Improvements to a Branch-Cut-and-Price Algorithm for the Exact Solution of Parallel Machines Scheduling Problems

Daniel Oliveira, Artur Pessoa

Universidade Federal Fluminense

Sep 29, 2016
XLVIII Simpósio Brasileiro de Pesquisa Operacional

Outline

The Parallel Machines Scheduling Problem

The Parallel Machines Scheduling Problem

- $J = \{1, \ldots, n\}$

The Parallel Machines Scheduling Problem

- $J = \{1, \ldots, n\}$
- $M = \{1, \ldots, m\}$

The Parallel Machines Scheduling Problem

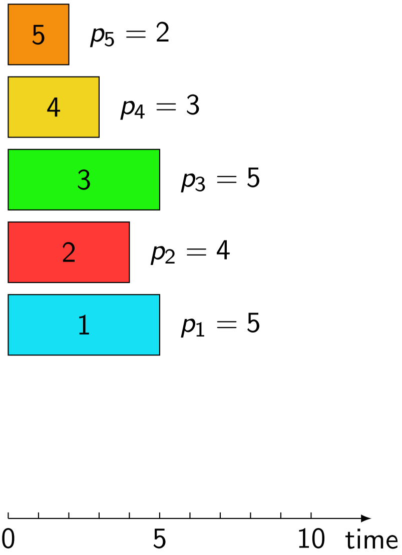
- $J = \{1, \ldots, n\}$
- $M = \{1, \ldots, m\}$
- Processing times p_j

The Parallel Machines Scheduling Problem

- $J = \{1, \ldots, n\}$
- $M = \{1, \ldots, m\}$
- Processing times p_j
- Cost $f_j(C_t)$

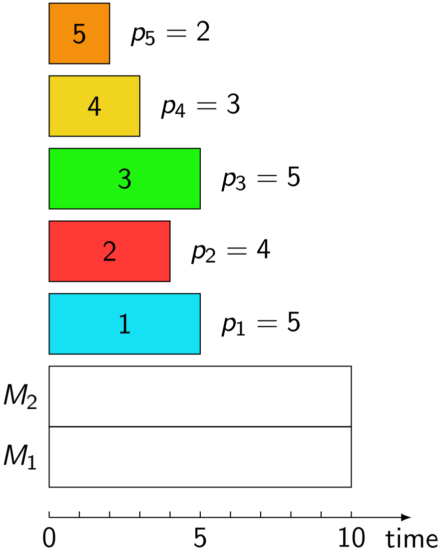
The Parallel Machines Scheduling Problem

- $J = \{1, \ldots, n\}$
- $M = \{1, \ldots, m\}$
- Processing times p_j
- Cost $f_j(C_t)$



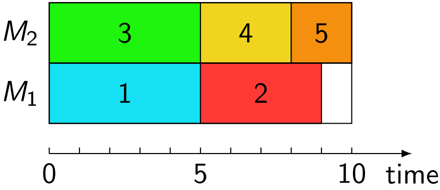
The Parallel Machines Scheduling Problem

- $J = \{1, \ldots, n\}$
- $M = \{1, \ldots, m\}$
- Processing times p_j
- Cost $f_j(C_t)$



The Parallel Machines Scheduling Problem

- $J = \{1, \ldots, n\}$
- $M = \{1, \ldots, m\}$
- Processing times p_j
- Cost $f_j(C_t)$

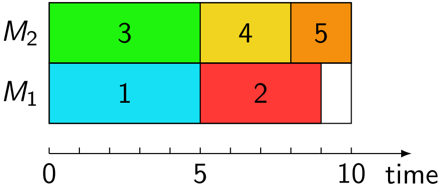


The Parallel Machines Scheduling Problem

- $J = \{1, \dots, n\}$
- $M = \{1, \dots, m\}$
- Processing times p_j
- Cost $f_j(C_t)$

Weighted Tardiness:

- Due dates d_j
- Weights w_j
- Minimize $\sum w_j T_j$



The Arc-Time-indexed Formulation

The Arc-Time-indexed Formulation

- Variables x_{ij}^t: job j succeeds job i at time t

The Arc-Time-indexed Formulation

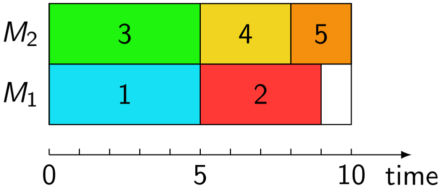
- Variables x_{ij}^t: job j succeeds job i at time t
- Schedules are paths in $G = (V, A)$

The Arc-Time-indexed Formulation

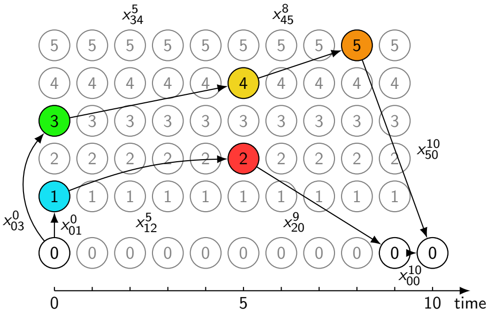
- Variables x_{ij}^{t}: job j succeeds job i at time t
- Schedules are paths in $G = (V, A)$
  - $V = \{(i, t)\}$
  - $A = \{((i, t - p_i), (j, t))\}$
  - $i, j \in J_0 = J \cup \{0\}$
  - $t \in \{0, \ldots, T\}$

The Arc-Time-indexed Formulation

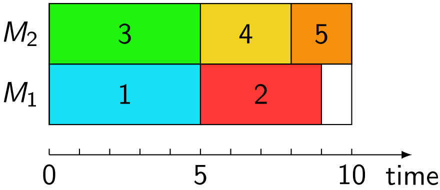
- Variables x_{ij}^t: job j succeeds job i at time t
- Schedules are paths in $G = (V, A)$
  - $V = \{(i, t)\}$
  - $A = \{((i, t - p_i), (j, t))\}$
  - $i, j \in J_0 = J \cup \{0\}$
  - $t \in \{0, \ldots, T\}$



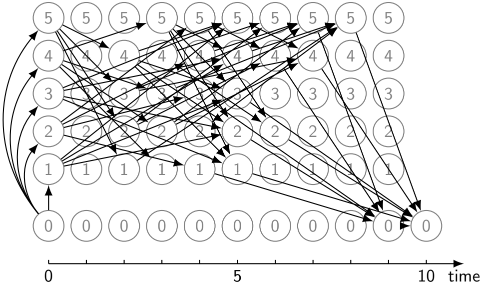
The Arc-Time-indexed Formulation



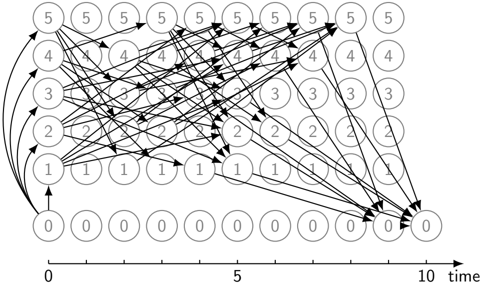
- Variables x^t_{ij}: job j succeeds job i at time t
- Schedules are paths in $G = (V, A)$
  - $V = \{(i, t)\}$
  - $A = \{((i, t - p_i), (j, t))\}$
  - $i, j \in J_0 = J \cup \{0\}$
  - $t \in \{0, \dots, T\}$



The Arc-Time-indexed Formulation

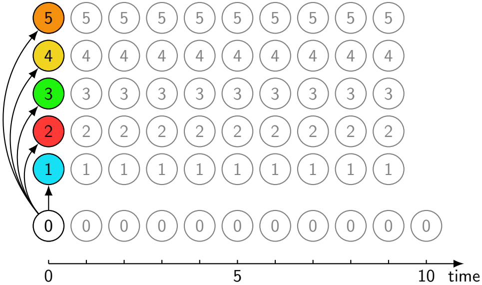


The Arc-Time-indexed Formulation



$$\min \sum_{i \in J_0} \sum_{j \in J \setminus \{i\}} \sum_{t=p_i}^{T-p_j} f_j\left(t+p_j\right) x_{ij}^t$$

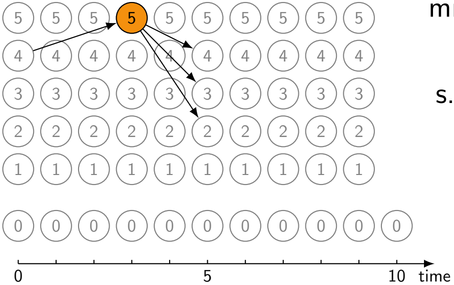
The Arc-Time-indexed Formulation



$$\min \quad \sum_{i \in J_0} \sum_{j \in J \setminus \{i\}} \sum_{t=p_i}^{T-p_j} f_j\left(t+p_j\right) x_{ij}^t$$

$$\text{s.t.} \quad \sum_{j \in J_0} x_{0j}^0 = m$$

The Arc-Time-indexed Formulation

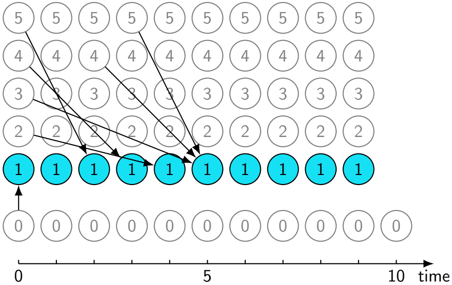


$$\min \quad \sum_{i \in J_0} \sum_{j \in J \setminus \{i\}} \sum_{t=p_i}^{T-p_j} f_j\,(t+p_j)\, x_{ij}^t$$

$$\text{s.t.} \quad \sum_{\substack{j \in J_0 \setminus \{i\},\\ t-p_j \geq 0}} x_{ji}^t - \sum_{\substack{j \in J_0 \setminus \{i\},\\ t+p_i+p_j \leq T}} x_{ij}^{t+p_i} = 0$$

$$(\forall i \in J;\ t = 0, \ldots, T - p_i)$$

The Arc-Time-indexed Formulation



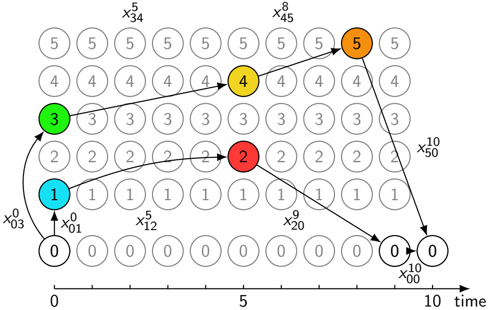
$$\min \quad \sum_{i \in J_0} \sum_{j \in J \setminus \{i\}} \sum_{t=p_i}^{T-p_j} f_j\left(t+p_j\right) x_{ij}^t$$

$$\text{s.t.} \quad \sum_{i \in J_0 \setminus \{j\}} \sum_{t=p_i}^{T-p_j} x_{ij}^t = 1 \quad (j \in J)$$

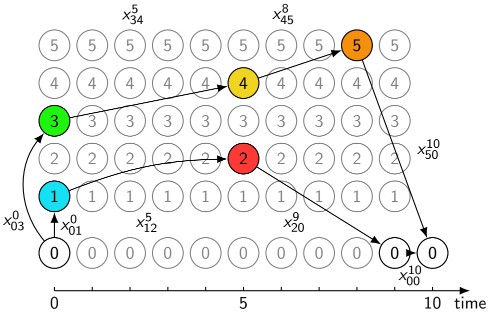
$$x \in Z^+$$

ATIF Reformulation

ATIF Reformulation

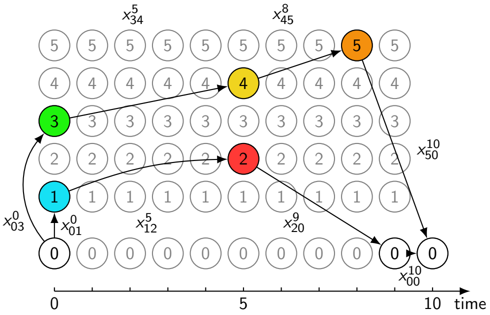


ATIF Reformulation



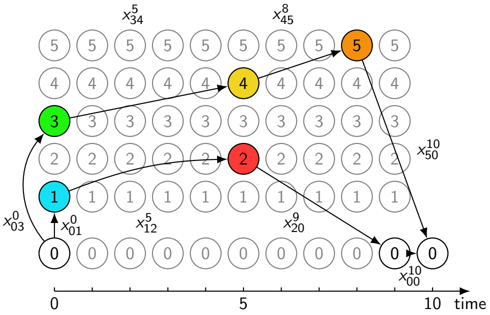
- Pseudo-Schedule: Path from $(0, 0)$ to $(0, T)$ in G

ATIF Reformulation



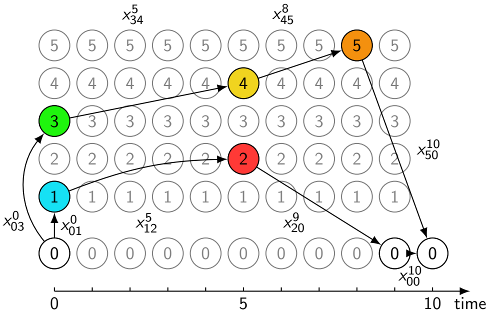
- Pseudo-Schedule: Path from $(0,0)$ to $(0,T)$ in G
- λ_p: pseudo-schedule p is part of the solution

ATIF Reformulation



- Pseudo-Schedule: Path from $(0,0)$ to $(0,T)$ in G
- λ_p: pseudo-schedule p is part of the solution
- $x^t_{ij} = \sum_{p \in P} q^{tp}_{ij} \lambda_p$ (q^{tp}_{ij} constant)

ATIF Reformulation



- Pseudo-Schedule: Path from $(0, 0)$ to $(0, T)$ in G
- λ_p: pseudo-schedule p is part of the solution
- $x^t_{ij} = \sum_{p \in P} q^{tp}_{ij} \lambda_p$ (q^{tp}_{ij} constant)
- Substituting in the ATIF without flow conservation

ATIF Reformulation

$$\min \quad \sum_{p \in P} \left(\sum_{(i,j)^t \in A} q_{ij}^{tp} f_j(t+p_j) \right) \lambda_p$$

$$\text{s.t.} \quad \sum_{p \in P} \left(\sum_{(j,i)^t \in A} q_{ji}^{tp} \right) \lambda_p = 1 \qquad (\forall\, i \in J)$$

$$\sum_{p \in P} \left(\sum_{(0,j)^0 \in A} q_{0j}^{0p} \right) \lambda_p = m$$

$$\lambda \geq 0$$

ATIF Reformulation

$$\min \quad \sum_{p \in P} \left(\sum_{(i,j)^t \in A} q_{ij}^{tp} f_j(t + p_j) \right) \lambda_p$$

$$\text{s.t.} \quad \sum_{p \in P} \left(\sum_{(j,i)^t \in A} q_{ji}^{tp} \right) \lambda_p = 1 \qquad (\forall\, i \in J)$$

$$\sum_{p \in P} \left(\sum_{(0,j)^0 \in A} q_{0j}^{0p} \right) \lambda_p = m$$

$$\lambda \geq 0$$

$$\sum_{a^t \in A} \alpha_{al}^t x_a^t \qquad \geq b_l \qquad (\forall\, l \in \{n+1, \ldots, r\})$$

ATIF Reformulation

$$\min \quad \sum_{p \in P} \left(\sum_{(i,j)^t \in A} q_{ij}^{tp} f_j(t + p_j) \right) \lambda_p$$

$$\text{s.t.} \quad \sum_{p \in P} \left(\sum_{(j,i)^t \in A} q_{ji}^{tp} \right) \lambda_p = 1 \qquad (\forall\, i \in J)$$

$$\sum_{p \in P} \left(\sum_{(0,j)^0 \in A} q_{0j}^{0p} \right) \lambda_p = m$$

$$\lambda \geq 0$$

$$\sum_{a^t \in A} \alpha_{al}^{t} \left(\sum_{p \in P} q_a^{tp} \lambda_p \right) \geq b_l \qquad (\forall\, l \in \{n+1, \ldots, r\})$$

ATIF Reformulation

$$\min \quad \sum_{p \in P} \left(\sum_{(i,j)^t \in A} q_{ij}^{tp} f_j(t + p_j) \right) \lambda_p$$

$$\text{s.t.} \quad \sum_{p \in P} \left(\sum_{(j,i)^t \in A} q_{ji}^{tp} \right) \lambda_p = 1 \qquad (\forall\, i \in J)$$

$$\sum_{p \in P} \left(\sum_{(0,j)^0 \in A} q_{0j}^{0p} \right) \lambda_p = m$$

$$\lambda \geq 0$$

$$\sum_{a^t \in A} \alpha_{al}^t \left(\sum_{p \in P} q_a^{tp} \lambda_p \right) \geq b_l \qquad (\forall\, l \in \{n+1, \ldots, r\})$$

- The cuts are robust: Pricing subproblem (shortest path) not changed

Branch-Cut-and-Price

Branch-Cut-and-Price

- Pricing
  - Shortest path from $(0, 0)$ to $(0, T)$ in G with arc lengths $\bar{c}_a^t$

Branch-Cut-and-Price

- Pricing
  - Shortest path from $(0, 0)$ to $(0, T)$ in G with arc lengths $\bar{c}_a^t$
- Fixing x_a^t variables by Reduced Costs after every 5 iterations

Branch-Cut-and-Price

- Pricing
  - Shortest path from $(0,0)$ to $(0,T)$ in G with arc lengths $\bar{c}_a^t$
- Fixing x_a^t variables by Reduced Costs after every 5 iterations
  - Prove that setting some $x_a^t = 1$ exceeds the best known solution cost

Branch-Cut-and-Price

- Pricing
  - Shortest path from $(0,0)$ to $(0,T)$ in G with arc lengths $\bar{c}_a^t$
- Fixing x_a^t variables by Reduced Costs after every 5 iterations
  - Prove that setting some $x_a^t = 1$ exceeds the best known solution cost
- Extended Capacity Cuts (Uchoa et al., 2008)

Branch-Cut-and-Price

- Pricing
  - Shortest path from $(0,0)$ to $(0,T)$ in G with arc lengths $\bar{c}_a^t$
- Fixing x_a^t variables by Reduced Costs after every 5 iterations
  - Prove that setting some $x_a^t = 1$ exceeds the best known solution cost
- Extended Capacity Cuts (Uchoa et al., 2008)
- Dual stabilization of Wentges (1997)

Branch-Cut-and-Price

- Pricing
  - Shortest path from $(0, 0)$ to $(0, T)$ in G with arc lengths $\bar{c}_a^t$
- Fixing x_a^t variables by Reduced Costs after every 5 iterations
  - Prove that setting some $x_a^t = 1$ exceeds the best known solution cost
- Extended Capacity Cuts (Uchoa et al., 2008)
- Dual stabilization of Wentges (1997)
- Strong branching: 8 possible choices

Branch-Cut-and-Price

- Pricing
  - Shortest path from $(0, 0)$ to $(0, T)$ in G with arc lengths $\bar{c}_a^t$
- Fixing x_a^t variables by Reduced Costs after every 5 iterations
  - Prove that setting some $x_a^t = 1$ exceeds the best known solution cost
- Extended Capacity Cuts (Uchoa et al., 2008)
- Dual stabilization of Wentges (1997)
- Strong branching: 8 possible choices
- After Root, if $|A| \leq 200.000$: Feed reduced ATIF to MIP Solver (CPLEX 11.1)

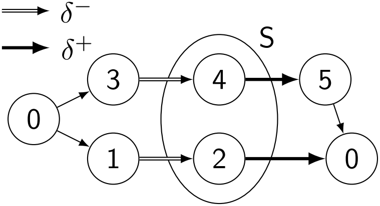
1. The Parallel Machines Scheduling Problem
2. The BCP of Pessoa, Uchoa, Poggi and Rodrigues (2010)
3. The Improved Algorithm
   - Newly Proposed Cuts over Extended Variables
   - Alternative Time-Indexed Formulations
   - New Cuts over Time-Indexed Formulations
4. Experiments
5. Conclusions

The Overload Elimination Cuts

The Overload Elimination Cuts

$$u^t = \sum_{a^t \in \delta^-(S)} x_a^t \quad (t = 1, \ldots, T)$$

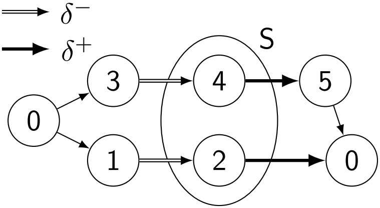
$$v^t = \sum_{a^t \in \delta^+(S)} x_a^t \quad (t = 1, \ldots, T)$$



The Overload Elimination Cuts

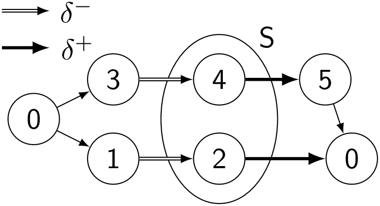
$$u^t = \sum_{a^t \in \delta^-(S)} x_a^t \quad (t = 1, \ldots, T)$$

$$v^t = \sum_{a^t \in \delta^+(S)} x_a^t \quad (t = 1, \ldots, T)$$



- For $m \geq 2$, $S \subseteq J$, and $t \in \{1, \ldots, t_{max}\}$:

The Overload Elimination Cuts



$$u^t = \sum_{a^t \in \delta^-(S)} x_a^t \quad (t = 1, \ldots, T)$$

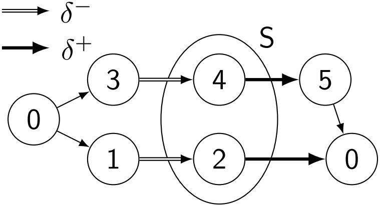
$$v^t = \sum_{a^t \in \delta^+(S)} x_a^t \quad (t = 1, \ldots, T)$$

- For $m \geq 2$, $S \subseteq J$, and $t \in \{1, \ldots, t_{max}\}$:

$$\sum_{q=t}^{t_1} v^q + \sum_{q=t_1+1}^{T} 2\,v^q - \sum_{\substack{q=\max\{t_1,\\ T-p(S)+m(t-1)+1\}}}^{T-1} u^q \geq 2,$$

$$t_1 = p(S) - t - (m-2)(t-1).$$

The Overload Elimination Cuts



$$u^t = \sum_{a^t \in \delta^-(S)} x_a^t \quad (t = 1, \ldots, T)$$

$$v^t = \sum_{a^t \in \delta^+(S)} x_a^t \quad (t = 1, \ldots, T)$$

- For $m \geq 2$, $S \subseteq J$, and $t \in \{1, \ldots, t_{max}\}$:

$$\sum_{q=t}^{t_1} v^q + \sum_{q=t_1+1}^{T} 2\, v^q - \sum_{\substack{q=\max\{t_1, \\ T-p(S)+m(t-1)+1\}}}^{T-1} u^q \geq 2,$$

$$t_1 = p(S) - t - (m-2)(t-1).$$

- Separation: A specialized genetic algorithm

Switching to a MIP Solver

Switching to a MIP Solver

- Solve Root Node $\rightarrow$ Branching

Switching to a MIP Solver

- Solve Root Node → ~~Branching~~ Feed Residual Model to CPLEX

Switching to a MIP Solver

- Solve Root Node → ~~Branching~~ Feed Residual Model to CPLEX
- Residual Model: Excludes variables fixed to zero

Switching to a MIP Solver

- Solve Root Node $\rightarrow$ ~~Branching~~ Feed Residual Model to CPLEX
- Residual Model: Excludes variables fixed to zero
- Pessoa et al. (2010): **ATIF** residual model

Switching to a MIP Solver

- Solve Root Node → ~~Branching~~ Feed Residual Model to CPLEX
- Residual Model: Excludes variables fixed to zero
- Pessoa et al. (2010): **ATIF** residual model
- Now: **TIF** residual model

Switching to a MIP Solver

- Solve Root Node $\rightarrow$ ~~Branching~~ Feed Residual Model to CPLEX
- Residual Model: Excludes variables fixed to zero
- Pessoa et al. (2010): **ATIF** residual model
- Now: **TIF** residual model
- Time-Indexed Formulation, Dyer and Wolsey (1991)

Switching to a MIP Solver

- Solve Root Node $\rightarrow$ ~~Branching~~ Feed Residual Model to CPLEX
- Residual Model: Excludes variables fixed to zero
- Pessoa et al. (2010): **ATIF** residual model
- Now: **TIF** residual model
- Time-Indexed Formulation, Dyer and Wolsey (1991)
  - Variables y_j^t: job j completes at time t

Switching to a MIP Solver

- Solve Root Node $\rightarrow$ ~~Branching~~ Feed Residual Model to CPLEX
- Residual Model: Excludes variables fixed to zero
- Pessoa et al. (2010): **ATIF** residual model
- Now: **TIF** residual model
- Time-Indexed Formulation, Dyer and Wolsey (1991)
  - Variables y_j^t: job j completes at time t

$$
\begin{aligned}
\min \quad & \sum_{j \in J} \sum_{t=p_j}^{T} f_j(t)\, y_j^t \\
\text{s.t.} \quad & \sum_{t=p_j}^{T} y_j^t = 1 && (j \in J) \\
& \sum_{j \in J} \sum_{t'=t}^{\min\{t+p_j-1,\, T\}} y_j^{t'} \leq m && (t = 1, \dots, T) \\
& y \in \{0, 1\}
\end{aligned}
$$

Alternative Time-Indexed Formulations

Alternative Time-Indexed Formulations

- How to enforce that no more that m machines are running?

Alternative Time-Indexed Formulations

- How to enforce that no more that m machines are running?
  - *R* Resource Constraints

Alternative Time-Indexed Formulations

- How to enforce that no more that m machines are running?
  - *R* Resource Constraints
  - *F* Network Flow

Alternative Time-Indexed Formulations

- How to enforce that no more that m machines are running?
  - *R* Resource Constraints
  - *F* Network Flow
- Variable definition

Alternative Time-Indexed Formulations

- How to enforce that no more that m machines are running?
  - *R* Resource Constraints
  - *F* Network Flow
- Variable definition
  - *y* Variables y_j^t: job j completes at time t

Alternative Time-Indexed Formulations

- How to enforce that no more that m machines are running?
  - R Resource Constraints
  - F Network Flow
- Variable definition
  - y Variables y_j^t: job j completes at time t
  - z Variables z_j^t: job j completes until time t ($y_j^t = z_j^t - z_j^{t-1}$)

Alternative Time-Indexed Formulations

- How to enforce that no more that m machines are running?
  - R Resource Constraints
  - F Network Flow
- Variable definition
  - y Variables y_j^t: job j completes at time t
  - z Variables z_j^t: job j completes until time t ($y_j^t = z_j^t - z_j^{t-1}$)
- 4 different time-indexed formulations (R_y, R_z, F_y, F_z)

Alternative Time-Indexed Formulations

$\underline{R_y}$ R_z F_y F_z

$$\min \quad \sum_{j \in J} \sum_{t=p_j}^{T} f_j(t)\, y_j^t$$

$$\text{s.t.} \quad \sum_{t=p_j}^{T} y_j^t = 1 \qquad (j \in J)$$

$$\sum_{j \in J} \sum_{t'=t}^{\min\{t+p_j-1,\, T\}} y_j^{t'} \le m \quad (t = 1, \ldots, T)$$

$$y \in \{0, 1\}$$

Alternative Time-Indexed Formulations

R_y $\quad$ $\underline{R_z}$ $\quad$ F_y $\quad$ F_z

$$
\begin{aligned}
\min \quad & \sum_{j \in J} \sum_{t=p_j}^{T} f_j(t) \left(z_j^t - z_j^{t-1} \right) & \\
\text{s.t.} \quad & z_j^{p_j - 1} = 0 & (j \in J) \\
& z_j^{t-1} \leq z_j^t & (j \in J;\ t = p_j, \ldots, T) \\
& z_j^T = 1 & (j \in J) \\
& \sum_{j \in J} \left(z_j^{\min\{t+p_j-1,\, T\}} - z_j^{t-1} \right) \leq m & (t = 1, \ldots, T) \\
& z \in \{0, 1\} &
\end{aligned}
$$

Alternative Time-Indexed Formulations

R_y R_z F_y F_z

$$\min \quad \sum_{j \in J} \sum_{t=p_j}^{T} f_j(t)\, y_j^t$$

$$\text{s.t.} \quad \sum_{t=p_j}^{T} y_j^t = 1 \qquad (j \in J)$$

$$\sum_{j \in J} \sum_{t'=t}^{\min\{t+p_j-1,\, T\}} y_j^{t'} \leq m \quad (t = 1, \ldots, T)$$

$$y \in \{0, 1\}$$

Alternative Time-indexed Formulations

R_y R_z F_y F_z

$$
\begin{aligned}
\min \quad & \sum_{j \in J} \sum_{t=p_j}^{T} f_j(t)\, y_j^t \\
\text{s.t.} \quad & \sum_{t=p_j}^{T} y_j^t = 1 && (j \in J) \\
& \sum_{j \in J} y_j^{p_j} = m \\
& \sum_{j \in J | t \geq p_j} y_j^t \geq \sum_{j \in J} y_j^{t+p_j} && (t = 1, \ldots, T) \\
& y \in \{0, 1\}
\end{aligned}
$$

Alternative Time-Indexed Formulations

R_y R_z F_y $\underline{F_z}$

$$\min \quad \sum_{j \in J} \sum_{t=p_j}^{T} f_j(t) \left(z_j^t - z_j^{t-1} \right)$$

$$\text{s.t.} \quad z_j^{p_j - 1} = 0 \qquad (j \in J)$$

$$z_j^{t-1} \leq z_j^t \qquad (j \in J;\ t = p_j, \ldots, T)$$

$$z_j^T = 1 \qquad (j \in J)$$

$$\sum_{j \in J} \left(z_j^{p_j} - z_j^{p_j - 1} \right) = m$$

$$\sum_{j \in J | t \geq p_j} \left(z_j^t - z_j^{t-1} \right) \geq \sum_{j \in J} \left(z_j^{t+p_j} - z_j^{t+p_j-1} \right) \qquad (t = 1, \ldots, T)$$

$$z \in \{0, 1\}$$

TIF Cuts by Projecting the ATIF Polytope

TIF Cuts by Projecting the ATIF Polytope

$$\sum_{i \in J \setminus \{j\}} y_i^t \geq y_j^{t+p_j} \qquad (j \in J;\ t = 1, \ldots, T)$$

TIF Cuts by Projecting the ATIF Polytope

$$\sum_{i \in J \setminus \{j\}} y_i^t \geq y_j^{t+p_j} \qquad (j \in J;\ t = 1, \ldots, T)$$

$$prec_t(j) = \{i \mid \ x_{ij}^t \text{ is not fixed}\}$$

TIF Cuts by Projecting the ATIF Polytope

$$\sum_{i \in J \setminus \{j\}} y_i^t \geq y_j^{t+p_j} \qquad (j \in J;\ t = 1, \ldots, T)$$

$$prec_t(j) = \{i \mid x_{ij}^t \text{ is not fixed}\}$$

$$prec_t(S) = \bigcup_{j \in S} prec_t(j)$$

TIF Cuts by Projecting the ATIF Polytope

$$\sum_{i \in J \setminus \{j\}} y_i^t \geq y_j^{t+p_j} \qquad (j \in J;\ t = 1, \ldots, T)$$

$$prec_t(j) = \{ i \mid \ x_{ij}^t \text{ is not fixed} \}$$

$$prec_t(S) = \bigcup_{j \in S} prec_t(j)$$

$$\sum_{i \in prec_t(S)} y_i^t \geq \sum_{j \in S} y_j^{t+p_j} \qquad (S \subset J;\ t = 1, \ldots, T)$$

TIF Cuts by Projecting the ATIF Polytope

$$\sum_{i \in J \setminus \{j\}} y_i^t \geq y_j^{t+p_j} \qquad (j \in J;\ t = 1, \ldots, T)$$

$$prec_t(j) = \{i \mid\ x_{ij}^t \text{ is not fixed}\}$$

$$prec_t(S) = \bigcup_{j \in S} prec_t(j)$$

$$\sum_{i \in prec_t(S)} y_i^t \geq \sum_{j \in S} y_j^{t+p_j} \qquad (S \subset J;\ t = 1, \ldots, T)$$

- Separation: Solving a Minimum Cut Problem in a directed graph

How much of the Integrality Gap is closed?

How much of the Integrality Gap is closed?

Table: Root relaxation and cut separation results

		BCP-PMWT		BCP-PMWT-OTI	
n	m	Avg. Gap	Avg. Time	Avg. Gap	Avg. Time
40	2	0.525%	78.0	0.235%	51.9
40	4	0.456%	23.4	0.448%	18.8
50	2	0.379%	256.8	0.276%	193.8
50	4	0.571%	67.8	0.583%	29.9
100	2	0.878%	6297.0	0.114%	3398.8
100	4	0.494%	984.0	0.322%	481.6

Which is the best TIF?

Which is the best TIF?

Table: Comparison of Alternative Time-Indexed Formulations

	Average LP Time (s)				Average MIP Time (s)*				# Solved**			
n	Fy	Ry	Fz	Rz	Fy	Ry	Fz	Rz	Fy	Ry	Fz	Rz
40	0.72	0.84	7.17	0.97	63.17	351.97	122.92	58.28	12	10	12	12
50	1.77	1.98	47.08	2.43	53.46	150.26	70.56	16.47	13	11	14	16

*average times uses only instances solved with all 4 TIFs in up to 3,600 seconds
**we tested 12 instances of 40 jobs and 17 instances of 50 jobs

How much help is the Variable Fixation?

How much help is the Variable Fixation?

Table: Effect of Variable Fixation in the Rz Time-Index Formulation – Summary

	Average LP Time (s)		Average MIP Time (s)*		# Solved		Total
n	Fix.	w/ Fix.	Fix.	w/ Fix.	Fix.	w/ Fix.	
40	0.74	22.54	11.11	561.59	12	10	12
50	2.04	105.84	11.63	496.61	17	9	17

*average times uses only instances solved by both in up to 3,600 seconds

How much help are the Projected Cuts?

How much help are the Projected Cuts?

Table: Effect of Projected Cuts in the Rz Time-Indexed Formulation – Summary

		ATIF	TIF		
n	m	Root Gap	1st LP Gap	Root Gap	Gap Improv.
100	2	0.114%	0.294%	0.249%	16.76%
100	4	0.322%	0.660%	0.646%	11.20%

Overall Results

Overall Results

Table: Full Results - Summary

n	m	# Inst.	BCP-PMWT # Solved	BCP-PMWT Avg.* Time	BCP-PMWT-OTI # Solved	BCP-PMWT-OTI Avg.* Time
40		50	50	357.9	50	48.1
50		50	50	5734.9	50	241.9
100	2	25	18	22523.8	21	7058.5
100	4	25	16	37667.7	22	5672.0

*average times uses only instances solved by both

Branching vs Switching to MIP Solver

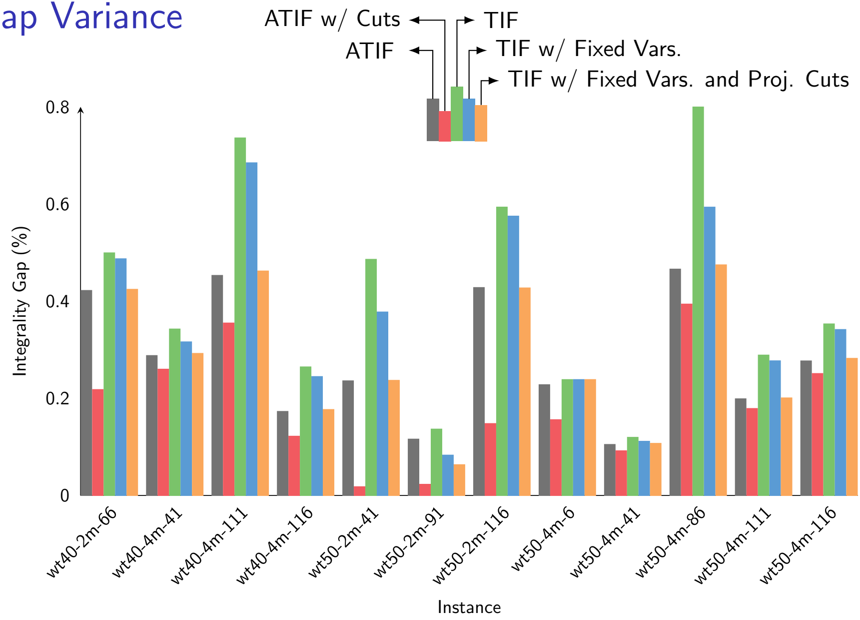
Branching vs Switching to MIP Solver

Table: BCP-PMWT-OTI Best Procedure

		BCP-PMWT				BCP-PMWT-OTI			
n	m	Root	BCP	ATIF MIP	Unsolved	Root	BCP	TIF MIP	Unsolved
40		38	2	10	0	38	1	11	0
50		33	4	13	0	33	3	14	0
100	2	13	2	3	7	16	1	4	4
100	4	7	5	4	9	7	1	14	3

Gap Variance

Gap Variance



Conclusions

Conclusions

- Results

Conclusions

- Results
  - 9 instances solved for the first time

Conclusions

- Results
  - 9 instances solved for the first time
  - 84.1% running time decrease for other instances