The Bin Packing and Vector Packing Problems: Two more VrpSolver applications.

Artur Pessoa[1] Ruslan Sadykov[2] Eduardo Uchoa[1]
François Vanderbeck[3]

[1] Universidade Federal Fluminense, Brazil
[2] Inria Bordeaux, France
[3] Atoptima, France

Autumun School on Advanced BCP Tools
Paris, France, November 22th

The Problem

Input

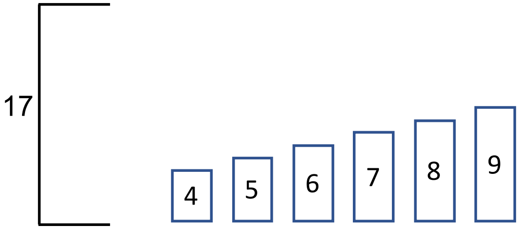
- Set T of items, set D of dimensions
- bin capacities Q^d ($d \in D$)
- item weights w_t^d ($t \in T, d \in D$)
- $D = \{1\}$ for Bin Packing
- $D = \{1, 2\}$ for (2D-Binary) Vector Packing

The Problem

Input

- Set T of items, set D of dimensions
- bin capacities Q^d ($d \in D$)
- item weights w_t^d ($t \in T, d \in D$)
- $D = \{1\}$ for Bin Packing
- $D = \{1, 2\}$ for (2D-Binary) Vector Packing

A Bin Packing Instance

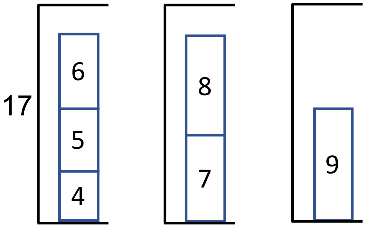


The Problem

Input

- Set T of items, set D of dimensions
- bin capacities Q^d $(d \in D)$
- item weights w_t^d $(t \in T, d \in D)$
- $D = \{1\}$ for Bin Packing
- $D = \{1, 2\}$ for (2D-Binary) Vector Packing

A Bin Packing Instance

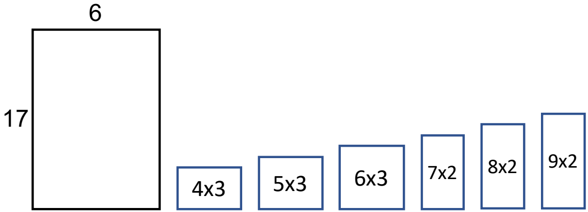


The Problem

Input

- Set T of items, set D of dimensions
- bin capacities Q^d ($d \in D$)
- item weights w_t^d ($t \in T, d \in D$)
- $D = \{1\}$ for Bin Packing
- $D = \{1, 2\}$ for (2D-Binary) Vector Packing

A Vector Packing Instance

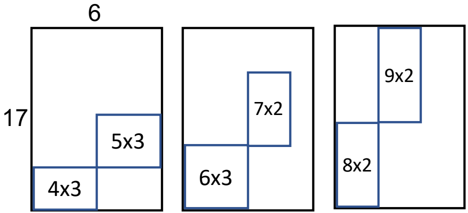


The Problem

Input

- Set T of items, set D of dimensions
- bin capacities Q^d $(d \in D)$
- item weights w_t^d $(t \in T, d \in D)$
- $D = \{1\}$ for Bin Packing
- $D = \{1, 2\}$ for (2D-Binary) Vector Packing

A Vector Packing Instance

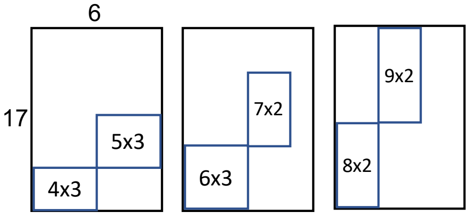


Decomposition

Input

- Set T of items, set D of dimensions
- bin capacities Q^d $(d \in D)$
- item weights w_t^d $(t \in T, d \in D)$
- $D = \{1\}$ for Bin Packing
- $D = \{1, 2\}$ for (2D-Binary) Vector Packing

A Vector Packing Instance



Formulation

h_t^p (constant): how many times item t is used in pattern $p \in P$

λ_p (variable): how many bins are filled with pattern p

$$\begin{aligned} \text{Min} \quad & \sum_{p \in P} \lambda_p && && \text{(1a)} \\ \text{S.t.} \quad & \sum_{p \in P} h_t^p \lambda_p \geq 1, && t = 1, \ldots, m, && \text{(1b)} \\ & \lambda_p \in \{0, 1\}, && p \in P. && \text{(1c)} \end{aligned}$$

P contains only feasible patterns
(p satisfying $\sum_{t \in T} w_t^d h_t^p \leq Q^d$, for all $d \in D$).

Relaxation

h_t^p (constant): how many times item t is used in pattern $p \in P$

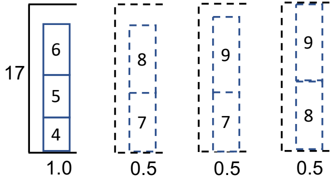
λ_p (variable): how many bins are filled with pattern p

$$\begin{aligned} \text{Min} \quad & \sum_{p \in P} \lambda_p && && \text{(2a)} \\ \text{S.t.} \quad & \sum_{p \in P} h_t^p \lambda_p \geq 1, && t = 1, \ldots, m, && \text{(2b)} \\ & \lambda_p \geq 0, && p \in P. && \text{(2c)} \end{aligned}$$

P contains only feasible patterns
(p satisfying $\sum_{t \in T} w_t^d h_t^p \leq Q^d$, for all $d \in D$).

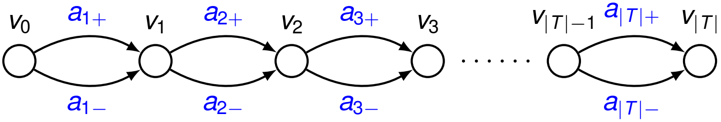
Relaxation: A Fractional Solution

$$\begin{aligned}
\text{Min} \quad & \lambda_{(4,5,6)} + \lambda_{(7,8)} + \lambda_{(7,9)} + \lambda_{(9,8)} + \cdots && \text{(3a)}\\
\text{S.t.} \quad & \lambda_{(4,5,6)} + \lambda_{(4)} + \cdots \geq 1, && \text{(3b)}\\
& \lambda_{(4,5,6)} + \lambda_{(5)} + \cdots \geq 1, && \text{(3c)}\\
& \lambda_{(4,5,6)} + \lambda_{(6)} + \cdots \geq 1, && \text{(3d)}\\
& \lambda_{(7,8)} + \lambda_{(7,9)} + \cdots \geq 1, && \text{(3e)}\\
& \lambda_{(7,8)} + \lambda_{(8,9)} + \cdots \geq 1, && \text{(3f)}\\
& \lambda_{(7,9)} + \lambda_{(8,9)} + \cdots \geq 1, && \text{(3g)}\\
& \lambda_p \geq 0, \qquad p \in P. && \text{(3h)}
\end{aligned}$$



The VrpSolver Model

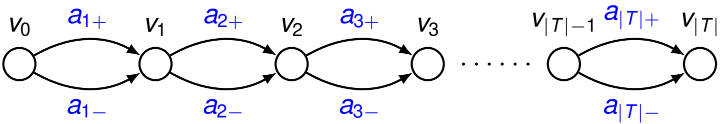
Graph G



- Resources $d \in D = R = R^M$ with consumption: $q_{a_{t+},d} = w_t^d$, $q_{a_{j-},d} = 0$, $t \in T, d \in D$
- Consumption bounds: $[l_{v_j,d}, u_{v_j,d}] = [0, Q^d]$, $t \in T$.
- Same as [Hessler et al., 2018] and [Wei et al., 2019].

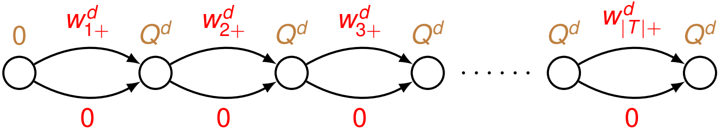
The VrpSolver Model

Graph G



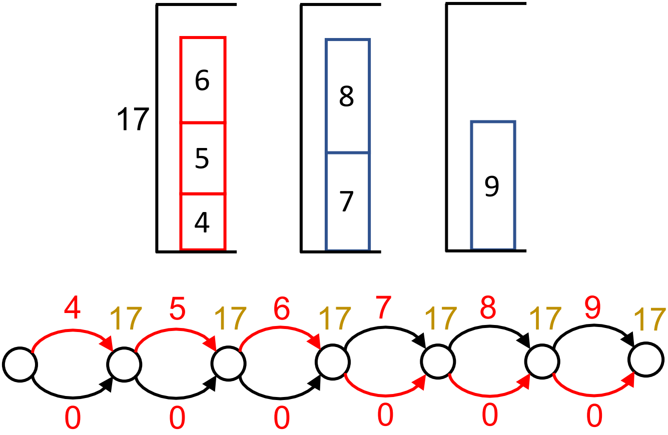
- Resources $d \in D = R = R^M$ with consumption: $q_{a_{t+},d} = w_t^d$, $q_{a_{j-},d} = 0$, $t \in T, d \in D$
- Consumption bounds: $[l_{v_j,d}, u_{v_j,d}] = [0, Q^d]$, $t \in T$.

RCSP Subproblem



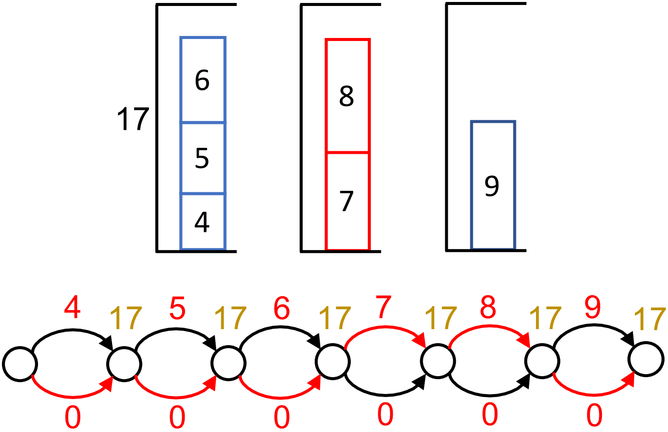
The VrpSolver Model

RCSP Subproblem



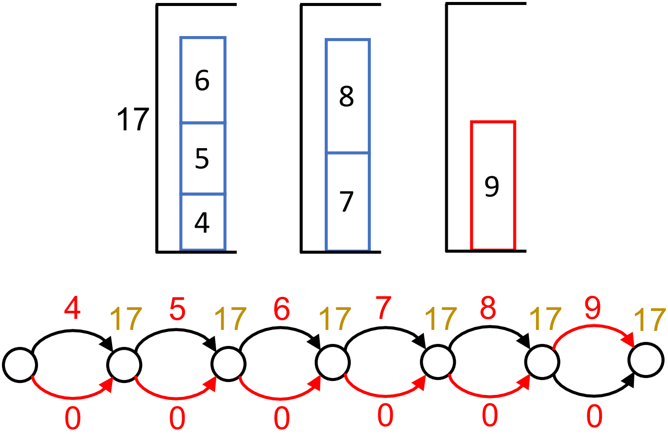
The VrpSolver Model

RCSP Subproblem



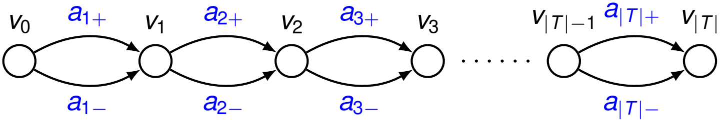
The VrpSolver Model

RCSP Subproblem



The VrpSolver Model

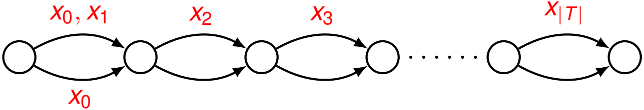
Graph G



- Mapping: $M(x_0) = \{a_{1+}, a_{1-}\}$, $M(x_t) = \{a_{t+}\}, t \in T$

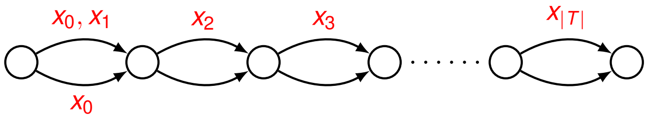
The VrpSolver Model

Arc mappings



The VrpSolver Model

Arc mappings



Formulation

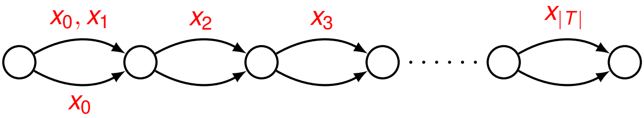
$$\begin{aligned} \text{Min} \quad & x_0 \\ \text{S.t.} \quad & x_t \geq 1, \qquad t \in T; \end{aligned}$$

Additional Elements

- Subproblem cardinality: $L = 0$, $U = \infty$
- Packing sets: $\mathcal{B} = \cup_{t \in T}\{\{a_{t+}\}\}$
- Branching over accumulated resource consumption and, if still needed, by Ryan and Foster rule
- Enumeration is on

The VrpSolver Model

Arc mappings



Formulation

$$\begin{aligned} \text{Min} \quad & x_0 \\ \text{S.t.} \quad & x_t \geq 1, \qquad t \in T; \end{aligned}$$

Additional Elements

- Subproblem cardinality: $L = 0$, $U = \infty$
- Packing sets: $\mathcal{B} = \cup_{t \in T}\{\{a_{t+}\}\}$
- Branching over accumulated resource consumption and, if still needed, by Ryan and Foster rule (never needed)
- Enumeration is on

Advanced Techniques borrowed from Vehicle Routing

- Bucket graph-based labelling algorithm for the RCSP pricing [Righini and Salani, 2006] [Sadykov et al., 2017]
- Automatic dual price smoothing stabilization [Wentges, 1997] [Pessoa et al., 2017]
- Reduced cost fixing of (bucket) arcs in the pricing problem [Ibaraki and Nakamura, 1994] [Irnich et al., 2010] [Sadykov et al., 2017]
- Limited-Memory Rank-1 Cuts [Jepsen et al., 2008] [Pecin et al., 2017b] [Pecin et al., 2017c] [Pecin et al., 2017a]
- Enumeration of elementary routes [Baldacci et al., 2008]
- Multi-phase strong branching [Pecin et al., 2017b]
- Generic (strong) diving heuristic [Sadykov et al., 2018]

Bin Packing: computational results

- Comparison with the best BCP [Wei et al., 2019] on AI and ANI instances by [Delorme et al., 2016].
- All instances by [Falkenauer, 1996] (up to 501 items) and by [Schoenfield, 2002] (up to 200 items, Hard28) solved in up to 3m37s (16s in the average) but pseudopolynomial formulations lead to better results.
- Initial upper bound is rounded-up lower bound plus one (easy for heuristics).
- Diving heuristic not enabled for ANI instances.

Bin Packing: computational results

Instance class	$\lvert J \rvert$	Best BCP		Our BCP	
		N	T	N	T
ANI200	201	50/50	14s	50/50	17s
ANI400	402	47/50	>7m16s	50/50	1m36s
ANI600	600	0/50	>1h	3/50	>58m
ANI800	801	0/50	>1h	0/50	>1h
AI200	202	50/50	4s	50/50	52s
AI400	403	46/50	>6m	46/50	>8m11s
AI600	601	27/50	>29m	35/50	>24m
AI800	802	15/50	>46m	26/50	>46m

- Best BCP: [Wei, Luo, Baldacci, and Lim, 2019]

Vertex Packing: computational results

Comparison with the state-of-the-art on the harderst 2-resources 200-items instances by [Caprara and Toth, 2001]

Algorithm	Class 1		Class 4	
	N	T	N	T
[Brandão and Pedroso, 2016]	10/10	2h07m	0/10	>2h
[Hu et al., 2017]	0/10	>10m	0/10	>10m
[Hessler et al., 2018]	3/10	>47m	0/10	>1h
Our BCP	10/10	2m42s	10/10	2m33s
	Class 5		Class 9	
[Brandão and Pedroso, 2016]	0/10	>2h	0/10	>2h
[Hu et al., 2017]	7/10	>6m	0/10	>10m
[Hessler et al., 2018]	7/10	>41m	0/10	>1h
Our BCP	10/10	12m10s	8/10	>27m

New Branching Scheme for Bin Packing

VrpSolver Model

- Branching over accumulated resource consumption and, if still needed, by Ryan and Foster rule

New Branching Scheme for Bin Packing

VrpSolver Model

- Branching over accumulated (disposable) resource consumption and, if still needed, by Ryan and Foster rule (never needed)

New Branching Scheme for Bin Packing

VrpSolver Model

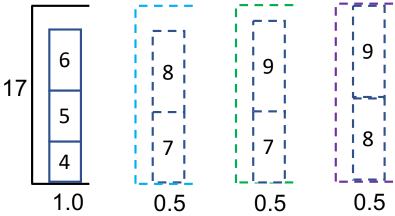
- Branching over accumulated (disposable) resource consumption and, if still needed, by Ryan and Foster rule (never needed)
- Advantages: keeps the princing structure (robust), allows stronger dominance rule.
- Disadvantage: subproblem solutions with positive capacity slacks may be feasible in both branches.

New Branching Scheme for Bin Packing

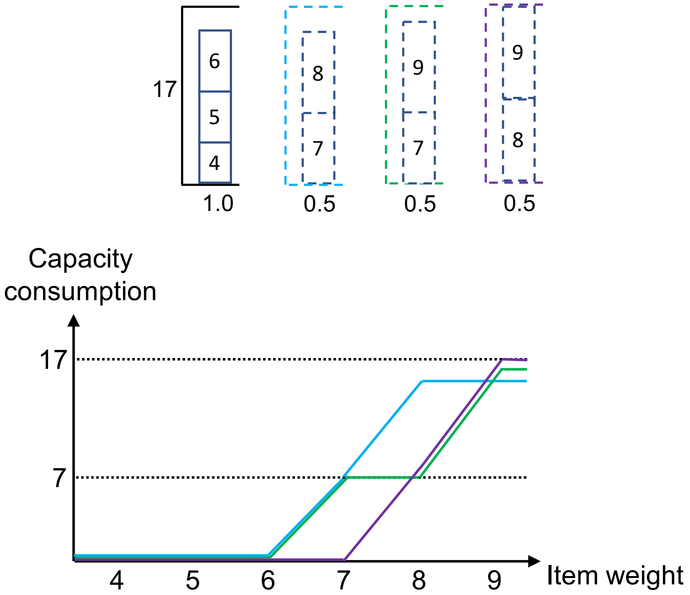
VrpSolver Model

- Branching over accumulated (disposable) resource consumption and, if still needed, by Ryan and Foster rule (never needed)
- Advantages: keeps the princing structure (robust), allows stronger dominance rule.
- Disadvantage: subproblem solutions with positive capacity slacks may be feasible in both branches.

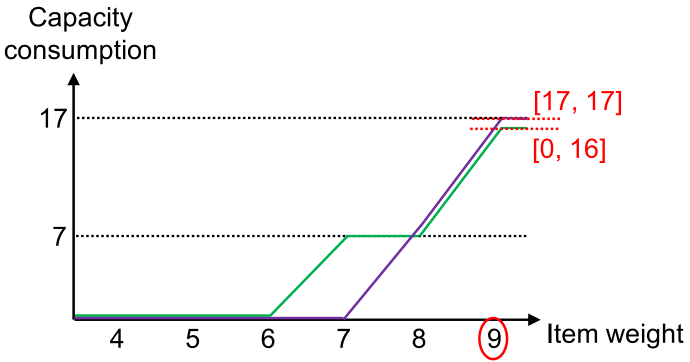
Example



New Branching Scheme for Bin Packing

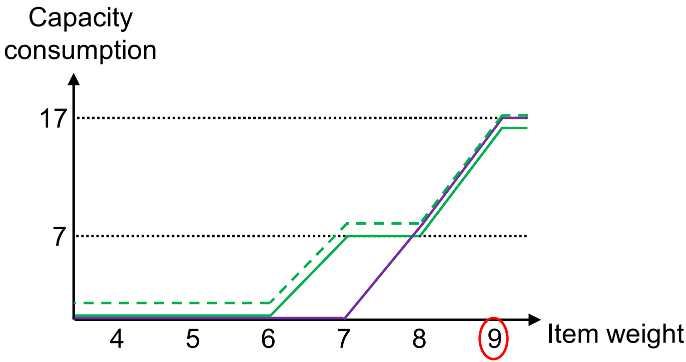


New Branching Scheme for Bin Packing



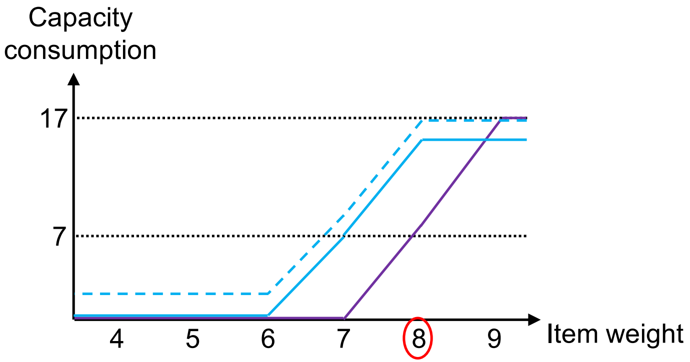
- Resource consumption bounds for branching on item 9 are $[0, 16]$ and $[17, 17]$
- Both paths can meet bounds $[17, 17]$ (disposable resources)

New Branching Scheme for Bin Packing



- Must consider both minimum and maximum resource consumptions
- No effective branching is possible for item 9

New Branching Scheme for Bin Packing



- Item 8 admits an effective branching

New Branching Scheme for Bin Packing

The scheme

- In an effective branching, for each branch, there is at least one non-zero λ variable whose path becomes infeasible.
- Some λ variables may still have paths feasible for both branches.
- Previously proposed in [Gélinas et al., 1995] for vehicle routing.

New Branching Scheme for Bin Packing

The scheme

- In an effective branching, for each branch, there is at least one non-zero λ variable whose path becomes infeasible.
- Some λ variables may still have paths feasible for both branches.
- Proposed in [Gélinas et al., 1995] for vehicle routing.

Theorem

- For Bin Packing, any fractional solution with no effective branching has the same projection as some convex combination of integer solutions in the arcs space.

New Branching Scheme for Bin Packing

Theorem

- Any fractional solution with no effective branching has the same projection as some convex combination of integer solutions in the arcs space.

Sketch of the proof

Step 1 Any fractional solution obtained with tight (disposable) resource consumption bounds has the same projection as some convex combination of integer solutions in the arcs space.

- The master problem relaxation is equivalent to a minimum cost flow problem.

New Branching Scheme for Bin Packing

Theorem

- Any fractional solution with no effective branching has the same projection as some convex combination of integer solutions in the arcs space.

Sketch of the proof

Step 2 Any fractional solution with no effective branching is feasible for some set of tight resource consumption bounds (and the theorem follows from Step 1).

- No effective branching $\Rightarrow$ for every item t, and consumption threshold q, every path is feasible either under consumption bound $\leq q - 1$ or $\geq q$.
- Let q^* be the maximum threshold the makes all paths feasible for consumption bound $\geq q^*$.
- For threshold $q^* + 1$ all paths are feasible for consumption bound $\leq q^*$.
- Thus, all paths are feasible for the tight consumption range $[q^*, q^*]$.

Conclusions and Future Work

Conclusions

- VrpSolver is an useful tool for testing (existing) advanced BCP techniques on new problems
- For Bin Packing, it is competitive, and for Vector Packing, superior to the state-of-the art.
- Such tests may inspire interesting new investigations (e.g. the new branching scheme)

Possible Extensions to the New Branching Scheme

- Bin Packing with Multiple Size Bins
- Cutting Stock
- Generic VrpSolver Model (multiple resources)

THANKS!

References I

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.

Brandão, F. and Pedroso, J. a. P. (2016).
Bin packing and related problems: General arc-flow formulation with graph compression.
Computers & Operations Research, 69:56 – 67.

Caprara, A. and Toth, P. (2001).
Lower bounds and algorithms for the 2-dimensional vector packing problem.
Discrete Applied Mathematics, 111(3):231 – 262.

Delorme, M., Iori, M., and Martello, S. (2016).
Bin packing and cutting stock problems: Mathematical models and exact algorithms.
European Journal of Operational Research, 255(1):1–20.

References II

Falkenauer, E. (1996).
A hybrid grouping genetic algorithm for bin packing.
Journal of Heuristics, 2(1):5–30.

Gélinas, S., Desrochers, M., Desrosiers, J., and Solomon, M. M. (1995).
A new branching strategy for time constrained routing problems with application to backhauling.
Annals of Operations Research, 61(1):91–109.

Hessler, K., Gschwind, T., and Irnich, S. (2018).
Stabilized branch-and-price algorithms for vector packing problems.
European Journal of Operational Research, 271(2):401 – 419.

Hu, Q., Zhu, W., Qin, H., and Lim, A. (2017).
A branch-and-price algorithm for the two-dimensional vector packing problem with piecewise linear cost function.
European Journal of Operational Research, 260(1):70 – 80.

References III

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

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.

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.

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

References IV

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

Pecin, D., Pessoa, A., Poggi, M., Uchoa, E., and Santos, H. (2017c).
Limited memory rank-1 cuts for vehicle routing problems.
Operations Research Letters, 45(3):206 – 209.

Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2017).
Automation and combination of linear-programming based stabilization techniques in column generation.
Technical Report hal-01077984, HAL Inria.

Righini, G. and Salani, M. (2006).
Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints.
Discrete Optimization, 3(3):255 – 273.

References V

Sadykov, R., Uchoa, E., and Pessoa, A. (2017).
A bucket graph based labeling algorithm with application to vehicle routing.
Cadernos do LOGIS 7, Universidade Federal Fluminense.

Sadykov, R., Vanderbeck, F., Pessoa, A., Tahiri, I., and Uchoa, E. (2018).
Primal heuristics for branch-and-price: the assets of diving methods.
INFORMS Journal on Computing, (Forthcoming).

Schoenfield, J. (2002).
Fast, exact solution of open bin packing problems without linear programming.
Technical report, US Army Space and Missile Defense Command.

References VI

Wei, L., Luo, Z., Baldacci, R., and Lim, A. (2019).
A new branch-and-price-and-cut algorithm for one-dimensional bin-packing problems.
INFORMS Journal on Computing.
Online.

Wentges, P. (1997).
Weighted dantzig–wolfe decomposition for linear mixed-integer programming.
International Transactions in Operational Research, 4(2):151–162.