Continuous cutting plane algorithms

Andrea Lodi
Cornell Tech
andrea.lodi@cornell.edu

(joint work with Didier Chételat)

Trends in Computational Discrete Optimization
April 28, 2023 @ ICERM

POLYTECHNIQUE MONTRÉAL

CORNELL TECH | HOME OF THE JACOBS TECHNION-CORNELL INSTITUTE

DATA SCIENCE FOR REAL-TIME DECISION-MAKING

Background

MILPs

As usual, we want to solve a mixed-integer linear program

$$\begin{aligned} \min \quad & c^t x + d^t z \\ \text{s.t.} \quad & Ax + Dz = b \\ & x, z \geq 0, \quad x \in \mathbb{Z}^k \end{aligned}$$

To solve these problems, it is customary to obtain dual bounds. The simplest way is through the LP relaxation

$$\begin{aligned} \min \quad & c^t x + d^t z \\ \text{s.t.} \quad & Ax + Dz = b \\ & x, z \geq 0 \end{aligned}$$

Valid inequalities

Inequality $\alpha^t x + \delta^t z \geq \beta$ valid for all MILP-feasible points

$$\{x, z | Ax + Dz = b, x, z \geq 0, x \in \mathbb{Z}^k\}$$

"Valid inequality"

Add it: same MILP, but possibly tighter LP (= better dual bound)

$$\begin{aligned} \min \quad & c^t x + d^t z \\ \text{s.t.} \quad & Ax + Dz = b \\ & \alpha^t x + \delta^t z \geq \beta \\ & x, z \geq 0, \quad x \in \mathbb{Z}^k \end{aligned} \quad \longrightarrow \quad \begin{aligned} \min \quad & c^t x + d^t z \\ \text{s.t.} \quad & Ax + Dz = b \\ & \alpha^t x + \delta^t z \geq \beta \\ & x, z \geq 0 \end{aligned}$$

Cutting planes

Cutting plane: valid inequality + some LP-feasible point (x*,z*) ($Ax^* + Dz^* = b, x^*, z^* \geq 0$) is cut off by the hyperplane if,

$$\alpha^t x^* + \delta^t z^* < \beta$$

There are algorithms that, given a MILP, can produce cutting planes in polynomial time.

GMI cutting plane algorithm

The prototype is Gomory's mixed-integer cutting plane algorithm.

1) Solve the LP relaxation of the problem, yielding a basis matrix B.
2) The GMI cuts are

$$\mathrm{tri}_{B^{-1}b}(B^{-1}A)x + \mathrm{abs}_{B^{-1}b}(B^{-1}D)z \geq 1$$

$$\left(\begin{array}{l} \mathrm{tri}_y(x) = \min\left(\frac{\{x\}}{\{y\}}, \frac{1-\{x\}}{1-\{y\}}\right) \\ \mathrm{abs}_y(x) = \max\left(\frac{x}{\{y\}}, \frac{-x}{1-\{y\}}\right) \end{array} \right)$$

Add them to the problem, yielding a tighter MILP.

$$\begin{array}{ll} \min & c^t x + d^t z \\ \text{s.t.} & Ax + Dz = b \\ & \mathrm{tri}_{B^{-1}b}(B^{-1}A)x + \mathrm{abs}_{B^{-1}b}(B^{-1}D)z \geq 1 \\ & x, z \geq 0, \quad x \in \mathbb{Z}^k \end{array}$$

=

$$\begin{array}{ll} \min & c^t x + {d_1}^t z_1 \\ \text{s.t.} & A_1 x + D_1 z_1 = b_1 \\ & x, z_1 \geq 0, \quad x \in \mathbb{Z}^k \end{array}$$

GMI cutting plane algorithm

3) We can compute the LP relaxation of this extended MILP, yielding its basis matrix B_1.

4) We can compute its GMI cuts

$$\mathrm{tri}_{B_1^{-1}b_1}(B_1^{-1}A_1)x + \mathrm{abs}_{B_1^{-1}b_1}(B_1^{-1}D_1)z_1 \geq 1$$

Again, these can be added to the MILP yielding a tighter MILP

$$\begin{aligned} \min \quad & c^t x + d_2{}^t z_2 \\ \text{s.t.} \quad & A_2 x + D_2 z_2 = b_2 \\ & x, z_2 \geq 0, \quad x \in \mathbb{Z}^k \end{aligned}$$

Etc.

GMI cutting plane algorithm

Note that in terms of A, D and **z**, rank-2 cuts are

$$\mathrm{tri}_{B_1^{-1}b_1}(B_1^{-1}A_1)x + \mathrm{abs}_{B_1^{-1}b_1}(B_1^{-1}D_1)z_1 \geq 1$$

$$\Big[\mathrm{tri}_{(B_1^{-1})_1{}^t b+(B_1^{-1})_2{}^t 1}\Big((B_1^{-1})_1{}^t A + (B_1^{-1})_2{}^t \mathrm{tri}_{B^{-1}b}(B^{-1}A)\Big) + \mathrm{abs}_{(B_1^{-1})_1{}^t b+(B_1^{-1})_2{}^t 1}(-(B_1^{-1})_2)^t \mathrm{tri}_{B^{-1}b}(B^{-1}A)\Big]x$$

$$+\Big[\mathrm{abs}_{(B_1^{-1})_1{}^t b+(B_1^{-1})_2{}^t 1}\Big((B_1^{-1})_1{}^t D + (B_1^{-1})_2{}^t \mathrm{abs}_{B^{-1}b}(B^{-1}D)\Big) + \mathrm{abs}_{(B_1^{-1})_1{}^t b+(B_1^{-1})_2{}^t 1}(-(B_1^{-1})_2)^t \mathrm{abs}_{B^{-1}b}(B^{-1}D)\Big]z$$

$$\geq \Big[1 + \mathrm{abs}_{(B_1^{-1})_1{}^t b+(B_1^{-1})_2{}^t 1}(-(B_1^{-1})_2)^t 1)\Big]$$

Continuous cuts optimization

GMI inequality “family”

We saw that the GMI rank-1 cuts were

$$\mathrm{tri}_{B^{-1}b}(B^{-1}A)x + \mathrm{abs}_{B^{-1}b}(B^{-1}D)z \geq 1$$

But actually, it is well known that for *any* v, W,

$$\mathrm{tri}_{v}(WA)x + \mathrm{abs}_{v}(WD)z \geq \mathrm{tri}_{v}(Wb)$$

are valid.

You get the classical GMI cuts as a special case by taking

$$W = B^{-1}, v = B^{-1}b$$

GMI inequality "family"

Similarly, for any v_1, W_1, v_2, $W_2=[W_{21}, W_{22}]$, the rank-2 GMI inequalities

$$\left[\mathrm{tri}_{v_2}\left({W_{21}}^t A + {W_{22}}^t \mathrm{tri}_{v_1}(W_1 A)\right) + \mathrm{abs}_{v_2}(-W_{22})^t \mathrm{tri}_{v_1}(W_1 A)\right] x$$

$$+\left[\mathrm{abs}_{v_2}\left({W_{21}}^t D + {W_{22}}^t \mathrm{abs}_{v_1}(W_1 D)\right) + \mathrm{abs}_{v_2}(-W_{22})^t \mathrm{abs}_{v_1}(W_1 D)\right] z$$

$$\geq \left[\mathrm{tri}_{v_2}\left({W_{21}}^t b + {W_{22}}^t \mathrm{tri}_{v_1}(W_1 b)\right) + \mathrm{abs}_{v_2}(-W_{22})^t \mathrm{tri}_{v_1}(W_1 b)\right]$$

are valid inequalities for the MILP, and we recover the classical GMI cuts by

$$v_1 = B^{-1} b,\; W_1 = B^{-1},\; v_2 = B_1^{-1} b_1,\; W_2 = B_1^{-1}$$

GMI inequality “family”

In general, we have families of inequalities for every “rank”, valid for the MILP by construction, parametrized by continuous parameters $\theta = (v_1, W_1, v_2, W_2, \ldots)$.

Question: what if, instead of choosing the parameters that the GMI separation algorithm tells us to take ($\theta = (B^{-1}b, B^{-1}, B_1^{-1}b_1, B_1^{-1}, \ldots)$), we try to find other, potentially better parameters θ?

Of course, it is not a new idea but we will try to do it differently.

Optimization

Criterion: try to find the parameters θ such that, when the inequalities are added to the MILP, the LP dual bound is as high as possible

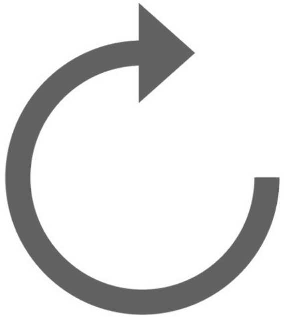
$$\max_{\theta} \; \text{LP-Val}_{\theta}(A, b, c)$$

Challenge: really nasty nonlinear continuous optimization problem.

Algorithm

Ad hoc two-step approach:

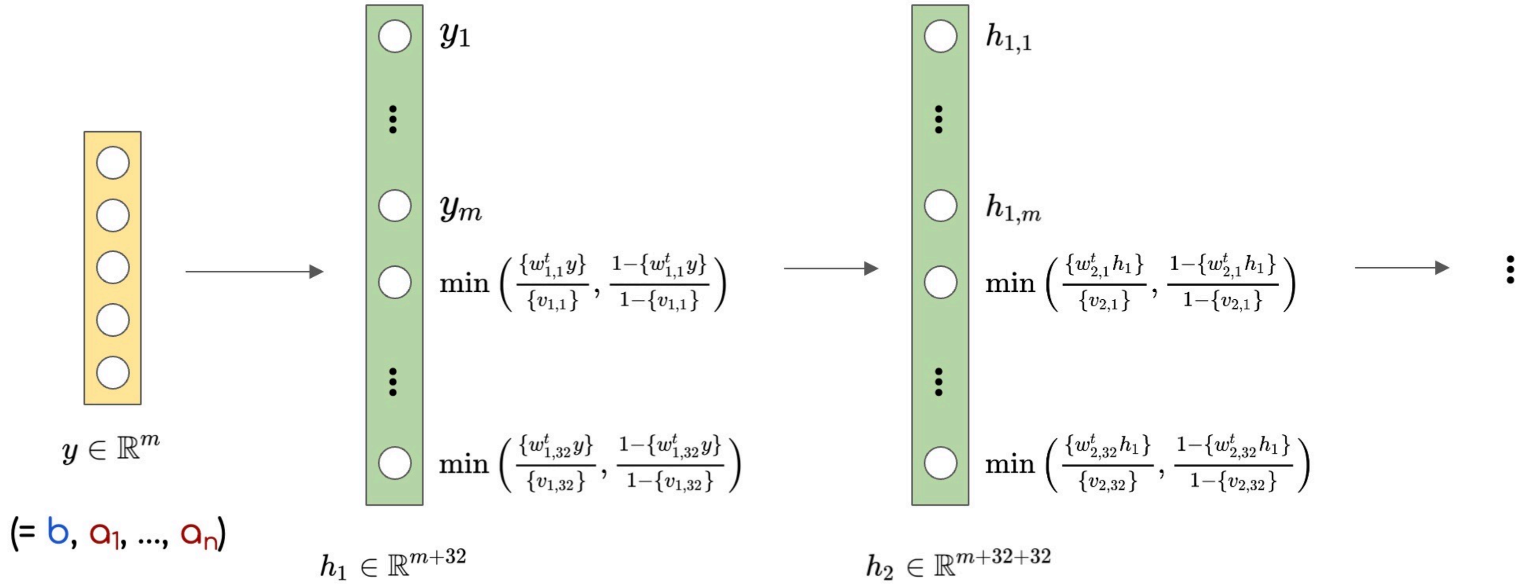
1. Solve the LP relaxation $LP_\theta(A, b, c)$.
2. Take a gradient step to make the GMI family inequalities $\Gamma_\theta x + \Delta_\theta z \geq \gamma_\theta$ cut off the LP solution (x^*, z^*):



$$\theta' \leftarrow \theta - \alpha \sum_i \nabla_\theta [\Gamma_\theta x^* + \Delta_\theta z^* - \gamma_\theta]_i$$

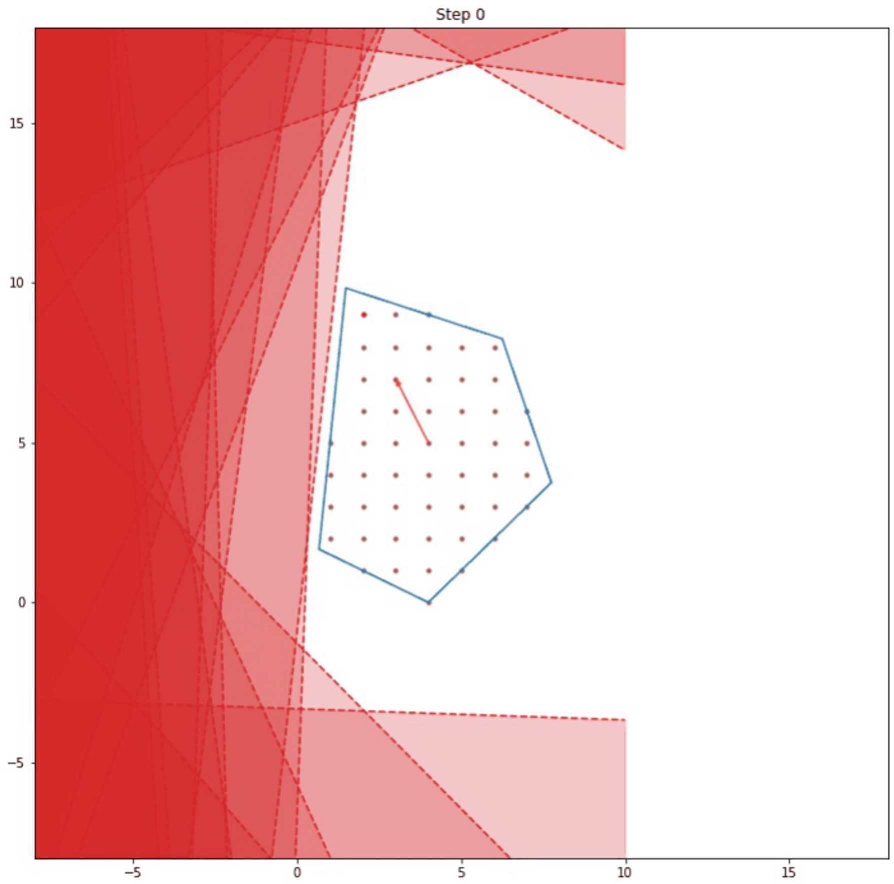
(α = small step size, e.g., 1e-3)

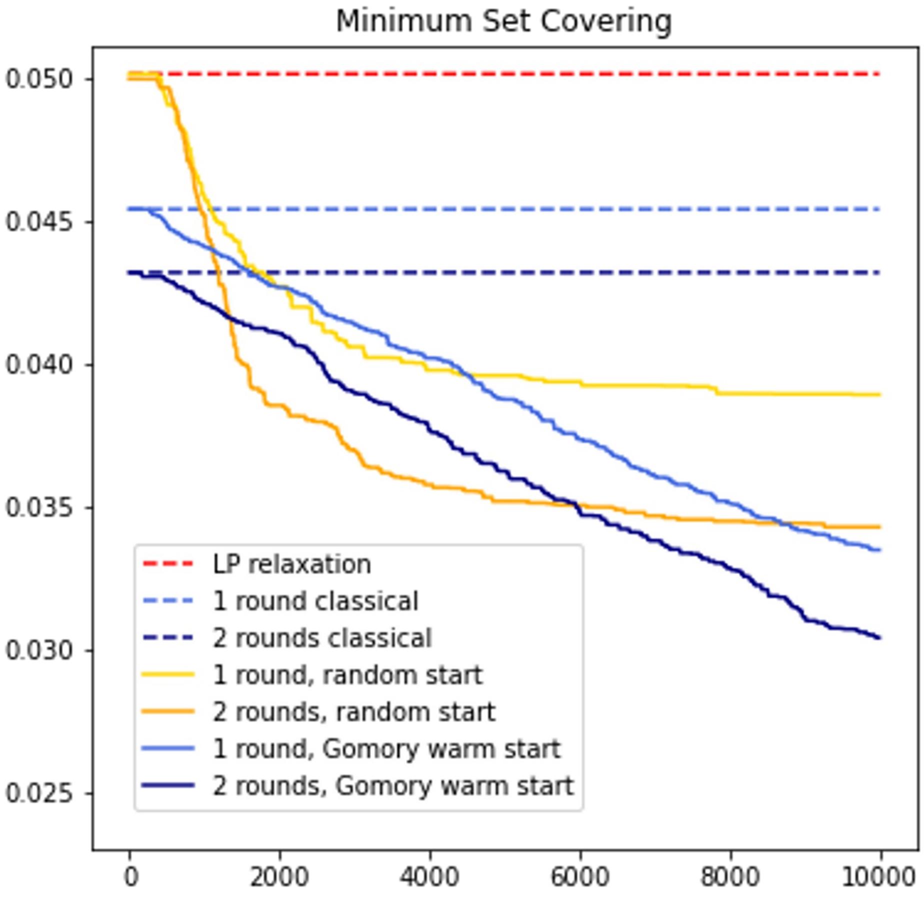
A computational environment for implementing the previous algorithm is that of a Neural Network



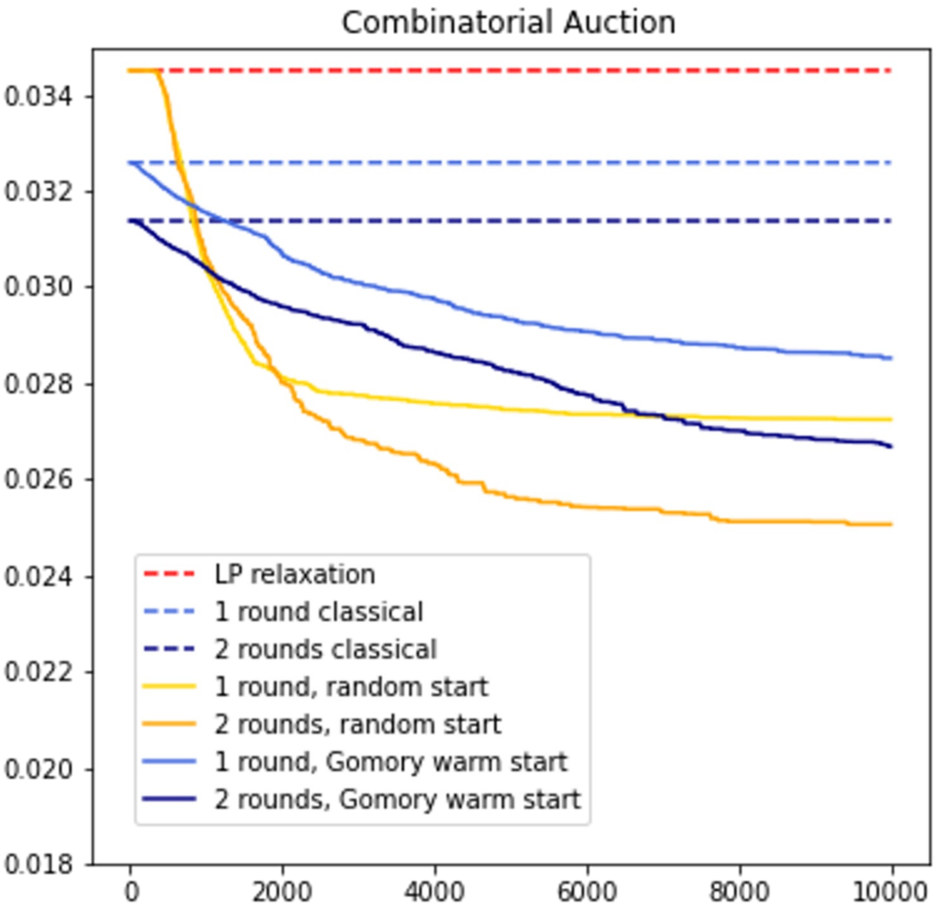
Experiments

32 rank-1 GMI inequalities, randomly initialized

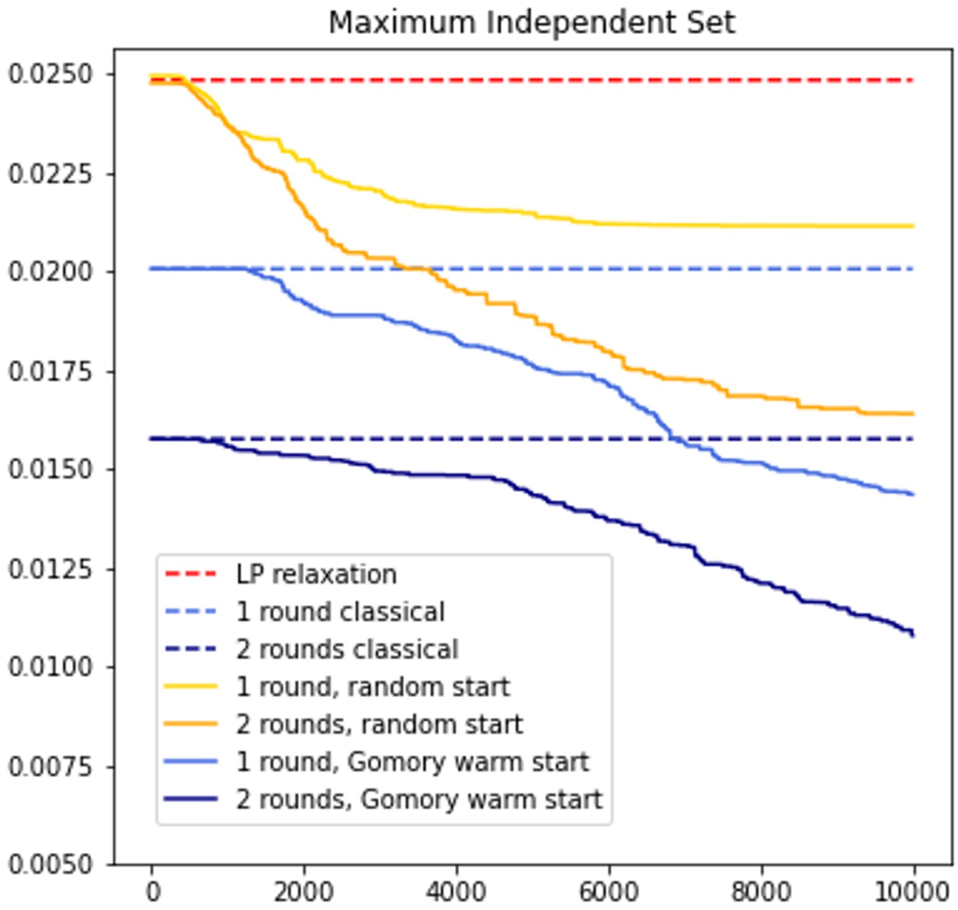




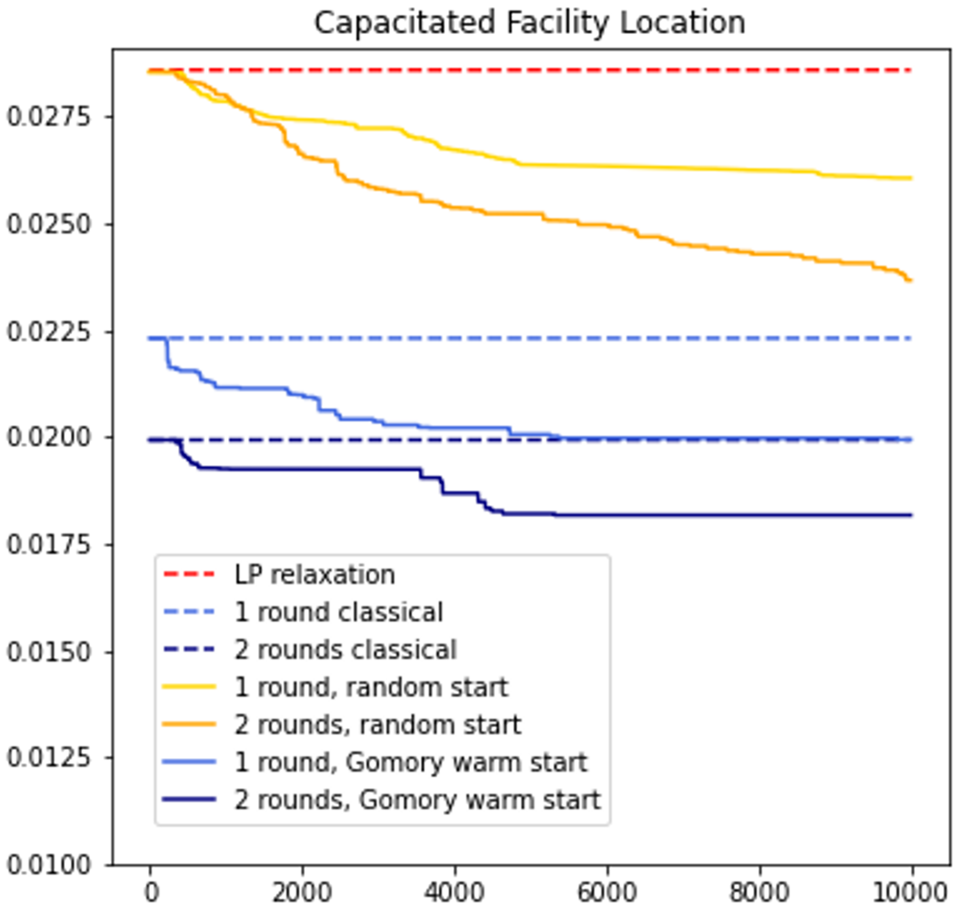
1000 variables
500 constraints



100 items
500 bids


Maximum Independent Set
0.0250
0.0225
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0
2000
4000
6000
8000
10000
LP relaxation
1 round classical
2 rounds classical
1 round, random start
2 rounds, random start
1 round, Gomory warm start
2 rounds, Gomory warm start

300 nodes



20 customers
10 facilities
Discrete assignment

Discussion

Cuts are added in rounds, **but** from one iteration to the other they are removed from the LP relaxation

- the bound is not necessarily monotone, but
- the size of the LP stays small

In some sense, 32 cuts per round are iteratively improved and made more robust (“distilled”), i.e., they are able to cut off simultaneously a cloud of solutions of the LP relaxation.

Differently from the classical cutting plane methods, it seems that the “memory” of the previous LPs remains (in some form) in the NN.

Subadditive neural networks

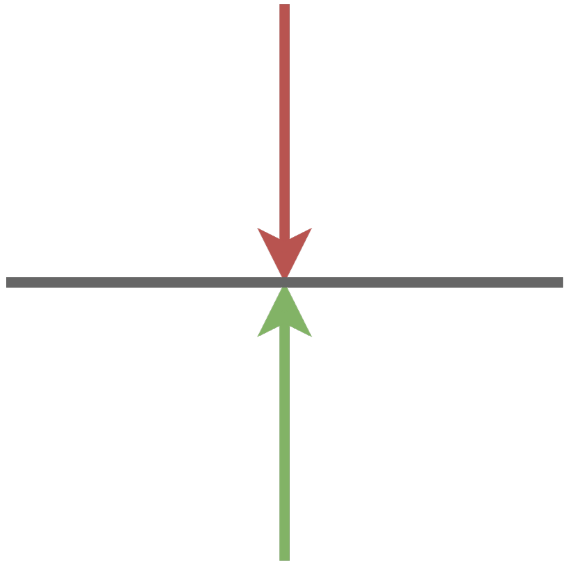
LP duality

It is well known that every LP

$$\begin{aligned} \min \quad & c^t x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

has an associated equivalent dual LP

$$\begin{aligned} \max \quad & w^t b \\ \text{s.t.} \quad & w^t A \leq c \end{aligned}$$



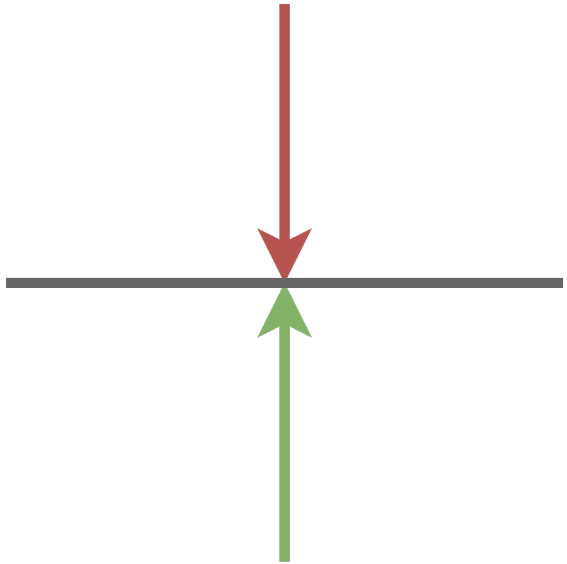
ILP duality

Every ILP

$$\min_{x \in \mathbb{R}^n} \quad c^t x$$
$$\text{s.t.} \quad Ax = b$$
$$x \geq 0, \quad x \in \mathbb{Z}^n$$

has an an associated equivalent infinite-dimensional, “continuous” problem

$$\max_{f:\mathbb{R}^m \mapsto \mathbb{R}} \quad f(b)$$
$$\text{s.t.} \quad f(A) \leq c$$
$$f \text{ is subadditive}$$



Subadditive functions

A subadditive function is a function such that

$$f(x+y) \leq f(x) + f(y)$$

Not necessarily differentiable, or even continuous, but to any subadditive function, we can associate its “upper directional derivative at zero” (UDDZ)

$$\bar{f}\,(y) = \limsup_{h \to 0+} \frac{f(hy)}{h}$$

Example

Recall "weighted tri" and "weighted abs" functions introduced to talk about GMI cuts, namely

$$\mathrm{tri}_v(x) = \min\left(\frac{\{x\}}{\{v\}}, \frac{1-\{x\}}{1-\{v\}}\right), \qquad \mathrm{abs}_v(x) = \max\left(\frac{x}{\{v\}}, \frac{-x}{1-\{v\}}\right)$$

It turns out that for any v and any W,

$$\mathrm{tri}_v(W^t y)$$

is subadditive in y, and that $\mathrm{abs}_v(Wy)$ is its UDDZ, i.e., $\mathrm{abs}_v(W^t y) = \overline{\mathrm{tri}_v(W^t y)}$

Connection

There is an interesting connection between continuous cut optimization and the subadditive dual.

Let us denote by A_θ, b_θ, c_θ the extended matrices obtained after adding K rounds of GMI valid inequalities, parametrized by $\theta = (v_1, W_1, \ldots, v_K, W_K)$.

We can rewrite our continuous cuts optimization problem as

$$\max_\theta \text{ LP-Val}_\theta(A, b, c) \equiv \max_\theta \begin{bmatrix} \min\limits_x & {c_\theta}^t x \\ \text{s.t.} & A_\theta x = b_\theta \\ & x \geq 0 \end{bmatrix} = \max_\theta \begin{bmatrix} \max\limits_w & w^t b_\theta \\ \text{s.t.} & w^t A_\theta \leq c_\theta \end{bmatrix}$$

by using LP duality, which, in turn, can be expanded as

Connection

$$= \begin{cases} \max\limits_{\theta, w} & w^t \left[b, \text{tri}_{v_1}(W_1 b), \text{tri}_{v_2}(W_2 [b, \text{tri}_{v_1}(W_1 b)]), \ldots\right] \\ \text{s.t.} & w^t \begin{bmatrix} A & 0 & 0 & \\ \text{tri}_{v_1}(W_1 A) & -I & 0 & \\ \text{tri}_{v_2}(W_2 \begin{bmatrix} A \\ \text{tri}_{v_1}(W_1 A) \end{bmatrix}) & \text{abs}_{v_2}(W_2 \begin{bmatrix} 0 \\ -I \end{bmatrix}) & -I & \\ & & & \ddots \end{bmatrix} \leq [c, 0, 0, \ldots] \end{cases}$$

Connection

Now, introduce the functions

$$f_k(y) = [y,\ \mathrm{tri}_{v_k}(W_k^t y)],$$

where [,] denotes concatenation.
Note that each f_k is subadditive, and has UDDZ

$$\bar{f}_k(y) = [y,\ \mathrm{abs}_{v_k}(W_k^t y)],$$

Then, our expression can be written (more or less) compactly as

Connection

$$= \left\{ \begin{array}{ll} \max\limits_{\theta,w} & w^t f_K(\cdots f_1(b)) \\ \text{s.t.} & w^t \left[f_K(\cdots f_1(A)), \quad \bar{f}_K(\cdots \bar{f}_2(\begin{bmatrix} 0 \\ -I \end{bmatrix})), \quad \bar{f}_K(\cdots \bar{f}_3(\begin{bmatrix} 0 \\ 0 \\ -I \end{bmatrix})), \cdots \right] \leq [c, 0, 0, \ldots] \end{array} \right.$$

$$= \left\{ \begin{array}{ll} \max\limits_{\theta,w} & w^t f_K(\cdots f_1(b)) \\ \text{s.t.} & w^t f_K(\cdots f_1(A)) \leq c \\ & w^t \bar{f}_K(\cdots \bar{f}_2(\begin{bmatrix} 0 \\ -I \end{bmatrix})) \leq 0 \\ & \cdots \\ & w^t \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -I \end{bmatrix} \leq 0 \end{array} \right.$$

Connection

Let us write

$$f_{w,\theta}(y) = w^t f_K \circ \cdots \circ f_1(y),$$

a function parametrized by $(w, \theta) = (w, v_1, W_1, \ldots, v_K, W_K)$.
Then, we really showed that

$$\max_{\theta} \text{LP-Val}_{\theta}(A, b, c) = \begin{cases} \max_{\theta,w} & f_{w,\theta}(b) \\ \text{s.t.} & f_{w,\theta}(A) \leq c, \\ & w^t \bar{f}_K \circ \bar{f}_2([0, -I]) \leq 0, \\ & \cdots \\ & w^t \bar{f}_K([0, -I]) \leq 0, \\ & w^t [0, -I] \leq 0. \end{cases}$$

Theorem 1. *Consider a function* $g = g_n \circ \cdots \circ g_1$*, where each* $g_k : \mathbb{R}^{m_{k-1}} \to \mathbb{R}^{m_k}$ *is of the form* $g_k(y) = [M_k y, \tilde{g}_k(y)]$ *for* M_k *an arbitrary matrix and* $\tilde{g}_k$ *a subadditive function. If the following criterion holds*

$$\begin{aligned} \overline{g_n \circ g_2}([0, -I]) &\leqslant 0, \\ &\cdots \\ \overline{g_n}([0, -I]) &\leqslant 0, \end{aligned} \tag{13}$$

then g *is subadditive.*

Connection

Therefore, we have

$$\max_{\theta} \ \text{LP-Val}_{\theta}(A, b, c) = \begin{array}{ll} \max_{\theta, w} & f_{w,\theta}(b) \\ \text{s.t.} & f_{w,\theta}(A) \leq c, \\ & f_{w,\theta} \text{ is subadditive by Theorem 1} \end{array}$$

i.e.,

continuous cuts optimization
= subadditive dual with a "subadditive neural net", whose subadditivity is guaranteed by the criterion of Theorem 1

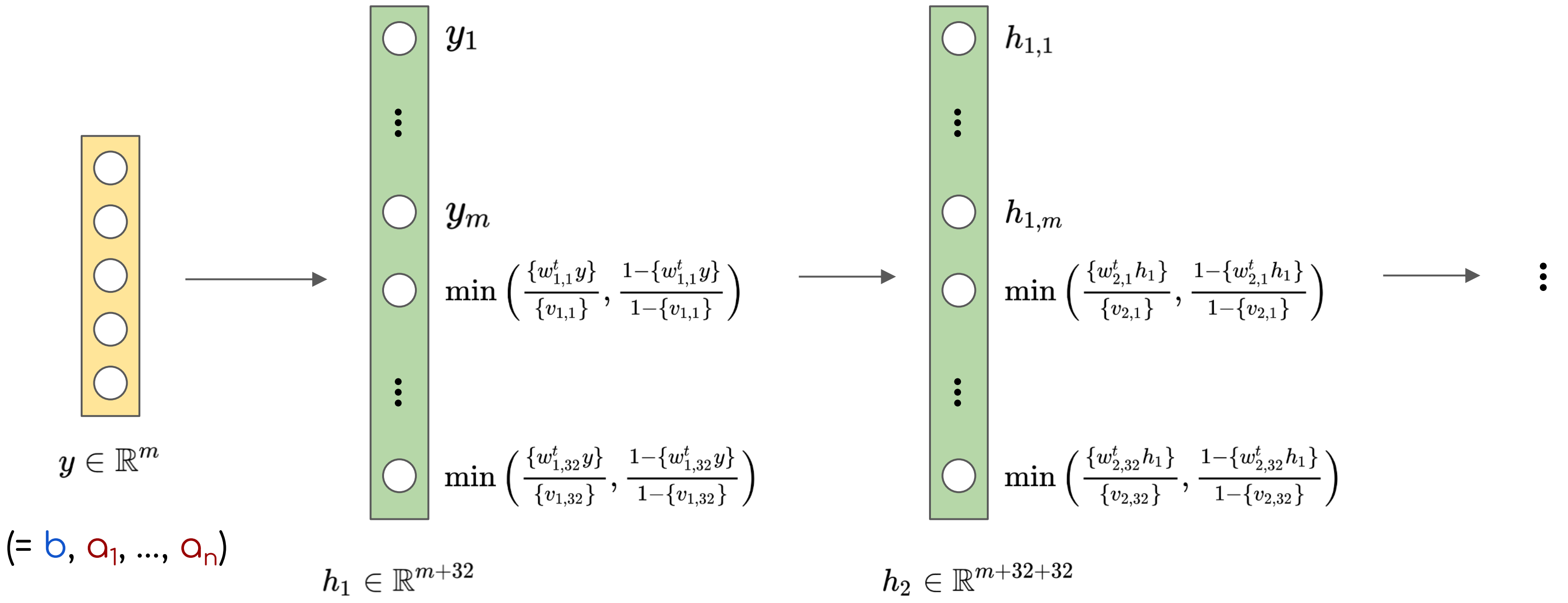
Connection

Also:

1. **classical GMI separation algorithm**
   = greedy layer-by-layer training of subadditive neural net

2. **continuous cuts optimization**
   = end-to-end training of subadditive neural net

Diagram of a "subadditive neural network" *f(y)*



Thank you!