$ into a series of subintervals $[0, s_0], [s_0, s_1], [s_1, s_2], \cdots, [s_{r-1}, s_r]$, where $s_r = S$.

Step 2: For subinterval $[0, s_0]$, we construct two schedules $\hat{\pi}^1$ and $\hat{\pi}^2$. For schedule $\hat{\pi}^1$, we put all the jobs after the maintenance activity according to the WSPT rule and for schedule $\hat{\pi}^2$ we put the job with minimal processing time and maximal weight before the maintenance activity and put the remaining jobs after the maintenance activity according to the WSPT rule. For each subinterval $[s_{i-1}, s_i]$, $i = 1, 2, \cdots, r$, we create an instance of minimization Knapsack problem by setting the knapsack size be s_i and each item associated each job with the processing time correspond to item size and weight correspond to item profit. By invoking an FPTAS which can be obtained by adapting an FPTAS for the classical maximization Knapsack problem (see, e.g., Kellerer, Pferschy and Pisinger [4]), we can determine the jobs that are scheduled before the maintenance activity, which are the items that were put into the knapsack. We use S_i^b to denote the job subset before the maintenance activity and S_i^a to denote the job subset after the maintenance activity. Then we construct schedule π_i as follows: first put the jobs in S_i^b before the maintenance activity according to the WSPT rule, then at time S start the maintenance activity with duration of $f(\sum_{i\in S_i^b} p_i)$ and finally schedule the jobs in S_i^a according to the WSPT rule. With the convenience, we denote π_i by $\pi_i = (S_i^b, MA, S_i^a)$, $i = 1, 2, \cdots, r$.

Step 3: From all the constructed schedules $\hat{\pi}^1, \hat{\pi}^2$, $\pi_i = (S_i^b, MA, S_i^a)$, $i = 1, 2, \cdots, r$, we choose the best as the output schedule and denote it as π, i.e., schedule π has the minimum total weighted completion time.

Theorem 1 *Algorithm H_1 is a $(2+\varepsilon)$-approximation algorithm for the problem $1, h_1, wldmt||\sum_i w_iC_i$ with the running time $O(n \log f(S)/\varepsilon^2 + T_f \log f(S) \log S/\varepsilon)$.*

Proof. Consider the optimal schedule π^*. We know that the sum of processing times of jobs scheduled before the maintenance activity in π^* must fall into one of the subintervals $[0, s_0]$, $[s_{i-1}, s_i]$, $i = 1, 2, \cdots, r$. If the sum of processing time of jobs scheduled before the maintenance activity in π^* falls into subinterval $[0, s_0]$, given $s_0 = \min_{i=1,2,\cdots,n} p_i$ there must be just two possible schedules for π^* which are the schedules $\hat{\pi}^1$ and $\hat{\pi}^2$ and we are done.

Next, we assume that the sum of processing times of jobs scheduled before the maintenance activity in π^* must fall into one of the subintervals $[s_{i-1}, s_i]$, $i = 1, 2, \cdots, r$.

Let S^b denote the job subset scheduled before the maintenance activity and S^a denote the job subset scheduled after the maintenance activity in π^*. Let $Z(\pi^*)$ denote the total weighted completion time of π^*. We have

$$Z(\pi^*) = Z(S^b) + Z(S^a) + (S + f(\sum_{i\in S^b} p_i))(\sum_{i\in S^a} w_i),$$

where $Z(S^b)$ denotes the total weighted completion time that we schedule the jobs in S^b on a single machine without maintenance according to WSPT rule and $Z(S^a)$ is defined as $Z(S^b)$.

Consider schedule π_i. Let $Z(\pi_i)$ denote the total weighted completion time of π_i. We have

$$Z(\pi_i) = Z(S_i^b) + Z(S_i^a) + (S + f(\sum_{i\in S_i^b} p_i))(\sum_{i\in S_i^a} w_i),$$

where $Z(S_i^b)$ and $Z(S_i^a)$ are defined as $Z(S^b)$ and $Z(S^a)$.

From the FPTAS of minimization Knapsack problem (Kellerer, Pferschy and Pisinger [4]), we have

$$(\sum_{i\in S_i^a} w_i) \le (1+\epsilon)(\sum_{i\in S^a} w_i).$$

From the split of subintervals, we have

$$f(\sum_{i\in S_i^b} p_i) \leq f(s_i) \leq (1+\varepsilon)f(s_{i-1}).$$

Clearly $f(\sum_{i\in S^b} p_i) \geq f(s_{i-1})$. Thus we have

$$f(\sum_{i\in S_i^b} p_i) \leq f(s_i) \leq (1+\varepsilon)f(\sum_{i\in S^b} p_i).$$

Finally, we obtain

$$\begin{aligned}(S+f(\sum_{i\in S_i^b} p_i))(\sum_{i\in S_i^a} w_i) &\leq (S+(1+\varepsilon)f(\textstyle\sum_{i\in S^b} p_i))(1+\varepsilon)(\sum_{i\in S^a} w_i)\\ &\leq (1+\varepsilon)^2(S+f(\textstyle\sum_{i\in S^b} p_i))(\sum_{i\in S^a} w_i).\end{aligned}$$

Because of $Z(\pi^*) \geq Z(\mathcal{J})$ and $Z(S_i^b) + Z(S_i^a) \leq Z(\mathcal{J})$ (where $Z(\mathcal{J})$ denotes the total weighted job completion time that we schedule the jobs in $\mathcal{J}$ on a single machine according to WSPT rule without maintenance), we achieve $Z(\pi_i) \leq (1+(1+\varepsilon)^2)Z(\pi^*)$. Slightly scaling the ε, we obtain $Z(\pi) \leq Z(\pi_i) \leq (2+\varepsilon)Z(\pi^*)$.

As for the running time of Algorithm H_1, in Step 1 it needs $O(T_f \log f(S) \log S/\log(1+\varepsilon))$ time to compute a series of $s_i, i = 1, 2, \cdots, r$ by using binary search, since $f(\cdot)$ is a non-decreasing function and its computing time is T_f. In Step 2 it totally invokes the FPTAS of minimization Knapsack problem (Kellerer, Pferschy and Pisinger [4]) at most $\log f(S)/\log(1+\varepsilon)$ times. For each invoking, it needs $O(n/\varepsilon)$ time (Kellerer, Pferschy and Pisinger [4]). Thus Algorithm H_1 runs in $O(n \log f(S)/\varepsilon^2 + T_f \log f(S) \log S/\varepsilon)$ time. □

3 An FPTAS

In this section, we focus on deriving an FPTAS for the $1, h_1, wldmt||\sum_i w_iC_i$ problem. To derive our FPTAS, we first consider a closely related scheduling problem $1, h_1||\sum_i w_iC_i$. The problem $1, h_1||\sum_i w_iC_i$, which was called as single-machine scheduling with an unavailable constraint to minimize the total weighted completion time, was first studied by Lee [6]. The only difference between problem $1, h_1||\sum_i w_iC_i$ and problem $1, h_1, wldmt||\sum_i w_iC_i$ is that the duration of the maintenance activity is fixed in the former but is variable in the latter. The problem $1, h_1||\sum_i w_iC_i$ has been well studied and admitted an FPTAS (Kellerer and Strusevich [5]). Next, we introduce our FPTAS for the $1, h_1, wldmt||\sum_i w_iC_i$ problem.

Algorithm H_2:

Step 1: Given an $\varepsilon > 0$ and let $s_0 = \min_{i=1,2,\cdots,n} p_i$. Compute a series of

$$s_i = \max_{s_{i-1}\leq t\leq S}\{t|f(t) \leq f(s_0)(1+\varepsilon)^i\}$$

by using binary search, $i = 1, 2, \cdots, r-1$, where $r = \lfloor \log_{1+\varepsilon} f(S) - \log_{1+\varepsilon} f(s_0)\rfloor + 1$. Split the interval $[0, S]$ into a series of subintervals $[0, s_0], [s_0, s_1], [s_1, s_2], \cdots, [s_{r-1}, s_r]$, where $s_r = S$.

Step 2: For subinterval $[0, s_0]$, we create two schedules $\hat{\pi}^1$ and $\hat{\pi}^2$. For schedule $\hat{\pi}^1$, we put all the jobs after the maintenance activity according to the WSPT rule and for schedule $\hat{\pi}^2$ we put the job with minimal processing time and maximal weight before the maintenance activity and put the remaining jobs after the maintenance according to the WSPT rule. For each subinterval $[s_{i-1}, s_i]$, $i = 1, 2, \cdots, r$, we construct an instance of problem $1, h_1||\sum_i w_iC_i$ by setting the the starting time of the maintenance activity be s_i and the duration of the maintenance activity be $(S - s_i + f(s_i))$. By invoking the FPTAS of the problem $1, h_1||\sum_i w_iC_i$ (Kellerer and Strusevich [5]), we can determine the jobs that scheduled before and after the maintenance activity. Again we use S_i^b and S_i^a to denote them. Then we construct schedule π_i as follows: first put the jobs in S_i^b before the maintenance activity according to the WSPT rule, then at time S start the maintenance activity with duration of $f(\sum_{i\in S_i^b} p_i)$ and finally schedule the jobs in S_i^a according to the WSPT rule. With the convenience, we denote π_i by $\pi_i = (S_i^b, MA, S_i^a)$, $i = 1, 2, \cdots, r$.

Step 3: From all the constructed schedules $\hat{\pi}^1, \hat{\pi}^2$, $\pi_i = (S_i^b, MA, S_i^a)$, $i = 1, 2, \cdots, r$, we choose the best as the output schedule and denote it as π, i.e., schedule π has the minimum total weighted completion time.

Theorem 2 *Algorithm H_2 is an FPTAS for problem $1, h_1, wldmt||\sum_i w_iC_i$ running in $O(n^4 \log f(S)/\varepsilon^3 + T_f \log f(S) \log S/\varepsilon)$ time.*

Proof. Consider an optimal schedule π^*. We know that the sum of processing times of jobs scheduled before the maintenance activity in π^* must fall into one of the subintervals $[0, s_0]$, $[s_{i-1}, s_i]$, $i = 1, 2, \cdots, r$. If the sum of processing time of jobs scheduled before the maintenance activity in π^* falls into subinterval $[0, s_0]$, given $s_0 = \min_{i=1,2,\cdots,n} p_i$ there must be just two possible schedules for π^* which are the schedules $\hat{\pi}^1$ and $\hat{\pi}^2$ and we are done.

Next, we assume that the sum of processing times of jobs scheduled before the maintenance activity in π^* must fall into one of the subintervals $[s_{i-1}, s_i]$, $i = 1, 2, \cdots, r$. Again let S^b denote the job subset scheduled before the maintenance activity and S^a denote the job subset scheduled after the maintenance activity in π^*. Define $Z(\pi^*)$ as the same in Section 2. We have

$$Z(\pi^*) = Z(S^b) + Z(S^a) + (S + f(\sum_{i\in S^b} p_i))(\sum_{i\in S^a} w_i),$$

where $Z(S^b)$ and $Z(S^a)$ are defined as the same in Section 2.

Consider schedule π_i. Define $Z(\pi_i)$ as the same in Section 2. We have

$$Z(\pi_i) = Z(S_i^b) + Z(S_i^a) + (S + f(\sum_{i\in S_i^b} p_i))(\sum_{i\in S_i^a} w_i),$$

where $Z(S_i^b)$ and $Z(S_i^a)$ are defined as the same in Section 2.

From the FPTAS of the problem $1, h_1||\sum_i w_iC_i$, since the schedule (S^b, MA, S^a) is a feasible solution for the problem $1, h_1||\sum_i w_iC_i$, we have

$$Z(S_i^b) + Z(S_i^a) + (s_i + (S - s_i + f(s_i)))(\sum_{i\in S_i^a} w_i)$$

$$\leq (1+\varepsilon)(Z(S^b)+Z(S^a)+(s_i+(S-s_i+f(s_i)))(\sum_{i\in S^a} w_i)),$$

i.e.,

$$Z(S_i^b)+Z(S_i^a)+(S+f(s_i)))(\sum_{i\in S_i^a} w_i)$$
$$\leq (1+\varepsilon)(Z(S^b)+Z(S^a)+(S+f(s_i)))(\sum_{i\in S^a} w_i)).$$

From the split of subintervals, we have

$$f(\sum_{i\in S_i^b} p_i) \leq f(s_i) \leq (1+\varepsilon)f(s_{i-1}).$$

Clearly $f(\sum_{i\in S^b} p_i) \geq f(s_{i-1})$. Thus we have

$$f(\sum_{i\in S_i^b} p_i) \leq f(s_i) \leq (1+\varepsilon)f(\sum_{i\in S^b} p_i).$$

Finally, we obtain

$$\begin{aligned}
& Z(S_i^b)+Z(S_i^a)+(S+f(\textstyle\sum_{i\in S_i^b} p_i))(\sum_{i\in S_i^a} w_i) \\
\leq\ & Z(S_i^b)+Z(S_i^a)+(S+f(s_i))(\textstyle\sum_{i\in S_i^a} w_i) \\
\leq\ & (1+\varepsilon)(Z(S^b)+Z(S^a)+(S+f(s_i))(\textstyle\sum_{i\in S^a} w_i)) \\
\leq\ & (1+\varepsilon)(Z(S^b)+Z(S^a)+(S+(1+\varepsilon)f(\textstyle\sum_{i\in S^b} p_i))(\sum_{i\in S^a} w_i)) \\
\leq\ & (1+\varepsilon)^2(Z(S^b)+Z(S^a)+(S+f(\textstyle\sum_{i\in S^b} p_i))(\sum_{i\in S^a} w_i)),
\end{aligned}$$

which implies

$$Z(\pi_i) \leq (1+\varepsilon)^2 Z(\pi^*).$$

Slightly scaling the ε, we obtain

$$Z(\pi) \leq Z(\pi_i) \leq (1+\varepsilon)Z(\pi^*).$$

Because the FPTAS of the problem $1, h_1||\sum_i w_iC_i$ provided by Kellerer and Strusevich [5] runs in $O(n^4/\varepsilon^2)$ time and we invoke it at most $\log f(S)/\ log(1+\varepsilon)$ times, the running time of Algorithm H_2 is $O(n^4 \log f(S)/\varepsilon^3 + T_f \log f(S) \log S/\varepsilon)$ time and thus Algorithm H_2 is an FPTAS (note that $f(\cdot)$ can be computed in polynomial time T_f by the assumption). $\square$

4 Concluding remarks

This paper investigates a single-machine scheduling problem with workload-dependent maintenance duration. The objective is to minimize the total weighted completion time. Under the assumption that the maintenance duration is an arbitrarily nonnegative and nondecreasing function on the workload and the duration function can be computed in polynomial time, we propose a $(2+\varepsilon)$-approximation algorithm and a fully polynomial time approximation scheme, which extends the previous results in [11].

Acknowlegment

The research was supported by National Natural Science Foundation of China (71502026 and 71501027), China Scholarship Council (201408330402), and the K.C. Wong Magna Foundation of Ningbo University.

References

[1] Bock, S., Briskorn, D., & Horbach, A. (2012). Scheduling flexible maintenance activities subject to job-dependent machine deterioration. Journal of Scheduling, 15, 565-578.

[2] Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics, 5, 236-287.

[3] Kubzin, M. A., & Strusevich, V. A. (2006). Planning machine maintenance in two-machine shop scheduling. Operations Research, 54(4), 789-800.

[4] Kellerer, H., Pferschy, U., & Pisinger, D. (2004). Knapsack Problems. Springer, Berlin.

[5] Kellerer, H., & Strusevich, V. A. (2010). Fully polynomial approximation schemes for a symmetric quadratic knapsack problem and its scheduling applications. Algorithmica, 57(4), 769-795.

[6] Lee, C. Y. (1996). Machine scheduling with an availability constraint. Journal of Global Optimization, 9(3-4), 395-416.

[7] Luo, W., Cheng, T. C. E., & Ji, M. (2015). Single-machine scheduling with a variable maintenance activity. Computers & Industrial Engineering, 79, 168-174.

[8] Ma, Y., Chu, C., & Zuo, C. (2010). A survey of scheduling with deterministic machine availability constraints. Computers & Industrial Engineering, 58(2), 199-211.

[9] Xu, D., Sun, K., & Li, H. (2008). Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan. Computers & Operations Research, 35, 1344-1349.

[10] Xu, D., Yin, Y., & Li, H. (2010). Scheduling jobs under increasing linear machine maintenance time. Journal of Scheduling, 13(4), 443-449.

[11] Xu, D., Wan, L., Liu, A., & Yang, D-L. (2015). Single machine total completion time scheduling problem with workload-dependent maintenance duration. Omega, 52, 101-106.