VRPSolver: a Generic Exact Solver for Vehicle Routing and Related Problems

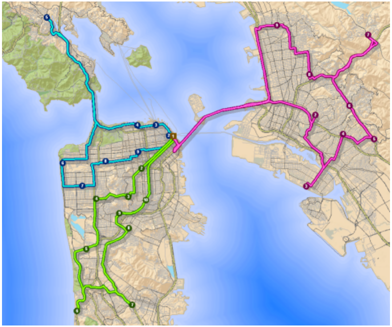
Eduardo Uchoa

Departamento de Engenharia de Produção
Universidade Federal Fluminense
Niterói - RJ, Brasil

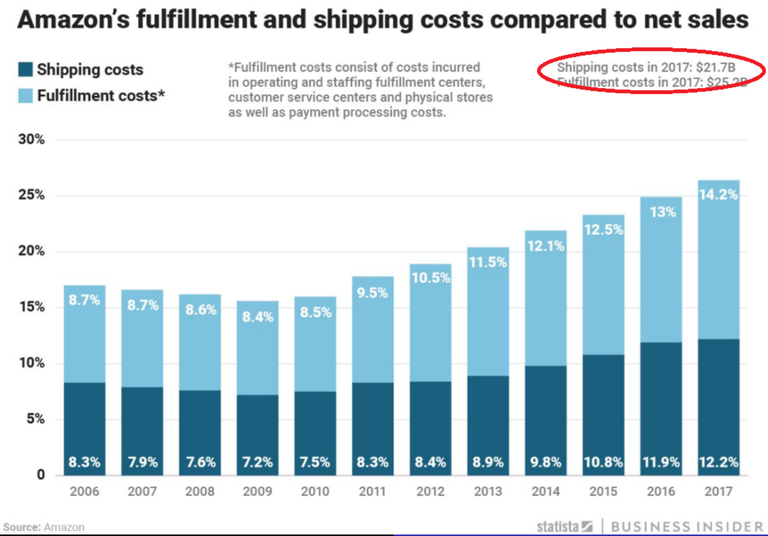
Vehicle Routing Problem (VRP)

One of the most widely studied in Combinatorial Optimization:

- +7,500 works published only in 2018 (Google Scholar), mostly heuristics
- Direct application in the real systems that distribute goods and provide services. Optimized routes can:
    - save a lot of money
    - reduce the environmental impacts of transportation



Example: In 2017, Amazon spent USD 21.7B in shipping, 14.2% of its net sales. As customers demand quicker service, this percentage is growing!



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.

Articles describing new variants appear every week

Outline of the presentation

Part I - Advances on Exact CVRP algorithms

- Review of the advances in the last 15 years

Part II - From CVRP to other classic VRP variants

Part III - A Generic Exact VRP Solver

- VRPSolver model
- Computational results
- Downloading and using VRPSolver

Conclusions: Perspectives on the use of exact algorithms in practice

Part I - Advances on Exact CVRP Algorithms

Capacitated Vehicle Routing Problem (CVRP)

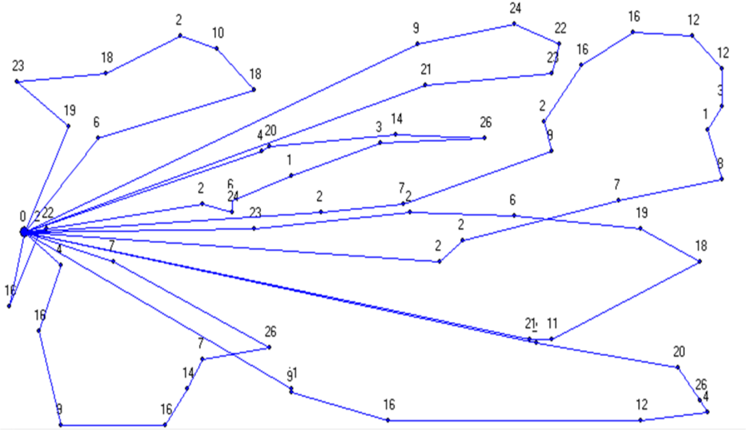
First (Dantzig and Ramser [1959]) and **most basic variant**:

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

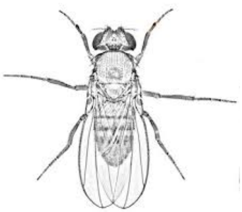
Solution: Set of routes from the depot, respecting the capacities and visiting all customers once; minimizing the total cost.

CVRP instance: $n = 61$, $Q = 100$, indicated demands, Euclidean distances

Optimal solution: $n = 61$, $Q = 100$, indicated demands, Euclidean distances

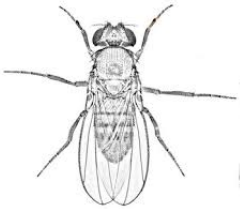


Why we care so much about the fruit fly?



Drosophila Melanogaster

Why we care so much about the fruit fly?



Widely used organism for research in genetics, physiology, and life history evolution. Eight Nobel prizes had been awarded for discoveries using Drosophila

Why we care so much about CVRP?

Common strategy in scientific research:

1. Study the simplest (but still representative!) case of a phenomenon
2. Generalize the discoveries for more complex cases

Historically, several important ideas on routing were first proposed on CVRP and later generalized for many other variants

Edge Formulation (Laporte and Nobert [1983])

Variable x_e indicates how many times e is used.

$$\min \sum_{e \in E} c_e x_e \quad (1)$$

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

$$\sum_{e \in \delta(S)} x_e \geq 2 \lceil \sum_{i \in S} d_i / Q \rceil \qquad \forall\ S \subseteq N, \quad (3)$$

$$x_e \in \{0, 1\} \qquad \forall\ e \in E \setminus \delta(0), \quad (4)$$

$$x_e \in \{0, 1, 2\} \qquad \forall\ e \in \delta(0). \quad (5)$$

Constraints (3) are *Rounded Capacity Cuts*

Branch-and-Cut (BC) Algorithms for CVRP

Extensive research for finding additional families of cuts for the Edge formulation:

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

The dominant approach 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]
- Lysgaard, Letchford, and Eglese [2004]

Best BC results

			LLE04		
Class	Size	#Ins	#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		

Size of the 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

Set Partitioning Formulation (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

$$\min \sum_{r\in\Omega} c_r \lambda_r \tag{6}$$

$$\text{S.t.} \quad \sum_{r\in\Omega} a_{ir}\lambda_r = 1 \qquad \forall\ i \in V_+, \tag{7}$$

$$\lambda_r \in \{0,1\} \qquad \forall\ r \in \Omega. \tag{8}$$

- Exponential number of variables $\Longrightarrow$ Column generation / Branch-and-Price (BP) algorithms
- Pricing elementary routes is strongly NP-hard $\Longrightarrow$ Relax Ω allowing some non-elementary (with cycles) routes

Even with elementary routes, **not a good CVRP formulation!**. Typical root gaps >**3%**, worse than **2%** of BC

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

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

is translated to

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

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

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

- **BCP, the combination of BC and BP can be much better than either of those techniques alone**

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

A crucial issue is the effect of the new dual variables in the pricing:

Robust Cut

Dual variables are translated into costs in the pricing. The subproblem structure does not change.

Non-robust Cut

Dual variables change the structure of pricing. Each added cut makes it harder.

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		

Robust BCP solved all literature instances with up to 134 customers. 3 larger instances remained open: M-n151-k12, M-n200-k16 e M-n200-k17.

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

Baldacci, Christofides and Mingozzi [2008]

- Uses **non-robust cuts**: Strong Capacity and Clique

Root gaps significantly reduced. Several tricks to keep pricing reasonably tractable.

New Key Idea

Instead of branching, algorithm finishes by **enumerating** all routes with reduced cost smaller than the gap. The SPF with only those routes is solved by CPLEX. This saves a lot of time in some instances.

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]

New Key Idea

Introduces *ng*-**routes**, an effective elementarity relaxation:

- For each $i \in V_+$, $NG(i) \subseteq V_+$ contains the *ng*-size closest customers. An *ng*-route can only revisit i if it passes first by a customer j such that $i \notin NG(j)$
- *ng*-size $= 8$ does not make pricing too hard and, in practice, eliminates most cycles

Non-robust Subset Row Cuts (Jepsen et al. [2008]) replace Cliques, smaller impact on pricing

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.

New Key Idea

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]

Uses Subset Row Cuts and *ng*-routes

New Key Idea

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 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 very complex BCP algorithm incorporating elements from **all** previously mentioned works.

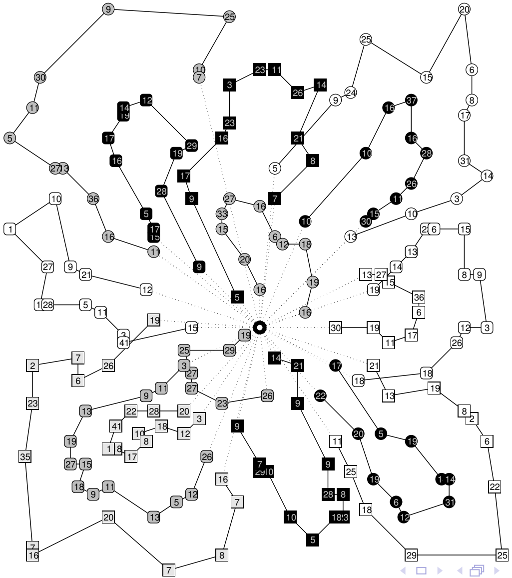
New Key Idea

The concept of **limited memory cuts** for greatly reducing the negative impact of non-robust cuts in the pricing.

M-n151-k12 solved in 3 minutes!

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

Optimal solution M-n200-k16, 10 hours CPU, cost: 1274 (Best heuristic solution: 1278)

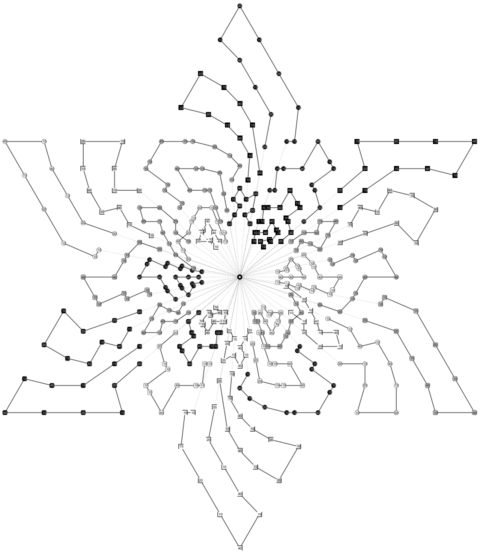


Golden, Wasil, Kelly and Chao [1998] proposed 12 CVRP instances, having from 240 to 483 customers.

- 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



New Instances

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

- Designed to mimic the 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

Results on Instances X

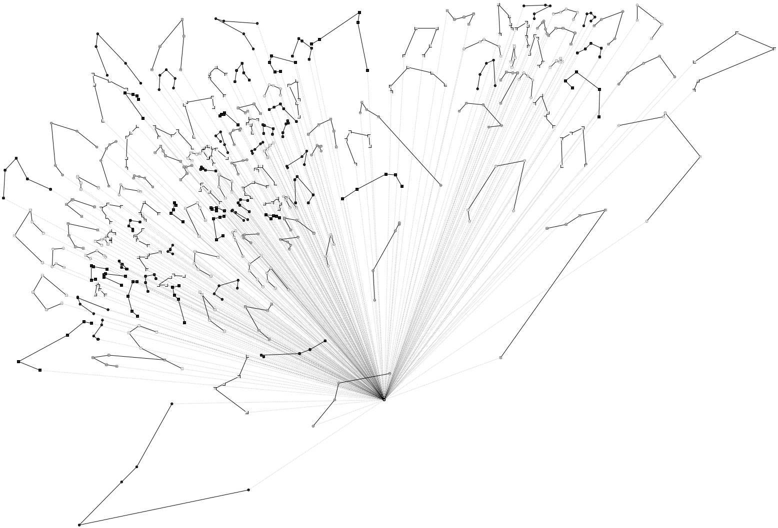
45 out of 100 instances could be solved (sometimes with special parameterization and very long runs):

- $100 \leq n < 200$: 22/22 (100%)
- $200 \leq n < 300$: 16/21 (76%)
- $300 \leq n < 500$: 6/25 (24%)
- $500 \leq n \leq 1000$: 1/32 (3%)

Smallest unsolved: X-n256-k16

Largest solved: X-n655-k131

Optimal solution X-655-k131, 2491 seconds CPU time, cost 106,780



Part II - From CVRP to other classic variants

VRP with Time Windows (VRPTW)

Results on the classic Solomon instances (100 customers)

- All solved, 55/56 at the root node

Results on Gehring-Homberger instances (200 customers)

- 51/60 solved, 27 for the first time

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

Heterogeneous Fleet VRP (HFVRP)

- Solves most instances with up to 200 costumers, two times more than previous methods

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

Capacitated Arc Routing Problem (CARP)

The classic multi-vehicle arc routing

Solves instances with up to 190 required edges, two times larger than previous methods.

- 23/24 Eglese instances, 11 for the first time
- 134/135 other instances from the literature

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

Multi-Trip VRP (MTVRP)

Vehicles can return to the depot for refilling

Solves instances with up to 50 customers, two times larger than previous methods.

R. Paradiso, R. Roberti, D. Laganà, and W Dullaert. An exact solution framework for multi-trip vehicle routing problems with time windows. *Operations Research*, Forthcoming, 2019

A Recent Survey

Other examples of state-of-the-art exact algorithms for VRP variants can be found in this survey:

L. Costa, C. Contardo, and G. Desaulniers. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53:946–985, 2019

The difficulty of creating state-of-the-art exact algorithms for new VRP variants

The BCP algorithms that are achieving the best results for the most VRP variants are very complex:

- Each of the previously mentioned algorithms took months to be constructed, even when built by a team of experts starting from an existing algorithm for another variant!
- There are intricate conceptual issues

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

Previous attempt (Desaulniers et al. [1998])

A framework where several variants are solved by the same BP algorithm over the SPF. The variants are modeled by defining different Ω routes sets, as the solutions of **Resource Constrained Shortest Path (RCSP) Problems**.

$$\min \sum_{r \in \Omega} c_r \lambda_r \tag{9}$$

$$\text{S.t.} \quad \sum_{r \in \Omega} a_{ir} \lambda_r = 1, \qquad \forall\ i \in V_+, \tag{10}$$

$$\lambda_r \in \{0, 1\} \qquad \forall\ r \in \Omega. \tag{11}$$

G. Desaulniers, J. Desrosiers, I. Ioachim, M. Solomon, F. Soumis, and D. Villeneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In *Fleet management and logistics*, pages 57–93. Springer, 1998

VRPSolver

A BCP solver for a generic model that encompasses a wide class of VRPs and even some other kinds of problems

Also defines routes as resource constrained shortest paths, but **allows an arbitrary Master structure (not only Set Partitioning)**. It also incorporates almost all advanced elements found in the recent VRP algorithms, including:

- *ng*-paths,
- Rank-1 cuts with limited memory,
- Rounded Capacity Cuts separators,
- Route enumeration to pools,
- Hierarchical Strong Branching,
- Automatic Dual Stabilization

A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck. A generic exact solver for vehicle routing and related problems. In *Proceedings of the 20th IPCO*, volume 11480, pages 354–369. Springer, 2019

Part III - VRPSolver

The Basic VRPSolver Model

Graphs for Resource Constrained Shortest Path (RCSP) generation (pricing)

Define **directed graphs** $G^k = (V^k, A^k)$, $k \in K$ (one graph per subproblem), having two **special vertices**: v^k_{source}, v^k_{sink} (identical or distinct). For each subproblem also define a set R^k of **resources** having:

- **Arc consumptions**: $q_{a,r} \in R$, $a \in A^k$, $r \in R^k$
- **Accumulated resource consumption interval limits**: $[l_{a,r}, u_{a,r}]$, $a \in A^k$, $r \in R^k$
  - May also be defined on vertices: $[l_{v,r}, u_{v,r}]$, $v \in V^k$, $r \in R^k$ (equivalent to defining $[l_{a,r}, u_{a,r}] = [l_{v,r}, u_{v,r}], \forall a \in \delta^-(v)$)

Let $V = \cup_{k \in K} V^k$ and $A = \cup_{k \in K} A^k$

Graphs for Resource Constrained Shortest Path (RCSP) generation (pricing)

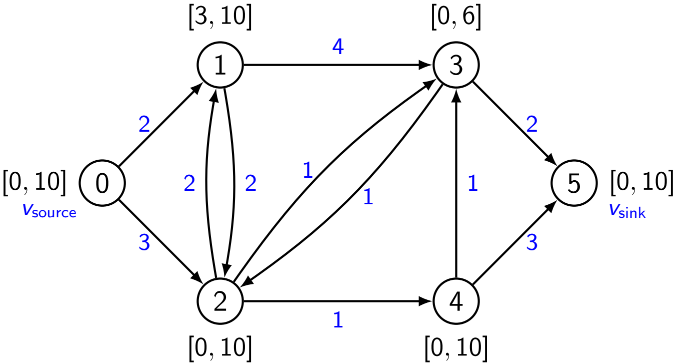
Resource Constrained Path

A path $p = (v^k_{\text{source}} = v_0, a_1, v_1, \ldots, a_{n-1}, v_{n-1}, a_n, v_n = v^k_{\text{sink}})$ over a graph G^k should have $n \geq 1$ arcs, $v_j \neq v^k_{\text{source}}$ and $v_j \neq v^k_{\text{sink}}$, $1 \leq j \leq n-1$, and is feasible if:

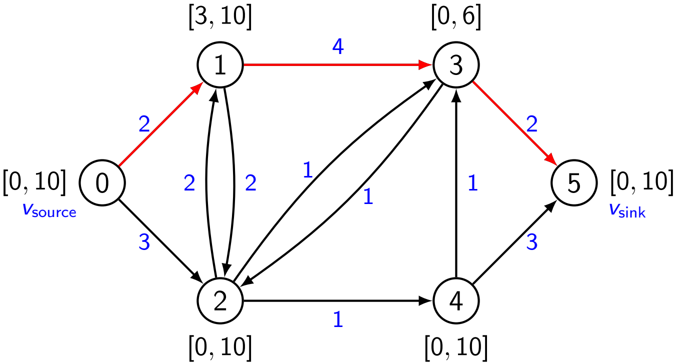
- for every $r \in R^k$, the accumulated resource consumption $S_{j,r}$ at visit j, $0 \leq j \leq n$, where $S_{0,r} = 0$ and $S_{j,r} = \max\{l_{a_j,r}, S_{j-1,r} + q_{a_j,r}\}$, does not exceed $u_{a_j,r}$.

- For each $k \in K$, P^k is the set of all resource constrained paths in G^k
- $P = \cup_{k \in K} P^k$

Example

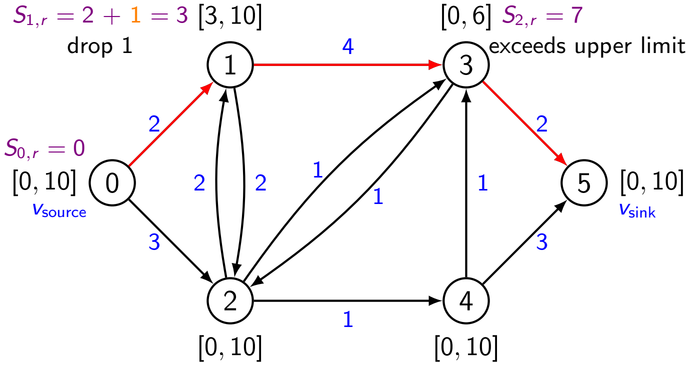


Example



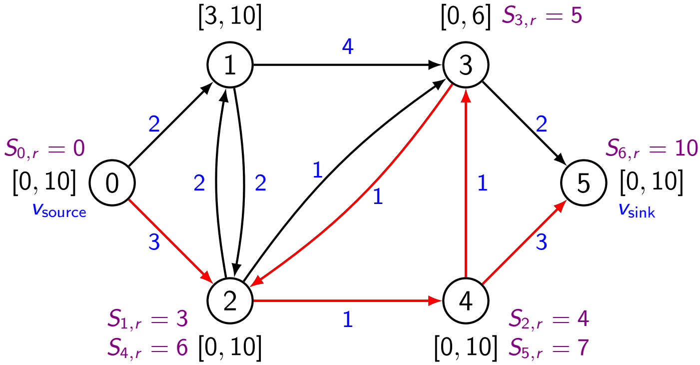
Path 1: 0 – 1 – 3 – 5

Example



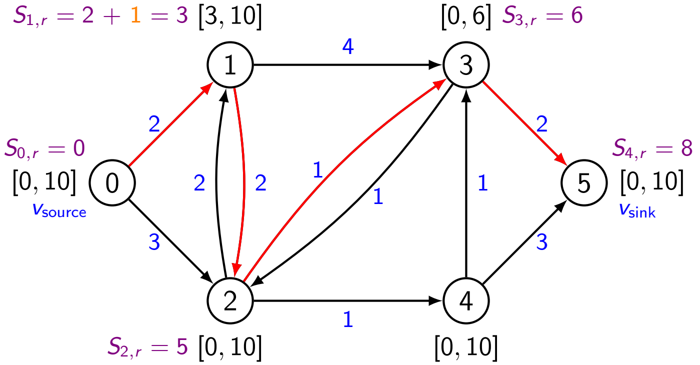
Path 1: 0 – 1 – 3 – 5 (Infeasible)

Example



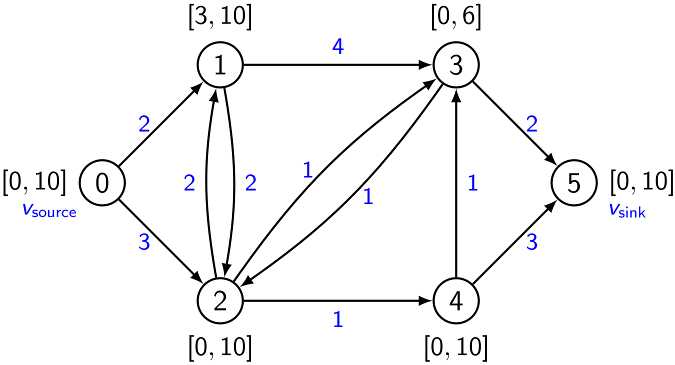
Path 1: 0 – 1 – 3 – 5 (Infeasible)
Path 2: 0 – 2 – 4 – 3 – 2 – 4 – 5

Example



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

Example



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)

Main vs Secondary resources

Set R^k is partitioned (by the modeler) into **main resources** R^k_M and **secondary resources** R^k_N

Main resources (R^k_M)

- It is mandatory the existence of at least one main resource. Maximum number of main resources: 2
- Consumptions should be non-negative
- Cycles with zero consumption on all main resources should not exist

The main resources make sure that sets P^k are finite, by preventing the infinite repetition of a subpath

Additional resource features

Disposable vs Non-disposable resources

- **Disposable** (default): it is possible to drop resource in order to satisfy the lower limit on accumulated consumption
- **Non-disposable**: it is not possible to drop resources

Defining resources as non-disposable make the RCSP much harder and should be avoided unless it is really needed for the model

Binary resources

All consumptions are 1, 0 or -1, all intervals are [0,0], [0,1] or [1,1].

Special implementation in bitsets

Basic Model: Variables and Mappings

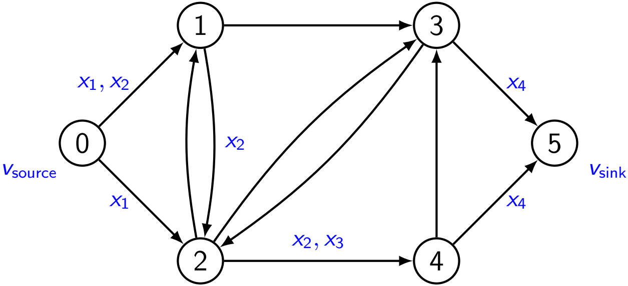
After defining the graphs, define continuous and/or integer variables:

1. 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.
2. Additional (non-mapped) y variables

Mapping

A "non-standard" concept used in VRPSolver to link the Master problem to the subproblems.

Mapping x_j, $1 \leq j \leq 4$ ($n_1 = 4$), variables



$M(x_1) = \{(0,1),(0,2)\}, M(x_2) = \{(0,1),(1,2),(2,4)\},$
$M(x_3) = \{(2,4)\}, M(x_4) = \{(3,5),(4,5)\}.$

$M^{-1}((0,1)) = \{x_1, x_2\}, M^{-1}((1,3)) = \{\emptyset\},$
$M^{-1}((2,4)) = \{x_2, x_3\}, \ldots, M^{-1}((4,5)) = \{x_4\}.$

Basic 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
L^k, U^k: given bounds on the number of paths from each graph

$$
\begin{aligned}
\text{Min} \quad & \sum_{j=1}^{n_1} c_j x_j + \sum_{s=1}^{n_2} f_s y_s && && \text{(12a)}\\
\text{S.t.} \quad & \sum_{j=1}^{n_1} \alpha_{ij} x_j + \sum_{s=1}^{n_2} \beta_{is} y_s \geq d_i, && i = 1, \ldots, m, && \text{(12b)}\\
& x_j = \sum_{k \in K} \sum_{p \in P^k} \left(\sum_{a \in M(x_j)} h_a^p \right) \lambda_p, && j = 1 \ldots, n_1, && \text{(12c)}\\
& L^k \leq \sum_{p \in P^k} \lambda_p \leq U^k, && k \in K, && \text{(12d)}\\
& \lambda_p \in Z_+, && p \in P, && \text{(12e)}\\
& x_j \in Z_+, \; y_s \in Z_+ && j = 1, \ldots, \bar{n}_1, \; s = 1, \ldots, \bar{n}_2. &&
\end{aligned}
$$

Basic Model: Formulation

Eliminating the x variables and relaxing the integrality constraints:

$$\text{Min} \quad \sum_{k \in K} \sum_{p \in P^k} \left(\sum_{j=1}^{n_1} c_j \sum_{a \in M(x_j)} h_a^p \right) \lambda_p + \sum_{s=1}^{n_2} f_s y_s \tag{13a}$$

$$\text{S.t.} \quad \sum_{k \in K} \sum_{p \in P^k} \left(\sum_{j=1}^{n_1} \alpha_{ij} \sum_{a \in M(x_j)} h_a^p \right) \lambda_p + \sum_{s=1}^{n_2} \beta_{is} y_s \geq d_i, \qquad i = 1, \ldots, m, \tag{13b}$$

$$L^k \leq \sum_{p \in P^k} \lambda_p \leq U^k, \qquad k \in K, \tag{13c}$$

$$\lambda_p \geq 0, \qquad p \in P. \tag{13d}$$

Basic Model: Formulation

Eliminating the x variables and relaxing the integrality constraints:

$$\text{Min} \quad \sum_{k\in K}\sum_{p\in P^k}\left(\sum_{j=1}^{n_1} c_j \sum_{a\in M(x_j)} h_a^p\right)\lambda_p + \sum_{s=1}^{n_2} f_s y_s \tag{13a}$$

$$\text{S.t.} \quad \sum_{k\in K}\sum_{p\in P^k}\left(\sum_{j=1}^{n_1} \alpha_{ij} \sum_{a\in M(x_j)} h_a^p\right)\lambda_p + \sum_{s=1}^{n_2} \beta_{is} y_s \geq d_i, \qquad i = 1,\ldots,m, \tag{13b}$$

$$L^k \leq \sum_{p\in P^k} \lambda_p \leq U^k, \qquad k \in K, \tag{13c}$$

$$\lambda_p \geq 0, \qquad p \in P. \tag{13d}$$

Master LP (13) is solved by column generation

Basic Model: Formulation

Let π_i, $1 \leq i \leq m$, be the dual variables of Constraints (13b), ν_+^k and ν_-^k, $k \in K$, the dual variables of Constraints (13c).

The reduced cost of an arc $a \in A$ is:

$$\bar{c}_a = \sum_{j \in M^{-1}(a)} c_j - \sum_{i=1}^{m} \sum_{j \in M^{-1}(a)} \alpha_{ij} \pi_i.$$

The reduced cost of a path $p \in P^k$ is:

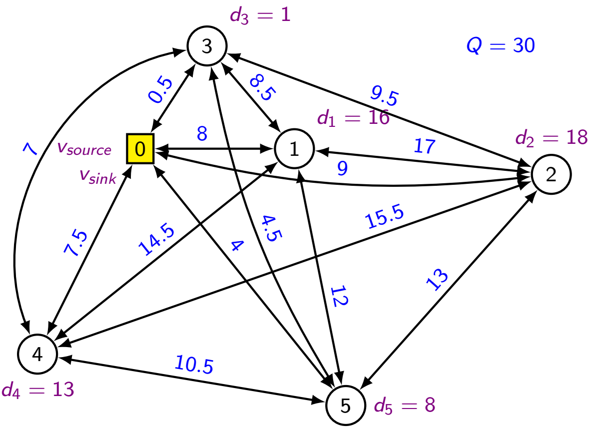
$$\bar{c}(p) = \sum_{j=1}^{n} \bar{c}_{a_j} - \nu_+^k - \nu_-^k.$$

So, the pricing subproblems correspond to finding, for each $k \in K$, a path $p \in P^k$ with minimum reduced cost.

Example: CVRP Model: Graph

Single graph

$G = (V, A)$, $A = \{(i,j),(j,i) : \{i,j\} \in E\}$, $v_{\text{source}} = v_{\text{sink}} = 0$;
$R = R_M = \{1\}$; $q_{a,1} = (d_i + d_j)/2$, $a = (i,j) \in A$ (define $d_0 = 0$);
$l_{i,1} = 0$, $u_{i,1} = Q$, $i \in V$;



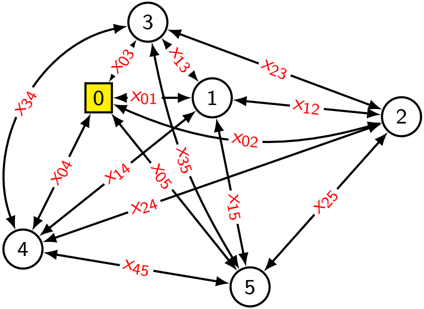
Example: CVRP Model: Formulation + Mapping

Integer variables x_e, $e \in E$.

$$\text{Min} \quad \sum_{e \in E} c_e x_e \tag{14a}$$

$$\text{S.t.} \quad \sum_{e \in \delta(i)} x_e = 2, \qquad i \in V_+. \tag{14b}$$

$L = 0,\ U = n;\ M(x_e) = \{(i,j),(j,i)\},\ e = \{i,j\} \in E.$



Example: Set of Paths

In this small CVRP example there are 18 paths (only counting elementary ones and only taking the cheapest path among those that visit the same customers)

λ_1: $0 \longrightarrow 3 \longrightarrow 0$

λ_2: $0 \longrightarrow 3 \longrightarrow 2 \longrightarrow 5 \longrightarrow 0$

λ_3: $0 \longrightarrow 2 \longrightarrow 5 \longrightarrow 0$

λ_4: $0 \longrightarrow 4 \longrightarrow 5 \longrightarrow 3 \longrightarrow 0$

λ_5: $0 \longrightarrow 3 \longrightarrow 2 \longrightarrow 0$

λ_6: $0 \longrightarrow 5 \longrightarrow 1 \longrightarrow 3 \longrightarrow 0$

λ_7: $0 \longrightarrow 1 \longrightarrow 3 \longrightarrow 4 \longrightarrow 0$

λ_8: $0 \longrightarrow 5 \longrightarrow 3 \longrightarrow 0$

λ_9: $0 \longrightarrow 4 \longrightarrow 5 \longrightarrow 0$

λ_{10}: $0 \longrightarrow 2 \longrightarrow 0$

λ_{11}: $0 \longrightarrow 4 \longrightarrow 1 \longrightarrow 3 \longrightarrow 0$

λ_{12}: $0 \longrightarrow 5 \longrightarrow 1 \longrightarrow 0$

λ_{13}: $0 \longrightarrow 3 \longrightarrow 4 \longrightarrow 0$

λ_{14}: $0 \longrightarrow 4 \longrightarrow 1 \longrightarrow 0$

λ_{15}: $0 \longrightarrow 5 \longrightarrow 0$

λ_{16}: $0 \longrightarrow 4 \longrightarrow 0$

λ_{17}: $0 \longrightarrow 3 \longrightarrow 1 \longrightarrow 0$

λ_{18}: $0 \longrightarrow 1 \longrightarrow 0$

Example: Resulting Complete MIP Formulation

$$
\begin{array}{ll}
\text{Min} & 21x_{01} + 58x_{02} + 17x_{03} + 33x_{04} + 49x_{05} + 37x_{12} + 19x_{13} + \ldots \\
\text{S.t.} & x_{01} + x_{12} + x_{13} + x_{14} + x_{15} = 2, \\
& x_{02} + x_{12} + x_{23} + x_{24} + x_{25} = 2, \\
& x_{03} + x_{13} + x_{23} + x_{34} + x_{35} = 2, \\
& x_{04} + x_{14} + x_{24} + x_{34} + x_{45} = 2, \\
& x_{05} + x_{15} + x_{25} + x_{35} + x_{45} = 2, \\
& x_{01} = \lambda_7 + \lambda_{12} + \lambda_{14} + \lambda_{17} + 2\lambda_{18}, \\
& x_{02} = \lambda_3 + \lambda_5 + 2\lambda_{10}, \\
& \vdots \\
& x_{45} = \lambda_4 + \lambda_9, \\
& 0 \leq \lambda_1 + \lambda_2 + \lambda_3 + \cdots + \lambda_{18} \leq 5, \\
& \lambda \in Z_+
\end{array}
$$

Substituting the x variables and relaxing integrality

$$
\begin{aligned}
\text{Min} \quad & 34\lambda_1 + 166\lambda_2 + 154\lambda_3 + 149\lambda_4 + 128\lambda_5 + 126\lambda_6 + 124\lambda_7 + \\
& 124\lambda_8 + 123\lambda_9 + 116\lambda_{10} + 114\lambda_{11} + 111\lambda_{12} + 101\lambda_{13} + 99\lambda_{14} + \\
& 98\lambda_{15} + 66\lambda_{16} + 57\lambda_{17} + 42\lambda_{18} \\
\text{S.t.} \quad & 2\lambda_6 + 2\lambda_7 + 2\lambda_{11} + 2\lambda_{12} + 2\lambda_{14} + 2\lambda_{17} + 2\lambda_{18} = 2 \\
& 2\lambda_2 + 2\lambda_3 + 2\lambda_5 + 2\lambda_{10} = 2 \\
& 2\lambda_1 + 2\lambda_2 + 2\lambda_4 + 2\lambda_5 + 2\lambda_6 + 2\lambda_7 + 2\lambda_8 + 2\lambda_{11} + 2\lambda_{13} + 2\lambda_{17} = 2 \\
& 2\lambda_4 + 2\lambda_7 + 2\lambda_9 + 2\lambda_{11} + 2\lambda_{13} + 2\lambda_{14} + 2\lambda_{16} = 2 \\
& 2\lambda_2 + 2\lambda_3 + 2\lambda_4 + 2\lambda_6 + 2\lambda_8 + 2\lambda_9 + 2\lambda_{12} + 2\lambda_{15} = 2 \\
& 0 \le \lambda_1 + \lambda_2 + \lambda_3 + \cdots + \lambda_{18} \le 5, \\
& \lambda \ge 0
\end{aligned}
$$

Equivalent to the classic Set Partitioning Formulation. Column generation is performed having a Resource Constrained Shortest Path in the defined graph as pricing subproblem.

Including Advanced Elements: Packing Sets

Collection of Packing Sets (arc version)

Let $\mathcal{B} \subset 2^A$ be a collection of mutually disjoint subsets of A such that the constraints:

$$\sum_{a \in B} \sum_{p \in P} h_a^p \lambda_p \leq 1, \quad B \in \mathcal{B}, \tag{15}$$

are satisfied by at least one optimal solution. In those conditions, we say that the elements of $\mathcal{B}$ are *packing sets*.

Packing sets generalize customers in the classical VRP variants, that can only be visited once

The definition of a proper $\mathcal{B}$ is part of the modeling

Collection of Packing Sets (vertex version)

In many cases it is more natural to define packing sets on vertices. Let $\mathcal{B}^{\mathcal{V}} \subset 2^V$ be a collection of mutually disjoint subsets of V such that the constraints:

$$\sum_{p \in P} \left(\sum_{v \in B} h_v^p \right) \lambda_p \leq 1, \quad B \in \mathcal{B}^{\mathcal{V}},$$

are satisfied by at least one optimal solution. In those conditions, we say that the elements of $\mathcal{B}^{\mathcal{V}}$ are *packing sets on vertices*.

Packing sets generalize customers in the classical VRP variants, that can only be visited once

The definition of a proper $\mathcal{B}^{\mathcal{V}}$ is part of the modeling

Generalizing ng-routes

The ng-routes (Baldacci et al. [2011a]) obtain a good compromise between bound quality and pricing difficulty.

Generalized ng-paths

For each arc $a \in A$, let $NG(a) \subseteq \mathcal{B}$ denote the *ng*-set of a. An *ng*-path may visit a given packing set B a second time only after passing by an arc a such that $B \notin NG(a)$.

Generalizing Limited Memory Rank-1 Cuts

The Rank-1 Cuts (R1Cs) (Pecin et al. [2017c], Bulhoes et al. [2018a]) are a generalization of the Subset Row Cuts by Jepsen et al. [2008]. They are further generalized as follows.

Given non-negative multipliers ρ_B for each $B \in \mathcal{B}$, a Chvátal-Gomory rounding of Constraints (59) yields:

$$\sum_{p \in P} \left\lfloor \sum_{B \in \mathcal{B}} \rho_B \sum_{a \in B} h_a^p \right\rfloor \lambda_p \leq \left\lfloor \sum_{B \in \mathcal{B}} \rho_B \right\rfloor . \tag{16}$$

Limited-memory avoids excessive impact in the pricing

Generalizing Path Enumeration

If gaps are sufficiently small, all paths without more than one arc in the same packing set can be enumerated to a pool, that is used to perform pricing by inspection.

Generalizes Baldacci et al. [2008] and Contardo and Martinelli [2014b].

Some Modeling Examples

Heterogeneous Fleet Vehicle Routing Problem (HFVRP)

- Undirected graph $G' = (V, E)$, $V = \{0, \ldots, n\}$, 0 is the depot, $V_+ = \{1, \ldots, n\}$ are the customers; positive demand d_i, $i \in V_+$; set of vehicle types $K = \{1, \ldots, m\}$; number of available vehicles u^k, $k \in K$; edge costs c_e^k, $e \in E$, $k \in K$; vehicle type capacity Q^k, $k \in K$.
- Find a minimum cost set of routes visiting all customers and such that the sum of the demands of the customers in a route does not exceed its vehicle type capacity. The number of routes for a vehicle type should not exceed its availability.

VRPSolver Model for Heterogeneous Fleet Vehicle Routing Problem (HFVRP)

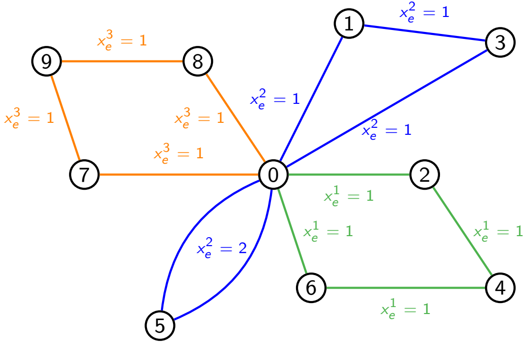
Graphs G^k

$G^k = (V^k, A^k)$, $V^k = \{v_0^k, \ldots, v_n^k\}$,
$A^k = \{(v_i^k, v_j^k), (v_j^k, v_i^k) : \{i, j\} \in E\}$;
$v^k_{\text{source}} = v^k_{\text{sink}} = v_0^k$, $k \in K$;
$R^k = R^k_M = \{r^k\}$;
$q_{a,1} = (d_i + d_j)/2$, $a = (v_i^k, v_j^k) \in A^k$, $k \in K$ (define $d_0 = 0$);
$l_{v_i^k, r^k} = 0, u_{v_i^k, r^k} = Q^k, v_i^k \in V^k$, $k \in K$.

Those graphs only differ by the value of Q^k, even if the costs are vehicle-dependent.

VRPSolver Model for Heterogeneous Fleet Vehicle Routing Problem (HFVRP)

Integer variables x_e^k, $e \in E$, $k \in K$ - how many times edge e is used in a route of a type k vehicle.



VRPSolver Model for Heterogeneous Fleet Vehicle Routing Problem (HFVRP)

Formulation + Mapping

Integer variables x_e^k, $e \in E$, $k \in K$.

$$\begin{aligned} \text{Min} \quad & \textstyle\sum_{k \in K} \sum_{e \in E} c_e^k x_e^k && \text{(17a)} \\ \text{S.t.} \quad & \textstyle\sum_{k \in K} \sum_{e \in \delta(i)} x_e^k = 2, \qquad i \in V_+; && \text{(17b)} \end{aligned}$$

$L^k = 0$, $U^k = u^k$; $M(x_e^k) = \{(v_i^k, v_j^k), (v_j^k, v_i^k)\}$, $e = \{i, j\} \in E$, $k \in K$.

Packing Sets

$\mathcal{B}^{\mathcal{V}} = \cup_{i \in V_+} \{\{v_i^k : k \in K\}\}$

VRPSolver Model for Heterogeneous Fleet Vehicle Routing Problem (HFVRP)

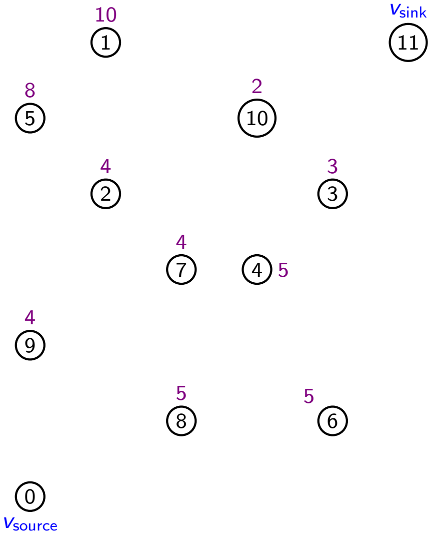
$$
\begin{aligned}
\text{Min} \quad & \sum_{k \in K} \sum_{e \in E} c_e^k x_e^k && && (18a)\\
\text{S.t.} \quad & \sum_{k \in K} \sum_{e \in \delta(i)} x_e^k = 2, && i \in V_+; && (18b)\\
& x_e^k = \sum_{p \in P^k} h_e^p \lambda_p, && e \in E, k \in K; && (18c)\\
& 0 \leq \sum_{p \in P^k} \lambda_p \leq u^k, && k \in K; && (18d)\\
& \lambda_p \in Z_+, && p \in P; && (18e)\\
& x_e^k \in Z_+, && e \in E, k \in K. && (18f)
\end{aligned}
$$

h_e^p = how many times edge e is used in p

Team Orienteering Problem (TOP)

- Directed graph $G = (V, A)$, $V = \{0, \ldots, n+1\}$, 0 and $n+1$ are the initial and final depots, respectively, $V_+ = \{1, \ldots, n\}$ are the customers; positive travel time t_a, $a \in A$; profit p_i, $i \in V_+$; maximum route duration T; and fleet size F.
- Find a set of at most F routes, each one starting at 0, ending at $n+1$ and not exceeding the maximum route duration, that visit each customer at most once and maximize the total profit of the visited customers.

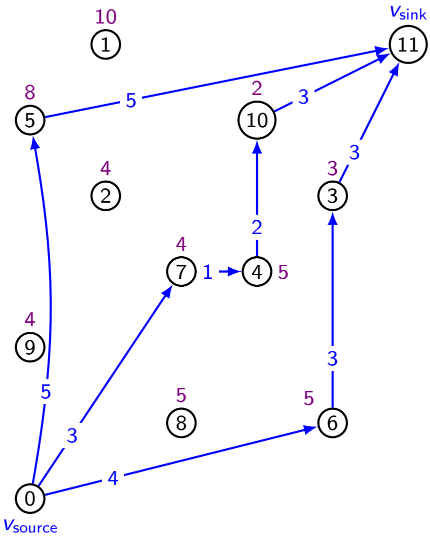
Team Orienteering Problem (TOP)



- Route duration: $T = 10$
- Fleet size: $F = 3$
- Number of customers: $n = 10$

Team Orienteering Problem (TOP)

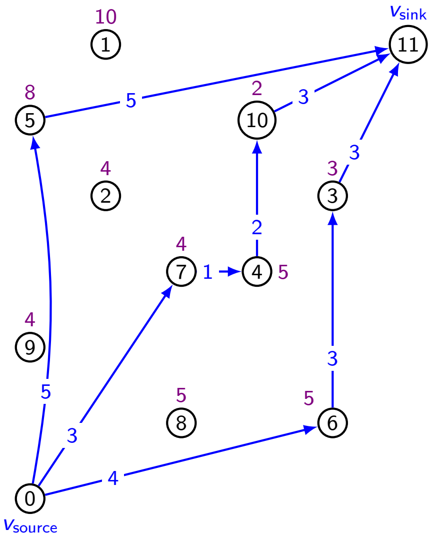
Some feasible paths



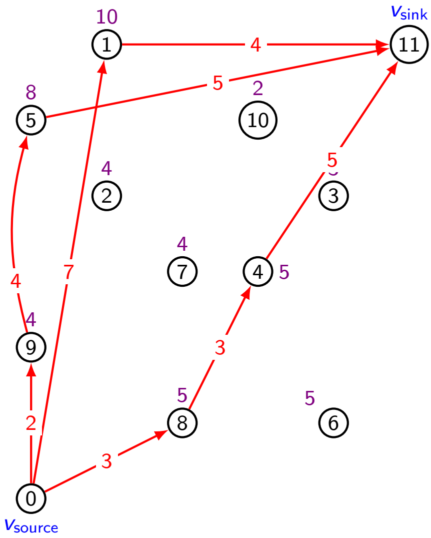
- Route duration: $T = 10$
- Fleet size: $F = 3$
- Number of customers: $n = 10$

Team Orienteering Problem (TOP)

Some feasible paths



Some infeasible paths



VRPSolver Model for Team Orienteering Problem (TOP)

Single graph

$G = (V, A)$; $v_{\text{source}} = 0, v_{\text{sink}} = n + 1$; $R = R^M = \{1\}$; $q_{a,1} = t_a$, $a = (i, j) \in A$; $l_{i,1} = 0, u_{i,1} = T, i \in V$.

Formulation + Mapping

Integer variables x_a, $a \in A$ and binary variables y_i, $i \in V_+$. The y variables, that indicate which customers are visited, are not mapped to any arc.

$$\text{Min} \quad -\sum_{i \in V_+} p_i y_i \tag{19a}$$

$$\text{S.t.} \quad \sum_{a \in \delta^-(i)} x_a = y_i, \qquad i \in V_+; \tag{19b}$$

$L = 0$, $U = F$; $M(x_a) = \{a\}$, $a \in A$;

Packing Sets

$\mathcal{B}^{\mathcal{V}} = \cup_{i \in V_+} \{\{i\}\}$

VRPSolver Model for Team Orienteering Problem (TOP)

$$\text{Min} \quad -\sum_{i \in V_+} p_i y_i \tag{20a}$$

$$\text{S.t.} \quad \sum_{a \in \delta^-(i)} x_a = y_i, \qquad i \in V_+; \tag{20b}$$

$$x_a = \sum_{p \in P} h_a^p \lambda_p, \qquad a \in A; \tag{20c}$$

$$0 \leq \sum_{p \in P} \lambda_p \leq F, \tag{20d}$$

$$\lambda, x \in Z_+, \tag{20e}$$

$$0 \leq y \leq 1. \tag{20f}$$

Substituting x and relaxing the integrality:

$$\text{Min} \quad -\sum_{i \in V_+} p_i y_i \tag{21a}$$

$$\text{S.t.} \quad \sum_{p \in P} \left(\sum_{a \in \delta^-(i)} h_a^p \right) \lambda_p = y_i, \qquad i \in V_+; \tag{21b}$$

$$0 \leq \sum_{p \in P} \lambda_p \leq F, \tag{21c}$$

$$\lambda \geq 0. \tag{21d}$$

$$0 \leq y \leq 1. \tag{21e}$$

Computational Results

Computational Experiments

- Solver optimization algorithms coded in C++ over BaPCod package (Vanderbeck et al. [2018])
- IBM CPLEX 12.9 used as LP solver
- Experiments on Xeon E5-2680 v3 2.50 GHz processors
- Models are defined using either a C++ interface or 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.	Gen. BCP	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. [2018b]	0 (1h)	Archetti et al. [2013]
VRPSL	Bulhoes	180	2h	**159 (16m)**	49 (90m)	Bulhoes et al. [2018b]	—	
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: Generic solver vs best specific solvers on 13 problems.

Additional Experiments on Some unsolved CVRP instances

Instance	Prev. BKS	Root LB	Nodes	Total Time	OPT
X-n284-k15	20226	20168	940	11.0 days	**<u>20215</u>**
X-n322-k28	29834	29731	1197	5.6 days	**29834**
X-n344-k43	42056	41939	2791	11.6 days	**<u>42050</u>**
X-n393-k38	38260	38194	1331	5.8 days	**38260**
X-n469-k138	221909	221585	8964	15.2 days	**<u>221824</u>**
X-n548-k50	86701	86650	337	2.0 days	**<u>86700</u>**

- Long runs, parameters calibrated per instance

Downloading and using the VRP Solver

VRPSolver available over Julia–JuMP

Available for academic use (vrpsolver.math.u-bordeaux.fr):

- Algorithms are bundled in a single pre-compiled docker
- Julia–JuMP user interface for modeling (open source), including several demos

Modeling on VRPSolver

1. Modeling a typical VRP variant requires less than 100 lines of Julia/JuMP code (not counting input/output). A user can build a working solver for a new variant in 1 day
2. Computer experiments for parameter tuning may be required for a better performance
3. Separation routines for problem specific cuts may be needed for less standard VRP variants. Examples where sophisticated specific cuts had to be devised for good performance:
    - Location-Routing Problem (Liguori et al. [2019])
    - Two-echelon CVRP (Marques et al. [2019])
4. VRPSolver has a built-in diving primal heuristic (Sadykov et al. [2018]). But it does not always work well. In those cases, a good performance depends on having some heuristic for providing external upper bounds.

Room for Creative Modeling

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

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

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

Conclusions and Perspectives

Future of VRPSolver $\longrightarrow$ Coluna

It is expected that future versions of VRPSolver will be built on top of **Coluna** (https://github.com/atoptima/Coluna.jl), an open source collaborative framework for branch-cut-and-price algorithms, coded in Julia, that is under development.

- No changes in VRPSolver user interface
- Migration will happen when Coluna has sufficient performance

VRP solvers

As expected from the economic importance of routing, there are scores of commercial and proprietary VRP solvers. The vast majority of them are based on heuristics.

There are also several free heuristics VRP solvers that can handle a significant number of variants, including:

- Google OR-Tools
- VRP Spreadsheet Solver (Güneș Erdoğan)
- LKH (Keld Helsgaun)
- OptaPlanner
- VROOM
- JSprit

Impact of Exact Algorithms in Practical VRP

Historically, exact solvers were rarely directly used in practical routing

1. Existing algorithms could not solve realistic-sized instances in reasonable times
    - Now most instances of the most classic VRPs with up to 200 customers can be solved
    - More importantly, instances with up to 100 customers can often 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 very hard task
    - Customizable codes with state-of-the-art performance will start to be available

We expect that exact algorithms will be much more used by VRP practitioners

Impact of Exact Algorithms in Practical VRP

However, exact solvers also have significant indirect practical impact:

1. Benchmarking heuristics
    - Knowing the optimal solution values for some instances with 200 costumers (even if it takes days to find them) allows for a much better assessment of the quality of a heuristic
    - Remark that if your real problem is slightly different from those already studied in academy, it is not possible to obtain solution values from the literature
2. Improving heuristics
    - The actual optimal solutions may give precious hints on how to improve, say, a local search algorithm.

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.

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.

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. Bulhoes, A. Pessoa, F. Protti, and E. Uchoa. On the complete set packing and set partitioning polytopes: Properties and rank 1 facets. *Operations Research Letters*, 46(4):389–392, 2018a.

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, 2018b.

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.

L. Costa, C. Contardo, and G. Desaulniers. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53:946–985, 2019.

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

G. Desaulniers, J. Desrosiers, I. Ioachim, M. Solomon, F. Soumis, and D. Villeneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In *Fleet management and logistics*, pages 57–93. Springer, 1998.

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.

R. Fukasawa, H. Longo, J. Lysgaard, M. Poggi, M. Reis, E. Uchoa, 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, R. Sadykov, and E. Uchoa. A branch-and-cut-and-price algorithm for the capacitated location-routing problem. In *Proceedings of the 10th TRISTAN*, pages 1–4, 2019.

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.

Guillaume Marques, Ruslan Sadykov, Jean-Christophe Deschamps, and Rémy Dupas. An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Computer & Operations Research*, 114, 2019.

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

R. Paradiso, R. Roberti, D. Laganà, and W Dullaert. An exact solution framework for multi-trip vehicle routing problems with time windows. *Operations Research*, Forthcoming, 2019.

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

D. Pecin, A. Pessoa, M. Poggi, and E. Uchoa. 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 E. Uchoa. 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.

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

A. Pessoa, R. Sadykov, and E. Uchoa. 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. In *Proceedings of the 20th IPCO*, volume 11480, pages 354–369. Springer, 2019.

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.

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

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

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*, (Forthcoming), 2018.

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.

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.