Overcoming Vanilla/Textbook CG

OeGOR Summer-School 2024, Krems

Timo Gschwind
Lehrstuhl für BWL, insb. Logistik

July 22 – 26, 2024

Agenda

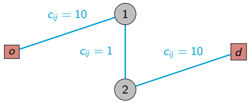
What is the right reformulation for your problem?

- What theory suggests:
  - → coupling constraints in the master
  - → one type of variable/subproblem for each block (*block-diagonal structure*)
  - → aggregation if possible
- Practice:
  - → choose reformulation such that overall approach performs best
    - > tradeoff: strength of the formulation vs. (practical) hardness of the subproblem
  - → rule of thumb:
    - > one type of variable/subproblem for each block
    - > aggregation if possible
    - > all constraints in the subproblem such that it remains *'well'* solvable → master needs to ensure remaining constraints (e.g., via constraints, cuts, branching)

D-W Reformulations

Example 1: VRPTW [Baldacci et al., 2011, Contardo et al., 2015, Pecin et al., 2017]

- Theory:
  - → master: partitioning constraints; fleet size constraint
  - → subproblem: elementary shortest path problem with resource constraints
- Practice:
  - → relax elementarity constraint in subproblem (→ *ng*-routes)
    - > subproblem much easier to solve
    - > formulation is weaker; example:



path $r_1 := o - 1 - 2 - 1 - 2 - d$,
cost $c_1 = 23$, RMP sol. $\lambda_1 = 1/2$

path $r_2 := o - 1 - 2 - d$,
cost $c_2 = 21$, RMP sol. $\lambda_2 = 1$

  - → elementarity automatically fulfilled for integer solutions
  - → Note: literature played a lot with this tradeoff

Example 2: VRP variant with some (for the subproblem) super complicated intra-route constraint: [Cherkesly et al., 2015, Gschwind, 2015]

- Relax this constraint in the subproblem
- Infeasible path elimination cuts in the master to forbid infeasible routes

Agenda

RMP Initialization

Why Provide a feasible RMP solution (feasible basic solution)
→ '*simplex*-feasibility'; not feasibility w.r.t. actual problem

How
1. (One or more) heuristic solutions
2. Any collection of columns
3. big-M: artificial columns with large costs
   - > VRPTW: cover a single customer (diff. to dedicated route?)

When At initialization of root node and potentially after each addition of cuts or branching decisions

Remarks:

1 + 2
- \+ may save many/some/few iterations
- \+ may provide global UB
- − may be difficult to guarantee RMP feasibility (in particular after cutting and branching)
- − columns/solutions for integer MP may not be useful for LP-relaxation

3
- \+ straightforward to implement
- \+ may be necessary anyway (after cutting and branching)

RMP Initialization

Some personal experience (with possible explanations):

- Typically: you don't win or loose here
  - $\rightarrow$ do not prioritize
- For VRPs: just stick with big-M, other strategies not worth it
  - $\rightarrow$ labeling heuristics for ESPPRC subproblem very effective
  - $\rightarrow$ the important iterations (exact pricings) are not saved
  - $\rightarrow$ maybe I am just doing it wrong. . .
- For packing (and related) problems: noticeable speedups with initialization by heuristic solutions
  - $\rightarrow$ simple heuristics provide good solutions
  - $\rightarrow$ root RMP solution *'close'* to integer optimal one

Heuristics are your friend!! [almost any paper on a good branch-and-price algorithm]

- No need to find the minimum reduced-cost column!
  - $\rightarrow$ any negative rdc column keeps the CG process going
- Maybe the most important component for many problems

Plenty of possibilities:

- Check columns of a column pool
- (Randomized) greedy/construction heuristics
- Local search
  - $\rightarrow$ start from basic variables $\rightarrow$ reduced cost of 0
- Metaheuristics
- Use exact pricer in heuristic fashion
  - $\rightarrow$ on reduced problem (e.g., on reduced graph)
  - $\rightarrow$ relax dominance conditions in DP
  - $\rightarrow$ keep only *'very'* promising states in DP/nodes in combinatorial B&B
  - $\rightarrow$ stop prematurely if negative rdc column found
  - $\rightarrow$ ...

Pricing Strategies

Overall pricing strategy:

- Tradeoff: speed of the pricing vs. quality and number of columns generated
  - $\rightarrow$ the ideal pricer is *'super fast'* and gives *'many' 'relevant'* columns
  - $\rightarrow$ also depends on how difficult the master is
- Build hierarchies of pricers
  - $\rightarrow$ increasing effort $\rightarrow$ exact pricer at the end
  - $\rightarrow$ ideally: exact pricer just called once to verify that no negative rdc column exists
- Generate multiple columns per iteration
  - $\rightarrow$ prefer pricing algorithms that give many columns
  - $\rightarrow$ stop pricing prematurely when reaching threshold of columns
  - $\rightarrow$ (try to look for) diverse columns
- Several subproblems:
  - $\rightarrow$ do not solve every one in each iteration
  - $\rightarrow$ test promising columns also for other subproblems

Agenda

Solving the Integer Problem

Identifying integer solutions: [Jans, 2010]

- If λ is integer, then x is integer
- If λ is fractional, x may still be integer!
- Beware (aggregation): x integer does not (always) imply x^k integer!

Obtaining optimal integer solutions: [Baldacci et al., 2011, Jepsen et al., 2008]

- Adding valid inequalities probably helps
- Branch-and-price and branch-price-and-cut
- Enumeration
  1. enumerate and add to RMP all columns that may be part of an optimal integer solution ($\rightarrow$ see *Variable Fixing*)
  2. solve RMP to integer optimality
- Combine enumeration and branch-and-price
- Other strategies?!

Branching Tree

Number of nodes that can be explored is typically limited: [Pecin et al., 2017]

- Try to avoid branching at all $\rightarrow$ strong valid inequalities, enumeration
- Build hierarchies of branching rules:
  - $\rightarrow$ add unnecessary rules (with a big impact, e.g., # vehicles in VRPs)
  - $\rightarrow$ most influential ones first
- More refined branching strategies:
  - $\rightarrow$ strong branching
    - $>$ me: not necessary to solve candidates completely (no CG)
  - $\rightarrow$ pseudo cost branching
  - $\rightarrow$ hybrid strategies
- Tree search strategy:
  - $\rightarrow$ best first: if lower bound needs to be raised (e.g., VRPs)
  - $\rightarrow$ depth first: 'just' need to find optimal integer solution (e.g., CS/BP)
- Early branching: branch before nodes are fully solved
  - $\rightarrow$ save expensive exact pricing iterations, avoid slow convergence
  - $\rightarrow$ possibly no valid bounds for the nodes

Lower Bounds

Let

- $z^\star_{MP}$ and $z^\star_{RMP}$ be the optimal solution values of the MP and RMP
- z^t_{RMP} be the optimal RMP solution value at CG iteration t and π^t the corresponding optimal dual solution
- K be the set of subproblems/types of variables
- κ^k be an upper bound on the total solution value of all variables of $k \in K$, i.e., $\sum \lambda^k \leq \kappa^k$
- $\tilde{c}^{k\star}(\pi)$ be the minimum reduced cost for dual values π

  $\rightarrow$ not available in case of heuristic pricing

The following holds true (assuming minimization)

- $z^{t-1}_{RMP} \geq z^t_{RMP} \geq z^\star_{RMP} = z^\star_{MP}$
- $z^t_{RMP} + \sum_{k \in K} \kappa^k \tilde{c}^\star(\pi^t) \leq z^\star_{MP}$
- if $|K| = 1$ and $c_\lambda = 1, \forall \lambda$: $\quad \frac{z^t_{RMP}}{1 - \tilde{c}^\star(\pi^t)} \leq z^\star_{MP}$

CG-based Heuristics

[Sadykov et al., 2019, Wahlen and Gschwind, 2023]

Why

1. You also/just want a heuristic
   - may provide very(!) good solutions/be competitive in general
   - good upper bounds may help the overall approach
   - can barely solve the root node in reasonable time
   - some are straightforward to implement
2. You do not want to implement a full branch-and-price
   - fair enough ($\rightarrow$ maybe check on generic frameworks)

How

1. Exact CG, no (complete) branching
   - restricted master heuristic: solve current RMP as MIP
     super simple, very good solutions for some problems
   - tree search facilitating integer solution ($+$ hard time limit), diving heuristics
2. Full branching, no exact CG
   - only use heuristic pricers
3. Any combination or more involved approaches

Agenda

Reduced Cost-based Variable Elimination/Fixing

[Bianchessi et al., 2024]

Nemhauser and Wolsey [1988], Proposition 2.1, page 389

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

Remarks:

- Criterion used for Enumeration ($\rightarrow$ see *Solving the Integer Problem*)
- (Practically) not directly applicable to CG for variable elimination
  - $\rightarrow$ Subproblem must prevent the re-generation of these variables
  - $\rightarrow$ Subproblem structure changes, gets much more complicated
- Requires a (good) upper bound
  - $\rightarrow$ guessing is possible, if none is available

Reduced Cost-based Variable Elimination/Fixing

Variable elimination in column generation/branch-and-price:

Proposition

Let

- $P[prop]$ be the subset of all variables P that satisfy some property *prop*
- and $\tilde{c}[prop](\pi) = \min_{p \in P[prop]} \tilde{c}_p(\pi)$ be the minimum reduced cost of any variable satisfying property *prop*.

If $\tilde{c}[prop](\pi) > UB - LB(\pi)$, then all λ_p, $p \in P[prop]$ can be eliminated.

Meaningful properties must meet the following criteria:

1. Elimination accelerates the overall algorithm

   $\rightarrow$ go well with/simplify the pricing algorithms

2. Minimum rdc $\tilde{c}[prop](\pi)$ can be effectively computed

Example: VRPTW

The following properties have been proposed:

1. Single arcs:
   - → for an arc $(i,j) \in A$, the route includes the arc (i,j) at least once
   - → eliminate arc from network; no modification of labeling algorithm
2. Two-arc sequences:
   - → for two arcs $(h,i),(i,j) \in A$, the route includes the sequence (h,i,j) at least once
   - → labeling algorithm needs to be adapted (more complicated); outweighed by much smaller number of possible routes
3. (Arc-specific) resource windows:
   - → the route visits a customer i with a certain resource value
     - > example (time windows): the route visits a customer i before (after/exactly at) some time t
   - → use adapted resource windows; no modification of labeling algorithm

Remark: In all cases, the computation of $\tilde{c}[prop](\pi)$ is (almost) a by-product of the bidirectional labeling algorithm for solving the subproblem.

Stabilization

[Pessoa et al., 2018, Gschwind and Irnich, 2016]

Why → CG process is driven by the dual values

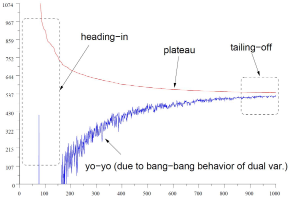
→ Several instability issues

> heading-in effect

> plateau effect

> tailing-off effect

> oscillation of dual values and dual bound



[Vanderbeck, 2005]

Idea Try to guide the duals to their optimal value faster

How → Very simple:

> replace partitioning by covering constraints if possible

→ Other methods:

> interior point stabilization (e.g., solve RMP with interior point method, convex combine several simplex solutions)

> box step / penalty function methods

> dual variable smoothing techniques

> dual-optimal inequalities

Stabilization – Set Covering vs. Set Partitioning

Set covering instead of set partitioning:

$$z_{MP}^{SP} = \min c^\top \lambda \qquad A\lambda = b \quad [\pi] \qquad \lambda \geq 0 \qquad \Rightarrow \pi \in \mathbb{R}$$

$$z_{MP}^{SP} = \min c^\top \lambda \qquad A\lambda \geq b \quad [\pi] \qquad \lambda \geq 0 \qquad \Rightarrow \pi \geq 0$$

- Effect: cut the dual space in half!
- Examples: VRPTW, bin packing, cutting stock, vertex coloring, . . .
- Prerequisites: heredity [Gschwind et al., 2021]
  - → subsets of feasible structures are again feasible structures (with the same or less cost)
  - → removing
    - > a customer from a feasible route,
    - > an item from a feasible bin or cutting pattern, or
    - > a vertex from an independent set

    results again in a feasible route, feasible bin or cutting pattern, or independent set (with identical or better cost)
  - → counterexamples: VRPTW minimizing waiting times or without triangle inequality, graph partitioning with (certain) relaxed cliques

Box step / penalty function methods:

Original primal:

$$\begin{array}{lrcl} \min & c^\top \lambda & & \\ & A\lambda & = & b \\ & \lambda & \geq & 0 \end{array}$$

⇒ relaxation of original primal

Original dual:

$$\begin{array}{lrcl} \max & b^\top \pi & & \\ & A^\top \pi & \leq & c \end{array}$$

⇒ restriction of original dual

- Basic idea: keep dual variables *close* to a *stability center*
- Dual variables π are restricted to the interval $[\delta^- - w^-, \delta^+ + w^+]$
- Deviation of π from the interval $[\delta^-, \delta^+]$ is penalized by ϵ^-, ϵ^+
- Surplus and slack variables y^- and y^+ perturb b by $\epsilon \in [\epsilon^-, \epsilon^+]$
- Need to tune parameters δ^-, δ^+, ϵ^-, ϵ^+
- Can be generalized to piecewise linear penalty functions

Dual variable smoothing techniques:

- Basic idea: smooth dual solution π^t of the current CG iteration by *previous* dual solutions
- Neame [2000]: $\bar{\pi}^t = \alpha\bar{\pi}^{t-1} + (1-\alpha)\pi^t$
- Wentges [1997]: $\bar{\pi}^t = \alpha\hat{\pi} + (1-\alpha)\pi^t$
  - $\rightarrow$ $\hat{\pi}$ is the current best known vector of duals
- Solve subproblem with smoothed dual prices $\bar{\pi}^t$ instead of π^t
  - $\rightarrow$ Columns are only added if they have negative rdc w.r.t. π^t
  - $\rightarrow$ otherwise: mispricing
- Level of smoothing is parametrized by single parameter $\alpha \in [0, 1)$

Stabilization – Dual-Optimal Inequalities

Dual-Optimal Inequalities:

Original primal model

$$\begin{aligned} z(P) &= \min c^\top \lambda \\ \text{s.t.} \quad & A\lambda = b \\ & \lambda \geq 0 \end{aligned}$$

Set of optimal solutions: P^*

Original dual model

$$\begin{aligned} z(D) &= \max b^\top \pi \\ \text{s.t.} \quad & A^\top \pi \leq c \end{aligned}$$

Set of optimal solutions: D^*

Basic idea: Restrict the dual space by additional constraints $E^\top \pi \leq e$.

Extended primal model

$$\begin{aligned} z(\tilde{P}) &= \min c^\top \lambda + e^\top y \\ \text{s.t.} \quad & A\lambda + Ey = b \\ & \lambda \geq 0, y \geq 0 \end{aligned}$$

Set of optimal solutions: $\tilde{P}^*$

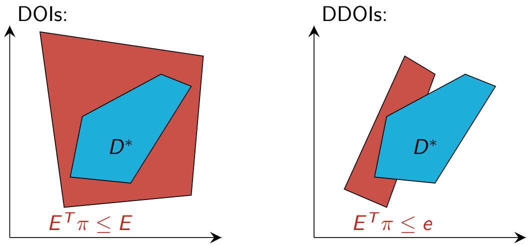
Extended dual model

$$\begin{aligned} z(\tilde{D}) &= \max b^\top \pi \\ \text{s.t.} \quad & A^\top \pi \leq c \\ & E^\top \pi \leq e \end{aligned}$$

Set of optimal solutions: $\tilde{D}^*$

Definition [Ben Amor et al., 2006]

If $D^* \subseteq \{\pi : E^\top \pi \leq e\}$ then $E^\top \pi \leq e$ is a (set of) dual-optimal inequalities (DOIs).
If $D^* \cap \{\pi : E^\top \pi \leq e\} \neq \varnothing$ then $E^\top \pi \leq e$ is a set of deep dual-optimal inequalities (DDOIs).



$\Rightarrow$ If the additional constraints $E^\top \pi \leq e$ are DOIs or DDOIs, then the original and extended models are equivalent!

DOIs for Cutting Stock (CS) and Bin Packing (BP) Problems

Gilmore and Gomory [1961]:

$$
\begin{aligned}
z(P_{CS}) \quad = \quad & \min \sum_{p \in \Omega} \lambda_p \\
\text{s.t.} \quad & \sum_{p \in \Omega} a_{ip} \lambda_p \geq b_i, \quad i \in I \\
& \lambda_p \geq 0, p \in \Omega
\end{aligned}
$$

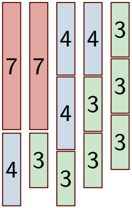
Dual model:

$$
\begin{aligned}
z(D_{CS}) \quad = \quad & \max \sum_{i \in I} b_i \pi_i \\
\text{s.t.} \quad & \sum_{i \in I} a_{ip} \pi_i \leq 1, \quad p \in \Omega \\
& \pi_i \geq 0, \quad i \in I
\end{aligned}
$$

Note: Integer demand and pattern coefficients b_i and a_{ip} for Cutting Stock
Binary demand and pattern coefficients b_i and a_{ip} for Bin Packing

Example (CS): Length $L = 11$, $I = \{1, 2, 3\}$ with $w_1 = 7, w_2 = 4, w_3 = 3$.

$$
\begin{aligned}
z(P_{CS}) \quad = \quad & \min \mathbf{1}^\top \lambda \\
\text{s.t.} \quad & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 1 & 0 \\ 0 & 1 & 1 & 2 & 3 \end{bmatrix} \lambda \geq b \\
& \lambda \in \mathbb{R}^5_+
\end{aligned}
$$



Stabilization – Dual-Optimal Inequalities

DOIs for Cutting Stock (CS) and Bin Packing (BP) Problems

Ranking Inequalities [Gilmore and Gomory, 1961]

Assume $I = \{1, \ldots, m\}$ with $w_1 \geq w_2 \geq \cdots \geq w_m$.
The Ranking Inequalities (RI) $\pi_i \geq \pi_{i+1}, 1 \leq i < m$ are DOIs for CS and DDOIs for BP.
The Equality Constraints $\pi_i = \pi_j$ if $w_i = w_j$ are DOIs for CS and DDOIs for BP.

Subset Inequalities [Valério de Carvalho, 2005, Gschwind and Irnich, 2016]

Let $i \in I$, $S \subseteq I \setminus \{i\}$, and $(t)_s \in \mathbb{Z}_+^S$ with $w_i \geq \sum_{s \in S} t_s w_s$.
The Weighted Subset Inequalities (WSI) $\pi_i \geq \sum_{s \in S} t_s \pi_s$ are DOIs for CS and generally <u>not</u> DDOIs for BP.

Remarks:

- Intuition: larger items are more difficult to cover/to include in a pattern
  - $\rightarrow$ they should have a higher dual price
- Corresponding primal columns (=DOI columns)?
  - $\rightarrow$ cost of zero, coefficients in covering constraints: -1 for item i and $+1$ for item $i+1$ (RI) or $+t_s$ for items $s \in S$ (WSI)
  - $\rightarrow$ allow (cost neutral) replacement of an item by others in a pattern
  - $\rightarrow$ allow implicit representation of pattern variables in the RMP

Stabilization – Dual-Optimal Inequalities

DOIs for Cutting Stock (CS) and Bin Packing (BP) Problems

Example (cont'd): Length $L = 11$, $I = \{1, 2, 3\}$ with $w_1 = 7, w_2 = 4, w_3 = 3$.

$$z(\tilde{P}_{CS}) = \min \mathbf{1}^\top \lambda + \mathbf{0}^\top y$$
$$\text{s.t.} \quad \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 1 & 0 \\ 0 & 1 & 1 & 2 & 3 \end{bmatrix} \lambda + \begin{bmatrix} -1 & 0 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} y \geq b$$
$$\lambda \in \mathbb{R}^5_+, \; y \in \mathbb{R}^2_+$$

and the corresponding (undominated) DOIs are

$$\pi_1 \geq \pi_2 + \pi_3 \qquad \text{and} \qquad \pi_2 \geq \pi_3$$

Remarks:

- Pattern column $(1, 1, 0)^\top$ and DOI columns $(-1, 1, 1)^\top$ and $(0, -1, 1)^\top$ are sufficient to (implicitly) represent all undominated patterns
- Recall for BP: WSI are not DDOIs
  - → combining pattern column $(1, 1, 0)^\top$ and DOI column $(-1, 1, 1)^\top$ results in infeasible (non-binary!) pattern $(0, 2, 1)^\top$
  - → (one) proof-scheme for DDOI-property: show that it is always possible to transform a solution with DOI columns into one with the same cost and without DOI columns

Stabilization – Dual-Optimal Inequalities

DOIs for Cutting Stock (CS) and Bin Packing (BP) Problems

Overall approach:

- A priori choice of some *static* DOIs
  - → ranking inequalities
  - → one subset inequality for each $i \in I$ with a set S of cardinality 2
- Dynamic generation of additional WSIs in each CG iteration
  - → separation of most violated WSIs is a by-product of solving the subproblem with dynamic programming
  - → multiple WSIs can be separated simultaneously
- For Bin Packing: *over-stabilization!*
  - → addition of inequalities that are generally not DDOIs
  - → recovery procedure:
    - > possible to construct a pure pattern solution → done!
    - > otherwise: delete remaining DOI columns and restart CG
  - → practically over-stabilization happend very rarely for BP

Question: Can you think of DDOIs for Vertex Coloring?

Agenda

How to build a good branch-and-price algorithm?

1. Have a working textbook branch-and-price
   - $\rightarrow$ check consistency of lower bounds!
2. Repeat: try to identify and resolve the bottleneck of the algorithm
   - $\rightarrow$ most of the time spent in the pricing
   - $\rightarrow$ most of the time spent in the RMP
   - $\rightarrow$ large number of CG iterations
   - $\rightarrow$ large number of branch-and-bound nodes

On Implementing a Good Algorithm

Things to try: [Desaulniers et al., 2002]

- Most of the time spent in the pricing:
  - → see *Pricing Strategies*
  - → weaker reformulation → relaxed subproblem
  - → less subproblems → aggregation
  - → if pricing gets much harder after cutting/branching
    - > less aggressive cutting (esp. non-robust cuts)
    - > rollback strategies
  - → variable fixing
- Most of the time spent in the RMP:
  - → different algorithm (primal/dual simplex, barrier, dual ascend)
  - → stabilization
  - → make it smaller
    - > remove unpromising columns every now and then → can always be re-generated later; maybe add to a column pool
    - > handle some constraints is a lazy fashion → remove from RMP and add dynamically only when violated
    - > add less columns per pricing

On Implementing a Good Algorithm

Things to try: [Desaulniers et al., 2002]

- Large number of CG iterations:
  - → multiple column pricing
    - > dozens to a few hundred may be added
  - → *better* columns
    - > complementary columns
    - > less zero entries
  - → stabilization
  - → early branching
- Large number of branch-and-bound nodes:
  - → improve the lower bounds
    - > stronger/more cuts
    - > stronger reformulation
    - > variable fixing may help
  - → better branching strategy
  - → enumeration

Agenda

Some Suggested Readings

Check *the book*! `https://www.gerad.ca/en/papers/G-2024-36`
→ really comprehensive and numerous pointers to further readings

Interested in VRPs?

- Survey on branch-and-price: [Costa et al., 2019]
- ESPPRC subproblem: [Irnich and Desaulniers, 2005]

Interested in very(!) comprehensive branch-and-price algorithm?

- [Pecin et al., 2017]
- [Pessoa et al., 2020]

Some useful frameworks:

- GCG https://gcg.or.rwth-aachen.de/
- BaPCod https://bapcod.math.u-bordeaux.fr/
- VRPsolver https://vrpsolver.math.u-bordeaux.fr/
- PathWyse https://github.com/pathwyse
- DIP https://github.com/coin-or/Dip
- ABACUS https://software.cs.uni-koeln.de/abacus/index.html

Roberto Baldacci, Aristide Mingozzi, and Roberto Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5): 1269—-1283, 2011.

Hatem Ben Amor, Jacques Desrosiers, and José Manuel Valério de Carvalho. Dual-optimal inequalities for stabilized column generation. *Operations Research*, 54 (3):454—-463, 2006.

Nicola Bianchessi, Timo Gschwind, and Stefan Irnich. Resource-window reduction by reduced costs in path-based formulations for routing and scheduling problems. *INFORMS Journal on Computing*, 36(1):224—-244, 2024.

Marilène Cherkesly, Guy Desaulniers, and Gilbert Laporte. Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and last-in-first-out loading. *Transportation Science*, 49(4):752—-766, 2015.

Claudio Contardo, Guy Desaulniers, and François Lessard. Reaching the elementary lower bound in the vehicle routing problem with time windows. *Networks*, 65(1): 88—-99, 2015.

Luciano Costa, Claudio Contardo, and Guy Desaulniers. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53(4):946—-985, 2019.

Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon. *Accelerating Strategies in Column Generation Methods for Vehicle Routing and Crew Scheduling Problems*, pages 309—-324. Springer US, 2002.

P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849—-859, 1961.

Timo Gschwind. A comparison of column-generation approaches to the synchronized pickup and delivery problem. *European Journal of Operational Research*, 247(1): 60—71, 2015.

Timo Gschwind and Stefan Irnich. Dual inequalities for stabilized column generation revisited. *INFORMS Journal on Computing*, 28(1):175—194, 2016.

Timo Gschwind, Stefan Irnich, Fabio Furini, and Roberto Wolfler Calvo. A branch-and-price framework for decomposing graphs into relaxed cliques. *INFORMS Journal on Computing*, 33(3):1070—1090, 2021.

S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 33–65. Springer-Verlag, 2005.

Raf Jans. Classification of dantzig–wolfe reformulations for binary mixed integer programming problems. *European Journal of Operational Research*, 204(2): 251—-254, 2010.

Mads Jepsen, Bjørn Petersen, Simon Spoorendonk, and David Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2):497—511, 2008.

Philip James Neame. *Nonsmooth Dual Methods in Integer Programming*. Phd thesis, Department of Mathematics and Statistics, University of Melbourne, Melbourne, Australia, 2000.

George Nemhauser and Laurence Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., June 1988. doi: 10.1002/9781118627372.

Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61—-100, 2017.

A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30(2):339—360, 2018.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183(1–2):483—523, 2020.

Ruslan Sadykov, François Vanderbeck, Artur Pessoa, Issam Tahiri, and Eduardo Uchoa. Primal heuristics for branch and price: The assets of diving methods. *INFORMS Journal on Computing*, 31(2):251—-267, 2019.

José M. Valério de Carvalho. Using extra dual cuts to accelerate column generation. *INFORMS Journal on Computing*, 17(2):175—-182, 2005.

François Vanderbeck. Implementing mixed integer column generation. In Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon, editors, *Column Generation*, pages 331–358. Springer US, 2005.

Julia Wahlen and Timo Gschwind. Branch-price-and-cut-based solution of order batching problems. *Transportation Science*, 57(3):756—-777, 2023.

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