Efficient Algorithms and Provably Good Solutions for NP-hard Scheduling Problems

Martin Skutella
(joint work with Sven Jäger)

May 12, 2021

schedulingseminar.com

Berlin Mathematics Research Center
MATH+

Side Note

A simple proof of the Moore-Hodgson Algorithm for minimizing the number of late jobs

Joseph Cheriyan[a,1,*], R. Ravi[b,2], Martin Skutella[c,3]

[a] C&O Dept., University of Waterloo, Waterloo, ON, Canada N2L 3G1
[b] Tepper School of Business, Carnegie Mellon University, Pittsburgh, USA
[c] Institute of Mathematics, Technische Universität Berlin, Germany

Abstract

The Moore-Hodgson Algorithm minimizes the number of late jobs on a single machine. That is, it finds an optimal schedule for the classical problem $1 \mid\mid \sum U_j$. Several proofs of the correctness of this algorithm have been published. We present a new short proof.

Keywords: Scheduling theory, Moore-Hodgson Algorithm, number of late jobs

arxiv.org/abs/2104.06210

How to Tackle NP-hard Scheduling Problems

Exact methods: find optimal solution at the cost of an exponential worst-case running time; sometimes work well in practice;

Examples: Dynamic Programming, Integer Programming, Constraint Programming, Branch and Bound, ...

Heuristic methods: work well in practice but usually do not come with a worst-case performance guarantee or running time bound;

Examples: local search, simulated annealing, genetic algorithms, greedy heuristics, machine learning, ...

Approximation algorithms: find in polynomial time a feasible solution with an a priori bound on the quality of the computed solution;

Examples: combinatorial algorithms, LP-based, primal-dual, greedy, local search, iterative rounding, ...

Approximation Algorithms

Definition.

i. An α-approximation algorithm for a minimization problem finds in polynomial time a feasible solution whose value is within a factor of α of the optimum. The factor $\alpha \geq 1$ is called performance ratio.
ii. A family of $(1+\varepsilon)$-approximation algorithms for each $\varepsilon > 0$ is a polynomial-time approximation scheme (PTAS).
iii. A PTAS whose running time is polynomial in the input size and $1/\varepsilon$ is a fully polynomial-time approximation scheme (FPTAS).

Examples:

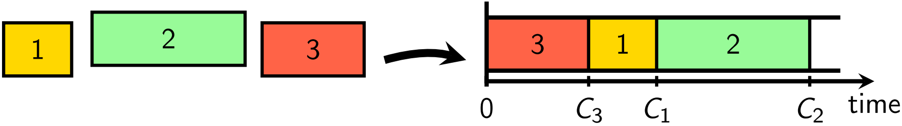
Scheduling identical parallel machines with makespan objective: $P\,|\,|\,C_{\max}$

- List scheduling is a 2-approximation algorithm (Graham 1966).
- List scheduling in order of non-increasing job sizes is a $4/3$-approximation algorithm (Graham 1969).
- FPTAS for fixed number of machines m (Horowitz & Sahni 1976)
- PTAS (Hochbaum & Shmoys 1987)

Total Weighted Completion Time Objective

Given: n jobs $j = 1, \ldots, n$, processing times $p_j > 0$, weights $w_j > 0$

Task: schedule jobs on a single machine; minimize $\sum_j w_j C_j$



Weighted Shortest Processing Time (WSPT) rule:

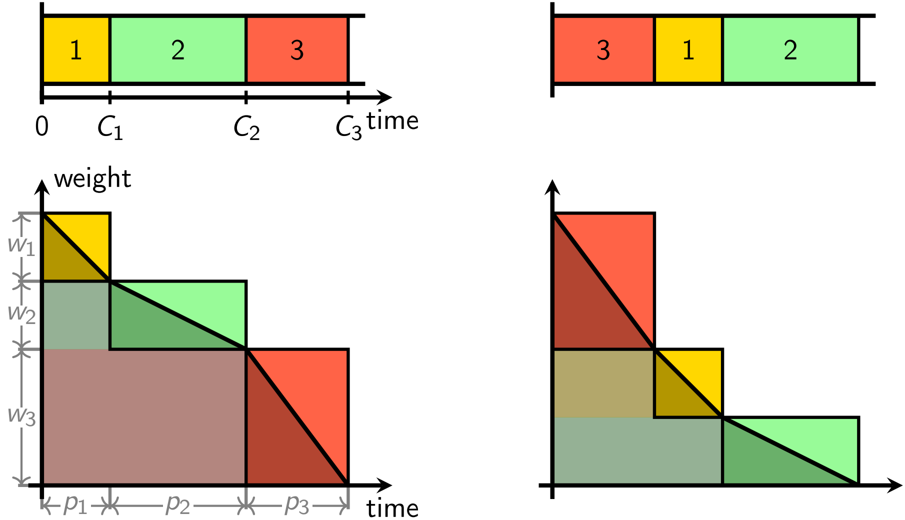
Theorem (Smith 1956).

Sequencing jobs in order of non-increasing ratios w_j/p_j is optimal.

"Photographer's Rule"

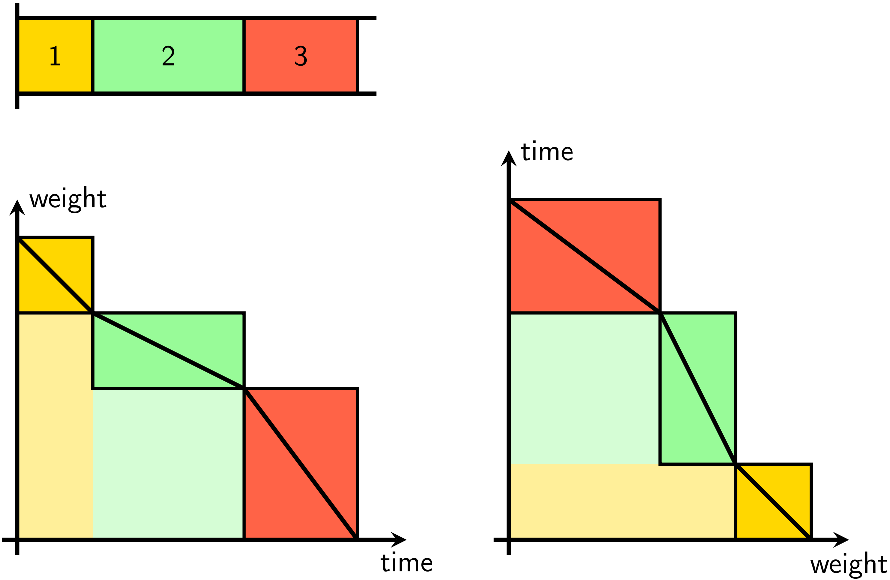
Proof of WSPT Rule via Two-Dimensional Gantt Charts

Eastman, Even & Isaacs 1964; Goemans & Williamson 2000



$^{w_j}/_{p_j}$ = diagonal slope of rectangle representing job j

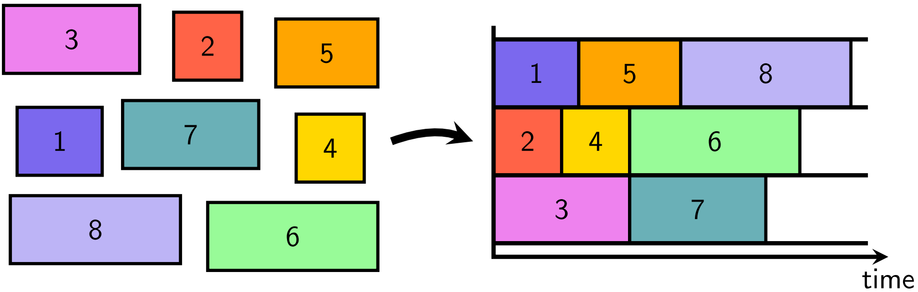
Swap Weights and Processing Times



Parallel Machine Scheduling to Minimize $\sum w_j C_j$

Given: n jobs $j = 1, \ldots, n$, processing times $p_j > 0$, weights $w_j > 0$

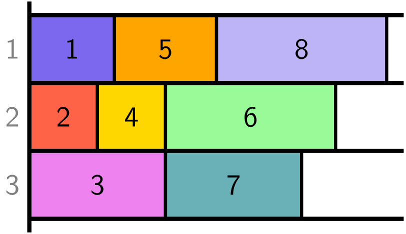
Task: schedule jobs on m parallel machines; minimize $\sum_j w_j C_j$



- weakly NP-hard for two machines (Bruno, Coffman & Sethi 1974)
- strongly NP-hard if m part of input (Garey & Johnson, problem SS13)
- FPTAS for fixed number of machines m (Sahni 1976)
- PTAS (Sk. & Woeginger 2000)

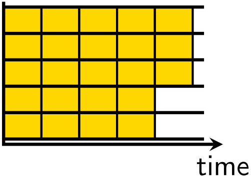
List Scheduling in Order of Non-Increasing w_j/p_j

$w_1/p_1 \geq w_2/p_2 \geq \ldots \geq w_n/p_n$



Theorem (Conway, Maxwell & Miller 1967).

Optimal if $w_j = 1$ for all j (or: $p_j = 1$ for all j).



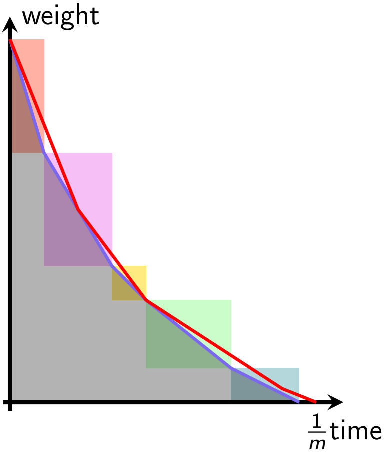
Theorem (Kawaguchi & Kyan 1986).

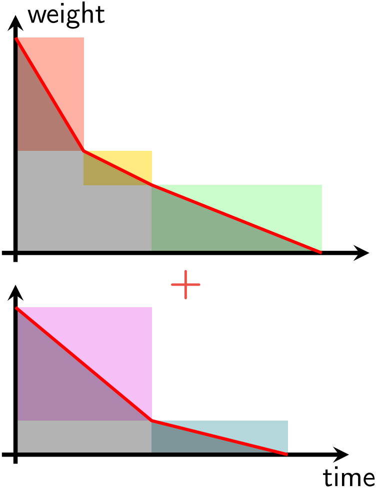
Tight performance ratio: $\frac{1+\sqrt{2}}{2} \approx 1.207$

Fast Single Machine Lower Bound

Lemma (Eastman, Even & Isaacs 1964).

$$\frac{1}{m}\Big(\mathrm{OPT}_1 - \frac{1}{2}\sum_j w_j p_j\Big) \;\leq\; \mathrm{OPT}_m - \frac{1}{2}\sum_j w_j p_j$$



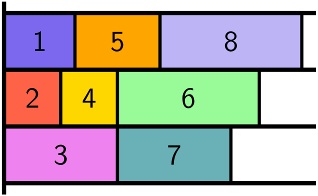


The Performance Ratio of WSPT is at most 3/2

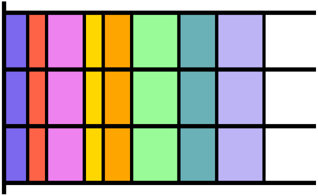
Lemma (Eastman, Even & Isaacs 1964).

$$\frac{1}{m}\Big(\mathrm{OPT}_1 - \frac{1}{2}\sum_j w_j p_j\Big) \;\leq\; \mathrm{OPT}_m - \frac{1}{2}\sum_j w_j p_j$$

WSPT



OPT_1 / m



WSPT start times $\leq$ single machine start times

Thus:

$$\mathrm{WSPT}_m \leq \frac{1}{m}\Big(\mathrm{OPT}_1 - \frac{1}{2}\sum_j w_j p_j\Big) + \sum_j w_j p_j$$

$$\leq \mathrm{OPT}_m + \frac{1}{2}\sum_j w_j p_j \;\leq\; \frac{3}{2}\,\mathrm{OPT}_m$$

Simplified and Refined Proof of the Kawaguchi-Kyan Bound

Theorem (Kawaguchi & Kyan 1986).

WSPT has performance ratio exactly $\frac{1+\sqrt{2}}{2} \approx 1.207$

Proof idea: explicit construction of worst-case instance (for $m \to \infty$)

Schwiegelshohn 2011: considerably simplified proof (but same idea)

Jäger & Sk. 2018/21: construction of worst-case instance for each fixed m

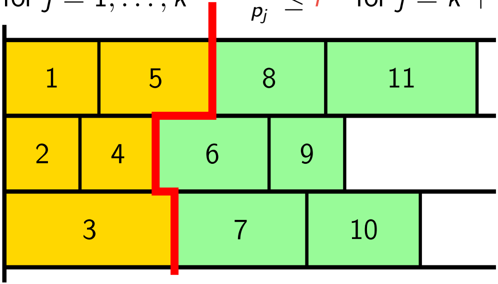
Sequence of reductions to worst-case instances with:

- i $w_j = p_j$ for all j
- ii at most $m - 1$ large jobs and many tiny jobs
- iii all large jobs are extra-large
- iv all extra-large jobs have same size

First Reduction: $w_j = p_j \ \forall j$

$\frac{w_j}{p_j} \geq R$ for $j = 1, \ldots, k$ $\qquad$ $\frac{w_j}{p_j} \leq r$ for $j = k+1, \ldots, n$

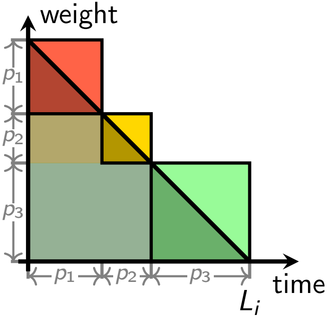
$R > r$



$$\sum_{j=1}^{n} w_j C_j = \frac{r}{R} \sum_{j=1}^{k} w_j C_j + \sum_{j=k+1}^{n} w_j C_j + \left(1 - \frac{r}{R}\right) \sum_{j=1}^{k} w_j C_j$$

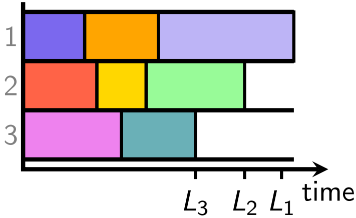
$$\Longrightarrow \qquad \frac{\text{WSPT}}{\text{OPT}} = \frac{A_{\text{WSPT}} + B_{\text{WSPT}}}{A_{\text{OPT}} + B_{\text{OPT}}} \leq \max\left\{\frac{A_{\text{WSPT}}}{A_{\text{OPT}}}, \frac{B_{\text{WSPT}}}{B_{\text{OPT}}}\right\}$$

Objective Function in Terms of Machine Loads (for $w_j = p_j$)



one machine i:

$$\sum_{j \to i} p_j C_j = \frac{1}{2} \Big(\underbrace{\sum_{j \to i} p_j}_{L_i} \Big)^2 + \frac{1}{2} \sum_{j \to i} {p_j}^2$$



m-machine schedule:

$$\sum_{j=1}^{n} p_j C_j = \frac{1}{2} \sum_{i=1}^{m} {L_i}^2 + \frac{1}{2} \sum_{j=1}^{n} {p_j}^2$$

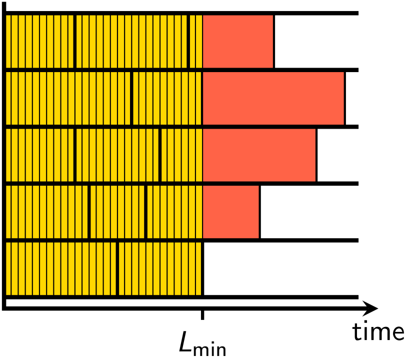
notice:

- $\sum_i L_i = \sum_j p_j$ (fixed)
- $\sum_i {L_i}^2$ minimal if $L_1 = \cdots = L_m$

Second Reduction: Large Jobs and 'Sand'™ (G. J. Woeginger)

$$\sum_j p_j C_j = \tfrac{1}{2} \sum_i {L_i}^2 + \tfrac{1}{2} \sum_j {p_j}^2$$

WSPT schedule



WSPT:

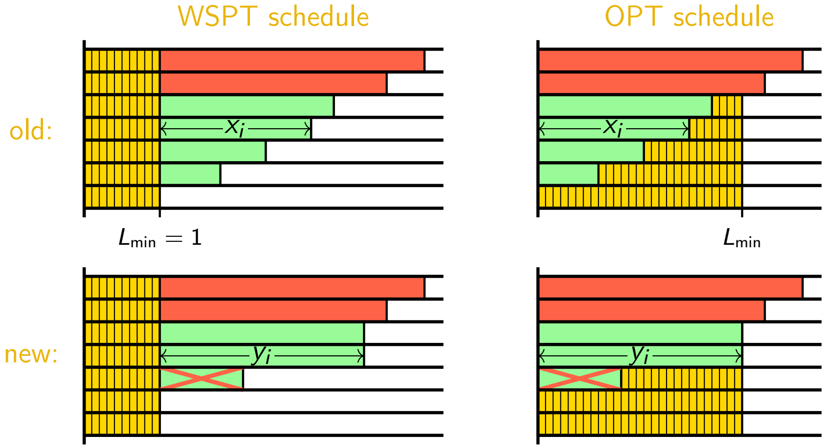
- $\sum_i {L_i}^2$ remains unchanged
- $\sum_j {p_j}^2$ decreased by $\delta \geq 0$

OPT:

- $\sum_i {L_i}^2$ unchanged or decreases
- $\sum_j {p_j}^2$ decreased by same $\delta \geq 0$

$\Longrightarrow \quad \dfrac{\text{WSPT}}{\text{OPT}}$ unchanged or increased

Third Reduction: Make Large Jobs Extra-Large

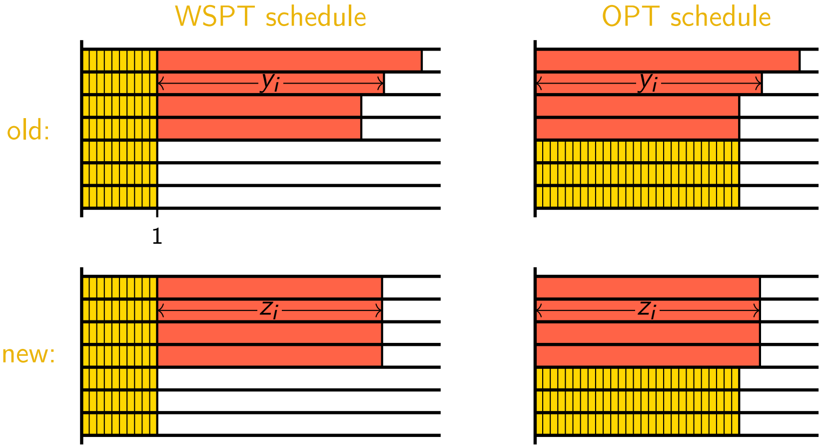


Increase in objective:

$$\frac{1}{2}\sum_i\big((1+y_i)^2 + {y_i}^2 - (1+x_i)^2 - {x_i}^2\big) \qquad \frac{1}{2}\sum_i\big({y_i}^2 - {x_i}^2\big) \geq 0$$

$$= \sum_i\big({y_i}^2 - {x_i}^2\big) \qquad \text{as } \sum_i x_i = \sum_i y_i$$

Fourth Reduction: All Extra-Large Jobs have Same Size

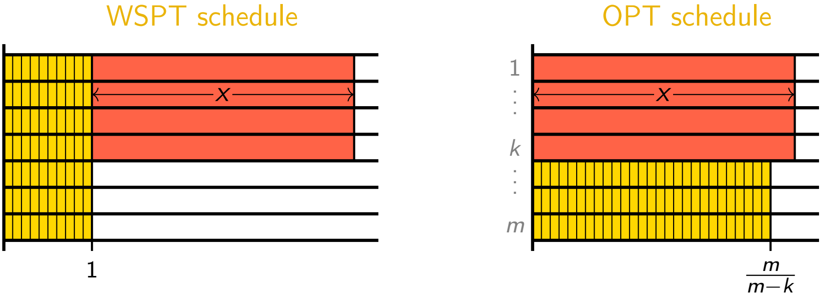


Increase in objective:

$$\frac{1}{2}\sum_i\Big((1+z_i)^2+{z_i}^2-(1+y_i)^2-{y_i}^2\Big) \qquad \sum_i\big({z_i}^2-{y_i}^2\big)\leq 0$$

$$=\sum_i\big({z_i}^2-{y_i}^2\big) \qquad \text{as } \sum_i z_i=\sum_i y_i$$

Analyzing the Performance Ratio



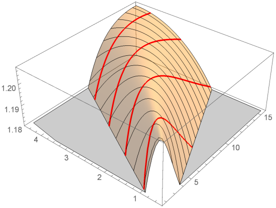
$$\text{WSPT} = \frac{m}{2} + k \cdot x(1+x) \qquad \text{OPT} = k \cdot x^2 + \frac{m^2}{2(m-k)}$$

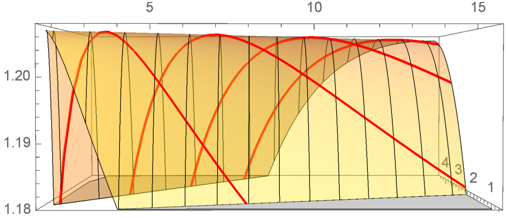
$$\frac{\text{WSPT}}{\text{OPT}} = \frac{(m-k)(2kx^2 + 2kx + m)}{(m-k)2kx^2 + m^2}$$

Observation: for fixed k, m, maximum ratio at $x = \dfrac{m}{\sqrt{k(2m-k)} - k}$

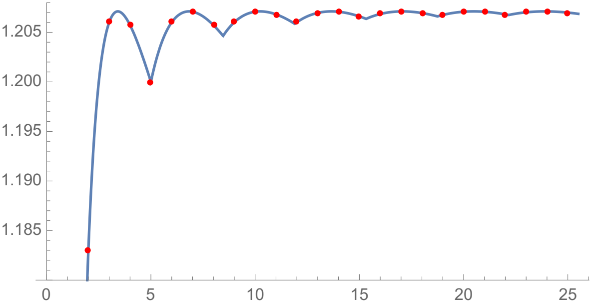
Worst-Case Instances

worst-case performance ratio for fixed m: $\max_k \left(1 - \frac{k}{2m} + \sqrt{\frac{k}{2m}\left(1 - \frac{k}{2m}\right)}\,\right)$



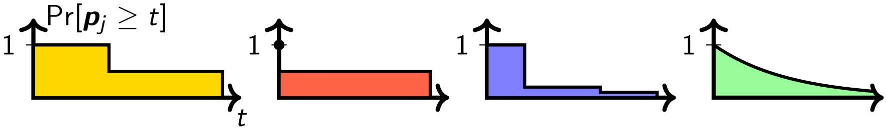


Observation: for each fixed m, maximum at $k = \left\lfloor \left(1 - \frac{1}{2}\sqrt{2}\right) m \right\rceil$.



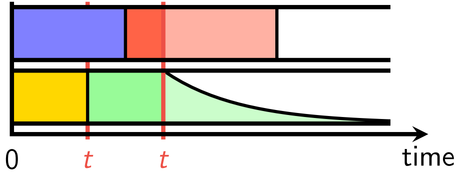
Stochastic Scheduling on Identical Parallel Machines

Given: distributions of independent random processing times $\boldsymbol{p}_j \geq 0$



Task: find m-machine scheduling policy minimizing $\mathrm{E}\left[\sum w_j C_j\right]$

- scheduling policy must be non-anticipative, i.e., decision made at time t may only depend on the information known at time t



Weighted Shortest Expected Processing Time (WSEPT)

WSEPT Rule

List scheduling in order of non-increasing $w_j / \mathrm{E}[\boldsymbol{p}_j]$.

- WSEPT is optimal for single machine (Rothkopf 1966)
- WSEPT has performance ratio $1 + \frac{1}{2}(1 + \Delta)$ with $\Delta \geq \frac{\mathrm{Var}[\boldsymbol{p}_j]}{\mathrm{E}[\boldsymbol{p}_j]^2}$ for all j. (Möhring, Schulz & Uetz 1999)
- WSEPT has no constant performance ratio. (Cheung, Fischer, Matuschke & Megow 2014; Im, Moseley & Pruhs 2015)
- WSEPT has performance ratio $1 + \frac{1}{2}(\sqrt{2} - 1)(1 + \Delta)$. (Jäger & Sk. 2018)

Open Problem

Online setting:

- jobs arrive one by one; must be immediately assigned to machines
- on each machine, assigned jobs are optimally sequenced (WSPT)

Algorithm MinIncrease

- assign job to machine minimizing increase of current objective value

Known results:

- MinIncrease has competitive ratio $\frac{3}{2} - \frac{1}{2m}$.
- If jobs arrive in order of non-increasing or non-decreasing w_j/p_j, then MinIncrease achieves competitive ratio $\frac{1}{2}(1+\sqrt{2})$.

Conjecture (Stougie 2017).

MinIncrease has competitive ratio $\frac{1}{2}(1+\sqrt{2})$.

Some References

- J. Bruno, E. G. Coffman, Jr., and R. Sethi: *Scheduling independent tasks to reduce mean finishing time*, Commun. ACM 17(7):382-387, 1974
- W. L. Eastman, S. Even, and I. M. Isaacs: *Bounds for the Optimal Scheduling of n Jobs on m Processors*, Management Science 11(2):268-279, 1964
- M. X. Goemans and D. P. Williamson: *Two-Dimensional Gantt Charts and a Scheduling Algorithm of Lawler*, SIAM J. Discrete Math. 13(3):281-294, 2000
- S. Jäger: *Approximation in Deterministic and Stochastic Machine Scheduling*, PhD thesis, TU Berlin, 2021
- S. Jäger, M. Skutella: *Generalizing the Kawaguchi-Kyan Bound to Stochastic Parallel Machine Scheduling*, STACS 2018: 43:1-43:14
- T. Kawaguchi and S. Kyan: *Worst Case Bound of an LRF Schedule for the Mean Weighted Flow-time Problem*, SIAM J. Comput. 15(4):1119-1129, 1986
- N. Megow, M. Uetz, and T. Vredeveld: *Models and Algorithms for Stochastic Online Scheduling*, Math. Oper. Res. 31(3):513-525, 2006
- R. H. Möhring, A. S. Schulz, and M. Uetz: *Approximation in Stochastic Scheduling: The Power of LP-Based Priority Policies*, J. ACM 46(6):924-942, 1999
- M. H. Rothkopf: *Scheduling with random service times*, Man. Sci. 12(9): 707-713, 1966
- S. K. Sahni: *Algorithms for Scheduling Independent Tasks*, J. ACM 23(1):116-127, 1976
- U. Schwiegelshohn: *An Alternative Proof of the Kawaguchi-Kyan Bound for the Largest-Ratio-First Rule*, Oper. Res. Lett. 39:255-259, 2011
- M. Skutella and G. J. Woeginger: *A PTAS for Minimizing the Total Weighted Completion Time on Identical Parallel Machines*, Math. Oper. Res. 25(1):63-75, 2000
- W. E. Smith: *Various Optimizers for Single-stage Production*, Naval Res. Logist. Quart. 3(1-2):59-66, 1956