

Short Communication

An improved delayed-start LPT algorithm for a partition problem on two identical parallel machines

Christos Koulamas ^{*}, George J. Kyparisis*Department of Decision Sciences and Information Systems, Florida International University, Miami, FL 33199, USA*

Received 5 October 2006; accepted 10 April 2007

Available online 19 April 2007

Abstract

We propose an off-line delayed-start LPT algorithm that sequences the first (longest) 5 jobs optimally and the remaining jobs according to the LPT principle on two identical parallel machines. We show that this algorithm has a sharper tight worst-case ratio bound than the traditional LPT algorithm for the sum of squares of machine completion times minimization problem.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Scheduling; Parallel machines; LPT; Approximation algorithms

1. Introduction

We consider off-line scheduling on two identical parallel machines with the objective of minimizing the sum of squares of the machine completion times. We assume that there are n jobs J_j with processing times p_j , $j = 1, \dots, n$ to be scheduled non-preemptively on two identical parallel machines. The completion time of job J_j is denoted as C_j , $j = 1, \dots, n$. The machine completion time of machine m_i , $i = 1, 2$, defined as the completion time of the last job scheduled on it, is denoted as C_{m_i} , $i = 1, 2$. We denote our problem as the $P2||\sum_{i=1}^2(C_{m_i})^2$ problem. For any heuristic algorithm A, let C_{\max}^A and C_{\min}^A denote the quantities obtained using algorithm A, where $C_{\max} = \max_{i=1,2}\{C_{m_i}\}$, $C_{\min} = \min_{i=1,2}\{C_{m_i}\}$, and let C_{\max}^* and C_{\min}^* denote the corresponding optimal quantities. We define the (relative) worst-case ratio bound of algorithm A for the $P2||\sum_{i=1}^2(C_{m_i})^2$ problem as

$$\rho_{\text{squares}}^A = \text{l.u.b.} \left\{ \frac{[(C_{\max}^A)^2 + (C_{\min}^A)^2] - [(C_{\max}^*)^2 + (C_{\min}^*)^2]}{(C_{\max}^*)^2 + (C_{\min}^*)^2} \right\}. \quad (1)$$

Graham (1969) proposed the longest processing time (LPT) algorithm for the $Pm||C_{\max}$ problem (the makespan minimization problem on m identical parallel machines). The LPT algorithm sorts all jobs in the

^{*} Corresponding author. Tel.: +1 305 348 3309; fax: +1 305 348 4126.

E-mail address: koulamas@fiu.edu (C. Koulamas).

non-increasing order of their processing times and assigns the next job in the list to the earliest available machine with ties broken in favor of the lowest numbered machine. Chandra and Wong (1975) used the LPT algorithm for the $Pm||\sum_{i=1}^m(C_{m_i})^2$ problem with $\rho_{\text{squares}}^{\text{LPT}} = \frac{1}{24}$; for $m = 2$ they obtained the sharper bound of approximately 0.0285. Leung and Wei (1995) improved the results of Chandra and Wong (1975) when $m > 2$.

The motivation for our paper is the observation that in a two identical parallel machines setting, the LPT algorithm is optimal for problem instances with up to 4 jobs and its worst-case performance occurs in problem instances with exactly 5 jobs. This observation implies that the worst-case performance of the LPT algorithm can be improved by scheduling the first (longest) 5 jobs optimally and then implementing the LPT rule for scheduling (heuristically) the remaining jobs.

In the next section, we present our delayed-start LPT algorithm A (with $O(n \log n + c)$ complexity, where c is a constant), that sequences the first (longest) 5 jobs optimally and the remaining jobs according to the LPT principle. We show that algorithm A has a sharper tight worst-case ratio bound than the traditional LPT algorithm for the $P2||\sum_{i=1}^2(C_{m_i})^2$ problem. More precisely, we show that $\rho_{\text{squares}}^A = \frac{1}{49}$ which improves the bound of 0.0285 of Chandra and Wong (1975).

An alternative theoretical approach for obtaining approximate solutions for the $Pm||\sum_{i=1}^m(C_{m_i})^2$ problem is to construct polynomial time approximation schemes (PTAS). Alon et al. (1997) constructed a PTAS for every fixed $p \geq 1$ for the $Pm||\sum_{i=1}^m(C_{m_i})^p$ problem which includes the $Pm||\sum_{i=1}^m(C_{m_i})^2$ problem as a special case when $p = 2$. However, PTAS are generally hard to implement from a practical standpoint.

We close this section by surveying additional related literature. He et al. (2000) proposed linear algorithms with sharper worst-case bounds than the LPT algorithm for the $P2||C_{\max}$ problem. However, their algorithms are hard to extend to the $P2||\sum_{i=1}^2(C_{m_i})^2$ problem. Several authors considered semi-online algorithms when it can be assumed that the jobs arrive in the LPT order but their actual processing times are unknown. We mention the work of Tan et al. (2005) for the semi-online version of the $P2||\sum_{i=1}^2(C_{m_i})^2$ problem.

2. Worst-case bound for an off-line delayed-start LPT algorithm A

The proposed algorithm A can be summarized as follows:

Algorithm A

Step 1: Sort all jobs $J_j, j = 1, \dots, n$, in the non-increasing order of their p_j values. Let S denote the resulting list (sequence) (without loss of generality, we assume that $S = \{1, 2, \dots, n\}$).

Step 2: If $n \leq 4$, then schedule all jobs according to the LPT rule.

If $n \geq 5$, then schedule the first (longest) 5 jobs in S optimally followed by the remaining jobs scheduled according to the LPT rule.

The running time of algorithm A is $O(n \log n + c)$, where $c = O(5)$ denotes the constant running time needed to sequence the first 5 jobs optimally. Actually, the comparison of the six schedules $[\{1\}, \{2, 3, 4, 5\}]$, $[\{1, 2\}, \{3, 4, 5\}]$, $[\{1, 3\}, \{2, 4, 5\}]$, $[\{1, 4\}, \{2, 3, 5\}]$, $[\{1, 5\}, \{2, 3, 4\}]$ and $[\{2, 3\}, \{1, 4, 5\}]$ respectively suffices to determine the optimal solution when there are only 5 jobs; the numbers in the first (second) set of brackets $\{\}$ denote the job allocation to m_1 (m_2) for each schedule.

Denote by $C_{\max}^A(k)$ and $C_{\min}^A(k)$ the maximum and the minimum machine completion time, respectively, after job $J_k, k = 1, \dots, n$, has been scheduled by algorithm A. Also, let $C_{\max}^*(k)$ and $C_{\min}^*(k)$ denote the corresponding optimal quantities. The quantity $\rho_{\text{squares}}^A(k)$ is defined analogously to (1) for a problem containing jobs $J_j, j = 1, \dots, k$. Clearly, $C_{\max}^A = C_{\max}^A(n)$, $C_{\max}^* = C_{\max}^*(n)$, $\rho_{\text{squares}}^A = \rho_{\text{squares}}^A(n)$, and so on for the other similar quantities. Denote the sum of processing times of the first k jobs in the LPT order as $P(k) = \sum_{j=1}^k p_j$. Then, $C_{\max}^*(k) + C_{\min}^*(k) = C_{\max}^A(k) + C_{\min}^A(k) = P(k) = \sum_{j=1}^k p_j$, which implies that

$$C_{\max}^A(k) - C_{\max}^*(k) = [P(k) - C_{\min}^A(k)] - [P(k) - C_{\min}^*(k)] = C_{\min}^*(k) - C_{\min}^A(k). \tag{2}$$

Furthermore,

$$\begin{aligned} \sum_{i=1}^2 [C_{m_i}^A(k)]^2 &= [C_{\max}^A(k)]^2 + [C_{\min}^A(k)]^2 = \frac{1}{2} [C_{\max}^A(k) - C_{\min}^A(k)]^2 + \frac{1}{2} [C_{\max}^A(k) + C_{\min}^A(k)]^2 \\ &= \frac{1}{2} [C_{\max}^A(k) - C_{\min}^A(k)]^2 + \frac{1}{2} [P(k)]^2 = \frac{1}{2} [2C_{\max}^A(k) - P(k)]^2 + \frac{1}{2} [P(k)]^2. \end{aligned} \tag{3}$$

Expression (3) is also valid when $C_{\max}^*(k)$, $C_{\min}^*(k)$ are substituted for $C_{\max}^A(k)$, $C_{\min}^A(k)$, respectively. The main result of this paper is stated next.

Theorem 1. *The worst-case ratio bound for algorithm A satisfies*

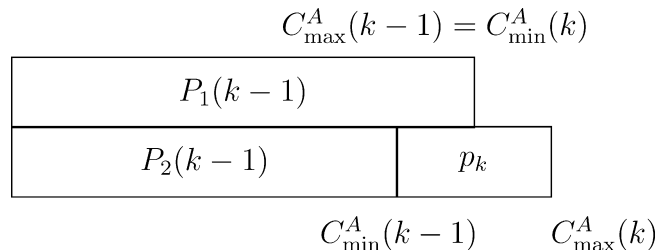
$$\rho_{\text{squares}}^A \leq \frac{1}{49} \tag{4}$$

and this bound is tight.

Proof. Each time algorithm A appends a new job J_k ($k = 6, \dots, n$) to the incumbent partial schedule (which is optimal when it contains up to 5 jobs) there are two possibilities: job J_k either becomes the makespan determining job or not. \square

Case 1. Job J_k becomes the makespan determining job, that is $C_{\max}^A(k) > C_{\max}^A(k - 1)$ for some $k, k = 6, \dots, n$.

Let $P_i(k)$, $i = 1, 2$, denote the sum of processing times of those jobs among the first k jobs in the LPT order that are scheduled on machine m_i by algorithm A as shown in the next figure.



Define

$$\Delta^A(k) = C_{\max}^A(k) - C_{\min}^A(k), \tag{5}$$

$$\Delta^*(k) = C_{\max}^*(k) - C_{\min}^*(k). \tag{6}$$

Clearly, $P(k) = P_1(k - 1) + P_2(k - 1) + p_k$. Observe that

$$\Delta^A(k) = P_2(k - 1) + p_k - P_1(k - 1) \leq p_k \tag{7}$$

as shown in the above figure, in which, without loss of generality, we have assumed that $P_2(k - 1) \leq P_1(k - 1)$. Denote by $n_i(k)$, $i = 1, 2$, the number of jobs among the first k jobs in the LPT order scheduled on machine m_i by algorithm A.

By combining $\sum_{i=1}^2 [C_{m_i}^*(k)]^2 \geq \frac{1}{2} [P(k)]^2$ with (3), (5) and (6) we obtain

$$\begin{aligned} \rho_{\text{squares}}^A(k) &= \text{l.u.b.} \left\{ \frac{\sum_{i=1}^2 [C_{m_i}^A(k)]^2 - \sum_{i=1}^2 [C_{m_i}^*(k)]^2}{\sum_{i=1}^2 [C_{m_i}^*(k)]^2} \right\} = \text{l.u.b.} \left\{ \frac{\frac{1}{2} [\Delta^A(k)]^2 - \frac{1}{2} [\Delta^*(k)]^2}{\sum_{i=1}^2 [C_{m_i}^*(k)]^2} \right\} \\ &\leq \text{l.u.b.} \left\{ \left[\frac{\Delta^A(k)}{P(k)} \right]^2 \right\}. \end{aligned} \tag{8}$$

The following subcases must be considered.

Subcase 1-1. Let $k \geq 7$. Then, since $p_1 \geq \dots \geq p_k$, $P(k) = \sum_{j=1}^k p_j \geq kp_k \geq 7p_k$.

Subcase 1-2. Let $k = 6$ and $n_2(k - 1) \geq 3$. Then, $P_1(k - 1) \geq P_2(k - 1) \geq 3p_k$, and thus $P(k) = P_1(k - 1) + P_2(k - 1) + p_k \geq 7p_k$.

Subcase 1-3. Let $k = 6$ and $n_2(k - 1) \leq 2$. Then, $n_1(k - 1) \geq 3$, that is $P_1(k - 1) \geq 3p_k$, and thus $P(k) = P_1(k - 1) + P_2(k - 1) + p_k = 2P_1(k - 1) + \Delta^A(k) \geq 6p_k + \Delta^A(k) \geq 7\Delta^A(k)$.

In Subcases 1-1 and 1-2 $P(k) \geq 7p_k$, which combined with (7) yields

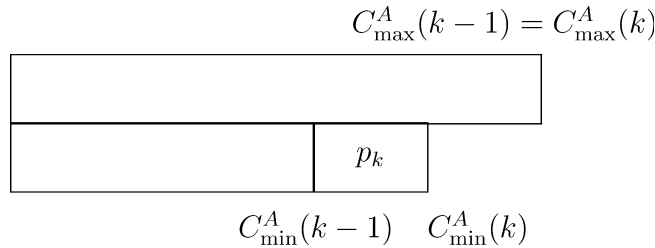
$$\frac{\Delta^A(k)}{P(k)} \leq \frac{p_k}{7p_k} = \frac{1}{7}. \tag{9}$$

In Subcase 1-3 $P(k) \geq 7\Delta^A(k)$, therefore

$$\frac{\Delta^A(k)}{P(k)} \leq \frac{\Delta^A(k)}{7\Delta^A(k)} = \frac{1}{7}. \tag{10}$$

The combination of inequalities (8)–(10) proves inequality (4) for Case 1.

Case 2. Job J_k does not become the makespan determining job, that is $C_{\max}^A(k) = C_{\max}^A(k - 1)$ for some k , $k = 6, \dots, n$.



In this case we first show that the following inequality holds for $k = 6, \dots, n$

$$\rho_{\text{squares}}^A(k) \leq \rho_{\text{squares}}^A(k - 1). \tag{11}$$

By utilizing expression (8), we can write

$$\rho_{\text{squares}}^A(k - 1) = \text{l.u.b.} \left\{ \frac{[C_{\max}^A(k - 1)]^2 + [C_{\min}^A(k - 1)]^2}{[C_{\max}^*(k - 1)]^2 + [C_{\min}^*(k - 1)]^2} - 1 \right\} \tag{12}$$

and

$$\rho_{\text{squares}}^A(k) = \text{l.u.b.} \left\{ \frac{[C_{\max}^A(k)]^2 + [C_{\min}^A(k)]^2}{[C_{\max}^*(k)]^2 + [C_{\min}^*(k)]^2} - 1 \right\}. \tag{13}$$

Our assumption of $C_{\max}^A(k) = C_{\max}^A(k - 1)$ implies that $C_{\min}^A(k) = C_{\min}^A(k - 1) + p_k$ and thus expression (13) can be written as

$$\rho_{\text{squares}}^A(k) = \text{l.u.b.} \left\{ \frac{[C_{\max}^A(k - 1)]^2 + [C_{\min}^A(k - 1) + p_k]^2}{[C_{\max}^*(k)]^2 + [C_{\min}^*(k)]^2} - 1 \right\}. \tag{14}$$

We show in the sequel that

$$[C_{\max}^*(k)]^2 + [C_{\min}^*(k)]^2 = [C_{\max}^*(k-1) + \epsilon]^2 + [C_{\min}^*(k-1) + p_k - \epsilon]^2, \tag{15}$$

where $\epsilon \geq 0$ depends on k . Define the quantities

$$\alpha = C_{\max}^A(k-1), \beta = C_{\min}^A(k-1), \gamma = C_{\max}^*(k-1), \delta = C_{\min}^*(k-1). \tag{16}$$

By combining expressions (12), (14), (15) and (16), inequality (11) can be written as

$$\text{l.u.b.} \left\{ \frac{(\alpha)^2 + (\beta + p_k)^2}{(\gamma + \epsilon)^2 + (\delta + p_k - \epsilon)^2} - 1 \right\} \leq \text{l.u.b.} \left\{ \frac{(\alpha)^2 + (\beta)^2}{(\gamma)^2 + (\delta)^2} - 1 \right\}. \tag{17}$$

Define the quantities

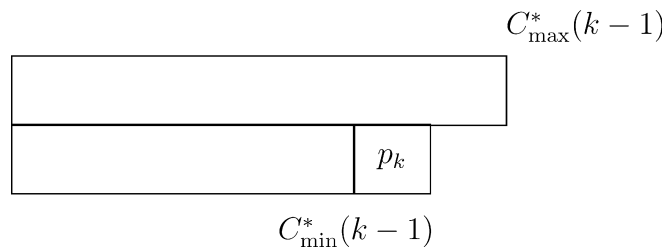
$$a = (\alpha)^2 + (\beta)^2, \quad b = (\gamma)^2 + (\delta)^2, \quad c = 2p_k\beta + p_k^2, \quad d = 2p_k\delta + p_k^2 + 2\epsilon(\gamma - \delta - p_k + \epsilon). \tag{18}$$

Using the expression in (18), inequality (17) can be written as

$$\text{l.u.b.} \left\{ \frac{a + c}{b + d} - 1 \right\} \leq \text{l.u.b.} \left\{ \frac{a}{b} - 1 \right\}. \tag{19}$$

Since $a \geq b > 0$, inequality (19) (which in turn leads to (11)) will be true if $d \geq c > 0$. In order to prove $d \geq c > 0$, we consider two subcases:

Subcase 2-1. $C_{\max}^*(k-1) \geq C_{\min}^*(k-1) + p_k$.



If we add job J_k to the optimal schedule for the first $k-1$ jobs by placing it on the machine with the minimum machine completion time, then the resulting schedule will still be feasible because $C_{\min}^*(k-1) + p_k \leq C_{\max}^*(k-1)$. Thus, $C_{\max}^*(k) \leq C_{\max}^*(k-1)$, and, because of $C_{\max}^*(k) \geq C_{\max}^*(k-1)$, we obtain

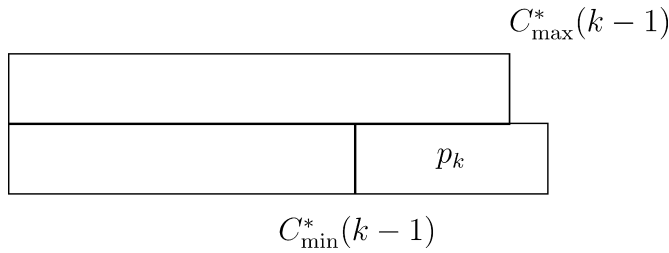
$$C_{\max}^*(k) = C_{\max}^*(k-1). \tag{20}$$

Moreover, equality (20) implies that

$$C_{\min}^*(k) = C_{\min}^*(k-1) + p_k. \tag{21}$$

The combination of (20) and (21) yields (15) for Subcase 2-1 with $\epsilon = 0$. Therefore, by (18), $c = 2p_k\beta + p_k^2$ and $d = 2p_k\delta + p_k^2$. This implies that $d \geq c > 0$ for Subcase 2-1.

Subcase 2-2. $C_{\max}^*(k - 1) < C_{\min}^*(k - 1) + p_k$.



In this subcase $C_{\max}^*(k - 1) \leq C_{\min}^*(k) \leq C_{\max}^*(k) \leq C_{\min}^*(k - 1) + p_k$. We define the quantity ϵ as

$$\epsilon = C_{\min}^*(k) - C_{\max}^*(k - 1) \geq 0. \tag{22}$$

By combining (22) with $C_{\max}^*(k) + C_{\min}^*(k) = C_{\max}^*(k - 1) + C_{\min}^*(k - 1) + p_k$, we obtain

$$C_{\max}^*(k) = C_{\min}^*(k - 1) + p_k - \epsilon. \tag{23}$$

The combination of (22) and (23) yields (15) for Subcase 2-2 with ϵ defined by (22). Since $C_{\max}^*(k) \geq C_{\min}^*(k)$ and $C_{\max}^*(k - 1) \geq C_{\min}^*(k - 1)$, equalities (22) and (23) imply that

$$0 \leq 2\epsilon \leq p_k, \tag{24}$$

for Subcase 2-2. Since $C_{\max}^A(k) = C_{\max}^A(k - 1)$, it is clear from the figure depicting the general Case 2 that

$$2C_{\min}^A(k - 1) + p_k \leq P(k - 1) = C_{\max}^*(k - 1) + C_{\min}^*(k - 1),$$

which implies that

$$0 \leq p_k - (\gamma - \delta) \leq 2\delta - 2\beta. \tag{25}$$

The quantities defined in (18) imply that the condition $d \geq c > 0$ is equivalent to

$$d - c = 2p_k\delta + p_k^2 + 2\epsilon(\gamma - \delta - p_k + \epsilon) - 2p_k\beta - p_k^2 = p_k(2\delta - 2\beta) - 2\epsilon[p_k - (\gamma - \delta)] + 2(\epsilon)^2 \geq 0. \tag{26}$$

The combination of inequalities (24) and (25) yields inequality (26), that is $d \geq c > 0$ for Subcase 2-2 as well.

In order to complete the proof for Case 2 we argue as follows. Since algorithm A schedules the first (longest) 5 jobs optimally, we only need to consider worst-case ratio bounds for problems with 6 or more jobs. Since the worst-case ratio bound in (11) is a nonincreasing function of k , inequality (4) (obtained for $k \geq 6$ for Case 1) is valid for Case 2 as well.

The following problem instance with $n = 6$ jobs and with job processing times p_j in the LPT list given as $\{3, 3, 2, 2, 2, 2\}$ demonstrates the tightness of our bound. The algorithm A schedule and the optimal schedule are depicted below.

3	3	2
2	2	2

Algorithm A schedule

3	2	2
3	2	2

Optimal schedule

It is easy to see that $C_{\max}^* = C_{\min}^* = 7$, $C_{\max}^A = 8$, $C_{\min}^A = 6$; therefore, $\rho_{\text{squares}}^A = \frac{1}{49}$.

3. Concluding remarks

Since the bound in Theorem 1 is a decreasing function of k , algorithm A can be made more accurate (with a sharper worst-case bound) at the expense of the additional computational effort needed to sequence a larger

job set optimally. However, this approach cannot be extended indefinitely since the required computational effort for sequencing jobs optimally quickly becomes prohibitive due to the well-known NP-hardness of the $P2||\sum_{i=1}^2(C_{m_i})^2$ problem.

References

- Alon, N., Azar, Y., Woeginger, G.J., Yadid, T., 1997. Approximation schemes for scheduling. In: Proceedings of the 8th Symposium on Discrete Algorithms, pp. 493–500.
- Chandra, A.K., Wong, C.K., 1975. Worst-case analysis of a placement algorithm related to storage allocation. *SIAM Journal of Computing* 4, 249–263.
- Graham, R.L., 1969. Bounds on multiprocessing timing anomalies. *SIAM Journal of Applied Mathematics* 17, 416–429.
- He, Y., Kellerer, H., Kotov, V., 2000. Linear compound algorithms for the partitioning problem. *Naval Research Logistics* 47, 593–601.
- Leung, J.Y.-T., Wei, W.-D., 1995. Tighter bounds on a heuristic for a partition problem. *Information Processing Letters* 56, 51–57.
- Tan, Z., He, Y., Epstein, L., 2005. Optimal on-line algorithms for the uniform machine scheduling problem with ordinal data. *Information and Computation* 196, 57–70.