A Comparative Study of Linear and Semidefinite Branch-and-Cut Methods for Solving the Minimum Graph Bisection Problem

Christoph Helmberg, TU Chemnitz

joint work with

Michael Armbruster, TU Chemnitz

Marzena Fügenschuh, TU Darmstadt

Alexander Martin, TU Darmstadt

- Problem and Polyhedral Results
- Sparse SDP Branch-and-Cut Techniques
- Numerical Results

(supported by the DFG)

A long term goal . . .

Solve large scale, structured quadratic 0-1 programming problems

$$\begin{array}{ll} \text{min} & x^T C x \\ \text{s.t.} & x^T A_i x = b_i, \quad i = 1, \ldots, m, \\ & x \in \{0,1\}^n \end{array}$$

by semidefinite programming relaxations.
[Shor87,Lov'aszSchrijver91,Lasserre200*,. . .]

A long term goal . . .

Solve large scale, structured quadratic 0-1 programming problems

$$\begin{array}{ll} \text{min} & x^T C x \\ \text{s.t.} & x^T A_i x = b_i, \quad i = 1, \ldots, m, \\ & x \in \{0,1\}^n \end{array}$$

by semidefinite programming relaxations.
[Shor87,Lov'aszSchrijver91,Lasserre200*,. . .]

More reasonable intermediate goals . . .

- $m = 0 \rightarrow$ max-cut (large scale: Ising models on grid graphs)

A long term goal . . .

Solve large scale, structured quadratic 0-1 programming problems

$$
\begin{array}{ll}
\text{min} & x^T C x \\
\text{s.t.} & x^T A_i x = b_i, \quad i = 1, \ldots, m, \\
 & x \in \{0,1\}^n
\end{array}
$$

by semidefinite programming relaxations.
[Shor87,Lov'aszSchrijver91,Lasserre200*,. . .]

More reasonable intermediate goals . . .

- $m = 0 \rightarrow$ max-cut (large scale: Ising models on grid graphs)
- $m = 1 \rightarrow$ graph bisection

“SDP is good in theory and on small dense instances,
but in sparse large scale instances LP is better.’

“SDP is good in theory and on small dense instances, but in sparse large scale instances LP is better.’

This is no longer true in this generality!

"SDP is good in theory and on small dense instances,
but in sparse large scale instances LP is better.'

This is no longer true in this generality!

For large scale sparse graph bisection problems
we compare LP and SDP in the same branch and cut-framework,
$\rightarrow$ SDP outperforms LP in many if not most instances.

Like "Simplex versus LP:"
- There will be an abundance of problems that are better solved by pure LP
- Computational SDP in large scale branch-and-cut is still in its infancy, we expect significant progress in the years to come

Emphasis of this work:
- develop sparse SDP branch-and-cut techniques in general
- advance and compare LP branch-and-cut for graph bisection
- NOT: develop efficient partition algorithms

Problem and Polyhedral Results

The Node Weighted Bisection Problem

- simple undirected Graph $G = (V, E)$,

  $V = \{1, \dots, n\}$, $E \subseteq \{ij : i, j \in V, i \neq j\}$

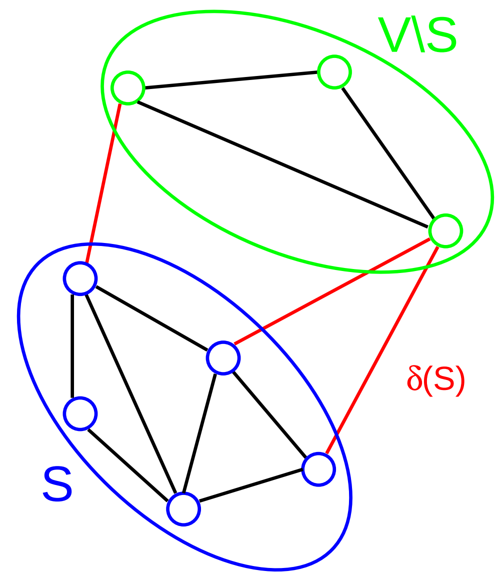
  node weights $\varphi_i \in \mathbb{N}_0$ for $i \in V$,

  capacity $F \in \mathbb{N}_0$,

- Find a bisection $(S, V \setminus S)$ with

  $\varphi(S) := \sum_{i \in S} \varphi_i \leq F$ and $\varphi(V \setminus S) \leq F$

  and $\delta(S)$ minimal (weights w_{ij})



$$P_B = \text{conv}\{\, y = \chi(\delta(S)) : S \subseteq V,\ \varphi(S) \leq F,\ \varphi(V \setminus S) \leq F\} \qquad \subseteq P_{\text{CU}}$$

IP-formulation:

suppose G contains a spanning star rooted at s

$$y_{ij} = \begin{cases} 1 & \text{if } ij \text{ is in the cut} \\ 0 & \text{otherwise} \end{cases}$$

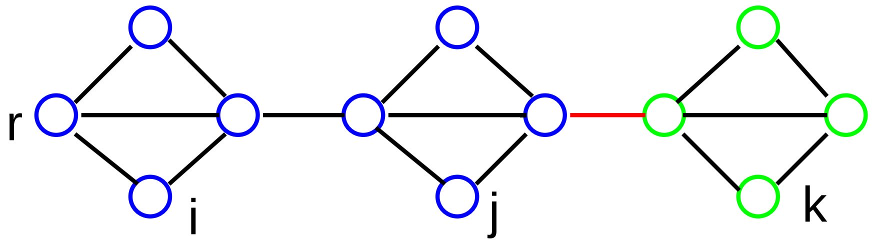
$$\begin{aligned}
\text{min} \quad & \sum_{ij} w_{ij} y_{ij} \\
\text{s.t.} \quad & \varphi_s + \sum_{v \neq s} \varphi_v (1 - y_{sv}) \leq F \\
& \sum_{v \neq s} \varphi_v y_{sv} \leq F \\
& \sum_{ij \in C \setminus U} y_{ij} + \sum_{ij \in U} (1 - y_{ij}) \geq 1 \qquad \text{cycle } C \subseteq E, \text{odd } U \subseteq C \\
& y \in \{0, 1\}^E
\end{aligned}$$

Related Polyhedral Investigations/Surveys in the Literature

- Cut Polytope: [DezaLaurent97]
- Node Capacitated Graph Partitioning: [FMdSWW96]
- Equipartition: [ConfortiRaoSassano90], [deSouza93]
- Knapsack: [Weismantel97]

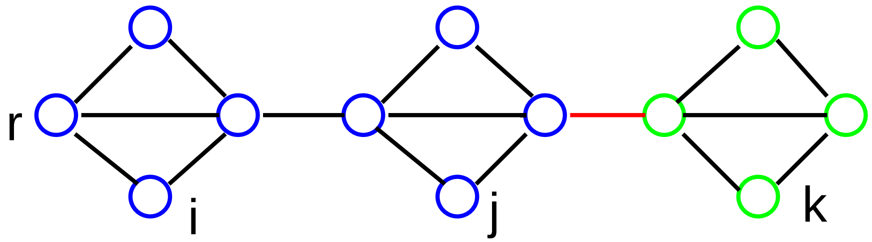
cycle inequalities of the cut polytope [BarahonaMahjoub86], tree inequalities [ConfortiRaoSassano90], star inequalities [ConfortiRaoSassano90], cycle inequalities of capacitated graph partitioning [ConfortiRaoSassano90], cycle with tails inequalities [FerreiraMartindeSouzaWeismantelWolsey96], suspended tree inequalities [LaurentdeSouza95], path block cycle inequalities [LaurentdeSouza95], cycle with ear inequalities [FerreiraMartindeSouzaWeismantelWolsey96], strengthened cycle with ear inequalities [FerreiraMartindeSouzaWeismantelWolsey96], knapsack tree inequalities [FerreiraMartindeSouzaWeismantelWolsey96] and strengthened knapsack tree inequalities [FerreiraMartindeSouzaWeismantelWolsey96]

Bounding the size of the side belonging to some root $r \in V$



Given $y = \delta(S)$, does i belong to the same side as r?

Bounding the size of the side belonging to some root $r \in V$



Given $y = \delta(S)$, does i belong to the same side as r?

Knapsack Tree Inequalities: [FMdSWW96]

Yes, if the shortest path P_{ri} from r to i consists of edges $y_e = 0$:

$$1 - \sum_{e \in P_{ri}} y_e \text{ positive } \Rightarrow \; i \text{ belongs to the same side as } r$$

$$\longrightarrow \qquad \varphi_r + \sum_{i \in V_T \setminus \{r\}} \varphi_i \left[1 - \sum_{e \in P_{ri}} y_e \right] \quad \leq \quad F$$

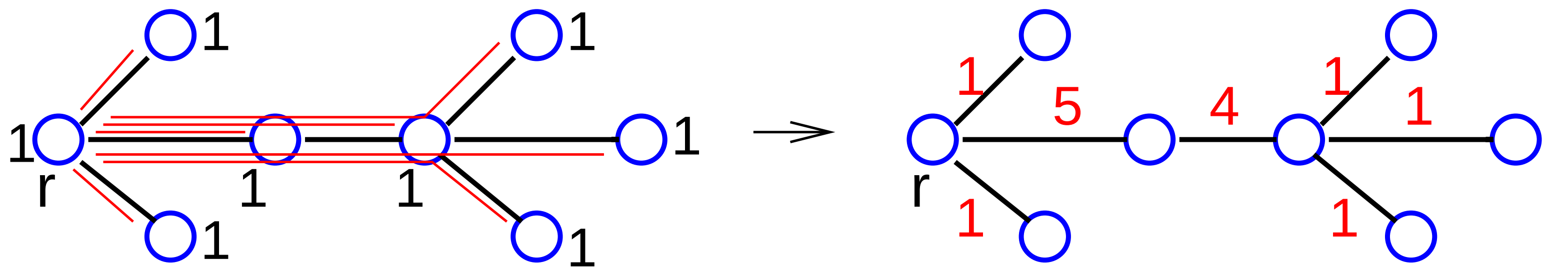
for some tree $T = (V_T, E_T)$ and its paths P_{ri} from r to i.

Collecting the weights of the edges along the paths from root r for each edge and applying "trivial" strengthening yields a *truncated knapsack tree ineq.*

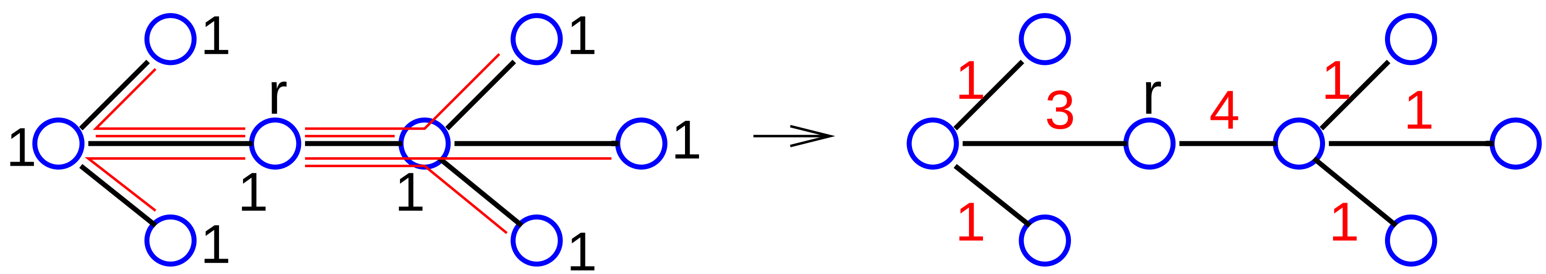
If $\sum a_v z_v \leq a_0$ is valid for $P_K := \text{conv}\{z \in \{0,1\}^V : \sum_{v \in V} \varphi_v z_v \leq F\}$ then

$$\sum_{e \in E_T} \alpha_e^r y_e \geq \alpha_0 \quad \text{with} \quad \alpha_0 := \sum_{v \in V_T} a_v - a_0 \quad \text{and} \quad \alpha_e^r := \min\{\sum_{v : e \in P_{rv}} a_v, \alpha_0\}$$

is valid for P_B. [FMdSWW96]



The Choice of the Root in Knapsack Tree Inequalities [AFHM07]



The Choice of the Root in Knapsack Tree Inequalities [AFHM07]

Best root in T is a *minimal root* $r \in \mathcal{R} = \operatorname*{Argmin}_{s \in V_T} \sum_{e \in E_T} \alpha_e^s$ (easy to find)

Theorem 1 *If $r \in \mathcal{R}$ then for all $s \in V_T$, $y \geq 0$,* $\sum_{e \in E_T} \alpha_e^s y_e \geq \sum_{e \in E_T} \alpha_e^r y_e$ $(\geq \alpha_0)$

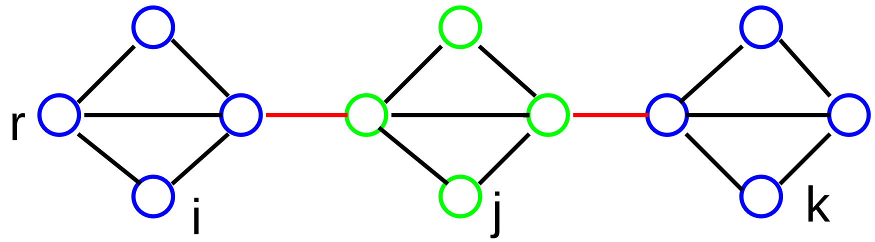
For $\varphi_v = 1$ $(v \in V)$ and root r each unreduced edge has weight $|V_e^r| = |\{v : e \in P_{rv}\}|$ (nodes "below" e) $\rightarrow$ minimal roots are "centered"

Theorem 2 *Assume $G = (V, E)$ is a tree, $\varphi_v = 1$ $(v \in V)$, $\frac{|V|}{2} + 1 \leq F < |V|$.*

$$\sum_{e \in E} \min\{|V_e^r|, |V| - F\} y_e \geq |V| - F$$

is facet-defining for P_B if and only if one of the following holds:
(a) r is a minimal root and each branchless path of length F contains a leaf
(b) $F = |V| - 1$.

Bounding the size of the side belonging to some root $r \in V$



Given $y = \delta(S)$, does i belong to the same side as r?

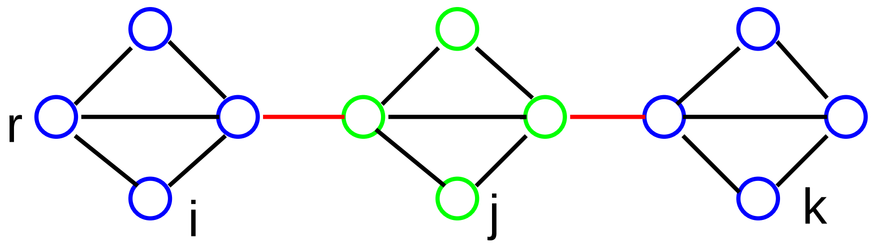
Knapsack Tree Inequalities:

Yes, if the shortest path P_{ri} from r to i consists of edges $y_e = 0$:

$$1 - \sum_{e \in P_{ri}} y_e \ \text{ positive } \Rightarrow \ i \text{ belongs to the same side as } r$$

$$\longrightarrow \qquad \varphi_r + \sum_{i \in V_T \setminus \{r\}} \varphi_i \left[1 - \sum_{e \in P_{ri}} y_e \right] \quad \leq \quad F$$

Bounding the size of the side belonging to some root $r \in V$



Given $y = \delta(S)$, does i belong to the same side as r?

Knapsack Tree Inequalities:
Yes, if the shortest path P_{ri} from r to i consists of edges $y_e = 0$:

$$1 - \sum_{e \in P_{ri}} y_e \text{ positive } \Rightarrow \; i \text{ belongs to the same side as } r$$

$$\longrightarrow \qquad \varphi_r + \sum_{i \in V_T \setminus \{r\}} \varphi_i \left[1 - \sum_{e \in P_{ri}} y_e \right] \quad \leq \quad F$$

Bisection Knapsack Walk Inequalities: (exploit bipartition) [AFHM07]
Yes, if there is a path P_{ri} with an <u>even</u> set $H_{ri} \subseteq P_{ri}$ of cut edges:

$$1 - \sum_{e \in P_{ri} \setminus H_{ri}} y_e - \sum_{e \in H_{ri}} (1 - y_e) \text{ positive } \Rightarrow \; i \text{ belongs to the same side as } r$$

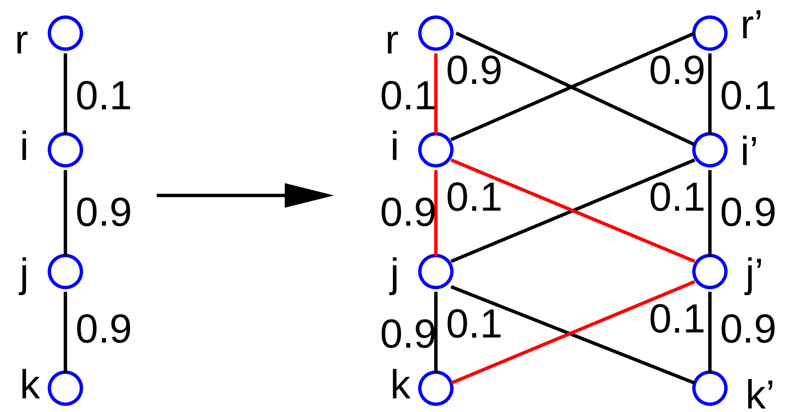
$$\varphi_r + \sum_{i \in V \setminus \{r\}} \varphi_i \left[1 - \sum_{e \in P_{rv} \setminus H_{ri}} y_e - \sum_{e \in H_{ri}} (1 - y_e) \right] \quad \leq \quad F$$

Finding the best path P_{ri} with even set $H_{ri} \subseteq P_{ri}$ [AFHM07]

For $y \in [0,1]^E$, root r and node i the goal is to maximize

$$1 - \sum_{e \in P_{ri} \setminus H_{ri}} y_e - \sum_{e \in H_{ri}} (1 - y_e)$$

simultaneously for all i by a shortest path tree in an auxiliary graph:



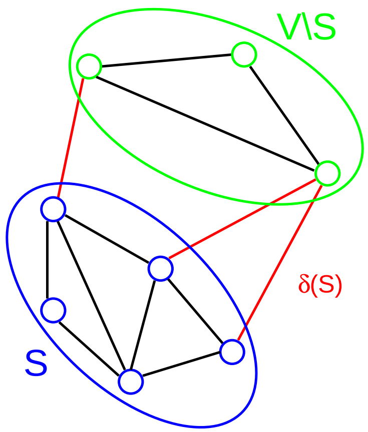
Note: • best *walk* P_{ri} and H_{ri} can be found in polynomial time
• they do not depend on the knapsack inequality $\varphi(S) \leq F$
$\rightarrow$ find paths first, then use knapsack separator $\rightarrow a(S) \leq a_0$

Alg. almost identical to cycle ineq. separation for P_{CUT} of [BM86]
- In the presence of cycle ineqs. a direct edge gives the best bound
- There are various (involved) strengthening techniques

Sparse
SDP Branch-and-Cut Techniques

SDP-Relaxation for Node Weighted Bisection

Represent partition by $x \in \{-1,1\}^n$ with $x_i = \begin{cases} 1 & i \in S \\ -1 & i \in V \setminus S \end{cases}$



$$|\varphi^T x| \le 2F - \varphi(V)$$

$$x_i x_j = \begin{cases} -1 & ij \in \delta(S) \\ 1 & \text{otherwise} \end{cases}$$

edge weights w into appropriate C,
$x^T C x = \langle Cx, x\rangle = \langle C, xx^T\rangle$

Relax $xx^T = [x_i x_j]$ to $X \succeq 0$

$$\min_{\substack{x \in \{-1,1\}^n \\ (\varphi^T x)^2 \le (2F - \varphi(V))^2}} x^T C x \quad \ge \quad \begin{array}{ll} \min & \langle C, X\rangle \\ \text{s.t.} & \text{diag}(X) = e \\ & \langle \varphi\varphi^T, X\rangle \le (2F - \varphi(V))^2 \\ & X \succeq 0 \\ & [\text{rank}(X) = 1] \end{array}$$

Shor87; Schrijver; DelormePoljak93
GoemansWilliamson95 (0.878 approx. for max-cut with $w \ge 0$)
FriezeJerrum97

For Branch-and-Cut we need

- a solver that
  - ◦ exploits structure (sparsity, low rank, . . .)
  - ◦ gives reliable bounds
  - ◦ produces primal approximate solutions (separation, heuristics)
  - ◦ allows warm starts for cutting planes
  - ◦ allows warm starts with branching decisions
- efficient separation routines (identical for LP and SDP)
- good primal heuristics (for SDP: GW-style with local improvement)
- good branching decisions

For SDP we use the spectral bundle method

([RendlRinaldiWiegele07] use bundle with elliptope oracle: not sparse)

Spectral Bundle Method

[H.Rendl00,H.Kiwiel02]

One can show

$$\begin{array}{rl} \max & \langle C, X\rangle \\ \text{s.t.} & \langle I, X\rangle = a \\ & \mathcal{A}X \le b \\ & X \succeq 0, \end{array} \quad = \quad \min_{y \in \mathbb{R}^m_+} \; a\lambda_{\max}(C - \mathcal{A}^T y) + \langle b, y\rangle$$

- The relaxation is of this form with $a = n$
- Any feasible y yields a valid dual bound
- The optimal y yields the optimal value of the semidefinite relaxation
- The matrix $C - \sum_i A_i y_i$ inherits structure of cost matrix and constraints [→ $\lambda_{\max}$ by iterative methods like Lanczos]

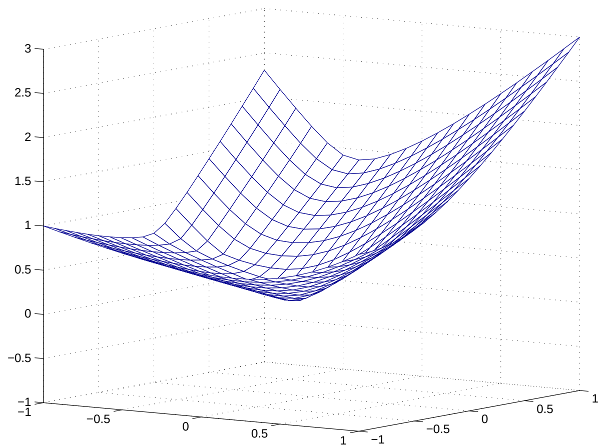
Solve the eigenvalue problem

→ minimize a nonsmooth convex function

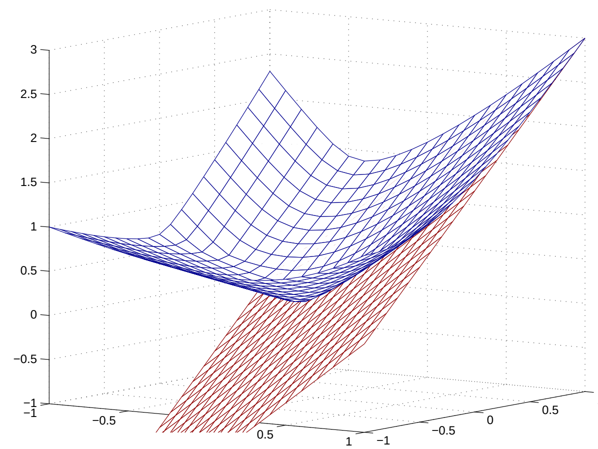
→ subgradient and bundle methods Hiriart-Urruty and Lemaréchal 1993

Proximal Bundle Method [Lemaréchal78,Kiwiel90]

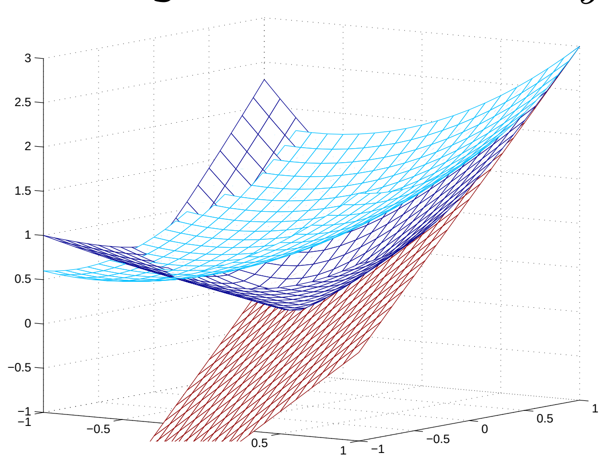
convex function



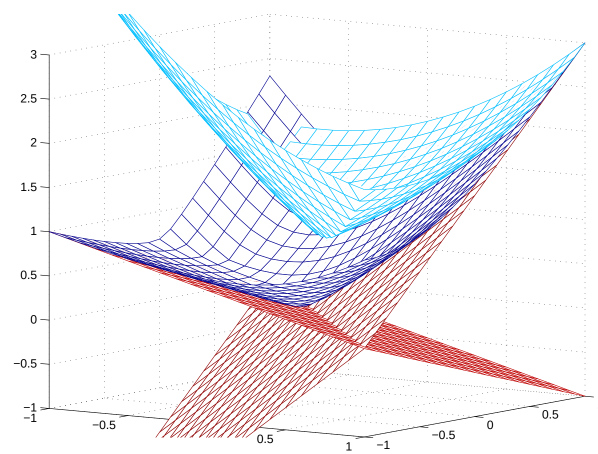
cutting plane model with $g \in \partial\varphi(\widehat{y})$



solve augmented model $\rightarrow y^+$



improve cutting plane model in y^+



Main steps

1. Given $\hat{y}$, find candidate by solving quadratic model

2. Evaluate function, determine subgradient

3. Decide on
   - null step
   - descent step

4. Update model and iterate

$\rightarrow$ Restarting no problem, any feasible y works . . .

Where is the primal? What about separation?

The Semidefinite Quadratic Model

For fixed slack variable η and center $\hat{y}$ solve

$$\text{(QSP)} \qquad \begin{array}{ll} \max & \langle C, X\rangle + \langle b - \eta - \mathcal{A}X, \hat{y}\rangle - \frac{1}{2u}\|b - \eta - \mathcal{A}X\|^2 \\ \text{s.t.} & X = PUP^T + \alpha\overline{W} \\ & \operatorname{tr} U + \alpha = a \\ & U \succeq 0, \alpha \geq 0. \end{array}$$

- P is an orthonormal matrix; minimal choice $P = v$ [v EV to $\lambda_{\max}$]
- $\overline{W}$ is a positive semidefinite matrix of trace 1
  e.g. last optimal solution of QSP, $\overline{W} = \overline{X}/n$ [need only $\mathcal{A}\overline{W}$, $\langle C, \overline{W}\rangle$]
- X satisfies $X \succeq 0$ and $\langle I, X\rangle = n$
- The new optimal $\overline{X}^+$ of (QSP) determines the next candidate y^+

Theorem 3 *[H. 2001]*
If the eigenvalue problem has an optimal solution then the algorithm generates a subsequence $K \subseteq \mathbb{N}$ so that all cluster points of $\overline{X}^k$, $k \in K$, are primal optimal solutions.

similar to Feltenmark and Kiwiel 2000

Spectral bundle and cutting planes [H.2004]

Main idea: separate with respect to $\overline{X} = PUP^T + \alpha\overline{W}$

Difficulty: $\overline{X}$ is 'never' feasible for all given constraints

$\longrightarrow$ the same inequalities may be separated again and again

$\longrightarrow$ separation routines can 'conceal' certain violated inequalities

One can show: no problem, if a maximum violation oracle is used

---

- Separating after almost every descent step performs great [[RenRinWie07] do it also after null steps]
- For SDP, separation outside the support of the graph may be very helpful $\rightarrow$ "support extension"
- Can be done efficiently even if $\overline{W}$ is kept sparse [use PUP^T as indicator]

Warm start after branching for spectral bundle

- Storing X or even P in every node of the BC-tree needs too much memory $\rightarrow$ keep only $y \in \mathbb{R}^m$
- Sometimes serious scaling problems occured if the branching constraint forces some y coordinates to move a lot ($\rightarrow$ slow progress!!!)
  [quadratic model term $\|\Delta y\|^2$]

Situation improved significantly with a scaling heuristic: [$\langle \Delta y, S^{-1} \Delta y \rangle$]

E.g., enforce $x_i = x_j$ by adding

$$\langle B_{ij}, X \rangle = 1 \quad \text{with} \quad B_{ij} = \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix} \begin{matrix} i \\ j \end{matrix} \quad = \mu_1 v_1 v_1^T + \mu_2 v_2 v_2^T$$

Consider $\lambda_{\max}(C - \mathcal{A}^T y - B_{ij} y_{m+1})$ [$\mathcal{A}^T y = \sum A_i y_i$]

Suppose μ_1 increases $\lambda_{\max}$, μ_2 decreases $\lambda_{\max}$

Coupled with those y_i of $\sum A_i y_i$ with $|v_j^T A_i v_j|$ big (normalized A_i):

$$s_{ii} = 1 + \varphi \begin{cases} \max\{(v_j^T A_i v_j)^2 : j = 1, 2\} & \text{if } v_1^T (C - \mathcal{A}^T y) v_1 \geq \lambda_{\max} - \delta \\ (v_2^T A_i v_2)^2 & \text{if } v_1^T (C - \mathcal{A}^T y) v_1 < \lambda_{\max} - \delta \end{cases}$$

Use this with $\varphi = 20$ for one descent step

Branching decision (general SDP)

- Strong branching too expensive
- Too few nodes for pseudo cost/reliability branching (see [AchterbergKochMartin05])

---

rule is based on PUP^T of $X = PUP^T + \overline{W}$

Idea: Find clusters in the vector labeling,
branch on those influencing the bound the most

$\bar{P} = PU^{\frac{1}{2}} \in \mathbb{R}^{n \times k}$ rows of $\bar{P} \rightarrow$ vector labelings

dominating direction: $v_{\max}$ to $\lambda_{\max}(\bar{P}^T \bar{P})$

Find closest row vector v_0: $\max\{|\bar{P}_{i,\bullet}^T v_{\max}|/\|\bar{P}_{i,\bullet}\| : i = 1, \ldots, n\}$

For $k = 1, 2, \ldots$ do:

- remove rows close to v_{k-1} with $|\bar{P}_{i,\bullet}^T v_{k-1}|/\|\bar{P}_{i,\bullet}\| \geq 0.2 \quad \rightarrow \quad I_{k-1}$
- find most orthogonal row v_k: $\min\{|\bar{P}_{i,\bullet}^T v_{k-1}|/\|\bar{P}_{i,\bullet}\|$: remaining rows $i\}$

Try each $\bar{X}_{k+} = \bar{P}_{k+}\bar{P}_{k+}^T + \overline{W}$ and $\bar{X}_{k-}$ with $\bar{P}_{k\pm} = \begin{cases} \pm v_1 & i \in I_k \\ \bar{P}_{i,\bullet} & i \notin I_k \end{cases}$

Branch on 1 and $\bar{k}$ with $\bar{k} \in \text{Argmax}(\langle C, X_{k+}\rangle + \langle C, X_{k-}\rangle)$

Numerical Results

The Setting of the Numerical Experiments

Used LP and SDP-relaxation in the same Branch&Cut-framework SCIP [thanks to Tobias Achterberg]

LP-relaxation

- basic relaxation: add a star into G if necessary and separate cycle ineqs.
- solve LPs using CPLEX

SDP-relaxation

- use same graph as LP
- canonical max-cut relaxation in $\{-1, 1\}$-variables ($\text{diag}(X) = e, X \succeq 0$)
- capacity constraint by $\langle \varphi\varphi^T, X \rangle \leq (2F - \varphi(V))^2$,
- solve dual by Spectral Bundle Method with primal aggregation [ConicBundle callable library]
- separate on primal aggregate w.r.t. the support, possibly enlarge the support

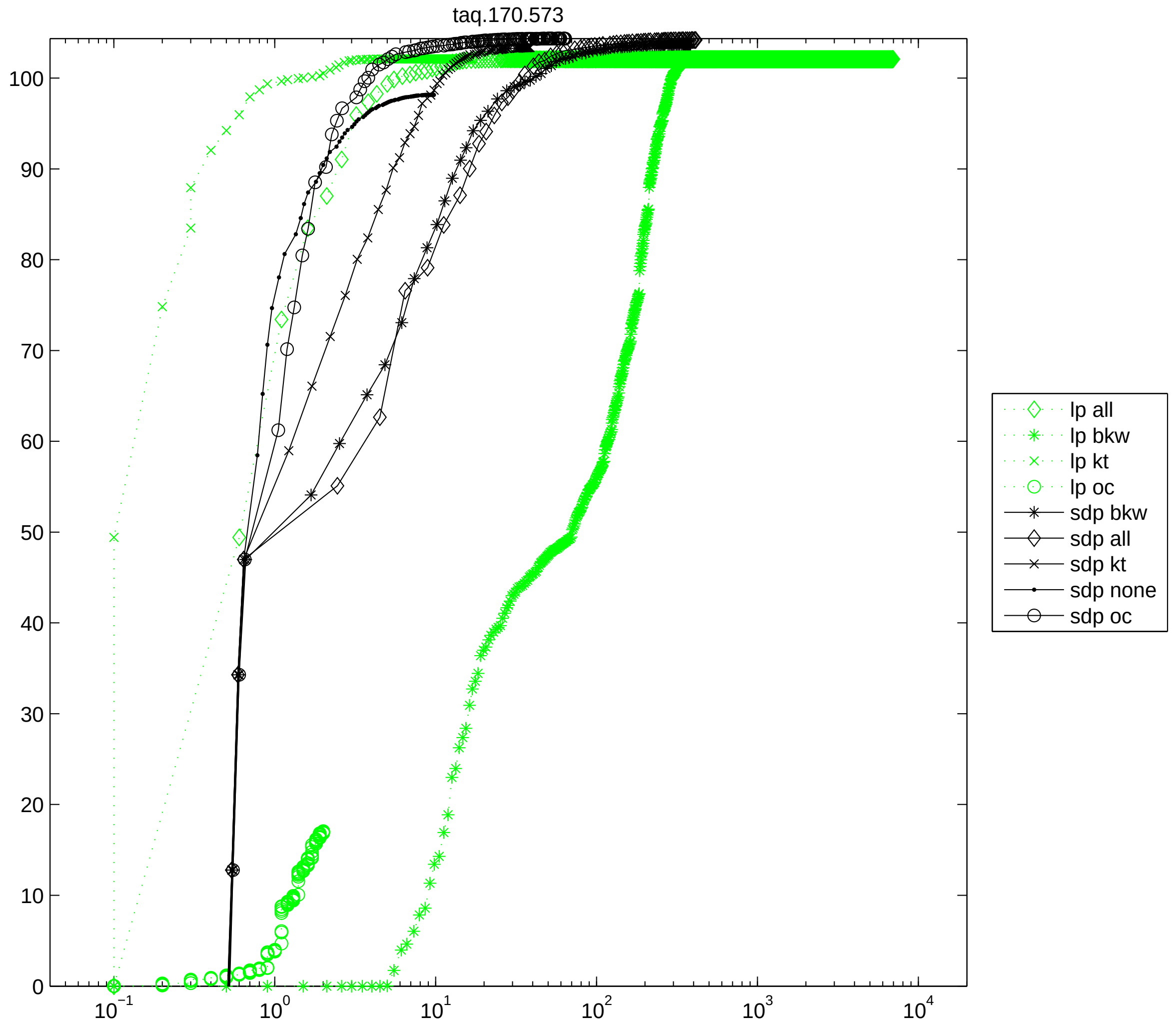
Separation routines

except for cycle ineq. (support extension!), both use the same separation routines for
knapsack star and bisection knapsack walk inequalities

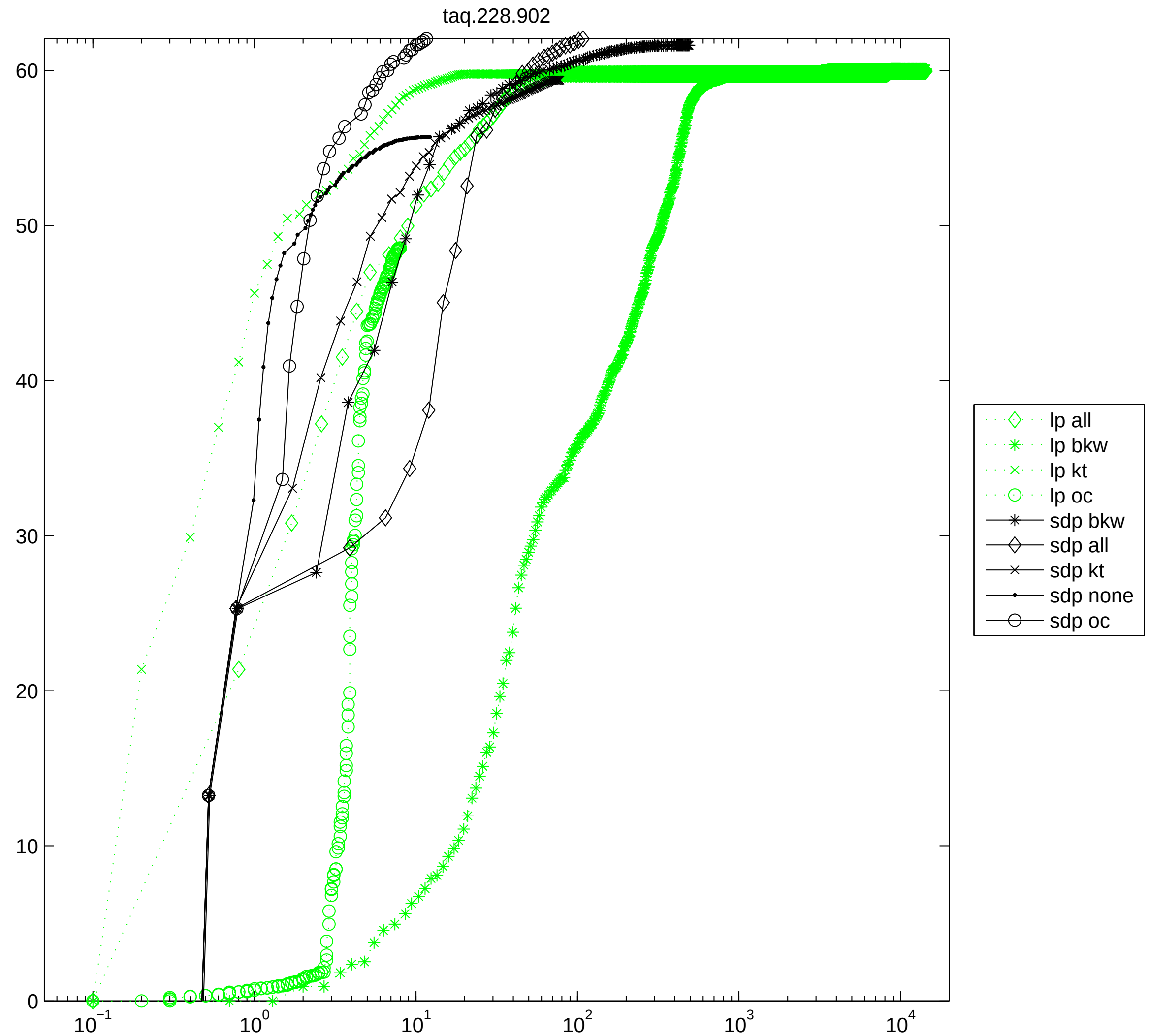
Root Node Lower Bound and Computation Time (VLSI design graphs)

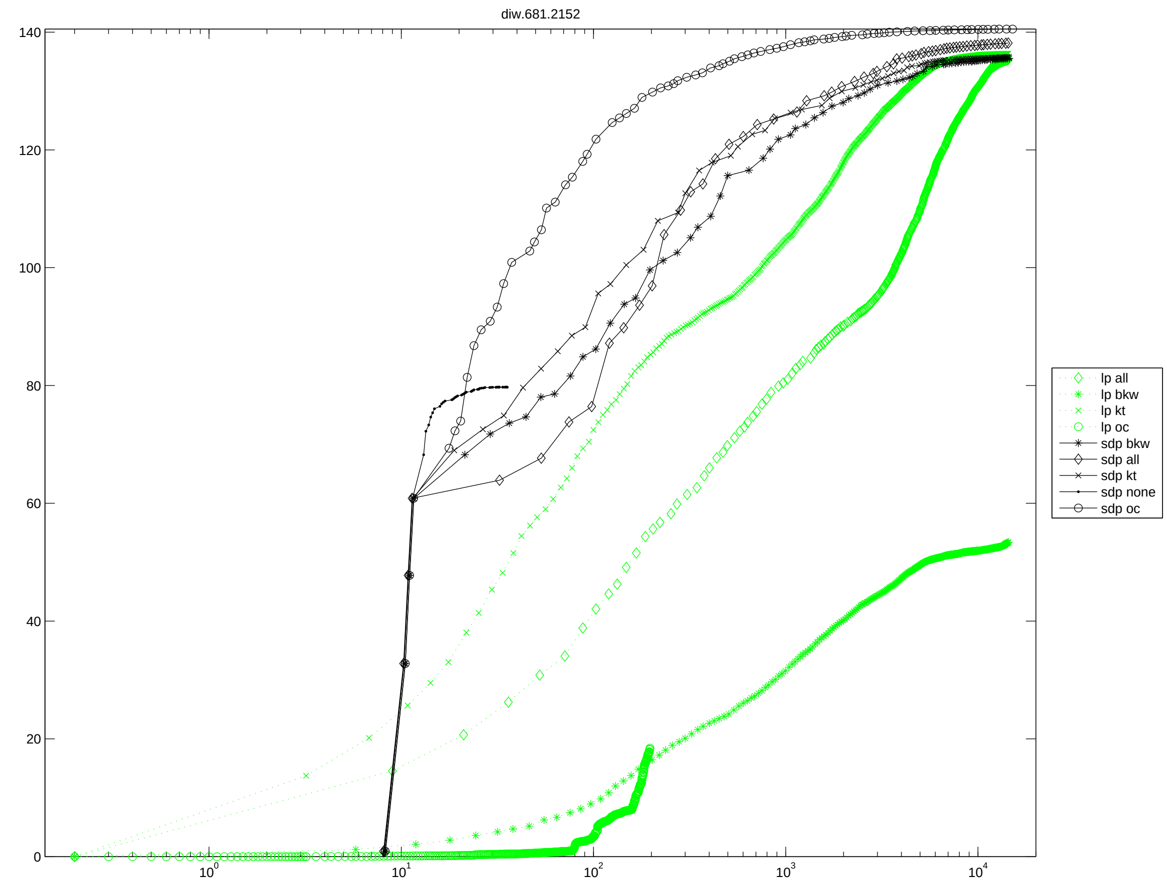
graph $n.m$	linear relaxation				semidefinite relaxation				
	cy	*cy+bkw*	*cy+kt*	*all*	*none*	*bkw*	*kt*	*cy*	
diw681.1494	18.4	134.9	136.3	135.9	77.3	135.1	134.7	140.6	1
taq1021.2253	23.1	74.1	113.2	113.9	60.1	112.9	112.4	116.8	1
dmxa1755.3686	0.0	42.6	87.1	91.1	37.5	89.3	89.0	92.9	
diw681.3104	34.8	238.4	744.4	829.2	630.7	954.6	935.0	988.8	9
taq334.3763	75.5	111.8	324.8	324.9	234.0	324.8	320.5	317.9	3
diw681.6402	46.8	136.2	304.6	306.3	280.8	320.7	310.4	319.7	3
gap2669.6182	8.6	28.3	71.1	73.7	35.8	72.7	74.0	74.0	
alut2292.6329	3.8	17.1	55.3	59.8	39.7	74.0	74.6	76.2	
taq1021.5480	74.1	154.4	639.8	689.9	1122.0	1510.8	1469.9	1540.6	15
dmxa1755.10867	20.4	31.7	137.1	138.6	94.6	143.0	142.0	143.5	1
alue6112.16896	0.0	8.2	7.8	31.0	52.9	117.6	99.9	135.1	
gap2669.24859	55.0	55.0	55.0	55.0	46.0	55.0	55.0	55.0	
taq1021.31641	151.8	215.1	372.5	374.6	359.4	386.6	301.8	398.6	3
alut2292.494500	559.0	740.3	1571.3	1966.2	53950	53374	46405	51071	47
mean lower bd	23	77	162	186	184	282	270	289	
mean time	40	8389	11646	10267	120	5684	3386	1476	5

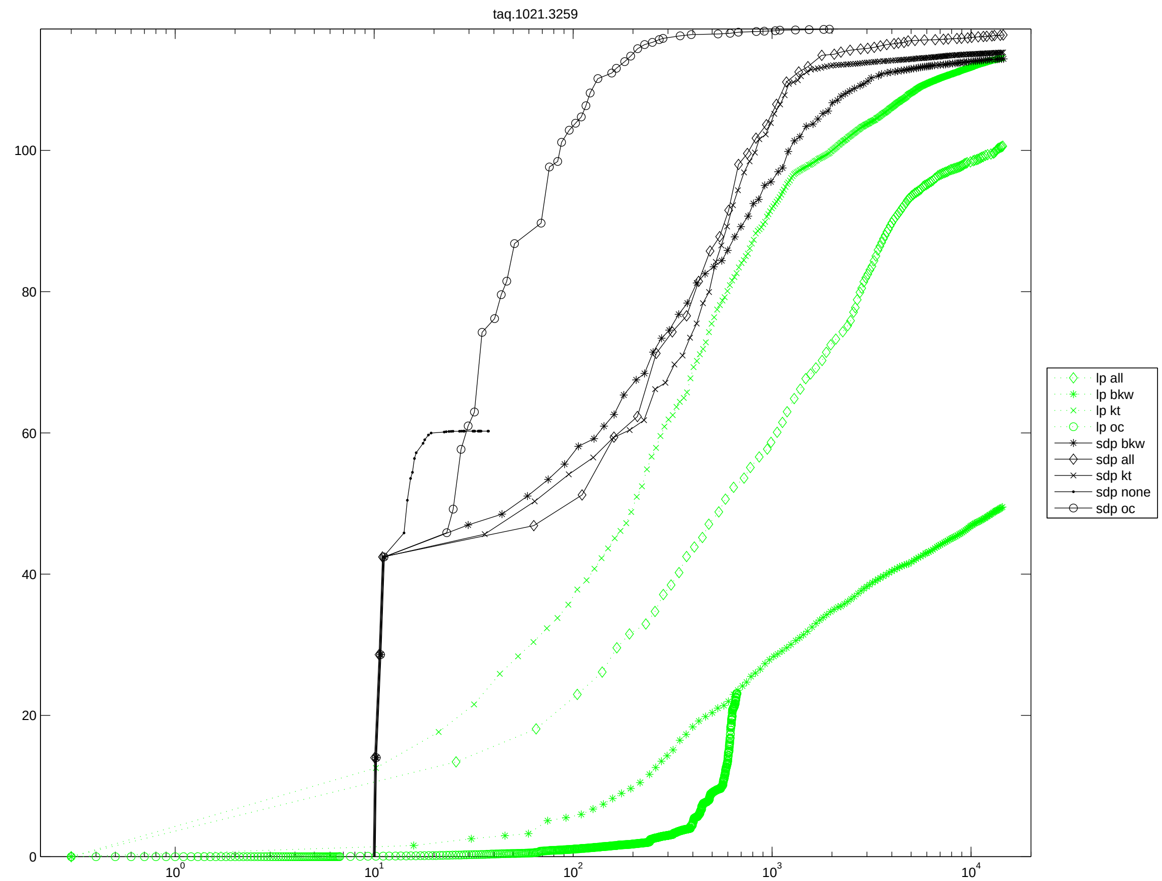
n number of nodes, m number of edges

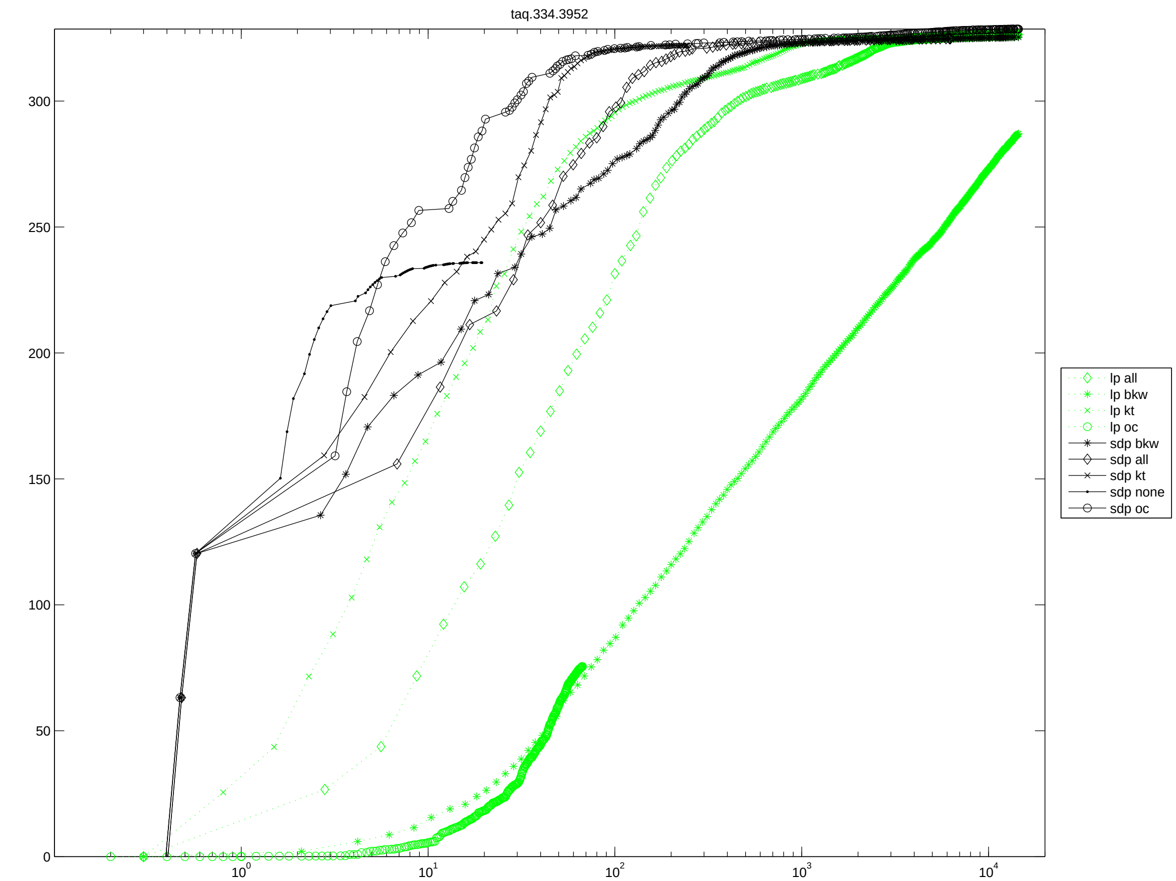


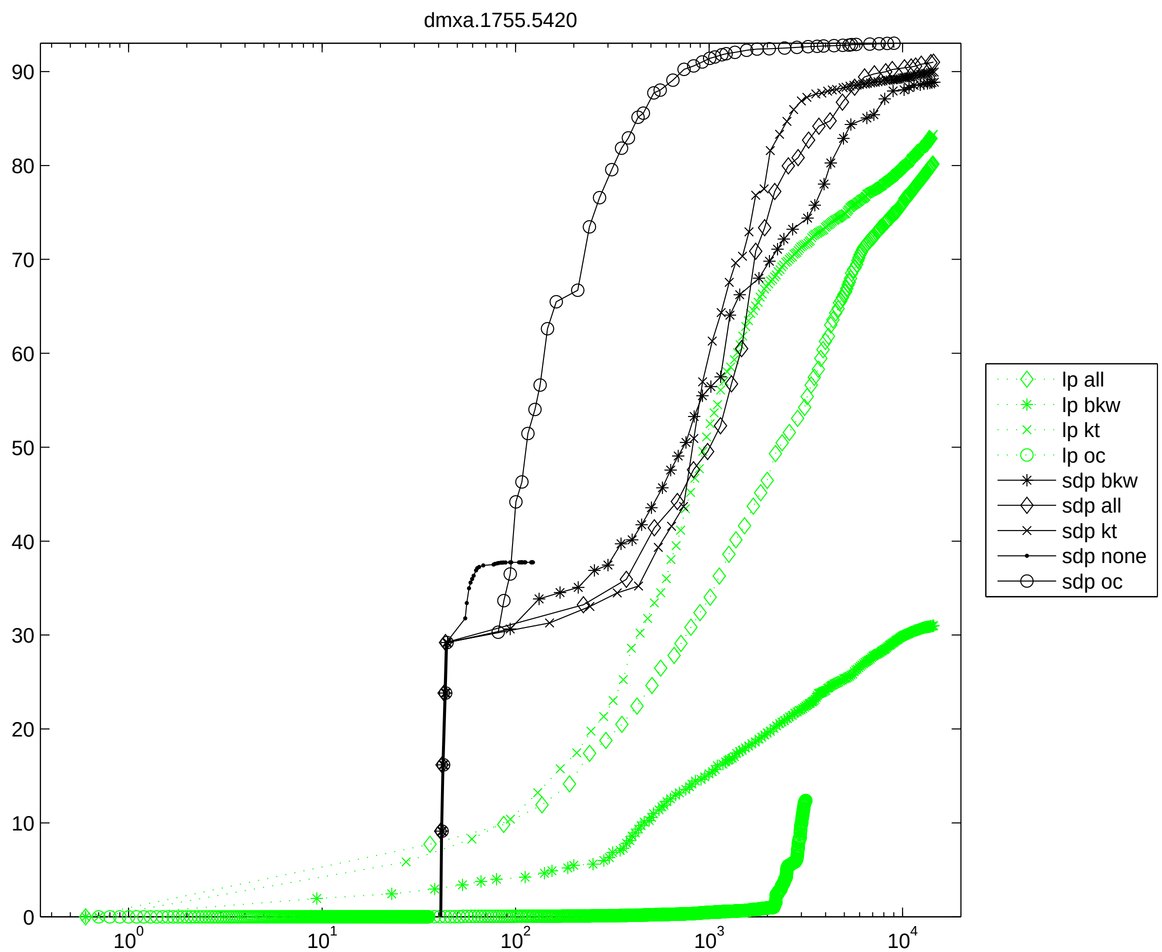
taq.170.573
0
10
20
30
40
50
60
70
80
90
100
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
lp all
lp bkw
lp kt
lp oc
sdp bkw
sdp all
sdp kt
sdp none
sdp oc

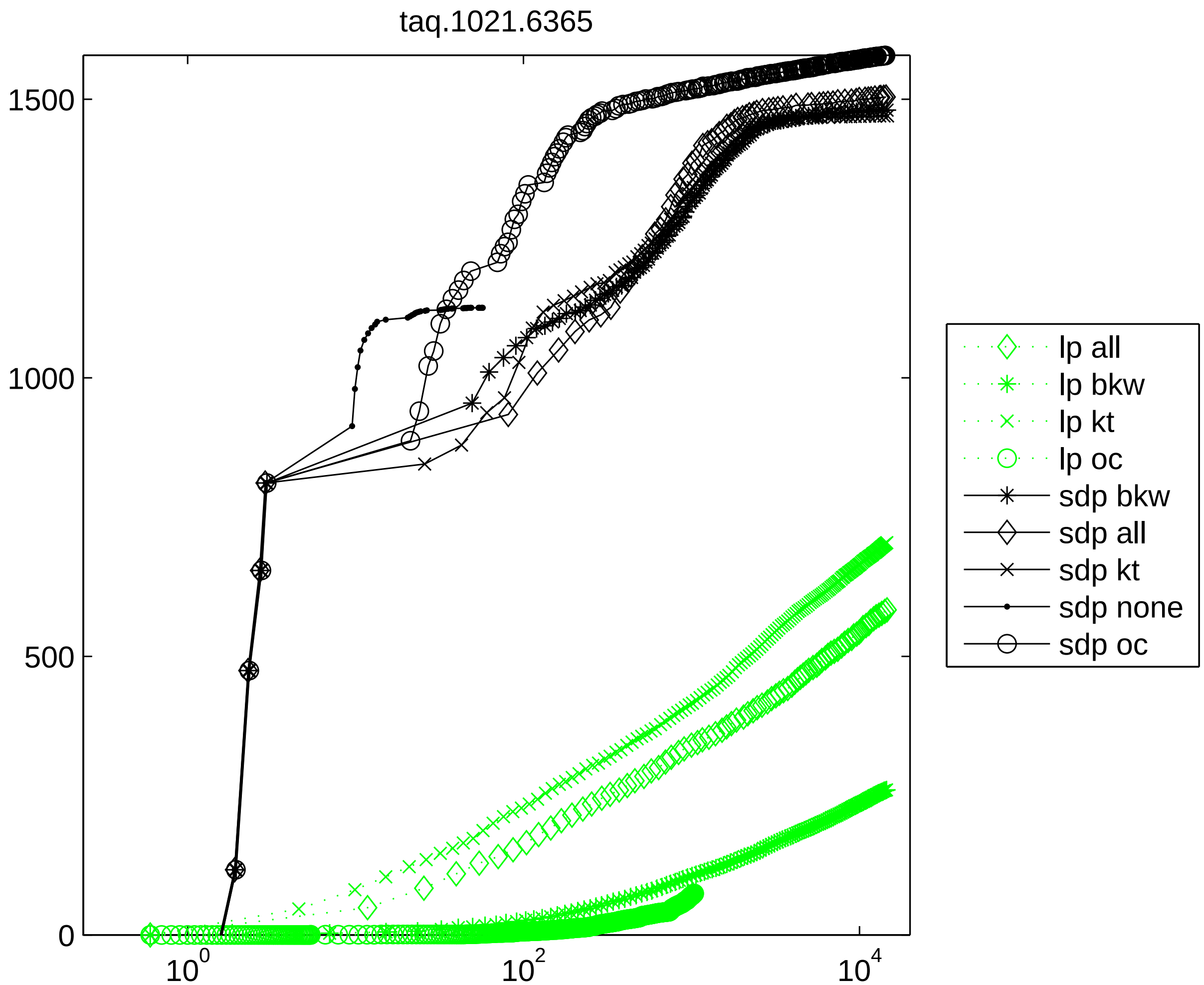

taq.228.902
0
10
20
30
40
50
60
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
lp all
lp bkw
lp kt
lp oc
sdp bkw
sdp all
sdp kt
sdp none
sdp oc


diw.681.2152
140
120
100
80
60
40
20
0
10^0
10^1
10^2
10^3
10^4
lp all
lp bkw
lp kt
lp oc
sdp bkw
sdp all
sdp kt
sdp none
sdp oc

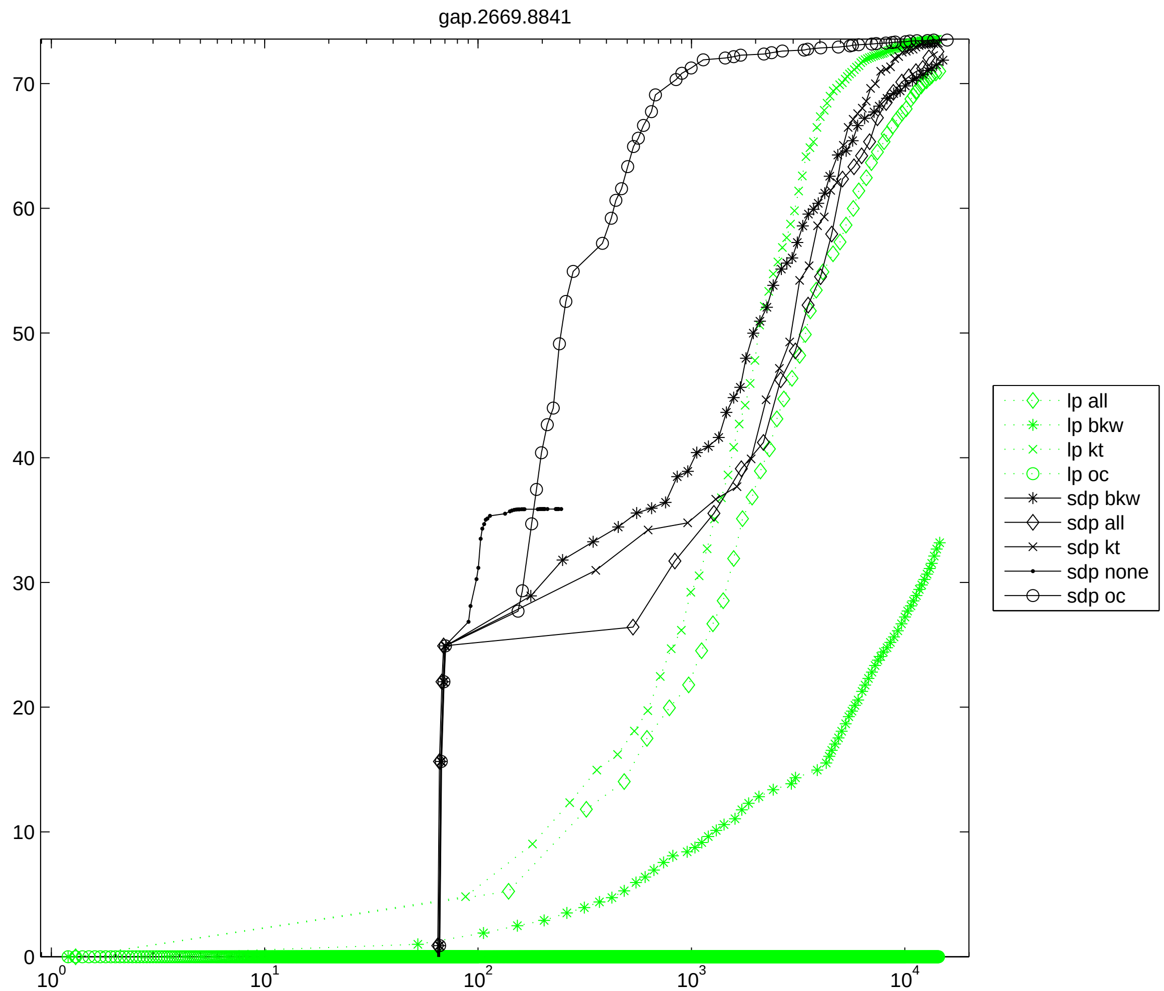

taq.1021.3259
0
20
40
60
80
100
10^0
10^1
10^2
10^3
10^4
lp all
lp bkw
lp kt
lp oc
sdp bkw
sdp all
sdp kt
sdp none
sdp oc


taq.334.3952
0
50
100
150
200
250
300
10^0
10^1
10^2
10^3
10^4
lp all
lp bkw
lp kt
lp oc
sdp bkw
sdp all
sdp kt
sdp none
sdp oc


dmxa.1755.5420
90
80
70
60
50
40
30
20
10
0
10^0
10^1
10^2
10^3
10^4
lp all
lp bkw
lp kt
lp oc
sdp bkw
sdp all
sdp kt
sdp none
sdp oc



taq.1021.6365
1500
1000
500
0
10^0
10^2
10^4
lp all
lp bkw
lp kt
lp oc
sdp bkw
sdp all
sdp kt
sdp none
sdp oc



gap.2669.8841
0
10
20
30
40
50
60
70
10^0
10^1
10^2
10^3
10^4
lp all
lp bkw
lp kt
lp oc
sdp bkw
sdp all
sdp kt
sdp none
sdp oc

Branch-and-Cut (VLSI design graphs)

graph $n.m$	linear relaxation					semidefinite relaxation			
	b&b nodes	*time*	*upper bound*	*lower bound*	*gap (%)*	*b&b nodes*	*time*	*upper bound*	*lower bound*
diw681.1494	1686	10 h	144	140.8	2	237	10 h	142	140.5
taq1021.2253	95	10 h	118	118.0	0	1	322 s	118	118.0
dmxa1755.3686	35	9 h	94	94	0	68	10 h	94	92.8
diw681.3104	1	10 h	1064	835.7	27	124	10 h	1011	1007.1
taq334.3763	351	4 h	342	342.0	0	2318	10 h	342	340.1
diw681.6402	3	10 h	357	315.2	13	159	10 h	331	329.2
gap2669.6182	1	4 h	74	74	0	1	651 s	74	74
alut2292.6329	1	10 h	77	69.5	10	96	10 h	77	76.1
taq1021.5480	1	10 h	2019	701.2	187	84	10 h	1650	1586.9
dmxa1755.10867	62	10 h	157	144.1	8	79	10 h	150	145.9
alue6112.16896	1	10 h	146	21.5	578	11	10 h	136	135.6
gap2669.24859	1	2525 s	55	55.0	0.0	1	491 s	55	55.0
taq1021.31641	1	10 h	426	375.3	13	9	10 h	404	399.0
alut2292.494500	1	*5 h	67815	1813.5	3639	1	5 h	67815	51880.0
geom. mean**	12	27144	208	156	8	42	17278	199	197

* Early termination due to a memory shortage, ** *alut2292.494500* excluded.

Branch-and-Cut (compiler design graphs [small!])

graph $n.m$	linear relaxation					semidefinite relaxation				
	# b&b nodes	*time (sec.)*	*upper bound*	*lower bound*	*gap (%)*	*# b&b nodes*	*time (sec.)*	*upper bound*	*lower bound*	*gap (%)*
cb.30.47	354	1	266	266	0	25166	540	266	266	[illegible]
cb.30.56	326	2	379	379	0	10276	256	379	379	[illegible]
cb.45.98	49	5	989	989	0	2995	438	989	989	[illegible]
cb.47.101	100	3	527	527	0	4403	433	527	527	[illegible]
cb.47.99	12	5	765	765	0	963	113	765	765	[illegible]
cb.61.187	785	81	2826	2826	0	10333	*5907	2826	2647	[illegible]

* Early termination due to a memory shortage.

Summary

- In Knapsack Tree Ineqs. the root should be chosen carefully
- Bisection Knapsack Walk Ineqs. are a specialization of Knapsack Trees,
  they are closely related to cycle inequalities,
  the best paths from the root to each node can be found in polynomial time
- For many bisection problems, current SDP-relaxation approaches
  are competitive if not superior to current LP-techniques,
  and this is also true for sparse large scale problems!

Open Problems

- Support extension proves extremely useful for SDP with cycle ineqs.
  What are good structural properties of this support?
- Are there structural indicators for when to use LP and when SDP?