A New Exact Approach for the Vehicle Routing Problem with Intermediate Replenishment Facilities

Roberto Wolfler Calvo[1], Paolo Gianessi[1], Lucas Létocart[1]

[1] LIPN UMR CNRS 7030, University Paris 13, 99, av J.B. Clement, 93430 Villetaneuse, France

7th Internation Workshop on Vehicle Routing, Intermodal Transportation and Related Areas

ROUTE 2014

Copenhagen, Denmark, june 4^{th}, 2014

Outline

Problem Definition

Data

- a set of customers, each one with a demand and a service time
- one depot, the base of the fleet of vehicles
- a set of replenishment facilities, with a recharge time each
- a set of vehicles of fixed capacity

Constraints

- each customers must be served by exactly one vehicle
- when empty, a vehicle can stop and recharge at a facility
- a rotation is the sequence of routes of a vehicle
- its rotation must start and end at the depot
- the total duration of its rotation must not exceed a given shift length

Objective function

Find a minimum cost set of routes

An example of instance and solution

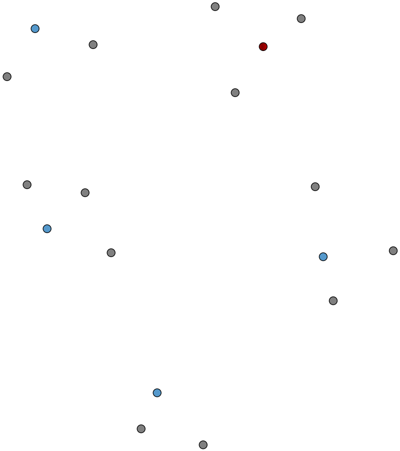

Figure 1: A VRPIRF instance: the depot (red), the facilities (blue), and the customers

An example of instance and solution

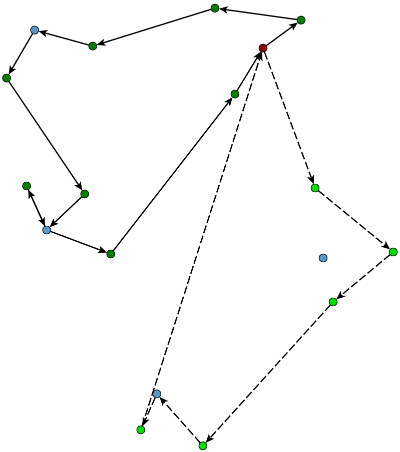

Figure 2: A solution to the previous instance

Literature Review

Exact methods

- A Branch&Cut on the VRP with satellite facilities ([2])
- A Branch&Price for the Multi-Depot VRP with Inter-Depot Routes ([4])

Heuristics

- A Hybrid Guided Local Search (VNS, Tabu Search) ([1])
- An Adaptive VNS for the VRP with Intermediate Stops ([5])
- A Tabu Search for the MDVRPI ([3])

Similarities with..

- Collection of waste (see [6], [7])
- VRPIRF belongs to the family of Multi-Depot VRPs (see [10], [11] for instance)
- it has some elements in common with Multi-Trip VRPs ([8], [9])

Notation I

Data

$V_c = \{1, ..., n\}$	$i, j \in V_c$	#, set, indexes of clients
$V_p = \{n+1, ..., n+f\}$	$p \in V_p$	#, set, index of replenishment facilities
$K = \{1, ..., n_K\}$	$k \in K$	#, set, index of vehicles
$V = \{0\} \cup V_c \cup V_p$	$v \in V$	entire node set (0 = depot node)
$A = V \times V_c \cup V_c \times V$	$ij \in A$	set of arcs
Q		capacity of vehicles
T		max duration of a rotation
q_i, τ_i	$i \in V_c$	demand and service time of clients
τ_p	$p \in P$	recharge time at facilities
d_{ij}, τ_{ij}	$ij \in A_0$	routing cost and travel time of arcs
$t_{ij} = \tau_i + \tau_{ij}$	$ij \in A$	extended time of arcs ($\tau_0 \equiv 0$)

Notation II

Set notation

$\mathcal{S}(V_c)$	collection $\{S \subset V_c : 2 \leq \lvert S \rvert \leq \lvert V_c \rvert - 2\}$ of customer subsets
$\delta^+(S)$, $\delta^-(S)$	cutsets of in- and outgoing arcs of $S \subseteq V_c$
$S_1 : S_2$	cutsets of arcs $ij : i \in S_1, j \in S_2$
$\overline{S}$	complementary set $V_c \setminus S$
$A(S)$	arcs with both endpoint in $S \subseteq V_c$

Decision variables and compact notation

$x_{ij}^k \in \{0,1\},\ k \in K,\ ij \in A \quad x_{ij}^k = 1 \Leftrightarrow$ vehicle k visits node j after node i

$y_p^k \in \{0,1\},\ k \in K,\ p \in V_p \quad y_p^k = 1 \Leftrightarrow$ vehicle k visits facility p at least once

$x^k(A'),\ A' \subseteq A$ aggregate sum $\sum_{ij \in A'} x_{ij}^k$

Further notation

$\kappa(S)$	min # of routes to serve clients in $S \in \mathcal{S}(V_c)$ (solution of BP)
$r(S)$	trivial lower bound $\lceil \frac{1}{Q} \sum_{i \in S} q_i \rceil$ on $\kappa(S)$

A Three-Index Formulation

$$
\begin{array}{llll}
\min & \sum_{k \in K} \sum_{ij \in A} d_{ij} x_{ij}^k & & \\
\text{s.t.} & \sum_{ij \in A} t_{ij} x_{ij}^k \leq T & \forall k \in K & \text{duration} \\
& \sum_{k \in K} x^k(\delta^+(i)) = 1 & \forall i \in V_C & \text{client service} \\
& x^k(\delta^+(i)) = x^k(\delta^-(i)) & \forall i \in V_C,\ k \in K & \\
& x^k(0 : V_c) \leq 1 & \forall k \in K & \text{depot degree} \\
& x^k(0 : V_c) = x^k(V_c : 0) & \forall k \in K & \\
& x^k(p : V_c) = x^k(V_c : p) & \forall k \in K & \text{facility degree} \\
& \sum_{k \in K} x^k(A(S)) \leq |S| - \kappa(S) & \forall S \in \mathcal{S}(V_c) & \text{capacity} \\
& x_{ip}^k \leq y_p^k & \forall i \in V_c,\ k \in K,\ p \in V_F & \text{activity} \\
& x^k(\overline{S} : S) \geq y_p^k & \forall k \in K,\ p \in V_F,\ S \subseteq V \setminus p & \text{connectivity} \\
& x^k(S : \overline{S}) \geq y_p^k & \forall k \in K,\ p \in V_F,\ S \subseteq V \setminus p & \\
& \sum_{k \in K} (x^k(\delta^-(0)) + \sum_{p \in V_p} x^k(\delta^-(p))) \geq \kappa(V_c) & \forall k \in K & \text{min \# vehicles} \\
& x^k(\delta^+(i)) \leq x^k(V_c : 0) & \forall i \in V_c,\ k \in K & \text{trick}
\end{array}
$$

Replenishment arcs I

(Almost) Back to routes

- with replenishment arcs, rotation becomes very similar to a route in classical CVRP
- the depot is the only node with in/outdegree greater than 1

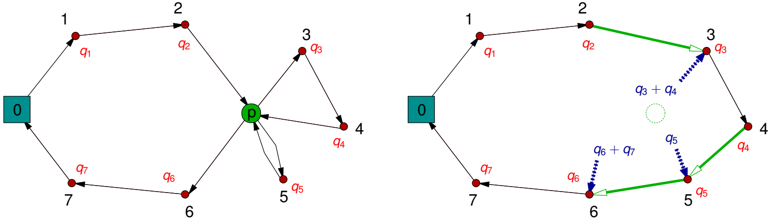

Figure 3: The same rotation with replenishment arcs (right) and without

Replenishment arcs II

Replenishment arcs allow to overcome the weakness of three-index model:

- now we can remove both activity variables y..
- ..and connectivity constraints

Connection..

we can impose the respect of connection constraints with a simple extension of classical SECs

$$(\forall S \in \mathcal{S}(V_c)) \quad x(A_0(S)) + w(A_P(S)) \leq |S| - 1$$

..meets capacity

with no change in the form of capacity constraints

$$(\forall S \in \mathcal{S}(V_c)) \quad x(A_0(S)) \leq |S| - \kappa(S)$$

Two index variables I

With vehicles

The three-index formulation has some drawbacks:

- symmetry issues
- very scattered fractionary solutions

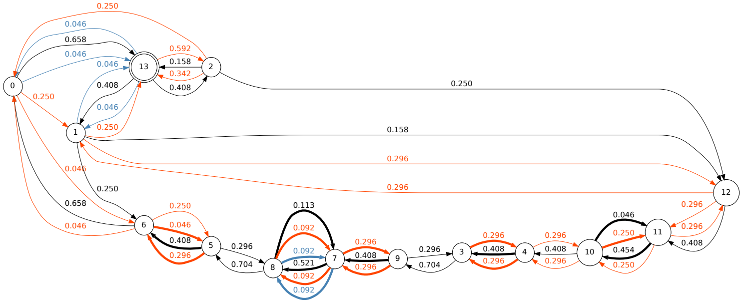

Figure 4: A three-index fractionary solution of the previous instance

Two index variables II

Without vehicles

If we introduce new continuous variables z and new constraints:

$$(\forall i \in V_c) \quad \sum_{v \in V \setminus i} z_{iv} = \sum_{v \in V \setminus i} z_{vi} + \sum_{v \in V \setminus i} t_{iv} x_{iv} + \sum_{j \in V_c \setminus i} u_{ij} w_{ij}$$

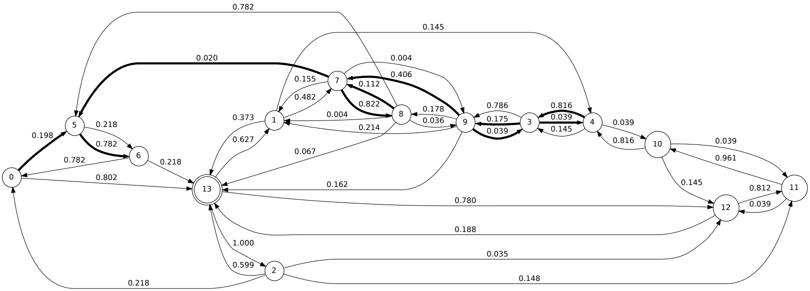

Figure 5: A two-index fractionary solution of the same instance

Arrival times example

How do arrival times work

In the example below:

- $x_{ij} = w_{ij} = 0$ except for $x_{01} = x_{12} = w_{23} = x_{34} = x_{45} = x_{50} = 1 \Rightarrow$ $z_{ij} = 0$ except for $z_{01}, z_{12}, z_{23}, z_{34}, z_{45}$ and z_{50} (z bounds)

How do z variables track time:

- $z_{01} = t_{01}$ (rotation start)
- for $i = 1$ (time track):

$$\sum_{v \in V \setminus \{1\}} z_{1v} = z_{12} = \sum_{v \in V \setminus \{1\}} z_{v1} + \sum_{v \in V \setminus \{1\}} t_{1v} x_{1v} + \sum_{j \in V_c \setminus \{1\}} u_{1j} w_{1j} = z_{01} + t_{12}$$

- similarly $z_{23} = z_{12} + u_{23}$, $z_{34} = z_{23} + t_{34}$, $z_{45} = z_{34} + t_{45}$ and $z_{50} = z_{45} + t_{50}$
- $z_{50} \leq T$ (shift duration)

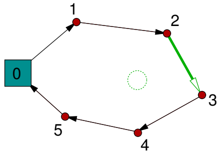

Figure 6: An instance with $f = 1$ and $n = 5$ and a solution with only one rotation. The vehicle performing it is ensured to be back at the depot within time T

A Two-Index Formulation I

New variables

x_{ij}	$\in \{0,1\}$,	$ij \in A_0$	$x_{ij} = 1 \Leftrightarrow$ node j follow node i in one same route
w_{ij}	$\in \{0,1\}$,	$ij \in A_P$	$w_{ij} = 1 \Leftrightarrow$ vehicle recharges in between clients i, j
z_{ij}	$\in \mathbb{R}$,	$i, j \in V$	arrival time at node j if its predecessor is node i

Compact notation

$x(A')$,	$A' \subseteq A_0$	aggregate sum $\sum_{ij \in A'} x_{ij}$
$w(A')$,	$A' \subseteq A_P$	aggregate sum $\sum_{ij \in A'} w_{ij}$

A Two-Index Formulation II

Model

min	$\sum_{ij \in A_0} d_{ij} x_{ij} + \sum_{ij \in A_P} e_{ij} w_{ij}$		
s.t.	$x(\delta_0^-(0)) + w(A_P)) \geq \kappa(V_c)$		min # of routes
	$x(\delta_0^+(i)) + w(\delta_P^+(i)) = 1$	$i \in V_c$	client service/1
	$x(\delta_0^-(i)) + w(\delta_P^-(i)) = 1$	$i \in V_c$	client service/2
	$x(\delta_0^-(0)) = x(\delta_0^+(0)) \leq n_K$		depot degree
	$x(A_0(S)) \leq \lvert S\rvert - \kappa(S)$	$S \in \mathcal{S}(V_c)$	capacity
	$x(A_0(S)) + w(A_P(S)) \leq \lvert S\rvert - 1$	$S \in \mathcal{S}(V_c)$	connection
	$z_{0i} = t_{0i} x_{0i}$	$i \in V_c$	rotation start
	$(t_{0i} + t_{ij}) x_{ij} + (t_{0i} + u_{ij}) w_{ij} \leq z_{ij}$	$i \in V_c, j \in V_c \setminus i$	z bounds/1
	$z_{ij} \leq (T - t_{j0})(x_{ij} + w_{ij})$	$i \in V_c, j \in V_c \setminus i$	z bounds/2
	$(t_{0i} + t_{i0}) x_{i0} \leq z_{i0}$	$i \in V_c$	z bounds/3
	$z_{i0} \leq T x_{i0}$	$i \in V_c$	shift duration
	$\sum_{v \in V \setminus i} z_{iv} = \sum_{v \in V \setminus i} z_{vi} + \sum_{v \in V \setminus i} t_{iv} x_{iv} + \sum_{j \in V_c \setminus i} u_{ij} w_{ij}$	$i \in V_c$	time track

Separation of Capacity Constraints I

Rounded Capacity Inequalities

capacity inequalities are separated in the form of rounded capacity inequalities (RCI) which replaces $\kappa(S)$ by $r(S)$

Graph transformation

- separation is performed with J.Lysgaard's package CVRPSEP (see [12])
- CVRPSEP requires the support graph to be symmetric
- a transformation of our support graph is therefore necessary

Separation of Capacity Constraints II

α-Transformation of the graph and Capacity Separation

- the routines are fed with the α-transformation of the graph: (α-separation):

$$(\forall i, j \in V_c, i < j) \;\; x_{ij}|^{\alpha} = x_{ij} + x_{ji} \qquad (\forall i \in V_c) \;\; x_{i0}|^{\alpha} = x_{0i} + x_{i0} + \sum_{j \in V_c \setminus i} (w_{ij} + w_{ji})$$

- on every set S which RCI is violated of more than a threshold θ we impose:

$$x(A_0(S)) \leq |S| - r(S) \qquad x(A_0(S)) + w(A_P(S)) \leq |S| - 1$$

- if $r(S) = 1$ the first constraint is not added since redundant

Separation of Capacity Constraints

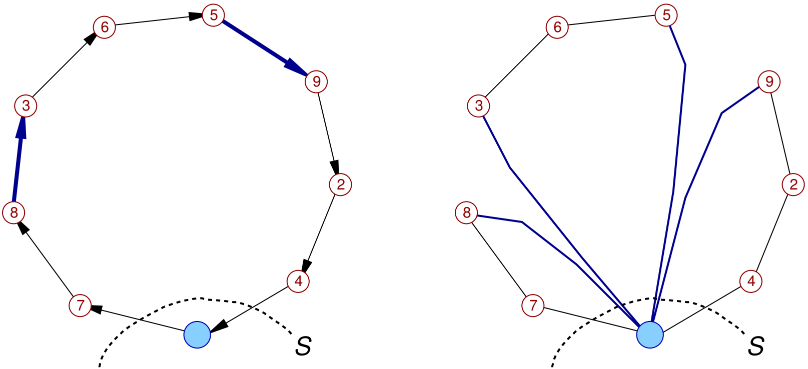

Figure 7: This new solution (left) will not be cut, as it violates neither the capacity contraint on *S* nor any other one: α-separation does not detect anything on its α-transformation (right).

Separation of Capacity Constraints

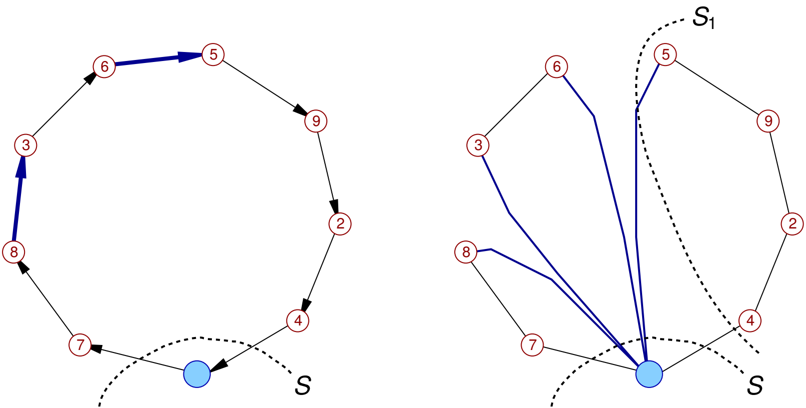

Figure 8: Even if it does not violate the capacity contraint on S, this new solution (left) would be cut, as α-separation would detect a violation on set S_1 (right).

Separation of Capacity Constraints

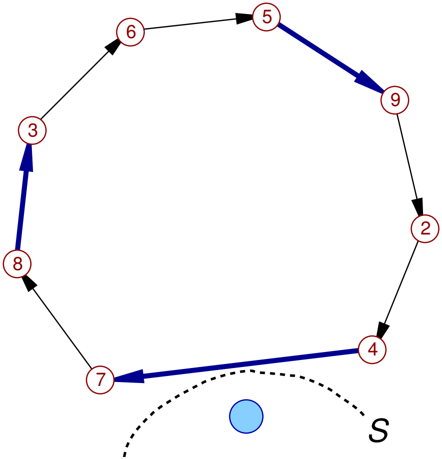

Figure 9: Another solution that would be cut in the following, as it violates the connection constraint on S imposed as a by-product of α-transformation (even if the capacity constraint is respected).

Separation of Connectivity Constraints

β-Transformation of the graph and Non-Connectivity Detection

- solutions respecting added RCIs could be nonconnected (see examples)
- to overcome this, immediately after α-separation, we perform another separation (β-separation) based on the β-transformation of the graph:

$$(\forall i, j \in V_c, i < j) \quad x_{ij}|^{\beta} = x_{ij} + x_{ji} + w_{ij} + w_{ji} \qquad (\forall i \in V_c) \quad x_{i0}|^{\beta} = x_{0i} + x_{i0}$$

- β-separation: a classical maxflow-based procedure to separate SECS
  - $(\forall i \in V_c)$ solve $0-i$ maxflow problem with $\mathbf{x}|^{\alpha}$ as support graph $\Rightarrow$ mincut S_i
  - if S_i has capacity < 1, impose $x(A_0(S_i)) + w(A_P(S_i)) \leq |S_i| - 1$
- connection constraint is replaced by its equivalent if it is the sparsest:

$$x(\delta_0^+(S)) + w(\delta_P^+(S)) \geq 1$$

Separation of Connectivity Constraints

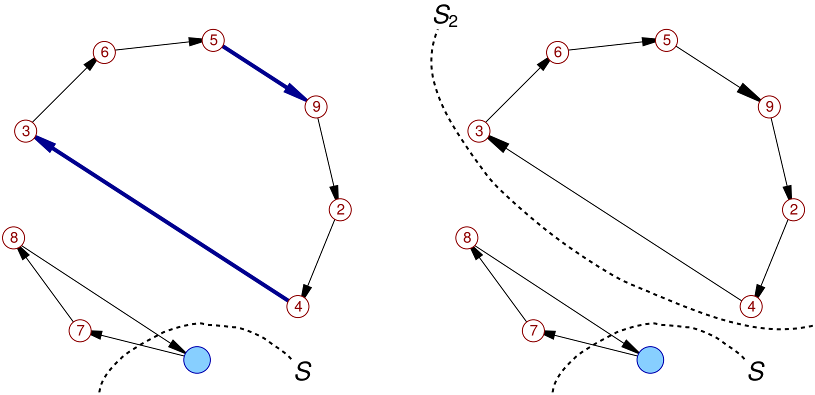

Figure 10: This solution (left) does not violate the capacity contraint on S, neither it can be cut by α-separation (the α-transformation is the same of Figure 7). However, β-separation will detect a violation on set S_2 and the solution will be cut.

Separation of other Constraints

Multistar inequalities

- these inequalities are originally in the form $\lambda x(E(N)) + x(N:S) \leq \gamma$
- $N \subset V_c$: nucleus, $S \subseteq V_c \setminus N$: satellites, λ and γ depend on $|N|$,$|S|$.
- separation is performed once again with CVRPSEP package
- cuts need to be adapted to the asymmetric case

Separation Strategy

Separation Algorithm

Both at root and at each node of the B&B tree (whether its solution is integer feasible or not) we follow these steps:

```
LPoptimize
while(true)
   perform α-separation
   if(there are sets whose capacity cut is violated by more than θ_α)
      add capacity and connection cuts on those sets and LPoptimize
      perform β-separation
      if(there are sets whose connection cut is violated by more than θ_β)
         add connection cuts on those sets and LPoptimize
      continue
   perform β-separation
   if(there are sets whose connection cut is violated by more than θ_β)
      add connection cuts on those sets and LPoptimize
      continue
   if(current solution is integer feasible) break
   perform separation of Multistar inequalities
   if(there are cuts which are violated by more than θ_μ)
      add those cuts and LPoptimize
      continue
   break
```

Computational Results

Benchmark Instances

- tests have been conducted on the instances of Crevier et al. (see [3]) and Tarantilis et al. (see [1])
- instances features range from:
  - 48 to 216 customers
  - 2 to 7 facilities
  - 2 to 8 vehicles

Computational Strategy

- on smaller instances (48 to 75 customers):
  - complete computation with a time limit from 3600s to 5400s on both the root note computation and the B&B search
  - the B&B search has been given the best known solution as initial UB
- on bigger instances (96+ customers):
  - only the root node computation has been conducted, time limit: always 3600s
  - the gap with the best known solution is reported

Computational Results

instance	t_{lim}	t_{root}	$t_{B\&C}$	UB_w	UB_b	LB	%
50c3d2v	3600	49	523	2239	2239	2239.00	0.00
50c3d4v	3600	11	3600	2400	2384	2185.87	8.31
50c3d6v	3600	42	3600	3031	3022	2701.95	10.59
50c5d2v	3600	108	3600	2640	2640	2626.75	0.50
50c5d4v	3600	59	3600	3120	3120	2914.98	6.57
50c5d6v	3600	37	3600	3583	3583	3205.15	10.55
50c7d2v	3600	169	3600	3388	3388	3361.62	0.78
50c7d4v	3600	136	3600	3416	3416	3369.19	1.37
50c7d6v	3600	70	3600	4132	4089	3681.10	9.98

Table 1: Results on Tarantilis instances with 50 customers.

Computational Results

instance	t_{lim}	t_{root}	$t_{B\&C}$	UB_w	UB_b	LB	%
75c3d2v	5400	1542	5400	2725	2725	2695.33	1.09
75c3d4v	5400	1089	5400	2793	2793	2737.86	1.97
75c3d6v	5400	1017	5400	3502	3502	3179.16	9.22
75c5d2v	5400	1830	5400	3425	3425	3343.64	2.38
75c5d4v	5400	391	5400	3620	3620	3423.63	5.42
75c5d6v	5400	755	5400	4254	4254	3939.69	7.39
75c7d2v	5400	1175	5400	3618	3618	3579.57	1.06
75c7d4v	5400	612	5400	3881	3871	3708.38	4.20
75c7d6v	5400	514	5400	4287	4287	3989.61	6.94

Table 2: Results on Tarantilis instances with 75 customers.

Computational Results

instance	t_{root}	UB_w	LB	%
100c3d3v	3600	3179	3101.97	2.42
100c3d5v	1463	3606	3178.38	11.86
100c3d7v	3600	4300	3853.87	10.38
100c5d3v	3600	4110	3998.63	2.71
100c5d5v	259	4466	4087.19	8.48
100c5d7v	3600	5206	4548.10	12.64
100c7d3v	3600	4278	4025.12	5.91
100c7d5v	293	4524	4072.14	9.99
100c7d7v	808	4961	4459.69	10.10

Table 3: Results on Tarantilis instances with 100 customers.

Computational Results

instance	t_{root}	UB_w	LB	%
125c4d3v	3600	3995	3876.89	2.96
125c4d5v	3600	4392	3981.17	9.35
125c4d7v	3600	4839	4206.77	13.07
125c6d3v	3600	4142	3999.19	3.45
125c6d5v	3600	4906	4327.09	11.80
125c6d7v	3600	5406	4628.97	14.37
125c8d3v	3600	4632	4386.89	5.29
125c8d5v	3600	5124	4575.70	10.70
125c8d7v	3600	5494	4591.21	16.43

Table 4: Results on Tarantilis instances with 125 customers.

Computational Results

instance	t_{root}	UB_w	LB	%
150c4d3v	3600	4134	3860.88	6.61
150c4d5v	3600	4724	3968.24	16.00
150c4d7v	1010	5263	4277.04	18.73
150c6d3v	793	4144	3737.52	9.81
150c6d5v	3600	4967	4354.96	12.32
150c6d7v	3600	5869	4844.69	17.45
150c8d3v	3600	4743	4495.64	5.22
150c8d5v	3600	5205	4599.48	11.63
150c8d7v	3600	5755	5129.98	10.86

Table 5: Results on Tarantilis instances with 150 customers.

Computational Results

instance	t_{root}	UB_w	LB	%
175c4d4v	3600	4810	4116.2	14.42
175c4d6v	3600	4940	4117.42	16.65
175c4d8v	1257	6046	4712.33	22.06
175c6d4v	3600	5121	4494.91	12.23
175c6d6v	3600	5527	4653.93	15.80
175c6d8v	3600	6185	5002.54	19.12
175c8d4v	3600	5987	4989.7	16.66
175c8d6v	3600	6090	5066.11	16.81
175c8d8v	3600	7042	5591.82	20.59

Table 6: Results on Tarantilis instances with 175 customers.

Computational Results

instance	$\lvert V_c \rvert$	$\lvert P \rvert$	$\lvert K \rvert$	t_{lim}	t_{root}	$t_{B\&C}$	UB_w	UB	LB	%
a1	48	3	6	3600	167	3600	1209	1209	1125.78	6.88
d1	48	4	5	3600	177	3602	1088	1088	1019.04	6.34
g1	72	5	5	5400	3055	5400	1224	1224	1170.50	4.37
j1	72	6	4	5400	842	5400	1161	1161	1105.94	4.74

Table 7: Results on Crevier instances with 48 to 72 customers.

Computational Results

instance	$\lvert V_c \rvert$	$\lvert P \rvert$	$\lvert K \rvert$	t_{root}	UB_w	LB	%
b1	96	3	4	3600	1272	1230.76	3.24
e1	96	4	5	810	1357	1330.48	1.95
h1	144	5	4	3600	1871	1515.21	19.02
k1	144	6	4	3600	1660	1562.32	5.88
c1	192	3	5	3600	1987	1714.55	13.71
f1	192	4	4	3600	1678	1425.60	15.04
i1	216	5	4	3600	2039	1734.12	14.95
l1	216	6	4	3600	1987	1645.27	17.20

Table 8: Results on Crevier instances with 96 to 216 customers.

Conclusions

Recap

- A new Branch&Cut algorithm for the VRPIRF has been presented, capable of very good gaps at the root node in comparison to the best known solutions
- on smaller instances, some new best known solutions have been found

Future work

- refinement of the code and introduction of other known cuts from the CVRP and other problem similar to VRPIRF
- study of specific valid inequalities derived from the structural properties of the problem

More to come..

A Branch&Price Algorithm for the same problem has been designed and its implementation is in progress

Thank you for your attention!

wolfler@lipn.fr gianessi@lipn.fr letocart@lipn.fr

References in the Literature I

MDVRPI, VRPIRF, VRPIS

[1] C. D. Tarantilis, E. E. Zachariadis, C. T. Kiranoudis.
A Hybrid Guided Local Search for the Vehicle-Routing Problem with Intermediate Replenishment Facilities.
INFORMS Journal on Computing, 20(1) : 154–168, 2008.

[2] J. F. Bard, M. Dror, L. Huang, P. Jaillet.
A branch and cut algorithm for the VRP with satellite facilities.
IIE Transactions, 30(9) : 821–834, 1998.

[3] B. Crevier, J.-F. Cordeau, G. Laporte.
The multi-depot vehicle routing problem with inter-depot routes.
European Journal of Operational Research, 176(2) : 756–773, 2007.

[4] I. Muter, J.-F. Cordeau, G. Laporte.
A Branch-and-Price Algorithm for the Multi-Depot Vehicle Routing Problem with Inter-Depot Routes.
submitted to Transportation Science, 2012.

[5] M. Schneider, A. Stenger, J. Hof.
An Adaptive VNS Algorithm for Vehicle Routing Problems with Intermediate Stops.
Technical Report LPIS-01/2014.

References in the Literature II

MTVRP

[6] E. Angelelli, M. G. Speranza
The periodic vehicle routing problem with intermediate facilities
European Journal of Operational Research, 137 : 233–247, 2002.

[7] A. Benjamin, J. Beasley.
Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities.
Computers and Operation Research, 37(12) : 2270–2280, 2013.

MTVRP

[8] A. Mingozzi, R. Roberti, P. Toth.
An exact algorithm for the multi-trip vehicle routing problem.
INFORMS Journal on Computing, 25(2) : 193–207, 2013.

[9] D. Cattaruzza, N. Absi, D. Feillet, T. Vidal.
A memetic algorithm for the multi trip vehicle routing problem.
European Journal of Operational Research, http://dx.doi.org/10.1016/j.ejor.2013.06.012, 2013.

References in the Literature III

MDVRP

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

[11] J.-F. Cordeau, M. Gendreau, G. Laporte.
A tabu search heuristic for periodic and multi-depot vehicle routing problems.
Networks, 30(2) : 105–119, 1997.

References in the Literature IV

Branch&Cut for CVRP

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

[13] A. N. Letchford, R. W. Eglese, J. Lysgaard.
Multistars, partial multistars and the capacitated vehicle routing problem.
Mathematical Programming, 94 (1) : 21–40, 2002.

Replenishment Arcs

[14] O. J. Smith, N. Boland, H. Waterer.
Solving shortest path problems with a weight constraint and replenishment arcs.
Computers & OR, 39(5) : 964–984, 2012.

ADVRP

[15] S. Almoustafa, S. Hanafi, N. Mladenovic.
New exact method for large asymmetric distance-constrained vehicle routing problem.
European Journal of Operational Research, 226(3) : 386–394, 2013.