

Cutting-Planes by projecting instead of separating

Daniel Porumbel

CEDRIC CS Lab, CNAM,
2 rue Conté, 75003 Paris, France

- 1 The general idea of projecting instead of separating
- 2 Linear robust optimization
- 3 The Benders reformulation
- 4 Column Generation for graph coloring
- 5 Conclusions

Input : a polytope P with prohibitively-many constraints

Goal : “upgrade” the standard Cutting-Planes (right) to a new method that uses projections inside the polytope P

- ★ The separation sub-problem will be upgraded to the projection sub-problem

Recall : Each iteration k corresponds to an outer approximation P_k of P and the Cutting-Planes has to separate $\text{opt}(P_k)$

Given $\mathbf{x} \in P$ and a direction $\mathbf{d} \in \mathbb{R}^n$, the projection of \mathbf{x} along \mathbf{d} asks to find the maximum step length t^* such that $\mathbf{x} + t^*\mathbf{d} \in P$

Using such **projections**, the new method generates a convergent sequence of inner solutions.

- 1 At each iteration k , the projection $\mathbf{x}_k \rightarrow \mathbf{d}_k$ generates a **contact point** $\mathbf{x}_k + t_k^*\mathbf{d}_k$ and a first-hit facet

Given $\mathbf{x} \in P$ and a direction $\mathbf{d} \in \mathbb{R}^n$, the projection of \mathbf{x} along \mathbf{d} asks to find the maximum step length t^* such that $\mathbf{x} + t^*\mathbf{d} \in P$

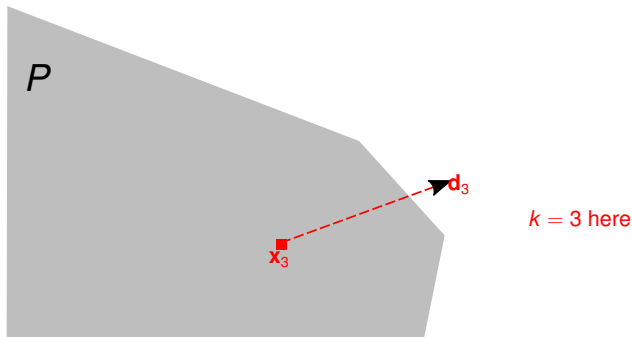
Using such **projections**, the new method generates a convergent sequence of inner solutions.

- 1 At each iteration k , the projection $\mathbf{x}_k \rightarrow \mathbf{d}_k$ generates a **contact point** $\mathbf{x}_k + t_k^*\mathbf{d}_k$ and a first-hit facet

Given $\mathbf{x} \in P$ and a direction $\mathbf{d} \in \mathbb{R}^n$, the projection of \mathbf{x} along \mathbf{d} asks to find the maximum step length t^* such that $\mathbf{x} + t^*\mathbf{d} \in P$

Using such **projections**, the new method generates a convergent sequence of inner solutions.

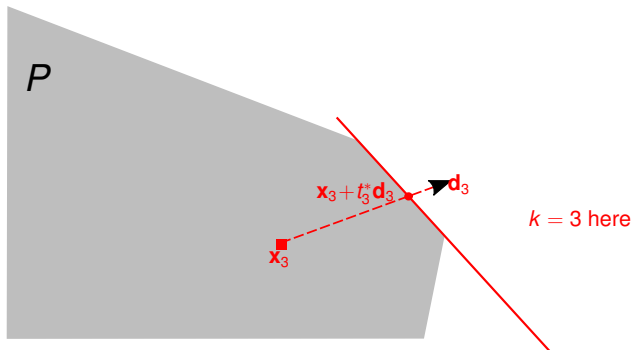
- 1 At each iteration k , the projection $\mathbf{x}_k \rightarrow \mathbf{d}_k$ generates a **contact point** $\mathbf{x}_k + t_k^*\mathbf{d}_k$ and a first-hit facet



Given $\mathbf{x} \in P$ and a direction $\mathbf{d} \in \mathbb{R}^n$, the projection of \mathbf{x} along \mathbf{d} asks to find the maximum step length t^* such that $\mathbf{x} + t^*\mathbf{d} \in P$

Using such **projections**, the new method generates a convergent sequence of inner solutions.

- 1 At each iteration k , the projection $\mathbf{x}_k \rightarrow \mathbf{d}_k$ generates a **contact point** $\mathbf{x}_k + t_k^*\mathbf{d}_k$ and a first-hit facet

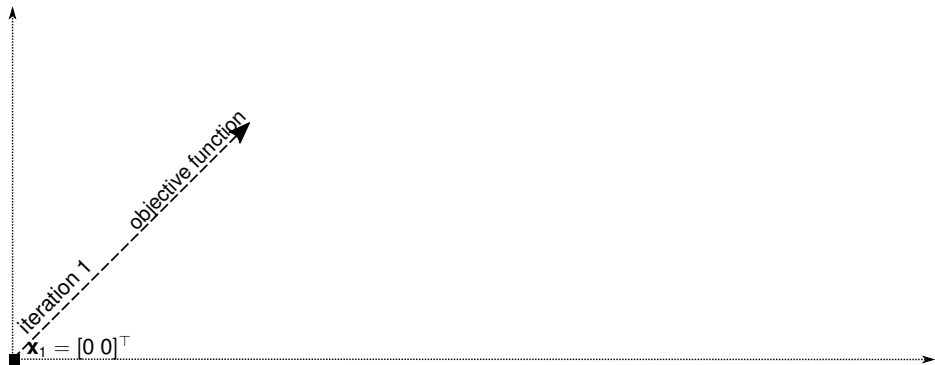


Problems addressed with the new method

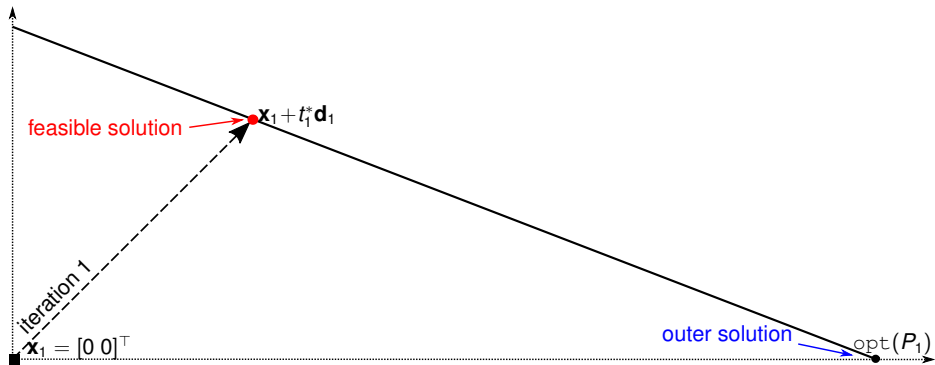
- 1 Graph coloring (dual polytope in Column Generation)
- 2 A Benders's `Cutting-Planes` problem (primal polytope)
- 3 A robust optimization problem (primal polytope)
- 4 Cutting-Stock with multiple lengths (dual polytope)

Different techniques have been used to solve the projection sub-problem for these different problems :

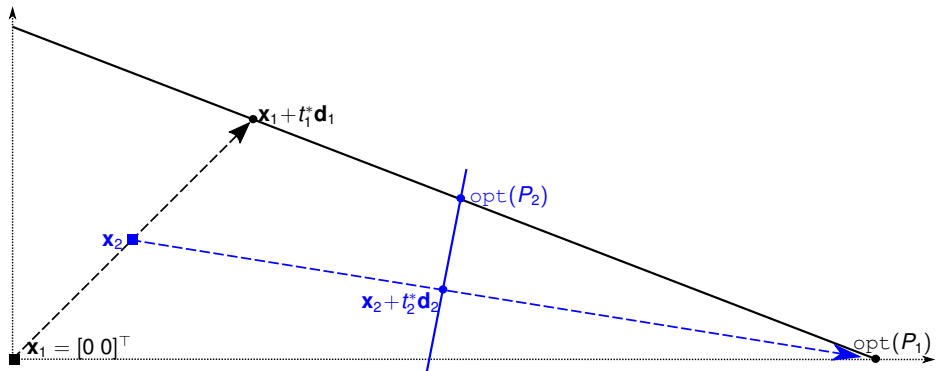
- The Charnes-Cooper transformation for 2, an ad-hoc method for 3 or Dynamic Programming for 4
- I'll focus on 1



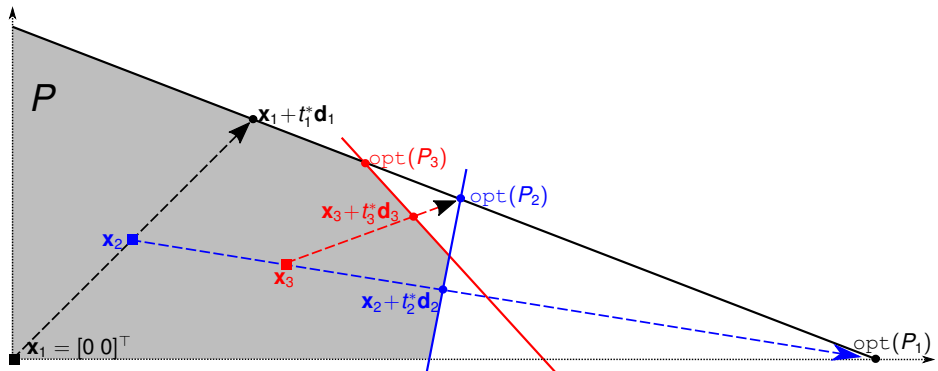
Iteration 1 : uncharted territory, follow objective function, i.e., advance along $\mathbf{x}_1 \rightarrow \mathbf{d}_1$ where \mathbf{d}_1 takes the value of the objective function



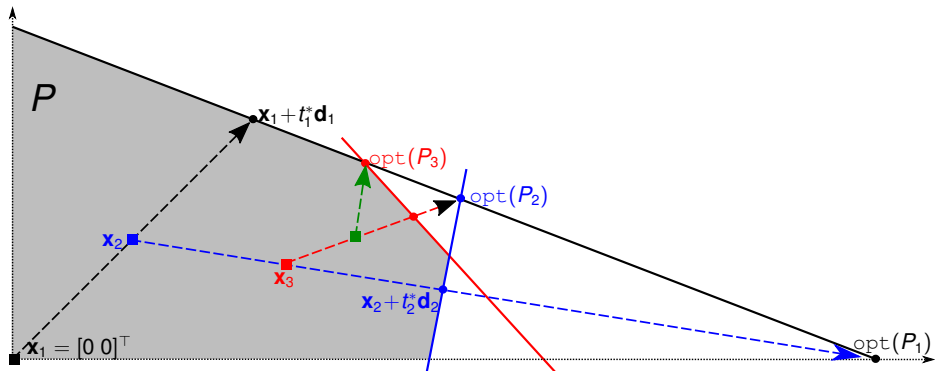
Iteration 1 : found a first outer solution $\text{opt}(P_1)$ and a first inner solution (contact point) $\mathbf{x}_1 + t_1^* \mathbf{d}_1$



Iteration 2 : an inner feasible solution (contact point) $\mathbf{x}_2 + t_2^* \mathbf{d}_2$ and a new outer solution. We take $\mathbf{d}_2 = \text{opt}(P_1) - \mathbf{x}_2$.



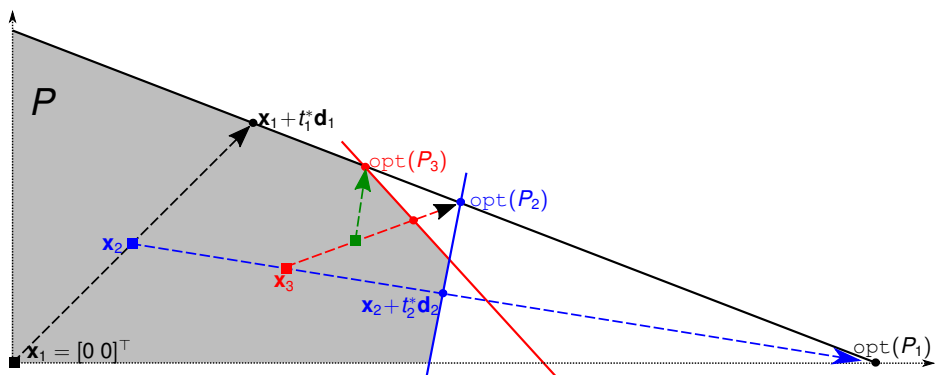
Iteration 3 : the feasible solution $\mathbf{x}_3 + t_3^* \mathbf{d}_3$ is almost optimal



Iteration 4 : optimality of $\text{opt}(P_3)$ proved

You can see the proposed method is convergent because it solves a separation problem on $\text{opt}(P_k)$ at each iteration k

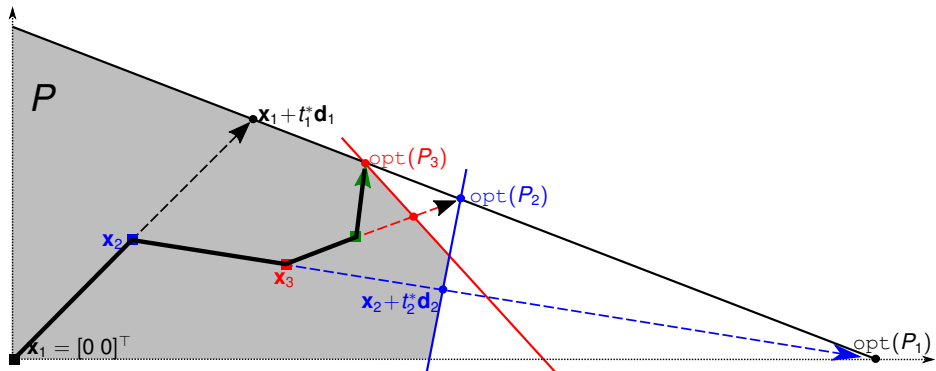
- The convergence proof takes two lines, cool !



Building on existing work [1,2], the new method was deliberately designed to be more **general** and when possible **simpler**

[1] Daniel Porumbel. Ray projection for optimizing polytopes with prohibitively many constraints in set-covering column generation. *Mathematical Programming*, 155(1) :147–197, 2016.

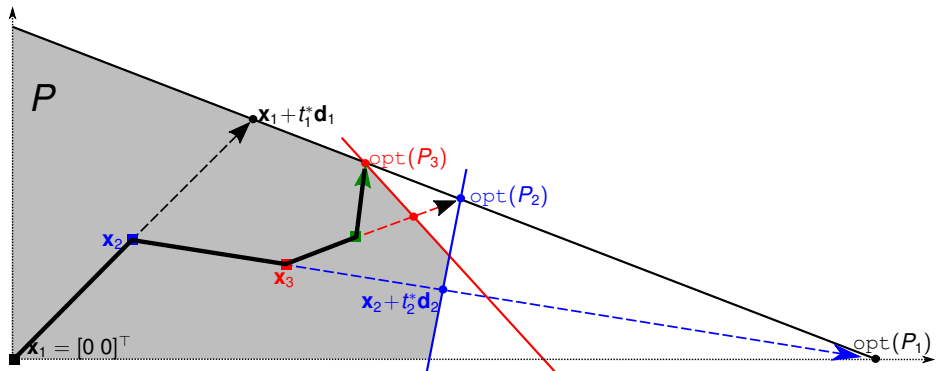
[2] Daniel Porumbel. From the separation to the intersection subproblem for optimizing polytopes with prohibitively many constraints in a Benders decomposition context. *Discrete Optimization*, 2018.



Notice the trajectory of the inner points — there is no built-in feature in the `Cutting-Planes` to generate inner points

- each x_k is a point on the last projected segment, i.e., between \mathbf{x}_{k-1} and $\mathbf{x}_{k-1} + t_{k-1}^* \mathbf{d}_{k-1}$
- in this example we choose : $x_k = \mathbf{x}_{k-1} + \frac{1}{2} \cdot t_{k-1}^* \mathbf{d}_{k-1}$

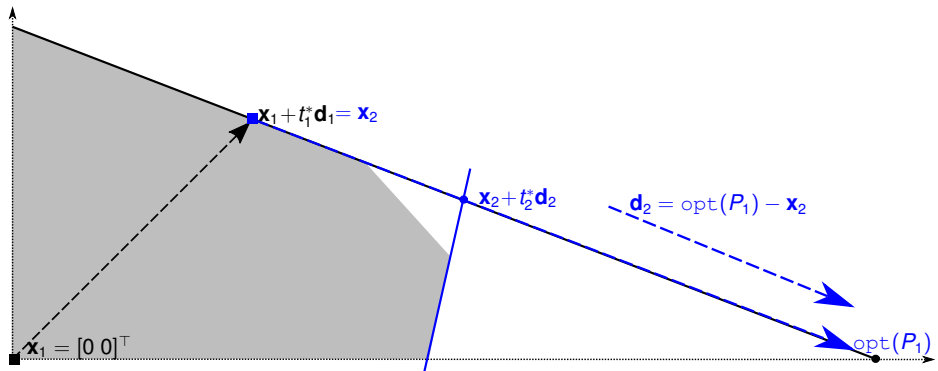
everything was like a movie until here : let's move to real life



Notice the trajectory of the inner points — there is no built-in feature in the `Cutting-Planes` to generate inner points

- each x_k is a point on the last projected segment, i.e., between \mathbf{x}_{k-1} and $\mathbf{x}_{k-1} + t_{k-1}^* \mathbf{d}_{k-1}$
- in this example we choose : $x_k = \mathbf{x}_{k-1} + \frac{1}{2} \cdot t_{k-1}^* \mathbf{d}_{k-1}$

everything was like a movie until here : let's move to real life



Notice the trajectory of the inner points — there is no built-in feature in the `Cutting-Planes` to generate inner points

- each x_k is a point on the last projected segment, i.e., between \mathbf{x}_{k-1} and $\mathbf{x}_{k-1} + t_{k-1}^* \mathbf{d}_{k-1}$
- we here choose the contact point : $x_k = \mathbf{x}_{k-1} + t_{k-1}^* \mathbf{d}_{k-1}$

everything was like a movie until here : let's move to real life

The projection sub-problem

We are given :

$$\max \{ \mathbf{b}^\top \mathbf{x} : \mathbf{a}^\top \mathbf{x} \leq c_a, \forall (\mathbf{a}, c_a) \in \text{Constr} \} = \max \{ \mathbf{b}^\top \mathbf{x} : \mathbf{x} \in P \}$$

Separation sub-problem on \mathbf{x}

$$\min \{ c_a - \mathbf{a}^\top \mathbf{x} : (\mathbf{a}, c_a) \in \text{Constr} \}$$

Projection sub-problem on $\mathbf{x} \rightarrow \mathbf{d}$

\implies find max t^* such that $\mathbf{a}^\top \mathbf{x} + t^* \cdot \mathbf{a}^\top \mathbf{d} \leq c_a \forall (\mathbf{a}, c_a) \in \text{Constr}$

$$t^* = \min \left\{ \frac{c_a - \mathbf{a}^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d}} : (\mathbf{a}, c_a) \in \text{Constr}, \mathbf{d}^\top \mathbf{a} > 0 \right\}.$$

The projection sub-problem

We are given :

$$\max \{ \mathbf{b}^\top \mathbf{x} : \mathbf{a}^\top \mathbf{x} \leq c_a, \forall (\mathbf{a}, c_a) \in \text{Constr} \} = \max \{ \mathbf{b}^\top \mathbf{x} : \mathbf{x} \in P \}$$

Separation sub-problem on \mathbf{x}

$$\min \{ c_a - \mathbf{a}^\top \mathbf{x} : (\mathbf{a}, c_a) \in \text{Constr} \}$$

Projection sub-problem on $\mathbf{x} \rightarrow \mathbf{d}$

\implies find $\max t^*$ such that $\mathbf{a}^\top \mathbf{x} + t^* \cdot \mathbf{a}^\top \mathbf{d} \leq c_a \forall (\mathbf{a}, c_a) \in \text{Constr}$

$$t^* = \min \left\{ \frac{c_a - \mathbf{a}^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d}} : (\mathbf{a}, c_a) \in \text{Constr}, \mathbf{d}^\top \mathbf{a} > 0 \right\}.$$

The projection sub-problem

We are given :

$$\max \{ \mathbf{b}^\top \mathbf{x} : \mathbf{a}^\top \mathbf{x} \leq c_a, \forall (\mathbf{a}, c_a) \in \text{Constr} \} = \max \{ \mathbf{b}^\top \mathbf{x} : \mathbf{x} \in P \}$$

Separation sub-problem on \mathbf{x}

$$\min \{ c_a - \mathbf{a}^\top \mathbf{x} : (\mathbf{a}, c_a) \in \text{Constr} \}$$

Projection sub-problem on $\mathbf{x} \rightarrow \mathbf{d}$

\implies find $\max t^*$ such that $\mathbf{a}^\top \mathbf{x} + t^* \cdot \mathbf{a}^\top \mathbf{d} \leq c_a \forall (\mathbf{a}, c_a) \in \text{Constr}$

$$t^* = \min \left\{ \frac{c_a - \mathbf{a}^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d}} : (\mathbf{a}, c_a) \in \text{Constr}, \mathbf{d}^\top \mathbf{a} > 0 \right\}.$$

- 1 The general idea of projecting instead of separating
- 2 **Linear robust optimization**
- 3 The Benders reformulation
- 4 Column Generation for graph coloring
- 5 Conclusions

Introducing the Robust Linear Program

Start from a standard LP with a feasible area described by :

$$\mathbf{a}^\top \mathbf{x} \leq c_a \quad \forall (\mathbf{a}, c_a) \in \text{Constr}_{\text{nom}}$$

For each nominal constraint $(\mathbf{a}, c_a) \in \text{Constr}_{\text{nom}}$ one can define a (huge) set of robust constraints

$$(\mathbf{a} + \mathbf{a}')^\top \mathbf{x} \leq c_a,$$

where any \mathbf{a}' belongs to a set a (reasonable) deviation of the nominal coefficients \mathbf{a} .

More exactly : $\mathbf{a}' \in \mathbb{R}^n$ is a vector with at maximum Γ non-zero components such that $a'_i \in \{-0.01 \cdot a_i, 0, 0.01 \cdot a_i\} \forall i \in [1..n]$.

- We write $\mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a})$

I only compare to the cutting-planes from [M. Fischetti and M. Monaci. Cutting plane versus compact formulations for uncertain (integer) linear programs. Mathematical Programming Computation, 4(3) :239–273, 2012.]

Introducing the Robust Linear Program

Start from a standard LP with a feasible area described by :

$$\mathbf{a}^\top \mathbf{x} \leq c_a \quad \forall (\mathbf{a}, c_a) \in \text{Constr}_{\text{nom}}$$

For each nominal constraint $(\mathbf{a}, c_a) \in \text{Constr}_{\text{nom}}$ one can define a (huge) set of robust constraints

$$(\mathbf{a} + \mathbf{a}')^\top \mathbf{x} \leq c_a,$$

where any \mathbf{a}' belongs to a set a (reasonable) deviation of the nominal coefficients \mathbf{a} .

More exactly : $\mathbf{a}' \in \mathbb{R}^n$ is a vector with at maximum Γ non-zero components such that $a'_i \in \{-0.01 \cdot a_i, 0, 0.01 \cdot a_i\} \forall i \in [1..n]$.

- We write $\mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a})$

I only compare to the cutting-planes from [M. Fischetti and M. Monaci. Cutting plane versus compact formulations for uncertain (integer) linear programs. Mathematical Programming Computation, 4(3) :239–273, 2012.]

Introducing the Robust Linear Program

Start from a standard LP with a feasible area described by :

$$\mathbf{a}^\top \mathbf{x} \leq c_a \quad \forall (\mathbf{a}, c_a) \in \text{Constr}_{\text{nom}}$$

For each nominal constraint $(\mathbf{a}, c_a) \in \text{Constr}_{\text{nom}}$ one can define a (huge) set of robust constraints

$$(\mathbf{a} + \mathbf{a}')^\top \mathbf{x} \leq c_a,$$

where any \mathbf{a}' belongs to a set a (reasonable) deviation of the nominal coefficients \mathbf{a} .

More exactly : $\mathbf{a}' \in \mathbb{R}^n$ is a vector with at maximum Γ non-zero components such that $a'_i \in \{-0.01 \cdot a_i, 0, 0.01 \cdot a_i\} \quad \forall i \in [1..n]$.

- We write $\mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a})$

I only compare to the cutting-planes from [M. Fischetti and M. Monaci. Cutting plane versus compact formulations for uncertain (integer) linear programs. *Mathematical Programming Computation*, 4(3) :239–273, 2012.]

The projection sub-problem

We need the maximum t such that

$$(\mathbf{a} + \mathbf{a}')^\top (\mathbf{x} + t\mathbf{d}) \leq c_a \quad \forall (\mathbf{a}, c_a) \in \text{Constr}_{\text{nom}}, \mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a}),$$

i.e., we consider all nominal constraints (\mathbf{a}, c_a) and all their deviations $\mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a})$

Let's work this formula for each nominal constraint (\mathbf{a}, c_a)

The projection sub-problem asks to minimize

$$\min_{\mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a})} \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}}$$

This could be written as a linear-fractional program (because \mathbf{a}' can be seen as belonging to a polytope) and solved with the Charnes-Cooper transformation.

The projection sub-problem

We need the maximum t such that

$$(\mathbf{a} + \mathbf{a}')^\top (\mathbf{x} + t\mathbf{d}) \leq c_a \quad \forall (\mathbf{a}, c_a) \in \text{Constr}_{\text{nom}}, \mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a}),$$

i.e., we consider all nominal constraints (\mathbf{a}, c_a) and all their deviations $\mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a})$

Let's work this formula for each nominal constraint (\mathbf{a}, c_a)

The projection sub-problem asks to minimize

$$\min_{\mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a})} \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}}$$

This could be written as a linear-fractional program (because \mathbf{a}' can be seen as belonging to a polytope) and solved with the Charnes-Cooper transformation.

The projection sub-problem

We need the maximum t such that

$$(\mathbf{a} + \mathbf{a}')^\top (\mathbf{x} + t\mathbf{d}) \leq c_a \quad \forall (\mathbf{a}, c_a) \in \text{Constr}_{\text{nom}}, \mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a}),$$

i.e., we consider all nominal constraints (\mathbf{a}, c_a) and all their deviations $\mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a})$

Let's work this formula for each nominal constraint (\mathbf{a}, c_a)

The projection sub-problem asks to minimize

$$\min_{\mathbf{a}' \in \text{Dev}_\Gamma(\mathbf{a})} \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}}$$

This could be written as a linear-fractional program (because \mathbf{a}' can be seen as belonging to a polytope) and solved with the Charness-Cooper transformation.

A step-by-step t -decreasing algorithm

Start with the t value given by the nominal constraint alone, *i.e.*, fix $\mathbf{a}' = \mathbf{0}$ in formula below :

$$t \leftarrow \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}} \quad \mathbf{a}' = \mathbf{0}_n \quad \frac{c_a - \mathbf{a}^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d}}$$

- 1 This t is not necessarily optimal because there might exist a different $\mathbf{a}' (\neq \mathbf{0}_n)$ such that

$$t > \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}}$$

equivalent to

$$t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}) > c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}$$

- 2 Solve $\min_{\mathbf{a}' \in \text{Dev}_r(\mathbf{a})} c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x} - t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d})$

- 3 Repeat from 1 while the optimum of above LP is below 0

A step-by-step t -decreasing algorithm

Start with the t value given by the nominal constraint alone, *i.e.*, fix $\mathbf{a}' = \mathbf{0}$ in formula below :

$$t \leftarrow \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}} \quad \mathbf{a}' = \mathbf{0}_n \quad \frac{c_a - \mathbf{a}^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d}}$$

- 1 This t is not necessarily optimal because there might exist a different $\mathbf{a}' (\neq \mathbf{0}_n)$ such that

$$t > \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}}$$

equivalent to

$$t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}) > c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}$$

- 2 Solve $\min_{\mathbf{a}' \in \text{Dev}_r(\mathbf{a})} c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x} - t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d})$

- 3 Repeat from 1 while the optimum of above LP is below 0

A step-by-step t -decreasing algorithm

Start with the t value given by the nominal constraint alone, *i.e.*, fix $\mathbf{a}' = \mathbf{0}$ in formula below :

$$t \leftarrow \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}} \quad \mathbf{a}' = \mathbf{0}_n \quad \frac{c_a - \mathbf{a}^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d}}$$

- 1 This t is not necessarily optimal because there might exist a different $\mathbf{a}' (\neq \mathbf{0}_n)$ such that

$$t > \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}}$$

equivalent to

$$t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}) > c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}$$

- 2 Solve $\min_{\mathbf{a}' \in \text{Dev}_r(\mathbf{a})} c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x} - t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d})$

- 3 Repeat from 1 while the optimum of above LP is below 0

A step-by-step t -decreasing algorithm

Start with the t value given by the nominal constraint alone, *i.e.*, fix $\mathbf{a}' = \mathbf{0}$ in formula below :

$$t \leftarrow \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}} \quad \mathbf{a}' = \mathbf{0}_n \quad \frac{c_a - \mathbf{a}^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d}}$$

- 1 This t is not necessarily optimal because there might exist a different $\mathbf{a}' (\neq \mathbf{0}_n)$ such that

$$t > \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}}$$

equivalent to

$$t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}) > c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}$$

- 2 Solve $\min_{\mathbf{a}' \in \text{Dev}_r(\mathbf{a})} c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x} - t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d})$

- 3 Repeat from 1 while the optimum of above LP is below 0

A step-by-step t -decreasing algorithm

Start with the t value given by the nominal constraint alone, *i.e.*, fix $\mathbf{a}' = \mathbf{0}$ in formula below :

$$t \leftarrow \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}} \quad \mathbf{a}' = \mathbf{0}_n \quad \frac{c_a - \mathbf{a}^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d}}$$

- 1 This t is not necessarily optimal because there might exist a different $\mathbf{a}' (\neq \mathbf{0}_n)$ such that

$$t > \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}}$$

equivalent to

$$t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}) > c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}$$

- 2 Solve $\min_{\mathbf{a}' \in \text{Dev}_r(\mathbf{a})} c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x} - t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d})$

- 3 Repeat from 1 while the optimum of above LP is below 0

A step-by-step t -decreasing algorithm

The resulting t is given as input to the next constraint of $\text{Constr}_{\text{nom}}$, to iteratively apply steps ① — ③ to all $\text{Constr}_{\text{nom}}$

- ① This t is not necessarily optimal because there might exist a different $\mathbf{a}' (\neq \mathbf{0}_n)$ such that

$$t > \frac{c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}}$$

equivalent to

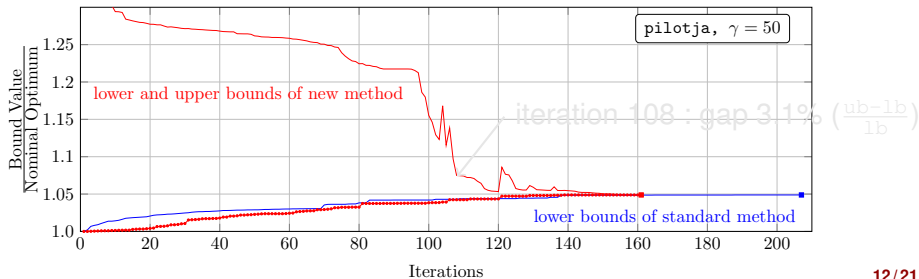
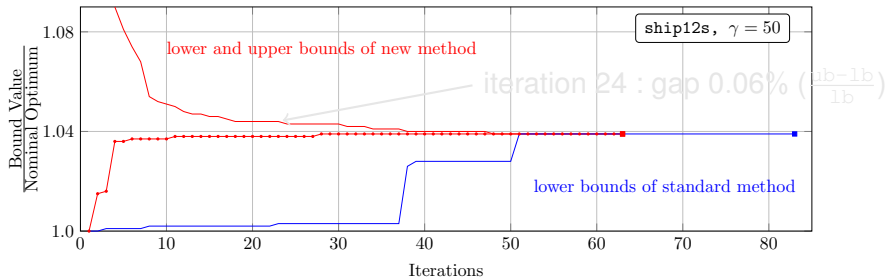
$$t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d}) > c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x}$$

- ② Solve $\min_{\mathbf{a}' \in \text{Dev}_T(\mathbf{a})} c_a - \mathbf{a}^\top \mathbf{x} - \mathbf{a}'^\top \mathbf{x} - t \cdot (\mathbf{a}^\top \mathbf{d} + \mathbf{a}'^\top \mathbf{d})$

- ③ Repeat from ① while the optimum of above LP is below 0

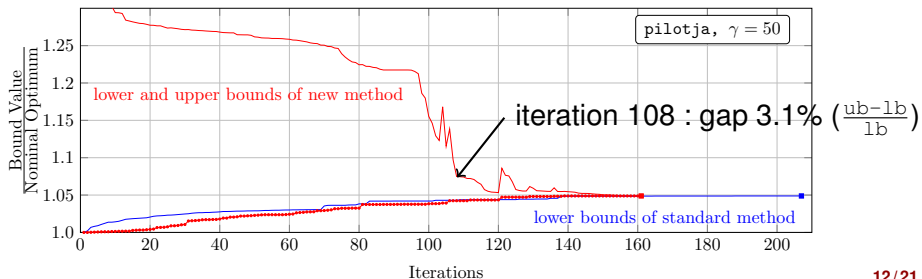
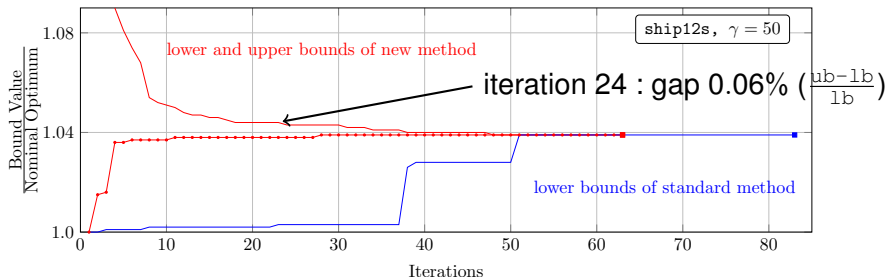
Results on the robust linear program

Each interior point is defined as : $\mathbf{x}_k = \mathbf{x}_{k-1} + \frac{1}{10} \cdot t_{k-1}^* \mathbf{d}_{k-1}$



Results on the robust linear program

Each interior point is defined as : $\mathbf{x}_k = \mathbf{x}_{k-1} + \frac{1}{10} \cdot t_{k-1}^* \mathbf{d}_{k-1}$



Plan

- 1 The general idea of projecting instead of separating
- 2 Linear robust optimization
- 3 The Benders reformulation
- 4 Column Generation for graph coloring
- 5 Conclusions

Du modèle de départ aux reformulations

coût des flux, on va utiliser $\mathbf{b} = \mathbf{0}$

$$\min \mathbf{d}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{y}$$

$$\mathbf{D}\mathbf{x} \geq \mathbf{e}$$

Contraintes de design

$$\mathbf{B}\mathbf{x} + \mathbf{A}\mathbf{y} \geq \mathbf{c}$$

$$\mathbf{x} \in \mathbb{Z}_+^n, \mathbf{y} \geq \mathbf{0}$$

Le flux \mathbf{y} doit pouvoir passer

\mathbf{x} est un nombre d'unités à faire fonctionner.

- câbles à monter
- entrepôts à ouvrir

\mathbf{y} est un coût des flux qui passent, coût d'affectations, etc

Du modèle de départ aux reformulations

coût des flux, on va utiliser $\mathbf{b} = \mathbf{0}$

$$\min \mathbf{d}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{y}$$

$$\mathbf{D}\mathbf{x} \geq \mathbf{e}$$

Contraintes de design

$$\mathbf{B}\mathbf{x} + \mathbf{A}\mathbf{y} \geq \mathbf{c}$$

$$\mathbf{x} \in \mathbb{Z}_+^n, \mathbf{y} \geq \mathbf{0}$$

Le flux \mathbf{y} doit pouvoir passer

Reformulation 1 :

$$\min \mathbf{d}^\top \mathbf{x} + \hat{z}$$

$$\mathbf{D}\mathbf{x} \geq \mathbf{e}$$

$$\hat{z} = \min \{ \mathbf{b}^\top \mathbf{y} : \mathbf{B}\mathbf{x} + \mathbf{A}\mathbf{y} \geq \mathbf{c}, \mathbf{y} \geq \mathbf{0} \}$$

$$\mathbf{x} \in \mathbb{Z}_+^n$$

Du modèle de départ aux reformulations

$$\begin{aligned} \min \quad & \mathbf{d}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{y} \\ \mathbf{Dx} \quad & \geq \mathbf{e} \\ \mathbf{Bx} + \mathbf{Ay} \quad & \geq \mathbf{c} \\ \mathbf{x} \in \mathbb{Z}_+^n, \quad & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

Reformulation 1 :

$$\begin{aligned} \min \quad & \mathbf{d}^\top \mathbf{x} + \hat{z} \\ \mathbf{Dx} \quad & \geq \mathbf{e} \\ \hat{z} = \min \quad & \{ \mathbf{b}^\top \mathbf{y} : \mathbf{Bx} + \mathbf{Ay} \geq \mathbf{c}, \mathbf{y} \geq \mathbf{0} \} \\ \mathbf{x} \in \mathbb{Z}_+^n \end{aligned}$$

On va dualiser ce PL

Du modèle de départ aux reformulations

coût des flux, on va utiliser $\mathbf{b} = \mathbf{0}$
 $\Rightarrow \hat{z} = 0$

$$\begin{aligned} \min \mathbf{d}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{y} \\ \mathbf{Dx} \geq \mathbf{e} \\ \mathbf{Bx} + \mathbf{Ay} \geq \mathbf{c} \\ \mathbf{x} \in \mathbb{Z}_+^n, \mathbf{y} \geq \mathbf{0} \end{aligned}$$

Reformulation 2 :

$$\begin{aligned} \min \mathbf{d}^\top \mathbf{x} + \hat{z} \\ \mathbf{Dx} \geq \mathbf{e} \\ \hat{z} = \max\{(\mathbf{c} - \mathbf{Bx})^\top \mathbf{u} : \mathbf{u} \in \mathcal{P}\}, \\ \mathbf{x} \in \mathbb{Z}_+^n \end{aligned}$$

$$\mathcal{P} = \{\mathbf{u} \geq \mathbf{0} : \mathbf{A}^\top \mathbf{u} \leq \mathbf{b}\}$$

Du modèle de départ aux reformulations

coût des flux, on va utiliser $\mathbf{b} = \mathbf{0}$
 $\Rightarrow \hat{z} = 0$

$$\begin{aligned} \min \mathbf{d}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{y} \\ \mathbf{Dx} \geq \mathbf{e} \\ \mathbf{Bx} + \mathbf{Ay} \geq \mathbf{c} \\ \mathbf{x} \in \mathbb{Z}_+^n, \mathbf{y} \geq \mathbf{0} \end{aligned}$$

Reformulation 3 :

$$\begin{aligned} \min \mathbf{d}^\top \mathbf{x} + \hat{z} \\ \mathbf{Dx} \geq \mathbf{e} \\ \hat{z} = \max\{(\mathbf{c} - \mathbf{Bx})^\top \mathbf{u} : \mathbf{u} \in \mathcal{P}\} \\ 0 \geq (\mathbf{c} - \mathbf{Bx})^\top \mathbf{u} \quad \forall \mathbf{u} \in \mathcal{P}, \\ \mathbf{x} \in \mathbb{Z}_+^n \\ \mathcal{P} = \{\mathbf{u} \geq \mathbf{0} : \mathbf{A}^\top \mathbf{u} \leq \mathbf{0}\} \end{aligned}$$

Du modèle de départ aux reformulations

$$\begin{aligned} \min \quad & \mathbf{d}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{y} \\ & \mathbf{D}\mathbf{x} \geq \mathbf{e} \\ & \mathbf{B}\mathbf{x} + \mathbf{A}\mathbf{y} \geq \mathbf{c} \\ & \mathbf{x} \in \mathbb{Z}_+^n, \mathbf{y} \geq \mathbf{0} \end{aligned}$$

Reformulation 4 :

$$\begin{aligned} \min \quad & \mathbf{d}^\top \mathbf{x} \\ & \mathbf{D}\mathbf{x} \geq \mathbf{e} \\ & (\mathbf{B}\mathbf{x})^\top \mathbf{u} \geq \mathbf{c}^\top \mathbf{u} \quad \forall \mathbf{u} \in \mathcal{P}, \\ & \mathbf{x} \in \mathbb{Z}_+^n \end{aligned}$$

Les coupes Benders sont définies par les rayons \mathbf{u} du polytope Benders \mathcal{P}

$$\mathcal{P} = \{ \mathbf{u} \geq \mathbf{0} : \mathbf{A}^\top \mathbf{u} \leq \mathbf{0} \}$$

Du modèle de départ aux reformulations

$$\min \mathbf{d}^\top \mathbf{x} + \mathbf{b}^\top \mathbf{y}$$

$$\mathbf{Dx} \geq \mathbf{e}$$

$$\mathbf{Bx} + \mathbf{Av} > \mathbf{c}$$

Projection $\mathbf{x} \rightarrow \mathbf{d}$

$$\max_{\mathbf{u} \in \mathcal{P}} \left\{ \frac{(\mathbf{Bx})^\top \mathbf{u} - \mathbf{c}^\top \mathbf{u}}{-\mathbf{(Bd)}^\top \mathbf{u}} : -\mathbf{(Bd)}^\top \mathbf{u} > 0 \right\}$$

Reformulation 4 :

$$\min \mathbf{d}^\top \mathbf{x}$$

$$\mathbf{Dx} \geq \mathbf{e}$$

$$(\mathbf{Bx})^\top \mathbf{u} \geq \mathbf{c}^\top \mathbf{u} \quad \forall \mathbf{u} \in \mathcal{P},$$

$$\mathbf{x} \in \mathbb{Z}_+^n$$

$$\mathcal{P} = \{\mathbf{u} \geq \mathbf{0} : \mathbf{A}^\top \mathbf{u} \leq \mathbf{0}\}$$

Résolution sous-problème d'intersection

Il faut résoudre

$$t^* = \max_{\mathbf{u} \in \mathcal{P}} \left\{ \frac{(\mathbf{Bx})^\top \mathbf{u} - \mathbf{c}^\top \mathbf{u}}{-(\mathbf{Bd})^\top \mathbf{u}} : -(\mathbf{Bd})^\top \mathbf{u} > 0 \right\}$$

$$\text{avec } \mathcal{P} = \{\mathbf{u} \geq \mathbf{0} : \mathbf{A}^\top \mathbf{u} \leq \mathbf{0}\}$$

Charnes–Cooper transformation :

$$\bar{\mathbf{u}} = \frac{\mathbf{u}}{-(\mathbf{Bd})^\top \mathbf{u}}$$

Using $\mathbf{u} \in \mathcal{P} \implies \mathbf{A}^\top \bar{\mathbf{u}} \leq \mathbf{0}, \bar{\mathbf{u}} \geq \mathbf{0}, -(\mathbf{Bd})^\top \bar{\mathbf{u}} = 1$, we obtain

$$t^* = \min \left\{ (\mathbf{Bx})^\top \bar{\mathbf{u}} - \mathbf{c}^\top \bar{\mathbf{u}} : \mathbf{A}^\top \bar{\mathbf{u}} \leq \mathbf{0}, \bar{\mathbf{u}} \geq \mathbf{0}, -(\mathbf{Bd})^\top \bar{\mathbf{u}} = 1 \right\}$$

Résultats

Instance		Projective Cutting-Planes				Standard Cutting-Planes		
OPT	Best IP	Iterations	Time [secs]	Time solve	Iterations	Time [secs]	Time solve	
	Sol	avg (dev)	avg (dev)	master		avg (dev)	avg (dev)	master
a	42.333	48	22.8 (1)	0.06 (0.002)	4.4%	35 (4.9)	0.09 (0.01)	5.5%
b	245.67	265	73.8 (2.7)	0.2 (0.006)	6.1%	131 (11.8)	0.4 (0.04)	8.7%
c	204.33	220	56.5 (1.5)	0.2 (0.004)	4.9%	78.5 (16)	0.2 (0.05)	5.8%
d	299.33	317	67.5 (3)	0.2 (0.01)	4.3%	104 (4.3)	0.4 (0.02)	6.1%
e	67.333	77	35.4 (0.8)	0.1 (0.006)	4.2%	39.5 (5.5)	0.1 (0.02)	5.5%
a	46		174 (27.4)	7.4 (5.8)	89.5%	229 (146)	9.6 (3)	95%
b	260		824 (206)	1073 (636)	99.5%	2987 (2427)	4129 (819)	99.8%
c	214		242 (27.1)	99 (31.6)	98.4%	526 (442)	378 (70.8)	99.6%
d	313		336 (53.4)	321 (103)	99.2%	1315 (1049)	2367 (469)	99.8%
e	74		1336 (138)	4907 (1640)	99.8%	2250 (1292)	6703 (2857)	99.9%

Les 5 dernières lignes concernent le modèle IP.

« dév » = déviation standard

Plan

- 1 The general idea of projecting instead of separating
- 2 Linear robust optimization
- 3 The Benders reformulation
- 4 Column Generation for graph coloring
- 5 Conclusions

We need to focus on the dual. The primal master is :

$$\begin{aligned} \min & \sum c_a y_a \\ \mathbf{x} : & \sum a_i y_a \geq b_i \quad \forall i \in [1..n] \\ & y_a \geq 0 \quad \forall (\mathbf{a}, c_a) \in \text{Constr} \end{aligned}$$

The dual LP is :

$$P \begin{cases} \max \mathbf{b}^T \mathbf{x} & (= \max \mathbf{1}_n^T \mathbf{x}) \\ y_a : \mathbf{a}^T \mathbf{x} \leq c_a, & \forall (\mathbf{a}, c_a) \in \text{Constr} \\ \mathbf{x} \geq \mathbf{0}_n \end{cases}$$

- $c_a = 1$ for each stable \mathbf{a} (each color counts once)

We re-write the graph coloring problem :

$$P \left\{ \begin{array}{l} \max \mathbf{1}_n^\top \mathbf{x} \\ \mathbf{a}^\top \mathbf{x} \leq 1, \text{ for any stable } \mathbf{a} \in \{0, 1\}^n \\ \mathbf{x} \geq \mathbf{0}_n \end{array} \right.$$

The projection sub-problem on $\mathbf{x} \rightarrow \mathbf{d}$

$$t^* = \min \left\{ \frac{1 - \mathbf{a}^\top \mathbf{x}}{\mathbf{a}^\top \mathbf{d}} : \mathbf{a} \in \text{STAB}, \mathbf{d}^\top \mathbf{a} > 0 \right\},$$

where STAB is the set of stables that can be written as

$$\begin{array}{l} a_i + a_j \leq 1 \quad \forall \{i, j\} \in E \\ a_i \in \{0, 1\} \quad \forall i \in V \end{array}$$

$$t^* = \min_{\mathbf{a}} \frac{1 - \mathbf{x}^\top \mathbf{a}}{\mathbf{d}^\top \mathbf{a}}$$
$$\mathbf{d}^\top \mathbf{a} > 0$$

$$\text{STAB} \begin{cases} a_i + a_j \leq 1, & \forall \{i, j\} \in E \\ a_i \in \{0, 1\} & \forall i \in [1..n] \end{cases}$$

$$t^* = \min_{\mathbf{a}} \frac{1 - \mathbf{x}^\top \mathbf{a}}{\mathbf{d}^\top \mathbf{a}}$$

$$\mathbf{d}^\top \mathbf{a} > 0$$

$$\text{STAB} \begin{cases} a_i + a_j \leq 1, & \forall \{i, j\} \in E \\ a_i \in \{0, 1\} & \forall i \in [1..n] \end{cases}$$

$$t^* = \min_{\mathbf{a}, \bar{\mathbf{a}}} \bar{\alpha} - \mathbf{x}^\top \bar{\mathbf{a}}$$

$$\bar{a}_i + \bar{a}_j \leq \bar{\alpha} \quad \forall \{i, j\} \in E$$

$$\mathbf{d}^\top \bar{\mathbf{a}} = 1$$

$$\bar{a}_i \in \{0, \bar{\alpha}\} \quad \forall i \in [1..n]$$

$$\bar{\alpha} \geq 0$$

$$\bar{\mathbf{a}} = \frac{\mathbf{a}}{\mathbf{d}^\top \mathbf{a}} \text{ and } \bar{\alpha} = \frac{1}{\mathbf{d}^\top \mathbf{a}}$$

This is a discrete Charnes–Cooper transformation !

The separation sub-problem is :

$$\min 1 - \mathbf{x}^\top \mathbf{a}$$

$$a_i + a_j \leq 1, \quad \forall \{i, j\} \in E$$

$$a_i \in \{0, 1\} \quad \forall i \in [1..n]$$

$$t^* = \min_{\mathbf{a}} \frac{1 - \mathbf{x}^\top \mathbf{a}}{\mathbf{d}^\top \mathbf{a}}$$

$$\mathbf{d}^\top \mathbf{a} > 0$$

$$\text{STAB} \begin{cases} a_i + a_j \leq 1, & \forall \{i, j\} \in E \\ a_i \in \{0, 1\} & \forall i \in [1..n] \end{cases}$$

$$t^* = \min_{\mathbf{a}, \bar{\mathbf{a}}} \bar{\alpha} - \mathbf{x}^\top \bar{\mathbf{a}}$$

$$\bar{a}_i + \bar{a}_j \leq \bar{\alpha} \quad \forall \{i, j\} \in E$$

$$\mathbf{d}^\top \bar{\mathbf{a}} = 1$$

$$\bar{a}_i \in \{0, \bar{\alpha}\} \quad \forall i \in [1..n]$$

$$\bar{\alpha} \geq 0$$

$$\bar{\mathbf{a}} = \frac{\mathbf{a}}{\mathbf{d}^\top \mathbf{a}} \text{ and } \bar{\alpha} = \frac{1}{\mathbf{d}^\top \mathbf{a}}$$

This is a discrete Charnes–Cooper transformation !

The separation sub-problem is :

$$\min 1 - \mathbf{x}^\top \mathbf{a}$$

$$a_i + a_j \leq 1, \quad \forall \{i, j\} \in E$$

$$a_i \in \{0, 1\} \quad \forall i \in [1..n]$$

$$t^* = \min_{\mathbf{a}} \frac{1 - \mathbf{x}^\top \mathbf{a}}{\mathbf{d}^\top \mathbf{a}}$$

$$\mathbf{d}^\top \mathbf{a} > 0$$

$$\text{STAB} \begin{cases} a_i + a_j \leq 1, & \forall \{i, j\} \in E \\ a_i \in \{0, 1\} & \forall i \in [1..n] \end{cases}$$

$$t^* = \min_{\mathbf{a}, \bar{\mathbf{a}}} \bar{\alpha} - \mathbf{x}^\top \bar{\mathbf{a}}$$

$$\bar{a}_i + \bar{a}_j \leq \bar{\alpha} \quad \forall \{i, j\} \in E$$

$$\mathbf{d}^\top \bar{\mathbf{a}} = 1$$

$$\bar{a}_i \in \{0, \bar{\alpha}\} \quad \forall i \in [1..n]$$

$$\bar{\alpha} \geq 0$$

$$\bar{\mathbf{a}} = \frac{\mathbf{a}}{\mathbf{d}^\top \mathbf{a}} \quad \text{and} \quad \bar{\alpha} = \frac{1}{\mathbf{d}^\top \mathbf{a}}$$

This is a discrete Charnes–Cooper transformation !

The separation sub-problem is :

$$\min 1 - \mathbf{x}^\top \mathbf{a}$$

$$a_i + a_j \leq 1, \quad \forall \{i, j\} \in E$$

$$a_i \in \{0, 1\} \quad \forall i \in [1..n]$$

The continuity-breaking
constraints remain similar.

$$t^* = \min_{\mathbf{a}} \frac{1 - \mathbf{x}^\top \mathbf{a}}{\mathbf{d}^\top \mathbf{a}}$$

$$\mathbf{d}^\top \mathbf{a} > 0$$

$$\text{STAB} \begin{cases} a_i + a_j \leq 1, & \forall \{i, j\} \in E \\ a_i \in \{0, 1\} & \forall i \in [1..n] \end{cases}$$

$$t^* = \min_{\mathbf{a}, \bar{\mathbf{a}}} \bar{\alpha} - \mathbf{x}^\top \bar{\mathbf{a}}$$

$$\bar{a}_i + \bar{a}_j \leq \bar{\alpha} \quad \forall \{i, j\} \in E$$

$$\mathbf{d}^\top \bar{\mathbf{a}} = 1$$

$$\bar{a}_i \in \{0, \bar{\alpha}\} \quad \forall i \in [1..n]$$

$$\bar{\alpha} \geq 0$$

$$\bar{\mathbf{a}} = \frac{\mathbf{a}}{\mathbf{d}^\top \mathbf{a}} \text{ and } \bar{\alpha} = \frac{1}{\mathbf{d}^\top \mathbf{a}}$$

This is a discrete Charnes–Cooper transformation !

The separation sub-problem is :

$$\min 1 - \mathbf{x}^\top \mathbf{a}$$

$$a_i + a_j \leq 1, \quad \forall \{i, j\} \in E$$

$$a_i \in \{0, 1\} \quad \forall i \in [1..n]$$

The edge inequalities could have been replaced by

any others imaginable

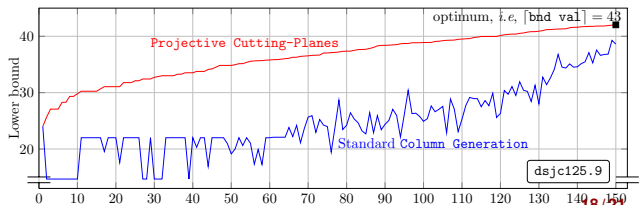
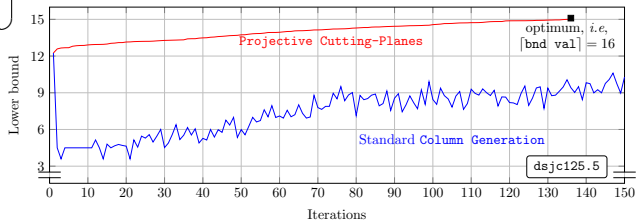
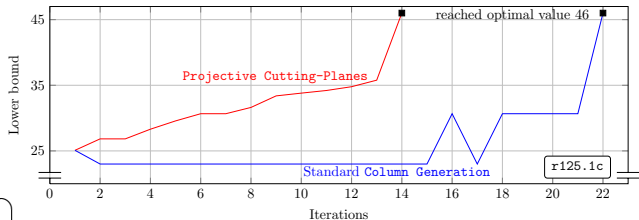
Each interior point \mathbf{x}_k yields a lower bound

We used

$$\mathbf{x}_k = \mathbf{x}_{k-1} + t_{k-1}^* \mathbf{d}_{k-1}$$

so that \mathbf{x}_k = the last contact point

For Column Generation, we used Lagrangian bounds



Further results on graph coloring

Each first row in black : the standard method
Each second row in red : the new method.

Three lower bounds found along the search

instance	beginning	mid iteration	last iteration
	iter:lb/time	iter:lb/time	iter:lb/time
r125.1	34 :2.35/0.06	36 :2.62/0.06	47 :5/0.08
	5 :2.35/0.18	17 :2.61/0.68	20 :5/0.80
dsjc125.5	253 :13.08/365	306 :14.27/634	378 :15.08/1101
	16 :13.04/213	62 :14.001/1288	136 :15.003/4077
dsjc125.9	70 :25.67/24.4	134 :34.13/78	171 :42.11/136
	2 :25.67/7.3	44 :34.11/109	150 :42.03/486

Plan

- 1 The general idea of projecting instead of separating
- 2 Linear robust optimization
- 3 The Benders reformulation
- 4 Column Generation for graph coloring
- 5 **Conclusions**

We proposed Projective Cutting-Planes :

- 1 The driving force is a sequence of inner solutions (a bit like in IPM) that are not available by default in standard Cutting-Planes
 - If one wants to calculate inner solutions during a standard Cutting-Planes, it may be possible but : 1) one has to apply some ad-hoc method and 2) such inner solutions will always remain a by-product of the algorithm
- 2 The lower bounds of the new method are monotonically increasing over the iterations : the infamous “yo-yo” effect is finished

-
- The new method discovered a new lower bound for a very well studied graph `dsjc250.1`
 - On the robust optimization problem, the new method is faster even in terms of CPU time

We proposed Projective Cutting-Planes :

- 1 The driving force is a sequence of inner solutions (a bit like in IPM) that are not available by default in standard Cutting-Planes
 - If one wants to calculate inner solutions during a standard Cutting-Planes, it may be possible but : 1) one has to apply some ad-hoc method and 2) such inner solutions will always remain a by-product of the algorithm
- 2 The lower bounds of the new method are monotonically increasing over the iterations : the infamous “yo-yo” effect is finished

-
- The new method discovered a new lower bound for a very well studied graph `dsjc250.1`
 - On the robust optimization problem, the new method is faster even in terms of CPU time

We proposed Projective Cutting-Planes :

- 1 The driving force is a sequence of inner solutions (a bit like in IPM) that are not available by default in standard Cutting-Planes
 - If one wants to calculate inner solutions during a standard Cutting-Planes, it may be possible but : 1) one has to apply some ad-hoc method and 2) such inner solutions will always remain a by-product of the algorithm
- 2 The lower bounds of the new method are monotonically increasing over the iterations : the infamous “yo-yo” effect is finished

-
- The new method discovered a new lower bound for a very well studied graph `dsjc250.1`
 - On the robust optimization problem, the new method is faster even in terms of CPU time