Tutorial: Modern Branch-and-Cut-and-Price for Vehicle Routing Problems

Ruslan Sadykov

Inria Bordeaux, France

Université Bordeaux, France

INOC 2019, June 14
Avignon, France

Plan of the talk

Introduction

Basic Branch-Cut-and-Price

Improvement Techniques

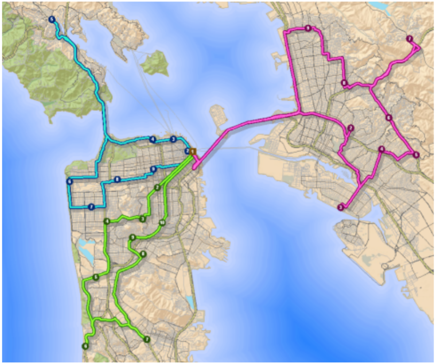
Generic BCP solver

Some Results and Perspectives

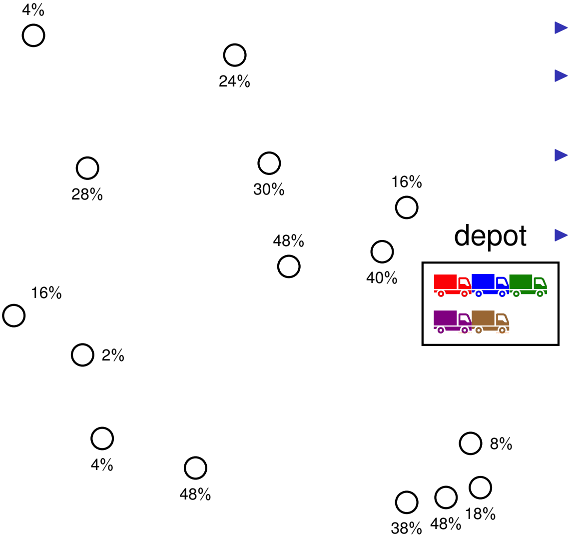
Contents

Vehicle Routing Problem (VRP)

- One of the most widely investigated optimization problems.
- Google Scholar finds +7,500 works published in 2018 (849 contain both “vehicle” and “routing” in the title)
- Direct application in the real-world systems that distribute goods and provide services

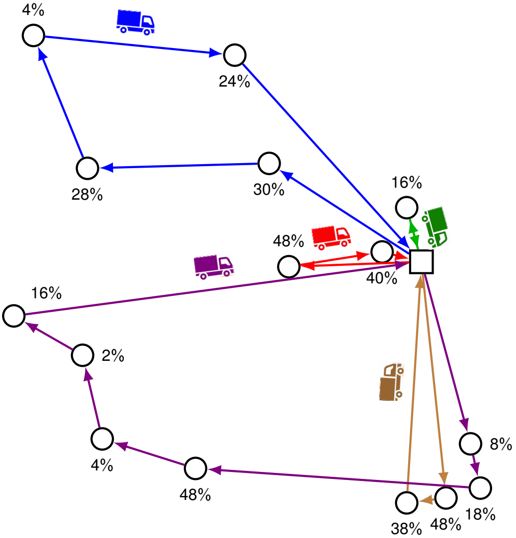


Capacitated Vehicle Routing Problem (CVRP)



- Depot
- K identical vehicles of capacity Q
- Clients $i \in V$ with demand d_i
- Matrix c of travelling costs

Capacitated Vehicle Routing Problem (CVRP)



- Depot
- K identical vehicles of capacity Q
- Clients $i \in V$ with demand d_i
- Matrix c of travelling costs

Minimize the total travelling cost

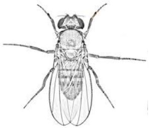
- such that every client is served
- total demand of clients served by the same vehicle does not exceed its capacity

Why we care so much about CVRP?

First [Dantzig and Ramser, 1959] and the most basic VRP variant.

Common strategy in scientific research

- Study the simplest (bust still representative!) case of a phenomenon
- Generalize the discoveries for more complex cases



Drosophila Melanogaster

Hundreds of VRP variants

Vehicle capacities, time windows, heterogeneous fleet, multiple depots, split delivery, pickup and delivery, backhauling, optional customer service, arc routing, alternative delivery options, service levels, etc, etc

Some history

- [Balinski and Quandt, 1964] set-partitioning formulation for CVRP
- [Laporte and Nobert, 1983] MIP formulation with edge variables, rounded capacity cuts, and branch-and-bound
- [Desrochers et al., 1992] first branch-and-price
- [Lysgaard et al., 2004] best branch-and-cut algorithm
- [Fukasawa et al., 2006] robust branch-cut-and-price
- [Baldacci et al., 2008] enumeration technique
- [Jepsen et al., 2008] (non-robust) subset-row cuts
- [Baldacci et al., 2011b] *ng*-route relaxation
- [Pecin et al., 2017b] limited-memory technique, best branch-cut-and-price
- [Poggi and Uchoa, 2014] [Costa et al., 2019] recent surveys

Some motivation

12th DIMACS Implementation

Challenge: Vehicle Routing Problems

In Memory of David S. Johnson

Competition rules will be presented in July

Contents

Resource constrained paths

- Complete directed graph $G = (V^0, A)$, $V^0 = \{0\} \cup V$.
- Capacity resource
- Resource consumption of arc $a = (i, j) \in A$ is d_j, $d_0 = 0$.
- Accumulated resource consumption interval for $v \in V^0$ is $[0, Q]$.

A set of feasible routes is modelled by set P of paths in G from node 0 to node 0 such that for each path $p \in P$

- each node $v \in V$ is visited at most once.
- accumulated resource consumption for every node v visited by p is within given intervals $[0, Q]$.

Set-partitioning formulation

- Variable x_a — arc $a \in A$ is used in the solution or not
- Variable λ_p — path $p \in P$ is used in the solution or not
- $h_a^p = 1$ if and only if path p contains arc a, otherwise 0
- $\delta^-(v)$ — set of arcs in A incoming to $v \in V$

$$
\begin{aligned}
\text{Min} \quad & \sum_{a \in A} c_a x_a \\
\text{S.t.} \quad & \sum_{a \in \delta^-(v)} x_a = 1, && v \in V, \\
& Bx \leq b, \\
& x_a = \sum_{p \in P} h_a^p \lambda_p, && a \in A, \\
& \sum_{p \in P^k} \lambda_p \leq K, \\
& x_a \in \{0, 1\}, && a \in A, \\
& \lambda_p \in \{0, 1\}, && p \in P.
\end{aligned}
$$

Set-partitioning formulation

- Variable x_a — arc $a \in A$ is used in the solution or not
- Variable λ_p — path $p \in P$ is used in the solution or not
- $h_a^p = 1$ if and only if path p contains arc a, otherwise 0
- $\delta^-(v)$ — set of arcs in A incoming to $v \in V$

$$
\begin{aligned}
\text{Min} \quad & \sum_{a \in A} c_a x_a \\
\text{S.t.} \quad & \sum_{a \in \delta^-(v)} x_a = 1, && v \in V, \\
& Bx \leq b, \\
& x_a = \sum_{p \in P} h_a^p \lambda_p, && a \in A, && (\pi_a) \\
& \sum_{p \in P^k} \lambda_p \leq K, && && (\mu) \\
& 0 \leq x_a \leq 1, && a \in A, \\
& 0 \leq \lambda_p \leq 1, && p \in P.
\end{aligned}
$$

Column and cut generation

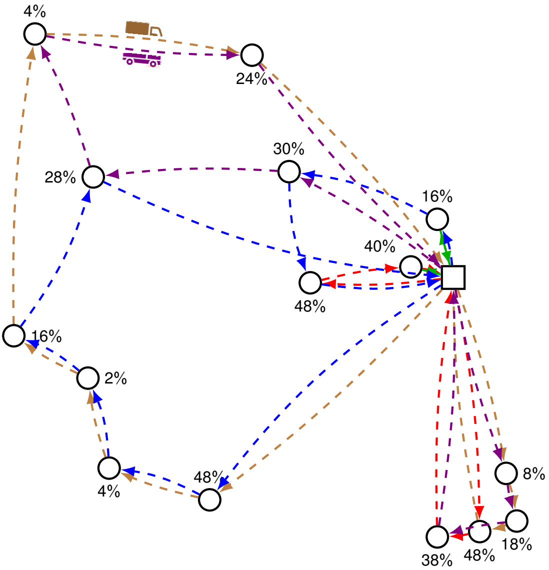
Linear Programming (LP) relaxation of the set-partitioning formulation (called Master Problem) is solved by column and cut generation:

1. For a subset of paths $P' \subset P$, define a Restricted Master Problem (RMP), containing subset P' of variables λ
2. Solve RMP by an LP solver, obtain an optimal primal solution $(\bar{x}, \bar{\lambda})$ and dual solution $(\bar{\pi}, \bar{\mu})$.
3. Solve the pricing problem to verify whether there is a variable λ_p with a negative reduced cost:

$$\min_{p \in P} \sum_{a \in A} \bar{\pi}_a h_a^p - \bar{\mu}. \tag{1}$$

4. If solution value of (1) is negative, add one or several variables λ_p to (RMP) and go to stage 2
5. Otherwise, run a separation algorithm to find constrains $Bx \leq b$ violated by $\bar{x}$. If violated inequalities are found, add them to (RMP) and go to stage 2, otherwise stop.

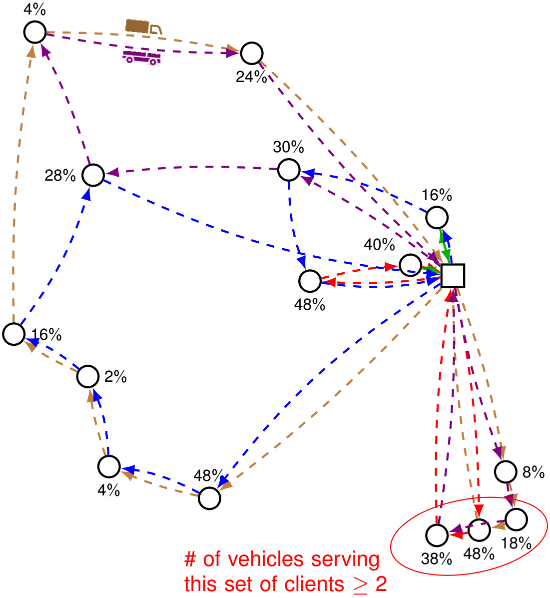
Column and cut generation: illustration



One continuous variable per feasible route.

Pricing problem is the Elementary Resource Constrained Shortest Path problem.

Column and cut generation: illustration



One continuous variable per feasible route.

Pricing problem is the Elementary Resource Constrained Shortest Path problem.

Additional constraints (cuts) are added to reduce the number of feasible non-integer solutions

Solving the pricing problem

Elementary resource constrained shortest path

Find a directed cycle in G starting at node 0, with accumulated resource consumption $\leq Q$, and minimizing the total arc reduced cost.

Labeling algorithm

- Every label represents a partial path starting from node 0.
- Label L contains
  - v^L — last visited vertex
  - $\bar{\pi}^L$ — current arc reduced cost
  - d^L — current accumulated resource consumption
  - $\mathcal{V}^L$ — set of visited vertices

Dominance

Label L dominates L' if $v^L = v^{L'}, \bar{\pi}^L \leq \bar{\pi}^{L'}, d^L \leq d^{L'}, \mathcal{V}^L \subseteq \mathcal{V}^{L'}$.
(any feasible completion of L' is feasible for L and has larger or the same reduced cost)

Basic labeling algorithm

$\mathcal{L} = \bigcup_{v \in V} \mathcal{L}_v$ — set of non-extended labels
$\mathcal{E} = \bigcup_{v \in V} \mathcal{E}_v$ — set of extended labels

---

$\mathcal{E} \leftarrow \emptyset$, $\mathcal{L} \rightarrow \{0, 0, 0, \{0\}\}$
while $\mathcal{L} \neq \emptyset$ **do**
    take a label L in $\mathcal{L}$, $v^L \neq 0$ or $\mathcal{V}^L = \emptyset$
    $\mathcal{L} \leftarrow \mathcal{L} \setminus \{L\}$, $\mathcal{E} \leftarrow \mathcal{E} \cup \{L\}$
    foreach $v \in V \setminus \mathcal{V}^L$ **do**
        extend L to L' along arc (v^L, v)
        if *L' is feasible and not dominated by a label in $\mathcal{L}_v \cup \mathcal{E}_v$* **then**
            $\mathcal{L} \leftarrow \mathcal{L} \cup \{L'\}$
            remove from $\mathcal{L}_v \cup \mathcal{E}_v$ all labels dominated by L'
    return labels L in $\mathcal{L}_0$ with $\bar{\pi}^L < \bar{\mu}$

Label-setting if labels are taken in a total order $\leq_{\text{lex}}$ such that

$$L \text{ extends to } L' \Rightarrow L \leq_{\text{lex}} L', \quad L \text{ dominates } L' \Rightarrow L \leq_{\text{lex}} L'$$

Otherwise, it is label-correcting

Robust cutting planes and branching

- Constraints of form $Bx \leq b$ are robust [Pessoa et al., 2008], i.e. their addition to the master does not change the structure of the pricing problem.
- The most important robust cutting planes are Rounded Capacity Cuts (RCC) [Laporte and Nobert, 1983]:

$$\sum_{a \in \delta^-(C)} x_a \geq \left\lceil \frac{\sum_{i \in C} d_i}{Q} \right\rceil, \quad \forall C \subseteq V.$$

- Several other robust cutting planes [Lysgaard et al., 2004] (not helpful within BCP)
- Branching on arc variables x suffices for integrality

Contents

Modern Branch-Cut-and-Price for Vehicle Routing

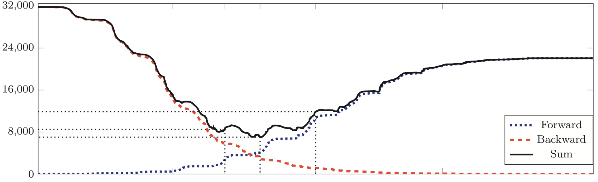
- Non-robust
  - Column and cut generation are interconnected
- Complex: not only Branch-Cut-and-Price, but

  Strong Branch-and-Price-and-Fix-and-Stabilize-and-Restrict-and-Cut-and-Enumerate-and-Heuristic-and-...
- Generic
  - Otherwise, it takes too much time to reimplement for every other problem variant.

Labeling algorithm enhancements

A subset of works tested on vehicle routing instances

- Keep track of vertices which cannot be visited instead of visited vertices in a label [Feillet et al., 2004]
- Resource discretization [Fukasawa et al., 2006]
- Bi-directional search [Righini and Salani, 2006]



[Tilk et al., 2017]

- "Pulse" algorithm: depth-first search and completion bounds [Lozano et al., 2016]
- "Bucketization" to limit the number of dominance checks [S. et al., 2017]

Non-elementary relaxations of the pricing problem

Weakens the column generation lower bound,
but keeps the BCP correct

- *q*-routes [Christofides et al., 1981]
- *k*-cycle elimination [Irnich and Villeneuve, 2006]
  (too expensive for $k \geq 5$)
- *ng*-routes [Baldacci et al., 2011b]

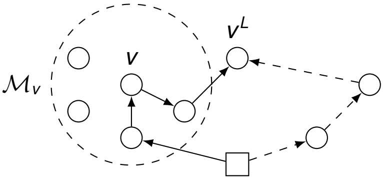
Non-elementary relaxations of the pricing problem

Weakens the column generation lower bound,
but keeps the BCP correct

- *q*-routes [Christofides et al., 1981]
- *k*-cycle elimination [Irnich and Villeneuve, 2006]
  (too expensive for $k \geq 5$)
- *ng*-routes [Baldacci et al., 2011b]

For each vertex $v \in V$, define a memory $\mathcal{M}_v$ of vertices which "remember" v.

If $v^L \notin \mathcal{M}_v$, v is removed from $\mathcal{V}^L$.
Sets $\mathcal{V}^L$ are smaller $\Rightarrow$ stronger domination



Small memories (of size ≈8-10) produce a tight relaxation of elementarity constraints for most instances.

Dynamic *ng*-route relaxation [Roberti and Mingozzi, 2014]

Even tighter relaxation can be obtained by dynamically increasing *ng*-memories.

Instance	Elementary bound		Dynamic *ng* bound	
	Gap	Time	Gap	Time
R202	0.72%	18	0.72%	58
R203	0.45%	72	0.45%	64
R204	0.88%	133	0.88%	76
R206	1.03%	45	1.04%	68
R207	0.42%	128	0.49%	79
R208	1.28%	267	1.34%	148
R209	1.57%	42	1.57%	33
R210	1.23%	34	1.23%	52
R211	1.61%	77	1.62%	54
RC204	0.49%	323	0.54%	131
RC207	1.62%	43	1.62%	38
RC208	1.21%	442	1.22%	66
Average	0.89%	151	0.91%	68

Table: Elementary bound [Lozano et al., 2016] vs. dynamic *ng* bound [S. et al., 2017] (hardest Solomon VRPTW instances)

Arc elimination using path-reduced costs [Irnich et al., 2010]

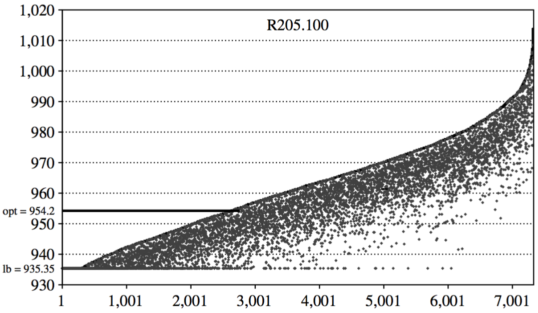
- Z_{RM} — optimum value of (MP) which gives the lower bound
- Z_{inc} — value of the incumbent integer solution
- $Z_{pricing}(a)$ — optimum solution value of the pricing problem solution, arc *a* being fixed to 1
- Arc *a* can be removed from the graph (it cannot take part of any improving solution) if

$$Z_{RM} + Z_{pricing}(a) \geq Z_{inc}$$

Arc elimination using path-reduced costs [Irnich et al., 2010]

- Z_{RM} — optimum value of (MP) which gives the lower bound
- Z_{inc} — value of the incumbent integer solution
- $Z_{pricing}(a)$ — optimum solution value of the pricing problem solution, arc a being fixed to 1
- Arc a can be removed from the graph (it cannot take part of any improving solution) if

$$Z_{RM} + Z_{pricing}(a) \geq Z_{inc}$$



A good heuristic is very important!

How we can find values $Z_{pricing}(a)$?

We perform forward and backward labeling algorithms. Then for each $a = (i, j)$,

- Let $\vec{\mathcal{L}}_i$ be set of forward labels at node i ($v^{\vec{L}} = i$)
- Let $\overleftarrow{\mathcal{L}}_j$ be set of backward labels at node j ($v^{\overleftarrow{L}} = j$)
- We find a pair of compatible labels $\vec{L} \in \vec{\mathcal{L}}_i$ and $\overleftarrow{L} \in \overleftarrow{\mathcal{L}}_j$:

$$d^{\vec{L}} + d^{\overleftarrow{L}} \leq Q, \quad \mathcal{V}^{\vec{L}} \cap \mathcal{V}^{\overleftarrow{L}} = \emptyset,$$

  minimizing $\bar{\pi}^{\vec{L}} + \bar{\pi}^{\overleftarrow{L}}$
- Then

$$Z_{pricing}(a) = \bar{\pi}^{\vec{L}} + \bar{\pi}^{\overleftarrow{L}} + \bar{\pi}_a - \mu$$

Subset-row cuts [Jepsen et al., 2008]

- Replacing variables x in set-partitioning constraints and relaxing to inequality:

$$\sum_{a \in \delta^-(v)} \sum_{p \in P} h_a^p \lambda_p \leq 1, \quad v \in V. \tag{2}$$

- Aggregating (2) for a set $C \subset V$, $|C| = 3$, with multiplier $\frac{1}{2}$:

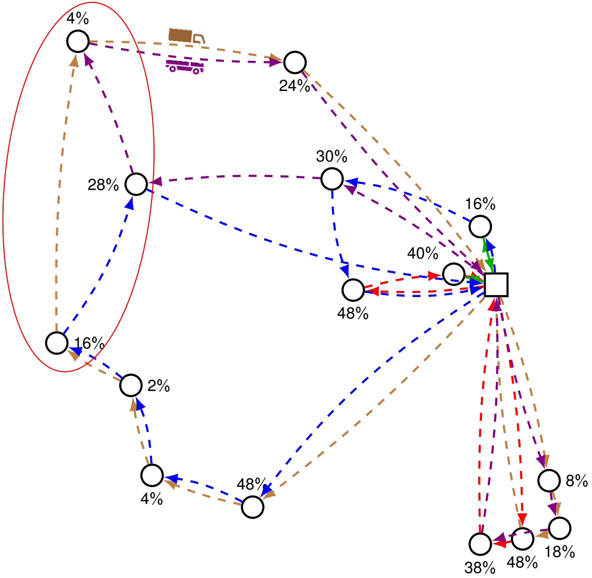
$$\sum_{p \in P} \frac{1}{2} \sum_{v \in C} \sum_{a \in \delta^-(v)} h_a^p \lambda_p \leq \frac{3}{2}, \tag{3}$$

- Performing Chvátal-Gomory rounding of (3):

$$\sum_{p \in P} \left\lfloor \frac{1}{2} \sum_{v \in C} \sum_{a \in \delta^-(v)} h_a^p \right\rfloor \lambda_p \leq 1,$$

Subset-row cuts: example of violation

of paths serving
at least two of these
three clients ≤ 1



Subset-row cuts: impact on the pricing problem

Coefficient of variable λ_p in cut η associated with subset C_η is

$$\left\lfloor \frac{1}{2} \sum_{v \in C_\eta} \sum_{a \in \delta^-(v)} h_a^p \right\rfloor .$$

Path p passes 0 or 1 times by a vertex in $C_\eta \rightarrow$ coefficient is 0.
Path p passes 2 or 3 times by a vertex in $C_\eta \rightarrow$ coefficient is 1, etc...

For each active cut $\eta \in \mathcal{N}$ we should keep binary state S_η^L in each label L.

Weaker dominance

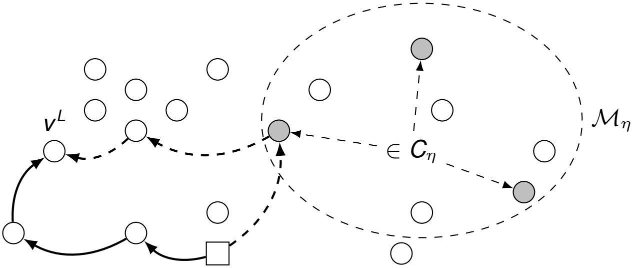
Given dual values $\nu_\eta > 0$, $\eta \in \mathcal{N}$, L dominates L' only if

$$\bar{\pi}^L \leq \bar{\pi}^{L'} - \sum_{\substack{\eta \in \mathcal{N}: \\ S_\eta^L > S_\eta^{L'}}} \nu_\eta \quad \left(\quad \text{instead of} \quad \bar{\pi}^L \leq \bar{\pi}^{L'} \right).$$

Limited memory technique [Pecin et al., 2017b]

For each active subset-row cut $\eta \in \mathcal{N}$, define a memory $\mathcal{M}_\eta$ of vertices which "remember" state S_η.

If $v^L \notin \mathcal{M}_\eta$, $S^L_\eta \leftarrow 0$.
States S^L contain more zeros $\Rightarrow$ stronger dominance



- Limited-memory cuts are weaker than full-memory ones
- However, the pricing problem difficulty is much smaller

Arbitrary cuts of Chvátal-Gomory rank 1

Chvátal-Gomory rounding using a vector p of multipliers:

$$\sum_{p \in P} \left\lfloor \sum_{v \in C} \sum_{a \in \delta^{-}(v)} p_v h_a^p \right\rfloor \lambda_p \leq \left\lfloor \sum_{v \in C} p_v \right\rfloor$$

All best possible multiplier vectors p for Chvátal-Gomory rounding of up to 5 constraints were found by [Pecin et al., 2017c].

Arbitrary cuts of Chvátal-Gomory rank 1

Chvátal-Gomory rounding using a vector p of multipliers:

$$\sum_{p \in P} \left\lfloor \sum_{v \in C} \sum_{a \in \delta^-(v)} p_v h_a^p \right\rfloor \lambda_p \leq \left\lfloor \sum_{v \in C} p_v \right\rfloor$$

All best possible multiplier vectors p for Chvátal-Gomory rounding of up to 5 constraints were found by [Pecin et al., 2017c].

	Gap(%)
Only CG (elementary routes)	2.63
+ robust cuts	0.98
+ 3SRCs	0.35
+ 4SRCs + 5SRCs	0.24
+ other R1Cs up to 5 rows	0.17

Enumeration of elementary routes [Baldacci et al., 2008]

- Z_{RM} — optimum value of (MP)
- Z_{inc} — value of the best known integer solution
- Reduced cost of $\lambda_p > Z_{inc} - Z_{RM} \Rightarrow$ it cannot participate in an improving solution.
- We enumerate all elementary routes with reduced cost $< Z_{inc} - Z_{RM}$ using a special labeling algorithm.
- If enumeration is successful, add all such variables λ_p to (RMP) and solve it using a MIP solver.
- If number of enumerated routes is large, we create a pool of routes, and solve the pricing problem by inspection [Contardo and Martinelli, 2014].

Strong branching [Røpke, 2012] [Pecin et al., 2017b]

In branch-cut-and-price, strong branching should be multi-phase

- Phase 0 — choose branching candidates (both from history and “fresh” ones)
- Phase 1 — resolve the (RMP) only without generating columns, reduce number of candidates
- Phase 2 — generated columns heuristically without generating cuts, reduce number of candidates
- Phase 3 — apply full column and cut generation for a small number of selected branching candidates, choose the best

Other important components

- Exploit forward-backward path symmetry if possible
- Heuristic column generation, call exact pricing as rare as possible
- Generate many columns at each iteration
- Clean-up (RMP) from time to time (remove columns)
- Stabilization is very important for instances with long routes
- Devise good heuristics for cut separation
- Stop cut generation if tailing-off is detected
- Stop non-robust cut generation if exact pricing is taking much time
- Rollback if pricing time is exploded
- Use primal heuristics if the initial solution is not close to the optimum.

Contents

Creating state-of-the-art algorithms for new VRP variants

- State-of-the-art BCP for CVRP is by far the most complicated BCP ever developed.
- Implementing such an algorithm takes months for an expert team, even if it is just an adaptation for another variant.
- One would like to have a generic algorithm that could be easily customised to many variants.
- Some attempts in the literature: [Desaulniers et al., 1998] [Baldacci and Mingozzi, 2009] [S. et al., 2017]

Generic BCP solver

Generic Branch-Cut-and-Price (BCP) state-of-the-art solver for Vehicle Routing Problems (VRPs) [Pessoa et al., 2019].

`vrpsolver.math.u-bordeaux.fr`

- Pre-compiled C++ code distributed in a docker image
- Open-source Julia-JuMP interface
- Demos for several VRPs are available

Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2019).
A generic exact solver for vehicle routing and related problems.
In Lodi, A. and Nagarajan, V., editors, *Integer Programming and Combinatorial Optimization*, volume 11480 of *Lecture Notes in Computer Science*, pages 354–369, Springer International Publishing.

Generic model used by VRPSolver

User should provide

- Graph(s) with the source and the sink
- For each graph, bounds for the number of paths in a solution
- Resource(s)
- Resource consumption(s) for arcs
- Accumulated resource consumption interval(s) for vertices
- Variables
- Mapping between arcs and variables
- Rounded Capacity Cuts (RCC) separators (optional)
- Separation routine(s) for problem-specific robust cuts (optional)

Generic model: collection of packing sets

Definition

A packing set is a subset of arcs (vertices) such that, in an optimal solution of the problem, at most one arc (vertex) in the subset appears at most once.

- Definition of packing sets is a part of modeling
- Packing sets generalize customers in CVRP
- Knowledge about packing sets allows the solver to use state-of-the-art techniques in a generalized form:
  - *ng*-routes
    - Distance matrix for packing sets is expected from the user to obtain initial *ng*-memories
  - Limited Memory Rank-1 Cuts
  - Elementary path enumeration
    - Additional technical condition to use enumeration

VRPSolver Julia-JuMP interface

```
using VRPSolver, JuMP
function build_model(data::DataCVRP)
   A = arcs(data) # set of arcs of the input graph G'
   n = nb_customers(data)
   V = [i for i in 1:n] # set of customers of the input graph G'
   V0 = [i for i in 0:n] # set of vertices of the graphs G' and G
   Q = veh_capacity(data)

   cvrp = VrpModel()
   @variable(cvrp.formulation, x[a in A], Int)
   @objective(cvrp.formulation, Min, sum(c(data,a) * x[a] for a in A))
   @constraint(cvrp.formulation, setpart[i in V], sum(x[a] for a in inc(data, i)) == 1.0)

   function build_graph() # Build the model directed graph G=(V,A)
      v_source = v_sink = 0
      G = VrpGraph(cvrp, V0, 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 A
         arc_id = add_arc(G, i, j, x[(i,j)])
         set_arc_consumption(G, arc_id, cap_res_id, d(data, j))
      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, [[distance(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
```

Problems which we modelled and solved

- Capacitated Vehicle Routing Problem (CVRP)
- CVRP with Time Windows
- Heterogeneous Fleet CVRP
- Multi-depot CVRP
- Pickup-and-Delivery Problem with Time Windows
- CVRP with Backhauls
- (Capacitated) Team Orienteering Problem
- Capacitated Profitable Tour Problem
- Vehicle Routing Problem With Service Levels
- Generalized Assignment Problem
- Vector Packing Problem
- Bin Packing Problem
- Capacitated Arc Routing Problem
- Robust CVRP with Demand Uncertainty
- Location-Routing Problem
- Two-Echelon Vehicle Routing Problem

VRPSolver : main use cases

- Benchmarking heuristic algorithms against the lower bound/optimal solution obtained by the solver
- Benchmarking exact algorithms against the solver
- Creating efficient models for new problem variants, both VRP and related ones (scheduling, network design, etc) (room for creative modelling!)
- Testing new families of (robust) cutting planes within a state-of-the-art Branch-Cut-and-Price algorithm.

Contents

Computational results

Problem	Data set	Number	Size	Time	Gen. BCP		Best Publication
CVRP	E-M	12	51-200	10h	12 (61s)	**12 (49s)**	[Pecin et al., 2017b]
	X	58	101-393	60h	**36 (147m)**	34 (209m)	[Uchoa et al., 2017]
VRPTW	Solomon Hard	14	100	1h	**14 (5m)**	13 (17m)	[Pecin et al., 2017a]
	Gehring Homb	60	200	30h	**56 (21m)**	50 (70m)	[Pecin et al., 2017a]
HFVRP	Golden	40	50-100	1h	**40 (144s)**	39 (287s)	[Pessoa et al., 2018]
MDVRP	Cordeau	11	50-360	1h	**11 (6m)**	11 (7m)	[Pessoa et al., 2018]
PDPTW	Ropke Cordeau	40	60-150	1h	**40 (5m)**	33 (17m)	[Gschwind et al., 2018]
	LiLim	30	200	1h	3 (56m)	**23 (20m)**	[Baldacci et al., 2011a]
TOP	Chao class 4	60	100	1h	**55 (8m)**	39 (15m)	[Bianchessi et al., 2018]
CTOP	Archetti	14	51-200	1h	**13 (7m)**	6 (35m)	[Archetti et al., 2013]
CPTP	Archetti open	28	51-200	1h	**24 (9m)**	0 (1h)	[Bulhoes et al., 2018]
VRPSL	Bulhoes et al.	180	31-200	2h	**159 (16m)**	49 (90m)	[Bulhoes et al., 2018]
GAP	OR-Lib class D	6	100-200	2h	5 (40m)	**5 (30m)**	[Posta et al., 2012]
	Nauss	30	90-100	1h	**25 (23m)**	1 (58m)	[Gurobi Optimization, 2017]
BPP	Falkenauer T	80	60-501	10m	80 (16s)	**80 (1s)**	[Brandão and Pedroso, 2016]
	Hard28	28	200	10m	28 (17s)	**28 (4s)**	[Delorme and Iori, 2018]
	AI	250	200-1000	1h	**160 (25m)**	140 (28m)	[Wei et al., 2019]
	ANI	250	200-1000	1h	**103 (35m)**	97 (40m)	[Wei et al., 2019]
VPP	Classes 1,4,5,9	40	200	1h	**38 (8m)**	13 (50m)	[Heßler et al., 2018]
CARP	Eglese	24	77-255	30h	**22 (36m)**	22 (43m)	[Pecin and Uchoa, 2019]

Table: Generic solver vs. best specific algorithms on 13 problems.

State-of-the-art performance and bottlenecks

Performance

- Now most instances of the most classic VRPs with up to 200 customers can be solved, some of them in a long time
- More importantly, instances with up to 100 customers can often be solved in less than 1 minute

Bottlenecks

- Separation of Chvátal-Gomory rank-1 cuts
- Premature branching due to the pricing problem difficulty
- Slow column generation convergence in some cases
- No efficient generic primal heuristics

Perspectives

- Up to now, exact algorithms were only used to benchmark heuristics
- This may change in the future, as customizable codes with state-of-the-art performance are starting to be available
- We expect that exact algorithms will be much more used by VRP practitioners

References I

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

Baldacci, R., Bartolini, E., and Mingozzi, A. (2011a).
An exact algorithm for the pickup and delivery problem with time windows.
Operations Research, 59(2):414–426.

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

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

References II

Baldacci, R., Mingozzi, A., and Roberti, R. (2011b).
New route relaxation and pricing strategies for the vehicle routing problem.
Operations Research, 59(5):1269–1283.

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

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

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

References III

Bulhoes, T., Hà, M. H., Martinelli, R., and Vidal, T. (2018).
The vehicle routing problem with service level constraints.
European Journal of Operational Research, 265(2):544 – 558.

Christofides, N., Mingozzi, A., and Toth, P. (1981).
Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations.
Mathematical Programming, 20(1):255–282.

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

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

References IV

- Dantzig, G. B. and Ramser, J. H. (1959).
  The truck dispatching problem.
  Management Science, 6(1):80–91.

- Delorme, M. and Iori, M. (2018).
  Enhanced pseudo-polynomial formulations for bin packing and cutting stock problems.
  INFORMS Journal on Computing, Forthcoming.

- Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, M. M., Soumis, F., and Villeneuve, D. (1998).
  A Unified Framework for Deterministic Time Constrained Vehicle Routing and Crew Scheduling Problems, pages 57–93.
  Springer US, Boston, MA.

- Desrochers, M., Desrosiers, J., and Solomon, M. (1992).
  A new optimization algorithm for the vehicle routing problem with time windows.
  Operations Research, 40(2):342–354.

References V

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004).
An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems.
Networks, 44(3):216–229.

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

Gschwind, T., Irnich, S., Rothenbächer, A.-K., and Tilk, C. (2018).
Bidirectional labeling in column-generation algorithms for pickup-and-delivery problems.
European Journal of Operational Research, 266(2):521 – 530.

Gurobi Optimization, L. (2017).
Gurobi optimizer reference manual, version 7.5.

References VI

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

Irnich, S., Desaulniers, G., Desrosiers, J., and Hadjar, A. (2010).
Path-reduced costs for eliminating arcs in routing and scheduling.
INFORMS Journal on Computing, 22(2):297–313.

Irnich, S. and Villeneuve, D. (2006).
The shortest-path problem with resource constraints and k-cycle elimination for $k \geq 3$.
INFORMS Journal on Computing, 18(3):391–406.

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

References VII

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

Lozano, L., Duque, D., and Medaglia, A. L. M. L. (2016).
An exact algorithm for the elementary shortest path problem with resource constraints.
Transportation Science, 50(1):348–357.

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

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

References VIII

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

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

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

Pessoa, A., de Aragão, Marcus, M. P., and Uchoa, E. (2008).
Robust branch-cut-and-price algorithms for vehicle routing problems.
In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 297–325. Springer US.

References IX

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

Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2019).
A generic exact solver for vehicle routing and related problems.
In Lodi, A. and Nagarajan, V., editors, *Integer Programming and Combinatorial Optimization*, volume 11480 of *Lecture Notes in Computer Science*, pages 354–369, Cham. Springer International Publishing.

Poggi, M. and Uchoa, E. (2014).
Chapter 3: New Exact Algorithms for the Capacitated Vehicle Routing Problem, pages 59–86.
SIAM Publications.

References X

Posta, M., Ferland, J. A., and Michelon, P. (2012).
An exact method with variable fixing for solving the generalized assignment problem.
Computational Optimization and Applications, 52:629–644.

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

Roberti, R. and Mingozzi, A. (2014).
Dynamic ng-path relaxation for the delivery man problem.
Transportation Science, 48(3):413–424.

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

References XI

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

Tilk, C., Rothenbächer, A.-K., Gschwind, T., and Irnich, S. (2017).
Asymmetry matters: Dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster.
European Journal of Operational Research, 261(2):530 – 539.

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

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