A Single Machine Scheduling Problem with Time Slot Costs

Weiya Zhong and Xiaolei Liu

Abstract. In this paper, we consider a single machine scheduling problem in which processing a job will incur some cost which is relevant with the time slots occupied by the job. The objective is to minimize the makespan plus the total time slot costs. We prove that the problem is strongly NP-hard and analyze a special case with non-increasing time slot costs.

1 Introduction and Problem Description

In traditional scheduling problems, the objective is to minimize the maximal completion time, the total weighted completion time, the number of tardy jobs, etc. However, in many practical scheduling problems, different costs associated with different time slots may happen.

In this paper, we study a single machine scheduling problem as follows: There are n jobs with processing times of $\{p_1, p_2, \ldots, p_n\}$ to be scheduled on a single machine with planning horizon of K units of time. No preemption is allowed. To ensure the feasibility, we assume that the total processing time $P = p_1 + p_2 + \ldots + p_n \leq K$. We also assume that all the parameters are integers. The cost of using the $k-$th time slot on the machine is π_k, $k = 1, 2, \ldots, K$ (the $k-$th time slot means time interval $[k-1, k]$). Suppose job j is scheduled in time interval $[s_j, s_j + p_j]$, then the time slot cost of job j is $\sum_{l=s_j+1}^{s_j+p_j} \pi_l$. The objective is to obtain a schedule such that the sum of the maximal job completion time (C_{max}) and the total time slot costs is minimized. The problem is denoted as $1 \mid slotcost \mid C_{max} + \sum_{j=1}^{n} \sum_{l=s_j+1}^{s_j+p_j} \pi_l$.

Weiya Zhong · Xiaolei Liu
Department of Mathematics Shanghai University, 99 Shangda Road,
BaoShan District, Shanghai
e-mail: wyzhong@shu.edu.cn, liuxiaoleiangel@yahoo.com.cn

Z. Qian et al. (Eds.): Recent Advances in CSIE 2011, LNEE 126, pp. 677–681.
springerlink.com

The variable time slot costs mean the status of a machine may change with time, which means our research may be related to two classes of scheduling models–scheduling with machine availability constraints[3, 4, 5]and scheduling with learning effects[6, 7, 8, 9]. In scheduling models with machine availability constraints, a machine can only process jobs when it is available. Thus, these problems can be included as an extreme case of our model where there are only two time slot costs–a normal one and a extremely high one. In scheduling models with learning effects, the processing time of a job is determined by the time when it is started and the cost of processing a job does not change with the machine speed. Wan and Qi[1] also study similar scheduling models with different objective functions.

This paper is organized as follows. In section 2, we prove the general problem is strongly NP-hard. In section 3, we analyze a special case. Section 4 concludes this paper.

2 Time Computational Complexity

In this section, we prove that the general problem is strongly NP-hard by reduction from 3-Partition problem[2].

3-Partition Problem. Given positive integers $a_1,\dots,a_{3q}$ and b with $\frac{b}{4} < a_j < \frac{b}{2}$, $j = 1,2,\dots,3q$ and $\sum_{j=1}^{3q} a_j = q \cdot b$, do there exist q pairwise disjoint three element subsets $S_i \subset \{1,2\dots,3q\}$ such that $\sum_{j\in S_i} a_j = b, i = 1,2,\dots,q$?

Theorem 1. *The problem* $1 \mid slotcost \mid C_{max} + \sum_{j=1}^{n} \sum_{l=s_j+1}^{s_j+p_j} \pi_l$ *is strongly NP-hard.*

Proof. The proof is done via a reduction from 3-partition problem. Given an instance of the 3-partition problem, we construct an instance of the decision version of $1 \mid slotcost \mid C_{max} + \sum_{j=1}^{n} \sum_{l=s_j+1}^{s_j+p_j} \pi_l$ as follows. Let there be $n = 3q$ jobs. The processing time of job j is a_j, $j = 1,2,\dots,3q$. The time slot cost function is

$$\pi_k = \begin{cases} 1, & \text{for } k = (t-1)b+t,\dots, \\ & \quad tb+t-1 (t = 1,2,\dots,q), \\ M, & \text{for } k = tb+t (t = 1,2,\dots,q-1) \\ & \quad \text{and } k \geq qb+q. \end{cases}$$

where M is a large integer, for example, $M = Z+1$, $Z = qb+qb-1$. The question asks whether there exists a schedule such that C_{max} is no more than the threshold value Z. Clearly, this transformation can be done in $O(nb)$ time, thus it is a pseudo-polynomial time transformation.

1. If the 3-partition problem has a feasible solution, we can schedule each group of three jobs in one of the q intervals, leaving all the interval $[tb+t-1, tb+t](t = 1,2\dots,q)$ with high slot costs idle. It can be verified that the objective value of this schedule is no more than Z.

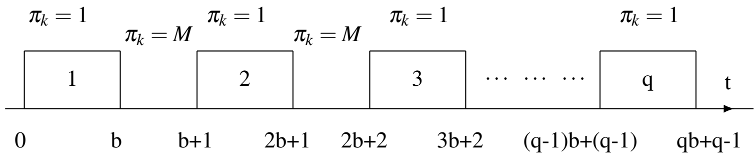

Fig. 1 Time slot costs pattern in the constructed instance

2. Suppose that there is a schedule with objective value no more than the threshold value Z. Then in this schedule, we have:

CLAIM 1: There should be no job scheduled in intervals $[tb+t-1, tb+t](t = 1,\ldots,q)$.

This is because the total cost of a schedule is larger(at least M)than the threshold value Z even if one unit time of a job is scheduled in intervals $[tb+t-1, tb+t](t = 1,2,\ldots,q)$. Hence, all the jobs must be scheduled in intervals $[(t-1)b+t-1, tb+t-1](t = 1,2,\ldots,q)$.

CLAIM 2: There must be exactly three jobs scheduled in each interval $[(t-1)b+t-1, tb+t-1](t = 1,2,\ldots,q)$. In fact, it is easy to show that in a schedule with objective value no more than Z :

a No more than three jobs can be scheduled in an interval $[(t-1)b+t-1, tb+t-1], t = 1,2,\ldots,q$.

Note that the size of job j is a_j, $\frac{b}{4} < a_j < \frac{b}{2}$, $j = 1,\ldots,3q$. If there are more than three jobs, then their total processing time will be greater than $4 \cdot \frac{b}{4} = b$. Clearly, they can't be scheduled in just one of the intervals $[(t-1)b+t-1, tb+t-1](t = 1,\ldots,q)$ with length of b. Thus, at least part of a job is scheduled in some intervals $[tb+t-1, tb+t], t \in \{1,2,\ldots,q\}$ and the total cost of the schedule is at least $M > Z$.

b No less than three jobs can be scheduled in an interval $[(t-1)b+t-1, tb+t-1], t = 1,2,\ldots,q$.

Note that the total number of jobs is $3q$ and the total number of intervals is q. If there are only one or two jobs scheduled in one of the interval $[(t-1)b+t-1, tb+t-1](t = 1,2,\ldots,q)$, then there must be more than three jobs scheduled in some interval $[(t-1)b+t-1, tb+t-1], t \in \{1,2,\ldots,q\}$, and this violates(a).

By Claim 1 and Claim 2, we know that there must be three jobs exactly scheduled in each of the intervals $[(t-1)b+t-1, tb+t-1], t = 1,2,\ldots,q$ since no preemption is allowed. This means 3-partition problem has a solution.

3 A Special Case with Non-increasing Time Slot Costs

In this section, we consider the special case of $1 \mid slotcost \mid C_{max} + \sum_{j=1}^{n} \sum_{l=s_j+1}^{s_j+p_j} \pi_l$ with non-increasing time slot costs.

It is necessary to consider the input formats of the time slot costs. The time slot costs π_k can be represented in two different ways. In the first case, the time slot costs are represented as a series of K discrete values. In such case, the input size of the time slot cost is in $O(K)$. Consequently, an algorithm with a running time bounded by polynomial function of n and K is regarded as a polynomial-time algorithm. In the second case, the time slot costs are given as a function of k with a closed form and each π_k can be calculated in a constant time. For example, $\pi_k = \eta^k \pi_0$, where π_0 is a constant. For such a representation of the time slot costs, its input size is $O(1)$. As a result, an algorithm with a running time bounded by a polynomial function of n and K is no longer a polynomial-time algorithm. Instead, a polynomial-time algorithm should have a running time bounded by a polynomial function of n and $\log K$.

Suppose in a feasible schedule S, $C_{max}(S)$ is the maximal completion time. It is easy to see that the job should be arranged in time interval $[C_{max}(S) - P, C_{max}(S)]$ and time interval $[0, C_{max}(S) - P]$ is idle, since the time slot costs are non-increasing. Let $x = C_{max}(S) - P$, then our problem is to

$$\begin{array}{c}\text{minimize } x + P + \sum_{l=x+1}^{x+P} \pi_l \\ \text{s.t. } x \in \{0, 1, \cdots, K - P\}\end{array}$$

We can get the optimal solution by computing $K - P + 1$ different values of $x + P + \sum_{l=x+1}^{x+P} \pi_l$ and select the minimal one. We can see that the problem can be solved in $O(K^2)$ time. If the time slot costs are represented as a series of K discrete values, the problem can be solved in polynomial time. If the time slot costs are given as a function of k with closed form and each π_k can be calculated in a constant time, the problem can be solved in pseudo-polynomial time.

4 Conclusion

In this paper, we study a single machine scheduling problem with variable time slot costs and the objective is to minimize the sum of maximal job completion time and total time slot costs. We prove that the general problem is strongly NP-hard and analyze a special case. For future research, we are interested related problems in parallel machine environment and flowshop environment because of it practical application. Besides, a machine speed change may also lead to a cost change for processing a job. We believe there is a strong need of research in the field.

References

1. Wan, G., Qi, X.: Scheduling with variable time slot costs. Naval Research Logistics 57, 159–171 (2010)
2. Garey, M.R., Johnson, D.S.: Computers and intractability: A guide to the theory of NP-completeness. Freeman, San Francisco (1979)
3. Lee, C.-Y.: Machine scheduling with availability constraints. In: Leung, J.Y.-T. (ed.) Handbook of scheduling: Algorithms, Models and Performance Analysis. ch. 22. CRC Press, Boca Raton (2004)

4. Lee, C.-Y., Liman, S.D.: Single-machine flow-time scheduling with scheduled maintenance. Acta Inform 29, 375–382 (1992)
5. Schmidt, G.: Scheduling with limited machine availability. Eur. J. Oper. Res. 121, 1–15 (2000)
6. Biskup, D.: A state-of-the-art review on scheduling with learning effects. Eur. J. Oper. Res. 188, 315–329 (2008)
7. Cheng, T.C.E., Ding, Q., Lin, B.M.T.: A concise survey of scheduling with time-dependent processing times. Eur. J. Oper. Res. 152, 1–13 (2004)
8. Janiak, A., Rudek, R.: The learning effect: Getting to the core of the problem. Inform. Process Lett. 103, 183–187 (2007)
9. Lin, B.M.T.: Complexity results for single-machine scheduling with positional learning effects. J. Oper. Res. Soc. 58, 1099–1102 (2007)