Worst-Case Analysis of LPT Scheduling on Small Number of Non-Identical Processors

Takuto Mitsunobu[1], Reiji Suda[1], and Vorapong Suppakitpaisarn[1]

The University of Tokyo, Japan

Abstract. The approximation ratio of the longest processing time (LPT) scheduling algorithm has been studied in several papers. While the tight approximation ratio is known for the case when all processors are identical, the ratio is not yet known when the processors have different speeds. In this work, we give a tight approximation ratio for the case when the number of processors is 3,4, and 5. We show that the ratio for those cases are no more than the lower bound provided by Gonzalez, Ibarra, and Sahni (SIAM J. Computing 1977). They are approximately 1.38 for three processors, 1.43 for four processors, and 1.46 for five processors.
Keywords: Scheduling, LPT Algorithm, Approximation Ratio

1 Introduction

Theoretical analyses of algorithms for scheduling problems have been conducted in several recent works such as [5, 10] (refer to [14] for a survey). Several computation environments are considered in the works, but we will focus on the following environments in this paper:

- **Offline:** All task information are available before the execution of a scheduling algorithm
- **Makespan minimization:** The scheduling algorithm aims to minimize the computation time needed by the processor that finish later than others.
- **No precedence constraints:** Tasks can be executed in any order.
- **Uniform processor:** Let the size of task i be $t(i)$ and let the speed of processor p be $s(p)$. Then, the calculation time of the task at the processor is $t(i)/s(p)$.

The above setting is called as $Q||C_{\max}$ in scheduling literature [14]. The problem is NP-hard even when we have two processors with the same speed [13, 4]. A polynomial time approximation scheme for this setting is known [9], but they are practically slow. We believe that it is more common to solve this problem using other scalable algorithms. Among those algorithms, the longest processing time (LPT) algorithm is one of the most well-known. The algorithm can be described as follows:

1. Set $w(p) = 0$ for any processor p.
2. Take i as one of the largest tasks that have not yet been assigned to any processor.

3. Take p as one of processors which minimizes $w(p) + t(i)/s(p)$.
4. Assign task i to processor p and set $w(p)$ as $w(p) + t(i)/s(p)$.
5. Go back to Step 2 until all tasks are assigned.

2 Previous Works

The approximation ratio of the LPT algorithm is analyzed in several works. Dobson [2] gave an instance where the approximation ratio of LPT is 1.512 and proved the worst-case approximation ratio of LPT is not greater than $19/12 \approx 1.583$. Friesen [3] gave an instance where the approximation ratio of LPT is 1.52 and proved the worst-case approximation ratio of LPT is not greater than $5/3 \approx 1.667$ by different approach from [2]. Later, Kovács [12] gave an instance where the worst-case approximation ratio of LPT is greater than 1.54 and less than $1 + \sqrt{3}/3 \approx 1.577$.

The ratio for several special cases are also studied in several papers. Those include the case where only one processor has a different speed [6, 11], the case all speeds of processors are a power of two [11], and the case where a ratio of speeds is a parameter [1].

In this talk, we consider a special case when the number of processors is a parameter. Suppose that the number of processors is m. We calculate the approximation algorithm in term of m. As the number of processors in distributed computation is usually small, we strongly believe that analyzing this special case is very important to understand the nature of the LPT algorithm.

Although its importance, there are only two previous works for this special case. Graham [7, 8] show that, when all processors are identical, the tight approximation ratio is $4/3 - 1/(3m)$. Gonzalez, Ibarra, and Sahni [6] show that the ratio is no larger than $2m/(m+1)$. Denote the unique positive root of the equation $2x^m - x^{m-1} - \cdots - x - 2 = 0$ by ρ_m. By a calculation, we have $\rho_2 \approx 1.28$, $\rho_3 \approx 1.38$, $\rho_4 \approx 1.43$, and $\rho_5 \approx 1.46$. The authors of [6] also gave a series of instances with m processors where the approximation ratio of LPT is ρ_m. In addition, it is shown in the same paper that, when $m = 2$, the tight ratio is ρ_2.

3 Our Contributions

We show that the lower bound given by Gonzalez, Ibarra, and Sahni [6] is tight also for $m = 3, 4, 5$. In other words, ρ_3, ρ_4, and ρ_5 are tight approximation ratios when the numbers of processors are $3, 4$, and 5 respectively. Formally, we show the following theorem:

Theorem 1. *When $m = 3, 4, 5$, the worst-case approximation ratio of LPT on uniform processors is ρ_m.*

Suppose that the number of tasks is n. An informal sketch of our proof is as follows:

1. We show that, if the worst approximation ratio is attained when $n = m + 1$, then the worst approximation is ρ_m.

2. We show that the worst ratio is not attained when $n = m + 2$.
3. We show that, when $m \in \{3, 4, 5\}$, the worst approximation ratio is attained only when $n \leq m + 2$.

Acknowledgement

The authors would like to thank Taku Onodera for several useful comments and ideas. The work of Vorapong Suppakitpaisarn is funded by JSPS Grant-in-Aid for Transformative Research Areas A grant number JP21H05845.

References

1. Chen, B.: Parametric bounds for LPT scheduling on uniform processors. Acta Mathematicae Applicatae Sinica **7**(1), 67–73 (1991)
2. Dobson, G.: Scheduling independent tasks on uniform processors. SIAM Journal on Computing **13**(4), 705–716 (1984)
3. Friesen, D.K.: Tighter bounds for LPT scheduling on uniform processors. SIAM Journal on Computing **16**(3), 554–560 (1987)
4. Garey, M.R., Johnson, D.S.: "strong"np-completeness results: Motivation, examples, and implications. Journal of the ACM (JACM) **25**(3), 499–508 (1978)
5. Ghalami, L., Grosu, D.: Scheduling parallel identical machines to minimize makespan: A parallel approximation algorithm. Journal of Parallel and Distributed Computing **133**, 221–231 (2019)
6. Gonzalez, T., Ibarra, O.H., Sahni, S.: Bounds for LPT schedules on uniform processors. SIAM journal on Computing **6**(1), 155–166 (1977)
7. Graham, R.L.: Bounds for certain multiprocessing anomalies. Bell system technical journal **45**(9), 1563–1581 (1966)
8. Graham, R.L.: Bounds on multiprocessing timing anomalies. SIAM journal on Applied Mathematics **17**(2), 416–429 (1969)
9. Hochbaum, D.S., Shmoys, D.B.: A polynomial approximation scheme for machine scheduling on uniform processors: using the dual approximation approach. In: FSTTCS'86. pp. 382–393. Springer (1986)
10. Jansen, K., Lassota, A., Maack, M.: Approximation algorithms for scheduling with class constraints. In: SPAA'20. pp. 349–357 (2020)
11. Kovács, A.: Tighter approximation bounds for LPT scheduling in two special cases. Journal of Discrete Algorithms **7**(3), 327–340 (2009)
12. Kovács, A.: New approximation bounds for LPT scheduling. Algorithmica **57**(2), 413–433 (2010)
13. Sahni, S.K.: Algorithms for scheduling independent tasks. Journal of the ACM (JACM) **23**(1), 116–127 (1976)
14. Suda, R.: A survey on task scheduling for heterogeneous parallel computing environments (in japanese). IPSJ Trans. on Advanced Computing Systems **47**(SIG 18), 92–114 (2006)