A PTAS for minimizing the total weighted completion time on identical parallel machines

MARTIN SKUTELLA

C.O.R.E. & TU Berlin

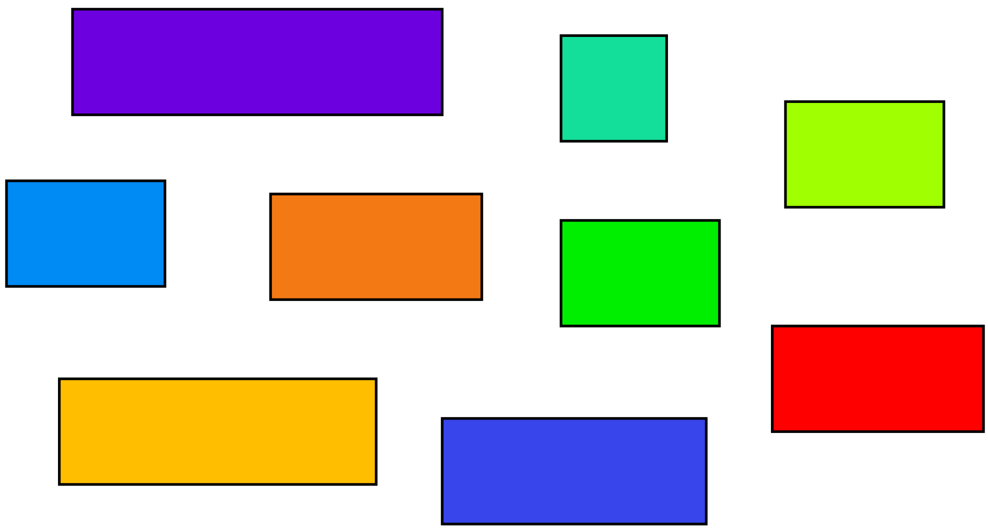
&

GERHARD J. WOEGINGER

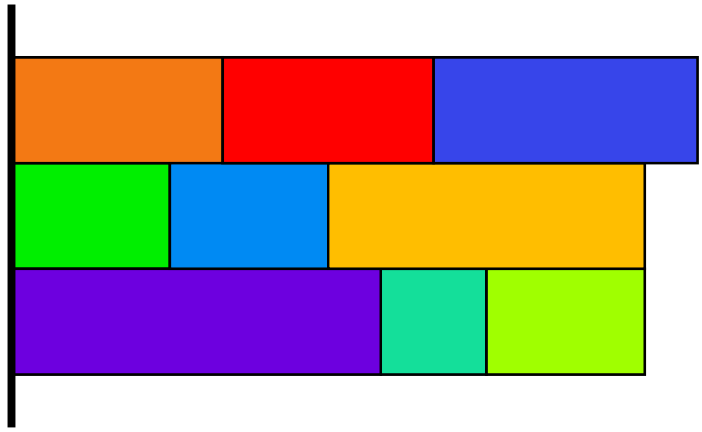
TU Graz

THE PROBLEM $P||\sum w_j C_j$

Given: n jobs, job j has size $p_j > 0$ and weight $w_j \geqslant 0$;
m identical parallel machines;



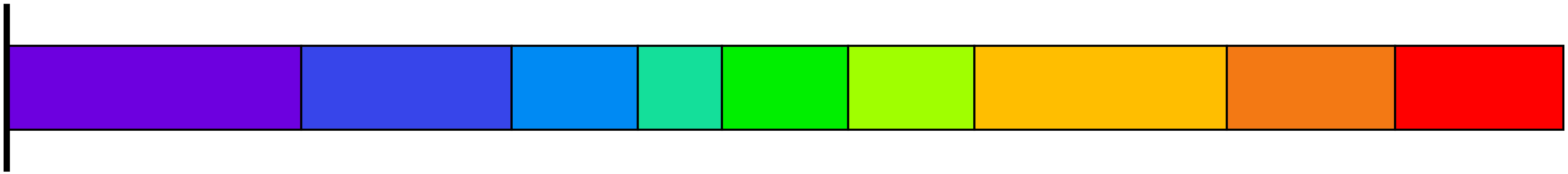
Goal: Construct a schedule that minimizes the total weighted completion time $\sum_{j=1}^{n} w_j C_j$



C_j denotes the completion time of job j

Complexity of the problem

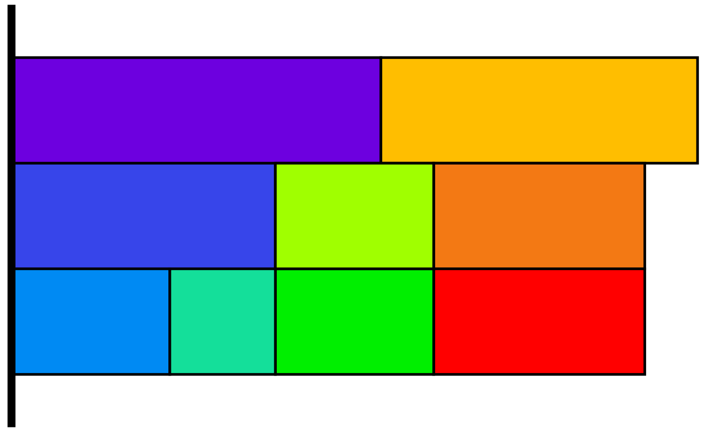
- For m part of the input, the problem is NP-hard in the strong sense.
- For constant $m \geqslant 2$, the problem is NP-hard but solvable in pseudo-polynomial time.
- For $m = 1$, the problem can be solved in polynomial time by Smith's Ratio Rule: *sequence the jobs in order of nonincreasing ratios* $\frac{w_j}{p_j}$



Color corresponds to ratio $\frac{w_j}{p_j}$ and thus indicates the importance of job j

Known approximation results

List scheduling in order of nonincreasing ratios $\frac{w_j}{p_j}$ on identical parallel machines:



Kawaguchi & Kyan '86:
The value of this schedule is at most $\frac{1}{2}(1+\sqrt{2}) \approx 1.21$ times the optimum and this bound is tight.
(Best known approximation result for many years.)

Sahni '76:
Fully polynomial-time approximation scheme for constant number m of machines.

POLYNOMIAL-TIME APPROXIMATION SCHEMES

A ρ–approximation algorithm is a polynomial-time algorithm which always produces a solution of value within ρ times the value of an optimal solution. The value ρ is called performance ratio of the algorithm.

A family of polynomial-time approximation algorithms with performance ratio $1+\varepsilon$ for all fixed $\varepsilon > 0$ is called a polynomial-time approximation scheme (PTAS).

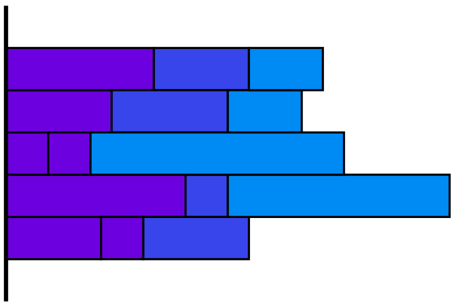
If the running times are even bounded by a polynomial in the input size and $\frac{1}{\varepsilon}$, then these algorithms build a fully polynomial time approximation scheme (FPTAS).

A PTAS for $P||\sum w_j C_j$

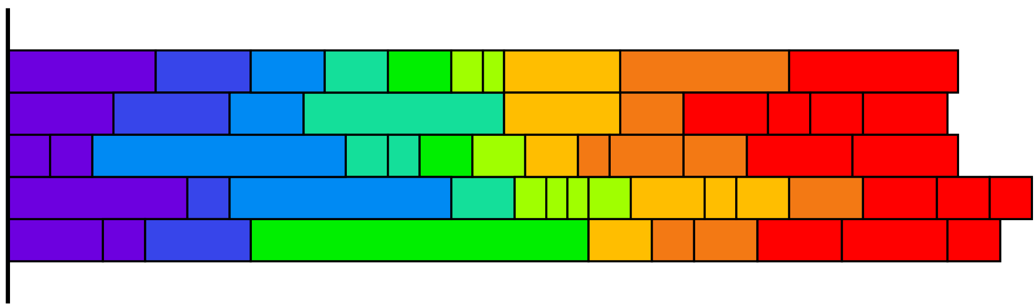
Step 1:
PTAS for a constant range of ratios:

$$\delta \leqslant \frac{w_j}{p_j} / \frac{w_k}{p_k} \leqslant \frac{1}{\delta}$$

for some constant $\delta > 0$ and for all jobs j, k.



Step 2:
PTAS for instances with arbitrary ratios $\frac{w_j}{p_j}$.

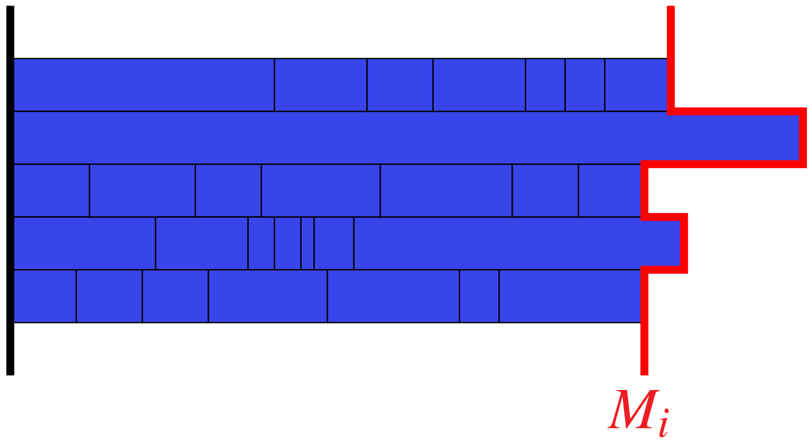


The result of AAWY '98

Alon, Azar, Woeginger, & Yadid '98 give a PTAS for the problem:

$$\text{minimize} \quad \sum_{i=1}^{m} M_i^2$$

where M_i denotes the finishing time of machine i.



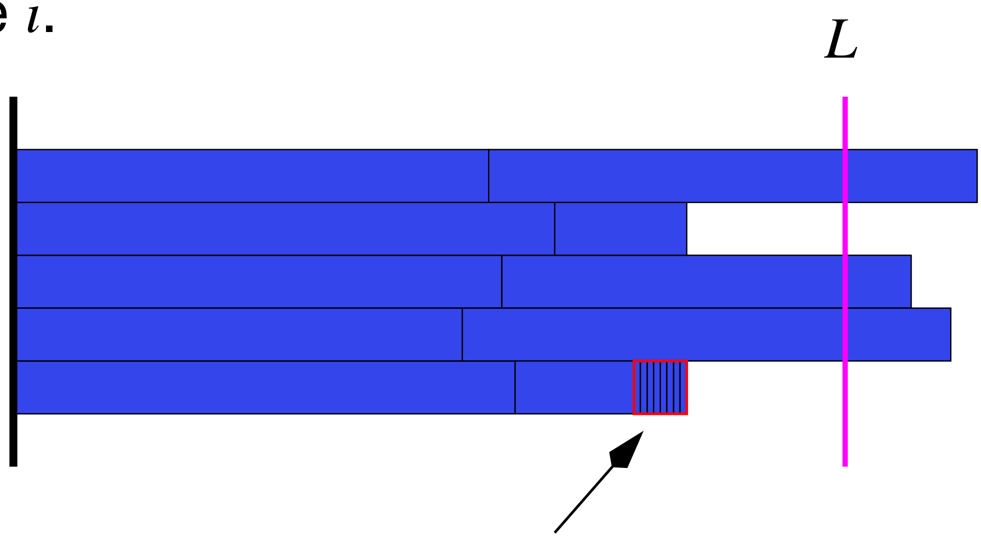
Observation: (Eastman, Even, & Isaacs '64)
If $\frac{w_j}{p_j} = 1$ for all jobs j, then

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

$\Longrightarrow$ PTAS for the case $\frac{w_j}{p_j} = c$ (constant) for all jobs j.

THE TECHNIQUE OF AAWY '98

- Eliminate huge jobs that are bigger than the average machine load $L := \frac{1}{m}\sum_{j=1}^{n} p_j$.
- In an optimal schedule $\frac{1}{2}L < M_i < 2L$ for each machine i.



- Cluster the tiny jobs to small jobs whose length is at least a constant fraction of L.

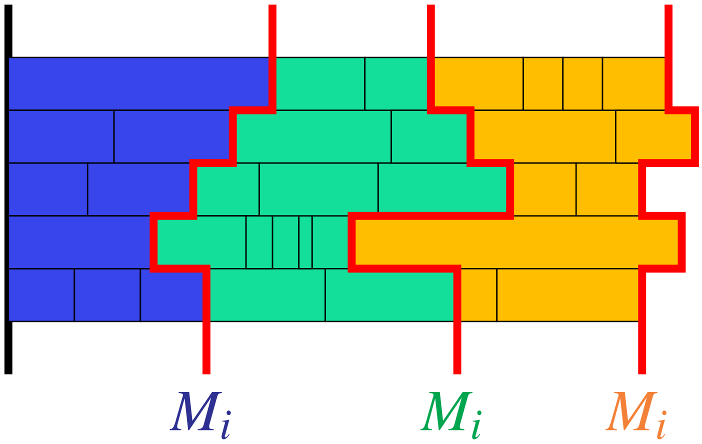
  $\Longrightarrow$ Only constant number of different job sizes.

  $\Longrightarrow$ Only constant number of possible schedules on each machine!

- The transformed instance can be solved in polynomial time, e. g., by dynamic programming (or integer linear programming in constant dimension).

The case of several ratios

Several ratios $\frac{w_j}{p_j} \in \{\gamma, \gamma, \gamma\}$ with $\gamma > \gamma > \gamma$:



The value of the schedule only depends on M_i, M_i, M_i:

$$\sum_{j=1}^{n} w_j C_j =$$

$$\frac{1}{2}\sum_{j=1}^{n} w_j p_j + \frac{\gamma}{2}\sum_{i=1}^{m} M_i^2 + \frac{\gamma - \gamma}{2}\sum_{i=1}^{m} M_i^2 + \frac{\gamma - \gamma}{2}\sum_{i=1}^{m} M_i^2$$

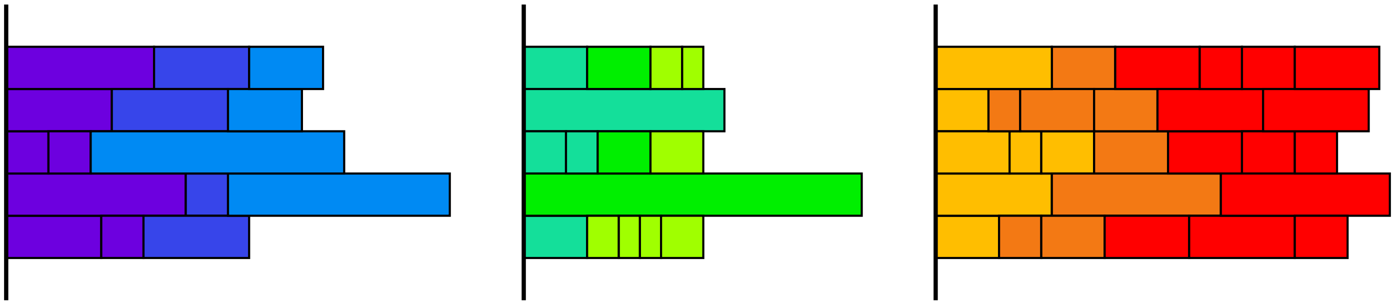
A generalization and modification of the technique of Alon et al. leads to a PTAS for the case of a constant range of ratios.

A PTAS for the general case

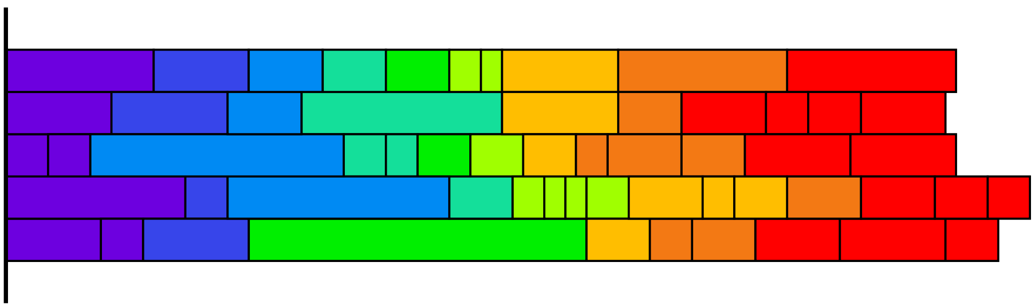
Main steps:

Partition the jobs into subsets according to ratios:

Compute near-optimal schedules for the subsets:



Concatenate those schedules “appropriately”:

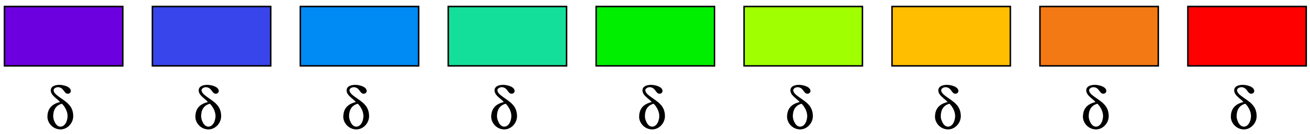


The aim:

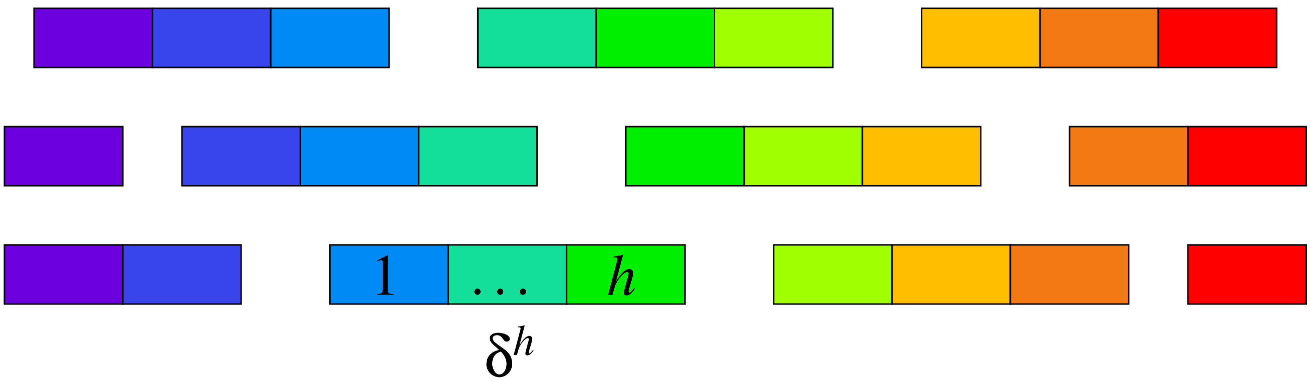
- Ratios of jobs are within constant range in each subset.
- Ratios of jobs in different subsets differ a lot.

The method:

- Start with fine partition of constant range δ:



- Turn it randomly into rougher partition of range δ^h by clustering groups of h subsets (for constant h):



The probability for two jobs with similar ratios to lie in different subsets is at most $\frac{1}{h}$ (small).

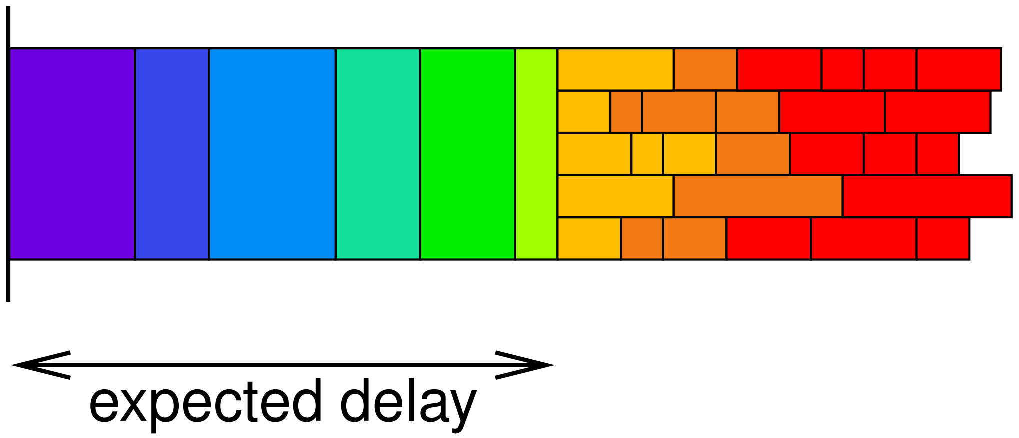
THE CONCATENATION STEP

The aim:

- Keep the machine loads balanced.
- In the analysis, we want to neglect the delay of jobs caused by more important jobs in "earlier" subsets.

The method:

- Randomly permute the machines in the schedule of each subset before concatenating.



The expected delay of the red jobs is the average machine load caused by the blue and green jobs.

THE ANALYSIS

Lemma: The value of an optimal schedule is at least the sum of the values of optimal schedules for all subsets of jobs.

$\Longrightarrow$ if we start with $(1+\varepsilon)$–approximations for the subsets, the sum is at most $(1+\varepsilon)opt$.

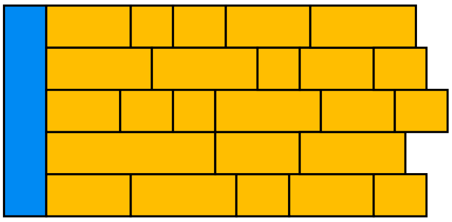
$\Longrightarrow$ The value of the computed schedule is at most $(1+\varepsilon)opt$ plus the additional cost caused by the delay of jobs in the concatenation step.

Theorem: The cost caused in the concatenation step can be neglected — it is in $O(\varepsilon)opt$.

Sketch of proof

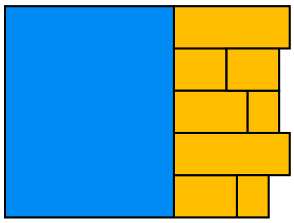
Consider the delay of a subset of jobs by another, more important subset:

Case 1: The ratios differ a lot and the delayed subset is “large”:



The delay can be neglected compared to the cost of the orange schedule.

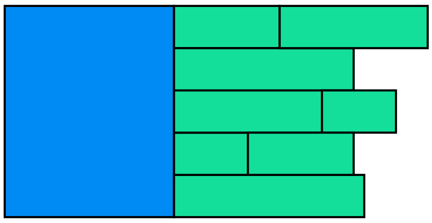
Case 2: The ratios differ a lot and the delayed subset is “small”:



The delay can be neglected compared to the cost of the blue schedule.

Sketch of proof (cont.)

Case 3: The ratios are similar:



The delay is of the same order as the sum of the cost of the blue and the green schedule; however, this case happens with negligibly small probability.

Even the sum of the delays over all pairs of subsets can be neglected; since the ratios of jobs are geometrically decreasing with the subsets, the sum can be bounded by geometric series.

q.e.d.

CONCLUDING REMARKS

- The algorithm can easily be derandomized.
- This is the first PTAS for a strongly NP-hard scheduling problem with minsum objective.
- Hochbaum & Shmoys '87 gave a PTAS for the problem to minimize the makespan (strongly NP-hard).
- Hoogeveen, Schuurman, & Woeginger showed that $R|r_j|\sum C_j$ cannot be approximated within arbitrary precision in polynomial time.
- Recently, several groups derived independently a PTAS for $1|r_j|\sum C_j$ and for more general problems. (Chekuri & Khanna; Karger & Stein; Queyranne & Sviridenko; Sk.)

  There exist PTASes for:

  - $P|r_j|\sum w_j C_j$ and $P|r_j, pmtn|\sum w_j C_j$,
  - $Rm|r_j|\sum w_j C_j$ and $Rm|r_j, pmtn|\sum w_j C_j$.