Cut-First Branch-and-Price-Second for the CARP

Workshop on Large Scale Optimization 2012
Vevey, Switzerland

Claudia Bode and Stefan Irnich
{claudia.bode,irnich}@uni-mainz.de
Chair for Logistics Management
Gutenberg School of Management and Economics
JOHANNES GUTENBERG UNIVERSITÄT MAINZ

November 21-22, 2012

Capacitated Arc-Routing Problem (CARP)

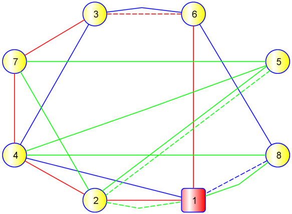
   

Notation: Defined on undirected Graph $G = (V, E)$

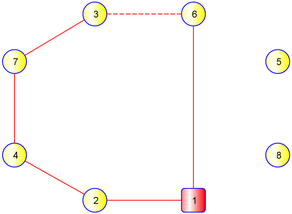
- demands $q_e \geq 0$ on edges $e \in E$
- required edges $E_R = \{e \in E : q_e > 0\} \subseteq E$
- K homogeneous vehicles stationed at depot d with capacity Q
- c_e cost for traversing through $e \in E$

Task: Find K minimum cost vehicles tours which start and end at depot d, service all required edges E_R, and respect the vehicle capacity Q.

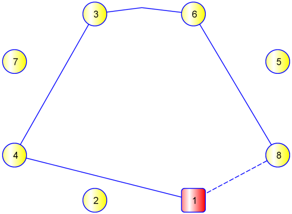
CARP solutions



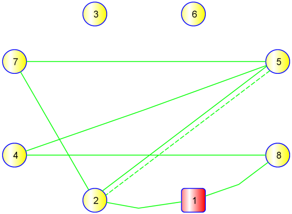
Complete solution



Route 1



Route 2



Route 3

Letchford, A. N. and Oukil, A. (2009). Exploiting sparsity in pricing routines for the capacitated arc routing problem, *Computers & Operations Research* **36**(7), 2320–2327.

- An approach that fully **exploits sparsity** of CARP instances should be superior
- Drawbacks of the proposed CG algorithm:
  1. Pure set-partitioning master program delivers worse bounds than known compact formulations
  2. Missing branching scheme

Branch-and-price for CARP is different than the node-routing case!

Two types of exact approaches:

1. Full exact methods determine an optimal integer solution and prove optimality by showing that its cost is a lower bound.
2. LB-based methods use heuristic solutions (often not computed within the presented approach) to prove optimality by showing that a computed lower bound matches with the heuristic's upper bound.

Methods presented in the Literature:

- **Two-Index Formulation** (Belenguer and Benavent, 1998)
- **One-Index Formulation** (Letchford, 1997; Belenguer and Benavent, 1998, 2003)
- **Set-Partitioning Formulation** (Gómez-Cabrero *et al.*, 2005)
- Transformation into Node-Routing Problem

Cut-First Branch-and-Price-Second

Key ideas:

- use strong LB from one-index formulation
- use aggregated set-partitioning formulation to avoid symmetric solutions
- exploit sparsity of real-world CARP graph;
  fast pricing on sparse network

Difficulties that had to be solved:

- negative reduced costs on deadheading edges (→dual optimal inequalities; Ben Amor *et al.* (2006))
- integer variables of aggregated formulation alone do not ensure integrality of tours
  (→complex branching scheme)

Overview: Cut-first bap-second

Input: CARP Instance

1. Phase 1: Cut
   Solve LP-relaxation of one-index formulation using a cutting-plane algorithm
   Let $\mathcal{S}$ be the set of active cuts at the end
   (odd-set, capacity, and disjoint-path inequalities)
2. Phase 2: Branch-and-Price
   Solve integer master program using CG und B&B
   Initialize MP with cuts $\mathcal{S}$

Output: Integer solution

Master problem derived from Dantzig-Wolfe decomposition of two-index formulation + cuts from one-index formulation + aggregation over vehicles:

$$
\begin{aligned}
\min \quad & \sum_{r \in \Omega} c_r^\top \lambda_r \\
\text{s.t.} \quad & \sum_{r \in \Omega} \bar{x}_{er} \lambda_r = 1 \quad \text{for all } e \in E_R \qquad (\text{dual } \pi_e) \\
& \sum_{r \in \Omega} d_{sr} \lambda_r \geq r_s \quad \text{for all } s \in \mathcal{S} \qquad (\text{dual } \beta_s) \\
& \sum_{r \in \Omega} \lambda_r = K \qquad (\text{dual } \mu) \\
& \lambda \geq \mathbf{0} \quad (\in \mathbb{R}^{|\Omega|})
\end{aligned}
$$

Reduced Costs (rdc) can be transformed back onto variables of the compact formulation:

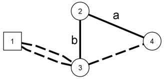
$$\tilde{c}_e^{serv} = c_e^{serv} - \pi_e \text{ for all } e \in E_R \qquad \text{and} \qquad \tilde{c}_e = c_e - \sum_{s \in \mathcal{S}} d_{es}\beta_s \text{ for all } e \in E.$$

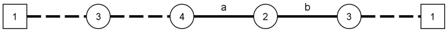
Dual optimality inequalities guarantee non-negative rdc $\tilde{c}_e \geq 0$ for deadheading (Bode and Irnich, 2012).

Pricing Problem:

$$\begin{aligned} \min \quad & \tilde{c}^{serv,\top} x + \tilde{c}^\top y - \mu \\ \text{s.t.} \quad & x(\delta_R(i)) + y(\delta(i)) = 2p_i \quad \text{for all } i \in V \\ & q^\top x \leq Q \\ & x(\delta_R(S)) + y(\delta(S)) \geq 2x_f \quad \text{for all } S \subseteq V \setminus \{d\},\ f \in E_R(S) \\ & p \in \mathbb{Z}_+^{|V|}, y \in \mathbb{Z}_+^{|E|}, x \in \{0,1\}^{|E_R|}, \end{aligned}$$

Example: $x_{23} = x_{24} = 1, y_{13} = 2, y_{34} = 1$ implies tour $1 - 3 - 4 \overset{a}{=} 2 \overset{b}{=} 3 - 1$





Pricing can be done on the **original network** (exploit sparsity, Letchford and Oukil (2009))

Extension along deadheading edges can be done with the Dijkstra Algorithm due to non-negative reduced costs $\tilde{c}_e \geq 0$.

Algorithm 1: Efficient Pricing Algorithm $\mathcal{O}\left(Q \cdot (|E| + |V| \log |V|)\right)$

for $q = 0, 1, 2, \ldots, Q$ **do**

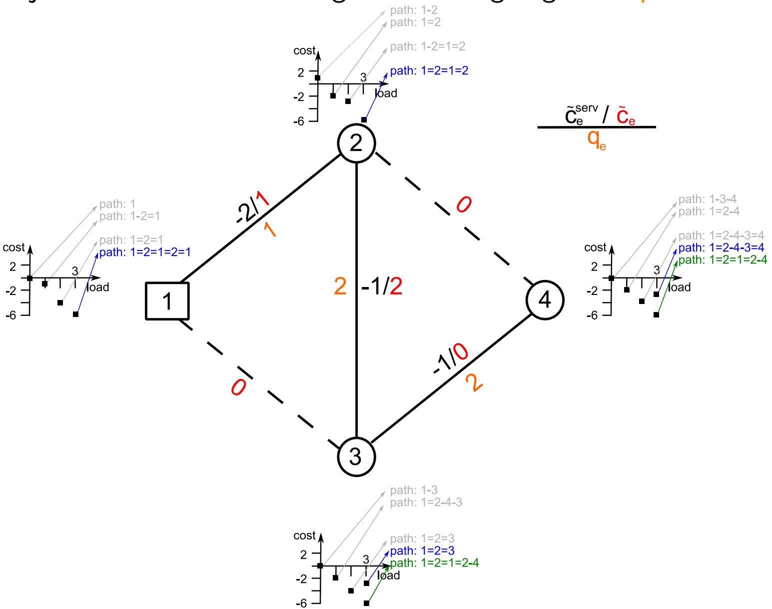
```
// Dijkstra-like extension
//    use rdc c̃_e ≥ 0
//    this creates new labels with identical load q
```

Extend labels (c, q) along deadheading edges and apply dominance algo

```
// Service extension
//    use rdc c̃_e^service
//    this creates only labels with higher load q' > q
```

Extend labels (c, q) along service edges and apply dominance algo

Dijkstra-like Extension along deadheading edges for $q = 3$



Pricing Relaxations

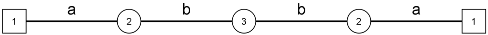
A tour is elementary iff no edge is serviced more than once.

- Nodes may be visited more than once.
- Edges may be traversed more than once, but only serviced once.

Tradeoff: The stronger the relaxation, the better the master program LB, but the harder the pricing problem.

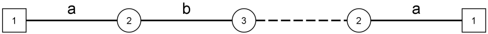
Non-elementary:

feasible:

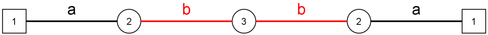


2-loop free:

feasible:



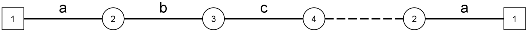
infeasible:



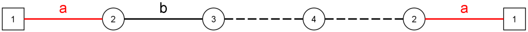
Pricing Relaxations

3-loop free:

feasible:

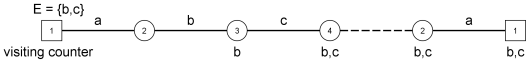


infeasible:

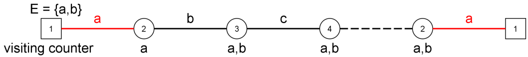


Partial elementary w.r.t. a set $\mathcal{E} \subseteq E_R$:

feasible w.r.t. $\mathcal{E} = \{b, c\}$:

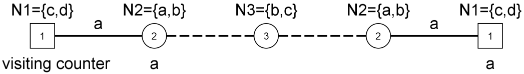


infeasible w.r.t. $\mathcal{E} = \{a, b\}$:

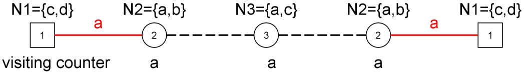


NG-route:

feasible:



infeasible:



Pricing Relaxations

Possible Relaxations (elementary problem is strongly $\mathcal{NP}$-hard):

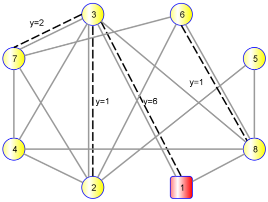
Relaxation	Efficiency Worst Case	Who?
Non-elementary	$\mathcal{O}\left(Q(\lvert E\rvert + \lvert V\rvert \log \lvert V\rvert)\right)$	Letchford and Oukil (2009)
2-loop free = 1-cycle free	Factor 2	based on Houck *et al.* (1980) and Benavent *et al.* (1992)
k-loop free for $k \geq 3$	Factor $k! \cdot (k-1)!$	based on Irnich and Villeneuve (2006)
partial elementarity w.r.t. $\mathcal{E} \subset E_R$	Factor $2^{\lvert\mathcal{E}\rvert}$	Desaulniers *et al.* (2008)
NG-route relaxation w.r.t. $N_i \subset E_R, i \in V$	Factor $2^{\max_i \lvert N_i\rvert}$	Baldacci *et al.* (2009)

Branching

Hierarchical branching scheme with **3 Types of Branching Rules:**

1. Branching on node degrees
   → select node with non-even node degree
2. Branching on edge flows
   → select node with fractional edge flow

Integral x and y variables do **not** guarantee integral route variables λ



Example: '=' is service and '–' is deadheading

Tour 1 (1=3=8=6–8=3–1)	$\lambda_1 = \frac{1}{3}$
Tour 2 (1=8=6–8=5=2=7–3–1)	$\lambda_2 = \frac{1}{3}$
Tour 3 (1–3–2=4=7=3–1)	$\lambda_3 = 1$
Tour 4 (1–3=4=8=3–1)	$\lambda_4 = \frac{1}{3}$
Tour 5 (1–3=2=6=7–3–1)	$\lambda_5 = 1$
Tour 6 (1=3–7=2=5=8=1)	$\lambda_6 = \frac{2}{3}$
Tour 7 (1–3=4=8=6–8=4=3–1)	$\lambda_7 = \frac{1}{3}$

⇒ 3rd class of branching rule is required

3. Branching on followers and non-followers

Branching on Follower Information

Example (cont'd): '=' is service and '–' is deadheading

Tour 1 (1=3=8=6–8=3–1) $\lambda_1 = \frac{1}{3}$
Tour 2 (1=8=6–8=5=2=7–3–1) $\lambda_2 = \frac{1}{3}$
Tour 3 (1–3–2=4=7=3–1) $\lambda_3 = 1$
Tour 4 (1–3=4=8=3–1) $\lambda_4 = \frac{1}{3}$
Tour 5 (1–3=2=6=7–3–1) $\lambda_5 = 1$
Tour 6 (1=3–7=2=5=8=1) $\lambda_6 = \frac{2}{3}$
Tour 7 (1–3=4=8=6–8=4=3–1) $\lambda_7 = \frac{1}{3}$

Consider edges $e = \{1, 3\}$ and $e' = \{3, 8\}$.

→ in $\frac{1}{3}$ of the cases e and e' are direct followers (Tour 1)

→ in $\frac{2}{3}$ of the cases e and e' are not direct followers (Tour 6)

Branching on Follower Information

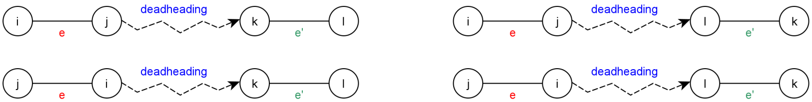
Branching on (non-)followers can be handled by **Network Modifications**:

- Deletions and/or additions of required edges
- Basic structure of the network remains unchanged

Let e and e' be two edges with fractional follower information.

Non-Follower branch: Forbid the consecutive service of these two edges
$\rightarrow$ **Associate same task** with e and e' and perform **2-loop elimination**

Follower branch: Enforce consecutive service
$\rightarrow$ Replace e and e' by **four new edges** modeling consecutive service



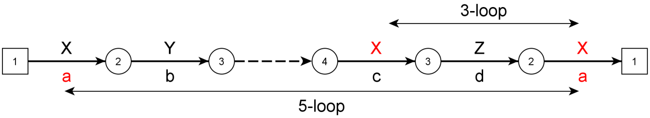
Branching scheme needs pricing problem relaxation that is able to handle two sets of tasks:

Tasks $\mathcal{T}^B$ for branching (2-loop-free tours)

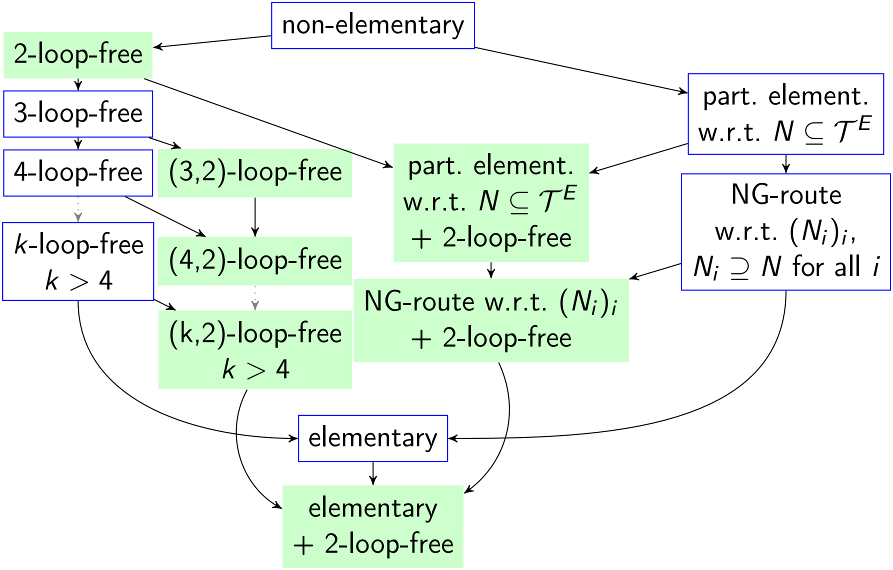
Tasks $\mathcal{T}^E$ for (approximating) elementary routes
(*k*-loop free, partial elementary, NG-route)

Example:

$\mathcal{T}^B = \{X, Y, Z, \dots\}$
$\mathcal{T}^E = \{a, b, c, \dots\}$



Branching and Pricing Relaxations



Note: weaker → stronger

Theoretical Results for $(k, 2)$-Loop Free Pricing

Possible Relaxations (elementary problem is strongly $\mathcal{NP}$-hard):

Relaxation	Efficiency		Who?
	Relaxation $\mathcal{R}$	Relaxation $(\mathcal{R}, 2)$	
Non-elementary	$\mathcal{O}(Q(\lvert E\rvert + \lvert V\rvert \log \lvert V\rvert))$	-	Letchford and Oukil (2009)
2-loop free $=$ 1-cycle free	Factor 2	Factor 2	based on Houck *et al.* (1980 and Benavent *et al.* (1992)
k-loop free for $k \geq 3$	Factor $k! \cdot (k-1)!$	Factor $(k+1)$	based on Irnich and Villeneuve (2006)
partial elementarity w.r.t. $\mathcal{E} \subset E_R$	Factor $2^{\lvert\mathcal{E}\rvert}$	Factor 2	Desaulniers *et al.* (2008)
NG-route relaxation w.r.t. $N_i \subset E_R, i \in V$	Factor $2^{\max_i \lvert N_i \rvert}$	Factor 2	Baldacci *et al.* (2009)

Acceleration Techniques

Heuristic Pricing:

- Reduced Networks (Elimination of Nodes and Arcs)
- Stronger Dominance
- Scaling
- Metaheuristics
- ...

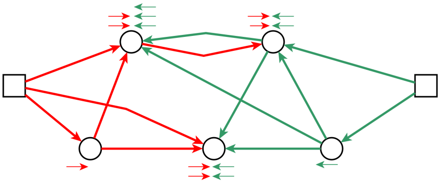
Acceleration of Exact Pricing Algorithms:

- Bidirectional Pricing (Righini and Salani, 2006)
- Bounding (Functions) (Baldacci *et al.*, 2011)

Bidirectional Pricing

Bidirectional Pricing (Righini and Salani, 2006):

1. Propagate FW up to "middle"
2. Propagate BW up to "middle"
3. Merge FW and BW labels

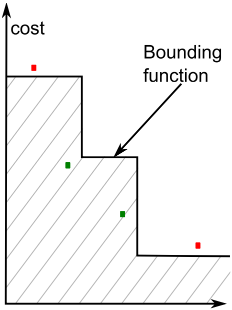


Some Findings:

- In undirected networks there is no difference between forward and backward labels
  - $\Rightarrow$ we save approx. factor 2 in label extension
  - $\Rightarrow$ overall factor 1.5–2; better for $(k, 2)$-loop relaxation
- 2-loop: Simple Merge in $\mathcal{O}(Q)$
  - $\Rightarrow$ guarantees identification of a minimum rdc path
  - $\Rightarrow$ but not all Pareto-optimal paths
- NG-route: Standard half-way test not applicable

Bounding

Bounding is possible with any relaxation of the pricing problem (Baldacci *et al.*, 2011):



Bounding function is computed using (backward) labeling with a proper relaxation

Pricing with	Relaxation for Bounding
2-loop	non-elementary scaled instance
3-loop	non-elementary 2-loop scaled instance
4-loop	non-elementary 2-loop or 3-loop scaled instance
NG-route + 2-loop	2-loop NG-route with subset of neighbors

Note: No Half-way Stop!

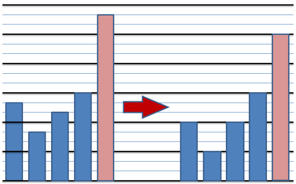
Scaling

Scaling can be used as a . . .

Relaxation:

Rounded down demand and capacity to next integer multiple of a factor f

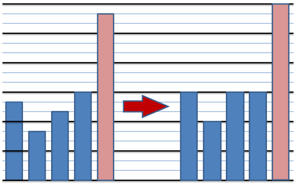
Such a relaxation can be used for bounding.



Restriction:

Rounded up demand and capacity to next integer multiple of a factor f

Can be used as a pricing heuristic.



Computational Setup:

- Computation times do not include Phase 1 (cutting), Time limit $TL = 4$ hours
- Best node first is node selection rule for B&B
- All Pricing Problems with bounding and solved bidirectionally
- NG-neighborhoods are generated dynamically based on iteratively solving the branch-and-price root node
  - all neighborhoods $N_i := \emptyset$ at start
  - select cycle $C = (i_1, i_2, \dots, i_k, i_1)$ with maximum flow
  - first and last serviced/required edges are identical $e = \{i_1, i_2\} = \{i_k, i_1\}$
  - task e is added to neighborhoods $N_{i_1}, N_{i_2}, \dots, N_{i_k}$
  - stop if bound on neighborhood size is exceeded

Best known Bounds

Table : Best Known Bounds for the egl Instances

instance	lb_{best}	computed by	ub_{best}	computed by	*opt*	proved by
egl-e2-b			6317	Brandão and Eglese (2008)	6317	own
egl-e3-b	7744	own	7775	Polacek *et al.* (2008)		
egl-e3-c	10244	Bartolini *et al.* (2012)	10292	Polacek *et al.* (2008)		
egl-e4-a	6408	Bode and Irnich (2012)	6444	Santos *et al.* (2010)		
egl-e4-b	8935	Bartolini *et al.* (2012)	8961	Bartolini *et al.* (2012)		
egl-e4-c	11512	own	11529	own		
egl-s2-a	9825	Bartolini *et al.* (2012)	9884	Santos *et al.* (2010)		
egl-s2-b	13017	Bartolini *et al.* (2012)	13100	Brandão and Eglese (2008)		
egl-s3-a	10165	own	10220	Santos *et al.* (2010)		
egl-s3-b	13648	Bartolini *et al.* (2012)	13682	Polacek *et al.* (2008)		
egl-s4-a	12153	own	12268	Santos *et al.* (2010)		
egl-s4-b	16113	own	16283	Fu *et al.* (2010)		
egl-s4-c	20430	Bartolini *et al.* (2012)	20481	Bartolini *et al.* (2012)		
egl-g1-a	976907	own	1049708	Martinelli *et al.* (2011)		
egl-g1-b	1093884	own	1140692	Martinelli *et al.* (2011)		
egl-g1-c	1212151	own	1282270	Martinelli *et al.* (2011)		
egl-g1-d	1341918	own	1420126	Martinelli *et al.* (2011)		
egl-g1-e	1482176	own	1583133	Martinelli *et al.* (2011)		
egl-g2-a	1067262	own	1129229	Martinelli *et al.* (2011)		
egl-g2-b	1185221	own	1255907	Martinelli *et al.* (2011)		
egl-g2-c	1311339	own	1417145	Martinelli *et al.* (2011)		
egl-g2-d	1446680	own	1516103	Martinelli *et al.* (2011)		
egl-g2-e	1581459	own	1701681	Martinelli *et al.* (2011)		

Best known Bounds

Table : Best Known Bounds for the bmcv Instances, Subsets C and E

instance	lb_{best}	computed by	ub_{best}	computed by	*opt*	proved by
C01	4145	own	4150	Beullens *et al.* (2003)		
C04			3510	Beullens *et al.* (2003)	3510	own
C09	5245	own	5260	Brandão and Eglese (2008)		
C11	4615	own	4630	Mei *et al.* (2009)		
C12	4235	own	4240	Beullens *et al.* (2003)		
C15	4920	own	4940	Beullens *et al.* (2003)		
C18	5580	Bartolini *et al.* (2012)	5620	Santos *et al.* (2010)		
C19			3115	Beullens *et al.* (2003)	3115	own
C21			3970	Beullens *et al.* (2003)	3970	own
C23	4075	own	4085	Beullens *et al.* (2003)		
C24			3400	Beullens *et al.* (2003)	3400	own
E01	4900	own	4910	Brandão and Eglese (2008)		
E09	5805	own	5820	Tang *et al.* (2009)		
E11					4650	own
E15	4200	own	4205	Santos *et al.* (2010)		
E16			3775	Beullens *et al.* (2003)	3775	own
E18			3835	Beullens *et al.* (2003)	3835	own
E19			3235	Beullens *et al.* (2003)	3235	own
E20			2825	Beullens *et al.* (2003)	2825	own
E23			3710	Beullens *et al.* (2003)	3710	own
E24			4020	Beullens *et al.* (2003)	4020	own

Best known Bounds

Table : Best Known Bounds for the bmcv Instances, Subsets D and F

instance	lb_{best}	computed by	ub_{best}	computed by	*opt*	proved by
D08			3045	Beullens *et al.* (2003)	3045	own
D14			3280	Beullens *et al.* (2003)	3280	own
D19			2400	Beullens *et al.* (2003)	2400	own
D21	3005	own	3050	Beullens *et al.* (2003)		
D23			3130	Beullens *et al.* (2003)	3130	own
D24	2705	own	2710	Beullens *et al.* (2003)		
F04			3485	Beullens *et al.* (2003)	3485	own
F08			3705	Beullens *et al.* (2003)	3705	own
F12			3395	Beullens *et al.* (2003)	3395	own
F18	3065	Bartolini *et al.* (2012)	3075	Beullens *et al.* (2003)		
F19	2515	own	2525	Beullens *et al.* (2003)		
F23			3005	Beullens *et al.* (2003)	3005	own

Conclusions and Outlook

Conclusions:

- "Easy" instances: best to be solved with 2-loop or 3-loop relaxation
- "Hard" instances: often best LBs with NG-route relaxation
  - dynamic generation of neighborhoods $N_i \subseteq E_R$
  - maximum neighborhood size 5, 6 or 7 works best
- Speedup by combining Bounding and Bidirectional Pricing gives a speedup by
  - 4-loop: approx. factor 8.0
  - NG-route/3-loop: approx. factor 4
- Partial Elementary is not competitive with NG-route relaxation

Outlook:

1. Detailed experiments with Strong Branching necessary
2. Integration Subset-row inequalities and other non-robust Cuts

Thank you for coming!

Questions?!

Baldacci, R., Mingozzi, A., and Roberti, R. (2009). Solving the vehicle routing problem with time windows using new state space relaxation and pricing strategies. Presented at AIRO 2008, EURO 2009, ISMP 2009, and AIRO 2009.

Baldacci, R., Mingozzi, A., and Roberti, R. (2011). New state-space relaxations for solving the traveling salesman problem with time windows. *INFORMS Journal on Computing*.

Bartolini, E., Cordeau, J.-F., and Laporte, G. (2012). Improved lower bounds and exact algorithm for the capacitated arc routing problem. *Mathematical Programming*, pages 1–44.

Belenguer, J. and Benavent, E. (1998). The capacitated arc routing problem: Valid inequalities and facets. *Computational Optimization and Applications*, **10**(2), 165–187.

Belenguer, J. and Benavent, E. (2003). A cutting plane algorithm for the capacitated arc routing problem. *Computers & Operations Research*, **30**, 705–728.

Ben Amor, H., Desrosiers, J., and Valério de Carvalho, J. (2006). Dual-optimal inequalities for stabilized column generation. *Operations Research*, **54**(3), 454–463.

Benavent, E., Campos, V., Corberán, A., and Mota, E. (1992). The capacitated arc routing problem: Lower bounds. *Networks*, **22**, 669–690.

Beullens, P., Muyldermans, L., Cattrysse, D., and Van Oudheusden, D. (2003). A guided local search heuristic for the capacitated arc routing problem. *European Journal of Operational Research*, **147**(3), 629–643.

Bode, C. and Irnich, S. (2012). Cut-first branch-and-price-second for the capacitated arc-routing problem. *Operations Research*, **(accepted for publication)**. Available at `http://logistik.bwl.uni-mainz.de/158.php`.

Brandão, J. and Eglese, R. (2008). A deterministic tabu search algorithm for the capacitated arc routing problem. *Computers & Operations Research*, **35**(4), 1112–1126.

Desaulniers, G., Lessard, F., and Hadjar, A. (2008). Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, **42**(3), 387–404.

Fu, H., Mei, Y., Tang, K., and Zhu, Y. (2010). Memetic algorithm with heuristic candidate list strategy for capacitated arc routing problem. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE.

Gómez-Cabrero, D., Belenguer, J., and Benavent, E. (2005). Cutting planes and column generation for the capacitated arc routing problem. presented at ORP3, Valencia, Spain.

Houck, D., Picard, J., Queyranne, M., and Vemuganti, R. (1980). The travelling salesman problem as a constrained shortest path problem: Theory and computational experience. *Opsearch*, **17**, 93–109.

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.

Letchford, A. (1997). *Polyhedral results for some arc routing problems*. Phd dissertation, Department of Management Science, Lancaster University.

Letchford, A. N. and Oukil, A. (2009). Exploiting sparsity in pricing routines for the capacitated arc routing problem. *Computers & Operations Research*, **36**(7), 2320–2327.

Martinelli, R., Pecin, D., Poggi, M., and Longo, H. (2011). A branch-cut-and-price algorithm for the capacitated arc routing problem. In P. Pardalos and S. Rebennack, editors, *Experimental Algorithms*, volume 6630 of *Lecture Notes in Computer Science*, pages 315–326. Springer Berlin / Heidelberg.

Mei, Y., Tang, K., and Yao, X. (2009). A global repair operator for capacitated arc routing problem. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, **39**(3), 723–734.

Polacek, M., Doerner, K., Hartl, R., and Maniezzo, V. (2008). A variable neighborhood search for the capacitated arc routing problem with intermediate facilities. *Journal of Heuristics*, **14**, 405–423. 10.1007/s10732-007-9050-2.

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.

Santos, L., Coutinho-Rodrigues, J., and Current, J. R. (2010). An improved ant colony optimization based algorithm for the capacitated arc routing problem. *Transportation Research Part B*, **44**(2), 246 – 266.

Tang, K., Mei, Y., and Yao, X. (2009). Memetic algorithm with extended neighborhood search for capacitated arc routing problems. *Evolutionary Computation, IEEE Transactions on*, **13**(5), 1151–1166.

Network Modification

Modifications become more intricate if several follower and non-follower constraints are active.

Example: Active follower decisions $F = \{(e_1, e_2), (e_3, e_4)\}$ and non-follower decisions $N = \{(e_1, e_3)\}$.

F and N partition E_R and one partition is $\{e_1, e_2, e_3, e_4\}$.

The set of all possible subsequences s is

$$\{(e_1, e_2), (e_3, e_4), (e_1, e_2, e_3, e_4), (e_1, e_2, e_4, e_3), (e_2, e_1, e_4, e_3)\}$$

These subsequences

- are represented by 4 edges each
- cost of a subsequence can be determined solving a DP