Exact Algorithms for Vehicle Routing: advances, challenges, and perspectives

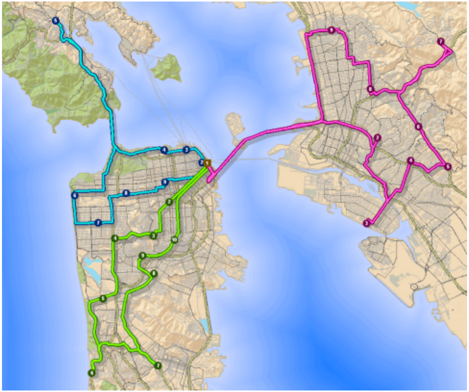
Eduardo Uchoa

Departamento de Engenharia de Produção
Universidade Federal Fluminense, Brazil
INRIA International Chair 2022-2026, Bordeaux

Vehicle Routing Problem (VRP)

One of the most widely studied in Combinatorial Optimization:

- +8,000 works published only in 2023 (Google Scholar), mostly heuristics
- Direct application in the real systems that distribute goods and provide services



Vehicle Routing Problem (VRP)

Reflecting the variety of real transportation systems, VRP literature is spread into hundreds of variants. For example, there are variants that consider:

- Vehicle capacities,
- Time windows,
- Heterogeneous fleets,
- Multiple depots,
- Split delivery, pickup and delivery, backhauling,
- Arc routing (Ex: garbage collection),
- etc, etc.

Emerging variants:

- Electric vehicle routing, drone routing, warehouse routing...

Traveling Salesman Problem (TSP)

The most famous NP-hard problem is a single-vehicle VRP variant!

Instance: Complete graph $G = (V, E)$. Edge $e \in E$ costs c_e.

Solution: A single route that visits each customer once and minimizes the total cost.

TSP Edge Formulation (Dantzig et al. [1954])

$$\min \sum_{e \in E} c_e x_e \tag{1}$$

$$\text{S.t.} \quad \sum_{e \in \delta(i)} x_e = 2 \qquad \forall\, i \in V, \tag{2}$$

$$\sum_{e \in \delta(S)} x_e \geq 2 \qquad \forall\, S \subset V, \tag{3}$$

$$x_e \in \{0,1\} \qquad \forall\, e \in E. \tag{4}$$

Constraints (3) are *Subtour Elimination Cuts*

TSP Computational Progress

Table 1.5 Milestones in the solution of TSP instances.

1954	G. Dantzig, R. Fulkerson, S. Johnson	49 cities	dantzig42
1971	M. Held and R. M. Karp	57 cities	[304]
1971	M. Held and R. M. Karp	64 cities	random points
1975	P. M. Camerini, L. Fratta, F. Maffioli	67 cities	random points
1975	P. Miliotis	80 cities	random points
1977	M. Grötschel	120 cities	gr120
1980	H. Crowder and M. W. Padberg	318 cities	lin318
1987	M. Padberg and G. Rinaldi	532 cities	att532
1987	M. Grötschel and O. Holland	666 cities	gr666
1987	M. Padberg and G. Rinaldi	1,002 cities	pr1002
1987	M. Padberg and G. Rinaldi	2,392 cities	pr2392

TSP Computational Progress - Concorde (Applegate, Bixby, Chvátal and Cook)

Table 1.6 Solution of TSP instances with Concorde.

1992	Concorde	3,038 cities	pcb3038
1993	Concorde	4,461 cities	fnl4461
1994	Concorde	7,397 cities	pla7397
1998	Concorde	13,509 cities	usa13509
2001	Concorde	15,112 cities	d15112
2004	Concorde	24,978 cities	sw24978
2004	Concorde with Domino-Parity	33,810 cities	pla33810
2006	Concorde with Domino-Parity	85,900 cities	pla85900

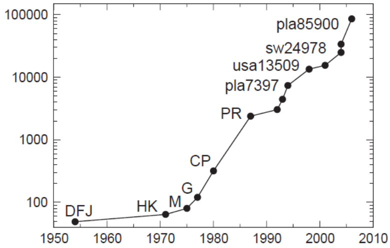


Figure 1.45 Further progress in the TSP, log scale.

TSP Computational Progress- Concorde (ABCC)

Table 1.6 Solution of TSP instances with Concorde.

1992	Concorde	3,038 cities	pcb3038
1993	Concorde	4,461 cities	fnl4461
1994	Concorde	7,397 cities	pla7397
1998	Concorde	13,509 cities	usa13509
2001	Concorde	15,112 cities	d15112
2004	Concorde	24,978 cities	sw24978
2004	Concorde with Domino-Parity	33,810 cities	pla33810
2006	Concorde with Domino-Parity	85,900 cities	pla85900

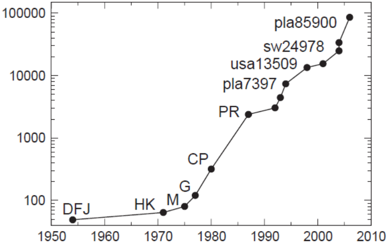


Figure 1.45 Further progress in the TSP, log scale.

pla85900: 135 years of CPU! A more practical result: most TSP instances with up to 1,000 cities are now solvable in a few minutes

TSP Computational Progress- Concorde (ABCC)

Table 1.6 Solution of TSP instances with Concorde.

1992	Concorde	3,038 cities	pcb3038
1993	Concorde	4,461 cities	fnl4461
1994	Concorde	7,397 cities	pla7397
1998	Concorde	13,509 cities	usa13509
2001	Concorde	15,112 cities	d15112
2004	Concorde	24,978 cities	sw24978
2004	Concorde with Domino-Parity	33,810 cities	pla33810
2006	Concorde with Domino-Parity	85,900 cities	pla85900

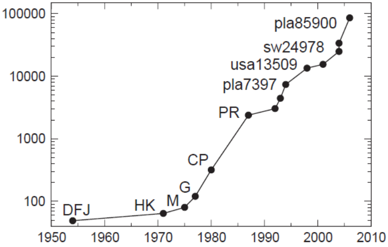


Figure 1.45 Further progress in the TSP, log scale.

Big success of "formulation + polyhedral investigation + separation procedures = branch-and-cut algorithm" paradigm

First (Dantzig and Ramser [1959]) and **most basic multi-vehicle variant**:

Instance: Complete graph $G = (V, E)$ with $V = \{0, \ldots, n\}$; 0 is the depot, $N = \{1, \ldots, n\}$ is the set of customers. Each edge $e \in E$ costs c_e. Each $i \in N$ demands d_i units. Homogeneous unlimited fleet of vehicles with capacity Q.

Solution: Routes from the depot, respecting the capacities and visiting all customers once; minimizing the total cost.

CVRP Edge Formulation (Laporte and Nobert [1983])

$$\min \sum_{e \in E} c_e x_e \tag{5}$$

$$\text{S.t.} \quad \sum_{e \in \delta(i)} x_e = 2 \qquad \forall\, i \in N, \tag{6}$$

$$\sum_{e \in \delta(S)} x_e \geq 2 \lceil \sum_{i \in S} d_i / Q \rceil \qquad \forall\, S \subseteq N, \tag{7}$$

$$x_e \in \{0,1\} \qquad \forall\, e \in E \setminus \delta(0), \tag{8}$$

$$x_e \in \{0,1,2\} \qquad \forall\, e \in \delta(0). \tag{9}$$

Constraints (7) are *Rounded Capacity Cuts*

Trying to emulate the TSP: BC Algorithms for CVRP

Extensive research on families of cuts:

- Framed Capacity, Strengthened Comb, Multistar, Extended Hypotour, etc.

Dominant approach (Naddef and Rinaldi [2002]) until early 2000's:

- Araque, Kudva, Morin, and Pekny [1994]
- Augerat, Belenguer, Benavent, Corberán, Naddef, and Rinaldi [1995]
- Blasum and Hochstättler [2000]
- Ralphs, Kopman, Pulleyblank, and Trotter Jr. [2003]
- Achuthan, Caccetta, and Hill [2003]
- Ralphs [2003]
- Wenger [2004]
- Lysgaard, Letchford, and Eglese [2004]

Best BC results

Class	Size	#Ins	LLE04 #Unsolved	Root gap	Avg Time(s)
A	36-79	22	7	2.06	6638
B	36-79	20	1	0.61	8178
E-M	50-199	12	9	2.10	39592
F	44-134	3	0	0.06	1016
P	14-100	24	8	2.26	11219
Total		81	25		
Processor			Intel Celeron 700MHz		

Avg Times only on solved instances. Smallest unsolved instance: 49 customers

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

Why BC is so much better on TSP than on CVRP?

Dantzig et al. [1954] TSP formulation is already quite strong

- Subtour Cuts separated in poly-time and typically obtain gaps $\leq 0.5\%$
- $+$ Blossom Cuts (2-matching) separated in poly-time, typical gaps $\leq 0.2\%$
- $+$ Fantastic heuristic separation of complex cuts, typical root gaps $\leq 0.05\%$ $\longrightarrow$ small BC trees (except on big instances)

Laporte and Nobert [1983] CVRP formulation not so strong

- Rounded Capacity Cuts separation already NP-hard. Excellent heuristics do exist (checked against exact MIP separation) and obtain typical gaps $\geq 3\%$
- $+$ all known cuts, typical root gaps around 2% $\longrightarrow$ very large BC trees on many instances with less than 100 customers

Set Partitioning Formulation – SPF (Balinski and Quandt [1964])

Ω is the set of routes, route r costs c_r, coefficient a_{ir} indicates how many times r visits customer i

$$\min \sum_{r\in\Omega} c_r \lambda_r \tag{10}$$

$$\text{S.t.} \quad \sum_{r\in\Omega} a_{ir}\lambda_r = 1 \qquad \forall\ i \in N, \tag{11}$$

$$\lambda_r \in \{0,1\} \qquad \forall\ r \in \Omega. \tag{12}$$

- Exponential number of variables $\Longrightarrow$ Column generation / Branch-and-Price (BP) algorithms
- Pricing elementary routes is strongly NP-hard $\Longrightarrow$ Relax Ω including some non-elementary q-routes (Christofides et al. [1979])

Combining Column Generation and Cut Separation

SPF linear relaxation is very good for VRPTW *with tight windows* (Desrosiers et al. [1984]) but **is quite weak for CVRP**:

- Typical root gaps >**3%**, worse than **2%** of BC root gap

Fukasawa et al. [2006] combined both methods. A cut over the edge variables

$$\sum_{e \in E} \alpha_e x_e \geq b,$$

is translated to

$$\sum_{r \in \Omega} \Big(\sum_{e \in E} \alpha_e a_{er} \Big) \lambda_r \geq b,$$

where a_{er} is the number of times that e is used in route r.

The combination of column generation with cuts defined over edges yields root gaps around **1%**.

Robust vs Non-robust Branch-Cut-and-Price (BCP)

A crucial point in combining column generation with cuts is the effect of the new dual variables in the pricing:

- A cut is **robust** when its dual variable can be translated into costs in the pricing. The subproblem structure does not change. Any cut over the edge variables is robust
- **non-robust** cuts change the pricing structure, each additional cut makes it harder. General cuts over the route variables are non-robust

Non-robust cuts are potentially stronger because a route variable carries much more information than an edge variable

Robust BCP results in FLL+06

		LLE04			FLL+06		
Class	#Ins	NS	Gap	T(s)	NS	Gap	T(s)
A	22	7	2.06	6638	0	0.81	1961
B	20	1	0.61	8178	0	0.47	4763
E-M	12	9	2.10	39592	3	1.19	126987
F	3	0	0.06	1016	0	0.06	2398
P	24	8	2.26	11219	0	0.76	2892
Total	81	25			3		
Processor		Intel Celeron 700MHz			Pentium 4 2.4GHz		

BCP solved all literature instances with up to 134 customers. Three larger instances remained open: M-n151-k12, M-n200-k16 e M-n200-k17.

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

Since then, all CVRP exact algorithms combine cut and column generation

Baldacci, Christofides and Mingozzi [2008]

- Uses **non-robust cuts**: Strengthened Capacity and Clique.
- Instead of branching, the algorithm finishes by **enumerating** all routes with reduced cost smaller than the gap. The SPF with only those routes is solved by a MIP solver

R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351–385, 2008

Baldacci, Mingozzi and Roberti [2011]

Introduces *ng*-**routes**, an effective elementarity relaxation better than *q*-routes.

Non-robust Subset Row Cuts (Chvátal-Gomory Cuts of Rank 1 over a small number of set partitioning constraints [Jepsen et al., 2008]) replace Cliques, smaller impact on pricing

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

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

Røpke [2012]

Back to Robust BCP, but already using *ng*-routes.

- Proposes a sophisticated and aggressive **strong branching**, reducing a lot the branch-and-bound trees

M-n151-k12 solved in 5 days!

S. Røpke. Branching decisions in branch-and-cut-and-price algorithms for vehicle routing problems. *Presentation in Column Generation 2012*, 2012

Contardo and Martinelli [2014]

- **Enumeration to a pool** with up to several million routes can be performed. After that, pricing is done by inspection in the pool.
  - Non-robusts cuts can be freely separated
  - As lower bounds improve, fixing by reduced costs reduce pool size
  - The problem is finished by a MIP solver only when pool size is much reduced

M-n151-k12 solved in 3 hours!

C. Contardo and R. Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146, 2014a

Pecin et al. [2014]

A complex BCP algorithm incorporating elements from **all** previously mentioned works and presenting some new ideas.

D. Pecin, A. Pessoa, M. Poggi, and Uchoa. E. Improved branch-cut-and-price for capacitated vehicle routing. In *Proceedings of the 17th IPCO*, pages 393–403. Springer, 2014

Pecin et al. [2014]: Cuts

- Robust Cuts
  - Rounded Capacity
  - Strengthened Comb
- Non-Robust Cuts
  - Subset Row
- Post-Enumeration Cuts
  - Cliques

Pecin et al. [2014]: Pricing

Most critical part of the BCP: dynamic programming labeling algorithm, handling:

- *ng*-routes
- The modifications induced by Subset Row Cuts

Features:

- Bi-directional search (Righini and Salani [2006]), using balanced mid-point
- Completion Bounds with under-evaluation (Contardo [2012])
- Fixing by reduced cost considering arc accumulated load (Pessoa et al. [2010])

Pecin et al. [2014]: Non-Robustness Control

Even with all care, non-robust cuts are indeed "non-robust":

- Pricing may be handling hundreds of Subset Row Cuts well. Then, the separation of a few dozen additional cuts makes the pricing 100x or even 1000x slower!
- When such a situation is detected, the algorithm **rolls back**, removing the "bad" cuts.

Pecin et al. [2014]: Miscellanea

Hybrid search strategy, combining branching and enumeration:

- Strong Branching (SB):
  - Hierarchical, 3 levels
  - Keeps full history to guide future decisions
  - Up to 200 candidates per node.
- Route enumeration:
  - Generates pool with up to 20M routes
  - Branching can be done over enumerated node
  - Uses MIP solver only when pool is reduced to less than 20K routes

Dual stabilization by smoothing (Wentges [1997])

Results in PPPU14

Class	#Ins	BMR11 US	BMR11 Gap	BMR11 T(s)	Rop12 US	Rop12 Gap	Rop12 T(s)	CM14 US	CM14 Gap	CM14 T(s)
A	22	0	0.13	30	0	0.57	53	0	0.09	59
B	20	0	0.06	67	0	0.25	208	0	0.08	34
E-M	12	3	0.49	303	2	0.96	44295	2	0.27	1548
F	3	1	0.11	164	0	0.25	2163	0	0.03	27772
P	24	0	0.23	85	0	0.69	280	0	0.18	240
Total	81	4			2			2		
Processor		Xeon X7350 2.93GHz			Core i7-2620M 2.7GHz			Xeon E5462 2.8GHz		

Class	#Ins	PPPU14 UnSolved	PPPU14 Gap	PPPU14 T(s)
A	22	0	0.03	5.6
B	20	0	0.04	6.2
E-M	12	0	0.19	3669
F	3	0	0.00	3679
P	24	0	0.07	33
Total	81	0		
Processor		Core i7-3770 3.4GHz		

M-n151-k12 solved in 3 minutes! M-n200-k16 and M-n200-k17 solved in a few hours

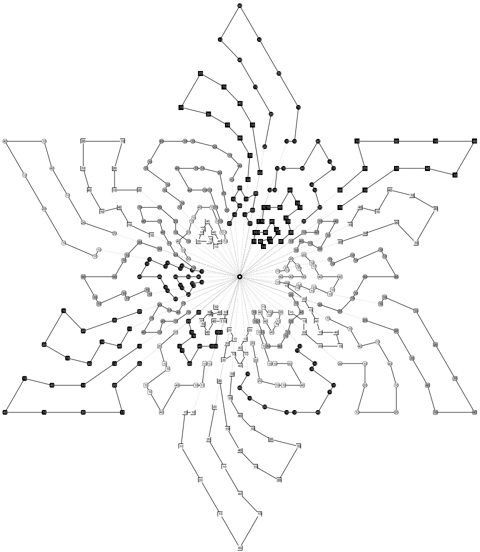
Golden Instances

Golden, Wasil, Kelly and Chao [1998] proposed 12 CVRP instances, having from 240 to 483 customers.

- Frequent in the heuristic literature
- Considered "out of reach" of exact algorithms

6 instances could be solved, with 240, 252, 300, 320, 360 and 420 customers.

Optimal solution Golden_20 (420 customers), 7 days CPU time, cost 1817.59; best heuristic 1817.86



Breakthrough in Pecin et al.[2014]: Limited Memory Cut

The concept of limited memory cut was pivotal for those improvements.

- The Subset Row Cuts are weakened in order to reduce their impact in the labeling algorithm used in the pricing
- However, those coefficients are dynamically adjusted in order to obtain the same bounds as regular SRCs

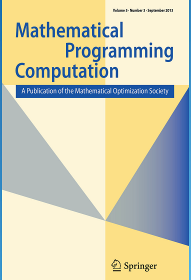
Advanced Branch-Cut-and-Price Algorithms

The advanced state-of-the-art BCPAs are those where the non-robust cuts and the pricing algorithm are jointly and symbiotically designed, in such a way that the pricing can handle a large number of very tailored non-robust cuts without becoming too inefficient.

- The presented **limited memory technique is good for the labeling algorithm**. If the pricing was being solved by another method (for example, by MIP), they would actually make the pricing harder!

Springer springer.com

This certificate is awarded at the 23rd International Symposium on Mathematical Programming, Bordeaux, France, July 2018



MPC Best Paper in 2017

The editorial board of MPC has chosen

Improved Branch-Cut-and-Price for Capacitated Vehicle Routing

by **Diego Pecin, Artur Pessoa, Marcus Poggi** and **Eduardo Uchoa**
MPC, volume 9, pp. 61-100, March 2017

Generic Exact VRPSolver

Creating advanced BCP for new VRP variants

D. Pecin, C. Contardo, G. Desaulniers, and Uchoa. E. New enhancements for exactly solving the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 29:489–502, 2017a

A. Pessoa, R. Sadykov, and Uchoa. E. Enhanced branch-cut-and-price algorithm for heterogeneous fleet vehicle routing problems. *European Journal of Operational Research*, 270:530–543, 2018

Diego Pecin and Eduardo Uchoa. Comparative analysis of capacitated arc routing formulations for designing a new branch-cut-and-price algorithm. *Transportation Science*, 53(6):1673–1694, 2019

Creating advanced BCP for new VRP variants

Creating a BCP algorithm for a new variant:

- Takes a lot of time, even when they adapt an already existing code for another variant
- Intricate conceptual issues for adapting some techniques for more complex variants

One would like to have a generic algorithm that could be easily customized to many variants

VRPSolver

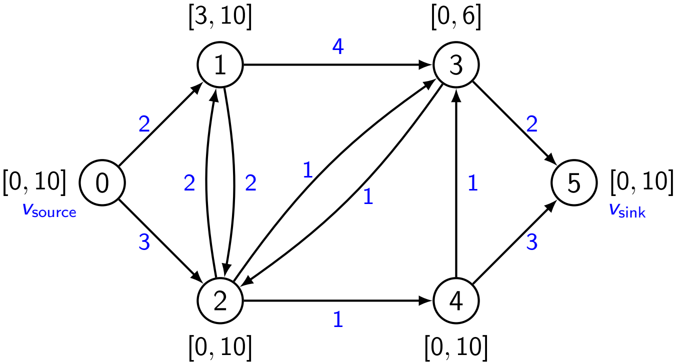
An advanced BCP solver for a generic model that encompasses a wide class of VRPs and several some other kinds of problems

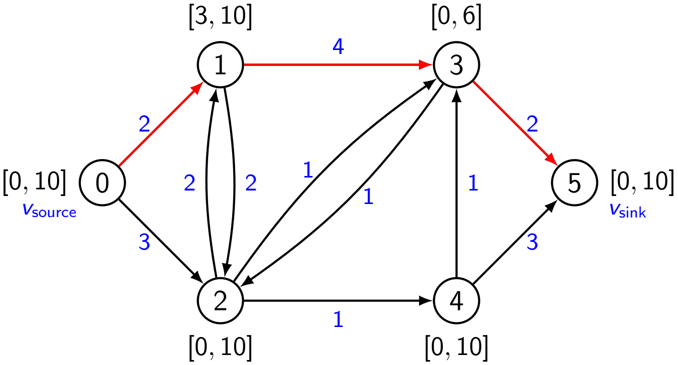
The VRPSolver model is really generic:

- Not based on “routing elements”, like customers, vehicles, depots, etc.
- Instead, it is **a general MIP + constraints forcing some variables to be linear combinations of resource constrained shortest paths in user-defined graphs**

A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183(1):483–523, 2020

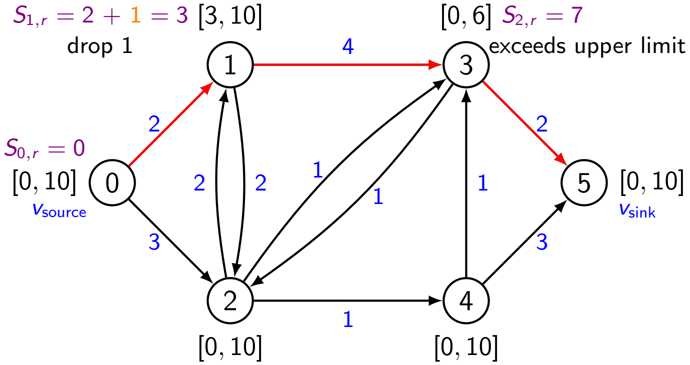
Resource Constrained Paths over a single resource





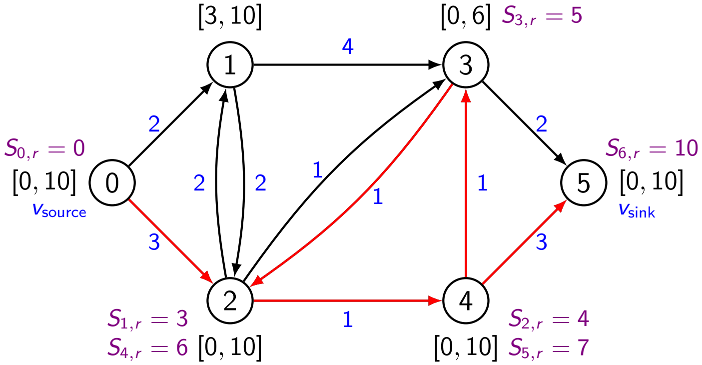
Path 1: 0 – 1 – 3 – 5

Resource Constrained Paths over a single resource



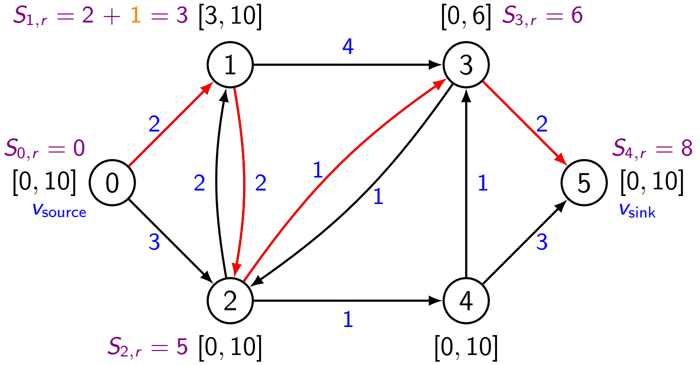
Path 1: 0 – 1 – 3 – 5 (Infeasible)

Resource Constrained Paths over a single resource

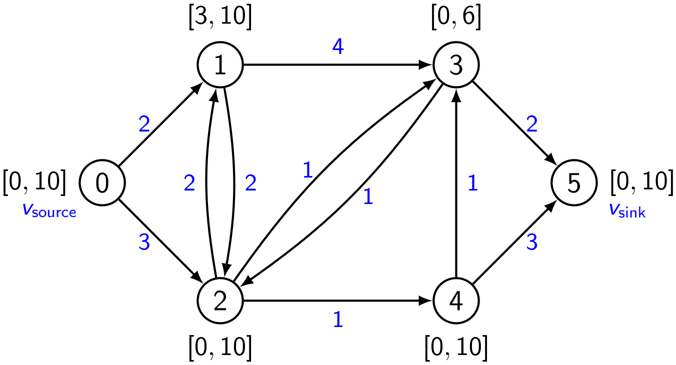


Path 1: 0 – 1 – 3 – 5 (Infeasible)
Path 2: 0 – 2 – 4 – 3 – 2 – 4 – 5

Resource Constrained Paths over a single resource



Path 1: 0 – 1 – 3 – 5 (Infeasible)
Path 2: 0 – 2 – 4 – 3 – 2 – 4 – 5 (Feasible)
Path 3: 0 – 1 – 2 – 3 – 5



Path 1: 0 – 1 – 3 – 5 (Infeasible)
Path 2: 0 – 2 – 4 – 3 – 2 – 4 – 5 (Feasible)
Path 3: 0 – 1 – 2 – 3 – 5 (Feasible)

Model: Variables and Mappings

After defining the graphs (with their set of resources, arc-consumptions, and intervals on accumulated consumption), part of the arcs are **mapped** to variables.

- Mapped x variables
  - Each variable x_j, $1 \leq j \leq n_1$, is mapped into a non-empty set $M(x_j) \subseteq A$.
  - The inverse mapping of arc a is $M^{-1}(a) = \{j | a \in M(x_j)\}$.
  - Some M^{-1} sets may be empty.

Model: Formulation

h_a^p (constant): how many times arc a is used in path p
λ_p (variable): how many times path p is used in the solution

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n_1} c_j x_j + \sum_{s=1}^{n_2} f_s y_s & & (13a)\\
\text{S.t.} \quad & \sum_{j=1}^{n_1} \alpha_{ij} x_j + \sum_{s=1}^{n_2} \beta_{is} y_s \geq d_i, & i = 1, \ldots, m, & \quad (13b)\\
& x_j = \sum_{k \in K} \sum_{p \in P^k} \left(\sum_{a \in M(x_j)} h_a^p \right) \lambda_p, & j = 1 \ldots, n_1, & \quad (13c)\\
& L^k \leq \sum_{p \in P^k} \lambda_p \leq U^k, & k \in K, & \quad (13d)\\
& \lambda_p \in Z_+, & p \in P, & \quad (13e)\\
& x_j \in Z,\ y_s \in Z & j = 1, \ldots, \bar{n}_1,\ s = 1, \ldots, \bar{n}_2. &
\end{aligned}
$$

Model: Formulation

Eliminating the x variables and relaxing the integrality constraints:

$$\min \quad \sum_{k\in K}\sum_{p\in P^k}\left(\sum_{j=1}^{n_1} c_j \sum_{a\in M(x_j)} h_a^p\right)\lambda_p + \sum_{s=1}^{n_2} f_s y_s \tag{14a}$$

$$\text{S.t.} \quad \sum_{k\in K}\sum_{p\in P^k}\left(\sum_{j=1}^{n_1} \alpha_{ij} \sum_{a\in M(x_j)} h_a^p\right)\lambda_p + \sum_{s=1}^{n_2} \beta_{is} y_s \geq d_i, \qquad i = 1,\ldots,m, \tag{14b}$$

$$L^k \leq \sum_{p\in P^k} \lambda_p \leq U^k, \qquad k \in K, \tag{14c}$$

$$\lambda_p \geq 0, \qquad p \in P. \tag{14d}$$

Model: Formulation

Eliminating the x variables and relaxing the integrality constraints:

$$\min \quad \sum_{k\in K}\sum_{p\in P^k}\left(\sum_{j=1}^{n_1} c_j \sum_{a\in M(x_j)} h_a^p\right)\lambda_p + \sum_{s=1}^{n_2} f_s y_s \tag{14a}$$

$$\text{S.t.} \quad \sum_{k\in K}\sum_{p\in P^k}\left(\sum_{j=1}^{n_1} \alpha_{ij} \sum_{a\in M(x_j)} h_a^p\right)\lambda_p + \sum_{s=1}^{n_2} \beta_{is} y_s \geq d_i, \qquad i = 1,\dots,m, \tag{14b}$$

$$L^k \leq \sum_{p\in P^k} \lambda_p \leq U^k, \qquad k \in K, \tag{14c}$$

$$\lambda_p \geq 0, \qquad p \in P. \tag{14d}$$

Master LP (14) is solved by column generation

Example: Generalized Assignment Problem (GAP)

Compact IP formulation:

$$\min \sum_{k \in K} \sum_{t \in T} c_t^k x_t^k \tag{15a}$$

$$\text{S.t.} \quad \sum_{k \in K} x_t^k = 1, \qquad t \in T; \tag{15b}$$

$$\sum_{t \in T} w_t^k x_t^k \leq Q^k, \qquad k \in K; \tag{15c}$$

$$x_t^k \in \{0, 1\}, \qquad t \in T, k \in K. \tag{15d}$$

VRPSolver Model for GAP

RCSP Graphs G^k (modeling binary knapsack problems)

$V^k = \{v^k_t : t = 0, \dots, |T|\}$;

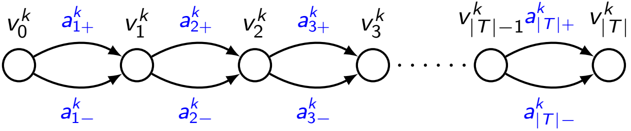
$A^k = \{a^k_{t+} = (v^k_{t-1}, v^k_t),\ a^k_{t-} = (v^k_{t-1}, v^k_t) : t = 1, \dots, |T|\}$;

$v^k_{\text{source}} = v^k_0,\ v^k_{\text{sink}} = v^k_{|T|}$;

$R^k = R^k_M = \{1\}$;

$q_{a^k_{t+},1} = w^k_t, q_{a^k_{t-},1} = 0, t \in T$;

$[l_{v^k_t,1}, u_{v^k_t,1}] = [0, Q^k],\ t \in T \cup \{0\}$.



VRPSolver Model for GAP

Formulation

Binary variables x_t^k, $t \in T$, $k \in K$.

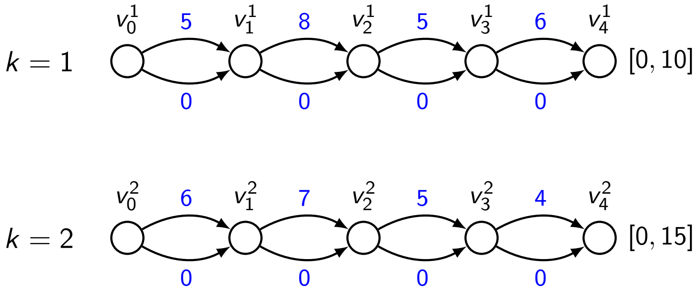
$$\min \quad \sum_{k \in K} \sum_{t \in T} c_t^k x_t^k \tag{16a}$$

$$\text{S.t.} \quad \sum_{k \in K} x_t^k = 1, \qquad t \in T; \tag{16b}$$

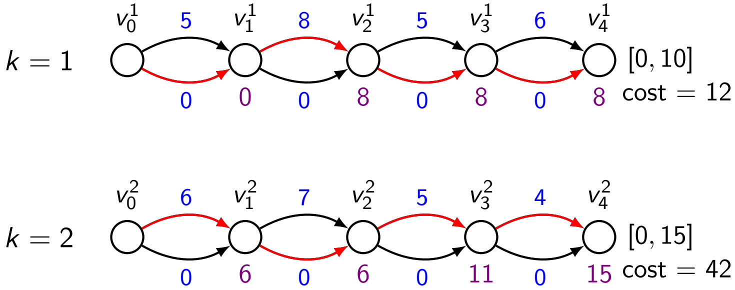
Mapping $M(x_t^k) = \{a_{t+}^k\}, t \in T, k \in K;\ L^k = 0,\ U^k = 1,\ k \in K.$

Example: VRPSolver Model for GAP

		cost (c_t^k)				load (w_t^k)				Q^k
tasks		**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	
machines	**1**	10	12	8	14	5	8	5	6	10
	2	15	10	9	18	6	7	5	4	15



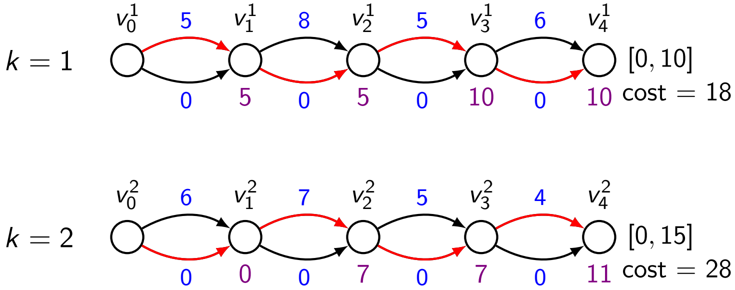
		cost (c_t^k)				load (w_t^k)				Q^k
tasks		**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	
machines	**1**	10	12	8	14	5	8	5	6	10
	2	15	10	9	18	6	7	5	4	15



The costs are actually defined over the x variables, the path costs are calculated via mapping

Example: VRPSolver Model for GAP

		cost (c_t^k)				load (w_t^k)				Q^k
tasks		**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	
machines	**1**	10	12	8	14	5	8	5	6	10
	2	15	10	9	18	6	7	5	4	15



The costs are actually defined over the x variables, the path costs are calculated via mapping

Example of Graphs for a GAP instance

		cost (c_t^k)				load (w_t^k)				Q^k
tasks		**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	
machines	**1**	10	12	8	14	5	8	5	6	10
	2	15	10	9	18	6	7	5	4	15

$$p_1 = (a^1_{1-},\ a^1_{2-},\ a^1_{3-},\ a^1_{4-}) \quad p_{10} = (a^2_{1-},\ a^2_{2-},\ a^2_{3+},\ a^2_{4-})$$
$$p_2 = (a^1_{1+},\ a^1_{2-},\ a^1_{3-},\ a^1_{4-}) \quad p_{11} = (a^2_{1-},\ a^2_{2-},\ a^2_{3-},\ a^2_{4+})$$
$$p_3 = (a^1_{1-},\ a^1_{2+},\ a^1_{3-},\ a^1_{4-}) \quad p_{12} = (a^2_{1+},\ a^2_{2+},\ a^2_{3-},\ a^2_{4-})$$
$$p_4 = (a^1_{1-},\ a^1_{2-},\ a^1_{3+},\ a^1_{4-}) \quad p_{13} = (a^2_{1+},\ a^2_{2-},\ a^2_{3+},\ a^2_{4-})$$
$$p_5 = (a^1_{1-},\ a^1_{2-},\ a^1_{3-},\ a^1_{4+}) \quad p_{14} = (a^2_{1+},\ a^2_{2-},\ a^2_{3-},\ a^2_{4+})$$
$$p_6 = (a^1_{1+},\ a^1_{2-},\ a^1_{3+},\ a^1_{4-}) \quad p_{15} = (a^2_{1-},\ a^2_{2+},\ a^2_{3+},\ a^2_{4-})$$
$$p_7 = (a^2_{1-},\ a^2_{2-},\ a^2_{3-},\ a^2_{4-}) \quad p_{16} = (a^2_{1-},\ a^2_{2+},\ a^2_{3-},\ a^2_{4+})$$
$$p_8 = (a^2_{1+},\ a^2_{2-},\ a^2_{3-},\ a^2_{4-}) \quad p_{17} = (a^2_{1-},\ a^2_{2-},\ a^2_{3+},\ a^2_{4+})$$
$$p_9 = (a^2_{1-},\ a^2_{2+},\ a^2_{3-},\ a^2_{4-}) \quad p_{18} = (a^2_{1+},\ a^2_{2-},\ a^2_{3+},\ a^2_{4+})$$

$$P^1 = \{p_1, p_2, p_3, p_4, p_5, p_6\}$$
$$P^2 = \{p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}\}$$

VRPSolver Model for GAP

IP model (corresponding to (13)):

$$
\begin{aligned}
\min \quad & \sum_{k\in K}\sum_{t\in T} c_t^k x_t^k && && (17a)\\
\text{S.t.} \quad & \sum_{k\in K} x_t^k = 1, && t\in T; && (17b)\\
& x_t^k = \sum_{p\in P^k : a_{t+}^k \in p} \lambda_p, && t\in T, k\in K; && (17c)\\
& 0 \le \sum_{p\in P^k} \lambda_p \le 1, && k\in K; && (17d)\\
& \lambda_p \in Z_+, && p\in P; && (17e)\\
& x_j \in Z, && t\in T, k\in K. && (17f)
\end{aligned}
$$

Eliminating the x variables and relaxing the integrality constraints:

$$\min \quad \sum_{k\in K}\sum_{p\in P^k}\left(\sum_{t:a^k_{t+}\in p} c^k_t\right)\lambda_p \tag{18a}$$

$$\text{S.t.} \quad \sum_{k\in K}\sum_{p\in P^k:a^k_{t+}\in p}\lambda_p = 1, \qquad t\in T; \tag{18b}$$

$$\sum_{p\in P^k}\lambda_p \le 1, \qquad k\in K; \tag{18c}$$

$$\lambda_p \ge 0, \qquad p\in P. \tag{18d}$$

Example: VRPSolver Model for GAP

$$
\begin{aligned}
\min\ z = \quad & 10x_1^1 + 12x_2^1 + 8x_3^1 + 14x_4^1 + 15x_1^2 + 10x_2^2 + 9x_3^2 + 18x_4^2 \\
\text{S.t.} \quad & x_1^1 + x_1^2 = 1, \\
& x_2^1 + x_2^2 = 1, \\
& x_3^1 + x_3^2 = 1, \\
& x_4^1 + x_4^2 = 1, \\
& x_1^1 = \lambda_2 + \lambda_6, \\
& x_2^1 = \lambda_3, \\
& x_3^1 = \lambda_4 + \lambda_6, \\
& x_4^1 = \lambda_5, \\
& x_1^2 = \lambda_8 + \lambda_{12} + \lambda_{13} + \lambda_{14} + \lambda_{18}, \\
& x_2^2 = \lambda_9 + \lambda_{12} + \lambda_{15} + \lambda_{16}, \\
& x_3^2 = \lambda_{10} + \lambda_{13} + \lambda_{15} + \lambda_{17} + \lambda_{18}, \\
& x_4^2 = \lambda_{11} + \lambda_{14} + \lambda_{16} + \lambda_{17} + \lambda_{18}, \\
& 0 \leq \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 \leq 1, \\
& 0 \leq \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{12} + \lambda_{13} + \lambda_{14} + \lambda_{15} + \lambda_{16} + \lambda_{17} + \lambda_{18} \\
& x \in Z_+^8, \lambda \in Z_+^{18}.
\end{aligned}
$$

Formulation (17)

Example: VRPSolver Model for GAP

Master LP (18) (identical to the Master LP obtained by standard DW decomposition):

$$
\begin{aligned}
\min \ z = \quad & 10\lambda_2 + 12\lambda_3 + 8\lambda_4 + 14\lambda_5 + 18\lambda_6 + 0\lambda_7 + 15\lambda_8 + 10\lambda_9 + 9\lambda_{10} \\
& +18\lambda_{11} + 25\lambda_{12} + 24\lambda_{13} + 33\lambda_{14} + 19\lambda_{15} + 28\lambda_{16} + 27\lambda_{17} + 42\lambda_{18} \\
\text{S.t.} \quad & \lambda_2 + \lambda_6 + \lambda_8 + \lambda_{12} + \lambda_{13} + \lambda_{14} + \lambda_{18} = 1, \\
& \lambda_3 + \lambda_9 + \lambda_{12} + \lambda_{15} + \lambda_{16} = 1, \\
& \lambda_4 + \lambda_6 + \lambda_{10} + \lambda_{13} + \lambda_{15} + \lambda_{17} + \lambda_{18} = 1, \\
& \lambda_5 + \lambda_{11} + \lambda_{14} + \lambda_{16} + \lambda_{17} + \lambda_{18} = 1, \\
& \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 \leq 1, \\
& \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{12} + \lambda_{13} + \lambda_{14} + \lambda_{15} + \lambda_{16} + \lambda_{17} + \lambda_{18} \leq 1, \\
& \lambda \geq 0.
\end{aligned}
$$

Optimal Solution: $z = 46$; $\lambda_6 = 1, \lambda_{16} = 1$

Computational Experiments

- VRPSolver algorithms coded in C++ over BaPCod package (Vanderbeck et al. [2018])
- Models are defined using a Julia–JuMP (Dunning et al. [2017]) based interface.

Tests over 13 problems: CVRP, VRPTW, HFVRP, Multi-Depot VRP (MDVRP), (Capacitated) Team Orienteering Problem (CTOP/TOP), Capacitated Profitable Tour Problem (CPTP), VRP with Service Level constraints (VRPSL), GAP, Vector Packing Problem (VPP), Bin Packing Problem (BPP) and CARP.

Computational results

Problem	Data set	#	T.L.	VRPSolver	Best Published		2nd Best Published	
CVRP	E-M	12	10h	12 (61s)	**12 (49s)**	Pecin et al. [2017b]	10 (432s)	Contardo et al. [2014]
	X	58	60h	**36 (147m)**	34 (209m)	Uchoa et al. [2017]	—	
VRPTW	Sol Hard	14	1h	**14 (5m)**	13 (17m)	Pecin et al. [2017a]	9 (39m)	Baldacci et al. [2011a]
	Hom 200	60	30h	**56 (21m)**	50 (70m)	Pecin et al. [2017a]	7 (-)	Kallehauge et al. [2006]
HFVRP	Golden	40	1h	**40 (144s)**	39 (287s)	Pessoa et al. [2018]	34 (855s)	Baldacci et al. [2009]
MDVRP	Cordeau	11	1h	**11 (6m)**	11 (7m)	Pessoa et al. [2018]	9 (25m)	Contardo et al. [2014]
PDPTW	RC	40	1h	**40 (5m)**	33 (17m)	Gschwind et al. [2018]	32 (14m)	Baldacci et al. [2011b]
	LiLim	30	1h	3 (56m)	**23 (20m)**	Baldacci et al. [2011b]	18 (27m)	Gschwind et al. [2018]
TOP	Chao 4	60	1h	**55 (8m)**	39 (15m)	Bianchessi et al. [2018]	30 (-)	El-Hajj et al. [2016]
CTOP	Archetti	14	1h	**13 (7m)**	7 (34m)	Archetti et al. [2013]	6 (35m)	Archetti et al. [2009]
CPTP	Archetti	28	1h	**24 (9m)**	0 (1h)	Bulhoes et al. [2018]	0 (1h)	Archetti et al. [2013]
VRPSL	Bulhoes	180	2h	**159 (16m)**	49 (90m)	Bulhoes et al. [2018]	—	
GAP	OR-Lib D	6	2h	5 (40m)	**5 (30m)**	Posta et al. [2012]	5 (46m)	Avella et al. [2010]
	Nauss	30	1h	**25 (23m)**	1 (58m)	Gurobi [2017]	0 (1h)	Nauss [2003]
VPP	1,4,5,9	40	1h	**38 (8m)**	13 (50m)	Heßler et al. [2018]	10 (53m)	Brandão et al. [2016]
BPP	Falk T	80	10m	80 (16s)	**80 (1s)**	Brandão et al. [2016]	80 (24s)	Belov et al. [2006,16]
	Hard28	28	10m	28 (17s)	**28 (7s)**	Belov et al. [2006,16]	26 (14s)	Brandão et al. [2016]
	AI	250	1h	**160 (25m)**	116 (35m)	Belov et al. [2006,16]	100 (40m)	Brandão et al. [2016]
	ANI	250	1h	**103 (35m)**	97 (40m)	Wei et al. [2019]	67 (45m)	Belov et al. [2006,16]
CARP	Eglese	24	30h	**22 (36m)**	22 (43m)	Pecin et al. [2019]	10 (237m)	Bartolini et al. [2013]

Table: VRPSolver vs best specific solvers on 13 problems.

VRPSolver on CVRP

Class X with 100 instances, ranging between 100 and 1000 customers:

- Designed to mimic a wide diversity of characteristics found in real applications
- Available at CVRPLIB (http://vrp.atd-lab.inf.puc-rio.br/index.php/en/)

E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858, 2017

VRPSolver on CVRP

55 out of 100 instances could be solved, sometimes with very special parameterization and very long runs (up to one month):

- $100 \leq n < 200$: 22/22 (100%)
- $200 \leq n < 300$: 19/21 (90%)
- $300 \leq n < 500$: 10/25 (40%)
- $500 \leq n \leq 1000$: 4/32 (12%)

Smallest unsolved: X-n280-k17

Largest solved: X-n856-k95

Optimal solution X-856-k95, 10 days of CPU time

VRP Solver available over Julia–JuMP or C++

The VRP solver is available for academic use (vrpsolver.math.u-bordeaux.fr):

- Algorithms bundled in a single pre-compiled docker (runs in every OS), with a Julia–JuMP user interface for modeling
- Or, no-docker more complex C++ interface for Linux

Unhappily, VRPSolver documentation, support, etc is far worse than it should be. We are now trying to fix that!

We are trying to make it a collaborative fully open-source project inspired by SCIP!

```julia
function build_model(data::DataCVRP)
   E = edges(data)
   n = nb_customers(data)
   V = [i for i in 0:n]
   V⁺ = [i for i in 1:n]
   Q = veh_capacity(data)
   cvrp = VrpModel()
   @variable(cvrp.formulation, x[e in E], Int)
   @objective(cvrp.formulation, Min, sum(c(data,e) * x[e] for e in E))
   @constraint(cvrp.formulation, deg[i in V⁺], sum(x[e] for e in δ(data, i)) == 2.0)
   function build_graph()
      v_source = v_sink = 0
      G = VrpGraph(cvrp, V, v_source, v_sink, (0, n))
      cap_res_id = add_resource(G, main = true)
      for i in V
         set_resource_bounds(G, i, cap_res_id, 0, Q)
      end
      for (i,j) in E
         arc_id = add_arc(G, i, j, [x[(i,j)]])
         set_arc_consumption(G, arc_id, cap_res_id, d(data, j))
         arc_id = add_arc(G, j, i, [x[(i,j)]])
         set_arc_consumption(G, arc_id, cap_res_id, d(data, i))
      end
      return G
   end
   G = build_graph()
   add_graph(cvrp, G)
   set_vertex_packing_sets(cvrp, [[(G,i)] for i in V⁺])
   define_packing_sets_distance_matrix(cvrp, [[dist(data, (i, j)) for j in V⁺] for i in V⁺])
   add_capacity_cut_separator(cvrp, [ ( [(G,i)], d(data, i) ) for i in V⁺], Q)
   set_branching_priority(cvrp, "x", 1)
   return (cvrp, x)
end
```

Success Cases - 45 papers, two in Operations Research

- Capacitated VRP (CVRP)
- CVRP with Time Windows
- Heterogeneous Fleet CVRP
- Multi-depot CVRP
- Pickup-and-Delivery Problem with Time Windows
- CVRP with Backhauls
- Multi-Trip VRP with Time Windows
- Team Orienteering Problem
- Capacitated Profitable Tour Problem
- VRP With Service Levels
- Clustered VRP
- Generalized VRP
- Cumulative VRP
- Split-Delivery VRP with Time Windows
- Joint Routing of Conventional and Electric Vehicles
- Multi-Period VRP with Workload Equity
- Generalized Assignment Problem
- Vector Packing Problem
- (Variable Size) Bin Packing Problem
- Capacitated Arc Routing Problem
- Robust CVRP with Demand Uncertainty
- Location-Routing Problem
- Two-Echelon VRP
- Black-and-White TSP
- Coloured TSP
- TSP with Hotels
- Differential Harvest Problem
- Parallel Machine Scheduling with Sum Objectives
- Prize-collecting Job Sequencing Problem with Secondary Resources
- Multi-Shuttle Crane Scheduling in Automated Storage and Retrieval Systems

Room for Creative Modeling

We believe users may find original ways of fitting new problems in the proposed model

- Not only VRP variants, possibly also problems from scheduling, packing, network design, etc.

Since VRP solving technology is quite advanced, there is a chance of obtaining better-than-existing-methods performance

Some works that used VRPSolver

T. Bulhões, R. Sadykov, A. Subramanian, and E. Uchoa. On the exact solution of a large class of parallel machine scheduling problems. *Journal of Scheduling*, 23(4):411–429, 2020

A. Pessoa, M. Poss, R. Sadykov, and F. Vanderbeck. Branch-cut-and-price for the robust capacitated vehicle routing problem with knapsack uncertainty. *Operations Research*, 69(3):739–754, 2021a

A. Pessoa, R. Sadykov, and E. Uchoa. Solving bin packing problems using VRPSolver models. *Operations Research Forum*, 2(2):1–25, 2021b

T. Adamo, G. Ghiani, P. Greco, and E. Guerriero. Properties and bounds for the single-vehicle capacitated routing problem with time-dependent travel times and multiple trips. In *10th ICORES*, pages 82–87, 2021

C. Damião, J.M. Silva, and E. Uchoa. A BCP algorithm for the cumulative capacitated vehicle routing problem. *4OR*, pages 1–25, 2021

Some works that used VRPSolver

A. Subramanyam, T. Cokyasar, J. Larson, and M. Stinson. Joint routing of conventional and range-extended electric vehicles in a large metropolitan network. *Transportation Research Part C*, 144:103830, 2022

I. Mohamed, W. Klibi, R. Sadykov, H. Şen, and F. Vanderbeck. The two-echelon stochastic multi-period capacitated location-routing problem. *European Journal of Operational Research*, 2022

G. Volte, E. Bourreau, R. Giroudeau, and O. Naud. Using VRPSolver to efficiently solve the differential harvest problem. *Computers & Operations Research*, page 106029, 2022

M. Freitas, J. M. Silva, and E. Uchoa. A unified exact approach for clustered and generalized vehicle routing problems. *Computers & Operations Research*, page 106040, 2022

Some works that used VRPSolver

P. Liguori, A.R. Mahjoub, G. Marques, R. Sadykov, and E. Uchoa. Nonrobust strong knapsack cuts for capacitated location routing and related problems. *Operations Research*, 71:1577–1595, 2023

I. Balster, T. Bulhões, P. Munari, A. Pessoa, and R. Sadykov. A new family of route formulations for split delivery vehicle routing problems. *Transportation Science*, 57:1359–1378, 2023

M. Roboredo, R. Sadykov, and E. Uchoa. Solving vehicle routing problems with intermediate stops using VRPSolver models. *Networks*, 81:399–416, 2023

R. Praxedes, T. Bulhões, A. Subramanian, and E. Uchoa. A unified exact approach for a broad class of vehicle routing problems with simultaneous pickup and delivery. *Computers & Operations Research*, 162:106467, 2024

VRPSolverEasy

We realized that modeling over VRPSolver was too difficult for most of its potential users

VRPSolverEasy, a simple interface for VRPSolver. Not fully general, can model the most standard VRP variants in terms depots, customers, links, and vehicle types

- Capacitated vehicles, customer time windows, heterogeneous fleet, multiple depots, open routes, optional customers with penalties, parallel links to model time/cost trade-offs, incompatibilities between vehicles and customers, customers with alternative locations and/or time windows.
- Free for all users

Najib Errami, Eduardo Queiroga, Ruslan Sadykov, and Eduardo Uchoa. VRPSolverEasy: a Python library for the exact solution of a rich vehicle routing problem. *INFORMS Journal on Computing*, 2023

Conclusions and Perspectives

Impact of Exact VRP algorithms in practice

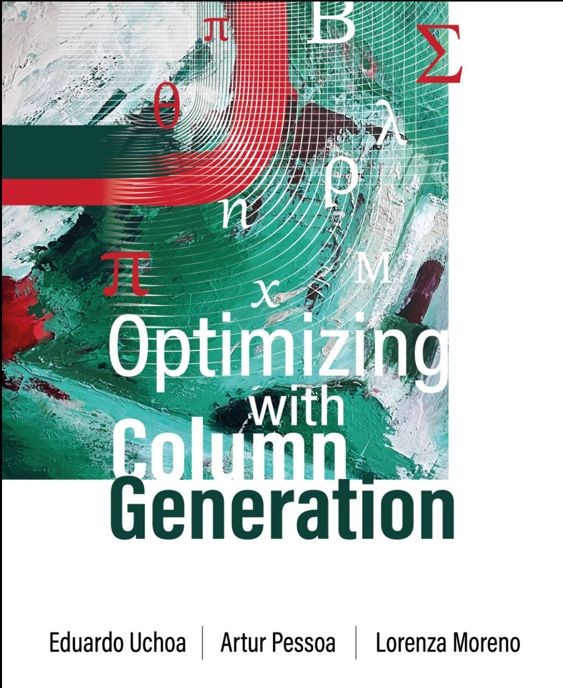
Historically, exact solvers were rarely used in practical routing

1. Existing algorithms could not solve realistic-sized instances in reasonable times
   - Now many instances of the most classic VRPs with up to 200 customers can be solved
   - More importantly, **many instances with up to 100 customers can be solved in a few minutes**
2. The real problems seldom correspond exactly to one of the classic variants. Creating a good exact code for a new variant is a hard task
   - Highly customizable codes with state-of-the-art performance are now available

We expect that exact algorithms will be much more used by practitioners, at least for benchmarking their heuristics

BRANCH-AND-PRICE

JACQUES DESROSIERS
MARCO LÜBBECKE
GUY DESAULNIERS
JEAN BERTRAND GAUTHIER


Optimizing
with
Column
Generation
Eduardo Uchoa | Artur Pessoa | Lorenza Moreno

“Optimizing with Column Generation: advanced branch-cut-and-price algorithms”

Cover design by Leonardo Viana over his 1m $\times$ 1m acrylic on canvas named “Ciclo”

Goals:

- Extensive historical research
- Beginner-friendly, starts from the basics
- Yet, provides in-depth coverage of the recent advanced BCP techniques
- Not only theoretical content, guidelines on the efficient practical implementation
- GitHub community of readers helping each other, sharing implementations for the project exercises, discussions

“Optimizing with Column Generation: advanced branch-cut-and-price algorithms”

Part I is 100% finished and available for download at:
https://optimizingwithcolumngeneration.github.io/

300 pages covering the fundamentals of CG

Thank you!

N. Achuthan, L. Caccetta, and S. Hill. An improved branch-and-cut algorithm for the capacitated vehicle routing problem. *Transportation Science*, 37:153–169, 2003.

T. Adamo, G. Ghiani, P. Greco, and E. Guerriero. Properties and bounds for the single-vehicle capacitated routing problem with time-dependent travel times and multiple trips. In *10th ICORES*, pages 82–87, 2021.

J. Araque, G. Kudva, T. Morin, and J. Pekny. A branch-and-cut algorithm for the vehicle routing problem. *Annals of Operations Research*, 50:37–59, 1994.

C. Archetti, D. Feillet, A. Hertz, and M G Speranza. The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, 60(6):831–842, Jun 2009.

C. Archetti, N. Bianchessi, and M.G. Speranza. Optimal solutions for routing problems with profits. *Discrete Applied Mathematics*, 161(4–5):547–557, 2013.

P. Augerat, J. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report 949-M, Université Joseph Fourier, Grenoble, France, 1995.

Pasquale Avella, Maurizio Boccia, and Igor Vasilyev. A computational study of exact knapsack separation for the generalized assignment problem. *Computational Optimization and Applications*, 45(3):543–555, 2010.

R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351–385, 2008.

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

Roberto Baldacci and Aristide Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2):347–380, 2009.

Roberto Baldacci, Enrico Bartolini, and Aristide Mingozzi. An exact algorithm for the pickup and delivery problem with time windows. *Operations Research*, 59(2):414–426, 2011b.

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

I. Balster, T. Bulhões, P. Munari, A. Pessoa, and R. Sadykov. A new family of route formulations for split delivery vehicle routing problems. *Transportation Science*, 57:1359–1378, 2023.

Enrico Bartolini, Jean-François Cordeau, and Gilbert Laporte. Improved lower bounds and exact algorithm for the capacitated arc routing problem. *Mathematical Programming*, 137(1): 409–452, Feb 2013.

G. Belov and G. Scheithauer. A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. *European Journal of Operational Research*, 171(1):85 – 106, 2006.

Nicola Bianchessi, Renata Mansini, and M. Grazia Speranza. A branch-and-cut algorithm for the team orienteering problem. *International Transactions in Operational Research*, 25(2): 627–635, 2018.

U. Blasum and W. Hochstättler. Application of the branch and cut method to the vehicle routing problem. Technical Report ZPR2000-386, Zentrum fur Angewandte Informatik Köln, 2000.

Filipe Brandão and João Pedro Pedroso. Bin packing and related problems: General arc-flow formulation with graph compression. *Computers & Operations Research*, 69:56 – 67, 2016.

T. Bulhões, R. Sadykov, A. Subramanian, and E. Uchoa. On the exact solution of a large class of parallel machine scheduling problems. *Journal of Scheduling*, 23(4):411–429, 2020.

Teobaldo Bulhoes, Minh Hoàng Hà, Rafael Martinelli, and Thibaut Vidal. The vehicle routing problem with service level constraints. *European Journal of Operational Research*, 265(2):544 – 558, 2018.

N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, *Combinatorial Optimization, A Wiley-Interscience Publication, Based on a series of lectures, given at the Summer School in Combinatorial Optimization, held in Sogesta, Italy,* *May 30th-June 11th, 1977, Chichester: Wiley, 1979, edited by Christofides, Nicos*, volume 1, pages 315–338, 1979.

C. Contardo. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. Technical report, Archipel-UQAM 5078, Université du Québec à Montréal, Canada, 2012.

C. Contardo and R. Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146, 2014a.

Claudio Contardo and Rafael Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146, 2014b.

C. Damião, J.M. Silva, and E. Uchoa. A BCP algorithm for the cumulative capacitated vehicle routing problem. *4OR*, pages 1–25, 2021.

G. Dantzig and R. Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.

George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410, 1954.

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

Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2): 295–320, 2017.

Racha El-Hajj, Duc-Cuong Dang, and Aziz Moukrim. Solving the team orienteering problem with cutting planes. *Computers & Operations Research*, 74:21 – 30, 2016.

Najib Errami, Eduardo Queiroga, Ruslan Sadykov, and Eduardo Uchoa. VRPSolverEasy: a Python library for the exact solution of a rich vehicle routing problem. *INFORMS Journal on Computing*, 2023.

M. Freitas, J. M. Silva, and E. Uchoa. A unified exact approach for clustered and generalized vehicle routing problems. *Computers & Operations Research*, page 106040, 2022.

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

Timo Gschwind, Stefan Irnich, Ann-Kathrin Rothenbächer, and Christian Tilk. Bidirectional labeling in column-generation algorithms for pickup-and-delivery problems. *European Journal of Operational Research*, 266(2):521 – 530, 2018.

Gurobi. Gurobi optimizer reference manual, version 7.5, 2017. URL http://www.gurobi.com.

Katrin Heßler, Timo Gschwind, and Stefan Irnich. Stabilized branch-and-price algorithms for vector packing problems. *European Journal of Operational Research*, 271(2):401 – 419, 2018.

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

B. Kallehauge, J. Larsen, and O.B.G. Madsen. Lagrangian duality applied to the vehicle routing problem with time windows. 33 (5):1464–1487, 2006.

G. Laporte and Y. Nobert. A branch and bound algorithm for the capacitated vehicle routing problem. *OR Spectrum*, 5(2):77–85, 1983.

P. Liguori, A.R. Mahjoub, G. Marques, R. Sadykov, and E. Uchoa. Nonrobust strong knapsack cuts for capacitated location routing and related problems. *Operations Research*, 71:1577–1595, 2023.

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

I. Mohamed, W. Klibi, R. Sadykov, H. Şen, and F. Vanderbeck. The two-echelon stochastic multi-period capacitated location-routing problem. *European Journal of Operational Research*, 2022.

Denis Naddef and Giovanni Rinaldi. Branch-and-cut algorithms for the capacitated vrp. In *The vehicle routing problem*, pages 53–84. SIAM, 2002.

Robert M. Nauss. Solving the generalized assignment problem: An optimizing and heuristic approach. *INFORMS Journal on Computing*, 15(3):249–266, 2003.

D. Pecin and Uchoa. E. Comparative analysis of capacitated arc routing formulations for designing a new branch-cut-and-price algorithm. *Transportation Science*, (Forthcoming), 2019.

D. Pecin, A. Pessoa, M. Poggi, and Uchoa. E. Improved branch-cut-and-price for capacitated vehicle routing. In *Proceedings of the 17th IPCO*, pages 393–403. Springer, 2014.

D. Pecin, C. Contardo, G. Desaulniers, and Uchoa. E. New enhancements for exactly solving the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 29: 489–502, 2017a.

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

Diego Pecin and Eduardo Uchoa. Comparative analysis of capacitated arc routing formulations for designing a new branch-cut-and-price algorithm. *Transportation Science*, 53(6): 1673–1694, 2019.

A. Pessoa, R. Sadykov, and Uchoa. E. Enhanced branch-cut-and-price algorithm for heterogeneous fleet vehicle routing problems. *European Journal of Operational Research*, 270:530–543, 2018.

A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183(1):483–523, 2020.

A. Pessoa, M. Poss, R. Sadykov, and F. Vanderbeck. Branch-cut-and-price for the robust capacitated vehicle routing problem with knapsack uncertainty. *Operations Research*, 69(3): 739–754, 2021a.

A. Pessoa, R. Sadykov, and E. Uchoa. Solving bin packing problems using VRPSolver models. *Operations Research Forum*, 2(2):1–25, 2021b.

Marius Posta, Jacques A. Ferland, and Philippe Michelon. An exact method with variable fixing for solving the generalized assignment problem. *Computational Optimization and Applications*, 52:629–644, 2012.

R. Praxedes, T. Bulhões, A. Subramanian, and E. Uchoa. A unified exact approach for a broad class of vehicle routing problems with simultaneous pickup and delivery. *Computers & Operations Research*, 162:106467, 2024.

T. Ralphs. Parallel branch and cut for capacitated vehicle routing. *Parallel Computing*, 29:607–629, 2003.

T. Ralphs, L. Kopman, W. Pulleyblank, and L. Trotter Jr. On the capacitated vehicle routing problem. *Mathematical Programming*, 94:343–359, 2003.

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

M. Roboredo, R. Sadykov, and E. Uchoa. Solving vehicle routing problems with intermediate stops using VRPSolver models. *Networks*, 81:399–416, 2023.

S. Røpke. Branching decisions in branch-and-cut-and-price algorithms for vehicle routing problems. *Presentation in Column Generation 2012*, 2012.

A. Subramanyam, T. Cokyasar, J. Larson, and M. Stinson. Joint routing of conventional and range-extended electric vehicles in a large metropolitan network. *Transportation Research Part C*, 144:103830, 2022.

E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858, 2017.

François Vanderbeck, Ruslan Sadykov, and Issam Tahiri. BaPCod — a generic Branch-And-Price Code, 2018. URL `https://realopt.bordeaux.inria.fr/?page_id=2`.

G. Volte, E. Bourreau, R. Giroudeau, and O. Naud. Using VRPSolver to efficiently solve the differential harvest problem. *Computers & Operations Research*, page 106029, 2022.

Laguna Wei, Zhixing Luo, Roberto Baldacci, and Andrew Lim. A new branch-and-price-and-cut algorithm for one-dimensional bin-packing problems. *INFORMS Journal on Computing*, Forthcoming, 2019.

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