Resource-Window Reduction by Reduced Costs in Path-based Formulations for Routing and Scheduling Problems

Column Generation, Montréal, 2023

Nicola Bianchessi Timo Gschwind Stefan Irnich

irnich@uni-mainz.de

Chair for Logistics Management
Gutenberg School of Management and Economics

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Overall Idea

Reduce resource windows (beyond preprocessing) with the help of reduced costs.

- Tighter upper bounds lead to less feasible labels

Overall Idea

Reduce resource windows (beyond preprocessing) with the help of reduced costs.

- Tighter upper bounds lead to less feasible labels
- Tighter lower bounds lead to more labels that are comparable

Overall Idea

Reduce resource windows (beyond preprocessing) with the help of reduced costs.

- Tighter upper bounds lead to less feasible labels
- Tighter lower bounds lead to more labels that are comparable
- Faster pricing accelerates overall BPC algorithm (major effect)

Reduce resource windows (beyond preprocessing) with the help of reduced costs.

- Tighter upper bounds lead to less feasible labels
- Tighter lower bounds lead to more labels that are comparable
- Faster pricing accelerates overall BPC algorithm (major effect)
- Some paths/columns of the RMP can be eliminated so that the lower bound of the (minor effect)

Overall Idea

Reduce resource windows (beyond preprocessing) with the help of reduced costs.

- Tighter upper bounds lead to less feasible labels
- Tighter lower bounds lead to more labels that are comparable
- Faster pricing accelerates overall BPC algorithm (major effect)
- Some paths/columns of the RMP can be eliminated so that the lower bound of the (minor effect)

Overall Idea

Reduce resource windows (beyond preprocessing) with the help of reduced costs.

- Tighter upper bounds lead to less feasible labels
- Tighter lower bounds lead to more labels that are comparable
- Faster pricing accelerates overall BPC algorithm (major effect)
- Some paths/columns of the RMP can be eliminated so that the lower bound of the (minor effect)

Take away:

- Direct resource-window reduction not helpful
- Consider arc-specific resource windows

Reduced Cost-based Variable Elimination/Fixing

Proposition 1 (Nemhauser and Wolsey (1988), Proposition 2.1, page 389)

Let UB be an upper bound on the optimal value of the minimization problem M, and let π be a dual solution to the linear relaxation of M providing a lower bound $LB(\pi)$.
If an integer variable $x \geq 0$ has reduced cost $\tilde{c}_x(\pi) > UB - LB(\pi)$, then $x = 0$ in every optimal solution to M, i.e., x can be eliminated.

Reduced Cost-based Variable Elimination/Fixing

Proposition 1 (Nemhauser and Wolsey (1988), Proposition 2.1, page 389)

Let UB be an upper bound on the optimal value of the minimization problem M, and let π be a dual solution to the linear relaxation of M providing a lower bound $LB(\pi)$.
If an integer variable $x \geq 0$ has reduced cost $\tilde{c}_x(\pi) > UB - LB(\pi)$, then $x = 0$ in every optimal solution to M, i.e., x can be eliminated.

No direct application in column generation:

- Forbidding the re-generation of one or several variables changes the structure of the pricing problem
- Effort of solving modified pricing problems is often too high
- Possible via network modification (Villeneuve and Desaulniers, 2005)

Notation

$$\min \sum_{p\in\mathcal{P}} c_p\lambda_p \tag{1a}$$

$$\text{s.t.} \sum_{p\in\mathcal{P}} a_{kp}\lambda_p = 1 \qquad \forall k \in K \qquad [\pi] \tag{1b}$$

$$\lambda_p \geq 0 \quad \text{integer} \qquad \forall p \in \mathcal{P} \tag{1c}$$

with

K: set of tasks to fulfill

$\mathcal{P}$: set of all resource feasible paths, underlying network $D = (V, A)$

MP: linear relaxation, i.e., $\lambda_p \geq 0$, $\lambda_p \in \mathbb{R}$

RMP: linear relaxation defined of subset $\mathcal{P}' \subset \mathcal{P}$

π: dual solution of MP/RMP

Reduced Cost-based Arc Fixing

Consider an arc $(i, j) \in A$ and the set $\mathcal{P}[ij]$ of all paths that contain the arc (i, j).

Reduced Cost-based Arc Fixing

Consider an arc $(i, j) \in A$ and the set $\mathcal{P}[ij]$ of all paths that contain the arc (i, j).

Proposition 2 (Irnich *et al.* (2010))

If the reduced cost $\tilde{c}_p(\pi)$ of all variables λ_p for $p \in \mathcal{P}[ij]$ fulfill $\tilde{c}_p(\pi) \geq UB - LB(\pi)$, then the arc (i, j) is not used in an optimal solution and can be eliminated.

Reduced Cost-based Arc Fixing

Consider an arc $(i,j) \in A$ and the set $\mathcal{P}[ij]$ of all paths that contain the arc (i,j).

Proposition 2 (Irnich *et al.* (2010))

If the reduced cost $\tilde{c}_p(\pi)$ of all variables λ_p for $p \in \mathcal{P}[ij]$ fulfill $\tilde{c}_p(\pi) \geq UB - LB(\pi)$, then the arc (i,j) is not used in an optimal solution and can be eliminated.

More general: Paths $\mathcal{P}[prop]$ is the subset of all $\mathcal{P}$ that fulfill a given property *prop*.

$$\tilde{c}[prop](\pi) = \min_{p \in \mathcal{P}[prop]} \tilde{c}_p(\pi)$$

Reduced Cost-based Arc Fixing

Consider an arc $(i, j) \in A$ and the set $\mathcal{P}[ij]$ of all paths that contain the arc (i, j).

Proposition 2 (Irnich *et al.* (2010))

If the reduced cost $\tilde{c}_p(\pi)$ of all variables λ_p for $p \in \mathcal{P}[ij]$ fulfill $\tilde{c}_p(\pi) \geq UB - LB(\pi)$, then the arc (i, j) is not used in an optimal solution and can be eliminated.

More general: Paths $\mathcal{P}[prop]$ is the subset of all $\mathcal{P}$ that fulfill a given property *prop*.

$$\tilde{c}[prop](\pi) = \min_{p \in \mathcal{P}[prop]} \tilde{c}_p(\pi)$$

Two important properties are:

- $[ij]$: For an arc $(i, j) \in A$, the path includes the arc (i, j) at least once;
- $[hij]$: For two arcs $(h, i), (i, j) \in A$, the path includes the sequence (h, i, j) at least once; (Desaulniers *et al.*, 2018)

The values $\tilde{c}[ij](\pi)$ can be effectively computed for all arcs $(i, j) \in A$ with the help of the forward REFs f_{ij}, backward REFs b_{ij}, and the merge operator m (return value of m is the reduced cost):

Reduced Cost-based Variable Elimination/Fixing

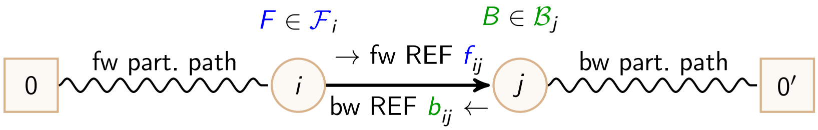
The values $\tilde{c}[ij](\pi)$ can be effectively computed for all arcs $(i,j) \in A$ with the help of the forward REFs f_{ij}, backward REFs b_{ij}, and the merge operator m (return value of m is the reduced cost):

- Full forward and full backward labeling gives label sets $(\mathcal{F}_i)$ and $(\mathcal{B}_i)$
- $f_{ij}(F) = b_{ij}(B) = \infty$, if infeasible; $m(\cdot,\cdot) = \infty$, if infeasible

The values $\tilde{c}[ij](\pi)$ can be effectively computed for all arcs $(i,j) \in A$ with the help of the forward REFs f_{ij}, backward REFs b_{ij}, and the merge operator m (return value of m is the reduced cost):

- Full forward and full backward labeling gives label sets $(\mathcal{F}_i)$ and $(\mathcal{B}_i)$
- $f_{ij}(F) = b_{ij}(B) = \infty$, if infeasible; $m(\cdot,\cdot) = \infty$, if infeasible

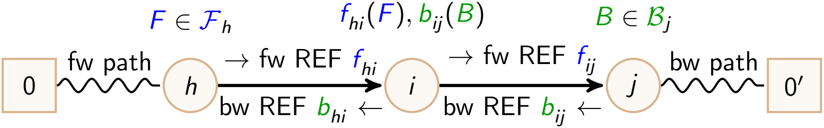
$$\tilde{c}[ij](\pi) = \min_{\substack{F \in \mathcal{F}_i, \\ B \in \mathcal{B}_j}} m(f_{ij}(F), B) = \min_{\substack{F \in \mathcal{F}_i, \\ B \in \mathcal{B}_j}} m(F, b_{ij}(B))$$



Reduced Cost-based Two-Arc Fixing

Two-arc fixing was suggested by Desaulniers *et al.* (2018):

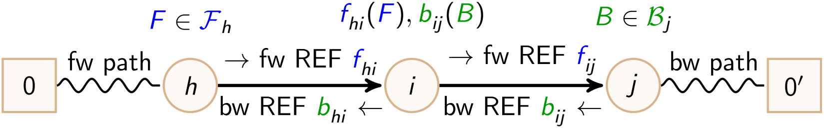
$$\tilde{c}[hij](\pi) = \min_{\substack{F \in \mathcal{F}_h, \\ B \in \mathcal{B}_j}} m(f_{hi}(F), b_{ij}(B))$$



Reduced Cost-based Two-Arc Fixing

Two-arc fixing was suggested by Desaulniers *et al.* (2018):

$$\tilde{c}[hij](\pi) = \min_{\substack{F \in \mathcal{F}_h, \\ B \in \mathcal{B}_j}} m(f_{hi}(F), b_{ij}(B))$$



- Two-arc sequences cannot be eliminated from the network
- Must be eliminated during label extension
- Modified dominance comparison between labels required

Time Windows

For time window VRPs (with TWs $[e_i, \ell_i]$, travel times τ_{ij}), the standard forward and backward REFs propagate the time attribute T^{time} in the following way:

$$T_j^{time} = f_{ij}^{time}(T_i) = \max\{e_j, T_i^{time} + \tau_{ij}\} \quad \text{feasible if} \quad T_j^{time} \leq \ell_j$$
$$T_i^{time} = b_{ij}^{time}(T_j) = \min\{\ell_i, T_j^{time} - \tau_{ij}\} \quad \text{feasible if} \quad T_i^{time} \geq e_i$$

Time Windows

For time window VRPs (with TWs $[e_i, \ell_i]$, travel times τ_{ij}), the standard forward and backward REFs propagate the time attribute T^{time} in the following way:

$$T_j^{time} = f_{ij}^{time}(T_i) = \max\{e_j, T_i^{time} + \tau_{ij}\} \quad \text{feasible if} \quad T_j^{time} \leq \ell_j$$
$$T_i^{time} = b_{ij}^{time}(T_j) = \min\{\ell_i, T_j^{time} - \tau_{ij}\} \quad \text{feasible if} \quad T_i^{time} \geq e_i$$

Three properties of resource-feasible paths $p \in \mathcal{P}$ related to a vertex $i \in V$ and an arbitrary point in time $t \in \mathbb{R}$:

$[T_i^{time} < t]$: The path can service/visit vertex i before t;

Time Windows

For time window VRPs (with TWs $[e_i, \ell_i]$, travel times τ_{ij}), the standard forward and backward REFs propagate the time attribute T^{time} in the following way:

$$T_j^{time} = f_{ij}^{time}(T_i) = \max\{e_j, T_i^{time} + \tau_{ij}\} \quad \text{feasible if} \quad T_j^{time} \leq \ell_j$$
$$T_i^{time} = b_{ij}^{time}(T_j) = \min\{\ell_i, T_j^{time} - \tau_{ij}\} \quad \text{feasible if} \quad T_i^{time} \geq e_i$$

Three properties of resource-feasible paths $p \in \mathcal{P}$ related to a vertex $i \in V$ and an arbitrary point in time $t \in \mathbb{R}$:

$[T_i^{time} < t]$: The path can service/visit vertex i before t; if i occurs several times in path p, at least one of the services must start strictly before time t;

Time Windows

For time window VRPs (with TWs $[e_i, \ell_i]$, travel times τ_{ij}), the standard forward and backward REFs propagate the time attribute T^{time} in the following way:

$$T_j^{time} = f_{ij}^{time}(T_i) = \max\{e_j, T_i^{time} + \tau_{ij}\} \quad \text{feasible if} \quad T_j^{time} \leq \ell_j$$
$$T_i^{time} = b_{ij}^{time}(T_j) = \min\{\ell_i, T_j^{time} - \tau_{ij}\} \quad \text{feasible if} \quad T_i^{time} \geq e_i$$

Three properties of resource-feasible paths $p \in \mathcal{P}$ related to a vertex $i \in V$ and an arbitrary point in time $t \in \mathbb{R}$:

- $[T_i^{time} < t]$: The path can service/visit vertex i before t; if i occurs several times in path p, at least one of the services must start strictly before time t;
- $[T_i^{time} > t]$: Likewise with a possible service start/visit at vertex i at a time strictly after time t;

Time Windows

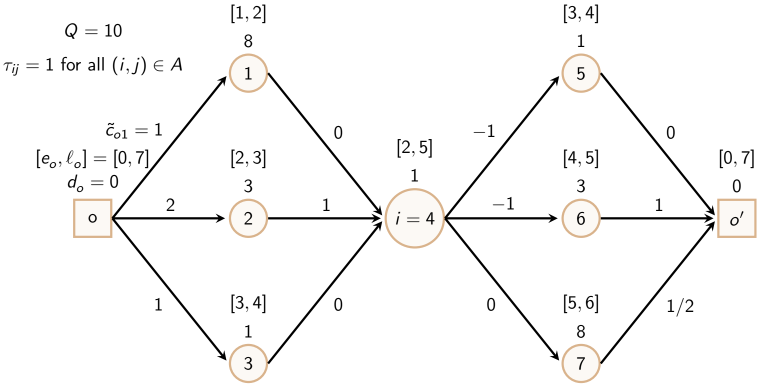
For time window VRPs (with TWs $[e_i, \ell_i]$, travel times τ_{ij}), the standard forward and backward REFs propagate the time attribute T^{time} in the following way:

$$
\begin{aligned}
T_j^{time} &= f_{ij}^{time}(T_i) = \max\{e_j, T_i^{time} + \tau_{ij}\} &\quad \text{feasible if} \quad & T_j^{time} \leq \ell_j \\
T_i^{time} &= b_{ij}^{time}(T_j) = \min\{\ell_i, T_j^{time} - \tau_{ij}\} &\quad \text{feasible if} \quad & T_i^{time} \geq e_i
\end{aligned}
$$

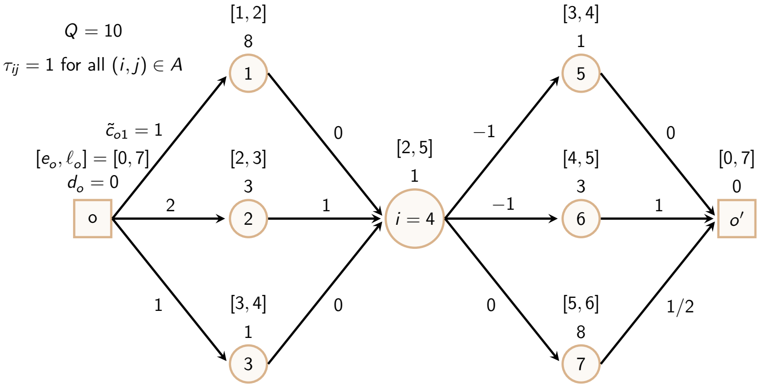
Three properties of resource-feasible paths $p \in \mathcal{P}$ related to a vertex $i \in V$ and an arbitrary point in time $t \in \mathbb{R}$:

- $[T_i^{time} < t]$: The path can service/visit vertex i before t; if i occurs several times in path p, at least one of the services must start strictly before time t;
- $[T_i^{time} > t]$: Likewise with a possible service start/visit at vertex i at a time strictly after time t;
- $[T_i^{time} = t]$: Likewise with a service/visit at vertex i starting exactly at time t;

Example: VRPTW; Consider paths and service times at vertex $i = 4$:



Example: VRPTW; Consider paths and service times at vertex $i = 4$:



Feasible path $p \in \mathcal{P}$	Possible values T_4^{time}	Reduced cost $\tilde{c}_p(\pi)$
$(o, 1, 4, 5, o')$	$[2, 3]$	0
$(o, 2, 4, 5, o')$	$[3, 3]$	2
$(o, 2, 4, 6, o')$	$[3, 4]$	3
$(o, 3, 4, 6, o')$	$[4, 4]$	1
$(o, 3, 4, 7, o')$	$[4, 5]$	1.5

Infeasible path	Reason for infeasibility
$(o, 1, 4, 6, o')$	load $\sum d_i = 12$
$(o, 1, 4, 7, o')$	load $\sum d_i = 17$
$(o, 2, 4, 7, o')$	load $\sum d_i = 12$
$(o, 3, 4, 5, o')$	time window at vertex 5

Time Windows

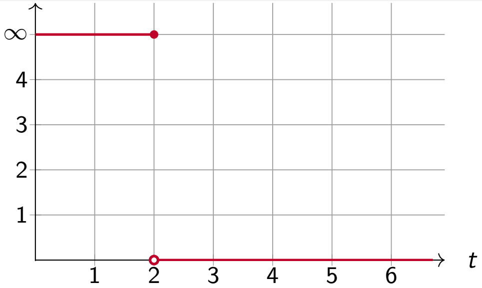
The values $\tilde{c}[T_i^{time} < t](\pi)$, $\tilde{c}[T_i^{time} < t](\pi)$, and $\tilde{c}[T_i^{time} = t](\pi)$ can be effectively computed for all vertices $i \in V$ with the help of the forward REFs f_{ij}, backward REFs b_{ij}, and the merge operator m (return value of m is reduced cost):

Time Windows

The values $\tilde{c}[T_i^{time} < t](\pi)$, $\tilde{c}[T_i^{time} < t](\pi)$, and $\tilde{c}[T_i^{time} = t](\pi)$ can be effectively computed for all vertices $i \in V$ with the help of the forward REFs f_{ij}, backward REFs b_{ij}, and the merge operator m (return value of m is reduced cost):

$$
\begin{aligned}
\tilde{c}[T_i^{time} < t](\pi) &= \min_{\substack{F \in \mathcal{F}_i : F^{time} < t \\ B \in \mathcal{B}_i}} m(F, B) \\
\tilde{c}[T_i^{time} > t](\pi) &= \min_{\substack{F \in \mathcal{F}_i, \\ B \in \mathcal{B}_i : B^{time} > t}} m(F, B) \\
\tilde{c}[T_i^{time} = t](\pi) &= \min_{\substack{F \in \mathcal{F}_i : F^{time} \leq t \\ B \in \mathcal{B}_i : B^{time} \geq t}} m(F, B)
\end{aligned}
$$

Time Windows

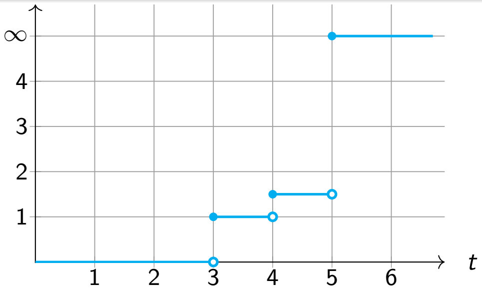


Feasible path $p \in \mathcal{P}$	Possible values T_4^{time}	Reduced cost $\tilde{c}_p(\pi)$
$(0,1,4,5,8)$	$[2,3]$	0
$(0,2,4,5,8)$	$[3,3]$	2
$(0,2,4,6,8)$	$[3,4]$	3
$(0,3,4,6,8)$	$[4,4]$	1
$(0,3,4,7,8)$	$[4,5]$	1.5

Proposition 3

The lower bound functions have the following properties in terms of $t \in \mathbb{R}$:

(i) $\tilde{c}[T_i^{time} < t](\pi)$ *is a non-increasing, piecewise constant function; it is continuous from the left;*



Feasible path $p \in \mathcal{P}$	Possible values T_4^{time}	Reduced cost $\tilde{c}_p(\pi)$
$(0, 1, 4, 5, 8)$	$[2, 3]$	0
$(0, 2, 4, 5, 8)$	$[3, 3]$	2
$(0, 2, 4, 6, 8)$	$[3, 4]$	3
$(0, 3, 4, 6, 8)$	$[4, 4]$	1
$(0, 3, 4, 7, 8)$	$[4, 5]$	1.5

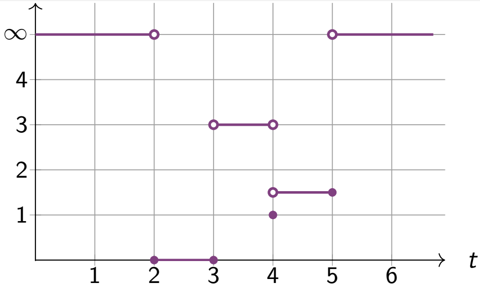
Proposition 3

The lower bound functions have the following properties in terms of $t \in \mathbb{R}$:

(i) $\tilde{c}[T_i^{time} < t](\pi)$ *is a non-increasing, piecewise constant function; it is continuous from the left;*

(ii) $\tilde{c}[T_i^{time} > t](\pi)$ *is a non-decreasing, piecewise constant function; it is continuous from the right;*

Time Windows



Feasible path $p \in \mathcal{P}$	Possible values T_4^{time}	Reduced cost $\tilde{c}_p(\pi)$
$(0, 1, 4, 5, 8)$	$[2, 3]$	0
$(0, 2, 4, 5, 8)$	$[3, 3]$	2
$(0, 2, 4, 6, 8)$	$[3, 4]$	3
$(0, 3, 4, 6, 8)$	$[4, 4]$	1
$(0, 3, 4, 7, 8)$	$[4, 5]$	1.5

Proposition 3

The lower bound functions have the following properties in terms of $t \in \mathbb{R}$:

(i) $\tilde{c}[T_i^{time} < t](\pi)$ *is a non-increasing, piecewise constant function; it is continuous from the left;*

(ii) $\tilde{c}[T_i^{time} > t](\pi)$ *is a non-decreasing, piecewise constant function; it is continuous from the right;*

(iii) $\tilde{c}[T_i^{time} = t](\pi)$ *is a piecewise constant function; it is not necessarily monotone;*

(iv) $\tilde{c}[T_i^{time} = t](\pi) \geq \max\{\tilde{c}[T_i^{time} \leq t](\pi), \tilde{c}[T_i^{time} \geq t](\pi)\}$ *holds true; strictly $>$ is possible.*

Time Windows

For any property [*prop*],

$$\begin{aligned} &\tilde{c}[prop](\pi) > UB - LB(\pi) \\ \Leftrightarrow \quad & LB[prop](\pi) := LB(\pi) + \tilde{c}[prop](\pi) > UB. \end{aligned}$$

Time Windows

For any property $[prop]$,

$$\begin{aligned} \tilde{c}[prop](\pi) &> UB - LB(\pi) \\ \Leftrightarrow \quad LB[prop](\pi) &:= LB(\pi) + \tilde{c}[prop](\pi) > UB. \end{aligned}$$

Proposition 4

The following rules are valid for all $t \in \mathbb{R}$:

(i) *If $LB[T_i^{time} < t](\pi) > UB$, the value e_i can be updated to $\max\{e_i, t\}$.*

(ii) *If $LB[T_i^{time} > t](\pi) > UB$, the value ℓ_i can be updated to $\min\{t, \ell_i\}$.*

For any property $[prop]$,

$$\begin{aligned}&\tilde{c}[prop](\pi) > UB - LB(\pi)\\ \Leftrightarrow \quad & LB[prop](\pi) := LB(\pi) + \tilde{c}[prop](\pi) > UB.\end{aligned}$$

Proposition 4

The following rules are valid for all $t \in \mathbb{R}$:

(i) *If $LB[T_i^{time} < t](\pi) > UB$, the value e_i can be updated to $\max\{e_i, t\}$.*

(ii) *If $LB[T_i^{time} > t](\pi) > UB$, the value ℓ_i can be updated to $\min\{t, \ell_i\}$.*

Remarks:

- We do not use property $[T_i^{time} = t]$, because the time windows would then be split into multiple smaller time windows per vertex.
- For different dual prices $\pi^1, \pi^2, \ldots, \pi^k$, the lower bound functions can be combined into

$$LB^*[prop] := \max_{s=1,\ldots,k} \{LB[prop](\pi^s)\}.$$

Load-Resource Windows

Standard REFs for capacity constraints start with an initial load of 0 for the forward and backward labeling and propagate the load-attributes as follows:

$$T_j^{load} = f_{ij}^{load}(T_i) = T_i^{load} + d_j \qquad \text{feasible if} \quad T_j^{load} \leq Q$$
$$T_i^{load} = b_{ij}^{load}(T_j) = T_j^{load} + d_i \qquad \text{feasible if} \quad T_i^{load} \leq Q$$

Load-Resource Windows

Standard REFs for capacity constraints start with an initial load of 0 for the forward and backward labeling and propagate the load-attributes as follows:

$$T_j^{load} = f_{ij}^{load}(T_i) = T_i^{load} + d_j \qquad \text{feasible if} \quad T_j^{load} \leq Q$$
$$T_i^{load} = b_{ij}^{load}(T_j) = T_j^{load} + d_i \qquad \text{feasible if} \quad T_i^{load} \leq Q$$

Problem: fw and bw values are not referring to the same resource window.

Load-Resource Windows

Standard REFs for capacity constraints start with an initial load of 0 for the forward and backward labeling and propagate the load-attributes as follows:

$$
\begin{aligned}
T_j^{load} &= f_{ij}^{load}(T_i) = T_i^{load} + d_j && \text{feasible if} \quad T_j^{load} \le Q \\
T_i^{load} &= b_{ij}^{load}(T_j) = T_j^{load} + d_i && \text{feasible if} \quad T_i^{load} \le Q
\end{aligned}
$$

Problem: fw and bw values are not referring to the same resource window.

Solution: In bw direction, consider the residual capacity and delay the demand propagation:

$$
\begin{aligned}
T_j^{l\tilde{oa}d} &= f_{ij}^{l\tilde{oa}d}(T_i) = \max\{d_j, T_i^{l\tilde{oa}d} + d_j\} && \text{feasible if} \quad T_j^{l\tilde{oa}d} \le \bar{Q}_j \\
T_i^{l\tilde{oa}d} &= b_{ij}^{l\tilde{oa}d}(T_j) = \min\{Q_i, T_j^{l\tilde{oa}d} - d_j\} && \text{feasible if} \quad T_i^{l\tilde{oa}d} \ge \bar{d}_i
\end{aligned}
$$

with initial value $\bar{d}_o = 0$ at origin o, and value $\bar{Q}_{o'} = Q$ at destination o'.

Results for VRPTW

Results for VRPTW

Instances: Solomon VRPTW benchmark with 25, 50 and 100 customers

- 86 instances solved in root node
- R208.100 and R211.100 not solved within 2 hours
- 80 instance in experiments

Results for VRPTW

Instances: Solomon VRPTW benchmark with 25, 50 and 100 customers

- 86 instances solved in root node
- R208.100 and R211.100 not solved within 2 hours
- 80 instance in experiments

BPC algorithm:

- Standard components: *preprocessing* (Desrochers *et al.*, 1992); *ng-route* relaxation (Baldacci *et al.*, 2011); *bidirectional labeling* with *dynamic HWP* (Tilk *et al.*, 2017); *partial pricing* with reduced networks (Desaulniers *et al.*, 2008); *arc fixing* (Irnich *et al.*, 2010); *limited-memory subset-row inequalities* (lm-SRIs, Pecin *et al.*, 2017) for subsets S of rows with $|S| = 3$; *branching* on #vehicle and arcs with best-first search
- $UB := opt + 1$
- time limit 7200 seconds (2 hours)

Results for VRPTW

Instances: Solomon VRPTW benchmark with 25, 50 and 100 customers

- 86 instances solved in root node
- R208.100 and R211.100 not solved within 2 hours
- 80 instance in experiments

BPC algorithm:

- Standard components: *preprocessing* (Desrochers *et al.*, 1992); *ng-route* relaxation (Baldacci *et al.*, 2011); *bidirectional labeling* with *dynamic HWP* (Tilk *et al.*, 2017); *partial pricing* with reduced networks (Desaulniers *et al.*, 2008); *arc fixing* (Irnich *et al.*, 2010); *limited-memory subset-row inequalities* (lm-SRIs, Pecin *et al.*, 2017) for subsets S of rows with $|S| = 3$; *branching* on #vehicle and arcs with best-first search
- $UB := opt + 1$
- time limit 7200 seconds (2 hours)

Computational setup:

- `C++` using the callable library of `CPLEX` 12.7.0 and compiled into 64-bit single-thread release code with Microsoft Visual Studio 2017
- 64-bit Microsoft Windows 10 computer with an Intel® Core™ i7-6700K clocked at 4.00 GHz and with 32 GB of RAM

Results for VRPTW

The following seven computational settings are analyzed:

- **AF**: arc fixing (AF), no resource-window reduction; this is the baseline setting;
- **AF/LD**: AF and resource-window reduction for the $\widetilde{load}$-attribute;
- **AF/TW**: AF and time-window reduction, i.e., for the *time*-attribute;
- **AF/LD.TW**: AF and resource-window reduction for $\widetilde{load}$ and *time*;
- **LD**: w/o AF, but with resource-window reduction for the $\widetilde{load}$-attribute;
- **TW**: w/o AF, but with time-window reduction, i.e., for the *time*-attribute;
- **LD.TW**: w/o AF, but with resource-window reduction for $\widetilde{load}$ and *time*

Results for VRPTW

		Computational Settings						
		AF	AF/LD	AF/TW	AF/LD.TW	LD	TW	LD.TW
Geometric mean time ratios		1.00	1.02	1.04	1.05	2.33	2.33	2.11
#Opt		78	78	77	78	72	73	75
#Unsolved		2	2	3	2	8	7	5
arcs eliminated (%)	min.	60.51	59.42	60.51	59.42			
	avg.	88.70	88.69	88.68	88.70			
	max.	96.67	96.43	96.49	96.43			
$\widetilde{load}$-window reduction (%)	min.		2.00		2.00	2.00		2.00
	avg.		12.80		12.91	13.10		13.06
	max.		33.92		35.74	38.70		36.40
time-window reduction (%)	min.			0.00	0.08		0.08	1.12
	avg.			14.50	14.67		14.26	14.20
	max.			30.25	32.04		31.80	32.57

Results for VRPTW

		Computational Settings						
		AF	AF/LD	AF/TW	AF/LD.TW	LD	TW	LD.TW
Geometric mean time ratios		1.00	1.02	1.04	1.05	2.33	2.33	2.11
#Opt		78	78	77	78	72	73	75
#Unsolved		2	2	3	2	8	7	5
arcs eliminated (%)	min.	60.51	59.42	60.51	59.42			
	avg.	88.70	88.69	88.68	88.70			
	max.	96.67	96.43	96.49	96.43			
$\widetilde{load}$-window reduction (%)	min.		2.00		2.00	2.00		2.00
	avg.		12.80		12.91	13.10		13.06
	max.		33.92		35.74	38.70		36.40
time-window reduction (%)	min.			0.00	0.08		0.08	1.12
	avg.			14.50	14.67		14.26	14.20
	max.			30.25	32.04		31.80	32.57

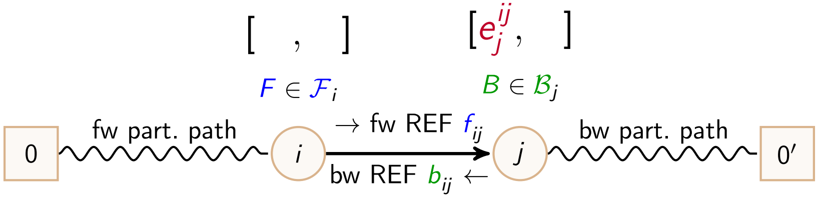
Results for VRPTW

		Computational Settings						
		AF	AF/LD	AF/TW	AF/LD.TW	LD	TW	LD.TW
Geometric mean time ratios		1.00	1.02	1.04	1.05	2.33	2.33	2.11
#Opt		78	78	77	78	72	73	75
#Unsolved		2	2	3	2	8	7	5
arcs eliminated (%)	min.	60.51	59.42	60.51	59.42			
	avg.	88.70	88.69	88.68	88.70			
	max.	96.67	96.43	96.49	96.43			
$\widetilde{load}$-window reduction (%)	min.		2.00		2.00	2.00		2.00
	avg.		12.80		12.91	13.10		13.06
	max.		33.92		35.74	38.70		36.40
time-window reduction (%)	min.			0.00	0.08		0.08	1.12
	avg.			14.50	14.67		14.26	14.20
	max.			30.25	32.04		31.80	32.57

Arc-Specific Resource Windows

Redefine the time-related parts of forward and backward REFs with the four arc-specific attributes $e_i^{ij}, e_j^{ij}, \ell_i^{ij}$, and ℓ_j^{ij} in the following way:

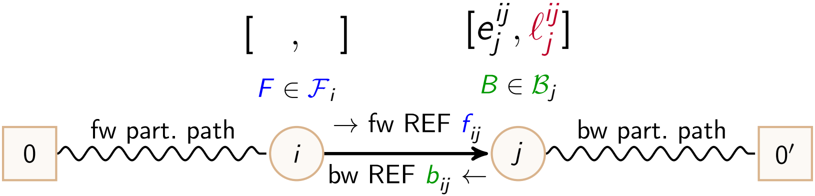
$$T_j^{time} = f_{ij}^{time}(T_i) = \max\{e_j^{ij}, T_i^{time} + \tau_{ij}\}$$



Arc-Specific Resource Windows

Redefine the time-related parts of forward and backward REFs with the four arc-specific attributes $e_i^{ij}, e_j^{ij}, \ell_i^{ij}$, and ℓ_j^{ij} in the following way:

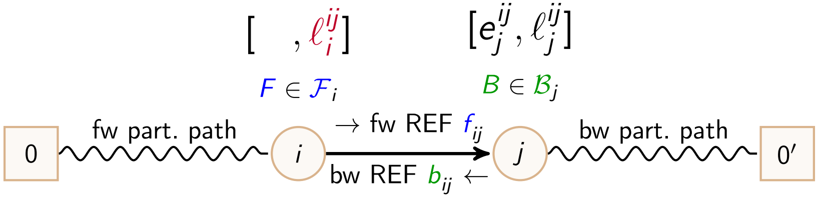
$$T_j^{time} = f_{ij}^{time}(T_i) = \max\{e_j^{ij}, T_i^{time} + \tau_{ij}\} \quad \text{feasible if} \quad T_j^{time} \leq \ell_j^{ij}$$



Arc-Specific Resource Windows

Redefine the time-related parts of forward and backward REFs with the four arc-specific attributes $e_i^{ij}, e_j^{ij}, \ell_i^{ij}$, and ℓ_j^{ij} in the following way:

$$T_j^{time} = f_{ij}^{time}(T_i) = \max\{e_j^{ij}, T_i^{time} + \tau_{ij}\} \quad \text{feasible if} \quad T_j^{time} \leq \ell_j^{ij}$$
$$T_i^{time} = b_{ij}^{time}(T_j) = \min\{\ell_i^{ij}, T_j^{time} - \tau_{ij}\}$$

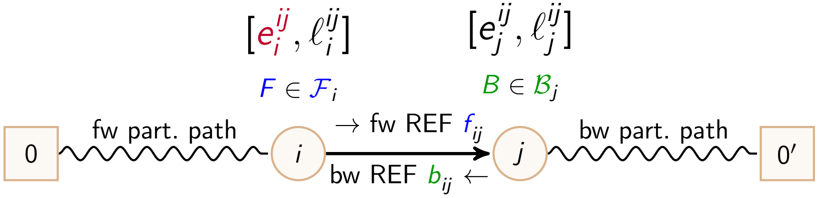


Arc-Specific Resource Windows

Redefine the time-related parts of forward and backward REFs with the four arc-specific attributes $e_i^{ij}, e_j^{ij}, \ell_i^{ij}$, and ℓ_j^{ij} in the following way:

$$T_j^{time} = f_{ij}^{time}(T_i) = \max\{e_j^{ij}, T_i^{time} + \tau_{ij}\} \quad \text{feasible if} \quad T_j^{time} \leq \ell_j^{ij}$$

$$T_i^{time} = b_{ij}^{time}(T_j) = \min\{\ell_i^{ij}, T_j^{time} - \tau_{ij}\} \quad \text{feasible if} \quad T_i^{time} \geq e_i^{ij}$$



Arc-Specific Resource Windows

New properties:

$[T_i^{time} < t, ij]$: All paths that include arc (i, j) and allow a start of service at vertex i before time t followed by the traversal of arc (i, j);

Arc-Specific Resource Windows

New properties:

- $[T_i^{time} < t, ij]$: All paths that include arc (i, j) and allow a start of service at vertex i before time t followed by the traversal of arc (i, j);
- $[ij, T_j^{time} < t]$: All paths that include arc (i, j) and allow a start of service at vertex j before time t after the traversal of arc (i, j).

Likewise for $> t$.

New properties:

$[T_i^{time} < t, ij]$: All paths that include arc (i,j) and allow a start of service at vertex i before time t followed by the traversal of arc (i,j);

$[ij, T_j^{time} < t]$: All paths that include arc (i,j) and allow a start of service at vertex j before time t after the traversal of arc (i,j).

Likewise for $> t$.

$$\tilde{c}[T_i^{time} < t, ij](\pi) = \min_{\substack{F \in \mathcal{F}_i : F^{time} < t \\ B \in \mathcal{B}_j}} m(F, b_{ij}(B))$$

$$\tilde{c}[T_i^{time} > t, ij](\pi) = \min_{\substack{F \in \mathcal{F}_i, \\ B \in \mathcal{B}_j : b_{ij}^{time}(B) > t}} m(F, b_{ij}(B))$$

$$\tilde{c}[ij, T_j^{time} < t](\pi) = \min_{\substack{F \in \mathcal{F}_i : f_{ij}^{time}(F) < t \\ B \in \mathcal{B}_j}} m(f_{ij}(F), B)$$

$$\tilde{c}[ij, T_j^{time} > t](\pi) = \min_{\substack{F \in \mathcal{F}_i, \\ B \in \mathcal{B}_j : B^{time} > t}} m(f_{ij}(F), B)$$

Arc-Specific Resource Windows

Proposition 5

The following four rules are valid for all $t \in \mathbb{R}$:

(i) *If $LB[T_i^{time} < t, ij](\pi) > UB$, the value e_i^{ij} can be updated to* $\max\{e_i^{ij}, t\}$.

(ii) *If $LB[T_i^{time} > t, ij](\pi) > UB$, the value ℓ_i^{ij} can be updated to* $\min\{t, \ell_i^{ij}\}$.

(iii) *If $LB[ij, T_j^{time} < t](\pi) > UB$, the value e_j^{ij} can be updated to* $\max\{e_j^{ij}, t\}$.

(iv) *If $LB[ij, T_j^{time} > t](\pi) > UB$, the value ℓ_j^{ij} can be updated to* $\min\{t, \ell_j^{ij}\}$.

Arc-Specific Resource Windows

Algorithm 1: Reduction Procedure for Arc-specific TWs

Input: Label sets $\mathcal{F}_i$ and $\mathcal{B}_i$ for all $i \in V$, upper bound UB, and arc-specific TWs $[e_i^{ij}, \ell_i^{ij}]$ and $[e_j^{ij}, \ell_j^{ij}]$ for all $(i,j) \in A$

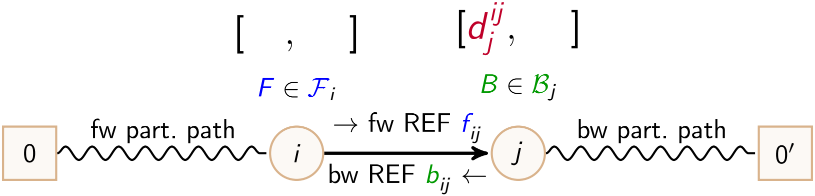
```
for (i,j) ∈ A do
    e_i = e_j ← ∞, ℓ_i = ℓ_j ← −∞;
    for F ∈ 𝓕_i do
        for B ∈ 𝓑_j do
            if m(f_ij(F), B) ≤ UB then
                e_i ← min{F^time, e_i},  e_j ← min{f_ij^time(F), e_j};
                ℓ_i ← max{b_ij^time(B), ℓ_i}, ℓ_j ← max{B^time, ℓ_j};
    e_i^ij ← max{e_i^ij, e_i}, e_j^ij ← max{e_j^ij, e_j};
    ℓ_i^ij ← min{ℓ_i^ij, ℓ_i}, ℓ_j^ij ← min{ℓ_j^ij, ℓ_j};
```

Output: Updated arc-specific TWs $[e_i^{ij}, \ell_i^{ij}]$ and $[e_j^{ij}, \ell_j^{ij}]$ for all $(i,j) \in A$

Arc-Specific Resource Windows

Redefine the load-related parts of forward and backward REFs with the four arc-specific attributes $d_i^{ij}, d_j^{ij}, Q_i^{ij}$, and Q_j^{ij} in the following way:

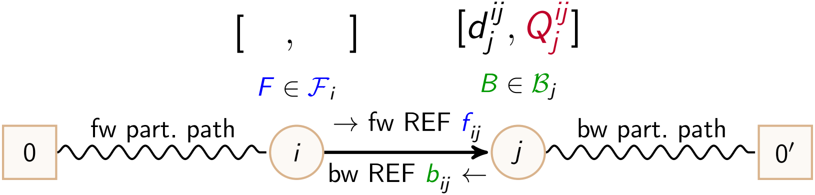
$$T_j^{\widetilde{load}} = f_{ij}^{\widetilde{load}}(T_i) = \max\{d_j^{ij}, T_i^{\widetilde{load}} + d_j\}$$



Arc-Specific Resource Windows

Redefine the load-related parts of forward and backward REFs with the four arc-specific attributes $d_i^{ij}, d_j^{ij}, Q_i^{ij}$, and Q_j^{ij} in the following way:

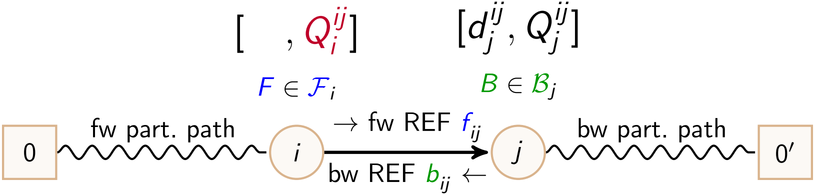
$$T_j^{\widetilde{load}} = f_{ij}^{\widetilde{load}}(T_i) = \max\{d_j^{ij}, T_i^{\widetilde{load}} + d_j\} \quad \text{feasible if} \quad T_j^{\widetilde{load}} \leq Q_j^{ij}$$



Redefine the load-related parts of forward and backward REFs with the four arc-specific attributes $d_i^{ij}, d_j^{ij}, Q_i^{ij}$, and Q_j^{ij} in the following way:

$$T_j^{\widetilde{load}} = f_{ij}^{\widetilde{load}}(T_i) = \max\{d_j^{ij}, T_i^{\widetilde{load}} + d_j\} \quad \text{feasible if} \quad T_j^{\widetilde{load}} \leq Q_j^{ij}$$

$$T_i^{\widetilde{load}} = b_{ij}^{\widetilde{load}}(T_j) = \min\{Q_i^{ij}, T_j^{\widetilde{load}} - d_j\}$$

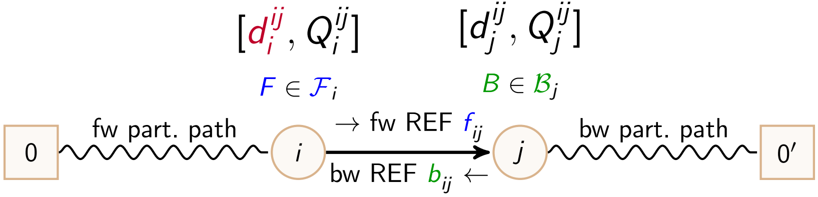


Arc-Specific Resource Windows

Redefine the load-related parts of forward and backward REFs with the four arc-specific attributes $d_i^{ij}, d_j^{ij}, Q_i^{ij}$, and Q_j^{ij} in the following way:

$$T_j^{l\tilde{o}ad} = f_{ij}^{l\tilde{o}ad}(T_i) = \max\{d_j^{ij}, T_i^{l\tilde{o}ad} + d_j\} \quad \text{feasible if} \quad T_j^{l\tilde{o}ad} \leq Q_j^{ij}$$

$$T_i^{l\tilde{o}ad} = b_{ij}^{l\tilde{o}ad}(T_j) = \min\{Q_i^{ij}, T_j^{l\tilde{o}ad} - d_j\} \quad \text{feasible if} \quad T_i^{l\tilde{o}ad} \geq d_i^{ij}$$

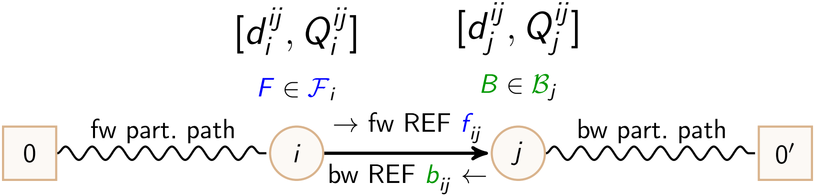


Arc-Specific Resource Windows

Redefine the load-related parts of forward and backward REFs with the four arc-specific attributes $d_i^{ij}, d_j^{ij}, Q_i^{ij}$, and Q_j^{ij} in the following way:

$$T_j^{l\widetilde{oa}d} = f_{ij}^{l\widetilde{oa}d}(T_i) = \max\{d_j^{ij}, T_i^{l\widetilde{oa}d} + d_j\} \quad \text{feasible if} \quad T_j^{l\widetilde{oa}d} \le Q_j^{ij}$$

$$T_i^{l\widetilde{oa}d} = b_{ij}^{l\widetilde{oa}d}(T_j) = \min\{Q_i^{ij}, T_j^{l\widetilde{oa}d} - d_j\} \quad \text{feasible if} \quad T_i^{l\widetilde{oa}d} \ge d_i^{ij}$$

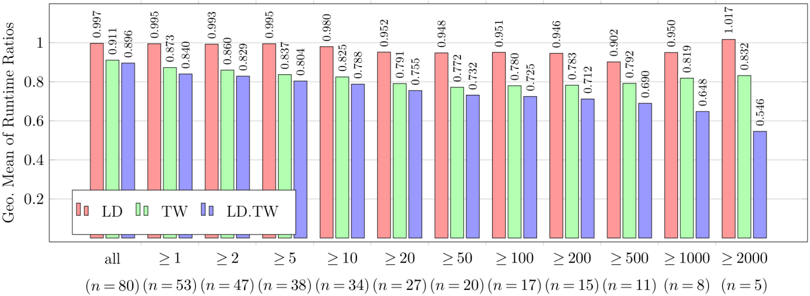


Initialization:

$$[d_i^{ij}, Q_i^{ij}] := [d_i, Q - d_j] \qquad \text{and} \qquad [d_j^{ij}, Q_j^{ij}] := [d_i + d_j, Q].$$

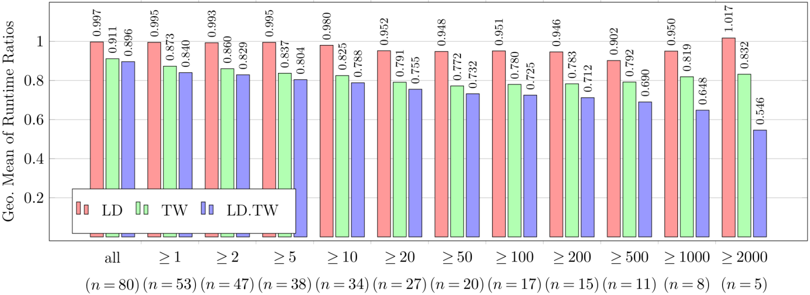
Arc-Specific Resource Windows

Results are grouped according to computation times, where $\geq rt$ indicates that only instances with a computation time t_{AF} of at least rt are considered.



Arc-Specific Resource Windows

Results are grouped according to computation times, where $\geq rt$ indicates that only instances with a computation time t_{AF} of at least rt are considered.

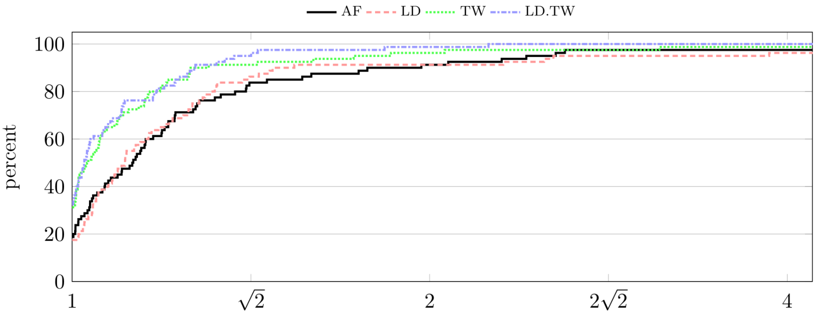


Remarks:

- Arc fixing (AF) is by-product of (i, j)-specific resource-window reduction: If the window of an arc $(i, j) \in A$ for some resource becomes an empty interval ($LHS > RHS$), the arc (i, j) is redundant (can be removed from the network).
- No explicit AF needed.

Arc-Specific Resource Windows

Performance profile:



Results for VRPTW and Simultaneous D&P

In the VRPTW with simultaneous deliveries and pickups, each customer $i \in N$ has to be visited exactly once. The service consists of

- a delivery of a non-negative quantity d_i
- a pickup of a non-negative quantity u_i.

Results for VRPTW and Simultaneous D&P

In the VRPTW with simultaneous deliveries and pickups, each customer $i \in N$ has to be visited exactly once. The service consists of

- a delivery of a non-negative quantity d_i
- a pickup of a non-negative quantity u_i.

For the resource *pick*, the resource windows can be initialized with the values

$$[q_i^{ij}, Q_i^{ij}] := [u_i, Q - u_j] \qquad \text{and} \qquad [q_j^{ij}, Q_j^{ij}] := [u_i + u_j, Q]$$

and for the resource *ml* with

$$[m_i^{ij}, M_i^{ij}] := [\max\{u_i, d_i\}, Q - d_j] \quad \text{and} \quad [m_j^{ij}, M_j^{ij}] := [\max\{d_i + d_j, u_i + d_j, u_i + u_j\}, Q].$$

Results for VRPTW and Simultaneous D&P

In the VRPTW with simultaneous deliveries and pickups, each customer $i \in N$ has to be visited exactly once. The service consists of

- a delivery of a non-negative quantity d_i
- a pickup of a non-negative quantity u_i.

For the resource *pick*, the resource windows can be initialized with the values

$$[q_i^{ij}, Q_i^{ij}] := [u_i, Q - u_j] \qquad \text{and} \qquad [q_j^{ij}, Q_j^{ij}] := [u_i + u_j, Q]$$

and for the resource *ml* with

$$[m_i^{ij}, M_i^{ij}] := [\max\{u_i, d_i\}, Q - d_j] \quad \text{and} \quad [m_j^{ij}, M_j^{ij}] := [\max\{d_i + d_j, u_i + d_j, u_i + u_j\}, Q].$$

The forward propagation of the resources *pick* and *ml* is performed with

$$\begin{aligned} T_j^{pick} &= f_{ij}^{pick}(T_i) = \max\{q_j^{ij}, T_i^{pick} + u_j \} && \text{feasible if} \quad T_j^{pick} \leq Q_j^{ij} \\ T_j^{ml} &= f_{ij}^{ml}(T_i) = \max\{m_j^{ij}, T_i^{pick} + u_j, T_i^{ml} + d_j\} && \text{feasible if} \quad T_j^{ml} \leq M_j^{ij}, \end{aligned}$$

while the backward propagation is performed with

$$\begin{aligned} T_i^{pick} &= b_{ij}^{pick}(T_j) = \min\{Q_i^{ij}, T_j^{pick} - u_j, T_j^{ml} - u_j\} && \text{feasible if} \quad T_i^{pick} \geq q_i^{ij} \\ T_i^{ml} &= b_{ij}^{ml}(T_j) = \min\{M_i^{ij}, T_j^{ml} - d_j\} && \text{feasible if} \quad T_i^{ml} \geq m_i^{ij}. \end{aligned}$$

Results for VRPTW and Simultaneous D&P

The following additional computational settings are analyzed:

- **AF/PI**: AF and resource-window reduction for the *pick*;
- **AF/ML**: AF and resource-window reduction for *ml*;
- **AF/All3**: AF and resource-window reduction for *pick*, *ml*, and *time*;
- **PI**: w/o AF, but with resource-window reduction for the *pick*;
- **ML**: w/o AF, but with resource-window reduction for *ml*;
- **All3**: w/o AF, but with resource-window reduction for *pick*, *ml*, and *time*;

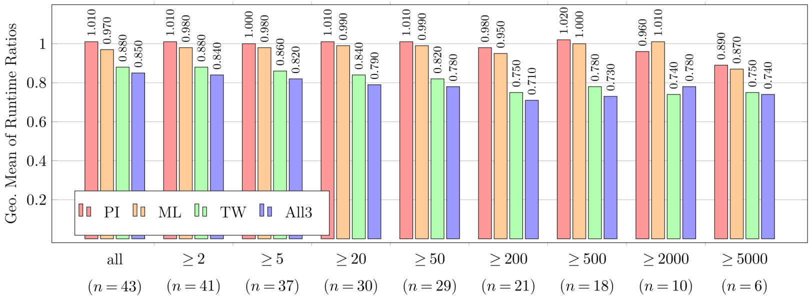
We use the VRPSDPTW instances of Hof and Schneider (2019): 43 instances in testset (not solved in root node; 12 instances with unknown objective value *opt*).

Results for VRPTW and Simultaneous D&P

		Computational Settings								
		AF	AF/PI	AF/ML	AF/TW	AF/All3	PI	ML	TW	All3
Geometric mean time ratios		1.00	0.95	1.00	0.98	1.00	1.74	1.67	1.67	1.54
#Opt		38	39	37	38	39	36	36	36	35
#Unsolved		5	4	6	5	4	7	7	7	8
arcs eliminated (%)	min.	46.50	46.76	46.50	46.50	46.63				
	avg.	87.89	88.01	87.96	87.80	87.78				
	max.	96.79	96.85	96.59	96.68	96.49				
pick-window reduction (%)	min.		0.03			0.02	0.02			0.02
	avg.		14.79			14.45	14.16			14.03
	max.		57.87			62.59	61.10			60.83
ml-window reduction (%)	min.			0.07		0.07		0.09		0.07
	avg.			16.70		16.29		15.84		15.94
	max.			66.48		66.83		65.03		66.52
time-window reduction (%)	min.				0.00	0.00			0.00	0.00
	avg.				10.72	10.80			10.49	9.89
	max.				32.51	32.65			29.63	29.37

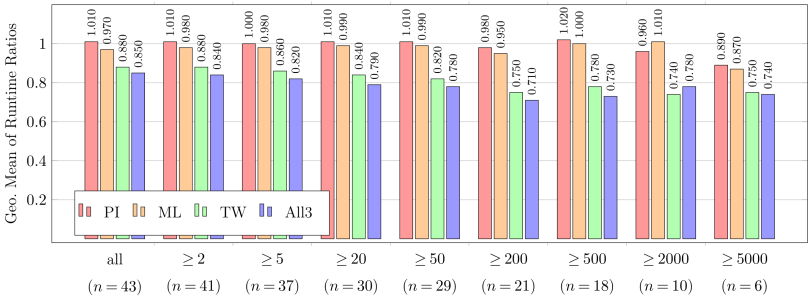
Arc-Specific Resource Windows

Results are grouped according to computation times, where $\geq rt$ indicates that only instances with a computation time t_{AF} of at least rt are considered.



Arc-Specific Resource Windows

Results are grouped according to computation times, where $\geq rt$ indicates that only instances with a computation time t_{AF} of at least rt are considered.



Remarks:

- Arc fixing (AF) is by-product of (i, j)-specific resource-window reduction: If the window of an arc $(i, j) \in A$ for some resource becomes an empty interval ($LHS > RHS$), the arc (i, j) is redundant (can be removed from the network).
- No explicit AF needed.

Findings:

- For resource-window reduction, forward and backward resource variables should provide lower and upper bounds, respectively.

Conclusions

Findings:

- For resource-window reduction, forward and backward resource variables should provide lower and upper bounds, respectively.
- Possible also for interdependent resource values (for simultaneous delivery and pickup, tour duration [=travel plus waiting time], etc.).

Conclusions

Findings:

- For resource-window reduction, forward and backward resource variables should provide lower and upper bounds, respectively.
- Possible also for interdependent resource values (for simultaneous delivery and pickup, tour duration [=travel plus waiting time], etc.).
- Arc-specific properties that simultaneously consider a given arc (i, j) and two resource windows at the endpoints i and j lead to more effective resource-window reduction procedures.

Thank you for listening!

Questions?!

References

Baldacci, R., Mingozzi, A., and Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, **59**(5), 1269–1283.

Desaulniers, G., Lessard, F., and Hadjar, A. (2008). Tabu search, partial elementarity, and generalized *k*-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, **42**(3), 387–404.

Desaulniers, G., Gschwind, T., and Irnich, S. (2018). Variable fixing based on two arc sequences in branch-price-and-cut algorithms. Presentation at 7th International Workshop on Freight Transportation and Logistics, ODYSSEUS 2018.

Desrochers, M., Desrosiers, J., and Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, **40**(2), 342–354.

Hof, J. and Schneider, M. (2019). An adaptive large neighborhood search with path relinking for a class of vehicle-routing problems with simultaneous pickup and delivery. *Networks*, **74**(3), 207–250.

Irnich, S., Desaulniers, G., Desrosiers, J., and Hadjar, A. (2010). Path-reduced costs for eliminating arcs in routing and scheduling. *INFORMS Journal on Computing*, **22**(2), 297–313.

Nemhauser, G. and Wolsey, L. (1988). *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc.

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.

References

Tilk, C., Rothenbächer, A.-K., Gschwind, T., and Irnich, S. (2017). Asymmetry matters: Dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster. *European Journal of Operational Research*, **261**(2), 530–539.

Villeneuve, D. and Desaulniers, G. (2005). The shortest path problem with forbidden paths. *European Journal of Operational Research*, **165**(1), 97–107.