Computational Approaches to Mixed Integer Second Order Cone Optimization (MISOCO)

Aykut Bulut[1]
Ted Ralphs[1]

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

INFORMS Annual Meeting,
3 November 2015

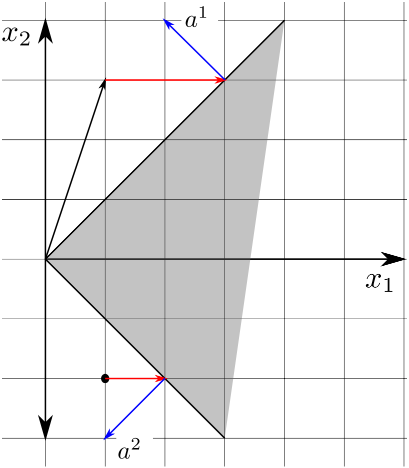
Outline

MISOCO definition

- We are interested in solving Mixed Integer Second Order Conic Optimization (MISOCO) problems.
- MISOCO is a generalization of Mixed Integer Linear Optimization (MILP).
- MISOCO can be formulated as follows,

$$
\begin{aligned}
\min \quad & c^\top x & & \\
s.t. \quad & Ax = b & & \\
& x \in \mathbb{L}^1 \times \cdots \times \mathbb{L}^k & & \\
& x_i \in \mathbb{R}_+ & & i \in I \\
& x_j \in \mathbb{Z}_+ & & j \in J.
\end{aligned}
\tag{MISOCO}
$$

Figure: Separation Example



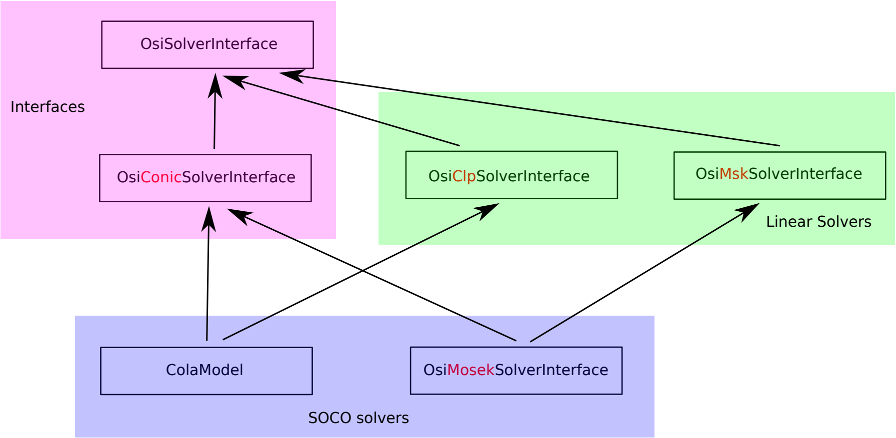
Outer Approximation Algorithm for SOCO

Solve linear relaxation (LP) of the problem.
if LP is infeasible **then**
    SOCO is infeasible, STOP.
end if
if LP is unbounded **then**
    while LP 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 LP.
    end while
end if
Get LP solution
while Solution is not feasbile for conic constraints **do**
    Add cuts using solution.
    Solve LP.
    if LP is infeasible **then**
        SOCO is infeasible, STOP.
    end if
    get LP solution.
end while
LP solution is optimal for SOCO, STOP.

COLA

- Implements outer approximation algorithm.
- All-written in `C++` language.
- Cola uses CLP to solve LP relaxations.
- Reads problems in Mosek's extended MPS format, uses COIN Utils for this.
- Implements *conic* OSI, which is an extension of OSI.
- *Conic* OSI can be used to build models.
- Takes advantage of simplex method's warm-start capabilities.

Figure: COLA's relationship to COIN-OR projects



COLA performance on CBLIB problems 1

Table: COLA statistics on CBLIB 2014

instance	Num C.	Larg. C.	Num Cuts	LP	CPU
chainsing-1000-1	2994	3	14479	11	13.01
classical_50_1	1	51	328	329	1.89
classical_200_1	1	201	1055	1056	114.11
estein4_A	9	3	36	7	0.01
robust_50_1	2	52	260	135	0.78
robust_200_1	2	202	960	500	64.86
shortfall_50_1	2	51	307	285	1.73
shortfall_100_1	2	101	533	503	11.44
shortfall_200_1	2	201	719	691	53.67
sssd-weak-30-8	24	3	165	9	0.03
turbine07	26	9	67	14	0.02
uflquad-nopsc-10-150	1500	3	14281	20	14.68
uflquad-nopsc-30-300	9000	3	83624	41	819.0

DisCO solver

- A branch and bound framework to solve MISOCO.
- Uses *conic* OSI to manipulate relaxation problems.
- By default it uses COLA to solve relaxations.
- Cplex, Mosek and IPOPT can also be used through conic OSI interface.
- Extends COIN-OR's High-Performance Parallel Search (CHiPPS) framework for conic problems.
- Similar design to Góez's ICLOPS (developed in his PhD work), major difference is it uses conic OSI.
- Simplex is used when COLA is chosen as solver.
- DisCO can use COIN-OR's CGL when COLA is chosen as solver.

Disjunctive Cuts by Belotti et al.

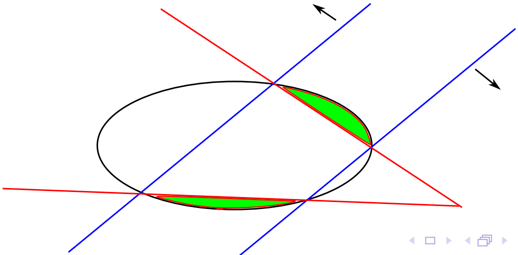
SOCO feasible region is given by

$$Ax = b$$
$$x \in \mathbb{L}^1 \times \cdots \times \mathbb{L}^k.$$

We convert the problem into the following form using $x = x^0 + Hw$,

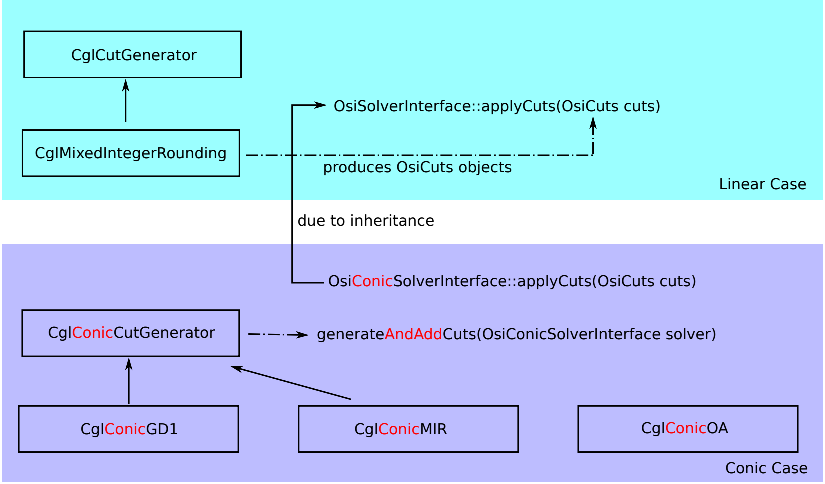
$$\begin{aligned} w^\top Q^i w + 2q^{i\top} w + \rho^i \leq 0 \quad & i \in \{1, \ldots, k\} \\ a^{i\top} w \geq \alpha^i \quad & i \in \{1, \ldots, k\}, \end{aligned}$$

where H is the null space basis of A and x^0 is any point such that $Ax^0 = b$.



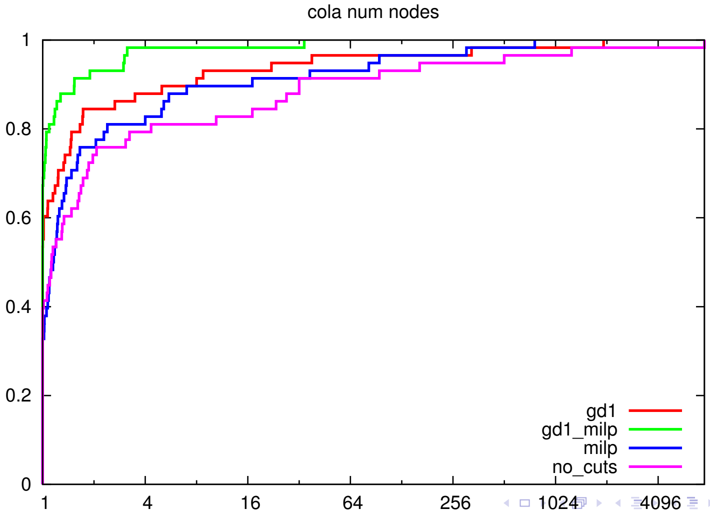
Implementing Conic Cuts

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



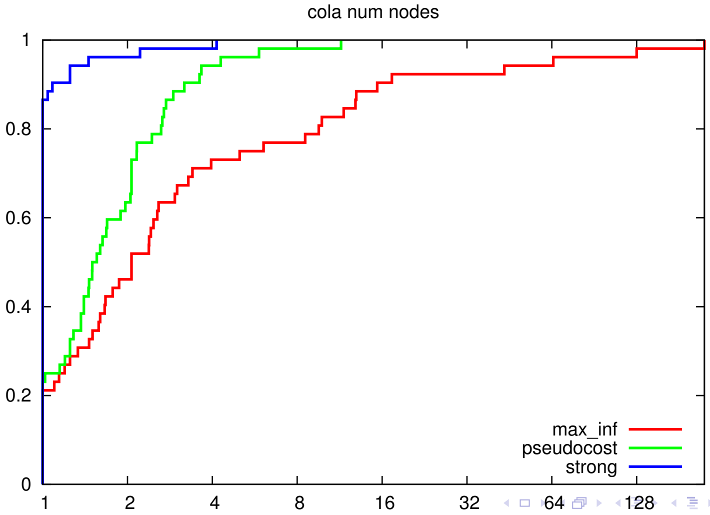
COLA with various cut strategies

Figure: COLA cut strategies, number of nodes



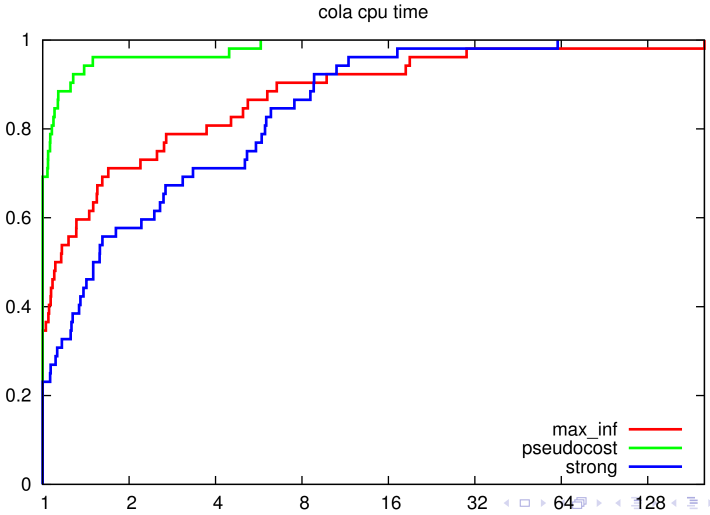
DisCO Branching Experiments with COLA–Nodes

Figure: COLA branching strategy number of nodes



DisCO Branching Experiments with COLA–CPU time

Figure: COLA branching strategy CPU time



DisCO Branching Experiments with Mosek–Nodes

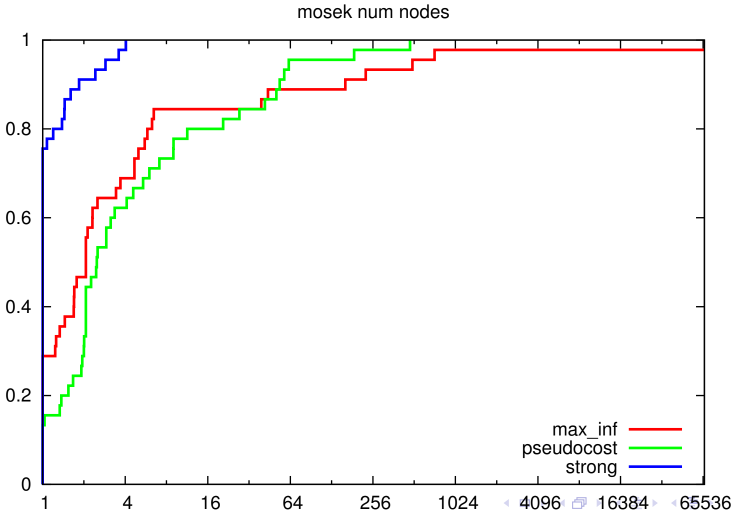
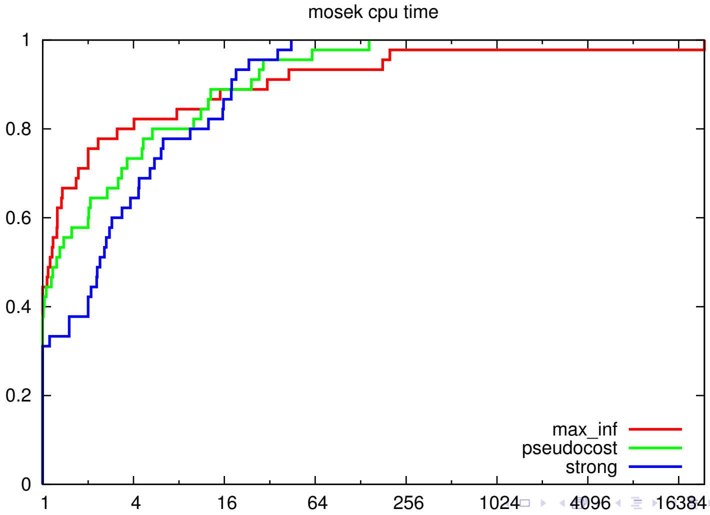


Figure: Mosek branching strategy, number of nodes

DisCO Branching Experiments with Mosek–CPU time

Figure: Mosek branching strategy, CPU time



DisCO, a Hybrid Approach

Algorithm 1 Hybrid Algorithm

```
Solve SOCO using IPM.
Approximate around IPM solution. Add approximating LP to node list.
while Node list is not empty do
    Pick a node, solve approximating LP.
    if LP solution is feasible for integrality and conic constraints then
        Store solution. Update lower bound.
    else
        if Fractional solution exists then
            Branch, add new nodes to list.
        else
            Call IPM solver.
            if Integer feasible then
                Store solution. Update lower bound.
            else
                Approximate around IPM solution.
                Branch, add new nodes to list.
            end if
        end if
    end if
end while
```

Contributions

- Outer approximation algorithm performance results on continuous problems.
- Testing outer approximation algorithm on discrete problems in a branch and bound framework.
- Software tools conic OSI, interface for Mosek, COLA, DisCO.
- Comparing performance of outer approximation method to IPM in branch and bound framework.
- Comparison of different branching strategies for MISOCO.
- Performance of disjunctive cuts given by Belotti et. al.
- Performance of MILP cuts when outer approximation is used.

Clone, Try, Contribute

`https://github.com/aykutbulut`
`https://github.com/coin-or`

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.

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, 2013.
ISSN 0166-218X.

Henrik A. Friberg.
Cblib 2014: A benchmark library for conic mixed-integer and continuous optimization.
Optimization Online, 2014.
URL http://www.optimization-online.org/DB_FILE/2014/03/4301.pdf.

Julio C. Góez.
Mixed Integer Second Order Cone Optimization Disjunctive Conic Cuts: Theory and experiments.
PhD thesis, Lehigh University, Bethlehem, PA, September 2013.

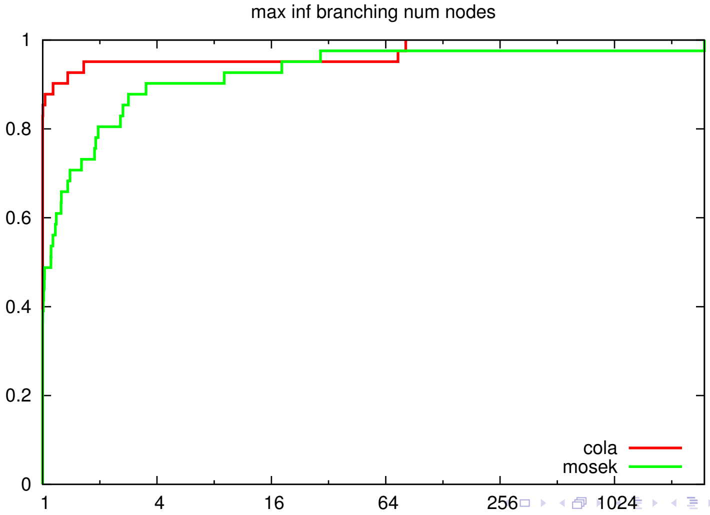
End of presentation

This is end of presentation!

Thank you for listening!

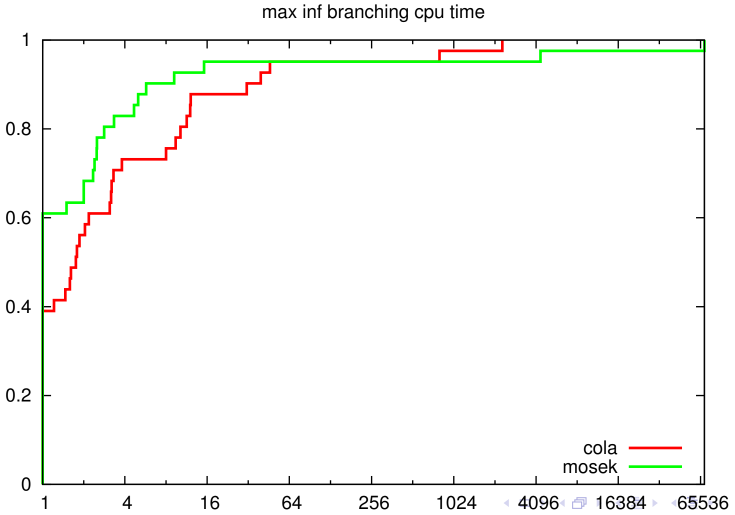
Maximum Infeasibility Branching–Nodes

Figure: COLA vs Mosek with maximum infeasibility, number of nodes



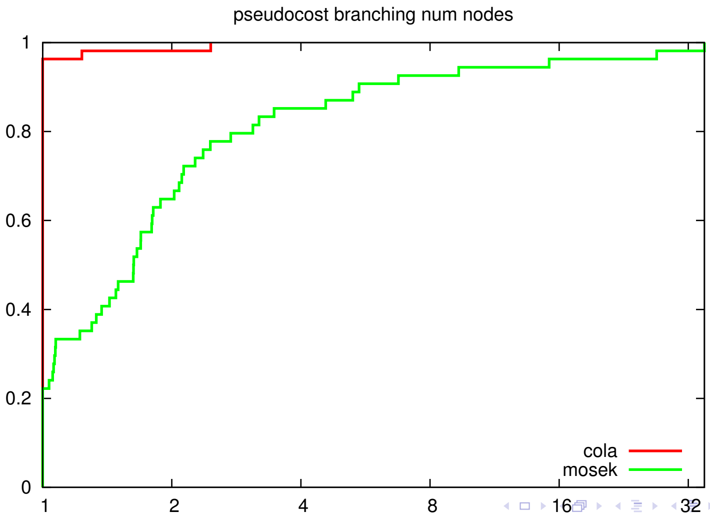
Maximum Infeasibility Branching–CPU time

Figure: COLA vs Mosek with maximum infeasibility, CPU time



Pseudocost Branching

Figure: COLA vs Mosek with pseudocost, number of nodes



Strong Branching–Nodes

Figure: COLA vs Mosek with strong branching, number of nodes

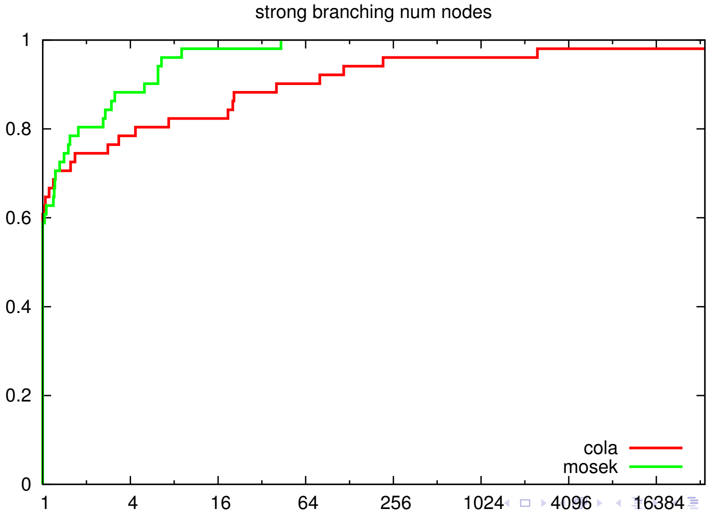
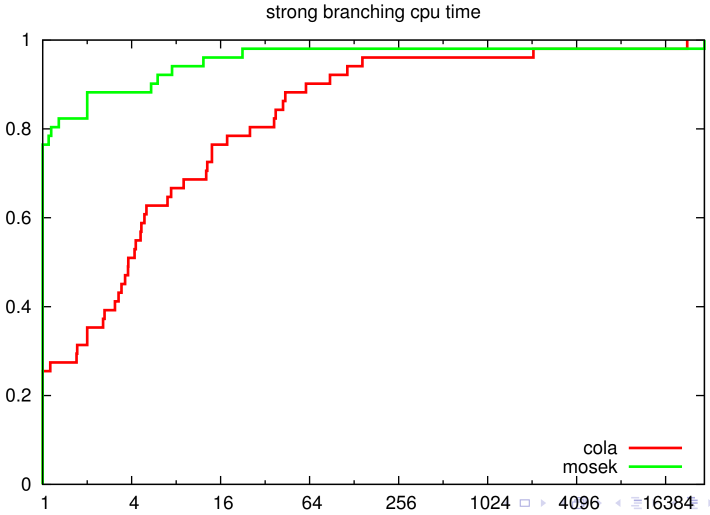


Figure: COLA vs Mosek with strong branching, CPU time



COLA–Pseudocost vs Mosek–Max Inf, Nodes

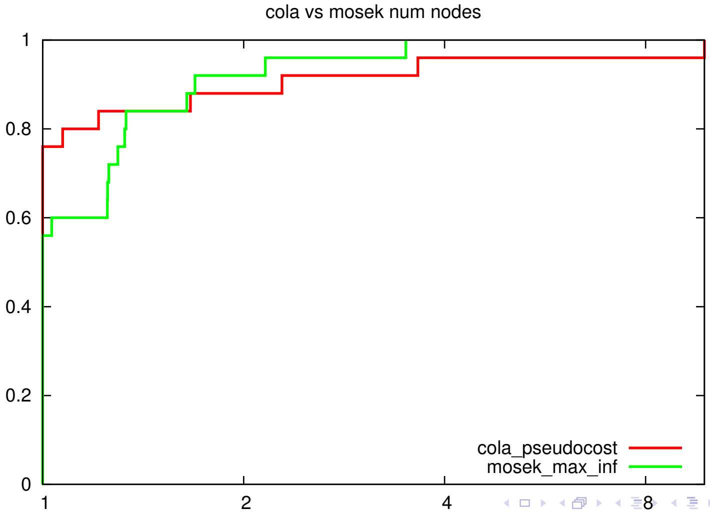


Figure: COLA pseudocost vs Mosek maximum infeasibility, number of nodes

COLA–Pseudocost vs Mosek–Max Inf, CPU time

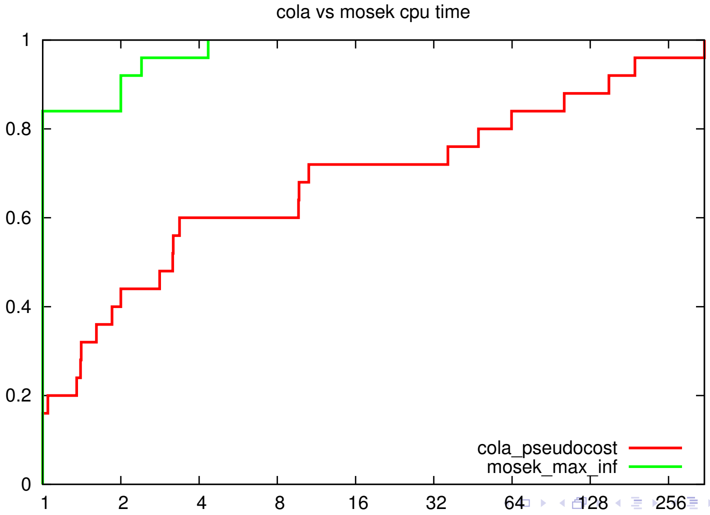


Figure: COLA pseudocost vs Mosek maximum infeasibility, CPU time