Limited-memory subset-row cuts for VRP

Diego Pecin

Polytechnique Montréal and GERAD, Canada

Joint work with
C. Contardo, G. Desaulniers, A. Pessoa, M. Poggi and E. Uchoa

Column Generation
Búzios, Brazil, May 23, 2016

Outline

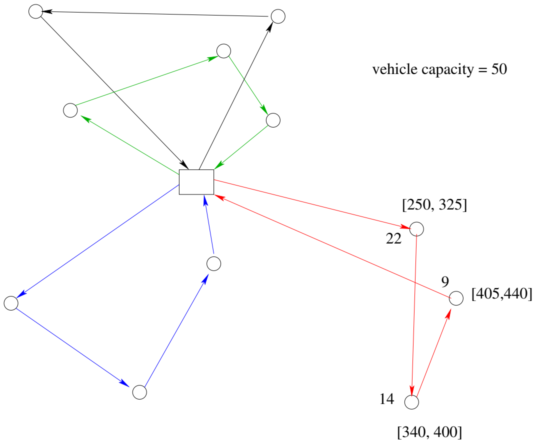
Capacitated vehicle routing problem (CVRP)

- Given
  - unlimited number of identical vehicles with a given capacity, housed in a single depot
  - set of customers with known demands (all pickups or all deliveries)
- Find vehicle routes such that
  - all customer demands are met
  - each customer is visited by a single vehicle
  - each route starts and ends at the depot
  - each route satisfies vehicle capacity
  - total cost (distance) is minimized

Vehicle routing problem with time windows (VRPTW)

- Given
  - unlimited number of identical vehicles with a given capacity, housed in a single depot
  - set of customers with known demands
  - a hard time window on service start time for each customer
    - Infeasible if upper bound is exceeded
    - Must wait until lower bound if vehicle arrives too early
- Find vehicle routes such that
  - all customer demands are met
  - each customer is visited by a single vehicle
  - each route starts and ends at the depot
  - each route satisfies vehicle capacity and time windows
  - total cost (distance) is minimized

Example of a solution for the VRPTW



Set Partitioning Formulation (SPF)

$$\min \sum_{r\in\Omega} c_r\lambda_r \qquad (1) \quad \text{total cost}$$

$$\text{s.t.} \sum_{r\in\Omega} a_i^r\lambda_r = 1, \qquad \forall\, i \in N \qquad (2) \quad \text{visit each customer}$$

$$\lambda_r \in \{0,1\}, \qquad \forall\, r \in \Omega \qquad (3) \quad \text{binary requirements}$$

- Ω is the set of routes, a_i^r is the number of times that customer i appears in route r
- Must be solved by column generation. The set Ω is often relaxed (allowing some non-elementary routes) in order to make the pricing subproblem more tractable

Set Partitioning Formulation

- Even if Ω only contains elementary routes, the linear relaxation of SPF is **not** strong enough for efficient branch-and-price
  - Except when routes are very constrained (e.g., very narrow time windows)
- SPF should be combined with cutting, yielding Branch-Cut-and-Price (BCP) algorithms

Cuts over Edge/Arc Formulations

- Depend of the specific VRP variant:
  - **CVRP:** Rounded Capacity, Strengthened Combs
  - **VRPTW:** 2-Path
- Improve significantly the relaxations. They are **robust**, their dual variables are translated into edge/arc costs in the pricing. Lead to efficient algorithms
- Seems to be exhausted. Really good new cuts not found in the last years
  - Exact separation of 2-Path is not practical for instances with large routes

Cuts over the Set Partitioning Formulation

- Valid for most VRP variants
- Potential for big improvements in the relaxations. However, they are **non-robust**, each added cut makes the pricing subproblem harder, quickly making it intractable

Subset row cuts (SRCs, Jepsen et al., 2008)

Given $C \subseteq N$ and a scalar multiplier p, the (C, p)-Subset Row Cut is:

$$\sum_{r \in \Omega} \left\lfloor p \sum_{i \in C} a_i^r \right\rfloor \lambda_r \leq \lfloor p|C| \rfloor \qquad (4)$$

Non-robust cut obtained by a Chvátal-Gomory rounding of $|C|$ constraints (2) in the SPF

3-SRCs: subsets of three customers ($|S| = 3$, $p = 1/2$)

$$\sum_{r \in \Omega} a_C^r \lambda_r \leq 1, \qquad \forall\, C \subseteq N, |C| = 3$$

where $a_C^r = 1$ if route r visits 2 or 3 customers in C

Subset row cuts (SRCs, Jepsen et al., 2008)

Given $C \subseteq N$ and a scalar multiplier p, the (C, p)-Subset Row Cut is:

$$\sum_{r \in \Omega} \left\lfloor p \sum_{i \in C} a_i^r \right\rfloor \lambda_r \leq \lfloor p|C| \rfloor \qquad (4)$$

Non-robust cut obtained by a Chvátal-Gomory rounding of $|C|$ constraints (2) in the SPF

3-SRCs: subsets of three customers ($|S| = 3$, $p = 1/2$)

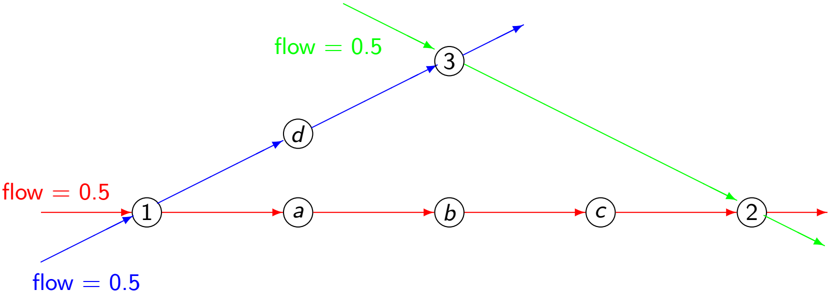
$$\sum_{r \in \Omega} a_C^r \lambda_r \leq 1, \qquad \forall\, C \subseteq N, |C| = 3$$

where $a_C^r = 1$ if route r visits 2 or 3 customers in C

Example of a 3-SRC

- $S = \{1, 2, 3\}$
- Covered by three paths with a flow of 0.5

$$\lambda_r + \lambda_b + \lambda_g \leq 1$$



3-SRCs

- Very efficient to close integrality gap
- Costly to handle in the labeling algorithm
  - For each SRC, you must keep track of the number of visited customers in S
  - Dominance rule is modified and less labels are dominated

Limited-memory 3-SRCs (Pecin et al., 2014)

- Weaker version
- λ_r receives coefficient 1 if
  - Route r visits at least two customers i_1 and i_2 in S
  - Nodes between i_1 and i_2 belong to a pre-specified subset of customers (the memory)
- In the previous example, memory $= \{1, 2, 3, a, b, c, d\}$

3-SRCs

- Very efficient to close integrality gap
- Costly to handle in the labeling algorithm
  - For each SRC, you must keep track of the number of visited customers in S
  - Dominance rule is modified and less labels are dominated

Limited-memory 3-SRCs (Pecin et al., 2014)

- Weaker version
- λ_r receives coefficient 1 if
  - Route r visits at least two customers i_1 and i_2 in S
  - Nodes between i_1 and i_2 belong to a pre-specified subset of customers (the memory)
- In the previous example, memory $= \{1, 2, 3, a, b, c, d\}$

Why it is good to reduce the set M as much as possible?

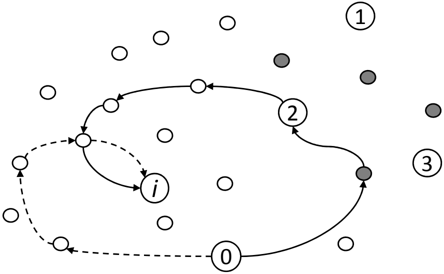


Figure: Solid path may only dominate the dashed path because the 3-SRC $\{1, 2, 3\}$ is already forgotten at customer i.

Limited Memory Subset Row Cuts (lm-SRCs)

Given $C \subseteq V_+$, a memory set M, $C \subseteq M \subseteq V_+$, and a scalar multiplier p, the limited memory (C, M, p)-Subset Row Cut is:

$$\sum_{r \in \Omega} \alpha(C, M, p, r)\lambda_r \leq \lfloor p|C| \rfloor \,, \tag{5}$$

where the coefficient of a route r is computed as:

function $\alpha(C,\ M,\ p,\ r)$
$\quad$ *coeff* $\leftarrow 0$, *state* $\leftarrow 0$
$\quad$ **for every vertex** $i \in r$ (in order) **do**
$\quad\quad$ **if** $i \notin M$ **then**
$\quad\quad\quad$ *state* $\leftarrow 0$
$\quad\quad$ **else if** $i \in C$ **then**
$\quad\quad\quad$ *state* $\leftarrow$ *state* $+ p$
$\quad\quad\quad$ **if** *state* ≥ 1 **then**
$\quad\quad\quad\quad$ *coeff* $\leftarrow$ *coeff* $+ 1$, *state* $\leftarrow$ *state* $- 1$
return *coeff*

Limited Memory Subset Row Cuts (lm-SRCs)

function $\alpha(C, M, p, r)$
  $\textit{coeff} \leftarrow 0$, $\textit{state} \leftarrow 0$
  for every vertex $i \in r$ (in order) **do**
    if $i \notin M$ **then**
      $\textit{state} \leftarrow 0$
    else if $i \in C$ **then**
      $\textit{state} \leftarrow \textit{state} + p$
      if $\textit{state} \geq 1$ **then**
        $\textit{coeff} \leftarrow \textit{coeff} + 1$, $\textit{state} \leftarrow \textit{state} - 1$
return *coeff*

- If $M = V_+$, the function returns $\lfloor p \sum_{i \in C} a_i^r \rfloor$
- Otherwise, the lm-SRC may be a weakening of the corresponding SRC

Main components of the BCP

- Pricing
  - bidirectional search
  - completion bounds + DSSR
  - fast and effective heuristics
- Variable Fixing
  - on arc $(i, j)^d$ for the CVRP and arc (i, j) for VRPTW
- Strong Branching
  - 3 levels, uses history of past branchings
- Route enumeration + Inspection pricing + MIP
  - from this point, cliques and SRCs are separeted
- Non-robustness control

Overall Results for (Pecin et al., 2014)

		LLE04			FLL+06			BCM08			BMR11		
Class	NP	Opt	Gap	T(s)	Opt	Gap	T(s)	Opt	Gap	T(s)	Opt	Gap	T(s)
A	22	15	2.06	6638	22	0.81	1961	22	0.2	118	22	0.13	30
B	20	19	0.61	8178	20	0.47	4763	20	0.16	417	20	0.06	67
E-M	12	3	2.10	39592	9	1.19	126987	8	0.69	1025	9	0.49	303
F	3	3	0.06	1046	3	0.14	2398				2	0.11	164
P	24	16	2.26	11219	24	0.76	2892	22	0.28	187	24	0.23	85
Total	81	56			78			72			77		
Processor		Celeron 700MHz (?)			Pent. 4 2.4GHz(561)			Pent. 4 2.6GHz(624)			X7350 2.93GHz(1108)		

		Con12			CM14			Rop12			PPPU14		
Class	NP	Opt	Gap	T(s)	Opt	Gap	T(s)	Opt	Gap	T(s)	Opt	Gap	T(s)
A	22	22	0.07	59	22	0.09	59	22	0.57	53	22	0.03	5.6
B	20	20	0.05	89	20	0.08	34	20	0.25	208	20	0.04	6.2
E-M	12	10	0.30	2807	10	0.27	1548	10	0.96	44295	12	0.19	3669
F	3	2	0.06	3	3	0.03	27722	3	0.25	2163	3	0.00	3679
P	24	24	0.13	43	24	0.18	240	24	0.69	280	24	0.07	32.7
Total	81	78			79			79			81		
Processor		E5462 2.8GHz(1204)			E5462 2.8GHz(1204)			i7-2620M 2.7GHz(1563)			E5-2667 3.3GHz(1996)		

Detailed Results: M-n151-k12

Algo	Machine	Root LB	Final LB	Total Time
BMR11	X7350 2.93GHz	1004.3	1004.3	380
Contardo12	E5462 2.8GHz	1012.5	**1015**	19699
CM14	E5462 2.8GHz	1012.6	**1015**	10110
Ropke12	i7-2620M 2.7GHz	1001.5	**1015**	417146
PPPU14	E5-2667 3.3GHz	1013.0	**1015**	186

Detailed Results: M-n200-k17

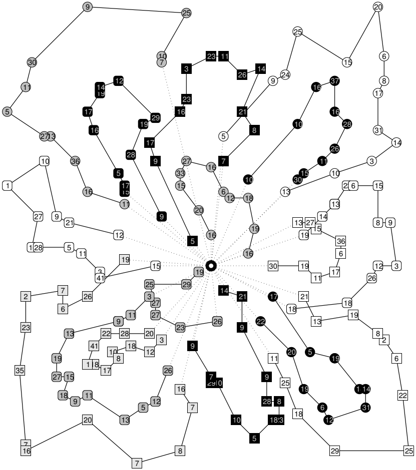
Algo	Machine	Root LB	Final LB	Total Time
BMR11	X7350 2.93GHz	1258.7	1258.7	436
Contardo12	E5462 2.8GHz	1265.1	1265.1	34350
CM14	E5462 2.8GHz	1269.2	1269.2	432000
Ropke12	i7-2620M 2.7GHz	1255.3	1261.4	7200
PPPU14	E5-2667 3.3GHz	1268.8	**1275**	3479

Detailed Results: M-n200-k16

Algo	Machine	Root LB	Final LB	Total Time
BMR11	X7350 2.93GHz	1256.6	1256.6	319
Contardo12	E5462 2.8GHz	1263.0	1263.0	265588
CM14	E5-2667 3.3GHz	1266.9	1266.9	432000
Ropke12	i7-2620M 2.7GHz	1253.0	1258.2	7200
PPPU14	E5-2667 3.3GHz	1266.6	**1274**	28620

Previous upper bound: 1278

Optimal solution of M-n200-k16 ($Q = 200$), cost 1274



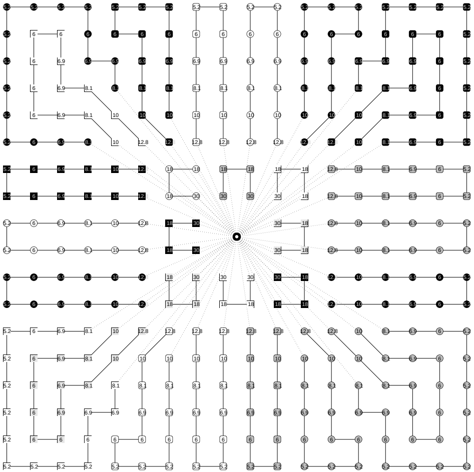
Golden Instances

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

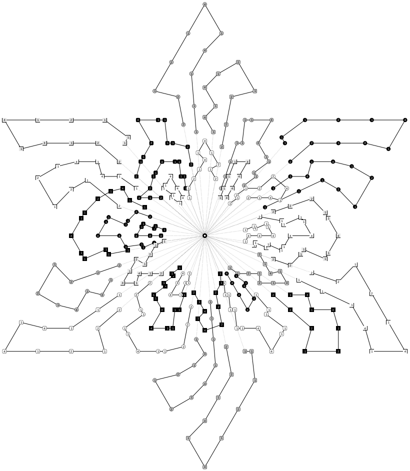
- Appear frequently in the literature on heuristic methods
- Until now, considered to be far beyond the reach of exact methods

(Pecin et al., 2014) could solve **four instances**, those with 240, 300, 320, and 360 customers.

Optimal solution of Golden_14 (320 customers, $Q = 1000$), cost 1080.55, 30 routes



Optimal solution of Golden_19 (360 customers, $Q = 20$), cost 1365.60, 33 routes



Comparison over hard instances (Pecin et al., 2014)

Instance	Algo	IUB	RLB1	RLB2	RT(s)	FLB	Nds	TT(s)
M-n151-k12	BMR11	1015	1004.3	-	380	1004.3	1	380
	Con12	1015	1008.9	1012.5	19041	**1015**	1	19699
	Rop12	1015	1001.5	-		**1015**	5268	417146
	CM14	1015	1011.2	1012.6	9942	**1015**	1	10110
	PPPU14	1015	1011.2	1013.0	180	**1015**	1	186
M-n200-k16	BMR11		1256.6	-	319	1256.6	1	319
	Con12	1278	1263.0	-	265589	1263.0	1	265589
	Rop12	1278	1253.0	-		1258.2	106	7200
	CM14	1278	1266.9	-	432000	1266.9	1	432000
	PPPU14	1278	1266.6	-	807	**1274**	75	28620
M-n200-k17	BMR11	1275	1258.7	-	436	1258.7	1	436
	Con12	1275	1265.1	-	34351	1265.1	1	34351
	Rop12	1276	1255.3	-		1261.4	144	7200
	CM14	1275	1269.2	-	432000	1269.2	1	432000
	PPPU14	1275	1268.8	-	695	**1275**	7	3479
G17 (240)	PPPU14	707.76	705.01	-	345	**707.76**	15	14693
G13 (252)	PPPU14	857.19	851.63	-	4803	851.63	1	4803
G9 (255)	PPPU14	579.71	576.86	-	9013	576.86	1	9013
G18 (300)	PPPU14	995.13	993.22	-	381	**995.13**	23	13769
G14 (320)	PPPU14	1080.55	1076.10	-	8124	**1080.55**	1175	$\approx$ 39 days
G10 (323)	PPPU14	736.26	731.28	-	25689	731.28	1	25689
G19 (360)	PPPU14	1365.60	1362.76	-	467	**1365.60**	339	162405

Improvements from (Pecin et al., 2015)

- (Pecin et al., 2015) present a computational polyhedral study to find the best possible inequalities that can be obtained from up to 5 rows of a SPF
- Experiments with classical CVRP instances show that the newly discovered multipliers lead to significantly improved dual bounds

Improvements from (Pecin et al., 2015)

Average gaps over a set of hard instances ranging from 36 to 199 customers. Full separation until convergence:

	Gap(%)
Only CG (elementary routes)	2.63
+ robust cuts	0.98
+ 3SRCs	0.35
+ 4SRCs + 5SRCs	0.24
Rank 1 Cuts up to 5 rows	0.17

The new cuts removed 30% of the residual gap. They can help to solve some larger open instances.

Golden_20 (420 customers)

Authors	BKS
(Vidal et al., 2012)	1818.32
(Groër et al., 2011)	1818.25
(Jin et al., 2014)	1817.89
(Liu and Li, 2014)	1817.86

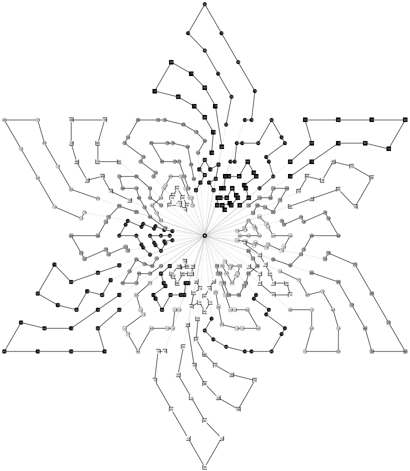
- Optimal solution: 1817.59
- Root LB: 1815.0 (1200 active Rank 1 cuts!)
  - Pecin et al., 2014 reaches the Root LB of 1814.4
- B&B Nodes: 370
- Total Time: 7 days (single core i7-3960X 3.30GHz)
  - We estimate between 2 or 3 months for (Pecin et al., 2014)

Golden_20 (420 customers)

Authors	BKS
(Vidal et al., 2012)	1818.32
(Groër et al., 2011)	1818.25
(Jin et al., 2014)	1817.89
(Liu and Li, 2014)	1817.86

- Optimal solution: 1817.59
- Root LB: 1815.0 (1200 active Rank 1 cuts!)
  - Pecin et al., 2014 reaches the Root LB of 1814.4
- B&B Nodes: 370
- Total Time: 7 days (single core i7-3960X 3.30GHz)
  - We estimate between 2 or 3 months for (Pecin et al., 2014)

Optimal solution of Golden_20, cost 1817.59



Improvements from (Pecin et al., 2016)

- New memory definition: Arc memory
- In the previous example, memory = $\{(1, a), (a, b), (b, c), (c, 2), (1, d), (d, 3), (3, 2)\}$
- Experiments with CVRP and VRPTW show that the **arc-memory can be decisive in some very hard instances**

Improvements from (Pecin et al., 2016)

Arc-memory for CVRP

Ins	UB	Arc-memory			Node-memory		
		LB	PT	TT	LB	PT	TT
G13 (252)	857.19	852.65	31	10674	851.89	41	10507

- Final Root LB: 853.0; B&B Nodes: 275; Total Time: 4.3 days

Arc-memory for VRPTW

Ins	UB	Arc-memory			Node-memory		
		LB	PT	TT	LB	PT	TT
R208 (100)	701.1	697.5	70	9621	697.3	796	63864
C2-2-3 (200)	1763.5	1747.0	24	12849	1747.1	536	74973
RC2-2-8 (200)	2151.2	2150.3	99	83057	2150.3	82	147585
C2-2-4 (200)	1695.0	1664.7	367	128684	1656.6	364	135024
RC2-2-9 (200)	2086.7	2069.6	213	275225	2065.6	295	270216
RC2-210 (200)	1989.3	1966.5	284	335007	1962.0	486	347495

Improvements from (Pecin et al., 2016)

Arc-memory for CVRP

Ins	UB	Arc-memory			Node-memory		
		LB	PT	TT	LB	PT	TT
G13 (252)	857.19	852.65	31	10674	851.89	41	10507

- Final Root LB: 853.0; B&B Nodes: 275; Total Time: 4.3 days

Arc-memory for VRPTW

Ins	UB	Arc-memory			Node-memory		
		LB	PT	TT	LB	PT	TT
R208 (100)	701.1	697.5	70	9621	697.3	796	63864
C2-2-3 (200)	1763.5	1747.0	24	12849	1747.1	536	74973
RC2-2-8 (200)	2151.2	2150.3	99	83057	2150.3	82	147585
C2-2-4 (200)	1695.0	1664.7	367	128684	1656.6	364	135024
RC2-2-9 (200)	2086.7	2069.6	213	275225	2065.6	295	270216
RC2-210 (200)	1989.3	1966.5	284	335007	1962.0	486	347495

100-customer Solomon instances (VRPTW)

Class	NI	JPSP08		DLH08		BMR11		Rop12		PCDU16	
		Opt	Time	Opt	Time	Opt	Time	Opt	Time	Opt	Time
C1	9	9	468	9	18	9	25	9	15	9	15
RC1	8	8	11004	8	2150	8	276	8	2907	8	52
R1	12	12	27412	12	2327	12	251	12	2040	12	31
C2	8	7	2795	8	2093	8	40	8	209	8	328
RC2	8	5	3204	6	15394	8	3767	8	2205	8	337
R2	11	4	35292	8	63068	10	28680	11	30592	11	6432
Total	56	45		51		55		56		56	
Average			13288		12920		5867		7209		1375
Processor		Opteron 2.6GHz		Pentium 4 3GHz		X7350 2.93GHz		i7-2620M 2.7GHz		X-ES2637 3.5GHz	

200-customer Gehring and Homberger instances (VRPTW)

Class	Opt	All-root			Solved					Unsolved	
		RGP	GP	T	RGP	GP	T	B	TT	Max	Min
C1	10	0.10	0.04	88	0.10	0.04	88	1.60	133	-	-
RC1	8	1.68	0.57	4674	1.56	0.45	3889	46.00	38940	1.27	0.82
R1	10	1.24	0.25	1451	1.24	0.25	1451	25.20	12854	-	-
C2	8	0.83	0.24	20552	0.54	0.00	4096	1.00	4096	1.75	0.64
RC2	7	1.29	0.24	123251	0.77	0.05	26362	1.57	27375	1.13	0.28
R2	8	0.86	0.08	59503	0.79	0.00	13397	1.25	13645	0.59	0.18
Total	51										
Average		1.00	0.24	34920	0.82	0.13	7274	13.03	15195		

Conclusions

- Non-robust cuts should be designed in order to minimize their impact on the specific kind of algorithm used in the pricing

- Limited-memory subset-row cuts together with several recent ideas for the CVRP could more than double the size of the largest instance solved so far
  - All the 81 instances from the traditional benchmark now are solved
  - 6 Golden et al. instances ranging from 240 to 420 customers solved

- Lm-SRCs also proved to be very useful to VRPTW instances
  - All the 56 100-customer Solomon instances now well-solved
  - 51 out of 60 200-customer instances solved

Conclusions

- Non-robust cuts should be designed in order to minimize their impact on the specific kind of algorithm used in the pricing
- Limited-memory subset-row cuts together with several recent ideas for the CVRP could more than double the size of the largest instance solved so far
    - All the 81 instances from the traditional benchmark now are solved
    - 6 Golden et al. instances ranging from 240 to 420 customers solved
- Lm-SRCs also proved to be very useful to VRPTW instances
    - All the 56 100-customer Solomon instances now well-solved
    - 51 out of 60 200-customer instances solved

Conclusions

- Non-robust cuts should be designed in order to minimize their impact on the specific kind of algorithm used in the pricing
- Limited-memory subset-row cuts together with several recent ideas for the CVRP could more than double the size of the largest instance solved so far
  - All the 81 instances from the traditional benchmark now are solved
  - 6 Golden et al. instances ranging from 240 to 420 customers solved
- Lm-SRCs also proved to be very useful to VRPTW instances
  - All the 56 100-customer Solomon instances now well-solved
  - 51 out of 60 200-customer instances solved

Publications

- Pecin, D.; Pessoa, A.; Poggi, M.; Uchoa, E. – **Improved Branch-Cut-and-Price for Capacitated Vehicle Routing**. Integer Programming and Combinatorial Optimization (IPCO), June 2014 (full paper to appear on Mathematical Programing Computational - MPC)
- Pecin, D.; Pessoa, A.; Poggi, M.; Uchoa, E.; H. Santos – **Limited Memory Rank-1 Cuts for Vehicle Routing Problems**. Presentation at the 19th Combinatorial Optimization Workshop, Aussois, France, 2015. Paper Submitted.
- Pecin, D.; Contardo, C.; Desaulniers, G.; Uchoa, E. – **New Enhancements for the Exact Solution of the Vehicle Routing Problem with Time Windows**. Cahiers du GERAD (G-2016-13), 2016. Paper submitted.

Obrigado!
Questions?