Asymptotic Optimality of LPT

Update on the Asymptotic Optimality of LPT

Anne Benoit
Louis-Claude Canon
Redouane Elghazi
Pierre-Cyrille Héam

ENS de Lyon, FEMTO-ST, Univ. Franche-Comté, CNRS

SCALE – April 13, 2022

Outline

Outline

Introduction

Classical scheduling $P||C_{max}$:

- n independent tasks with costs $p_1, \ldots, p_n$;
- m identical processors;
- we want to minimize the time C_{max} at which we finish executing the last task.

Introduction

Classical scheduling $P||C_{max}$:

- n independent tasks with costs $p_1, \ldots, p_n$;
- m identical processors;
- we want to minimize the time C_{max} at which we finish executing the last task.

We study the greedy heuristic *Longest Processing Time* (LPT):

- the tasks are greedily executed, from the longest one;
- simple, low complexity;
- good performance in theory and in practice.

Introduction

Classical scheduling $P||C_{max}$:

- n independent tasks with costs $p_1, \ldots, p_n$;
- m identical processors;
- we want to minimize the time C_{max} at which we finish executing the last task.

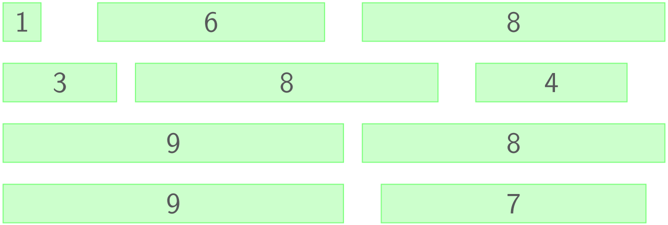
We study the greedy heuristic *Longest Processing Time* (LPT):

- the tasks are greedily executed, from the longest one;
- simple, low complexity;
- good performance in theory and in practice.

We study its asymptotic behavior for specific workload distributions.

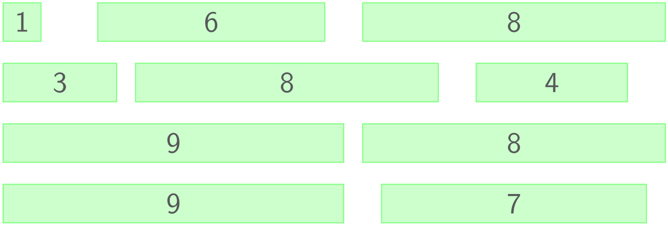
Example

Example instance, with 10 tasks and 3 processors:

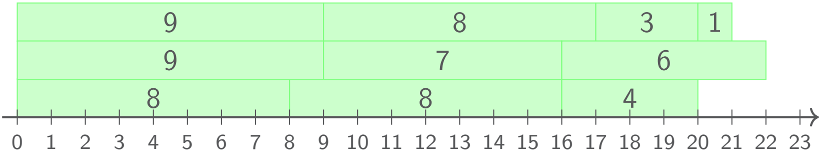


Example

Example instance, with 10 tasks and 3 processors:



The solution of LPT:



Motivation

We study one of the most basic scheduling problems:

- no precedence graph;
- no communication cost;
- machines are identical;
- the execution times of tasks are fully given (offline);
- each task is executed on a single processor.

Motivation

We study one of the most basic scheduling problems:

- no precedence graph;
- no communication cost;
- machines are identical;
- the execution times of tasks are fully given (offline);
- each task is executed on a single processor.

By refining our understanding of the behavior of LPT in this simple case, we hope to better understand why LPT behaves well or not in more complex setups.

Contributions

- theoretical study of LPT under the constraint that the distribution of the workload is a Uniform integer composition:
  - the costs of the tasks are not independently and identically distributed variables;
  - the tasks may have a minimum non-zero cost.
- empirical study for several distributions:
  - we found the literature to lack empirical assessments for the tightness of theoretical bounds.

Studied algorithms

- LPT: Largest Processing Time, with complexity $O(n \log n)$;
- LS: List Scheduling, with complexity $O(n \log m)$;
- MD: Median Discriminated, with complexity $O(n \log m)$;
- SLACK[1], with complexity $O(n \log n)$;
- LDM[2], with complexity $O(n(\log n + m \log m))$.

[1] Della Croce and Scatamacchia, “The Longest Processing Time rule for identical parallel machines revisited”, 2020.

[2] Michiels et al., “Performance Ratios for the Karmarkar-Karp Differencing Method”, 2003.

Existing results

Problem	Distribution	Studied quantity	Convergence/rate
$P\|\|C_{\max}$	$\mathcal{U}(0,1)$	$E[\mathrm{LPT}]/E[\mathrm{OPT}^*]$	$1+O(m^2/n^2)$ [3]
$P\|\|C_{\max}$	$F(x)=x^a$, $0<a<\infty$	$\mathrm{LPT}-\mathrm{OPT}$	$O((\log\log(n)/n)^{\frac{1}{a}})$ almost surely (a.s.) [4]
$P\|\|C_{\max}$	as above	$E[(\mathrm{LPT}-\mathrm{OPT})^q]$	$O((1/n)^{\frac{a}{q}})$ [4]
		...	

[3] Coffman, Frederickson, and Lueker, "Probabilistic analysis of the LPT processor scheduling heuristic", 1982.

[4] Frenk and Rinnooy Kan, "The rate of convergence to optimality of the LPT rule", 1986.

Outline

Uniform integer composition

We present the distribution $\mathbb{D}_W$:

Definition ($\mathbb{D}_W$)

- the total amount of work $W = \sum p_i$ is fixed;
- this work is split among the tasks;
- of all the possible splitting, a list of tasks L is chosen uniformly;

- example of $\mathbb{D}_3$: (1,1,1), (1,2), (2,1), (3) with probability $\frac{1}{4}$ each;
- example situation: corresponds to a large application with unknown parallelization;
- easy to sample.

Uniform integer composition

Theorem

Algorithms LPT and SLACK are optimal for $\mathbb{D}_W$, with probability $1 - O\left(\frac{1}{W}\right)$. For any fixed number of machines m, for L a list of tasks generated according to $\mathbb{D}_W$,

$$\mathbb{P}(\mathsf{LPT}(L, m) = \mathsf{OPT}(L, m)) = 1 - O\left(\frac{1}{W}\right), \quad \textit{and}$$

$$\mathbb{P}(\mathsf{SLACK}(L, m) = \mathsf{OPT}(L, m)) = 1 - O\left(\frac{1}{W}\right), \quad \textit{and}$$

$$\mathbb{P}(\mathsf{LDM}(L, m) = \mathsf{OPT}(L, m)) = 1 - O\left(\frac{1}{W}\right), \quad \textit{and}$$

$$\mathbb{P}(\mathsf{MD}(L, m) \leq \mathsf{OPT}(L, m) + 1) = 1 - O\left(\frac{1}{W}\right).$$

Uniform integer composition with a minimum

We present the distribution $\mathbb{D}_{W,p_{min}}$:

Definition ($\mathbb{D}_{W,p_{min}}$)

- the total amount of work $W = \sum p_i$ is fixed;
- this work is split among the tasks execution time at least p_{min} each;
- of all the possible splitting, a list of tasks L is chosen uniformly;

- example of $\mathbb{D}_{6,2}$: (2,2,2), (2,4), (3,3), (4,2) with probability $\frac{1}{4}$ each;
- harder to sample than the previous version: we use dynamic programming.

Uniform integer composition with a minimum

Theorem

Let m be a fixed number of machines. One has, for a list of tasks L generated according to $\mathbb{D}_{W,p_{\min}}$,

$$\mathbb{P}(|\mathrm{LPT}(L,m) - \mathrm{OPT}(L,m)| \leq p_{\min}) \underset{W\to+\infty}{\longrightarrow} 1 \quad \textit{and}$$

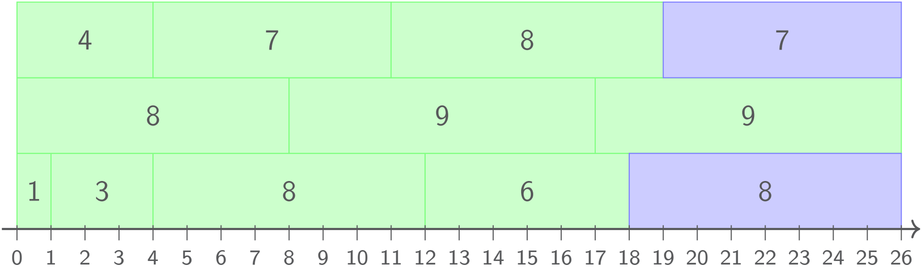
$$\mathbb{P}(|\mathrm{SLACK}(L,m) - \mathrm{OPT}(L,m)| \leq p_{\min}) \underset{W\to+\infty}{\longrightarrow} 1.$$

$$\mathbb{P}(|\mathrm{LDM}(L,m) - \mathrm{OPT}(L,m)| \leq p_{\min}) \underset{W\to+\infty}{\longrightarrow} 1.$$

Outline

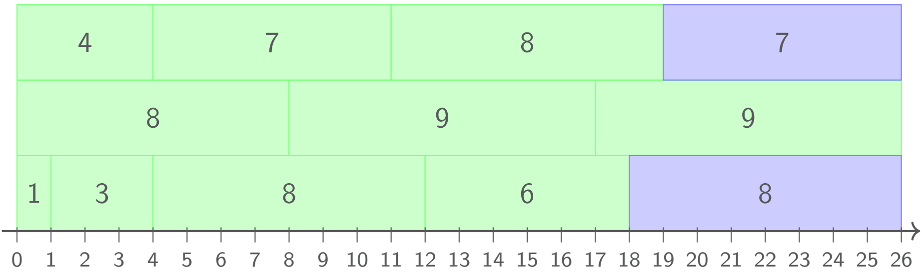
Empirical results – Optimality transform technique

The optimal solution for an instance is usually unknown.
In order to still have the optimality ratio of a solution, we add tasks so that we know an optimal solution.



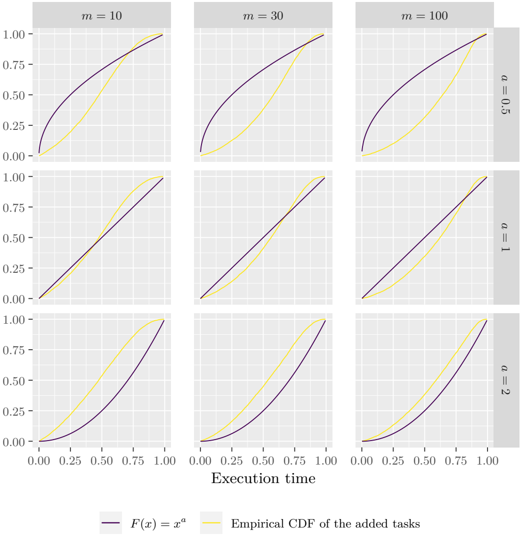
Empirical results – Optimality transform technique

The optimal solution for an instance is usually unknown.
In order to still have the optimality ratio of a solution, we add tasks so that we know an optimal solution.



Can we measure the proximity between the original tasks and the "padding" tasks?

Empirical results – Optimality transform results



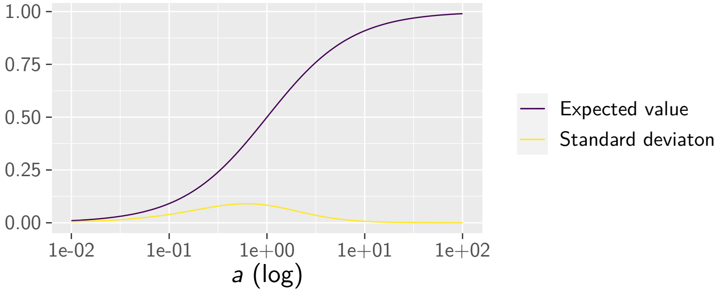
Empirical results

We have 3 sets of experiments. We try LPT and the related heuristics on:

1. instances with distribution function $F(x) = x^a$ with $a > 0$. We compare the results to the asymptotic bound found by Frenk and Rinnooy Kan[5]. They state that LPT $-$ OPT should decrease in $O((\frac{\log\log(n)}{n})^{\frac{1}{a}})$ almost surely;
2. realistic instances based on the Parallel Workload Archive (PWA);
3. instances with distribution following the Uniform integer composition without a minimum ($\mathbb{D}_W$) and with a minimum ($\mathbb{D}_{W,p_{min}}$).

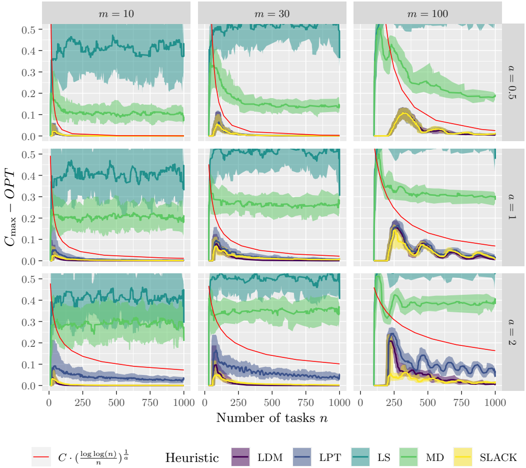
[5] Frenk and Rinnooy Kan, "The rate of convergence to optimality of the LPT rule", 1986.

Empirical results – $F(x) = x^a$



Expected value and standard deviation of a random variable with cumulative distribution function $F(x) = x^a$ as a function of a with $0 < a < \infty$.

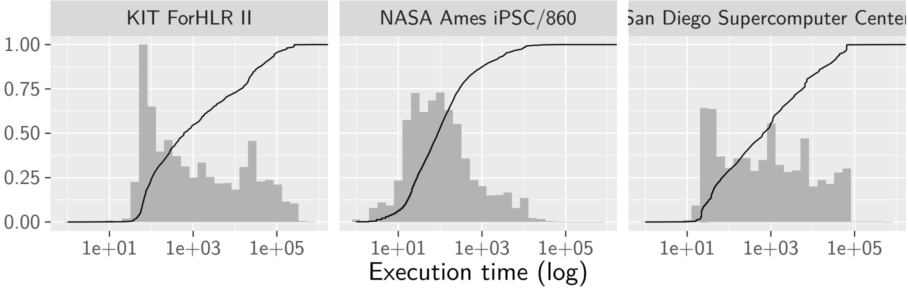
Empirical results – $F(x) = x^a$



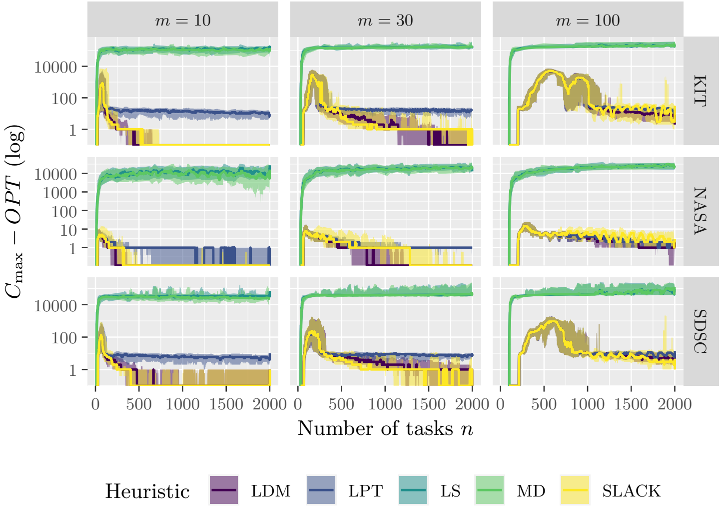
Empirical results – Realistic instances

To create realistic instances, we computed the empirical Cumulative Distribution Function from real instances.

Distributions of task costs for the KIT ForHLR II and NASA Ames iPSC/860 instances.



Empirical results – Realistic instances



Empirical results – Unif. integer comp. ($\mathbb{D}_W$)

Distribution in percentages of the absolute errors observed for LS and MD with W from 10 to 9999 and $m \in \{10, 30, 100\}$.

abs. err.	LS	MD	abs. err.	LS	MD
0	3.4	67.7	6	9.1	0.04
1	10.7	30.3	7	5.0	$<$ 0.01
2	17.0	0.96	8	2.7	0
3	18.1	0.58	9	1.5	0
4	17.3	0.28	10	0.7	0
5	13.7	0.08	$>$10	0.6	0

LPT and SLACK were always optimal.

Empirical results – Unif. integer comp. ($\mathbb{D}_{W,p_{min}}$)

Results on the difference between the $C_{\max}$ computed by the heuristics and a lower bound of the optimal makespan OPT. Each line is related to different $D_{W,p_{\min}}$. The results are presented as max – avg – std. Each value is obtained with W from 10 to 9999 and for $m \in \{10, 30, 100\}$.

$p_{\min}$	LPT	MD	SLACK	LDM
3	2 – 0.93 – 0.70	5 – 1.48 – 0.56	2 – 0.61 – 0.60	2 – 0.57 – 0.57
5	4 – 1.84 – 1.23	9 – 2.69 – 0.87	5 – 1.11 – 0.86	4 – 1.07 – 0.82
7	6 – 2.71 – 1.80	13 – 3.65 – 1.15	6 – 1.60 – 1.17	6 – 1.56 – 1.11
10	9 – 3.99 – 2.65	15 – 5.23 – 1.80	11 – 2.32 – 1.62	9 – 2.29 – 1.57

femto-st SCIENCES & TECHNOLOGIES

Empirical results – Unif. integer comp. ($\mathbb{D}_{W,p_{min}}$)

Results on the difference between the $C_{\max}$ computed by the heuristics and a lower bound of the optimal makespan OPT. Each line is related to different $D_{W,p_{\min}}$. The results are presented as max – avg – std. Each value is obtained with W from 10 to 9999 and for $m \in \{10, 30, 100\}$.

$p_{\min}$	LPT	MD	SLACK	LDM
3	2 – 0.93 – 0.70	5 – 1.48 – 0.56	2 – 0.61 – 0.60	2 – 0.57 – 0.57
5	4 – 1.84 – 1.23	9 – 2.69 – 0.87	5 – 1.11 – 0.86	4 – 1.07 – 0.82
7	6 – 2.71 – 1.80	13 – 3.65 – 1.15	6 – 1.60 – 1.17	6 – 1.56 – 1.11
10	9 – 3.99 – 2.65	15 – 5.23 – 1.80	11 – 2.32 – 1.62	9 – 2.29 – 1.57

Empirical results – Unif. integer comp. ($\mathbb{D}_{W,p_{min}}$)

Results on the difference between the $C_{\max}$ computed by the heuristics and a lower bound of the optimal makespan OPT. Each line is related to different $D_{W,p_{\min}}$. The results are presented as max – avg – std. Each value is obtained with W from 10 to 9999 and for $m \in \{10, 30, 100\}$.

$p_{\min}$	LPT	MD	SLACK	LDM
3	2 – 0.93 – 0.70	5 – 1.48 – 0.56	2 – 0.61 – 0.60	2 – 0.57 – 0.57
5	4 – 1.84 – 1.23	9 – 2.69 – 0.87	5 – 1.11 – 0.86	4 – 1.07 – 0.82
7	6 – 2.71 – 1.80	13 – 3.65 – 1.15	6 – 1.60 – 1.17	6 – 1.56 – 1.11
10	9 – 3.99 – 2.65	15 – 5.23 – 1.80	11 – 2.32 – 1.62	9 – 2.29 – 1.57

SLACK is better than LPT and LDM is slightly better than SLACK.

femto-st SCIENCES & TECHNOLOGIES

Outline

femto-st SCIENCES & TECHNOLOGIES

Conclusion

What has been done:

- Definition and theoretical analysis of a distribution that is non-independent that supports a minimum execution time p_{min}: the Uniform integer composition ($\mathbb{D}_W$ and $\mathbb{D}_{W,p_{\min}}$);
- Empirical comparison of five heuristics (LPT, MD, LS, SLACK, LDM) under different distributions ($F(x) = x^a$, realistic instance, Uniform integer composition);
- Empirical evaluation of the bound on the convergence of LPT under the distribution $F(x) = x^a$. We found this bound to be tight.

Conclusion

What has been done:

- Definition and theoretical analysis of a distribution that is non-independent that supports a minimum execution time p_{min}: the Uniform integer composition ($\mathbb{D}_W$ and $\mathbb{D}_{W,p_{\min}}$);
- Empirical comparison of five heuristics (LPT, MD, LS, SLACK, LDM) under different distributions ($F(x) = x^a$, realistic instance, Uniform integer composition);
- Empirical evaluation of the bound on the convergence of LPT under the distribution $F(x) = x^a$. We found this bound to be tight.

Future work:

- introduce uncertainty in the costs p_i of the tasks;
- explore dependent cost distributions other than the Uniform integer composition;
- explore known independent distributions, but with a minimum execution time.

femto-st SCIENCES & TECHNOLOGIES

Thank you for your attention

redouane.elghazi@femto-st.fr