Project-and-Lift for the Perspective Reformulation: How Serendipity Brought Us to a Free Lunch

Antonio Frangioni

Dipartimento di Informatica, Università di Pisa

with F. Furini, C. Gentile

Aussois 2013
January 9, 2013

Outline

The Hydro-thermal Unit Commitment problem

- Set P of thermal units and H of hydro (cascade) units
- Discretized time horizon $\mathcal{T}$, energy demand $\bar{d}_t$ for $t \in \mathcal{T}$
- Complicated technical constraints
- Operate the set of available units over $\mathcal{T}$ so as to satisfy demand
- A (not particularly smart) MIQP model of a "simple" version
  - $u^i_t \in \{0, 1\}$: ON/OFF state of thermal unit $i \in P$
  - $p^i_t \in \mathbb{R}_+$: power level of thermal unit $i \in P$
  - $q^j_t \in \mathbb{R}_+$: water discharge for hydro unit $j \in H(h)$ for cascade $h \in H$
- Objective function:

$$f(p, u) = \sum_{i \in P} \left(s^i(u^i) + \sum_{t \in \mathcal{T}} \left(a^i_t (p^i_t)^2 + b^i_t p^i_t + c^i_t u^i_t \right) \right) \qquad (1)$$

  - nonlinear convex energy cost ($a^i_t > 0$), fixed costs
  - time-dependent start-up costs $s^i(u^i)$ (only a few extra constraints and continuous variables with nifty formulation[1])

[1] Nowak, Römisch "Stochastic Lagrangian Relaxation Applied to Power Scheduling in a Hydro-thermal System Under Uncertainty", *Annals of Operations Research*, 2000

A MIQP formulation of UC

- Thermal units:

$$\bar{p}^i_{min} u^i_t \leq p^i_t \leq \bar{p}^i_{max} u^i_t \qquad t \in \mathcal{T} \quad (2)$$

$$p^i_t \leq p^i_{t-1} + u^i_{t-1}\Delta^i_+ + (1 - u^i_{t-1})\bar{l}^i \qquad t \in \mathcal{T} \quad (3)$$

$$p^i_{t-1} \leq p^i_t + u^i_t \Delta^i_- + (1 - u^i_t)\bar{u}^i \qquad t \in \mathcal{T} \quad (4)$$

$$u^i_t \leq 1 - u^i_{r-1} + u^i_r \qquad t \in \mathcal{T},\ r \in [t - \tau^i_+, t-1] \quad (5)$$

$$u^i_t \geq 1 - u^i_{r-1} - u^i_r \qquad t \in \mathcal{T},\ r \in [t - \tau^i_-, t-1] \quad (6)$$

- Hydro units:

$$0 \leq q^j_t \leq \bar{q}^j_{max} \qquad t \in \mathcal{T} \quad (7)$$

$$\bar{v}^j_{min} \leq v^j_t \leq \bar{v}^j_{max} \qquad t \in \mathcal{T} \quad (8)$$

$$v^j_t - v^j_{t-1} = \bar{w}^j_t - w^j_t - q^j_t + \textstyle\sum_{k \in \mathcal{S}(j)} \left(q^k_{t-t_{kj}} + w^k_{t-t_{kj}} \right) \qquad t \in \mathcal{T} \quad (9)$$

- Demand satisfaction (α^j = constant power-to-discharged water):

$$\textstyle\sum_{i \in P} p^i_t + \sum_{h \in H} \sum_{j \in H(h)} \alpha^j q^j_t = \bar{d}_t \qquad t \in \mathcal{T} \qquad (10)$$

Traditional solution approaches

- Large-scale (large $|P|$, $|H|$, $|\mathcal{T}|$) Mixed-Integer Quadratic Program to be solved in a few minutes
- Traditionally intractable for general-purpose MIQP/MILP solvers
- Traditional alternative: Lagrangian Relaxation of demand constraints (10), Lagrangian heuristic[2]
- Results actually quite good, especially if compared with `Cplex`
- A pesky Referee's comment: may the problem for Cplex be the Quadratic part? If so, piecewise-linearize f[3]
- Should this work? On the outset, we didn't see why ...

[2] F., Gentile, Lacalandra "Solving Unit Commitment Problems with General Ramp Contraints", *IJEPES*, 2008

[3] Carrión, Arroyo "A Computationally Efficient Mixed-integer Linear Formulation for the Thermal Unit Commitment Problem" *IEEE Transactions on Power Systems*, 2006

Results of the MILP formulation

. . . but it did, big times!

	MIQP			MILP				
	first	best	gap	time	gap	ftime	fgap	nodes
20	24	2229	0.29	3.72	0.36		1.00	0
50	249	1491	0.22	21.93	0.21	15.98	0.36	0
75	447	1514	0.10	56.31	0.20	47.08	1.62	10
100	940	2327	0.13	94.09	0.17	69.75	2.18	16
150	2348	2483	0.24(1)	218.69	0.12	177.35	6.58	16
200	3600	3600	* (5)	267.78	0.09	247.12	1.85	6

- Stopping tolerance at 0.5% (and invalid lower bound)
- Again, inherent gap vastly worse (and invalid anyway)
- Most of the difference is in the heuristic, as the LB is weaker (. . .)

Comparing MILP and LR

		RCDP			Cplex MILP					
p	h	time	gap	iter	time	gap	ftime	fgap	nodes	LPs
10	0	0.75	0.99	187	0.95	0.33		1.18	0	23
20	0	1.83	0.46	189	3.72	0.36		1.00	0	23
50	0	4.84	0.28	195	21.93	0.21	15.98	0.36	0	25
75	0	9.41	0.34	206	56.31	0.20	47.08	1.62	10	59
100	0	14.74	0.33	213	94.09	0.17	69.75	2.18	16	76
150	0	21.20	0.17	277	218.69	0.12	177.35	6.58	16	115
200	0	34.80	0.09	317	267.78	0.09	247.12	1.85	6	87
20	10	1.76	0.39	170	93.53	0.21		0.59	140	258
50	20	6.36	0.06	160	17.98	0.06	17.98	0.06	0	60
75	35	15.01	0.04	198	96.86	0.11	96.86	0.11	170	300
100	50	24.74	0.04	209	130.86	0.06	130.86	0.06	180	266
150	75	37.41	0.02	189	467.62	0.06	467.62	0.06	300	554
200	100	50.91	0.01	175	427.71	0.05	427.71	0.05	205	321

- Cplex primal heuristic impressively effective despite the LB being much worse . . . or is it?

Here Comes the Serendipity Moment

- Testing the MILP formulation to appease the pesky Referee (who happens to be right, albeit for the wrong reason: bad enough already)
- The LB of the MILP must be lower than that of the MIQP (which is lower than that of the LR)

Here Comes the Serendipity Moment

- Testing the MILP formulation to appease the pesky Referee (who happens to be right, albeit for the wrong reason: bad enough already)
- The LB of the MILP must be lower than that of the MIQP (which is lower than that of the LR)
- Instead it is way higher, but still valid (lower than the LR one)

Here Comes the Serendipity Moment

- Testing the MILP formulation to appease the pesky Referee (who happens to be right, albeit for the wrong reason: bad enough already)
- The LB of the MILP must be lower than that of the MIQP (which is lower than that of the LR)
- Instead it is way higher, but still valid (lower than the LR one)
- IT MUST BE A BUG ...

Here Comes the Serendipity Moment

- Testing the MILP formulation to appease the pesky Referee (who happens to be right, albeit for the wrong reason: bad enough already)
- The LB of the MILP must be lower than that of the MIQP (which is lower than that of the LR)
- Instead it is way higher, but still valid (lower than the LR one)
- IT MUST BE A BUG . . .
  and instead is a (n involuntary) reformulation which improves the LB!

Here Comes the Serendipity Moment

- Testing the MILP formulation to appease the pesky Referee (who happens to be right, albeit for the wrong reason: bad enough already)
- The LB of the MILP must be lower than that of the MIQP (which is lower than that of the LR)
- Instead it is way higher, but still valid (lower than the LR one)
- IT MUST BE A BUG . . .
  and instead is a (n involuntary) reformulation which improves the LB!
- After lots of head scratching, here's what had happened

The general MINLP framework

- Convex function f, Mixed-Integer NonLinear Program fragment

$$\min \{ f(p) + cu \ : \ Ap \leq bu \ , \ u \in \{0,1\} \ \} \qquad (11)$$

$p \in \mathcal{P} = \{ p \in \mathbb{R}^n \ : \ Ap \leq b \}$ compact $\equiv \{ p \ : \ Ap \leq 0 \} = \{0\}$

[4] F., Gentile "Perspective Cuts for a Class of Convex 0-1 Mixed Integer Programs", *Math. Prog.*, 2006

The general MINLP framework

- Convex function f, Mixed-Integer NonLinear Program fragment

$$\min \{ \, f(p) + cu \; : \; Ap \leq bu \; , \; u \in \{0,1\} \, \} \tag{11}$$

$p \in \mathcal{P} = \{ \, p \in \mathbb{R}^n \; : \; Ap \leq b \, \}$ compact $\equiv \{ \, p \; : \; Ap \leq 0 \, \} = \{0\}$

- Equivalently, minimize the nonconvex function

$$f(p,u) = \begin{cases} 0 & \text{if } u = 0 \text{ and } p = 0 \\ f(p) + c & \text{if } u = 1 \text{ and } Ap \leq b \\ +\infty & \text{otherwise} \end{cases} \tag{12}$$

[4] F., Gentile "Perspective Cuts for a Class of Convex 0-1 Mixed Integer Programs", *Math. Prog.*, 2006

The general MINLP framework

- Convex function f, Mixed-Integer NonLinear Program fragment

$$\min \{ f(p) + cu \;:\; Ap \leq bu \;,\; u \in \{0,1\} \} \tag{11}$$

$p \in \mathcal{P} = \{ p \in \mathbb{R}^n \;:\; Ap \leq b \}$ compact $\equiv \{ p \;:\; Ap \leq 0 \} = \{0\}$

- Equivalently, minimize the nonconvex function

$$f(p,u) = \begin{cases} 0 & \text{if } u = 0 \text{ and } p = 0 \\ f(p) + c & \text{if } u = 1 \text{ and } Ap \leq b \\ +\infty & \text{otherwise} \end{cases} \tag{12}$$

- Best possible convex relaxation of (11): use the convex envelope[4]

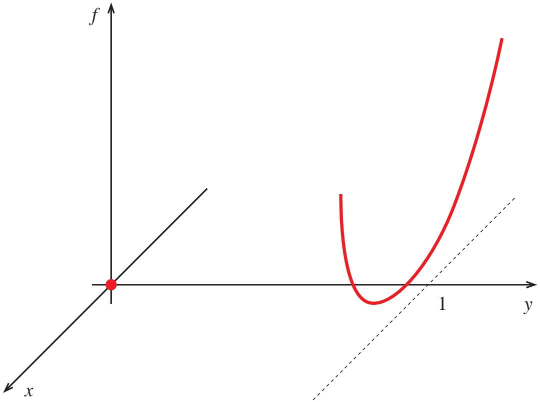
$$h(p,u) = \begin{cases} 0 & \text{if } p = 0 \text{ and } u = 0, \\ uf(p/u) + cu & \text{if } Ap \leq bu, \;\; u \in (0,1], \\ +\infty & \text{otherwise.} \end{cases} \tag{13}$$

(convex function minorizing $f(p,u)$ with smallest possible epigraph)

[4] F., Gentile "Perspective Cuts for a Class of Convex 0-1 Mixed Integer Programs", *Math. Prog.*, 2006

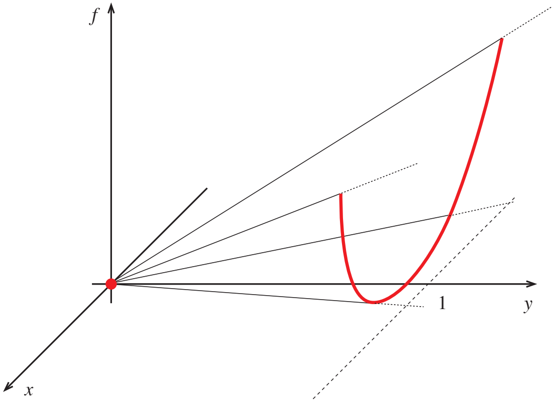
The Perspective what?

- $h(p, u)$ is a section of the perspective function $f(x, \lambda) = \lambda f(x/\lambda)$



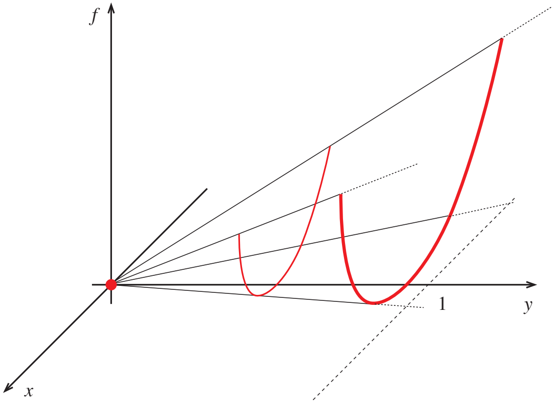
The Perspective what?

- $h(p, u)$ is a section of the perspective function $f(x, \lambda) = \lambda f(x/\lambda)$



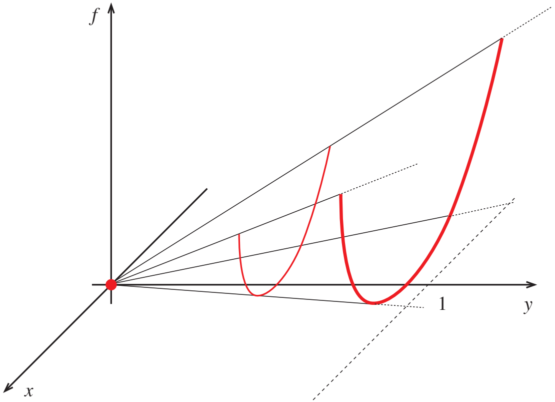
The Perspective what?

- $h(p, u)$ is a section of the perspective function $f(x, \lambda) = \lambda f(x/\lambda)$



The Perspective what?

- $h(p, u)$ is a section of the perspective function $f(x, \lambda) = \lambda f(x/\lambda)$

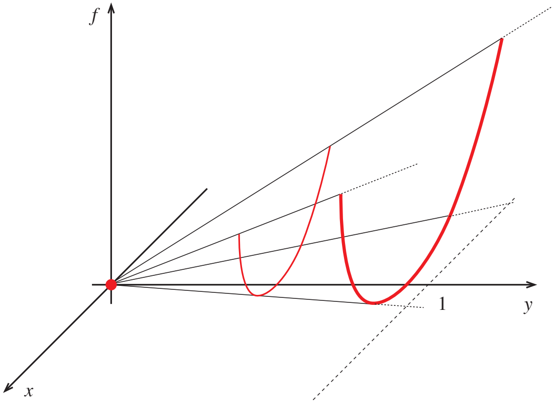


- $h(p, u)$ convex but much more nonlinear than $f(p) + cu$

  example: $f(p) = ap^2 + bp \quad \Rightarrow \quad h(p, u) = (a/u)p^2 + bp + cu$

The Perspective what?

- $h(p, u)$ is a section of the perspective function $f(x, \lambda) = \lambda f(x/\lambda)$



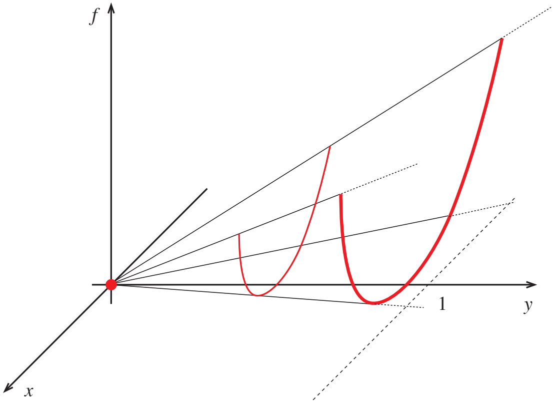
- $h(p, u)$ convex but much more nonlinear than $f(p) + cu$

example: $f(p) = ap^2 + bp \quad \Rightarrow \quad h(p, u) = (a/u)p^2 + bp + cu$

notes: I) $a/u > a$ for $u < 1$;

The Perspective what?

- $h(p, u)$ is a section of the perspective function $f(x, \lambda) = \lambda f(x/\lambda)$



- $h(p, u)$ convex but much more nonlinear than $f(p) + cu$

  example: $f(p) = ap^2 + bp \quad \Rightarrow \quad h(p, u) = (a/u)p^2 + bp + cu$

  notes: I) $a/u > a$ for $u < 1$; II) for $a = 0$ nothing happens

The Perspective Relaxation (Reformulation)

- A convex, continuous, more nonlinear program

$$\min \left\{ \; u f(p/u) + cu \;\; : \;\; Ap \leq bu \; , \;\; u \in \{0,1\} \; \right\} \tag{14}$$

$u \in \{0,1\}^n \Longrightarrow$ a reformulation! (if $0f(0/0) = 0$)

The Perspective Relaxation (Reformulation)

- A convex, continuous, more nonlinear program

$$\min \{ \; uf(p/u) + cu \; : \; Ap \leq bu \; , \; u \in \{0, 1\} \; \} \tag{14}$$

$u \in \{0, 1\}^n \implies$ a reformulation! (if $0f(0/0) = 0$)

- Lower bound much better, but how to solve it efficiently?

The Perspective Relaxation (Reformulation)

- A convex, continuous, more nonlinear program

$$\min \{ \ uf(p/u) + cu \ : \ Ap \leq bu \ , \ u \in \{0,1\} \ \} \qquad (14)$$

$u \in \{0,1\}^n \Longrightarrow$ a reformulation! (if $0f(0/0) = 0$)

- Lower bound much better, but how to solve it efficiently?

- Every convex function is the supremum of its affine minorants

$(v, p, u) \in epi\ h \iff Ap \leq bu$, $u \in [0,1]$, and $\forall \bar{p} \in \mathcal{P}$

$$v \geq f(\bar{p}) + c + [s \ , \ c + f(\bar{p}) - s\bar{p}] \begin{bmatrix} p - \bar{p} \\ u - 1 \end{bmatrix} \qquad \forall s \in \partial f(\bar{p}) \quad (15)$$

(infinitely many inequalities, at least one for each $\bar{p} \in \mathcal{P}$)

The Perspective Relaxation (Reformulation)

- A convex, continuous, more nonlinear program

$$\min \{ \ uf(p/u) + cu \ : \ Ap \leq bu \ , \ u \in \{0,1\} \ \} \qquad (14)$$

$u \in \{0,1\}^n \Longrightarrow$ a reformulation! (if $0f(0/0) = 0$)

- Lower bound much better, but how to solve it efficiently?

- Every convex function is the supremum of its affine minorants

$(v, p, u) \in epi\ h \iff Ap \leq bu$, $u \in [0,1]$, and $\forall \bar{p} \in \mathcal{P}$

$$v \geq f(\bar{p}) + c + [s \ , \ c + f(\bar{p}) - s\bar{p}] \begin{bmatrix} p - \bar{p} \\ u - 1 \end{bmatrix} \qquad \forall s \in \partial f(\bar{p}) \quad (15)$$

(infinitely many inequalities, at least one for each $\bar{p} \in \mathcal{P}$)

- The quadratic case: Perspective Cuts

$$v \geq (2a\bar{p} + b)p + (c - a\bar{p}^2)u \qquad \forall \bar{p} \in [p_{min}, p_{max}] \qquad (16)$$

(we happened to have inadvertently slipped in a pair of these)

Impact of the Perspective Relaxation (P/C)

P/C				CPLEX				
r.time	r.gap	time	nodes	r.time	r.gap	time	nodes	gap
4.17	0.28	15.61	3	1.41	2.36	10000	264179	1.27
4.29	0.13	4.53	1	1.80	0.49	62	1205	-
2.07	0.69	178.12	136	0.85	1.24	216	4083	-
8.64	0.28	37.14	4	1.61	2.40	10000	331732	1.43
8.42	0.20	23.75	2	1.71	1.63	10000	245582	0.87
6.71	0.24	12.59	2	1.58	1.37	10000	268516	0.73
4.83	0.28	12.71	3	0.87	2.23	10000	475400	1.45
5.97	0.18	19.35	3	1.74	1.06	6137	189898	-
6.73	0.23	44.35	44	1.55	2.60	10000	337915	1.69
7.96	0.26	141.69	73	1.64	2.28	10000	286651	1.02
5.98	0.28	48.98	57	1.48	1.77	7642	240516	0.85

- Root gap node greatly reduced at a small expense in running time
- Small instances ($p = 20$), no ramp constraints, gap $= 0.1\%$
- Nice thing is: you only need a `cutcallback`

Extension to nonseparable functions

- Another pesky referee wanted results on another problem than UC
- Mean-Variance problem with min and max buy-in thresholds

$$\min \left\{ x^T Q x \;\middle|\; \begin{array}{l} ex = 1 \;\;,\;\; \mu x \geq \rho \;, \\ l_i y_i \leq x_i \leq u_i y_i \;,\; y_i \in \{0,1\} \;\; i = 1, \ldots, n \end{array} \right\}$$

$\mu =$ expected return, $Q =$ covariance matrix, $\rho =$ desired return

- But f is nonseparable, so Perspective Reformulation not applicable

Extension to nonseparable functions

- Another pesky referee wanted results on another problem than UC
- Mean-Variance problem with min and max buy-in thresholds

$$\min \left\{ x^T Q x \;\middle|\; \begin{array}{l} ex = 1 \;\;,\;\; \mu x \geq \rho \;, \\ l_i y_i \leq x_i \leq u_i y_i \;,\; y_i \in \{0,1\} \;\; i = 1, \ldots, n \end{array} \right\}$$

$\mu =$ expected return, $Q =$ covariance matrix, $\rho =$ desired return

- But f is nonseparable, so Perspective Reformulation not applicable
- Dirty trick: choose $D \succeq 0$ diagonal s.t. $R = Q - D \succeq 0$

$$\min \left\{ x^T D x + z^T R z \;\middle|\; \begin{array}{l} ex = 1 \;\;,\;\; \mu x \geq \rho \;\;,\;\; z = x \\ l_i y_i \leq x_i \leq u_i y_i \;,\; y_i \in \{0,1\} \;\; i = 1, \ldots, n \end{array} \right\}$$

Move nonseparability to new variables p, let D "as large as possible"

- Trivial choice: $D = \lambda_{min}(Q) I$

The instances

- 30 randomly-generated instances for each $n \in \{200, 300, 400\}$
- $\mu_i \in [0.002, 0.01]$, $l_i \in [0.075, 0.125]$, $u_i \in [0.375, 0.425]$ (uniformly)
- Q = well-known random generator of [Pardalos, Rodgers '90]
- Effectiveness of Perspective Relaxation heavily impacted by

$$S = \text{average} \left\{ \frac{Q_{ii} - \sum_{j \neq i} |Q_{ij}|}{Q_{ii}} \; : \; i = 1, \ldots, n \right\}$$

  (dominance index)
- For each n, three classes of instances (10 each):
  - "+" instances, $S \approx 0.6$ (diagonally dominant)
  - "0" instances, $S \approx 0$ (diagonally quasi-dominant)
  - "−" instances, $S \approx -0.5$ (not diagonally dominant)
- Available at `http://www.di.unipi.it/optimize/Data`

Results of P/C

	P/C				Cplex			
	time	nodes	d.gap	r.gap	nodes	p.gap	d.gap	r.gap
200^+	904	7.7e+4		6.48	1.9e+7	0.14	45.33	85.63
200^0	320	2.8e+4		6.10	8.5e+6	0.38	51.27	84.47
200^-	3306	2.6e+5	0.02	6.69	8.9e+6	0.24	42.09	78.88
300^+	2061	9.3e+4		5.62	4.0e+6	0.41	64.68	92.01
300^0	1715	7.1e+4		6.28	3.6e+6	0.43	59.91	87.87
300^-	2797	9.4e+4	0.05	7.04	3.0e+6	0.53	45.11	78.77
400^+	4756	1.1e+5	0.10	6.15	1.9e+6	1.03	61.47	89.06
400^0	7421	1.6e+5	0.16	6.53	1.5e+6	1.18	68.68	90.03
400^-	6901	1.4e+5	0.36	6.49	1.5e+6	1.60	65.88	88.47

- Root node gap divided by 10: $< 8\%$ w.r.t. $> 80\%$
- Effectiveness worsens as Q less dominant, could not even solve all 200 instances (though much better than Cplex: none solved in 10000s)
- Perhaps a better D could help?

Choosing D via SDP

- Assuming $tr(D)$ the relevant metric, the "largest" D solves

$$\max \left\{ \sum_{i=1}^{n} d_i \;:\; Q - \sum_{i=1}^{n} d_i (e_i e_i^T) \succeq 0 \;,\; d \geq 0 \right\}$$
$$\min \left\{ tr(QX) \;:\; diag(X) \geq e \;,\; X \succeq 0 \right\}$$

  dual pair of SemiDefinite (= convex = easy) Problems

- Several efficient, open-source SDP codes

- Interesting relaxation: removing $d \geq 0$ in the primal gives

$$\min \left\{ tr(QX) \;:\; diag(X) = e \;,\; X \succeq 0 \right\}$$

  - $d^* > 0$ anyway in all our tests
  - most often faster to solve in practice by all codes
  - *constant trace* = max eigenvalue problem, specialized approaches

- Is the SDP running time low? Is this worth?

Comparison of SDP codes

	d_{max}	d_{min}	d_{avg}	ME	CSDP $\geq$	CSDP $=$	DSDP $\geq$	DSDP $=$	SDPA $\geq$	SDPA $=$	SDPLR $\geq$	SDPLR $=$	SB $=$
200^+	1.96	0.97	1.47	0.13	3.12	2.98	1.86	0.10	1.81	0.29	3.71	2.23	23.77
200^0	1.93	0.90	1.41	0.13	3.03	2.99	1.87	0.10	1.68	0.29	3.72	2.79	16.39
200^-	1.86	0.87	1.37	0.13	3.00	2.95	1.86	0.10	1.62	0.40	2.30	2.19	16.58
300^+	1.97	0.97	1.47	0.23	10.54	9.84	4.92	0.26	5.33	0.73	13.20	5.02	69.13
300^0	1.93	0.91	1.42	0.23	10.91	9.55	4.99	0.26	4.97	0.71	8.58	9.08	46.01
300^-	1.69	0.89	1.29	0.23	10.91	9.62	5.10	0.26	5.11	0.72	5.67	5.53	41.82
400^+	1.98	0.97	1.47	0.39	31.03	29.28	10.56	0.52	5.02	1.40	17.48	21.60	146.07
400^0	1.93	0.93	1.43	0.39	37.24	31.27	10.86	0.52	11.46	1.37	21.80	11.93	94.62
400^-	1.87	0.89	1.38	0.39	36.77	31.61	10.75	0.52	11.10	1.38	15.10	21.11	90.07

- On average 50% better than λ_{min}, worst case $\approx$ few % worse
- Results getting worse as Q less diagonally dominant
- Times not much worse using right code and model
- Is it worth?

Impact on the B&C

	SDP				ME			
	time	nodes	d.gap	r.gap	time	nodes	d.gap	r.gap
200^+	164	1.2e+4		1.14	904	7.7e+4		6.48
200^0	161	1.1e+4		2.14	320	2.8e+4		6.10
200^-	1902	1.3e+5		3.65	3306	2.6e+5	0.02	6.69
300^+	818	2.9e+4		4.54	2061	9.3e+4		5.62
300^0	856	2.7e+4		1.97	1715	7.1e+4		6.28
300^-	1709	5.2e+4		2.68	2797	9.4e+4	0.05	7.04
400^+	2264	7.0e+4		4.79	4756	1.1e+5	0.10	6.15
400^0	4378	7.2e+4	0.10	2.29	7421	1.6e+5	0.16	6.53
400^-	6311	1.0e+5	0.23	3.06	6901	1.4e+5	0.36	6.49

- root node gap halved+ w.r.t. ME
- All instances up to $n = 300$ solved to optimality within 10000s
- Effectiveness still worsens as Q less dominant, can't solve a few 400^-

An Alternative: The Conic Program Reformulation

- $\lambda f(x/\lambda)$ is SOCP-representable if f is [Ben Tal '02]

[5] F., Gentile "A Computational Comparison of [. . .]: SOCP vs. Cutting Planes" *Op. Res. Letters*, 2009

An Alternative: The Conic Program Reformulation

- $\lambda f(x/\lambda)$ is SOCP-representable if f is [Ben Tal '02]
- Some pesky people insisted in ignoring (16) and using instead

$$\min \left\{ t + bp + cu : \sqrt{ap^2 + (t-u)^2/4} \leq (t+u)/2 \ldots u \in \{0,1\} \right\}$$

  a Mixed-Integer Second-Order Cone Program: Cplex can solve it ...

[5] F., Gentile "A Computational Comparison of [...]: SOCP vs. Cutting Planes" *Op. Res. Letters*, 2009

An Alternative: The Conic Program Reformulation

- $\lambda f(x/\lambda)$ is SOCP-representable if f is [Ben Tal '02]

- Some pesky people insisted in ignoring (16) and using instead

  $$\min \left\{ t + bp + cu : \sqrt{ap^2 + (t-u)^2/4} \leq (t+u)/2 \ldots u \in \{0,1\} \right\}$$

  a Mixed-Integer Second-Order Cone Program: Cplex can solve it ...

		P/C					CP				
p	h	gap	nodes	LPs	time	t/LP	gap	nodes	CPs	time	t/CP
10	0		4.3e2	7.8e2	14	0.018		5.8e2	1.0e3	20	0.021
20	0		5.0e4	5.8e4	6805	0.094		6.6e4	7.5e4	13392	0.145
50	0	0.08	1.7e5	2.1e5	86400	0.421	0.08	9.1e4	1.1e5	86400	0.781
20	10		1.1e4	1.3e4	161	0.014		1.4e4	1.8e4	626	1.937
50	20		5.5e5	6.6e5	29874	0.037	0.00	5.0e5	6.1e5	86400	0.460
75	35	0.01	8.5e5	1.0e6	73076	0.073	0.01	1.8e5	2.2e5	86400	0.314

... but it's not a great idea (24h = very unrealistic, gap = 0.01%)[5]

[5] F., Gentile "A Computational Comparison of [...]: SOCP vs. Cutting Planes" *Op. Res. Letters*, 2009

Comparing CP with P/C on MV

	P/C				CP				
	nodes	QPs	time	t/QP	nodes	CPs	time	t/CP	gap
200^+	1.9e4	1.9e4	194	0.0008	9.2e3	1.1e3	17961	1.578	0.15(1)
200^0	1.7e4	1.8e4	90	0.0007	2.7e4	3.2e4	30785	1.648	0.32(2)
200^-	1.2e5	1.3e5	835	0.0006	1.6e4	1.9e5	55144	1.719	1.02(5)
300^+	3.4e4	3.5e4	433	0.0014	1.1e4	1.4e4	72075	8.334	0.58(7)
300^0	3.1e5	3.3e4	378	0.0019	1.0e4	1.3e4	59591	4.464	0.53(6)
300^-	5.5e5	5.8e4	654	0.0014	1.1e4	1.3e4	66863	5.272	0.81(7)
400^+	7.9e4	8.2e4	2066	0.0032	4.7e3	5.9e3	61810	10.397	1.01(6)
400^0	2.3e5	2.4e5	3974	0.0020	6.1e3	7.6e3	83782	10.588	1.79(9)
400^-	3.3e5	3.4e5	8092	0.0026	6.3e3	7.9e3	80382	10.764	2.71(8)

- Same SDP solved for CP and P/C
- QPs awesomely faster than CPs to solve ("quadratic simplex") even invoking the built-in SOCP linearization(?)
- Faster bound $\Longrightarrow$ more nodes $\Longrightarrow$ faster convergence
- Best case (very unstructured problem)

Alternatives to P/C and CP to solve the PR

- We tried quite a few (Newton, . . .) before finding CP

Alternatives to P/C and CP to solve the PR

- We tried quite a few (Newton, ...) before finding CP
- The (failed) Newton idea eventually led us to projecting

Alternatives to P/C and CP to solve the PR

- We tried quite a few (Newton, ...) before finding CP
- The (failed) Newton idea eventually led us to projecting
- Only works under strong assumptions (often true):
  - A1 p is a single variable, $\bar{p}_{min} \geq 0$
  - A2 f is quadratic
  - A3 there are no constraints linking different u
- Basic idea: recast the PR as

$$\min \{ \; z(p) \; : \; p \in [0, \bar{p}_{max}] \; \}$$

where $z(p)$ (partial minimization of a convex function $\Rightarrow$ convex) is

$$z(p) = bp + \min_u \{ \; ap^2/u + cu \; : \; u\bar{p}_{min} \leq p \leq u\bar{p}_{max} \; , \; p \in [0,1] \; \}$$

(intuition: the projection will be "less nonlinear")

Derivation

- Algebraic characterization of $z(p)$ out of optimal solution $u^*(p)$[6]
- In turn, $u^*(p)$ out of unconstrained minimizer $\tilde{u}(p)$, i.e., solution to

$$\frac{\partial h(p,u)}{\partial u} = c - ap^2/p^2 = 0$$

(if any)

  - if $\tilde{u}(p)$ is feasible, then $u^*(p) = \tilde{u}(p)$
  - otherwise, $u^*(p)$ is the projection of $\tilde{u}(p)$ over the feasible region (easy because the feasible region is very simple)
  - note: $ap^2/u^2 \geq 0$

- Basically, a few algebraic computations depending on a, c, ...

[6] F., Gentile, Grande, Pacifici "Projected Perspective Reformulations with Applications in Design Problems" *Operations Research* 2010

The form of $z(p)$

1) $c \leq 0 \Rightarrow \tilde{u}(p)$ undefined $\Rightarrow u^*(p) = 1 \Rightarrow$

$$\boxed{z(p) = ap^2 + bp + c}$$

2) $c > 0 \Rightarrow \tilde{u}(p) = p\sqrt{a/c}$

2.1) $\tilde{u} \leq p/\bar{p}_{max} \Leftrightarrow \bar{p}_{max} \leq \sqrt{c/a} \Leftrightarrow u^*(p) = p/\bar{p}_{max} \Rightarrow$

$$\boxed{z(p) = (\ b + a\bar{p}_{max} + c/\bar{p}_{max}\ )\,p}$$

2.2) $0 \geq \tilde{u}(p) \geq p/\bar{p}_{max} \Leftrightarrow \bar{p}_{max} \geq \sqrt{c/a}\ (\geq \bar{p}_{min})$.

- $(\bar{p}_{max} \geq)\ p \geq \sqrt{c/a}\ (\geq 0) \Rightarrow \tilde{u}(p) \geq 1 \Rightarrow u^*(p) = 1$;
- $0 \leq p \leq \sqrt{c/a}\ (\leq \bar{p}_{max}) \Rightarrow \tilde{u}(p) \leq 1 \Rightarrow u^*(p) = \tilde{u}(p)$.

$$\Rightarrow \boxed{z(p) = \begin{cases} (\ b + 2\sqrt{ac}\ )\,p & \text{if } 0 \leq p \leq \sqrt{c/a} \\ ap^2 + bp + c & \text{if } \sqrt{c/a} \leq p \leq \bar{p}_{max} \end{cases}}$$

The form of $z(p)$

1) $c \leq 0 \Rightarrow \tilde{u}(p)$ undefined $\Rightarrow u^*(p) = 1 \Rightarrow$

$$\boxed{z(p) = ap^2 + bp + c}$$

2) $c > 0 \Rightarrow \tilde{u}(p) = p\sqrt{a/c}$

2.1) $\tilde{u} \leq p/\bar{p}_{max} \Leftrightarrow \bar{p}_{max} \leq \sqrt{c/a} \Leftrightarrow u^*(p) = p/\bar{p}_{max} \Rightarrow$

$$\boxed{z(p) = (\ b + a\bar{p}_{max} + c/\bar{p}_{max}\ )p}$$

2.2) $0 \geq \tilde{u}(p) \geq p/\bar{p}_{max} \Leftrightarrow \bar{p}_{max} \geq \sqrt{c/a}\,(\geq \bar{p}_{min})$.

- $(\bar{p}_{max} \geq)\ p \geq \sqrt{c/a}\,(\geq 0) \Rightarrow \tilde{u}(p) \geq 1 \Rightarrow u^*(p) = 1$;
- $0 \leq p \leq \sqrt{c/a}\,(\leq \bar{p}_{max}) \Rightarrow \tilde{u}(p) \leq 1 \Rightarrow u^*(p) = \tilde{u}(p)$.

$$\Rightarrow \boxed{z(p) = \begin{cases} (\ b + 2\sqrt{ac}\ )p & \text{if } 0 \leq p \leq \sqrt{c/a} \\ ap^2 + bp + c & \text{if } \sqrt{c/a} \leq p \leq \bar{p}_{max} \end{cases}}$$

- $z(p)$ convex differentiable piecewise-quadratic with ≤ 2 pieces

Application: Quadratic (1-Commodity) Network Design

- Directed graph $G = (N, A)$, deficit b_i for $i \in N$, arc capacity $\bar{p}_{ij}$ with fixed-charge cost $c_{ij} > 0$, quadratic routing cost $b_{ij} p_{ij} + a_{ij} p_{ij}^2$

$$
\begin{aligned}
\min \quad & \textstyle\sum_{(i,j)\in A} c_{ij} u_{ij} + b_{ij} p_{ij} + a_{ij} p_{ij}^2 \\
& \textstyle\sum_{(j,i)\in A} p_{ji} - \sum_{(i,j)\in A} p_{ij} = b_i \quad i \in N \\
& 0 \leq p_{ij} \leq \bar{p}_{ij} u_{ij} \;\; , \;\; u_{ij} \in \{0, 1\} \quad (i,j) \in A
\end{aligned}
\tag{17}
$$

- In continuous relaxation, $u_{ij} = p_{ij} / \bar{p}_{ij} \Longrightarrow$ a quadratic flow problem
- Very weak bound, PR improves it but destroys flow structure

Application: Quadratic (1-Commodity) Network Design

- Directed graph $G = (N, A)$, deficit b_i for $i \in N$, arc capacity $\bar{p}_{ij}$ with fixed-charge cost $c_{ij} > 0$, quadratic routing cost $b_{ij} p_{ij} + a_{ij} p_{ij}^2$

$$\begin{array}{ll} \min & \sum_{(i,j) \in A} c_{ij} u_{ij} + b_{ij} p_{ij} + a_{ij} p_{ij}^2 \\ & \sum_{(j,i) \in A} p_{ji} - \sum_{(i,j) \in A} p_{ij} = b_i \quad i \in N \\ & 0 \leq p_{ij} \leq \bar{p}_{ij} u_{ij} \;\;,\;\; u_{ij} \in \{0, 1\} \quad (i,j) \in A \end{array} \tag{17}$$

- In continuous relaxation, $u_{ij} = p_{ij} / \bar{p}_{ij} \Longrightarrow$ a quadratic flow problem
- Very weak bound, PR improves it but destroys flow structure
- P²R: Separable Convex Quadratic MCF on $G' = (N, A')$ with $|A'| \leq 2|A|$ (same nodes, duplicated arcs)

$$\begin{array}{ll} \min & \sum_{(i,j) \in A'} b'_{ij} r_{ij} + a'_{ij} r_{ij}^2 \\ & \sum_{(j,i) \in A'} r_{ji} - \sum_{(i,j) \in A'} r_{ij} = b_i \quad i \in N \\ & 0 \leq r_{ij} \leq \bar{r}_{ij} \quad (i,j) \in A' \end{array}$$

Computational results

- Randomly generated Network Design instances:
  - standard DIMACS random MCF problem generator (`netgen`)
  - linear costs randomly generated in a given interval
  - quadratic/fixed costs randomly generated with respect to linear costs (two different ways, "h" – high and "l" – low)
- Not as difficult as expected (many solved at root node)
- 2-pieces linear/quadratic P^2/R solved with `CPLEX-qpopt`
  $\Rightarrow$ not really a specialized algorithm
- Using "true" MCF solver possible with further piecewise-linearization but static version not competitive (dynamic may be)
- P^2/R very efficient (but helped by few branching/cuts)
- Non P/R B&C and Conic Program formulation of P/R much slower

Network Design — the table

name	time	nodes	t/n	time	nodes	t/n	gap
	P^2/R			CPLEX			
2000-h-h	0.10	1	0.10	690.09	101868	0.11	0.00
2000-h-l	45.42	278	1.10	1031.75	141485	0.01	0.06
2000-l-h	0.09	1	0.09	858.22	131954	0.03	0.00
2000-l-l	8.78	63	0.10	1036.79	140877	0.01	0.04
3000-h-h	0.15	1	0.15	1041.96	88541	0.01	0.00
3000-h-l	71.02	269	0.17	1051.93	73591	0.01	0.12
3000-l-h	0.15	1	0.15	988.74	89209	0.12	0.00
3000-l-l	19.05	79	0.16	1062.45	85878	0.01	0.04
	P/C			CP			
2000-h-h	57.09	7	13.84	895.70	8	207.60	0.01
2000-h-l	51.60	348	0.72	252.98	36	27.65	0.00
2000-l-h	42.3	6	16.57	525.35	9	63.35	0.00
2000-l-l	20.60	131	0.51	252.82	193	40.02	0.00
3000-h-h	117.30	11	18.90	564.41	2	407.97	0.01
3000-h-l	140.47	584	1.39	366.95	27	36.76	0.00
3000-l-h	101.18	12	12.01	372.16	4	89.53	0.01
3000-l-l	45.43	153	0.89	292.41	83	62.39	0.00

Approximated P^2R: Project&Lift

- Approximating the PR may work
- Biggest roadblocks in P^2R:
  - separability of u (UC does not have it)
  - need for a special structure to be exploited (MV does not have it)
  - no $u \Longrightarrow$ home-made branch, cut, fixing, ...
- But for non-separable u, computing the projection is too complicated

Approximated P^2R: Project&Lift

- Approximating the PR may work
- Biggest roadblocks in P^2R:
  - separability of u (UC does not have it)
  - need for a special structure to be exploited (MV does not have it)
  - no $u \Longrightarrow$ home-made branch, cut, fixing, ...
- But for non-separable u, computing the projection is too complicated
- Possible solution: project as if u were separable, re-introduce them

Approximated P^{2}R: Project&Lift

- Approximating the PR may work
- Biggest roadblocks in P^{2}R:
  - separability of u (UC does not have it)
  - need for a special structure to be exploited (MV does not have it)
  - no $u \Longrightarrow$ home-made branch, cut, fixing, ...
- But for non-separable u, computing the projection is too complicated
- Possible solution: project as if u were separable, re-introduce them
- Meanwhile, a few useful generalizations:
  - Extend to nonquadratic f provided that for fixed $g^{\pm}$

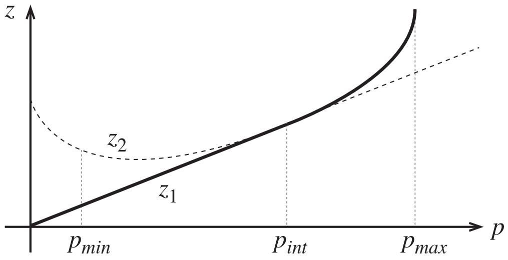
$$\tilde{u}(p) = \begin{cases} pg^{+} & \text{if } p \geq 0 \\ -pg^{-} & \text{if } p \leq 0 \end{cases} \quad (18)$$

    is the unique stationary point of $h(p, u)$ with respect to u
    (many cases: p^k, e^p, Kleinrock delay function, ...)
  - Extend to $\bar{p}_{min} < 0$ (4-pieces z instead of 2-pieces one)

Project&Lift

- The general form ($\bar{p}_{min} \geq 0$): for $p_{int} \in \{ \bar{p}_{min} \, , \, 1/g^+ \, , \, \bar{p}_{max} \}$

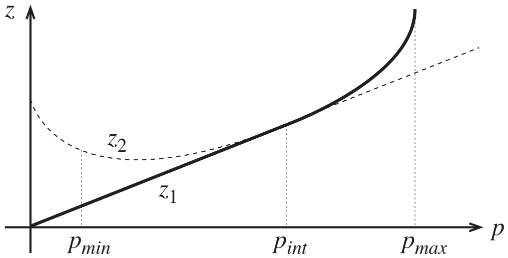
$$z(p) = \begin{cases} z_1(p) = \left(\, b + f(p_{int})/p_{int} + c/p_{int} \, \right) p & 0 \leq p \leq p_{int} \\ z_2(p) = f(p) + bp + c & p_{int} \leq p \leq \bar{p}_{max} \end{cases}$$



Project&Lift

- The general form ($\bar{p}_{min} \geq 0$): for $p_{int} \in \{ \; \bar{p}_{min} \; , \; 1/g^+ \; , \; \bar{p}_{max} \; \}$

$$z(p) = \begin{cases} z_1(p) = \left(\; b + f(p_{int})/p_{int} + c/p_{int} \; \right) p & 0 \leq p \leq p_{int} \\ z_2(p) = f(p) + bp + c & p_{int} \leq p \leq \bar{p}_{max} \end{cases}$$



- Theorem: can be written as a NLP with the u "making" z_1:

$$z(p) = \begin{cases} \min \;\; h(u,q) = & uz(p_{int}) + z_2(q + p_{int}) - z(p_{int}) \\ & (\bar{p}_{min} - p_{int})u \leq q \leq (\bar{p}_{max} - p_{int})u \\ & p = p_{int}u + q \qquad , \qquad u \in [0,1] \end{cases}$$

Approximated Projected Perspective Reformulation

- With the integrality constraints, a reformulation of the block

$$\begin{aligned} \min \quad & uz(p_{int}) + z_2(q + p_{int}) - z(p_{int}) \\ & (\bar{p}_{min} - p_{int})u \leq q \leq (\bar{p}_{max} - p_{int})u \\ & p = p_{int}u + q \qquad , \qquad u \in \{0,1\} \end{aligned}$$

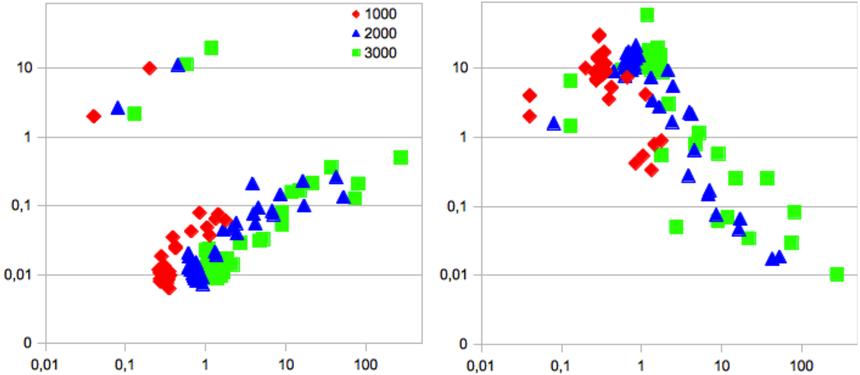
  just use this instead of the original constraints

- Not as strong as PR, but same number of variables and easy to do
- 4-pieces version for $\bar{p}_{min} < 0$ (duplicate u)
- Just properly translating one variable improves the LB:
  blatant violation of the no-free-lunch principle!
- Funny observation: $f(p) = ap^2$, just redefine $p = p_{int}u + q$ and use

$$u^2 = u \qquad\qquad qu = q$$

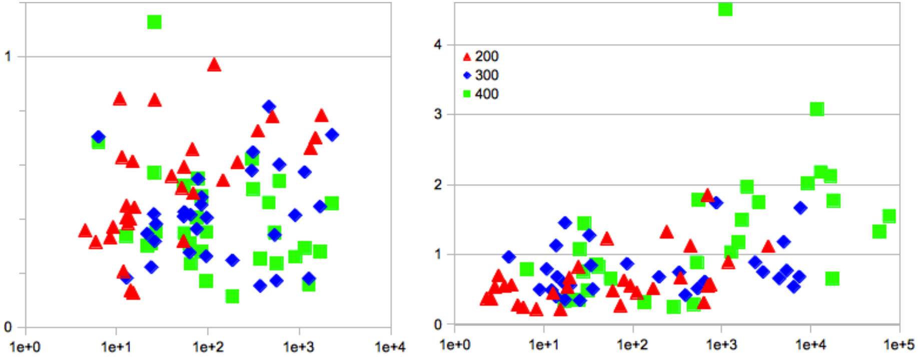
  valid because $u \in \{0,1\}$ and strengthening (RLT?)

Computational results: ND



- left: (AP^2R time) / (P/C time) plotted against AP^2R time
- right = (AP^2R time) / (P^2R time) plotted against AP^2R time

Computational results: MV



- (AP^{2}R time) / (P/C time) plotted against AP^{2}R time
- right = (tight) constraint on max assets (10), not separable

Computational results: UC

		AP²R									
		NCNH		CNH		CH		B&C			
t	h	time	dgap	time	dgap	time	pgap	nodes	time	pgap	gap
10	0	0.31	1.49	1.50	0.29	2.59	0.04	1229	284	0.00	0.01
20	0	0.75	1.25	6.97	0.29	13.99	0.15	4635	9999	0.01	0.17
50	0	3.19	1.19	50.53	0.22	65.12	0.23	1078	9999	0.02	0.23
20	10	0.88	0.58	3.04	0.15	7.91	0.16	16477	1078	0.00	0.01
50	20	2.91	0.58	18.97	0.09	28.33		3780	9999	0.00	0.07
75	35	5.46	0.49	39.18	0.06	45.73		1727	9999	0.03	0.08
		PC									
10	0	0.17	1.48	0.99	0.23	1.25	0.40	365	17	0.00	0.01
20	0	0.49	1.24	3.93	0.25	5.38		15607	4851	0.00	0.02
50	0	2.85	1.16	16.59	0.19	20.63		14286	9986	0.00	0.13
20	10	0.52	0.56	1.92	0.13	3.14	0.51	8107	240	0.00	0.01
50	20	2.05	0.57	6.17	0.07	13.11		66945	6649	0.00	0.02
75	35	4.19	0.48	11.23	0.05	20.22	0.08	57456	9999	0.00	0.02

- No Cuts No Heuristic, Cuts No Heuristic, Cuts & Heuristic, B&C
- Sometimes a better root node solution is found, all the rest is bad

Conclusions

- Errors can be useful, be glad for pesky referees/fellow researchers :-)

[7] Stubbs, Mehrotra "A branch-and-cut method for 0-1 mixed convex programming" *Math. Prog.*, 1999

[8] Khajavirad, Sahinidis "Convex envelopes generated from finitely many compact convex sets" *Math. Prog.* 2013

[9] Luedtke, Namazifar, Linderoth "Some results on the strength of relaxations of multilinear functions" *TR UW-Madison*, 2010

[10] F., Gentile, Lacalandra "Sequential Lagrangian-MILP Approaches for Unit Commitment Problems" *International Journal of Electrical Power and Energy Systems* 2011

Conclusions

- Errors can be useful, be glad for pesky referees/fellow researchers :-)
- Still lots of work to do in PR (nonseparable, . . .),
  several clever people did[7]/does it[8,9]

---

[7] Stubbs, Mehrotra "A branch-and-cut method for 0-1 mixed convex programming" *Math. Prog.*, 1999

[8] Khajavirad, Sahinidis "Convex envelopes generated from finitely many compact convex sets" *Math. Prog.* 2013

[9] Luedtke, Namazifar, Linderoth "Some results on the strength of relaxations of multilinear functions" *TR UW-Madison*, 2010

[10] F., Gentile, Lacalandra "Sequential Lagrangian-MILP Approaches for Unit Commitment Problems" *International Journal of Electrical Power and Energy Systems* 2011

Conclusions

- Errors can be useful, be glad for pesky referees/fellow researchers :-)
- Still lots of work to do in PR (nonseparable, . . .),
  several clever people did[7]/does it[8,9]
- (Energy) problems motivate good methodological research,
  good methodologies help solving real-world problems

[7] Stubbs, Mehrotra "A branch-and-cut method for 0-1 mixed convex programming" *Math. Prog.*, 1999

[8] Khajavirad, Sahinidis "Convex envelopes generated from finitely many compact convex sets" *Math. Prog.* 2013

[9] Luedtke, Namazifar, Linderoth "Some results on the strength of relaxations of multilinear functions" *TR UW-Madison*, 2010

[10] F., Gentile, Lacalandra "Sequential Lagrangian-MILP Approaches for Unit Commitment Problems" *International Journal of Electrical Power and Energy Systems* 2011

Conclusions

- Errors can be useful, be glad for pesky referees/fellow researchers :-)
- Still lots of work to do in PR (nonseparable, ...), several clever people did[7]/does it[8,9]
- (Energy) problems motivate good methodological research, good methodologies help solving real-world problems
- MINLP + decomposition: lots to be invented[10]

[7] Stubbs, Mehrotra "A branch-and-cut method for 0-1 mixed convex programming" *Math. Prog.*, 1999

[8] Khajavirad, Sahinidis "Convex envelopes generated from finitely many compact convex sets" *Math. Prog.* 2013

[9] Luedtke, Namazifar, Linderoth "Some results on the strength of relaxations of multilinear functions" *TR UW-Madison*, 2010

[10] F., Gentile, Lacalandra "Sequential Lagrangian-MILP Approaches for Unit Commitment Problems" *International Journal of Electrical Power and Energy Systems* 2011

Conclusions

- Errors can be useful, be glad for pesky referees/fellow researchers :-)
- Still lots of work to do in PR (nonseparable, ...), several clever people did[7]/does it[8,9]
- (Energy) problems motivate good methodological research, good methodologies help solving real-world problems
- MINLP + decomposition: lots to be invented[10]
- MINLP a funny place to be in: nonlinear + combinatorial = many useful structures

---

[7] Stubbs, Mehrotra "A branch-and-cut method for 0-1 mixed convex programming" *Math. Prog.*, 1999

[8] Khajavirad, Sahinidis "Convex envelopes generated from finitely many compact convex sets" *Math. Prog.* 2013

[9] Luedtke, Namazifar, Linderoth "Some results on the strength of relaxations of multilinear functions" *TR UW-Madison*, 2010

[10] F., Gentile, Lacalandra "Sequential Lagrangian-MILP Approaches for Unit Commitment Problems" *International Journal of Electrical Power and Energy Systems* 2011

Commercial

- PGMO project on “hard” hydro units
- Claudia D’Ambrosio (LIX), Claudio Gentile (IASI-CNR) [, F.]
- Looking for talented MILP expert, one-year post-doc
- Applications welcome, please spread the word

THANKS!