Improved Approximation Algorithms for Non-Preemptive Scheduling Problems via Time-Indexed LP Relaxations

Shi Li, *University at Buffalo*

Sep 12, 2017, UC Berkeley

Outline

1. Introduction
2. 1.8786-Approximation for $R|r_j|\sum_j w_j C_j$
3. $(2 + 2\ln 2)$-Approximation for $P|\text{prec}|\sum_j w_j C_j$

Non-Preemptive Scheduling Problems to Minimize Weighted Completion Time

- set J of jobs, set M of machines, w_j = weight of job j

Non-Preemptive Scheduling Problems to Minimize Weighted Completion Time

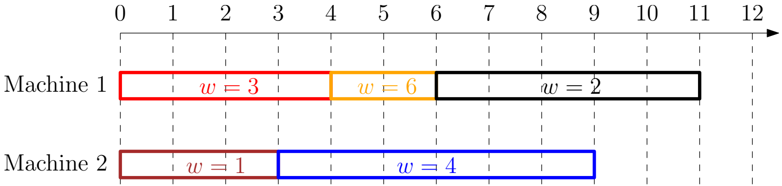
- set J of jobs, set M of machines, $w_j =$ weight of job j
- schedule J non-preemptively on M to minimize

$$\sum_{j \in J} w_j C_j, \text{ where } C_j = \text{completion time of } j.$$

Non-Preemptive Scheduling Problems to Minimize Weighted Completion Time

- set J of jobs, set M of machines, w_j = weight of job j
- schedule J non-preemptively on M to minimize

$$\sum_{j \in J} w_j C_j, \text{ where } C_j = \text{completion time of } j.$$



$$\sum_{j \in J} w_j C_j = 3 \times 4 + 6 \times 6 + 2 \times 11 + 1 \times 3 + 4 \times 9 = 109$$

$\alpha|\beta|\gamma$-Notation of Graham et al.

$\alpha|\beta|\gamma$-Notation of Graham et al.

α denotes machine model:

$\alpha|\beta|\gamma$-Notation of Graham et al.

α denotes machine model:

- P: identical machines, p_j = size of j

$\alpha|\beta|\gamma$-Notation of Graham et al.

α denotes machine model:

- P: identical machines, p_j = size of j
- R: unrelated machines, $p_{i,j}$ = size of j on machine i

$\alpha|\beta|\gamma$-Notation of Graham et al.

α denotes machine model:

- P: identical machines, $p_j =$ size of j
- R: unrelated machines, $p_{i,j} =$ size of j on machine i
- Q: related machines, $p_{i,j} = p_j / s_i$

$\alpha|\beta|\gamma$-Notation of Graham et al.

α denotes machine model:

- P: identical machines, p_j = size of j
- R: unrelated machines, $p_{i,j}$ = size of j on machine i
- Q: related machines, $p_{i,j} = p_j / s_i$

β denotes additional constraints:

$\alpha|\beta|\gamma$-Notation of Graham et al.

α denotes machine model:

- P: identical machines, $p_j =$ size of j
- R: unrelated machines, $p_{i,j} =$ size of j on machine i
- Q: related machines, $p_{i,j} = p_j / s_i$

β denotes additional constraints:

- r_j: each job j has an arrival time r_j

$\alpha|\beta|\gamma$-Notation of Graham et al.

α denotes machine model:

- P: identical machines, $p_j =$ size of j
- R: unrelated machines, $p_{i,j} =$ size of j on machine i
- Q: related machines, $p_{i,j} = p_j / s_i$

β denotes additional constraints:

- r_j: each job j has an arrival time r_j
- prec: precedence constraints,
  $j \prec j'$ requires j' to start after j completes

$\alpha|\beta|\gamma$-Notation of Graham et al.

α denotes machine model:

- P: identical machines, p_j = size of j
- R: unrelated machines, $p_{i,j}$ = size of j on machine i
- Q: related machines, $p_{i,j} = p_j / s_i$

β denotes additional constraints:

- r_j: each job j has an arrival time r_j
- prec: precedence constraints,
  $j \prec j'$ requires j' to start after j completes
- $p_j = 1$: all jobs have the same size

$\alpha|\beta|\gamma$-Notation of Graham et al.

α denotes machine model:

- P: identical machines, $p_j =$ size of j
- R: unrelated machines, $p_{i,j} =$ size of j on machine i
- Q: related machines, $p_{i,j} = p_j / s_i$

β denotes additional constraints:

- r_j: each job j has an arrival time r_j
- prec: precedence constraints,
  $j \prec j'$ requires j' to start after j completes
- $p_j = 1$: all jobs have the same size

γ denotes objective:

$\alpha|\beta|\gamma$-Notation of Graham et al.

α denotes machine model:

- P: identical machines, p_j = size of j
- R: unrelated machines, $p_{i,j}$ = size of j on machine i
- Q: related machines, $p_{i,j} = p_j/s_i$

β denotes additional constraints:

- r_j: each job j has an arrival time r_j
- prec: precedence constraints,
  $j \prec j'$ requires j' to start after j completes
- $p_j = 1$: all jobs have the same size

γ denotes objective:

- in this talk it is always $\sum_j w_j C_j$

List of Our Results

Problems	Prev. approx. ratio	Our approx. ratio
$P\|\text{prec}\|\sum_j w_jC_j$	4 [1]	$2+2\ln 2$
$P\|\text{prec}, p_j=1\|\sum_j w_jC_j$	3 [1]	$1+\sqrt{2}$
$Q\|\text{prec}\|\sum_j w_jC_j$	$O(\log m)$ [2]	$O(\frac{\log m}{\log\log m})$
$R\|\|\sum_j w_jC_j$	$1.5-\epsilon$ [3]	$1.5-\frac{1}{6000}$
$R\|r_j\|\sum_j w_jC_j$	2 [4]	1.8786

1 [Munier-Queyranne-Schulz' 98]

2 [Chudak-Shmoys' 97]

3 [Bansal-Srinivasan-Svensson' 16]

4 [Skutella' 01], [Schulz and Skutella' 02]

List of Our Results

Problems	Prev. approx. ratio	Our approx. ratio
$P\|\text{prec}\|\sum_j w_j C_j$	4 [1]	$2 + 2\ln 2$
$P\|\text{prec}, p_j = 1\|\sum_j w_j C_j$	3 [1]	$1 + \sqrt{2}$
$Q\|\text{prec}\|\sum_j w_j C_j$	$O(\log m)$ [2]	$O(\frac{\log m}{\log\log m})$
$R\|\|\sum_j w_j C_j$	$1.5 - \epsilon$ [3]	$1.5 - \frac{1}{6000}$
$R\|r_j\|\sum_j w_j C_j$	2 [4]	1.8786

1 [Munier-Queyranne-Schulz' 98]

2 [Chudak-Shmoys' 97]

3 [Bansal-Srinivasan-Svensson' 16]

4 [Skutella' 01], [Schulz and Skutella' 02]

- Based on [Im-Li, FOCS 2016] and [Li, FOCS 2017]

List of Our Results

Problems	Prev. approx. ratio	Our approx. ratio
$P\|\text{prec}\|\sum_j w_jC_j$	4 [1]	$2+2\ln 2$
$P\|\text{prec}, p_j=1\|\sum_j w_jC_j$	3 [1]	$1+\sqrt{2}$
$Q\|\text{prec}\|\sum_j w_jC_j$	$O(\log m)$ [2]	$O(\frac{\log m}{\log\log m})$
$R\|\|\sum_j w_jC_j$	$1.5-\epsilon$ [3]	$1.5-\frac{1}{6000}$
$R\|r_j\|\sum_j w_jC_j$	2 [4]	1.8786

1 [Munier-Queyranne-Schulz' 98]

2 [Chudak-Shmoys' 97]

3 [Bansal-Srinivasan-Svensson' 16]

4 [Skutella' 01], [Schulz and Skutella' 02]

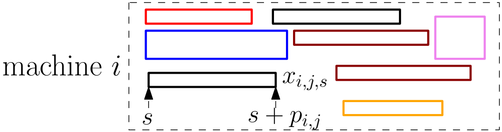
- Based on [Im-Li, FOCS 2016] and [Li, FOCS 2017]

Main Theme: Time-Indexed LP Relaxation

- $x_{i,j,s}$, $i \in M, j \in J, s \geq r_j$: whether job j is scheduled on i with starting time s

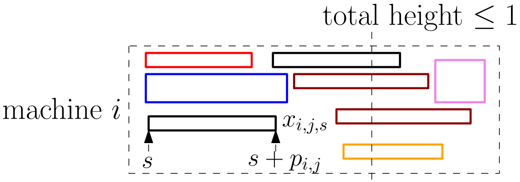
Main Theme: Time-Indexed LP Relaxation

- $x_{i,j,s}$, $i \in M, j \in J, s \geq r_j$: whether job j is scheduled on i with starting time s



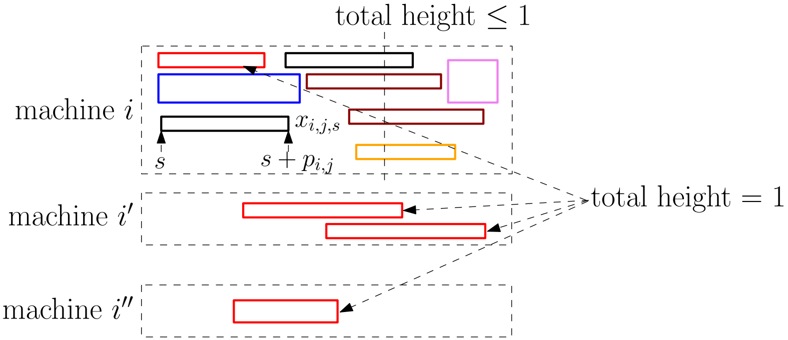
Main Theme: Time-Indexed LP Relaxation

- $x_{i,j,s}$, $i \in M, j \in J, s \geq r_j$: whether job j is scheduled on i with starting time s



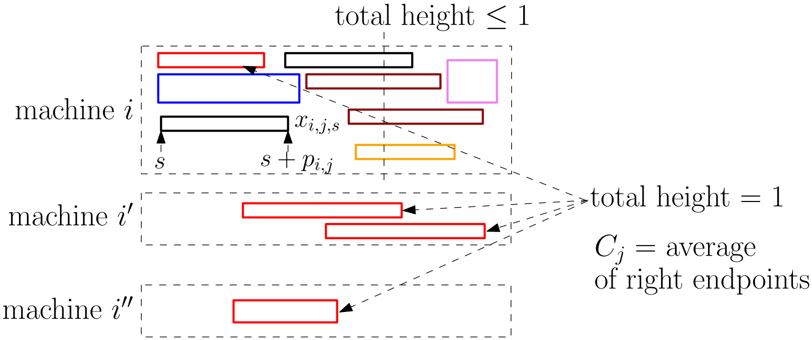
Main Theme: Time-Indexed LP Relaxation

- $x_{i,j,s}$, $i \in M, j \in J, s \geq r_j$: whether job j is scheduled on i with starting time s



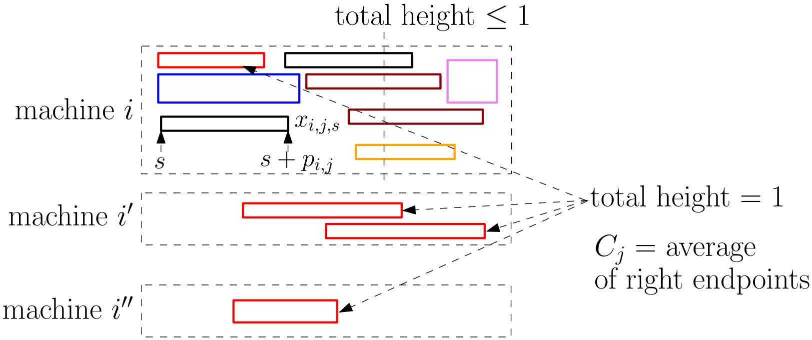
Main Theme: Time-Indexed LP Relaxation

- $x_{i,j,s}$, $i \in M, j \in J, s \geq r_j$: whether job j is scheduled on i with starting time s



Main Theme: Time-Indexed LP Relaxation

- $x_{i,j,s}$, $i \in M, j \in J, s \geq r_j$: whether job j is scheduled on i with starting time s



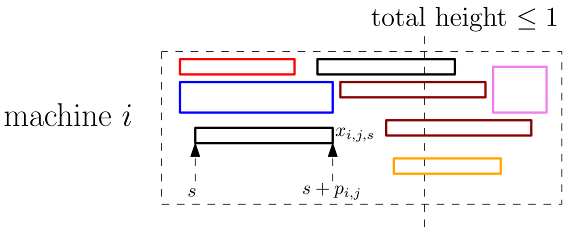
- to capture $j \prec j'$: $C_{j'} \geq C_j + p_{j'}$

Outline

1. Introduction
2. 1.8786-Approximation for $R|r_j|\sum_j w_j C_j$
3. $(2 + 2\ln 2)$-Approximation for $P|\text{prec}|\sum_j w_j C_j$

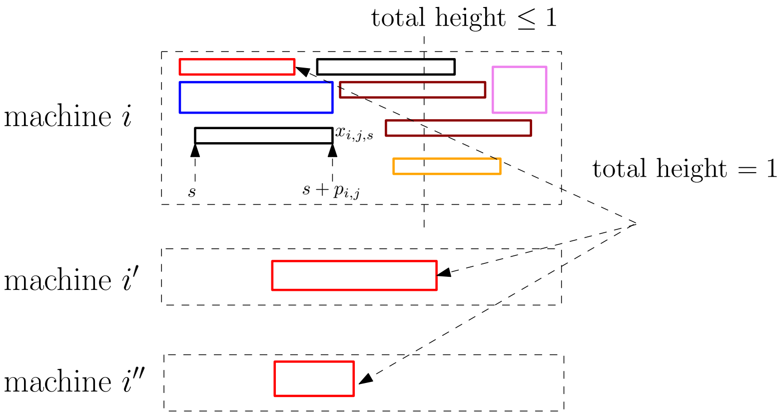
Recall the problem $R|r_j|\sum_j w_jC_j$

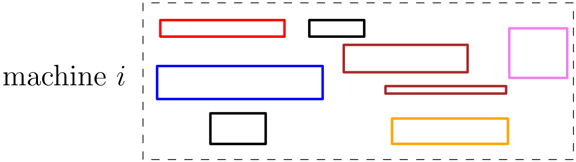
- R: unrelated machine scheduling
- r_j: job j has arrival time r_j



Recall the problem $R|r_j|\sum_j w_j C_j$

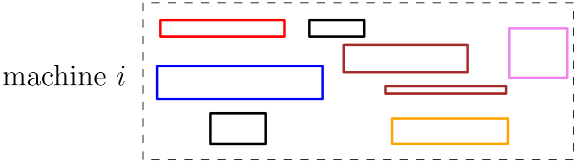
- R: unrelated machine scheduling
- r_j: job j has arrival time r_j





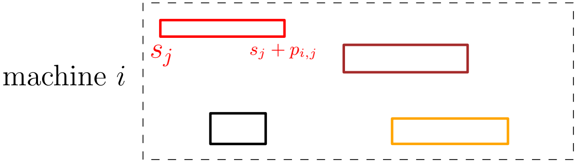
2-Approx. Algorithm

1. solve time-indexed LP relaxation to obtain $\{x_{i,j,s}\}$



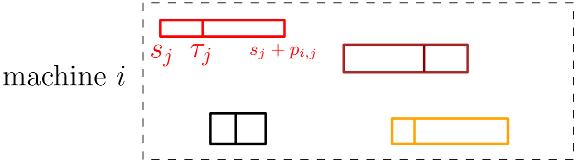
2-Approx. Algorithm

1. solve time-indexed LP relaxation to obtain $\{x_{i,j,s}\}$
2. for each $j \in J$, independently do:



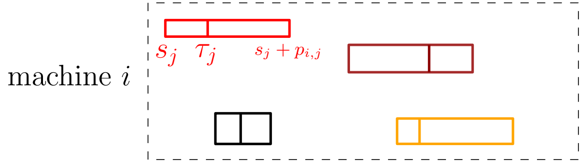
2-Approx. Algorithm

1. solve time-indexed LP relaxation to obtain $\{x_{i,j,s}\}$
2. for each $j \in J$, independently do:
3. let $(i_j, s_j) = (i, s)$ with probability $x_{i,j,s}$



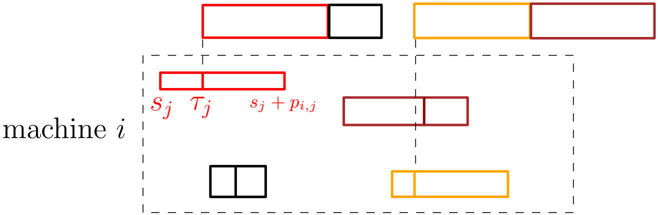
2-Approx. Algorithm

1. solve time-indexed LP relaxation to obtain $\{x_{i,j,s}\}$
2. for each $j \in J$, independently do:
3. let $(i_j, s_j) = (i, s)$ with probability $x_{i,j,s}$
4. choose τ_j from $[s_j, s_j + p_{i_j,j}]$ uniformly at random



2-Approx. Algorithm

1. solve time-indexed LP relaxation to obtain $\{x_{i,j,s}\}$
2. for each $j \in J$, independently do:
3. let $(i_j, s_j) = (i, s)$ with probability $x_{i,j,s}$
4. choose τ_j from $[s_j, s_j + p_{i_j,j}]$ uniformly at random
5. for each $i \in M$, schedule jobs assigned to i in increasing order of τ_j, pretending τ_j's are arrival times



2-Approx. Algorithm

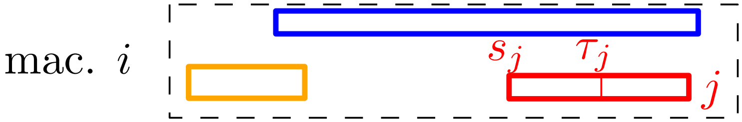
1. solve time-indexed LP relaxation to obtain $\{x_{i,j,s}\}$
2. for each $j \in J$, independently do:
3. let $(i_j, s_j) = (i, s)$ with probability $x_{i,j,s}$
4. choose τ_j from $[s_j, s_j + p_{i_j,j}]$ uniformly at random
5. for each $i \in M$, schedule jobs assigned to i in increasing order of τ_j, pretending τ_j's are arrival times

Proof of 2-Approximation

Lemma $\mathbb{E}[C_j] \leq 2 \sum_{i,s} x_{i,s}(s + p_{i,j})$

Proof of 2-Approximation

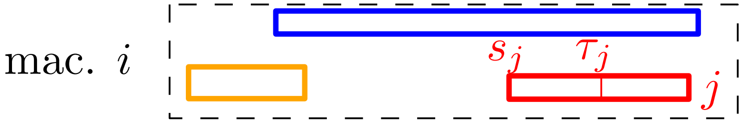
Lemma $\mathbb{E}[C_j] \le 2\sum_{i,s} x_{i,s}(s + p_{i,j})$



- Fixing $j, i_j = i, s_j$ and τ_j and condition on them

Proof of 2-Approximation

Lemma $\mathbb{E}[C_j] \le 2\sum_{i,s} x_{i,s}(s + p_{i,j})$

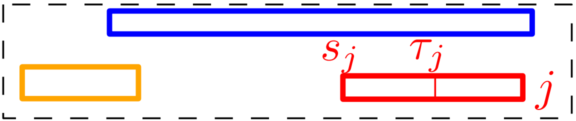


- Fixing $j, i_j = i, s_j$ and τ_j and condition on them
- $C_j = \sum_{j' \ne j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \big(|\text{idle time}| \text{ before } \tau_j\big) + p_{i,j}$

Proof of 2-Approximation

Lemma $\mathbb{E}[C_j] \leq 2\sum_{i,s} x_{i,s}(s + p_{i,j})$

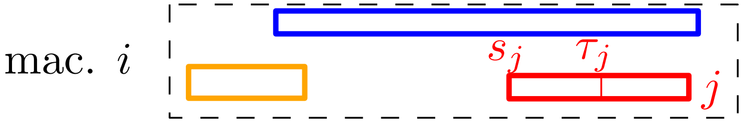
mac. i



- Fixing $j, i_j = i, s_j$ and τ_j and condition on them
- $C_j = \sum_{j' \neq j: i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j\right) + p_{i,j}$
- $\mathbb{E}[\text{first term} | i_j = i, s_j, \tau_j] \leq \tau_j$

Proof of 2-Approximation

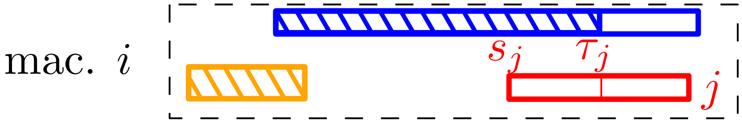
Lemma $\mathbb{E}[C_j] \le 2\sum_{i,s} x_{i,s}(s + p_{i,j})$



- Fixing $j, i_j = i, s_j$ and τ_j and condition on them
- $C_j = \sum_{j' \neq j: i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j\right) + p_{i,j}$
- $\mathbb{E}[\text{first term} | i_j = i, s_j, \tau_j] \le \tau_j$ second term $\le \tau_j$

Proof of 2-Approximation

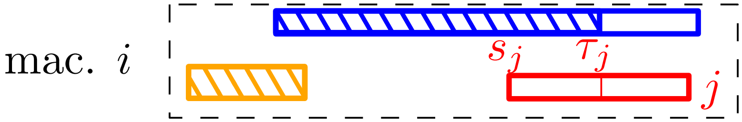
Lemma $\mathbb{E}[C_j] \le 2\sum_{i,s} y_{i,s}(s + p_{i,j})$



- Fixing $j, i_j = i, s_j$ and τ_j and condition on them
- $\mathbb{E}[C_j | i_j = i, s_j, \tau_j] \le 2\tau_j + p_{i,j}$

Proof of 2-Approximation

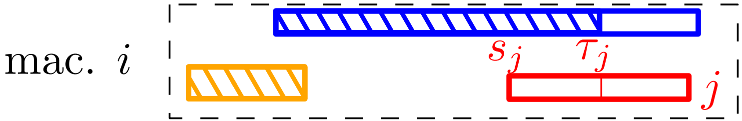
Lemma $\mathbb{E}[C_j] \leq 2\sum_{i,s} y_{i,s}(s + p_{i,j})$



- Fixing $j, i_j = i, s_j$ and τ_j and condition on them
- $\mathbb{E}[C_j | i_j = i, s_j, \tau_j] \leq 2\tau_j + p_{i,j}$
- $\mathbb{E}[\tau_j | i_j = i, s_j] = s_j + p_{i,j}/2$

Proof of 2-Approximation

Lemma $\mathbb{E}[C_j] \le 2\sum_{i,s} y_{i,s}(s + p_{i,j})$

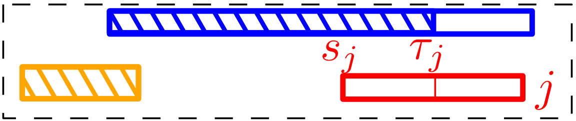


- Fixing $j, i_j = i, s_j$ and τ_j and condition on them
- $\mathbb{E}[C_j | i_j = i, s_j, \tau_j] \le 2\tau_j + p_{i,j}$
- $\mathbb{E}[\tau_j | i_j = i, s_j] = s_j + p_{i,j}/2$
- $\mathbb{E}[C_j | i_j = i, s_j] \le 2(s_j + p_{i,j}/2) + p_{i,j} = 2(s_j + p_{i,j})$

Proof of 2-Approximation

Lemma $\mathbb{E}[C_j] \le 2\sum_{i,s} y_{i,s}(s + p_{i,j})$

mac. i



- Fixing $j, i_j = i, s_j$ and τ_j and condition on them
- $\mathbb{E}[C_j | i_j = i, s_j, \tau_j] \le 2\tau_j + p_{i,j}$
- $\mathbb{E}[\tau_j | i_j = i, s_j] = s_j + p_{i,j}/2$
- $\mathbb{E}[C_j | i_j = i, s_j] \le 2(s_j + p_{i,j}/2) + p_{i,j} = 2(s_j + p_{i,j})$
- $\mathbb{E}[C_j] \le 2\sum_{i,s} x_{i,j,s}(s + p_{i,j})$

$$C_j = \sum_{j' \neq j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j\right) + p_{i,j}$$

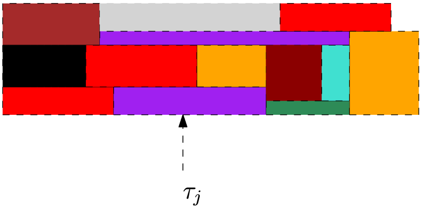
- Assuming machine i is “fully packed”

$$C_j = \sum_{j' \neq j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j\right) + p_{i,j}$$

- Assuming machine i is “fully packed”
- Bounding $\left(|\text{idle time}| \text{ before } \tau_j\right)$ by τ_j too pessimistic?

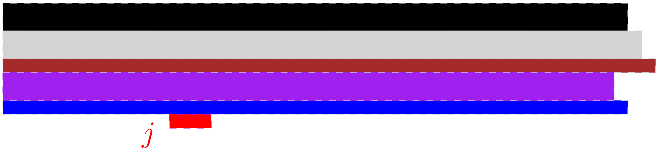
$$C_j = \sum_{j' \neq j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j\right) + p_{i,j}$$

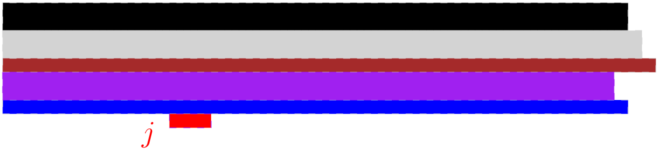
- Assuming machine i is "fully packed"
- Bounding $\left(|\text{idle time}| \text{ before } \tau_j\right)$ by τ_j too pessimistic?
- If many rectangles have small length: many jobs are before τ_j with high prob. $\Rightarrow$ some busy intervals



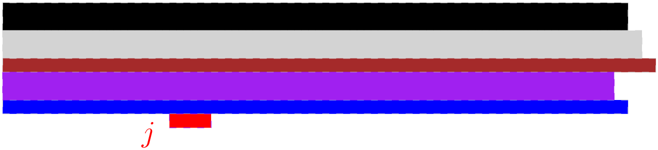
$$C_j = \sum_{j' \neq j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j\right) + p_{i,j}$$

- Assuming machine i is "fully packed"
- Bounding $\left(|\text{idle time}| \text{ before } \tau_j\right)$ by τ_j too pessimistic?
- If many rectangles have small length: many jobs are before τ_j with high prob. $\Rightarrow$ some busy intervals
- The only bad case: most rectangles have length $>> \tau_j$



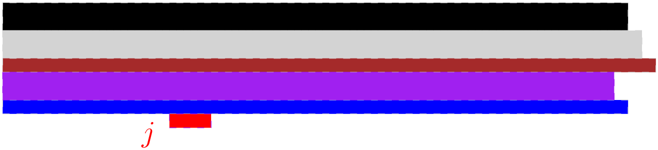


Improving the Factor of 2



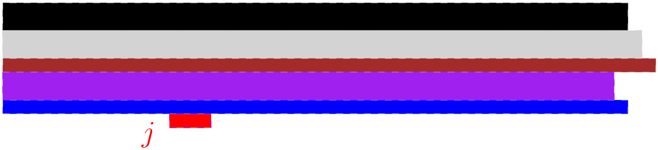
Improving the Factor of 2

- Choose each τ_j from a different distribution (instead of uniformly between s_j and $s_j + p_{i_j,j}$)
  - Shift mass from left to center: long jobs are less likely to delay j



Improving the Factor of 2

- Choose each τ_j from a different distribution (instead of uniformly between s_j and $s_j + p_{i_j,j}$)
  - Shift mass from left to center: long jobs are less likely to delay j
  - Shift mass from right to center: not increase $\mathbb{E}[\tau_j | i_j, s_j]$



Improving the Factor of 2

- Choose each τ_j from a different distribution (instead of uniformly between s_j and $s_j + p_{i_j,j}$)
  - Shift mass from left to center: long jobs are less likely to delay j
  - Shift mass from right to center: not increase $\mathbb{E}[\tau_j | i_j, s_j]$

Choosing each τ_j uniformly from $[s_j + \delta p_{i,j}, s_j + (1-\delta)p_{i,j}]$ for small $\delta > 0$ can already improve the ratio 2!

Analyze Effect of Using Non-Uniform Distributions

- Let $\boldsymbol{\Theta}$ be distribution over $[0,1]$ for shifting parameters
- i.e, $\tau_j = s_j + \theta_j p_{i,j}$, where $\theta_j \sim \boldsymbol{\Theta}$

$$C_j = \sum_{j' \neq j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j\right) + p_{i,j}$$

Analyze Effect of Using Non-Uniform Distributions

- Let $\boldsymbol{\Theta}$ be distribution over $[0,1]$ for shifting parameters
- i.e, $\tau_j = s_j + \theta_j p_{i,j}$, where $\theta_j \sim \boldsymbol{\Theta}$

$$C_j = \sum_{j' \neq j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j \right) + p_{i,j}$$

- Bound first two terms by $\sum_{\text{rectangles } R}(\text{contribution of } R)$

Analyze Effect of Using Non-Uniform Distributions

- Let $\mathbf{\Theta}$ be distribution over $[0,1]$ for shifting parameters
- i.e, $\tau_j = s_j + \theta_j p_{i,j}$, where $\theta_j \sim \mathbf{\Theta}$

$$C_j = \sum_{j' \neq j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j\right) + p_{i,j}$$

- Bound first two terms by $\sum_{\text{rectangles } R}(\text{contribution of } R)$
- Final ratio depends on

$$\max_{\text{rectangles } R} \frac{\text{expected contribution of } R}{\text{area of } R \text{ before } \tau_j},$$

which depends on of $\mathbf{\Theta}$

Analyze Effect of Using Non-Uniform Distributions

- Let $\boldsymbol{\Theta}$ be distribution over $[0,1]$ for shifting parameters
- i.e, $\tau_j = s_j + \theta_j p_{i,j}$, where $\theta_j \sim \boldsymbol{\Theta}$

$$C_j = \sum_{j' \neq j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \big(|\text{idle time}| \text{ before } \tau_j\big) + p_{i,j}$$

- Bound first two terms by $\sum_{\text{rectangles } R}(\text{contribution of } R)$
- Final ratio depends on

$$\max_{\text{rectangles } R} \frac{\text{expected contribution of } R}{\text{area of } R \text{ before } \tau_j},$$

  which depends on of $\boldsymbol{\Theta}$
- Use computer program to find best $\boldsymbol{\Theta}$

Analyze Effect of Using Non-Uniform Distributions

- Let $\mathbf{\Theta}$ be distribution over $[0,1]$ for shifting parameters
- i.e, $\tau_j = s_j + \theta_j p_{i,j}$, where $\theta_j \sim \mathbf{\Theta}$

$$C_j = \sum_{j' \neq j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j\right) + p_{i,j}$$

- Bound first two terms by $\sum_{\text{rectangles } R} (\text{contribution of } R)$
- Final ratio depends on

$$\max_{\text{rectangles } R} \frac{\text{expected contribution of } R}{\text{area of } R \text{ before } \tau_j},$$

which depends on of $\mathbf{\Theta}$

- Use computer program to find best $\mathbf{\Theta}$
- Find good fit for the distribution

Analyze Effect of Using Non-Uniform Distributions

- Let $\mathbf{\Theta}$ be distribution over $[0,1]$ for shifting parameters
- i.e, $\tau_j = s_j + \theta_j p_{i,j}$, where $\theta_j \sim \mathbf{\Theta}$

$$C_j = \sum_{j' \neq j : i_{j'} = i, \tau_{j'} < \tau_j} p_{i,j'} + \left(|\text{idle time}| \text{ before } \tau_j\right) + p_{i,j}$$

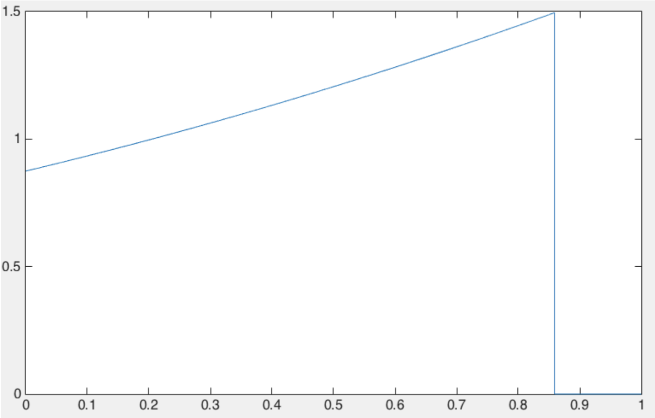
- Bound first two terms by $\sum_{\text{rectangles } R}(\text{contribution of } R)$
- Final ratio depends on

$$\max_{\text{rectangles } R} \frac{\text{expected contribution of } R}{\text{area of } R \text{ before } \tau_j},$$

which depends on of $\mathbf{\Theta}$

- Use computer program to find best $\mathbf{\Theta}$
- Find good fit for the distribution
- Purely analytical proof for approximation ratio

Probability Density Function for Θ



$$f(\theta) = \begin{cases} 0.1702\theta^2 + 0.5768\theta + 0.8746 & \text{if } 0 \leq \theta \leq 0.85897 \\ 0 & \text{if } 85897 < \theta \leq 1 \end{cases}$$

Summary for $R|r_j|\sum_{j\in J} w_jC_j$

Thm There is a 1.8786-approx. for $R|r_j|\sum_{j\in J} w_jC_j$.

Algorithm for $R|r_j|\sum_{j\in J} w_jC_j$

1. solve time-indexed LP relaxation to obtain $\{x_{i,j,s}\}$
2. for each $j \in J$, independently do:
3. let $(i_j, s_j) = (i, s)$ with probability $x_{i,j,s}$
4. choose θ_j from $\mathbf{\Theta}$, and let $\tau_j = s_j + \theta_j p_{i_j,j}$
5. for each $i \in M$, schedule jobs assigned to i in increasing order of τ_j, pretending τ_j's are releasing times

Outline

1. Introduction

2. 1.8786-Approximation for $R|r_j|\sum_j w_j C_j$

3. $(2 + 2\ln 2)$-Approximation for $P|\text{prec}|\sum_j w_j C_j$

$P|\text{prec}|\sum_j w_jC_j$

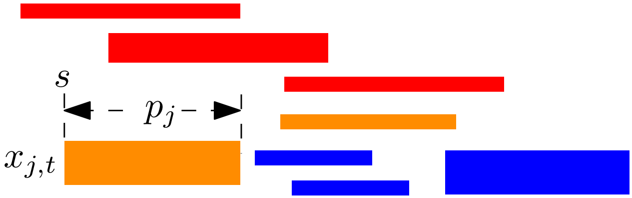
- P: identical machines (instead of unrelated machines)
- prec: we have precedence constraints

Time-Indexed LP Relaxation

- $x_{j,s} \in [0,1]$: j scheduled on interval $(s, s+p_j]$?

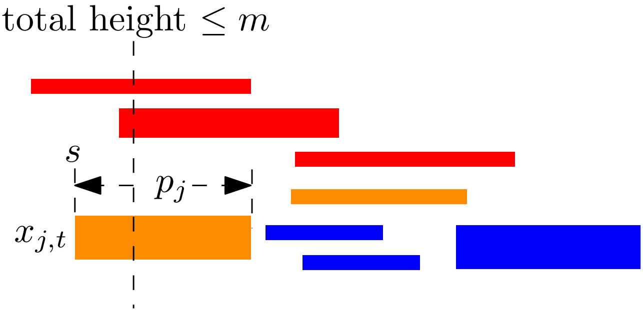
Time-Indexed LP Relaxation

- $x_{j,s} \in [0,1]$: j scheduled on interval $(s, s+p_j]$?



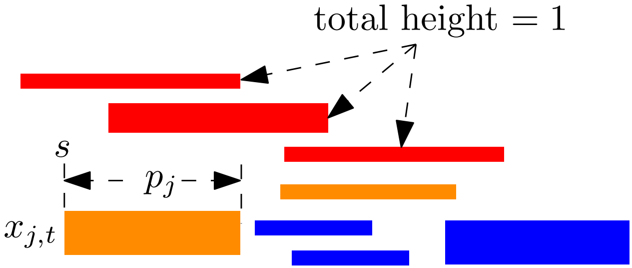
Time-Indexed LP Relaxation

- $x_{j,s} \in [0,1]$: j scheduled on interval $(s, s+p_j]$?



Time-Indexed LP Relaxation

- $x_{j,s} \in [0,1]$: j scheduled on interval $(s, s+p_j]$?



List Scheduling Algorithm Gives 4-Approximation

1. solve LP and let C_j be fractional completion time of j

List Scheduling Algorithm Gives 4-Approximation

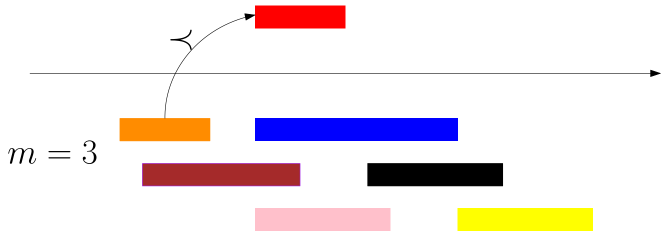
1. solve LP and let C_j be fractional completion time of j
2. for every job j in non-decreasing order of $C_j - p_j/2$

List Scheduling Algorithm Gives 4-Approximation

1. solve LP and let C_j be fractional completion time of j
2. for every job j in non-decreasing order of $C_j - p_j/2$
3. schedule j as early as possible subject to machine-capacity constraints and precedence constraints.

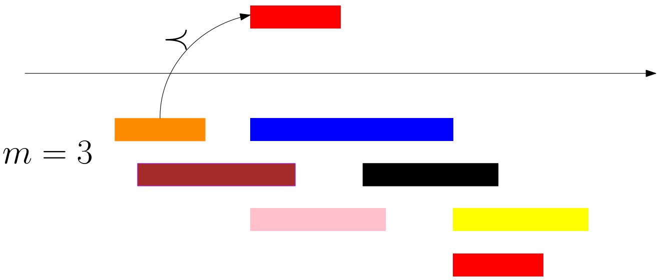
List Scheduling Algorithm Gives 4-Approximation

1. solve LP and let C_j be fractional completion time of j
2. for every job j in non-decreasing order of $C_j - p_j/2$
3. schedule j as early as possible subject to machine-capacity constraints and precedence constraints.



List Scheduling Algorithm Gives 4-Approximation

1. solve LP and let C_j be fractional completion time of j
2. for every job j in non-decreasing order of $C_j - p_j/2$
3. schedule j as early as possible subject to machine-capacity constraints and precedence constraints.



Analysis of List-Scheduling Algorithm

- $\tilde{C}_j$: completion time of j given by algorithm

Analysis of List-Scheduling Algorithm

- $\tilde{C}_j$: completion time of j given by algorithm
- $\tilde{C}_j = I + B$
- $I =$ total idle time before $\tilde{C}_j$
- $B =$ total busy time before $\tilde{C}_j$

Analysis of List-Scheduling Algorithm

- $\tilde{C}_j$: completion time of j given by algorithm
- $\tilde{C}_j = I + B$
- $I =$ total idle time before $\tilde{C}_j$
- $B =$ total busy time before $\tilde{C}_j$

Lemma $I \leq 2C_j$.

Analysis of List-Scheduling Algorithm

- $\tilde{C}_j$: completion time of j given by algorithm
- $\tilde{C}_j = I + B$
- $I =$ total idle time before $\tilde{C}_j$
- $B =$ total busy time before $\tilde{C}_j$

Lemma $I \leq 2C_j$.

Lemma $B \leq 2C_j$.

Lemma $B \leq 2C_j$.

Proof.

Lemma $B \leq 2C_j$.

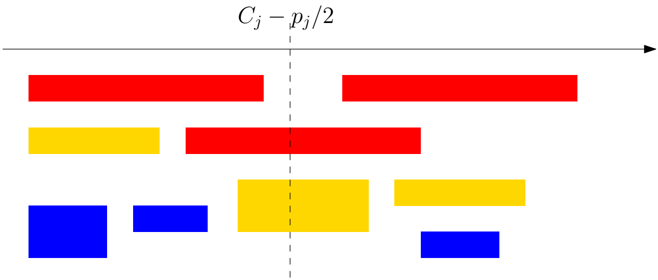
Proof.

- $C_j - p_j/2$: horizontal mass center of rectangles for j

Lemma $B \leq 2C_j$.

Proof.

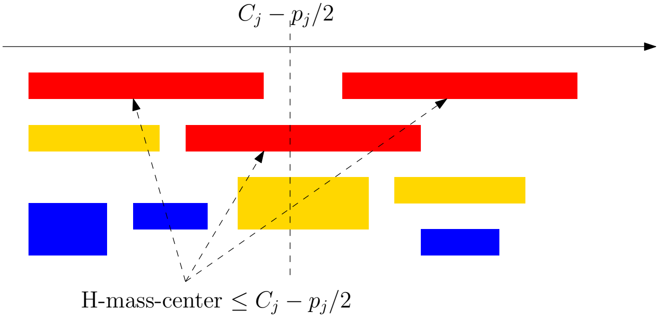
- $C_j - p_j/2$: horizontal mass center of rectangles for j



Lemma $B \leq 2C_j$.

Proof.

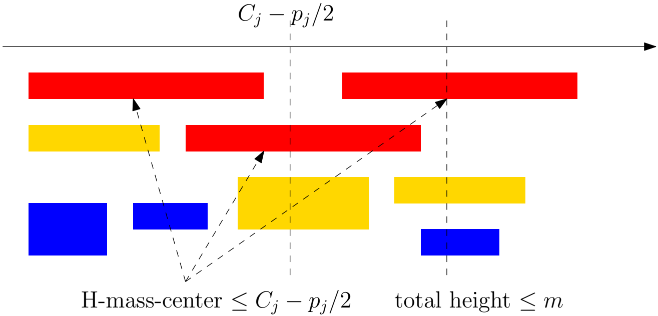
- $C_j - p_j/2$: horizontal mass center of rectangles for j



Lemma $B \leq 2C_j$.

Proof.

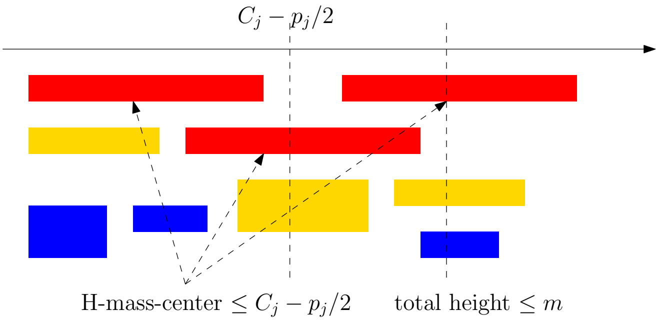
- $C_j - p_j/2$: horizontal mass center of rectangles for j



Lemma $B \leq 2C_j$.

Proof.

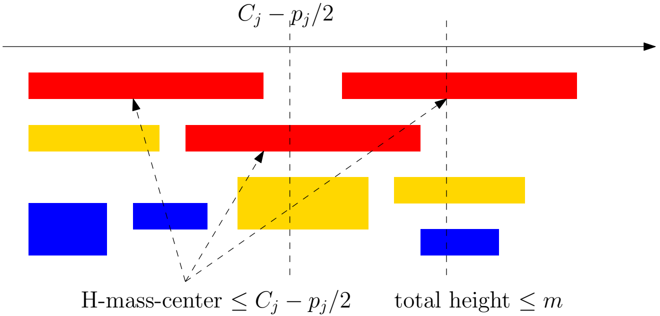
- $C_j - p_j/2$: horizontal mass center of rectangles for j
- $\sum_{j' \text{ considerred before } j} p_{j'} =$ total area $\leq 2m(C_j - p_j/2)$



Lemma $B \le 2C_j$.

Proof.

- $C_j - p_j/2$: horizontal mass center of rectangles for j
- $\sum_{j' \text{ considerred before } j} p_{j'} =$ total area $\le 2m(C_j - p_j/2)$
- length of busy time $\frac{2m(C_j - p_j/2)}{m} \le 2C_j$ □



Our Algorithm

1. solve LP and let C_j be fractional completion time of j
2. choose $\theta \sim_R [0, 1/2]$
3. for every job j in non-decreasing order of $C_j - p_j + \theta p_j$
4. schedule j as early as possible subject to machine-capacity constraints and precedence constraints.

Our Algorithm

1. solve LP and let C_j be fractional completion time of j
2. choose $\theta \sim_R [0, 1/2]$
3. for every job j in non-decreasing order of $C_j - p_j + \theta p_j$
4. schedule j as early as possible subject to machine-capacity constraints and precedence constraints.

Lemma $I \leq \frac{1}{1-\theta} C_j$

Our Algorithm

1. solve LP and let C_j be fractional completion time of j
2. choose $\theta \sim_R [0, 1/2]$
3. for every job j in non-decreasing order of $C_j - p_j + \theta p_j$
4. schedule j as early as possible subject to machine-capacity constraints and precedence constraints.

Lemma $I \leq \frac{1}{1-\theta} C_j$

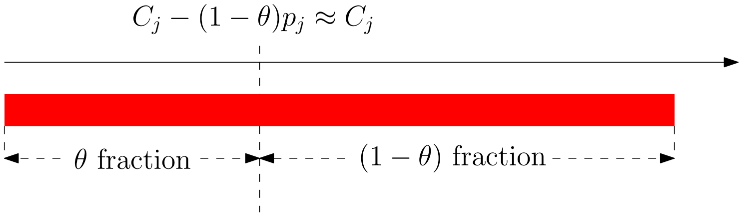
Lemma $B \leq \frac{1}{\theta} C_j$

Our Algorithm

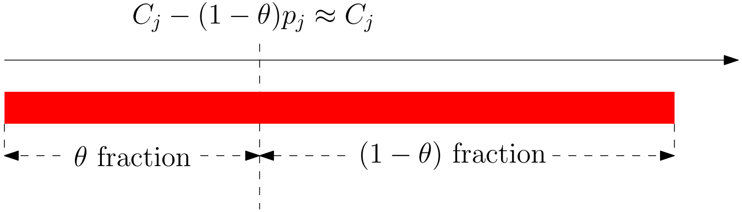
1. solve LP and let C_j be fractional completion time of j
2. choose $\theta \sim_R [0, 1/2]$
3. for every job j in non-decreasing order of $C_j - p_j + \theta p_j$
4. schedule j as early as possible subject to machine-capacity constraints and precedence constraints.

Lemma $I \leq \frac{1}{1-\theta} C_j$

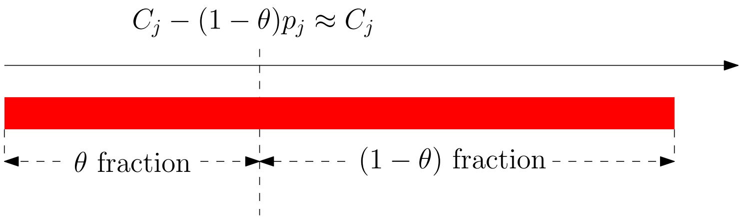
Lemma $B \leq \frac{1}{\theta} C_j$



Lemma $B \leq \frac{1}{\theta} C_j$

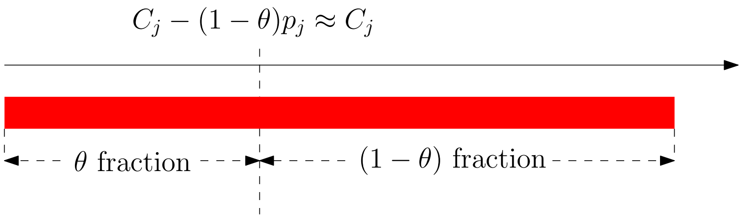


Lemma $B \leq \frac{1}{\theta} C_j$



- Bound can not be tight for every $\theta \in [0, 1/2]$!

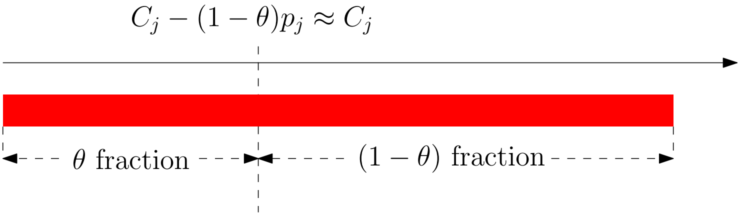
Lemma $B \leq \frac{1}{\theta} C_j$



- Bound can not be tight for every $\theta \in [0, 1/2]$!

Lemma $\mathbb{E}_{\theta \sim_R [0,1/2]} B \leq 2C_j$.

Lemma $B \leq \frac{1}{\theta} C_j$



- Bound can not be tight for every $\theta \in [0, 1/2]$!

Lemma $\mathbb{E}_{\theta \sim_R [0,1/2]} B \leq 2C_j$.

- Approximation ratio $= 2 + \mathbb{E}_{\theta \sim_R [0,1/2]} \frac{1}{1-\theta} = 2 + 2\ln 2$

Summary for $P|\text{prec}|\sum_j w_j C_j$

Thm There is a $(2 + 2\ln 2)$-approximation for $P|\text{prec}|\sum_j w_j C_j$.

1. solve LP and let C_j be fractional completion time of j
2. choose $\theta \sim_R [0, 1/2]$
3. for every job j in non-decreasing order of $C_j - p_j + \theta p_j$
4. schedule j as early as possible subject to machine-capacity constraints and precedence constraints.

Open Problems

- Better approximation for $R|r_j|\sum_j w_j C_j$? (APX-hardness known)

Open Problems

- Better approximation for $R|r_j|\sum_j w_j C_j$? (APX-hardness known)
- $(1.5 - c)$-approximation for $R||\sum_j w_j C_j$ with a reasonable $c > 0$? (APX-hardness known)

Open Problems

- Better approximation for $R|r_j|\sum_j w_j C_j$? (APX-hardness known)
- $(1.5 - c)$-approximation for $R||\sum_j w_j C_j$ with a reasonable $c > 0$? (APX-hardness known)
- 2-approximation for $P|\text{prec}, p_j = 1|\sum_j w_j C_j$? (UGC-2-hardness known)

Open Problems

- Better approximation for $R|r_j|\sum_j w_jC_j$? (APX-hardness known)
- $(1.5-c)$-approximation for $R||\sum_j w_jC_j$ with a reasonable $c > 0$? (APX-hardness known)
- 2-approximation for $P|\text{prec}, p_j = 1|\sum_j w_jC_j$? (UGC-2-hardness known)
- Improving our $O(\log m/\log\log m)$-approximation for $Q|\text{prec}|\sum_j w_jC_j$?