Combinatorial Algorithms for Minimizing the Weighted Sum of Completion Times on a Single Machine

James Davis [1]   Rajiv Gandhi[2]   Vijay Khothari

[1]Department of Operations Research
Cornell University
jamesmariodavis@gmail.com

[2]Department of Computer Science
Rutgers University - Camden
rajivg@camden.rutgers.edu

November 8, 2010

Single Machine Scheduling (Off-line Problem)

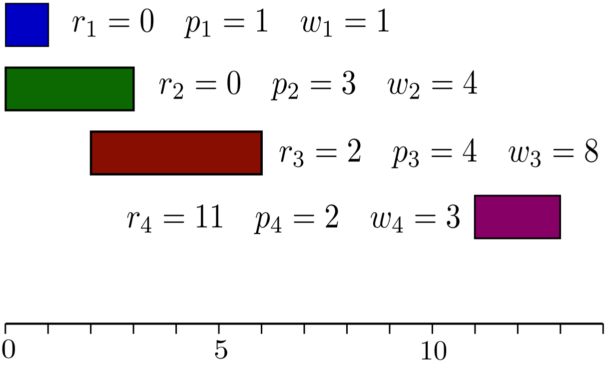
Input:

- J, set of jobs
- p_j, processing times
- $r_j \geq 0$, release dates
- w_j, weights

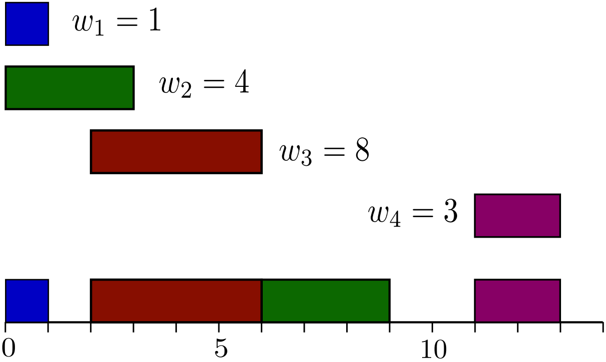
Goal:

- Schedule jobs on a single machine
- Non-preemptive schedule
- No job can be scheduled before it's release date
- Minimize $\sum_j w_j C_j$, weighted sum of completion times

Single Machine Scheduling (Off-line Problem)



Single Machine Scheduling (Off-line Problem)



Cost $= 1 \cdot 1 + 6 \cdot 8 + 9 \cdot 4 + 13 \cdot 3$

Single Machine Scheduling (Online Problem)

Input:

- Jobs, J
- $p_j, r_j, w_j \quad \forall j \in J$

Goal:

- Schedule jobs on a single machine
- Non-preemptive schedule
- Minimize $\sum_j w_j C_j$, weighted sum of completion times
- Aware of job j at time r_j
- Schedule fixed at time t without knowledge of jobs s.t. $r_j > t$

LP Formulation

Notation ($S \subseteq J$):

- Sum of processing times

$$p(S) = \sum_{j \in S} p_j$$

- Sum of squared processing times,

$$p^2(S) = \sum_{j \in S} p_j^2$$

Simple LP Formulation

C_j denotes the completion time of job j

$$
\begin{aligned}
\min \quad & \sum_{j \in J} w_j C_j \\
\text{subject to} \quad & C_j \geq r_j + p_j, && \forall j \in J \\
& \sum_{j \in S} p_j C_j \geq \frac{p(S)^2 + p^2(S)}{2}, && \forall S \subseteq J \\
& C_j \geq 0, && \forall j \in J
\end{aligned}
$$

Previous Work

Single Machine Scheduling:

- NP-Hard (Lenstra et al.)
- Off-line PTAS known (Afrati et al.)
- Off-line 1.6853-Appx. Alg. via LP rounding (Goemans et al.)
- Online 1.6853-Appx. Alg. via LP rounding (Goemans et al.)

Using Simple LP (provide upper bound on int. gap):

- Off-line 3-Appx. Alg. via LP rounding (Hall et al.)

Our Contribution

Use Simple LP (provide upper bound on int. gap):

- Off-line $(1+\sqrt{2})(\approx 2.42)$-Approximation Algorithm
- Online 3-Approximation Algorithm

Main Criteria

Recall Notation:

- $p(S) = \sum_{j \in S} p_j$

New Notation:

- $j^* \in J$ has highest release date (r_j value)
- $j' \in J$ has lowest $\frac{w_j}{p_j}$ value

Main Criteria

$$r_{j^*} > p(J)$$

List Algorithm

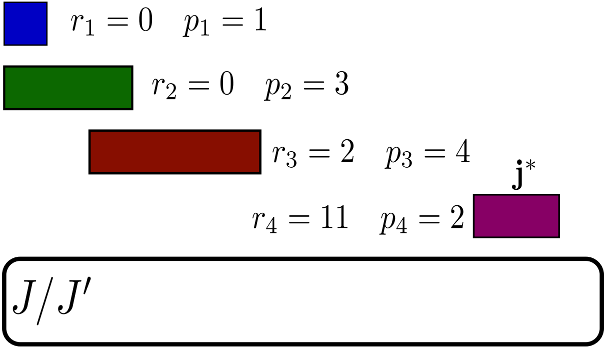
Off-line List Algorithm

$J' \leftarrow J$
while $J' \neq \emptyset$ **do**
  $j^* \leftarrow$ job with largest r_j value
  if $r_{j^*} > p(J')$ **then**
    Remove j^* from J'
  else if $r_{j^*} \leq p(J')$ **then**
    $j' \leftarrow j \in J'$ with lowest $\frac{w_j}{p_j}$ value
    Remove j' from J'
  end if
end while
Schedule jobs in the reverse order that they were removed from J'

Intuitive Analysis

If $r_{j^*} > p(J')$ then C_{j^*} is approximately dominated by r_{j^*}

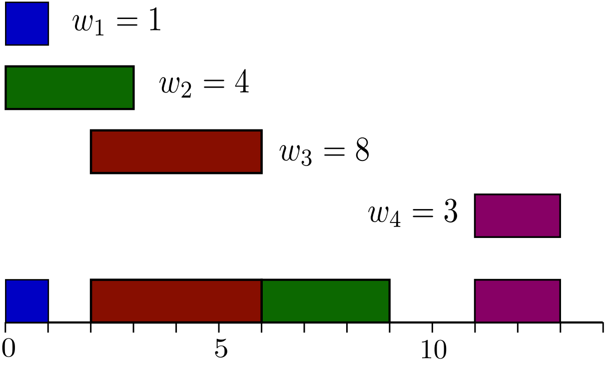
$$r_{j^*} = 11 > 10 = p(J')$$



Intuitive Analysis

If $r_{j^*} > p(J')$ then C_{j^*} is approximately dominated by r_{j^*}

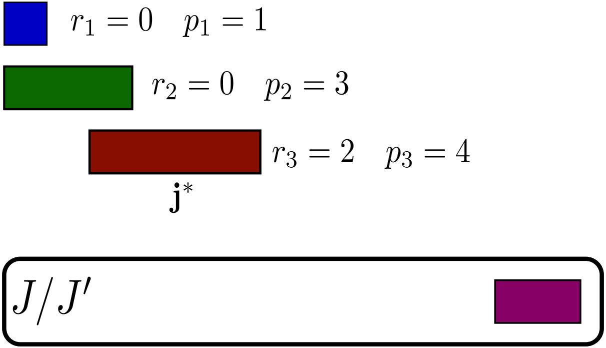
$$r_{j^*} = 11 > 10 = p(J')$$



Intuitive Analysis

If $r_{j^*} \leq p(J')$ then C_{j^*} is approximately dominated by $p(J')$

$$r_{j^*} = 2 \leq 8 = p(J')$$

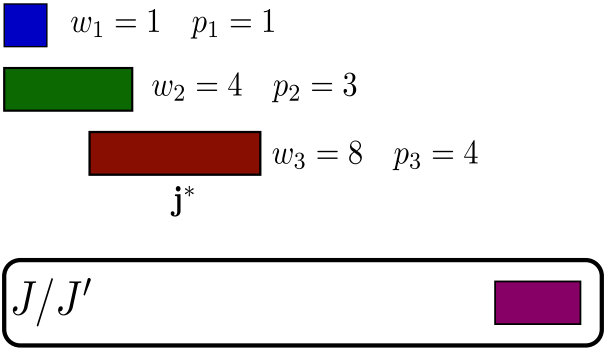


Intuitive Analysis

Smith's Rule

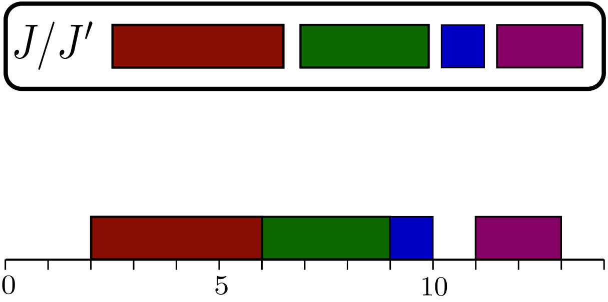
Schedule the most useless job last (lowest $\frac{w_j}{p_j}$)

j' is job 1 ($\frac{w_1}{p_1} = 1$)



Intuitive Analysis

Final Schedule



Simple LP Formulation

C_j denotes the completion time of job j

$$
\begin{aligned}
\min \quad & \sum_{j \in J} w_j C_j \\
\text{subject to} \quad & C_j \geq r_j + p_j, && \forall j \in J \\
& \sum_{j \in S} p_j C_j \geq \frac{p(S)^2 + p^2(S)}{2}, && \forall S \subseteq J \\
& C_j \geq 0, && \forall j \in J
\end{aligned}
$$

Simple Dual

Introduce $\beta_S \quad \forall S \subseteq J$ and $\alpha_j \quad \forall j \in J$

$$\begin{aligned}
\max \; & \sum_{j \in J} \alpha_j (r_j + p_j) + \sum_{S \subseteq J} \beta_S \left(\frac{p(S)^2 + p^2(S)}{2} \right) & \\
\text{subject to} \; & \alpha_j + p_j \sum_{S : j \in S} \beta_S \leq w_j, & \forall j \in J \\
& \alpha_j \geq 0, & \forall j \in J \\
& \beta_S \geq 0, & \forall S \subseteq J
\end{aligned}$$

Primal-Dual Algorithm

Simplified Setting:

- Single Iteration (job set J)
- Increase α_{j^*} or β_J

$r_{j^*} > p(J')$

- Increase α_{j^*}
- $\alpha_{j^*} = w_{j^*}$

$r_{j^*} \leq p(J')$

- Increase β_J
- $\beta_J = \frac{w_{j'}}{p_{j'}}$

Dual Constraint

$$\alpha_j + p_j \sum_{S: j \in S} \beta_S \leq w_j$$

Primal-Dual Algorithm

Off-line Primal-Dual Algorithm

$J' \leftarrow J$
while $J' \neq \emptyset$ **do**
  $j^* \leftarrow$ job with largest r_j value
  if $r_{j^*} > p(J')$ **then**

$$\alpha_{j^*} \leftarrow w_{j^*} - p_{j^*} \sum_{S: j^* \in S} \beta_S$$

    Remove j^* from J'
  else if $r_{j^*} \leq p(J')$ **then**
    $j' \leftarrow j \in J'$ with lowest $\frac{w_j}{p_j}$ value
    $\beta_{J'} \leftarrow \frac{w_{j'}}{p_{j'}} - \sum_{S: j \in S} \beta_S$
    Remove j' from J'
  end if
end while
Schedule jobs in the reverse order that they were removed from J'

Primal-Dual Algorithm

Theorem

Using the off-line primal dual algorithm:

$$\text{Cost of Schedule} \;\leq (1+\sqrt{2})\cdot \;\text{Dual Feasible Solution} \;\leq (1+\sqrt{2})\cdot \;\text{OPT}$$

Thank You!