SÉMINAIRE ENSTA

December, 11th 2017

Semi-infinite Programming

The Blankenship-Mitsos-Djelassi Strategy

G. Chabert

Outline

SIP

Find

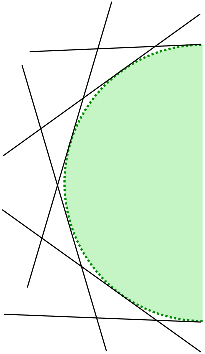
$$f^* := \min_x f(x) \quad \text{s.t.} \quad g(x,y) \leq 0, \quad \forall y \in \mathcal{Y}$$

where $\mathcal{Y}$ is infinite.

Example :

$$g(x,y) = \cos(y)x_1 + \sin(y)x_2 - 1$$

$$\mathcal{Y} = [0, 2\pi]$$



A SIP problem is equivalent to a NLP problem with a *max* inequality constraint :

SIP (min-max formulation)

Find

$$f^* := \min_x f(x) \quad \text{s.t.} \quad g^*(x) \leq 0$$

with

$$g^*(x) = \max_{y \in \mathcal{Y}} g(x, y).$$

The SIP strategy presented here is not a branch & prune strategy.

It is an iterative scheme that calculates a sequence of lower bounds (f^-) and upper bounds (f^+ and argmin) until

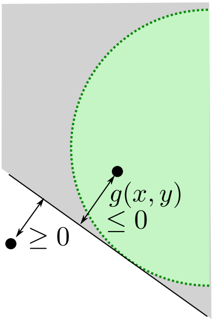
$$f^+ - f^- < \varepsilon_f.$$

However, it resorts to a global NLP solver at each iteration.

We assume the precision ϵ_{NLP} of the NLP solver satisfies

$$\epsilon_{NLP} \ll \varepsilon_f.$$

In our graphical examples, for each y, we assume that $g(x, y)$ is the signed distance between x and $g(., y) = 0$.

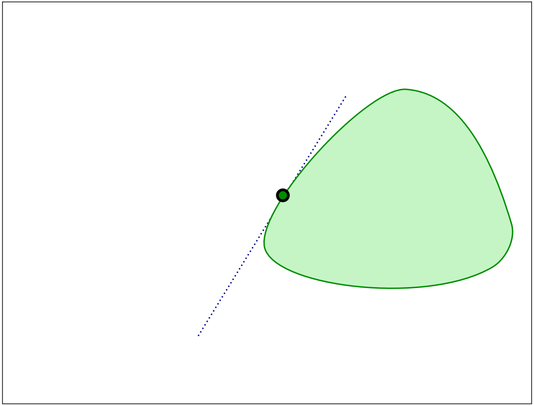


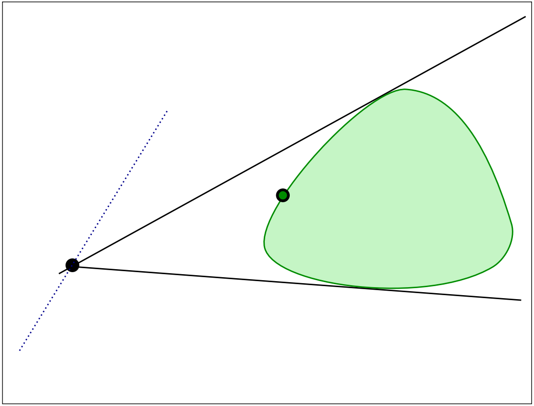
The iteration is based on a discretization of $\mathcal{Y}$ noted $\mathcal{Y}_D$ that is populated during solving.

Outline

Lower bounding is based on a technique by Blakenship which :

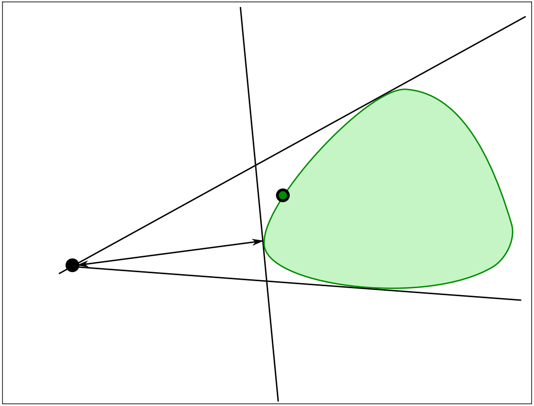
1. run the NLP solver to minimize f with $y \in \mathcal{Y}_D$
2. update the lower bound f^- (if better)
3. add into $\mathcal{Y}_D$ the value of y that maximizes the violation at the point found.





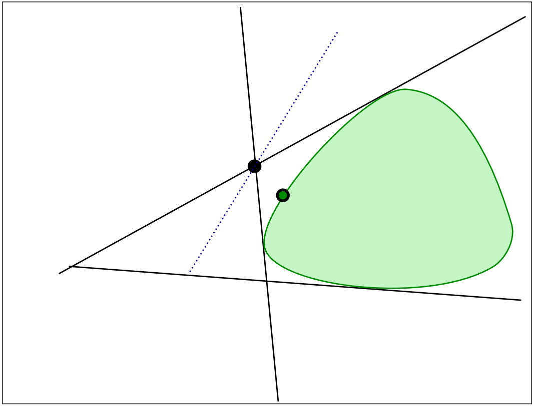
LBD problem

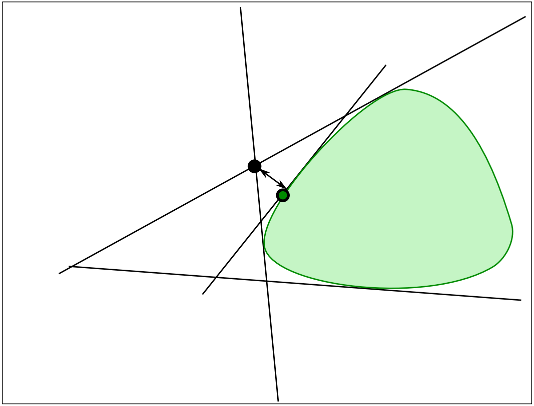
$$f_{LBD} := \min_x f(x) \quad \text{s.t.} \quad g(x,y) \leq 0, \quad \forall y \in \mathcal{Y}_D$$

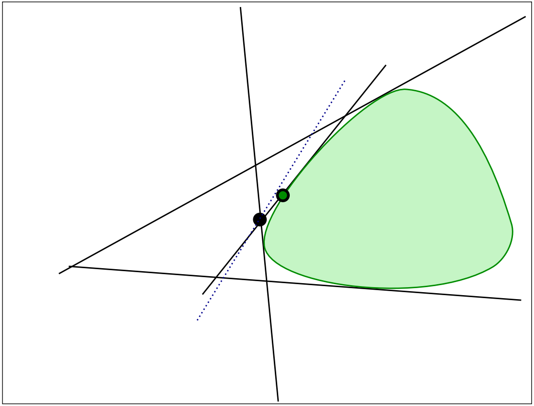


LLP problem

$$g_{LLP} := \max_{y \in \mathcal{Y}} g(x, y)$$







Note : If

$$g_{LLP}^{-} < 0 < g_{LLP}^{+}$$

nothing can be done and the LLP problem will have to be solved again for x_{LBD}, with a higher accuracy.

However, not immediatly (an upper bounding step is performed first). We cannot loop because $g^*(x_{LBD})$ may actually be arbitrarily close to 0.

The precision of the NLP solver when solving LLP cannot be set a priori : the sequence of $g^*(x_{LBD})$ is not monotonic and it would anyway require sensitivity analysis at the optimum.

The convergence of f_{LBD} to the expected value (i.e., $f^* - \epsilon_f$ or greater) is not straightfoward.

If we denote by x_k the sequence of points x_{LBD}, then either x_k turns to be SIP-feasible for one k and it's done, or we have to prove that

$$g^*(x_k) \to 0.$$

which is not true in general. Some compacity/continuity assumptions are required.

Note that the sequence $g^*(x_k)$ is not necessarily monotonously decreasing, in any case.

The main argument behind the proof is the following :

If we denote y_k the argmax of the (first) *LLP* problem successfully solved for x_k then

$$g(x_k, y_k) > 0 \tag{1}$$

and, by construction,

$$\forall k, \quad \forall l > k, \quad g(x_l, y_k) \leq 0. \tag{2}$$

If we now assume that the domain for x is compact, we can extract a converging sub-sequence of x_k. And if we further assume that g is uniformly continuous, we can then deduce from (1) and (2) that

$$g(x_k, y_k) \to 0$$

which means that the limit point x_k is SIP-feasible.

So $\mathcal{Y}$ has to be populated in a specific way.

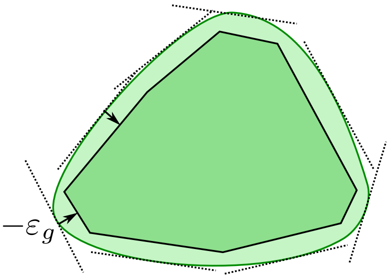
Otherwise, convergence is lost.

Outline

The principle is to consider a *restriction of the relaxation*.

UBD problem

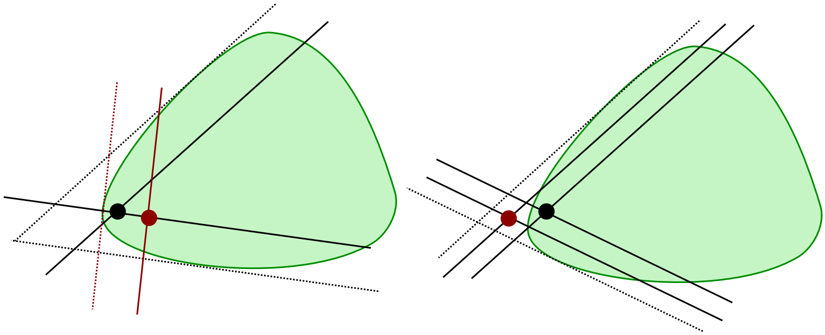
$$f_{UBD} := \min_x f(x) \quad \text{s.t.} \quad g(x,y) \le -\varepsilon_g, \quad \forall y \in \mathcal{Y}_D$$



The idea is to get an upper bound of f^* of the required precision, by decreasing ε_g and populating $\mathcal{Y}$.

The difficulty is that :

- for a fixed ε_g we deteriorate the criterion by populating $\mathcal{Y}$
- for a fixed $\mathcal{Y}$ we may lose the SIP-feasibility by decreasing ε_g.



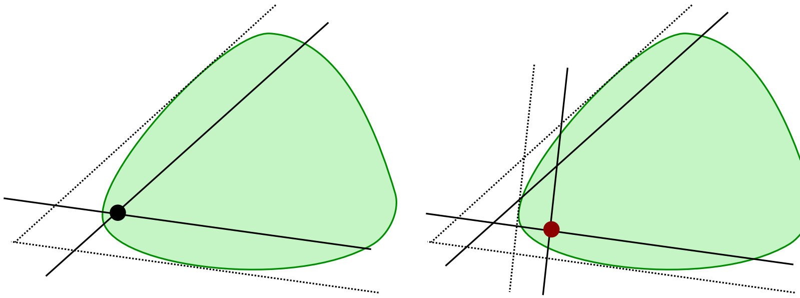
Still, if we keep on decreasing ε_g and populating $\mathcal{Y}$ (in the appropriate way) the process eventually succeed.

More precisely, a strategy proven to converge is the following.

- We solve the UBD program for a given ε_g and $\mathcal{Y}$ and obtain x_{UBD}.
- If it is infeasible, ε_g is decreased.
- Otherwise, we solve the LLP program for x_{UBD}.
  - if $g^+_{LLP} \leq 0$, x_{UBD} is SIP-feasible, f^+ is updated and ε_g is decreased
  - otherwise we add y^*_{LLP} in $\mathcal{Y}$

Note : Contrary to *LBD*, the precision ϵ_{LLP} required for LLP can be fixed here a priori to any value $< \varepsilon_g$ (we don't need to handle the case $g^-_{LLP} < 0 < g^+_{LLP}$) !

However, the sequence of valid upper bounds is not monotonous (and, contrary to LBD, this is regardless of the precision of the NLP solver).



Convergence is even less trivial and requires an additionnal assumption (existence of a Slater point).

Like before, we don't know a priori the final value of ε_g for a given ε_f.

In practice, we are faced to the following dilemma :

- Decreasing ε_g too slowly leads to poor convergence of the overwhole iteration
- Decreasing ε_g too quickly leads to a dense popuation of $\mathcal{Y}$ and a lack of upper bounds

Outline

The oracle problem allows to boost convergence.

The key idea is to fix a guess f_{ORA} for the objective, e.g. :

$$f_{ORA} = \frac{1}{2}\Big(f_{LBD} + f_{UBD}\Big)$$

and to look for the point x that minimizes the violation η of the constraints inside the current discretization $\mathcal{Y}$ while satisfying

$$f(x) \leq f_{ORA}.$$

ORA problem

$$f_{ORA} := \min_{x,\eta} \ \eta \quad \text{s.t.} \quad \begin{cases} f(x) \leq f_{ORA} \\ g(x,y) \leq \eta, \quad \forall y \in \mathcal{Y}_D \end{cases}$$

ORA problem

$$f_{ORA} := \min_{x,\eta} \ \eta \quad \text{s.t.} \quad \begin{cases} f(x) \le f_{ORA} \\ g(x,y) \le \eta, \quad \forall y \in \mathcal{Y}_D \end{cases}$$

- If the minimum η^* is positive, f_{ORA} is a valid lower bound !
- If the minimum η^* is negative and x_{ORA} is SIP-feasible, then f_{ORA} is a valid upper bound.
- Better, in case of SIP-feasibility, ε_g of UBD can be set η^*

Note : solving UBD with $\varepsilon_g = \eta^*$ can give a better bound than f_{ORA}, at least in theory.

Outline

The Blankenship approach replaces the SIP problem by a sequence of (increasingly strengthened) relaxations.

This approach could also be used as such for upper bounding as it eventually provides SIP-feasible points (in general). Indeed, looking for the argmin of UBD already introduces a *restriction of the relaxation*.

But this strategy would over-populate $\mathcal{Y}$.

The Mitsos upper-bouding strategy alleviates this phenomenon by over-restricting the relaxation.

The oracle introduces a dichotomic principle in this approach. But the dimension of the subproblem is increased.

Thanks !