A Branch-and-Cut-and-Price algorithm for a large class of parallel machine scheduling problems

Teobaldo Bulhões[2] **Ruslan Sadykov**[1] Eduardo Uchoa[2]
Anand Subramanian[3]

[1] Inria Bordeaux and Univ. Bordeaux, France

[2] Univ. Federal Fluminense, Brazil

[3] Univ. Federal da Paraíba, Brazil

Journées GOThA/Bermudes
Tours, September 26, 2017

Contents

The scheduling problem we want to solve

- Set M of **unrelated** machines
- n jobs, each job $j \in J = \{1, \ldots, n\}$ has
  - processing time p_j^k, dependent on the machine
  - release and due dates r_j and d_j
  - earliness and tardiness unitary penalties α_j and β_j
- Given completion time C_j of job $j \in J$ in the schedule, its cost is

$$\alpha_j E_j + \beta_j T_j = \alpha_j \cdot \max\{0, d_j - C_j\} + \beta_j \cdot \max\{0, C_j - d_j\}$$

- There is a **sequence-dependent setup time** $s_{i,j}^k$ if job j is scheduled immediately after job i on machine k.
- The objective is to minimize the **total earliness/tardiness cost**.
- Problem's notation:

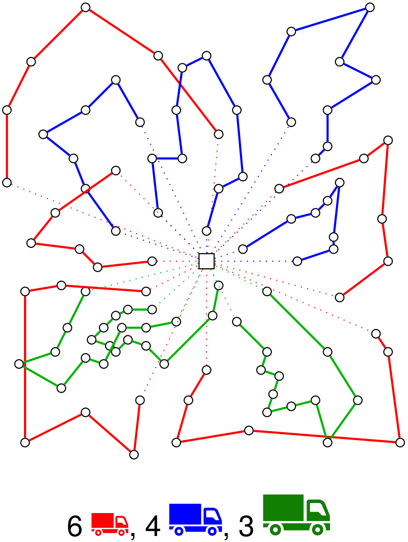
$$R|r_j, s_{ij}^k| \sum_j \alpha_j E_j + \beta_j T_j$$

Heterogeneous Vehicle Routing with Time Windows

Set I of customers, each $i \in I$ with demand d_i, service time s_i and time window $[r_i, d_i]$.

Set M of vehicle types, each $k \in M$ with a depot $i_{|I|+k}$ with U_k vehicles of capacity Q_k, with fixed cost f_u, travel costs c_{ij}^k and travel distances d_{ij}^k for each pair $(i, j) \in I \cup M$ of customers/depots.

Objective: minimize the total fixed and travel cost.



Similarities between problems

HVRP	**UMSP**
Heterogeneous vehicles	Unrelated machines
Vehicle routes	Machine schedules
Capacity resource	One job at a time
Time resource	Time resource
Service times	Job processing times
Distances between customers	Setup times
Minimizing vehicle cost and total travelled distance	Minimizing “just-in-time” penalty (sort of “soft” time windows)

State-of-the-art exact algorithms for Vehicle Routing

Instances with 100–150 customers are routinely solved to optimality

Pecin, D., Pessoa, A., Poggi, M., and Uchoa, E. (2017).
Improved branch-cut-and-price for capacitated vehicle routing.
Mathematical Programming Computation, 9(1):61–100.

Pecin, D., Contardo, C., Desaulniers, G., and Uchoa, E. (2017).
New enhancements for the exact solution of the vehicle routing problem with time windows.
INFORMS Journal on Computing, 29(3):489–502.

Pessoa, A., Sadykov, R., and Uchoa, E. (2017).
Enhanced branch-cut-and-price algorithm for heterogeneous fleet vehicle routing problems.
Cadernos do LOGIS, number 3.

Existing exact approaches in the literature for scheduling on parallel machines with sum criteria

- $R \mid s_{ij}^k \mid \sum \alpha_j E_j + \beta_j T_j$ Only MIP formulations, up to 5 machines and 12 jobs.
- $R \mid\mid \sum T_j$ A branch-and-bound [Shim and Kim, 2007], up to 5 machines and 20 jobs.
- $R \mid\mid \sum w_j T_j$ A branch-and-bound [Liaw et al., 2003], up to 4 machines and 18 jobs.
- $Q \mid s_{ij}^k \mid \sum E_j + T_j$ A MIP and a Benders decomposition [Balakrishnan et al., 1999], up to 20 jobs.
- $P \mid s_f \mid \sum T_j$ A branch-and-bound [Schaller, 2014], up to 3 machines and 14 jobs.
- $P \mid r_j \mid \sum w_j T_j$ A branch-and-bound [Jouglet and Savourey, 2011], up to 5 machines and 20 jobs
- $P \mid\mid \sum w_j T_j$ A Branch-Cut-and-Price [Pessoa et al., 2010], up to 4 machines and 100 jobs.
- $R \mid a_k, r_j, s_{ij}^k \mid \sum w_j T_j$ A branch-and-price [Lopes and de Carvalho, 2007], up to 50 machines and 150 jobs

Contents

Set covering (master) formulation

- Ω_k — set of pseudo-schedules for machine $k \in M$
- a_j^ω — number of times that job j appears in pseudo-schedule ω.
- c_ω — cost of pseudo-schedule ω.
- Binary variable $\lambda_k^\omega = 1$ if and only if pseudo-schedule ω is assigned to machine $k \in M$

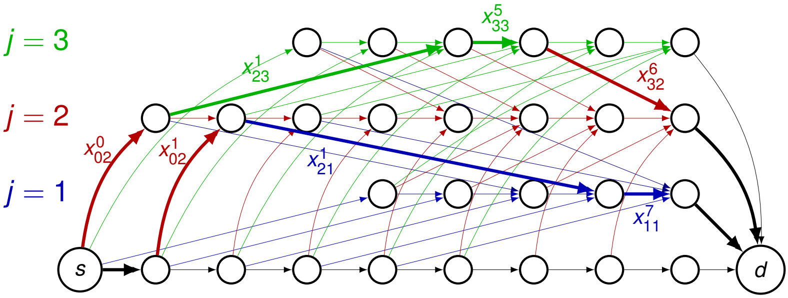
$$
\begin{aligned}
\min \sum_{k \in M} \sum_{\omega \in \Omega_u} c_\omega \lambda_s & & \\
\sum_{k \in M} \sum_{\omega \in \Omega_u} a_j^\omega \lambda_\omega & = 1, & \forall j \in J, \\
\sum_{\omega \in \Omega_k} \lambda_\omega & \leq 1, & \forall k \in M, \\
\lambda_\omega & \in \{0, 1\}, & \forall \omega \in \Omega_k, \forall k \in M.
\end{aligned}
$$

Pricing subproblem for machine $k \in M$

Extended graph G_k

Arc (i, j, t) — setup time between job i and j is started at time t, and job j is started at time $t + s_{ij}^k$

Variable x_{ij}^t — arc (i, j, t) in the solution or not



$J = \{1, 2, 3\}$, $T = 8$, $p_1 = 4$, $p_2 = 1$, $p_3 = 3$, $s_{ij} = 1, \forall i, j \in J$

Pseudo-schedules 0-2-3-2-0 and 0-2-1-0 are shown

Pricing subproblem: the labelling algorithm

Given dual solution π of the restricted master problem, the pricing subproblem is

$$\min_{\omega \in \Omega_k} \bar{c}_\omega = c_\omega - \sum_{j \in J} a_j^\omega \pi_j = \sum_{\substack{i,j \in J,\ i \neq j \\ t \in T}} \left(c_j^{t+s_{ij}+p_j} - \pi_j \right) \cdot x_{ij}^t$$

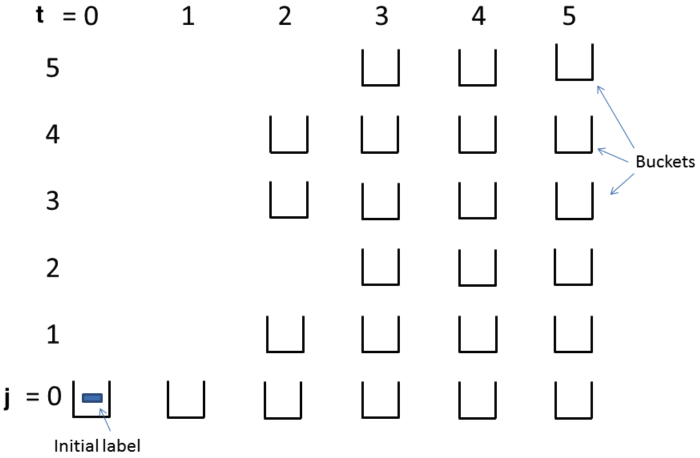
i.e. the shortest path problem in the extended graph.

Labels

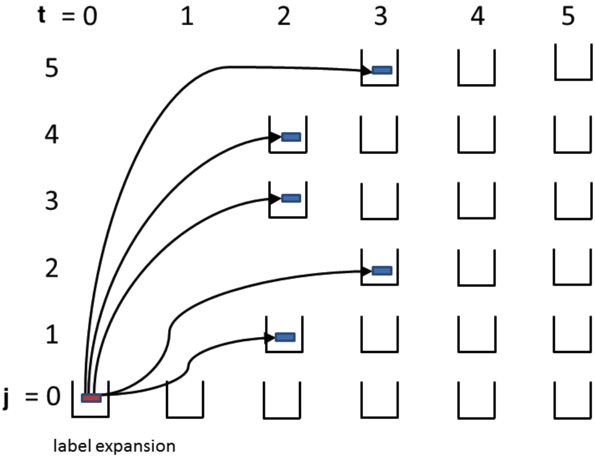
Each label $L = (\bar{c}^L, j^L, t^L)$ represents a partial pseudo-schedule.
It dominates another label L' if

$$j^L = j^{L'},\ t^L = t^{L'},\ \bar{c}^L \leq \bar{c}^{L'}$$

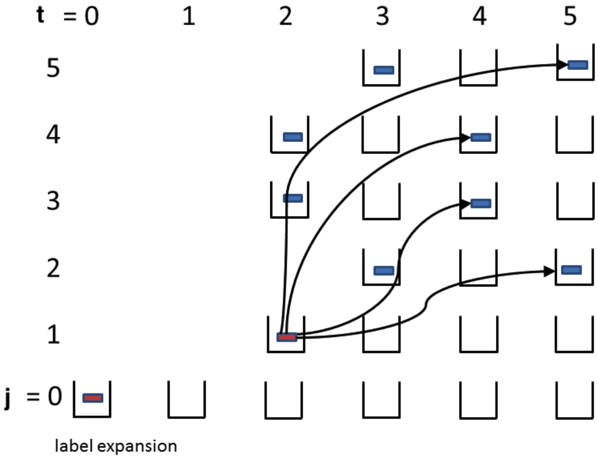
The labeling algorithm



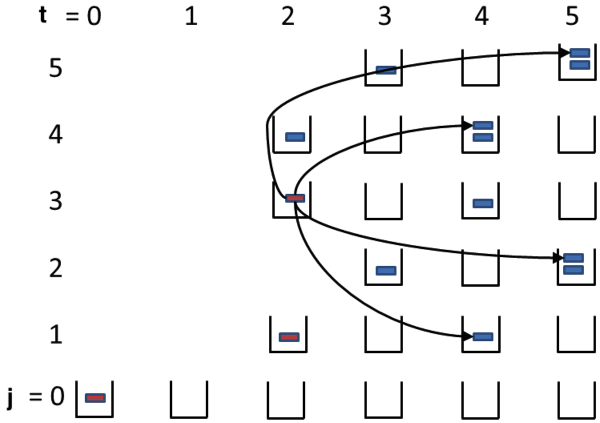
The labeling algorithm



The labeling algorithm

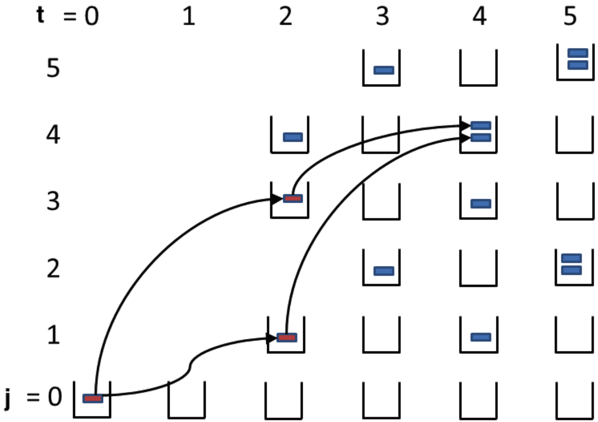


The labeling algorithm



Do both labels need to be kept in bucket (4,4)?

The labeling algorithm



The labels represent partial paths 0-1-4 and 0-3-4

Fixing of arc variables by reduced costs

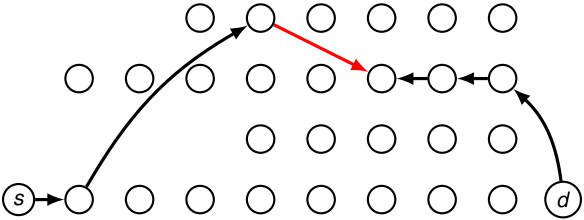
- Z_{RM} — optimal value of the current restricted master.
- Z^k_{sub} — minimum reduced cost for machine $k \in M$.
- Lagrangian lower bound: $Z_{RM} + \sum_{k \in M} Z^k_{sub}$.
- Z_{inc} — value of the best known integer solution.
- $Z^k_{sub}(a)$ — current minimum reduced cost of a path containing arc $a \in G_k$.
- Arc a can be removed (it cannot take part of any improving solution) if

$$Z^k_{sub}(a) + \sum_{k' \in M \setminus \{k\}} Z^{k'}_{sub} + Z_{RM} \geq Z_{inc}.$$

- A good heuristic is very important!

Computing $Z^k_{sub}(a)$ [Ibaraki and Nakamura, 1994]

How to compute the shortest path passing through arc $a = (i, j, t) \in G_k$?



1. $F(i, t)$ — the value of the shortest path from s to node (i, t)
2. $B(k, t + s^k_{ij} + p^k_j)$ — the value of the shortest path from d to node $(j, t + s^k_{ij} + p^k_j)$
3. $Z^k_{sub}(a = (i, j, k)) = F(i, t) + B(j, t + s^k_{ij} + p^k_j) + \overline{c}_j^{\,t+s^k_{ij}+p^k_j}$.

Ibaraki, T. and Nakamura, Y. (1994).
A dynamic programming method for single machine scheduling.
European Journal of Operational Research, 76(1):72 – 82.

Dual price smoothing stabilization

- $\overline{\pi}$ — current dual solution of the restricted master
- π^* — dual solution giving the best Lagrangian bound so far
- We solve the pricing problem using the dual vector

$$\pi' = (1 - \alpha) \cdot \overline{\pi} + \alpha \cdot \pi^*,$$

  where $\alpha \in [0, 1)$.
- Parameter α is automatically adjusted in each column generation iteration using the sub-gradient of the Lagrangian function at π' [Pessoa et al., 2017].

Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2017).
Automation and combination of linear-programming based stabilization techniques in column generation.
INFORMS Journal on Computing, (Forthcoming).

Branching

- Branching on aggregated arc variables

$$\sum_{0 \leq t \leq T} x_{ij}^{tk} \in \{0, 1\},$$

  i.e. job i immediately precedes job j on machine k or not
- Multi-phase strong branching is used
- Branching history is kept is used through pseudo-costs

Contents

Subset-Row Cuts (SRCs) [Jepsen et al., 2008]

Given $C \subseteq J$ and a multiplier ρ, the (C, ρ)-Subset Row Cut is:

$$\sum_{k \in M} \sum_{\omega \in \Omega_k} \left\lfloor \rho \sum_{i \in C} a_j^\omega \right\rfloor \lambda_\omega \leq \lfloor \rho |C| \rfloor$$

Special case of **Chvátal-Gomory rank-1 cuts** obtained by rounding of $|C|$ set-packing constraints in the master

Here we use only 1-row and 3-row cuts with $\rho = \frac{1}{2}$.
We separate them by enumeration.

Mads Jepsen and Bjorn Petersen and Simon Spoorendonk and David Pisinger (2008).
Subset-Row Inequalities Applied to the Vehicle-Routing Problem with Time Windows.
Operations Research, 56(2):497–511.

Impact on the pricing problem

Given dual value $\nu_\gamma < 0$ for each active subset row cut $\gamma \in \Gamma$, defined for subset C_γ of jobs, modified reduced cost of pseudo-schedule $\omega \in \Omega_k$ is :

$$\bar{c}_\omega = \sum_{i,j \in J,\, t \in T} \left(c_j^{t+s_{ij}+p_j} - \pi_j \right) \cdot x_{ij}^t - \sum_{\gamma \in \Gamma} \left\lfloor \frac{1}{2} \cdot \sum_{\substack{j \in C_\eta,\, i \in J, \\ i \neq j,\, t \in T}} x_{ij}^t \right\rfloor .$$

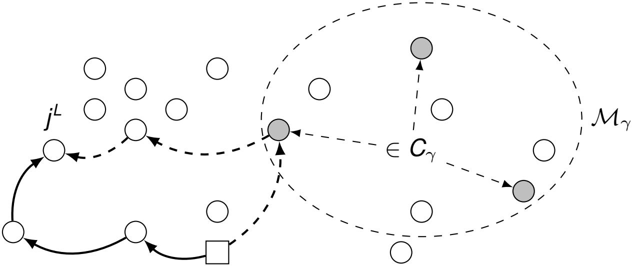
Each cut adds to labels an additional binary state S_γ^L (parity of the number of times jobs in C_γ appear in the partial schedule L), resulting in a weaker domination:

$$\bar{c}^L - \sum_{\gamma \in \Gamma:\, S_\gamma^L > S_\gamma^{L'}} \nu_\gamma \leq \bar{c}^{L'} \quad \text{instead of} \quad \bar{c}^L \leq \bar{c}^{L'}$$

Limited memory cuts [Pecin et al., 2017]

For each active cut $\gamma \in \Gamma$, define a memory $\mathcal{M}_\gamma$ of vertices (jobs) which "remember" state S_γ.

If $j^L \notin \mathcal{M}_\gamma$, then $S^L_\gamma \leftarrow 0$.
Vectors S^L are sparser ⇒ stronger domination



Pecin, D., Pessoa, A., Poggi, M., and Uchoa, E. (2017).
Improved branch-cut-and-price for capacitated vehicle routing.
Mathematical Programming Computation, 9(1):61–100.

Contents

Results for $R \mid r_j, s_{ij}^k \mid \sum \alpha_j E_j + \beta_j T_j$, small setup times

Initial heuristic and instances by [Kramer and Subramanian, 2017]

Size		With cuts						BKS	
n	m	#Solved	Root Gap (%)	Gap (%)	Root Time	Total Time	#Nodes	Improv. (%)	#New
40	2	60/60	0.01	0.00	4m	4m	1.1	0.12	22
60	2	60/60	0.32	0.00	23m	28m	3.5	0.33	46
60	3	60/60	0.86	0.00	16m	35m	10.6	0.48	47
80	2	60/60	0.23	0.00	1h12m	1h37m	5.7	0.14	41
80	4	48/60	1.69	0.52	37m	4h33m	92.0	0.26	50

Size		Without cuts					
n	m	#Solved	Root Gap (%)	Gap (%)	Root Time	Total Time	#Nodes
40	2	60/60	1.72	0.00	3m	6m	44.8
60	2	59/60	1.99	0.05	13m	1h55m	412.8
60	3	60/60	2.23	0.00	10m	1h13m	361.5

Kramer, A. and Subramanian, A. (2017).
A unified heuristic and an annotated bibliography for a large class of earliness-tardiness scheduling problems.
Technical report, Universidad Federal da Paraíba, Brazil.

Results for $R \mid r_j, s_{ij}^k \mid \sum \alpha_j E_j + \beta_j T_j$, larger setup times

Initial heuristic and instances by [Kramer and Subramanian, 2017]

Size		With cuts						BKS	
n	m	#Solved	Root Gap (%)	Gap (%)	Root Time	Total Time	#Nodes	Improv. (%)	#New
40	2	60/60	0.43	0.00	13m	16m	2.8	0.76	46
60	2	58/60	2.22	0.06	48m	2h56m	23.2	1.34	58
60	3	45/60	4.29	1.21	29m	5h45m	85.8	1.56	55
80	2	28/60	2.89	1.32	1h59m	9h49m	48.8	0.80	54
80	4	10/60	5.17	3.91	1h18m	10h58m	120.4	0.39	27

Size		Without cuts					
n	m	#Solved	Root Gap (%)	Gap (%)	Root Time	Total Time	#Nodes
40	2	60/60	4.08	0.00	5m	24m	172.6
60	2	43/60	4.71	1.21	23m	7h06m	1246.2
60	3	37/60	5.99	2.14	18m	7h05m	1702.3

Kramer, A. and Subramanian, A. (2017).
A unified heuristic and an annotated bibliography for a large class of earliness-tardiness scheduling problems.
Technical report, Universidad Federal da Paraíba, Brazil.

Results for $R \,||\, \sum \alpha_j E_j + \beta_j T_j$

Size		Our Branch-Cut-and-Price						BKS		[Şen and Bülbül, 2015]		
n	m	Solv.	Root Gap(%)	Gap (%)	Root Time	Total Time	Nod. num.	Impr. (%)	New	Solv.	Gap (%)	Time
40	2	60/60	0.04	0.00	2m	5m	3.4	0.00	0	26/60	0.16	1m
60	2	60/60	0.04	0.00	9m	12m	3.3	0.00	1	7/60	0.89	2m
60	3	60/60	0.05	0.00	6m	7m	2.9	0.01	5	7/60	0.82	2m
80	2	59/60	0.02	0.00	28m	40m	5.4	0.00	3	2/60	0.90	2m
80	4	60/60	0.11	0.00	15m	16m	3.9	0.07	15	0/60	4.54	4m
90	3	60/60	0.05	0.00	29m	34m	4.7	0.03	20	1/60	2.52	3m
100	5	59/60	0.20	0.02	31m	57m	26.7	0.10	27	0/60	8.83	5m
120	3	56/60	0.16	0.04	1h54m	3h00m	16.7	0.07	22	0/60	4.12	3m
120	4	58/60	0.23	0.01	1h24m	2h12m	17.7	0.17	31	0/60	6.98	4m

With subset row cuts, root gap is 6 times smaller (40 and 60 jobs instances).

In 30 minutes, CPLEX solved 49/60 inst. with 40 jobs, 36/120 inst. with 60 jobs, 3/120 inst. with 80 jobs, 2/60 inst. with 90 jobs.

Şen, H. and Bülbül, K. (2015).
A strong preemptive relaxation for weighted tardiness and earliness/tardiness problems on unrelated parallel machines.
INFORMS Journal on Computing, 27(1):135–150.

Final remarks

- First use of non-robust cuts for scheduling problems
- Significant computational improvement over the existing exact approaches for the problem
  - scales up to 4 machines and 80 jobs for “generic” instances with setup times
  - solves 532/540 instances without setup times with up to 4 machines and 120 jobs
- Need more testing on “less generic” instances
- Ways to improve results:
  - A better heuristic for generic instances is needed!
  - First convergence is very slow
  - More balanced branching
  - Separation for rank-1 cuts with 4 and more rows
  - Enumeration [Baldacci et al., 2008]
  - Avoid discretisation

References I

Balakrishnan, N., Kanet, J. J., and Sridharan, V. (1999).
Early/tardy scheduling with sequence dependent setups on uniform parallel machines.
Computers and Operations Research, 26(2):127 – 141.

Baldacci, R., Christofides, N., and Mingozzi, A. (2008).
An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts.
Mathematical Programming, 115:351–385.

Şen, H. and Bülbül, K. (2015).
A strong preemptive relaxation for weighted tardiness and earliness/tardiness problems on unrelated parallel machines.
INFORMS Journal on Computing, 27(1):135–150.

Ibaraki, T. and Nakamura, Y. (1994).
A dynamic programming method for single machine scheduling.
European Journal of Operational Research, 76(1):72 – 82.

References II

Jepsen, M., Petersen, B., Spoorendonk, S., and Pisinger, D. (2008).
Subset-row inequalities applied to the vehicle-routing problem with time windows.
Operations Research, 56(2):497–511.

Jouglet, A. and Savourey, D. (2011).
Dominance rules for the parallel machine total weighted tardiness scheduling problem with release dates.
Computers and Operations Research, 38(9):1259 – 1266.

Kramer, A. and Subramanian, A. (2017).
A unified heuristic and an annotated bibliography for a large class of earliness-tardiness scheduling problems.
Technical report, Universidad Federal da Paraíba, Brazil.

Liaw, C.-F., Lin, Y.-K., Cheng, C.-Y., and Chen, M. (2003).
Scheduling unrelated parallel machines to minimize total weighted tardiness.
Computers and Operations Research, 30(12):1777 – 1789.

References III

Lopes, M. J. P. and de Carvalho, J. V. (2007).
A branch-and-price algorithm for scheduling parallel machines with sequence dependent setup times.
European Journal of Operational Research, 176(3):1508 – 1527.

Pecin, D., Pessoa, A., Poggi, M., and Uchoa, E. (2017).
Improved branch-cut-and-price for capacitated vehicle routing.
Mathematical Programming Computation, 9(1):61–100.

Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2017).
Automation and combination of linear-programming based stabilization techniques in column generation.
INFORMS Journal on Computing, (Forthcoming).

Pessoa, A., Uchoa, E., de Aragão, M. P., and Rodrigues, R. (2010).
Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems.
Mathematical Programming Computation, 2:259–290.

References IV

Schaller, J. E. (2014).
Minimizing total tardiness for scheduling identical parallel machines with family setups.
Computers and Industrial Engineering, 72:274 – 281.

Shim, S.-O. and Kim, Y.-D. (2007).
Minimizing total tardiness in an unrelated parallel-machine scheduling problem.
Journal of the Operational Research Society, 58(3):346–354.