Practical relevance of the state-of-the-art exact VRP solvers

Ruslan Sadykov

Inria Bordeaux, France

Université Bordeaux, France

Joint work with **Eduardo Uchoa, Artur Pessoa, Eduardo Queiroga, Najib Errami**

DIMACS Workshop on Computational Approaches to Vehicle Routing

May 22, 2023

Plan of the talk

Exact SOTA algorithms for vehicle routing and their performance

POPMUSIC matheuristic for VRPs

VRPSolverEasy Python package

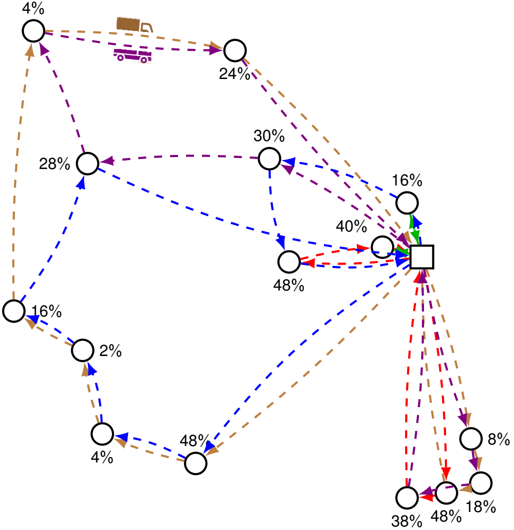
Contents

Set-partitioning formulation

- Set of feasible routes $=$ set P of resource constrained paths in graph $G = (V \cup \{0\}, A)$
- $h_a^p = 1$ if and only if path $p \in P$ contains arc a, otherwise 0
- Variable x_a — arc $a \in A$ is used in the solution or not
- Variable λ_p — path $p \in P$ is used in the solution or not

$$
\begin{aligned}
\text{Min} \quad & \sum_{a \in A} c_a x_a \\
\text{S.t.} \quad & \sum_{a \in \delta(v)} x_a = 2, && v \in V, \\
& Bx \leq b, && \text{(add. constraints and robust cuts)} \\
& D\lambda \leq d, && \text{(non-robust cuts)} \\
& x_a = \sum_{p \in P} h_a^p \lambda_p, && a \in A, \\
& \sum_{p \in P} \lambda_p \leq K, && \\
& x_a, \lambda_p \in \{0, 1\}, && a \in A, p \in P.
\end{aligned}
$$

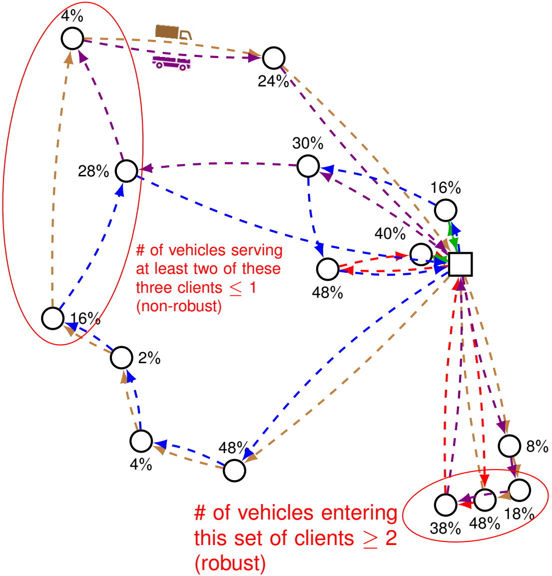
Column and cut generation: illustration



One continuous variable per feasible route.

Pricing problem is the Resource Constrained Shortest Path problem.

Column and cut generation: illustration



One continuous variable per feasible route.

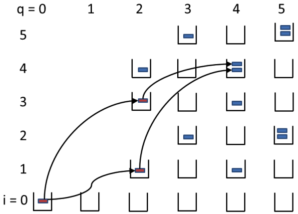
Pricing problem is the Resource Constrained Shortest Path problem.

Additional constraints (cuts) are added to reduce the integrality gap

Nodes (customers) are generalized to packing sets.

Resource constrained shortest path problem (RCSP)

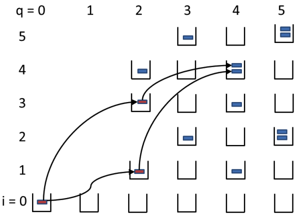
Labeling algorithm



- Enumeration of partial paths
- Relies heavily on domination
- Bi-directional search
- Buckets
- Completion bounds

Resource constrained shortest path problem (RCSP)

Labeling algorithm



- Enumeration of partial paths
- Relies heavily on domination
- Bi-directional search
- Buckets
- Completion bounds

Problem restriction using reduced cost arguments

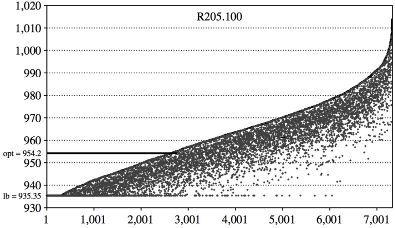


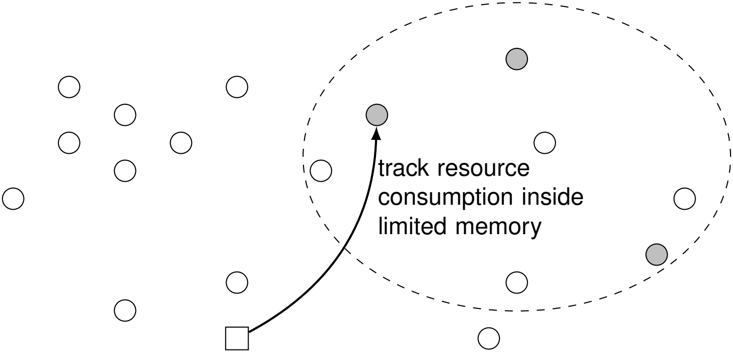
Figure from [Irnich et al., 2010]

- Remove arcs contained only in paths with reduced cost $>$ primal-dual gap
- Enumerate all elementary paths with with reduced cost $\leq$ primal-dual gap

Pricing problem relaxation: limited-memory concept

Specific RCSP instances

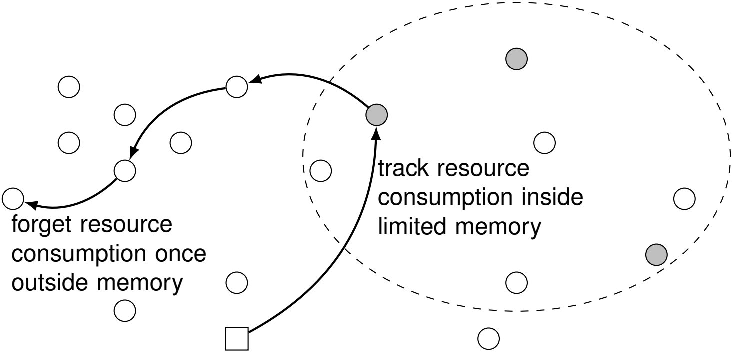
Few global resources but hundreds of local resources.



Pricing problem relaxation: limited-memory concept

Specific RCSP instances

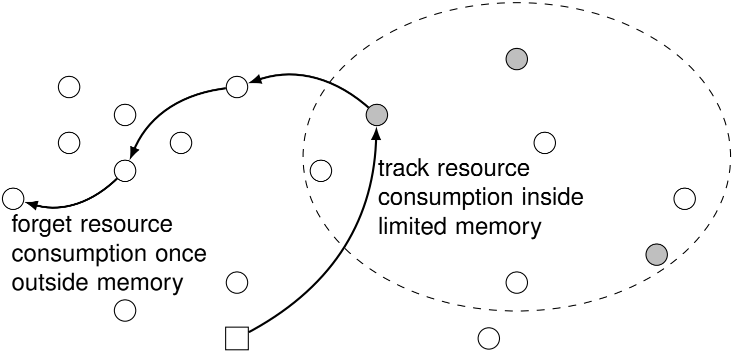
Few global resources but hundreds of local resources.



Pricing problem relaxation: limited-memory concept

Specific RCSP instances

Few global resources but hundrends of local resources.



Dynamic relaxation

Restrict relaxation after column generation convergence: increase limited memories

Heuristics and branching

Generic heuristics: diving and restricted master IP

- Work fine only for instances with short routes
- Cannot efficiently use diving heuristic due to resource relaxation and a significant primal-dual gap.
- Preliminary path enumeration with false gap is required before running restricted master heuristic.

Branching

- Strong branching is important
- Aggregated branching may be important

Some history

- [Balinski and Quandt, 1964] set-partitioning formulation for CVRP
- [Laporte and Nobert, 1983] branch-and-cut, rounded capacity cuts
- [Desrosiers et al., 1984] first branch-and-price
- [Lysgaard et al., 2004] best branch-and-cut algorithm
- [Fukasawa et al., 2006] robust branch-cut-and-price
- [Baldacci et al., 2008] path enumeration technique
- [Jepsen et al., 2008] (non-robust) subset-row cuts
- [Baldacci et al., 2011] *ng*-route relaxation
- [Pecin et al., 2017] limited-memory technique, best branch-cut-and-price
- Latest survey: [Costa et al., 2019]
- [Pessoa et al., 2020] VRP generic model and exact solver

World record for the CVRP exact solving

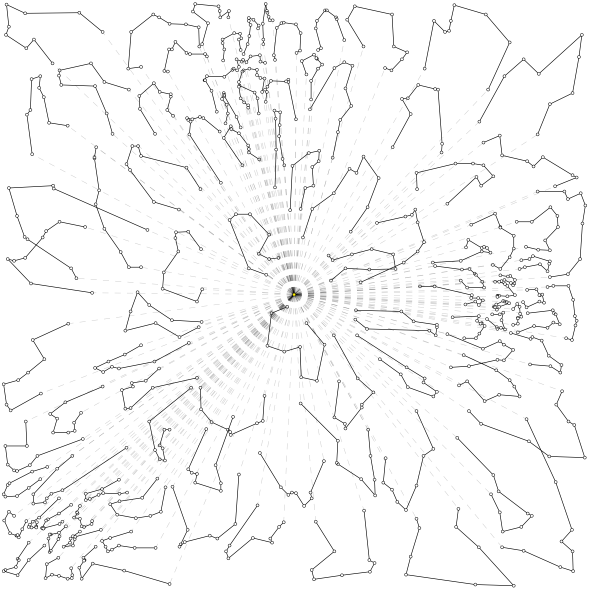
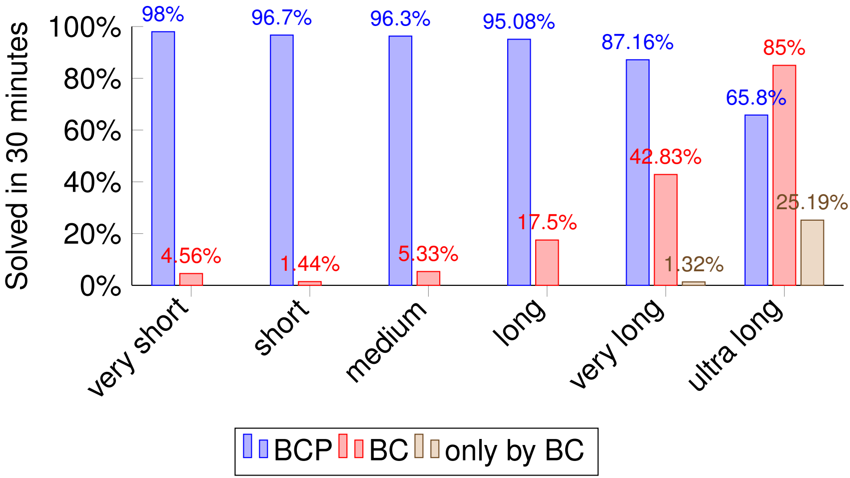


Figure: Optimal solution for X-n865-k95 (solved in ≈20 days)

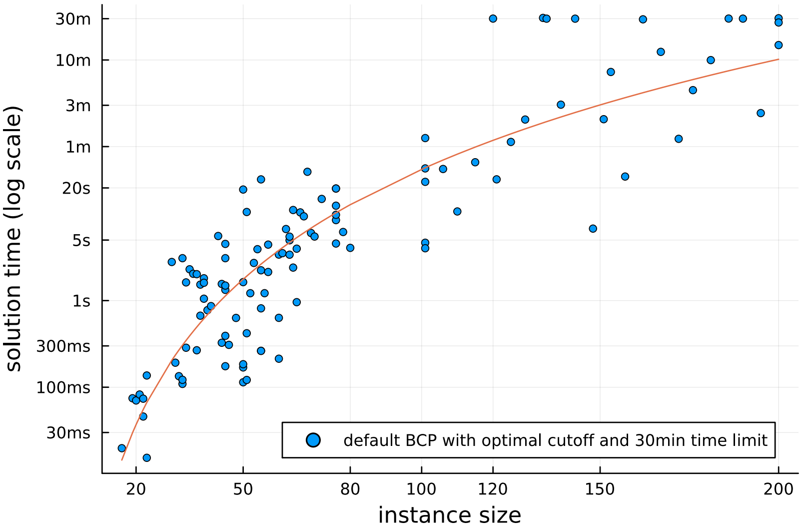
Computational comparison between BCP and BC

10'000 instances with 100 customers [Queiroga et al., 2022].



All 10'000 instances are solved to optimality thanks to cluster (aggregated) branching [Uchoa and Silva, 2022].

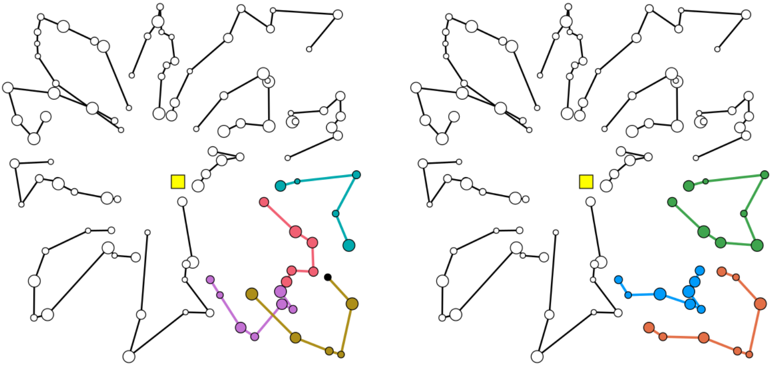
Solutions times (for CVRP)



Contents

POPMUSIC for CVRP: illustration

Partial OPtimization Metaheuristic Under Special Intensification Conditions [Taillard and Voss, 2002]



(a) Initial solution and a constructed subproblem. Seed client is marked in black.

(b) Improved solution after finding a better subsolution

POPMUSIC matheuristic for VRPs

Design choices

- Progressive increase of the subproblem size.
- Use of heuristic BCP
  - Time limit
  - False gap mechanism
  - Restricted master heuristic inside BCP

POPMUSIC matheuristic for VRPs

Design choices

- Progressive increase of the subproblem size.
- Use of heuristic BCP
  - Time limit
  - False gap mechanism
  - Restricted master heuristic inside BCP

Using heuristic instead of BCP

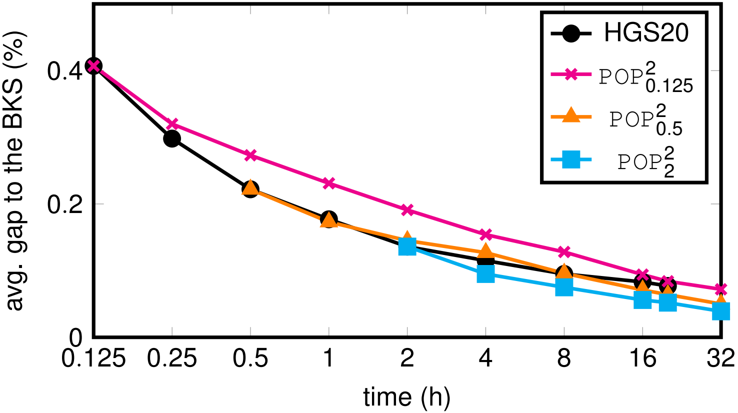
- We used [Vidal, 2022] heuristic instead of BCP for the DIMACS VRP challenge!
- Extensively studied in [Santini et al., 2023]
- However, less generic implementation.
- How to set the termination criteria for the subproblem?

Computational comparison for the CVRP

HGS20 : [Vidal, 2022].

$\texttt{POP}^2_x$: POPMUSIC-BCP starting with *x*-hour solution of [Vidal, 2022].

X instances with $300 < n \leq 1000$ customers

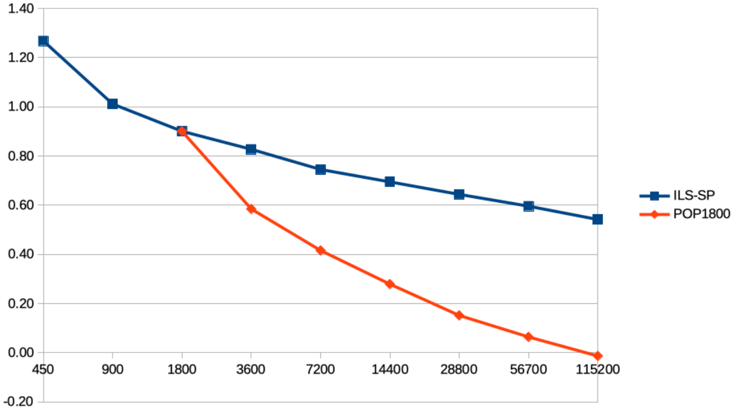


Computational results for the CVRP with backhauls

ILS-SP : [Subramanian and Queiroga, 2020].

POP1800: POPMUSIC-BCP starting with 30-minutes solution of [Subramanian and Queiroga, 2020]

Instances with 300–1000 clients.

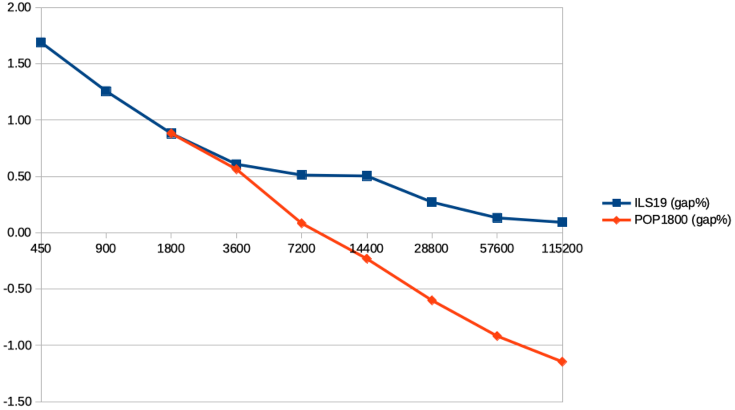


Computational results for the HFVRP

ILS19 : [Penna et al., 2019].

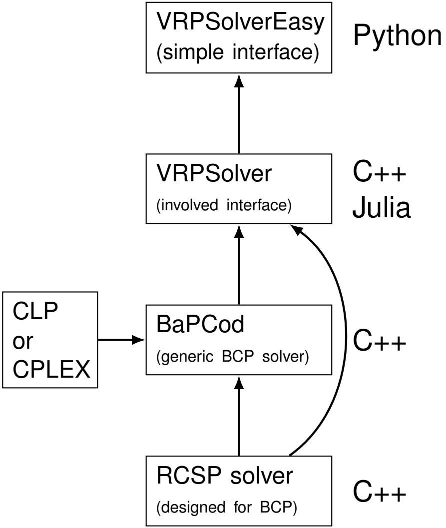
POP1800: POPMUSIC-BCP starting with 30-minutes solution of [Penna et al., 2019].

Instances of type XH with 300–1000 clients, both limited and unlimited fleet.

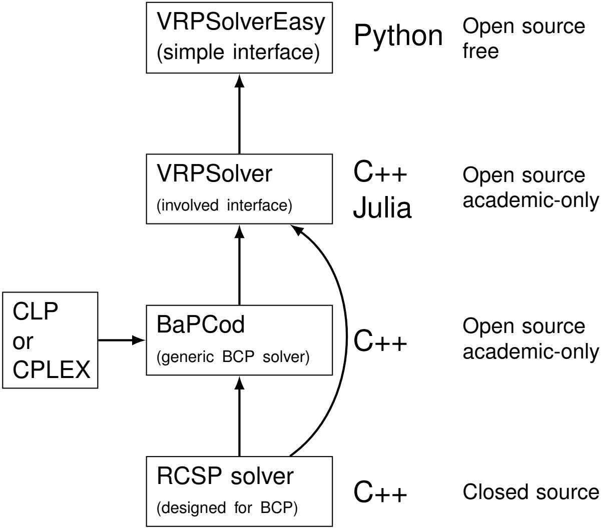


Contents

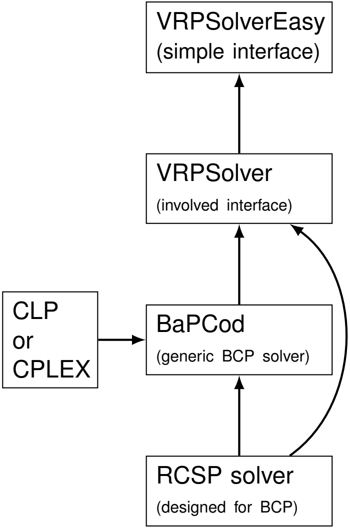
Four layers of VRPSolver implementation



Four layers of VRPSolver implementation



Four layers of VRPSolver implementation



Python	Open source free	depots, vehicles, customers
C++ Julia	Open source academic-only	variables, constraints, graphs, resources, packing sets
C++	Open source academic-only	a 15-years old mess
C++	Closed source	labels, buckets, VRP cut separation heuristics

VRPSolverEasy: compromise between generality and simplicity

`https://github.com/inria-UFF/VRPSolverEasy`

“OR-free” interface

Depots, customers, vehicles, links instead of variables, constraints, graphs, resources.

VRP variants covered

CVRP, VRPTW, HFVRP, MDVRP, OVRP, TOP, parallel links, site-dependent VRP, roaming delivery locations, and combinations.

VRP variants potentially covered

Arc routing, clustered VRP, generalized VRP, VRP with backhauls, Multi-trip VRP, Location-routing.

VRPSolverEasy: code example

```python
from VRPSolverEasy.src import solver
model = solver.Model()
model.add_depot(id=0)
for i in range(data.nb_customers):
  model.add_customer(id=i+1, demand=data.cust_demands[i])
for i,cust_i in enumerate(data.cust_coordinates):
  for j in range(i + 1, len(data.cust_coordinates)):
    dist = euclidean_distance(cust_i[0], cust_i[1],
                              data.cust_coordinates[j][0],
                              data.cust_coordinates[j][1])
    model.add_link(start_point_id=i+1,
                   end_point_id=j+1,
                   distance=dist)
model.add_vehicle_type(id=1, start_point_id=0,
                       end_point_id=0,
                       max_number=data.nb_customers,
                       capacity=data.vehicle_capacity,
                       var_cost_dist=1)
model.solve()
if model.solution.is_defined():
  print(model.solution)
```

VRPSolverEasy: computational results

Initial solution by OR-Tools run for $n/2$ sec. (included in solution time)

VRPSolverEasy time limit is 30 minutes

CVRP-Small, VRPTW-Solomon, HFVRP-Classic: $\leq$ 100 customers

gap_I, **gap_F** — OR-Tools gap, VRPSolverEasy gap

Problem	**Dataset**	Without built-in heuristic				With built-in heuristic			
		#opt	**Time**	**gap_I**	**gap_F**	**#opt**	**Time**	**gap_I**	**gap_F**
CVRP	Small	87/87	33.8	0.99	0.00	87/87	31.0	0.99	0.00
	X$\leq$ 200	14/26	844.9	4.36	2.35	16/26	617.3	4.36	1.96
VRPTW	Solomon	53/56	109.0	14.20	0.14	54/56	112.2	14.20	0.10
	HG-200	27/60	881.0	9.02	4.63	28/60	879.7	9.16	4.78
HFVRP	Classic	31/40	234.5	10.35	0.95	36/40	154.2	10.32	0.34
	XH$\leq$ 200	2/22	1730.4	18.07	13.44	8/22	1088.6	18.65	8.25
Total		215/291				229/291			

Practical relevance of exact VRP solvers

Possible now

- Finally easy to use!
- Small and moderate size (up to $\approx$100 customers)
- Not so long routes (up to $\approx$15 customers per route)
- For these instances
  - More generic than SOTA heuristic VRP solvers
  - More efficient than generic VRP solvers (OR-Tools, LocalSolver)

Perspectives

- Industrial-quality codes are needed!
- Parallel matheuristics for larger instances
- Hybrid heuristics which use dual information from column and cut generation

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.

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.

Balinski, M. and Quandt, R. (1964).
On an integer program for a delivery problem.
Operations Research, 12(2):300–304.

Costa, L., Contardo, C., and Desaulniers, G. (2019).
Exact branch-price-and-cut algorithms for vehicle routing.
Transportation Science, Forthcoming.

References II

Desrosiers, J., Soumis, F., and Desrochers, M. (1984).
Routing with time windows by column generation.
Networks, 14(4):545–565.

Fukasawa, R., Longo, H., Lysgaard, J., Aragão, M. P. d., Reis, M., Uchoa, E., and Werneck, R. F. (2006).
Robust branch-and-cut-and-price for the capacitated vehicle routing problem.
Mathematical Programming, 106(3):491–511.

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.

References III

Laporte, G. and Nobert, Y. (1983).
A branch and bound algorithm for the capacitated vehicle routing problem.
Operations-Research-Spektrum, 5(2):77–85.

Lysgaard, J., Letchford, A. N., and Eglese, R. W. (2004).
A new branch-and-cut algorithm for the capacitated vehicle routing problem.
Mathematical Programming, 100(2):423–445.

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.

Penna, P. H. V., Subramanian, A., Ochi, L. S., Vidal, T., and Prins, C. (2019).
A hybrid heuristic for a broad class of vehicle routing problems with heterogeneous fleet.
Annals of Operations Research, 273(1):5–74.

References IV

Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2020).
A generic exact solver for vehicle routing and related problems.
Mathematical Programming, 183:483–523.

Queiroga, E., Sadykov, R., Uchoa, E., and Vidal, T. (2022).
10,000 optimal CVRP solutions for testing machine learning based heuristics.
In *AAAI-22 Workshop on Machine Learning for Operations Research (ML4OR)*.

Santini, A., Schneider, M., Vidal, T., and Vigo, D. (2023).
Decomposition strategies for vehicle routing heuristics.
INFORMS Journal on Computing, Ahead of print.

Subramanian, A. and Queiroga, E. (2020).
Solution strategies for the vehicle routing problem with backhauls.
Optimization Letters, 14(8):2429–2441.

References V

Taillard, E. D. and Voss, S. (2002).
POPMUSIC — Partial OPtimization Metaheuristic under Special Intensification Conditions, pages 613–629.
Springer US, Boston, MA.

Uchoa, E. and Silva, J. a. M. P. (2022).
Branching on clustered vrp instances.
In *32nd EURO Conference*.

Vidal, T. (2022).
Hybrid genetic search for the cvrp: Open-source implementation and swap* neighborhood.
Computers & Operations Research, 140:105643.