Valid Inequalities for Mixed Integer Second Order Conic Optimization

Aykut Bulut[1]
Ted Ralphs[1]

[1]COR@L Lab, Department of Industrial and Systems Engineering, Lehigh University

INFORMS Annual Conference
November 2014

Outline

Outline

Outline

Outline

MISOCO definition

- We are interested in solving Mixed Integer Second Order Conic Optimization (MISOCO) problems.
- MISOCO is a generalization of Mixed Integer Linear Optimization (MILO).

We formulate MISOCO problem as follows,

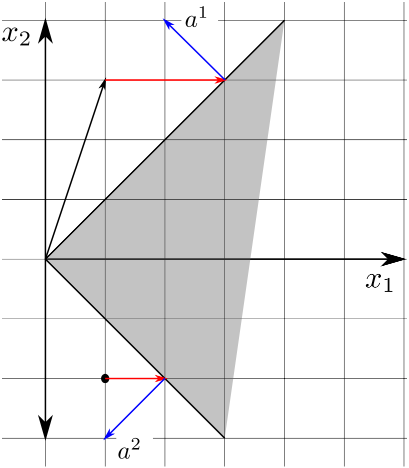
$$\begin{aligned}
\min \quad & c^\top x \\
s.t. \quad & Ax = b \\
& x \in \mathbb{K} \\
& x_i \in \mathbb{R}_+ \quad i \in I \\
& x_j \in \mathbb{Z}_+ \quad j \in J.
\end{aligned} \tag{MISOCO}$$

Outer Approximation Algorithm for SOCO

```
Solve linear relaxation (LO) of the problem.
if LO is infeasible then
    SOCO is infeasible, STOP.
end if
if LO is unbounded then
    while LO is unbounded do
        Determine direction of unboundedness
        if Direction is feasible for all conic constraints then
            SOCO is unbounded, STOP.
        else
            Add cuts using direction of unboundedness.
        end if
        Solve LO.
    end while
end if
Get LO solution
while Solution is not feasbile for conic constraints do
    Add cuts using solution.
    Solve LO.
    if LO is infeasible then
        SOCO is infeasible, STOP.
    end if
    get LO solution.
end while
LO solution is optimal for SOCO, STOP.
```

Separating Infeasible Directions/Solutions

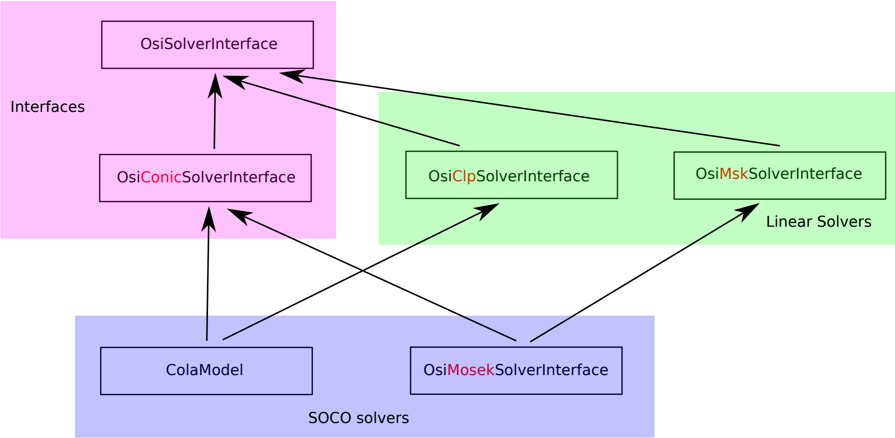
Figure: Separation Example



COLA

- Uses linear outer approximation algorithm.
- All-written in `C++` language.
- Inherits COIN-OR's CLP solver, solves using simplex.
- Has COIN-OR's solver interface and extends it for conic needs.
- Reads problems in extended MPS format, uses COIN Utils for this.
- Takes advantage of simplex method's warm-start capabilities.

Figure: COLA's relationship to COIN-OR projects



COLA Performance Statistics

- NC: Number of Conic constraints the instance has.
- LC: Size of Largest Conic constraint.
- US: number of Unboundedness Supports generated during solution.
- MUS: Maximum number of Unboundedness Supports generated for a cone.
- SS: Separation Supports generated.
- MSS: Maximum number of Separation Supports generated for a cone.
- NLO: Number of linear optimization problems solved.
- CPU: CPU seconds spent during execution of COLA.

COLA Computational Results

Table: COLA statistics on Góez's random instances

instance	NC	LC	US	MUS	SS	MSS	LO	CPU
r12c15k5i10	5	3	0	0	15	4	5	0.01
r14c18k3i9	3	6	4	2	28	13	16	0.01
r17c30k3i12	3	10	12	4	192	66	74	0.07
r17c20k5i15	5	4	0	0	8	3	4	0.0
r22c30k10i20	10	3	5	1	26	5	8	0.02
r22c40k10i20	10	4	16	2	83	13	22	0.03
r23c45k3i21	3	15	13	5	397	140	148	0.25
r27c50k5i25	5	10	16	4	315	69	77	0.11
r32c45k15i30	15	3	8	1	36	4	6	0.0
r32c60k15i30	15	4	29	3	169	15	32	0.02
r52c75k5i35	5	15	7	2	293	71	74	0.15

COLA performance on CBLIB problems 1

Table: COLA statistics on CBLIB 2014 Part 1

instance	NC	LC	US	MUS	SS	MSS	LO	CPU
chainsing-1000-1	2994	3	0	0	14479	10	11	13.01
classical_200_1	1	201	0	0	1055	1055	1056	114.11
classical_50_1	1	51	0	0	328	328	329	1.89
estein4_A	9	3	0	0	36	6	7	0.01
estein4_B	9	3	0	0	44	6	9	0.02
estein4_C	9	3	0	0	60	10	11	0.02
estein4_nr22	9	3	0	0	41	6	7	0.0
estein5_A	18	3	0	0	109	11	14	0.02
estein5_nr21	18	3	0	0	99	9	11	0.03
pp-n1000-d10000	1000	3	0	0	16107	18	19	7.19
pp-n100-d10000	100	3	0	0	1613	18	19	0.12
pp-n10-d10000	10	3	0	0	161	17	18	0.02
robust_50_1	2	52	0	0	260	134	135	0.78
robust_100_1	2	102	0	0	577	297	298	7.64
robust_200_1	2	202	0	0	960	499	500	64.86
shortfall_100_1	2	101	0	0	533	502	503	11.44
shortfall_100_2	2	101	0	0	674	630	631	19.28
shortfall_100_3	2	101	0	0	573	527	528	12.55
shortfall_200_1	2	201	0	0	719	690	691	53.67
shortfall_200_2	2	201	0	0	876	841	842	77.74
shortfall_50_1	2	51	0	0	307	284	285	1.73
shortfall_50_2	2	51	0	0	344	320	321	2.13
shortfall_50_3	2	51	0	0	451	408	409	3.58

COLA performance on CBLIB problems 2

Table: COLA statistics on CBLIB 2014 Part 2

instance	NC	LC	US	MUS	SS	MSS	LO	CPU
sssd-strong-25-8	24	3	0	0	243	13	15	0.07
sssd-strong-30-8	24	3	0	0	260	13	14	0.07
sssd-weak-20-8	24	3	0	0	171	8	9	0.03
sssd-weak-25-8	24	3	0	0	171	8	9	0.04
sssd-weak-30-8	24	3	0	0	165	8	9	0.03
turbine07_aniso	25	3	0	0	53	9	11	0.01
turbine07GF	25	3	0	0	10	4	5	0.0
turbine07_lowb_aniso	25	3	0	0	64	10	12	0.03
turbine07_lowb	27	9	0	0	81	8	9	0.02
turbine07	26	9	0	0	67	12	14	0.02
turbine54GF	119	3	0	0	25	10	11	0.05
turbine54	120	9	0	0	220	11	13	0.05
uflquad-nopsc-10-150	1500	3	0	0	14281	16	20	14.68
uflquad-nopsc-20-150	3000	3	0	0	29063	17	30	74.84
uflquad-nopsc-30-100	3000	3	0	0	29108	23	39	66.91
uflquad-nopsc-30-150	4500	3	0	0	42809	19	39	156.26
uflquad-nopsc-30-200	6000	3	0	0	55650	19	40	332.71
uflquad-nopsc-30-300	9000	3	0	0	83624	16	41	819.0
uflquad-psc-10-150	1500	3	0	0	10837	13	23	14.08
uflquad-psc-20-150	3000	3	0	0	18164	15	37	70.09
uflquad-psc-30-100	3000	3	0	0	16595	22	49	69.07
uflquad-psc-30-150	4500	3	0	0	23675	19	50	128.58
uflquad-psc-30-200	6000	3	0	0	33972	19	50	291.65
uflquad-psc-30-300	9000	3	0	0	54083	19	50	978.33

DietCOLA

- A naive branch and bound framework to solve MISOCO.
- Uses conic OSI to manipulate relaxation problems.
- By default it uses COLA to solve relaxations.
- Mosek can also be used through conic OSI interface.
- Uses COIN-OR's Abstract Library for Parallel Search (ALPS) for search.
- Similar design to Góez's ICLOPS (developed in his PhD work), major difference is it uses conic OSI and COLA.
- Since it uses COLA, it uses simplex.

DietCOLA Performance Statistics

- NNFproc: Number of Nodes Fully processed.
- NNPproc: Number of Nodes Partially processed.
- NNbran: Number of Nodes branched,
- NNprun: Number of Nodes pruned,
- NNleft: Number of Nodes in B&B tree at termination,
- TD: Tree Depth,
- CPU: CPU seconds spent during B&B search.

DietCOLA Random Problems

Table: DietCOLA performance on Góez random instances

instance	NNFproc	NNPProc	NNbran	NNprun	NNleft	TD	CPU
r12c15k5i10	49	0	24	0	0	14	0.01
r12c15k5i15	31	0	15	0	0	12	0.02
r14c18k3i9	2822	5	1413	0	0	28	2.6
r14c18k3i12	1131	24	577	0	0	31	0.6
r14c18k3i15	806	11	408	0	0	31	0.47
r14c18k3i18	288	13	150	0	0	25	0.14
r17c30k3i12	20	18	19	1	0	9	1.14
r17c30k3i15	1989	1571	1780	1	0	20	746.74
r17c20k5i15	61	0	30	0	0	16	0.02
r17c20k5i20	45	0	22	0	0	16	0.01
r22c30k10i20	4459	114	2286	0	0	27	1.52
r22c40k10i20	117	25	71	1	0	11	0.44
r22c40k10i30	146788	51247	99017	0	0	39	2306.78
r23c45k3i21	263	262	262	0	0	19	396.42
r23c45k3i24	4277	4074	4175	0	0	24	16252.78
r27c50k5i25	328	309	318	0	0	26	80.76
r32c45k15i30	2399	508	1453	0	0	29	1.3
r32c45k15i45	213652	6753	110202	0	0	65	190.39
r32c60k15i30	58	50	54	1	0	17	0.46

DietCOLA performance on CBLIB 2014 instances

instance	NNFproc	NNPProc	NNbran	NNprun	NNleft	TD	CPU
estein4_A	31	0	15	0	0	4	0.21
estein4_B	31	0	15	0	0	4	0.14
estein4_C	31	0	15	0	0	4	0.22
estein4_nr22	31	0	15	0	0	4	0.12
estein5_A	971	0	485	0	0	11	35.04
estein5_B	430	1	215	0	0	11	11.38
estein5_C	301	134	217	0	0	11	4.53
estein5_nr1	829	30	429	0	0	11	33.92
estein5_nr21	971	0	485	0	0	11	42.87
pp-n10-d10000	2047	0	1023	0	0	10	3.04
pp-n10-d10	189	79	134	1	0	13	1.57
robust_50_3	161	12	86	0	0	15	3862.64
sssd-strong-20-4	16845	1210	9027	0	0	32	133.4
sssd-strong-25-4	7804	1463	4633	0	0	35	49.3
turbine07_lowb_aniso	1645	354	999	0	0	39	35.09
turbine07_lowb	788	244	517	3	0	31	34.11
turbine54GF	12	0	6	1	0	6	0.04
uflquad-nopsc-10-100	140	133	136	0	0	10	4121.23
uflquad-nopsc-10-150	158	147	152	0	0	10	8688.63
uflquad-psc-10-100	5	2	3	0	0	3	11.34
uflquad-psc-10-150	4	1	2	0	0	2	10.51
uflquad-psc-20-100	10	3	6	0	0	4	88.32
uflquad-psc-20-150	5	2	3	0	0	3	61.74
uflquad-psc-30-100	11	8	9	0	0	6	223.53
uflquad-psc-30-150	3	0	1	0	0	1	17.85
uflquad-psc-30-200	5	2	3	0	0	3	131.13
uflquad-psc-30-300	11	4	7	0	0	4	1124.53

DietCOLA versus Mosek Discrete Solver

Table: Comparison of DietCOLA to Mosek

instance	DietCOLA time	Mosek time
robust_50_3	3862.64	0.93
sched_200_100_orig	1242.41	6.44
uflquad-psc-30-300	1124.53	58.17
uflquad-nopsc-10-100	4121.23	6.48
uflquad-nopsc-10-150	8688.63	10.01
uflquad-nopsc-20-100	NaN	206.44

Reasons of poor performance;

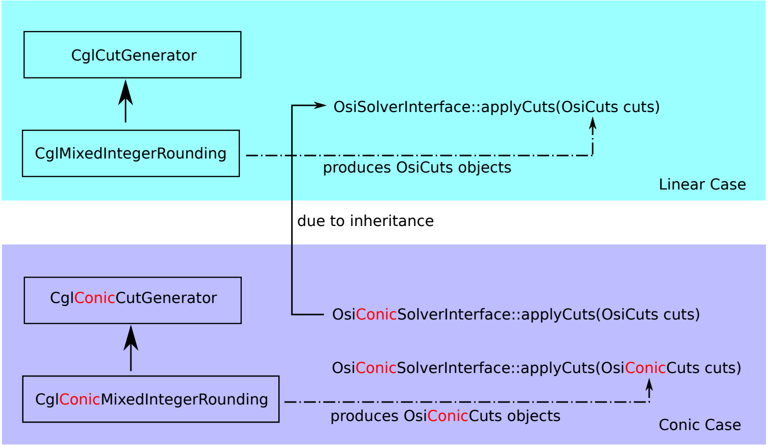
1. lack of cut management for COLA,
2. lack of cut generators for MISOCO feasible set,
3. lack of heuristics,
4. lack of carefully set node selection/searching,
5. lack of parallelization.

Valid Inequalities for MISOCO

We have the following cuts for MISOCO problems,

- Conic mixed-integer rounding (MIR) cuts given by [Atamturk and Narayanan(2010)] for general mixed integer case,
- Conic Gomory cuts given by [Çezik and Iyengar(2005)] for mixed 0–1 problems,
- Convex cuts defined by [Stubbs and Mehrotra(1999)] for mixed 0–1 convex problems,
- **D**isjunctive **c**onic **c**uts (DCC) and **d**isjunctive **cy**lindirical **c**uts (DCyC) defined by [Belotti et al.(2013a)Belotti, Góez, Pólik, Ralphs, and Terlaky] for general mixed integer case,
- Two term disjunctions defined by [Kılınç-Karzan and Yıldız(2014)] for general mixed integer case.

Figure: Conic CGL's relationship to COIN-OR projects



Conic MIR

Assume conic constraints in the following form,

$$\|Ax + Gy - b\| \leq d^\top x + e^\top y - h.$$

Introduce variables $(t_1, t_{2,m+1}) \in \mathbb{R} \times \mathbb{R}^m$,

$$\begin{aligned} t_1 &\leq d^\top x + e^\top y - h, \\ t_{i+1} &\geq |a_i x + g_i y - b_i| \quad i = 1, \ldots, m, \\ t_1 &\geq \|t_{2:m+1}\|. \end{aligned}$$

For a fixed row we define following set,

$$\mathcal{S} := \{(x, y, t) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p \times \mathbb{R} : t \geq |ax + gy - b|\}.$$

Following is an MIR inequality for $\mathcal{S}$ for $\alpha > 0$,

$$\sum_{j=1}^{n} \varphi_{f_\alpha}\left(\frac{a_j}{\alpha}\right) x_j - \varphi_{f_\alpha}\left(\frac{b}{\alpha}\right) \leq \frac{(t + y^+ - y^-)}{|\alpha|}.$$

Conic MIR implementation for COLA case

Extending formulations for a SOC in canonical form does not yield MIR cuts.

$$\|x_{2:n}\| \leq x_1$$

When we extend,

$$\begin{aligned} x_1 &\leq t_1, \\ |x_i| &\leq t_i \qquad i = 2, \ldots, n, \\ t_1 &\geq \|t_{2:m+1}\|. \end{aligned}$$

Then for an integer variable k that is basic and a cone member we can write,

$$\left|\left(B^{-1}b\right)_k - \left(B^{-1}N\right)_k x_N\right| \leq t_k$$

Modify cone constraint as,

$$x_1 \geq \|(x_2, \ldots, t_k, \ldots, x_n)\|.$$

MILP cuts for MISOCO

- COLA approximates conic constraints with supporting hyperplanes around optimal solution.
- Since COLA uses simplex method, COIN-OR's CGL can be used to generate cuts.
- We use CGL to generate cuts in the root node. We generate cuts from
    - `CglKnapsackCover`,
    - `CglSimpleRounding`,
    - `CglGMI`,
    - `CglGomory`,
    - `CglMixedIntegerRounding`,
    - `CglMixedIntegerRounding2`

  classes untill the bound given by relaxation problem does not improve.

MILP cuts performance measures

- NumCut: Number of MILP cuts added in the root node.
- NNwC: Number of nodes with MILP cuts.
- NNwoC: Number of nodes without MILP cuts.
- Gap: Optimality gap closed by the cuts in percentage.

Effectiveness of MILP cuts for MISOCO-1

Table: MILP cut performance on random problem instances

problem name	NumCut	NNwC	NNwoC	Gap (%)
r12c15k5i10	12	47	49	0.1
r12c15k5i15	27	1	31	0.2
r14c18k3i9	10	2822	2822	0.2
r14c18k3i12	87	989	1131	0.1
r14c18k3i15	101	1177	806	0.2
r14c18k3i18	145	669	288	0
r17c30k3i12	17	20	20	0
r17c30k3i15	23	2021	1989	0.4
r17c20k5i15	55	8	61	0.4
r17c20k5i20	74	18	45	0
r22c30k10i20	20	3506	4459	0
r22c40k10i20	12	120	117	0.1
r22c40k10i30	45	79185	146788	0
r23c45k3i21	32	262	263	0
r23c45k3i24	38	5734	4277	0.2
r27c50k5i25	35	332	328	0
r32c45k15i30	24	12	2399	0
r32c45k15i45	119	854	213652	0.6
r32c60k15i30	23	50	58	0

Table: MILP cut performance on CBLIB problem instances

problem name	NumCut	NNwC	NNwoC	Gap (%)
estein4_A	112	28	31	26
estein4_B	94	31	31	27
estein4_C	104	28	31	41
estein4_nr22	202	31	31	47
estein5_A	28	927	971	0
estein5_B	30	394	430	0
estein5_C	22	434	301	0
estein5_nr1	26	681	829	0
estein5_nr21	30	971	971	0
robust_50_3	12	844	161	0
turbine07_lowb_aniso	45	2234	1645	0
turbine07_lowb	36	656	788	0
turbine54GF	11	2	12	10
uflquad-nopsc-10-100	701	138	140	0.7
uflquad-nopsc-10-150	339	158	158	0.2
uflquad-nopsc-20-100	2480	-	-	2.7
uflquad-nopsc-20-150	2391	-	-	1.3
uflquad-nopsc-30-100	9222	-	-	19
uflquad-nopsc-30-150	4212	-	-	4
uflquad-nopsc-30-200	3446	-	-	2
uflquad-nopsc-30-300	3031	-	-	1
uflquad-psc-10-100	3	5	5	0
uflquad-psc-10-150	5	4	4	0
uflquad-psc-20-100	9	10	10	0
uflquad-psc-20-150	5	5	5	0
uflquad-psc-30-100	5	11	11	0
uflquad-psc-30-150	1	3	3	0
uflquad-psc-30-200	4	5	5	0
uflquad-psc-30-300	9	11	11	0

Contributions

- Outer approximation algorithm performance results on continuous problems.
- Testing outer approximation algorithm on discrete problems in a branch and bound framework.
- Implementation details for conic MIR.
- Software tools conic OSI, COLA/DietCOLA.

Future Work

- Comparing performance of outer approximation method to IPM in branch and bound framework.
- Comparing performance of different cutting procedures.
- Implementing Conic CGL.
- Exploring/Implementing pre-processing/heuristic procedures for MISOCO that may improve the performance of software tools provided.
- Blending all these different procedures to come up with a reliable setting that performs good most of the time.
- Improving COLA with cut management, improve numerics.

References

Alper Atamturk and Vishnu Narayanan.
Conic mixed-integer rounding cuts.
Mathematical Programming, 122(1):1–20, 2010.
ISSN 0025-5610.
doi: 10.1007/s10107-008-0239-4.
URL http://dx.doi.org/10.1007/s10107-008-0239-4.

Alper Atamtürk, Gemma Berenguer, and Zuo-Jun Shen.
A conic integer programming approach to stochastic joint location-inventory problems.
Operations research, 60(2):366–381, 2012.

P Belotti, J C Góez, I Pólik, T K Ralphs, and T Terlaky.
On families of quadratic surfaces having fixed intersection with two hyperplanes.
161, 2013a.
URL http://coral.ie.lehigh.edu/~ted/files/papers/Quadratic11.pdf.

Pietro Belotti, Julio C. Góez, Imre Pólik, Ted K. Ralphs, and Tamás Terlaky.
On families of quadratic surfaces having fixed intersections with two hyperplanes.
Discrete Applied Mathematics, 161(16–17):2778–2793, 2013b.
ISSN 0166-218X.

Dimitris Bertsimas and Romy Shioda.
Algorithm for cardinality-constrained quadratic optimization.
Computational Optimization and Applications, 43(1):1–22, 2009.

M.T. Çezik and G. Iyengar.
Cuts for mixed 0-1 conic programming.
Mathematical Programming, 104(1):179–202, 2005.
ISSN 0025-5610.
doi: 10.1007/s10107-005-0578-3.
URL http://dx.doi.org/10.1007/s10107-005-0578-3.

End of presentation

This is end of presentation!

Thank you for listening!