

Robust Branch-and-Cut-and-Price for the CMST Problem and Extended Capacity Cuts

Eduardo Uchoa, Engenharia de Produção, UFF, Brazil.

Ricardo Fukasawa, ISYE, Georgia Tech, USA.

Jens Lysgaard, Aarhus School of Business, Denmark.

Marcus Poggi de Aragão, Informática, PUC-Rio, Brazil.

Diogo Andrade, RUTCOR, Rutgers, USA.

Part I: Context

Robust branch-and-cut-and-price

- A branch-and-bound algorithm where:
 - Each node can solve a linear program with an exponential number of both variables and constraints,
 - Neither branching nor separation (cutting) ever increase the complexity of the pricing subproblem.

Robust branch-and-cut-and-price

A number of researchers noted that cuts expressed in terms of variables from a suitable original formulation do not disturb the pricing and constructed the first robust BCPs:

Vanderbeck (1998) - lot-sizing.

Kim, Barnhart, Ware, Reinhardt (1999) - a network design problem.

Kohl, Desrosiers, Madsen, Solomon, Soumis (1999) - VRPTW.

Van der Akker, Hurkens, Savelsbergh (2000) - a scheduling problem.

Felici, Gentile, Rinaldi (2000) - a supply-chain problem.

Barnhart, Hane, Vance (2001) - integer multiflow.

Robust Branch-and-Cut-and-Price

E

Our group obtained good results with RBCP in the following problems:

Capacitated Minimum Spanning Tree – Fukasawa, Porto, Poggi de Aragão, Uchoa (2003).

Generalized Assignment – Pigatti, Poggi de Aragão, Uchoa (2003).

Capacitated Vehicle Routing – Fukasawa, Lysgaard, Poggi de Aragão, Reis, Uchoa, Werneck (2004).

Capacitated Arc Routing – Longo, Poggi de Aragão, Uchoa (2004).

Bin Packing/Cutting Stock – Belov, Letchford, Uchoa (2005).

The Capacitated Minimum Spanning Tree Problem (CMST)

■ Given:

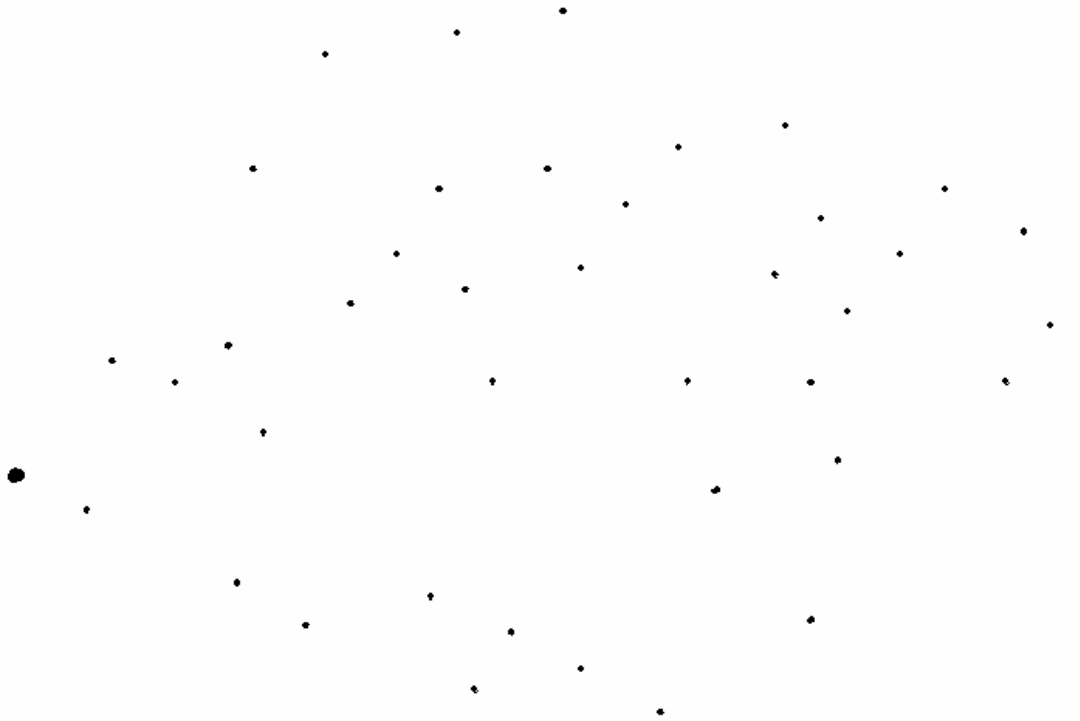
- An undirected graph $G=(V,E)$,
- Costs c_e for each edge in E ,
- Non-negative integral weights (or demands) d_v for each vertex in V ,
- A maximum capacity C ,
- A vertex designed as “the root”.

■ Find:

- A minimum cost spanning tree where the total weight of the vertices in each subtree hanging from the root does not exceeds C .

The Capacitated Minimum Spanning Tree Problem (CMST)

- Example: (unit demands, $C=10$)



The Capacitated Minimum Spanning Tree Problem (CMST)

- NP-hard problem. Instances with as few as 50 vertices can already be difficult to solve.
- Two cases:
 - Unit demand instances
 - Non-unit demand instances

Some Exact Algorithms

- Branch-and-cut over the directed formulation – Hall (1996)
- Lagrangean relaxation – Frangioni, Pretolani, Scutella (1999)
- Relax-and-cut – Bergson, Lucena, Maculan (2002)
- Branch-and-cut over hop-indexed formulations – Gouveia, Martins (1999, 2002, 2005)

A formulation for the CMST on a directed graph

- Replace each edge $(i,j) \in E$ by two arcs (i,j) and (j,i) , with cost $c_{ij} = c_{ji} = c_e$
- Let 0 be the root node
- Let $V_+ = \{1, \dots, n\}$
- Define binary variables
 - $X_a = 1$ if arc a belongs to an arborescence rooted at 0.

The directed formulation for the CMST

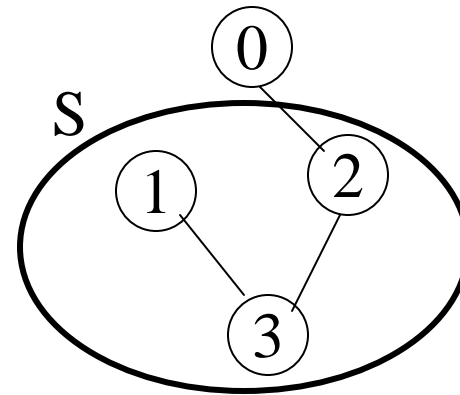
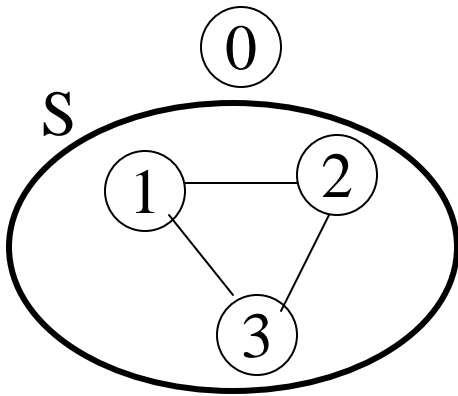
$$\begin{aligned} \min \quad & \sum_{a \in A} c_a x_a \\ \text{s.t.} \quad & \sum_{a \in d^-(i)} x_a = 1, \forall i \in V_+ \\ & \sum_{a \in d^-(S)} x_a \geq \left\lceil \frac{d(S)}{C} \right\rceil, \forall S \subseteq V_+ \\ & x_a \in \{0,1\}, \forall a \in A \end{aligned}$$

All vertices have exactly one incident arc

Capacity Cuts:
Eliminate cycles and enforce maximum tree weight

Violated Capacity Cuts

C=2, unit demands, root = 0



$$\left\lceil \frac{d(S)}{C} \right\rceil = 2 > \sum_{a \in d^-(S)} x_a = 1$$

The q-tree formulation

- Let T be the set of all arborescences with degree one at the root and not violating the capacity C .
- Define, for each subtree $t \in T$, the variable λ_t . The x variables can now be expressed as a combination of the λ 's.

A formulation with many variables

$$\begin{aligned} \min \quad & \sum_{a \in A} c_a x_a \\ \text{s.t.} \quad & \sum_{t \in T: a \in t} I_t - x_a = 0, \forall a \in A \\ & \sum_{a \in d^-(i)} x_a = 1, \forall i \in V_+ \\ & I_t \geq 0, \forall t \in T \\ & x_a \in \{0,1\}, \forall a \in A \end{aligned}$$

Defining q-trees

- The pricing on the previous formulation must find a minimum cost arborescence not violating the capacity C .
- Strongly NP-hard problem.
- Relax to allow cycles. The, so-called, “q-trees” can be found in pseudo-polynomial time ($O(n^2C^2)$) by dynamic programming.

Capacity Cuts

- As any other cut over the x variables, capacity cuts can also be introduced in the formulation.

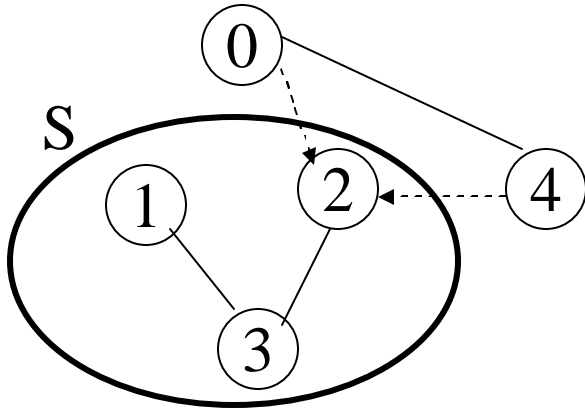
Root Cutset Cuts

- Let S be a subset of V_+ and define $k = \left\lceil \frac{d(S)}{C} \right\rceil$
- Let $A = \{i \in V \setminus S \mid \lceil (d(S) + d(i))/C \rceil = k\}$
- and $B = (V \setminus S) \setminus A$

$$\frac{k+1}{k} \sum_{a \in \{d^-(S) \cap d^+(A)\}} x_a + \sum_{a \in \{d^-(S) \cap d^+(B)\}} x_a \geq k+1, \forall S \subseteq V_+$$

Root Cutset cuts

C=3, unit demands, root=0



$$k = \left\lceil \frac{d(S)}{C} \right\rceil = 1$$

$$A = \{0\}, B = \{4\}$$

$$2 \sum_{a \in \{d^-(S) \cap d^+(A)\}} x_a + \sum_{a \in \{d^-(S) \cap d^+(B)\}} x_a = 1.5 < 2$$

VIOLATED

Other Cuts ?

- BCP over the q-tree formulation and using those two families of cuts (FPPU03).
 - Better bounds than previous approaches (specially for non-unitary demands). Many open instances could be solved.
 - But several other instances, with as few as 80 vertices and unitary demands, remained unsolved.

How to find new cuts to improve this BCP ?

Other Known Cuts

Other families of cuts (multistars) are known for the CMST (Araque, Hall, Magnanti, Gouveia).

They also may be interpreted as an attempt to capture the idea that “already wasted” arcs should count less for the capacity of a set S .

Other authors reported that those cuts have little impact on bounds.

The hop-indexed formulation

- The HIF is a fixed-charge flow formulation (Gouveia, Martins, 1999) for the unitary demand case. Binary variables x_{ij}^h indicate that arc (i,j) is h “hops” away from the root.
- Leads to large LPs, $O(m.C)$ variables and constraints.
- Combined with CCs and RCCs gives good bounds.

Cuts for the HIF

- Gouveia and Martins (2005) proposed a new family of cuts, a generalization of CCs taking advantage of the hop-indexed variables.
- Same idea: “already wasted” arcs count less for the capacity of a set S .
- Some improvements on the bounds.

Part II: Extended Capacity Cuts on the BCP

General Question

How to find new effective cuts on problems where BCP is being used ?

Some Observations on BCP

Suppose that the pricing subproblem is being solved by dynamic programming. We can introduce one variable to each transition between states.

1. Cuts over those variables do not change the pricing subproblem.
2. The size of the LPs that are actually solved does not change too.
3. The large number of new variables may allow complex cuts to be expressed in a simple way.

Some Observations on BCP

If you are already solving the pricing by dynamic programming, extending the formulation to include the transition variables is a “free lunch”.

Examples

- On CVRP we can have binary variables x_{ij}^d saying that a vehicle arrives in j from i with a remaining capacity of d units.
- Similar variables for the CMST.
- On VRPTW we can have variables indexed by both capacity and time.

The Base Equalities valid for capacity-indexed variables

- For each vertex i in V_+ :

$$\sum_{a \in d^-(i)} \sum_{d=1}^C d \cdot x_a^d - \sum_{a \in d^+(i)} \sum_{d=1}^{C-1} d \cdot x_a^d = d(i)$$

Summing over a set S in V_+ :

$$\sum_{a \in d^-(S)} \sum_{d=1}^C d \cdot x_a^d - \sum_{a \in d^+(S)} \sum_{d=1}^{C-1} d \cdot x_a^d = d(S)$$

- A combination of q-trees always satisfies the Base Equalities. However they are a very rich source of cuts.

Extended Capacity Cuts (ECCs)

- An ECC over a set S is any inequality valid for

$$\left\{ \begin{array}{l} \sum_{a \in \mathbf{d}^-(S)} \sum_{d=1}^C d \cdot x_a^d - \sum_{a \in \mathbf{d}^+(S)} \sum_{d=1}^{C-1} d \cdot x_a^d = d(S) \\ x \text{ binary} \end{array} \right\}$$

Capacity Cuts are ECCs

- Relax to greater than equal, divide by C:

$$\sum_{a \in d^-(S)} \sum_{d=1}^C (d/C) \cdot x_a^d - \sum_{a \in d^+(S)} \sum_{d=1}^{C-1} (d/C) \cdot x_a^d \geq d(S)/C$$

- Perform integral rounding

$$\sum_{a \in d^-(S)} \sum_{d=1}^C x_a^d \geq \lceil d(S)/C \rceil$$

Note that the original variables $x_a = \sum_{d=1}^C x_a^d$

Other known cuts are ECCs

- Root cutsets and multistars are dominated by ECCs obtained by integral rounding.
- The hop-indexed cuts by GM05 can be shown to be dominated by ECCs.

How to find new interesting cuts ?

Homogeneous ECCs

- Given a set S , define

$$y_d = \sum_{a \in d^-(S)} x_a^d \quad \forall d = 1 \dots C$$

$$z_d = \sum_{a \in d^+(S)} x_a^d \quad \forall d = 1 \dots C - 1$$

$$D = d(S)$$

- HECCs are cuts over the above y and z variables.

HECCs obtained by rounding

The integer rounding of

$$\sum_{d=1}^C d \cdot y_d - \sum_{d=1}^{C-1} d \cdot z_d \leq D$$

with multipliers $r = a/b$, a and b in the range $1 \dots C$, already gives a reasonable family of cuts.

Obtaining better HECCs

When C is small, one can compute the facets of

$$P(C, D) = \left\{ \begin{array}{l} \sum_{d=1}^C d \cdot y_d - \sum_{d=1}^{C-1} d \cdot z_d = D, \\ \sum_{d=1}^C y_d \leq |S|, \\ (y, z) \in Z_+^{2C-1} \end{array} \right\}$$

Facet HECCs

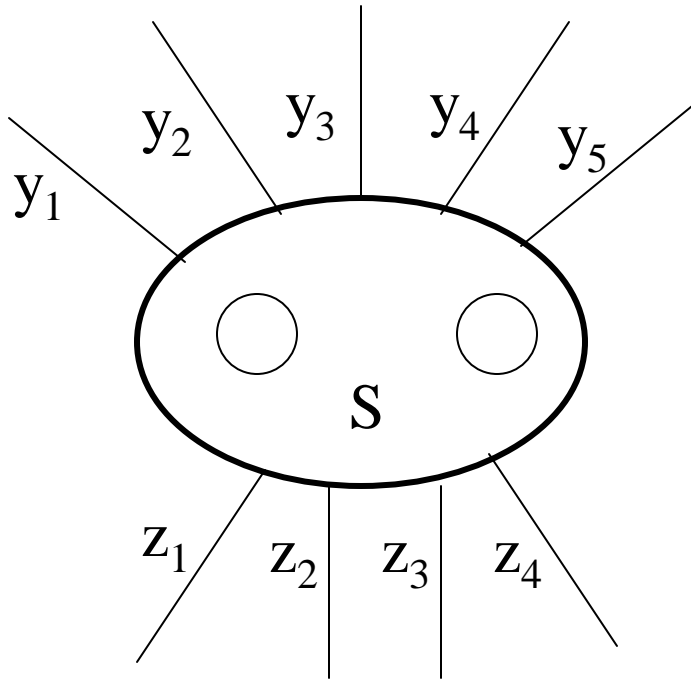
When C is small, polyhedra $P(C,D)$ can be described by few facets. The value of D (we tested up to 10) has little effect on the number of facets.

Examples:

- $P(5,D)$ s have 4 to 6 non-trivial facets.
- $P(10,D)$ s have about 300 non-trivial facets.

$$P(5,2) =$$

$$\left\{ \begin{array}{l} y_1 + 2y_2 + 3y_3 + 4y_4 + 5y_5 - z_1 - 2z_2 - 3z_3 - 4z_4 = 2, \\ y_1 + y_2 + y_3 + y_4 + y_5 \leq 2, \end{array} \right. \quad (y, z) \in \mathbb{Z}_+^9$$



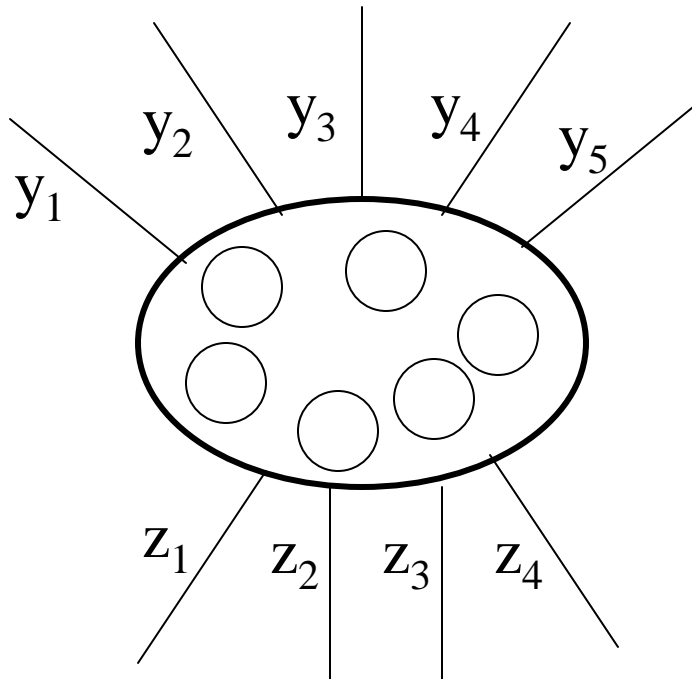
$$\mathbf{C=5, d(S)=2}$$

$$P(5,2) =$$

$$\left\{ \begin{array}{l} y_1 + 2y_2 + 3y_3 + 4y_4 + 5y_5 - z_1 - 2z_2 - 3z_3 - 4z_4 = 2, \\ y_1 + y_2 + y_3 + y_4 + y_5 \leq 2, \\ y_1 + 2y_2 + 2y_3 + 3y_4 + 4y_5 - z_2 - 2z_3 - 2z_4 \geq 2 \quad (r = 2/3), \\ y_1 + 2y_2 + 2y_3 + 3y_4 + 3y_5 - z_2 - z_3 - 2z_4 \geq 2 \quad (r = 3/5), \\ y_1 + 2y_2 + 2y_3 + 2y_4 + 3y_5 - z_3 - 2z_4 \geq 2, \\ y_1 + 2y_2 + 2y_3 + 2y_4 + 2y_5 - z_4 \geq 2, \\ y_1 + 2y_2 + 2y_3 + 3y_4 + 4y_5 - z_1 - 2z_2 - 2z_3 - 3z_4 \leq 2, \\ y_1 + 2y_2 + 2y_3 + 2y_4 + 3y_5 - z_1 - 2z_2 - 2z_3 - 2z_4 \leq 2, \\ (y, z) \geq 0 \end{array} \right.$$

$$P(5,6) =$$

$$\left\{ \begin{array}{l} y_1 + 2y_2 + 3y_3 + 4y_4 + 5y_5 - z_1 - 2z_2 - 3z_3 - 4z_4 = 6, \\ y_1 + y_2 + y_3 + y_4 + y_5 \leq 6, \end{array} \quad (y, z) \in \mathbb{Z}_+^9 \right\}$$



$$\mathbf{C=5, d(S)=6}$$

$$P(5,6)=$$

$$\left\{ \begin{array}{l} y_1 + 2y_2 + 3y_3 + 4y_4 + 5y_5 - z_1 - 2z_2 - 3z_3 - 4z_4 = 6, \\ y_1 + y_2 + y_3 + y_4 + y_5 \leq 6, \\ y_1 + y_2 + y_3 + y_4 + y_5 \geq 2 \quad (r = 1/5, \text{ a CC}), \\ y_1 + 2y_2 + 2y_3 + 2y_4 + 3y_5 - z_3 - 2z_4 \geq 4, \\ y_1 + 2y_2 + 2y_3 + 2y_4 + 3y_5 - z_2 - 2z_3 - 2z_4 \leq 4, \\ y_1 - y_5 + z_1 + z_2 + z_3 + 2z_4 \geq 0, \\ (y, z) \geq 0 \end{array} \right.$$

Conj: A CC only defines a facet of $P(C,D)$ when $D = kC + 1$

Obtaining HECCs

When C is large (say, $C > 100$) rounded HECCs became weaker. We believe that better cuts can be obtained by applying well-chosen sub-additive functions over the base equalities

$$\sum_{d=1}^C d \cdot y_d - \sum_{d=1}^{C-1} d \cdot z_d = D$$

Not tested yet.

Computational Results

A BCP including HECCs was tested over the OR-Lib unitary instances. They have 40, 80, 120 and 160 vertices and $C=5$, 10 and 20.

- Almost all instances with 40 vertices are now solved in the root node.
- For larger instances the duality gap is typically halved.

Other ECCs being investigated

- Semi-Homogeneous ECCs.
- Local ECCs.

“Second Generation” RBCPs ?

RBCP also including cuts over the dynamic programming transition variables seems to be a promising approach.

Part III:

Extended Capacity Cuts without column generation

The Capacity-Indexed Formulation (CIF) for the CMST

$$\begin{aligned}
 \min \quad & \sum_{a \in A} c_a x_a \\
 \text{s.t.} \quad & \sum_{a \in \mathbf{d}^-(i)} x_a = 1, \quad \forall i \in V_+ \\
 & x_a = \sum_{d=1}^C x_a^d, \quad \forall a \subseteq A \\
 & \sum_{a \in \mathbf{d}^-(i)} \sum_{d=1}^C d \cdot x_a^d - \sum_{a \in \mathbf{d}^+(i)} \sum_{d=1}^{C-1} d \cdot x_a^d = d(i), \quad \forall i \in V_+ \\
 & x_a^d \in \{0,1\}, \quad \forall a \in A, d = 1 \dots C
 \end{aligned}$$

The CIF for the CMST

- Reasonably compact for small values of C (say, up to 20); mC variables, but only $2n$ constraints.
- Although its linear relaxation is weak, the addition of HECCs make the bounds close to those obtained by the BCP over the q -trees.
- The small difference is usually compensated by the faster solution of a node.
- Several open instances could be solved in this way.

Results over OR-LIB instances with $C=5$

- Separating HECC facets.
- All instances solved, seven of them were formerly open. In five cases the best heuristic solution was not optimal.

Results over OR-LIB instances with C=5

Size	Avg. Gap (%)	Avg. Root Time (s)	Avg. Total Time (s)	# Instances / # F. Open
40	0.13	2.27	2.32	20/0
80	0.13	28	90	15/0
120	0.16	141	569	10/5
160	0.13	294	754	2/2

Pentium 2.4GHz

Results over OR-LIB instances with $C=10$

- Separating rounded HECCs.
- Three formerly open instances solved.
Most instances with 120 and 160
vertices still have to be run.

Results over OR-LIB instances with C=10

Size	Avg. Gap (%)	Avg. Root Time (s)	Avg. Total Time (s)	# Instances / # F. Open
40	0.19	39	60	20/0
80	0.36	317	4435	10/2
120	-	-	-	10/9
160	-	-	-	2/2

Bounds and times on formerly open instance te80-2 C=10

GM (2005) : 1609.95, 4039s (AMD 1GHz)
q-trees+CCs+RCCs: 1613.18, 435s
q-trees+ HECCs : 1623.14, 1442s
CIF + HECCs : 1620.98, 537s

Opt: **1639**, 23309s = 6.5h (11995 nodes)

Rounding HECCs with D up to 10.

Bounds and times on formerly open instance te80-3 C=10

GM(2005)	:	1666.58,	6563 s (AMD 1GHz).
q-trees+CCs+RCCs	:	1669.51,	215 s
q-trees+ HECCs	:	1675.20,	846 s
CIF+HECCs	:	1673.28,	286 s

Opt: **1687**, 14880s = 4.1h (6332 nodes)

Rounding HECCs with D up to 10.

Bounds and times on formerly open instance te120-1 C=10

GM (2005)	: 891.23, 6027 s (AMD 1GHz).
q-trees+CCs+RCCs	: 892.88, 415 s
q-trees+HECCs	: 898.01, 2142 s
CIF+HECCs	: 896.71, 662 s

Opt: **904**, 44372s = 12.3h (11584 nodes)

Rounding HECCs with D up to 10.

Capacity-indexed formulations for other problems

- VRPs, Steiner, facility location, network design, etc.
- Many known cuts on the usual variables can be shown to be equivalent or dominated by rounded ECCs.

The Fixed Charge Network Flow Problem

$$\begin{aligned} \min \quad & \sum_{a \in A} c_a x_a + \sum_{a \in A} f_a w_a \\ \text{s.t.} \quad & \sum_{a \in d^-(i)} x_a - \sum_{a \in d^+(i)} x_a = d(i) \quad , \forall i \in V \\ & 0 \leq x_a \leq u_a w_a \quad , \forall a \in A \\ & w \text{ binary} \end{aligned}$$

Suppose d and u integral

FCNF Capacity-Indexed Reformulation

$$\begin{aligned} \min \quad & \sum_{a \in A} \sum_{d=1}^{u_a} (c_a d + f_a) x_a^d \\ \text{s.t.} \quad & \sum_{a \in \mathbf{d}^-(i)} \sum_{d=1}^{u_a} d \cdot x_a^d - \sum_{a \in \mathbf{d}^+(i)} \sum_{d=1}^{u_a} d \cdot x_a^d = d(i) \quad , \forall i \in V \\ & \sum_{d=1}^{u_a} x_a^d \leq 1 \quad , \forall a \in A \\ & x \text{ binary} \end{aligned}$$

FCNF Capacity-Indexed Reformulation

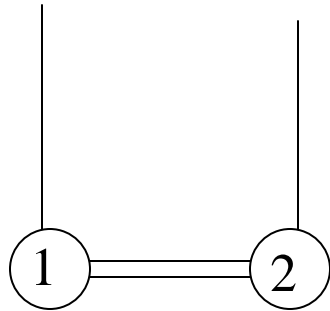
- Same value of LP relaxation, but now ECCs can be used. The rhs D on base equalities can also be zero or negative.
- The CIFs for CMST, VRP, ... are particular cases.
- Results shown with the CIF over the CMST only uses general FCNF HECCs. The separation procedures know nothing about CMST !

Some interesting questions

- When is not possible to compute ECC facets, how we can obtain cuts better than those by rounding ? Can we explore the very special structure of the base equality ?
- How ECCs compare with the known cuts for the FCNF (flow covers, ...) over the usual variables ?

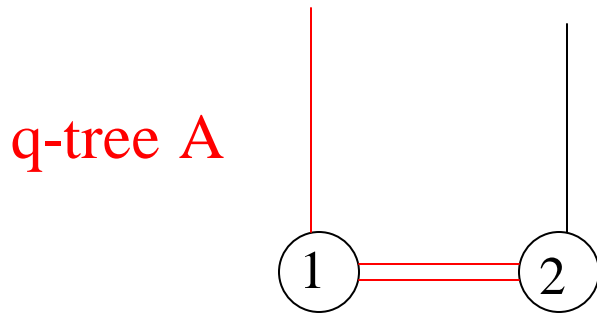
Extra Slides

Part of typical q-tree fractional solution, $C \geq 3$



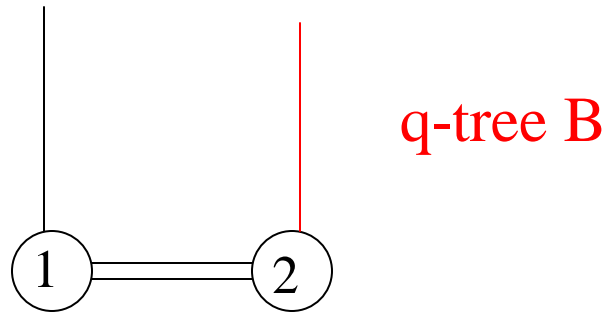
$$0.5 \times (\dots -1 -2 -1) + 0.5 \times (\dots -2)$$

Part of typical q-tree fractional solution, $C \geq 3$



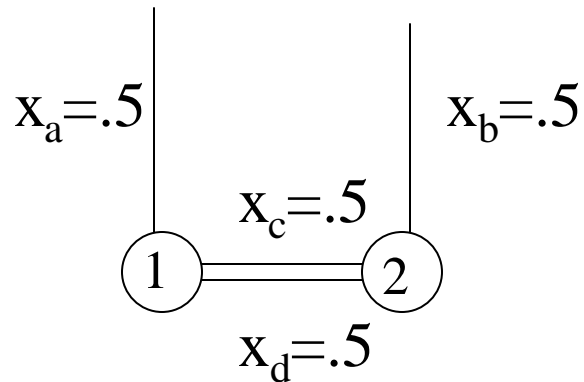
$$0.5 \times (\dots -1 -2 -1) + 0.5 \times (\dots -2)$$

Part of typical q-tree fractional solution, $C \geq 3$



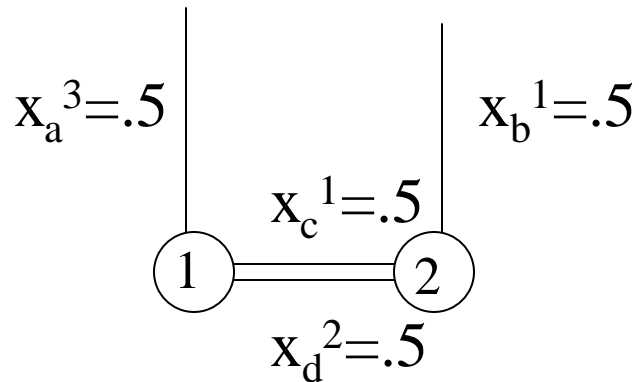
$$0.5 \times (\dots -1 -2 -1) + 0.5 \times (\dots -2)$$

Viewing this fractional solution over x variables



Looking only at those x variables, no cut is possible:
this solution is a convex combination of valid trees
(...,1-2) and (...,2-1).

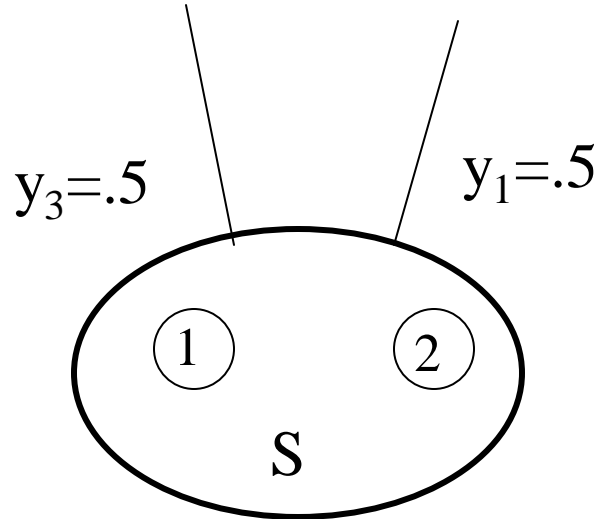
Viewing this fractional solution over capacity-indexed variables



An arc a with capacity 3 is entering a set $S = \{1, 2\}$ with demand 2. Some arc with capacity 1 or 2 must leave S !

Ad hoc cut: $x_a^3 \leq Sx_{(1,j)}^2 + Sx_{(1,j)}^2 + Sx_{(2,k)}^1$

Viewing this fractional solution over y and z variables



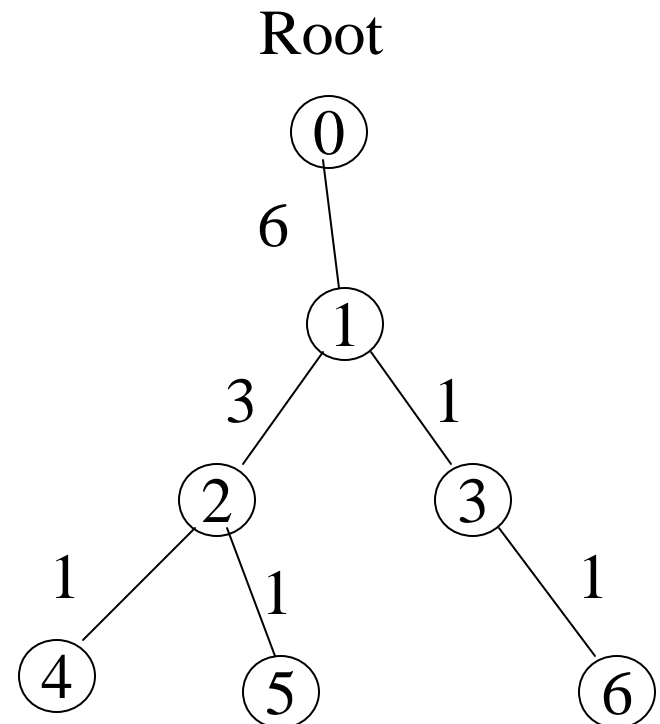
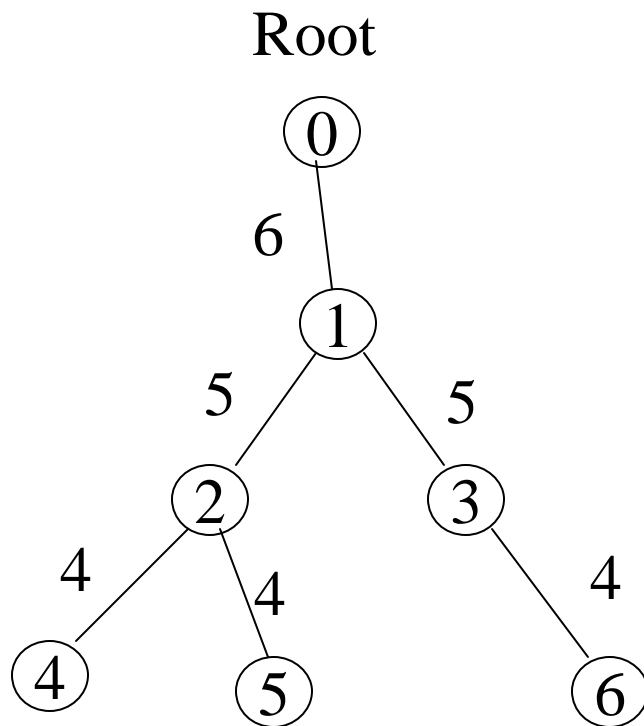
Rounded HECC: $y_1 + 2 y_2 + 2 y_3 + \dots - \dots \geq 2$

CMST : Why ECCs are stronger than cuts over hop-indexed vars

Bounds over hops

C=6, Unit demands

Actual Capacities



Acknowledgements

- The first author thanks Gleb Belov, Adam Letchford and Luis Gouveia for valuable discussions.