Optimality-Based Discretization Methods for the Global Optimization of Nonconvex Semi-Infinite Programs

Preprint: arXiv:2303.00219

Evren M. Turan

Norwegian University of Science and Technology

PanOptiC View on Global Optimization, March 10, 2023

Joint work with: Johannes Jäschke (NTNU),
Rohit Kannan (LANL)

Funding: Norwegian Research Council, Center for Nonlinear Studies at LANL

Semi-Infinite Programs (SIPs)

SIP: finite number of variables, *infinite* number of constraints

Design Centering

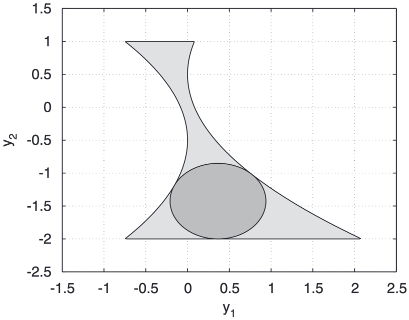
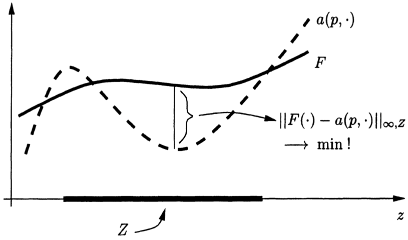


Fig. 1. A disk $B(x^\star)$ with maximal area in a container C.

Chebyshev Approximation



Other applications: adversarial ML, robust optimization, model reduction, ...

Formulation

Want *global* solutions to:

$$\begin{aligned} \min_{x \in X} \; & f(x) \\ \text{s.t.} \; & g(x, y) \le 0, \quad \forall y \in Y \end{aligned} \tag{SIP}$$

- nonempty compact sets $X \subset \mathbb{R}^{n_x}$ and $Y \subset \mathbb{R}^{n_y}$
- $f : \mathbb{R}^{n_x} \to \mathbb{R}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$ are continuous
- no convexity/concavity assumptions on f, g, X, and Y

Formulation

Want *global* solutions to:

$$\min_{x \in X} \; f(x) \qquad \text{(SIP)}$$
$$\text{s.t. } g(x, y) \leq 0, \quad \forall y \in Y$$

- nonempty compact sets $X \subset \mathbb{R}^{n_x}$ and $Y \subset \mathbb{R}^{n_y}$
- $f : \mathbb{R}^{n_x} \to \mathbb{R}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$ are continuous
- no convexity/concavity assumptions on f, g, X, and Y

Challenge: checking feasibility requires *global* solution of the lower-level problem

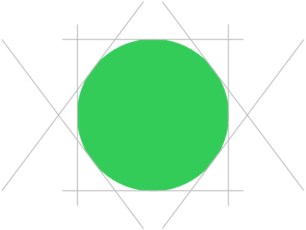
$$G(x) := \max_{y \in Y} \; g(x, y) \qquad \text{(LLP)}$$

- $\bar{x} \in X$ feasible $\iff$ $G(\bar{x}) \leq 0$

Simple Examples of Semi-Infinite Constraints

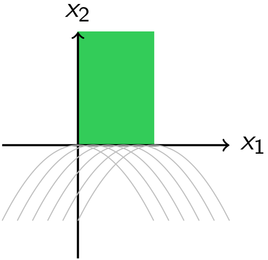
A ball as a semi-infinite constraint

$$x_1^2 + x_2^2 \leq 1,$$
$$\iff \sum_{i=1}^{2} 2y_i(x_i - y_i) \leq 0, \ \forall y \text{ s.t. } \|y\| = 1$$



Another semi-infinite constraint

$$x_1 \in [0, 1],$$
$$x_2 \in [-1000, 1000],$$
$$x_2 \geq -(x_1 - y)^2, \ \forall y \in [-1, 1]$$



Discretization-Based Lower Bounds

Consider an iteratively refined finite subset $Y_k \subsetneq Y_{k+1} \subsetneq Y$ (Kelley, 1960; Blankenship and Falk, 1976)

$$\begin{aligned} \min_{x \in X} \ & f(x) \\ \text{s.t. } & g(x,y) \leq 0, \quad \forall y \in Y_k \end{aligned} \tag{LBP}$$

- Can be solved directly using off-the-shelf global solvers
- Under mild assumptions every optimum of a sequence of discretizations (LBP) converges to an optimum of (SIP)

Discretization-Based Lower Bounds

Consider an iteratively refined finite subset $Y_k \subsetneq Y_{k+1} \subsetneq Y$ (Kelley, 1960; Blankenship and Falk, 1976)

$$\begin{aligned} \min_{x \in X} \ & f(x) \\ \text{s.t.} \ & g(x,y) \leq 0, \quad \forall y \in Y_k \end{aligned} \tag{LBP}$$

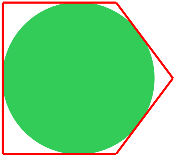
- Can be solved directly using off-the-shelf global solvers
- Under mild assumptions every optimum of a sequence of discretizations (LBP) converges to an optimum of (SIP)
- Many adaptations and generalizations (Djelassi et al., 2021)

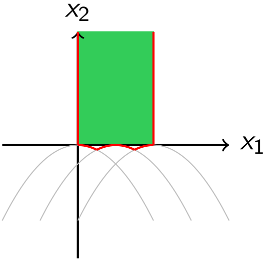
Discretization-Based Lower Bounds

Consider an iteratively refined finite subset $Y_k \subsetneq Y_{k+1} \subsetneq Y$ (Kelley, 1960; Blankenship and Falk, 1976)

$$\begin{aligned} \min_{x \in X} \ & f(x) \\ \text{s.t.} \ & g(x, y) \leq 0, \quad \forall y \in Y_k \end{aligned} \tag{LBP}$$

- Can be solved directly using off-the-shelf global solvers
- Under mild assumptions every optimum of a sequence of discretizations (LBP) converges to an optimum of (SIP)
- Many adaptations and generalizations (Djelassi et al., 2021)





How to Populate the Discretization Y_k?

Assume we are given discretization Y_{k-1} from iteration $k-1$ with corresponding (LBP) solution $x^k \in X$

How to Populate the Discretization Y_k?

Assume we are given discretization Y_{k-1} from iteration $k-1$ with corresponding (LBP) solution $x^k \in X$

- Conventional Approach: Add $y^k \in Y$ to Y_{k-1} to exclude x^k (based on feasibility arguments), e.g., by solving

$$y^k \in \arg\max_{y \in Y} \; g(x^k, y) \qquad \text{(LLP)}$$

How to Populate the Discretization Y_k?

Assume we are given discretization Y_{k-1} from iteration $k-1$ with corresponding (LBP) solution $x^k \in X$

- Conventional Approach: Add $y^k \in Y$ to Y_{k-1} to exclude x^k (based on feasibility arguments), e.g., by solving

$$y^k \in \underset{y \in Y}{\arg\max}\ g(x^k, y) \tag{LLP}$$

- Our Approach: Add $y^k \in Y$ to Y_{k-1} that results in the highest lower bound (based on bound improvement)

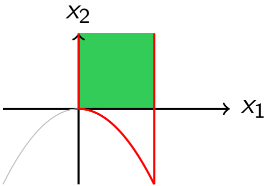
How to Populate the Discretization Y_k?

Assume we are given discretization Y_{k-1} from iteration $k-1$ with corresponding (LBP) solution $x^k \in X$

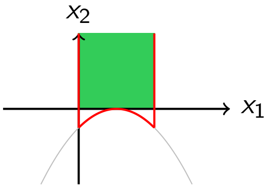
- Conventional Approach: Add $y^k \in Y$ to Y_{k-1} to exclude x^k (based on feasibility arguments), e.g., by solving

$$y^k \in \arg\max_{y \in Y} \; g(x^k, y) \tag{LLP}$$

- Our Approach: Add $y^k \in Y$ to Y_{k-1} that results in the highest lower bound (based on bound improvement)
  We directly optimize the discretization for the best bound!



$\downarrow$
$\min x_2$



Conventional approach: $LBD = -1$

Our approach: $LBD = -0.25$

The Blankenship and Falk (1976) Algorithm

A Feasibility-Based Discretization Method

Recall

$$\min_{x \in X} \; f(x) \qquad \text{(LBP)}$$
$$\text{s.t. } g(x,y) \le 0, \quad \forall y \in Y_{k-1}$$

$$G(x) := \max_{y \in Y} g(x,y) \qquad \text{(LLP)}$$

The Blankenship and Falk (1976) Algorithm

A Feasibility-Based Discretization Method

Recall
$$\min_{x \in X} \; f(x) \quad \text{s.t.} \; g(x,y) \le 0, \quad \forall y \in Y_{k-1} \tag{LBP}$$

$$G(x) := \max_{y \in Y} \; g(x,y) \tag{LLP}$$

1. **Initialize**: $Y_0 = \emptyset$, $k = 1$.
2. **Solve** (LBP) to obtain candidate solution $x^k \in X$.
3. **Solve** (LLP) at $x = x^k$ to obtain solution $y^k \in Y$ and $G(x^k)$.
4. **If** $G(x^k) > 0$: $Y_k \leftarrow Y_{k-1} \cup \{y^k\}$, $k \leftarrow k + 1$ goto 2.

   Else: terminate with x^k as a solution to (SIP).

The Blankenship and Falk (1976) Algorithm

A Feasibility-Based Discretization Method

Recall
$$\min_{x \in X} f(x) \quad \text{s.t. } g(x,y) \le 0, \quad \forall y \in Y_{k-1} \tag{LBP}$$

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

1. **Initialize**: $Y_0 = \emptyset$, $k = 1$.
2. **Solve** (LBP) to obtain candidate solution $x^k \in X$.
3. **Solve** (LLP) at $x = x^k$ to obtain solution $y^k \in Y$ and $G(x^k)$.
4. **If** $G(x^k) > 0$: $Y_k \leftarrow Y_{k-1} \cup \{y^k\}$, $k \leftarrow k+1$ goto 2.

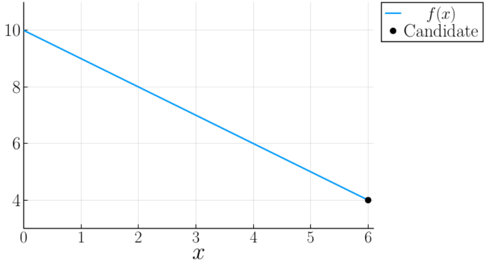
   Else: terminate with x^k as a solution to (SIP).

- Discretization updated using (LLP) solution (feasibility cuts)
- Two global solves per iteration in general
- "Workhorse" bounding method for global optimization of SIPs

The Blankenship and Falk (1976) Algorithm

A Feasibility-Based Discretization Method

1. **Initialize**: $Y_0 = \emptyset$, $k = 1$.
2. **Solve** (LBP) to obtain candidate solution $x^k \in X$.
3. **Solve** (LLP) at $x = x^k$ to obtain solution $y^k \in Y$ and $G(x^k)$.
4. **If** $G(x^k) > 0$: $Y_k \leftarrow Y_{k-1} \cup \{y^k\}$, $k \leftarrow k + 1$ goto 2.
   Else: terminate with x^k as a solution to (SIP).



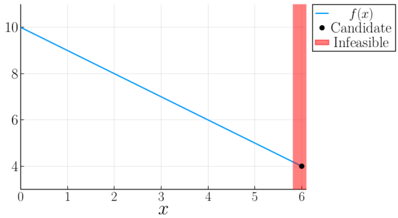
PROBLEM DP from Mitsos (2009)

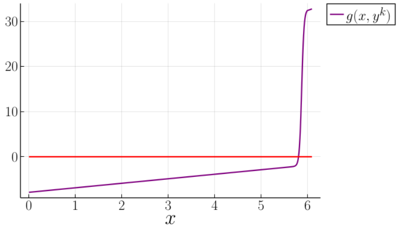
$$X = [0,6],\ Y = [2,6],\ f(x) = 10 - x,\ g(x,y) = \frac{y^2}{1 + \exp(-40(x-y))} + x - y - 2$$

The Blankenship and Falk (1976) Algorithm

A Feasibility-Based Discretization Method

1. **Initialize**: $Y_0 = \emptyset$, $k = 1$.
2. **Solve** (LBP) to obtain candidate solution $x^k \in X$.
3. **Solve** (LLP) at $x = x^k$ to obtain solution $y^k \in Y$ and $G(x^k)$.
4. **If** $G(x^k) > 0$: $Y_k \leftarrow Y_{k-1} \cup \{y^k\}$, $k \leftarrow k+1$ goto 2.
   Else: terminate with x^k as a solution to (SIP).





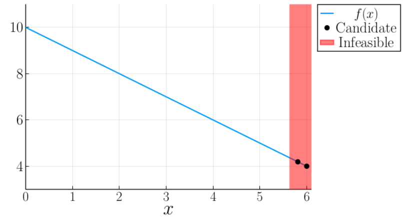
PROBLEM DP from Mitsos (2009)

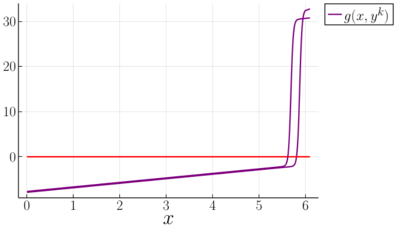
$$X = [0,6],\ Y = [2,6],\ f(x) = 10 - x,\ g(x,y) = \frac{y^2}{1+\exp(-40(x-y))} + x - y - 2$$

The Blankenship and Falk (1976) Algorithm

A Feasibility-Based Discretization Method

1. **Initialize**: $Y_0 = \emptyset$, $k = 1$.
2. **Solve** (LBP) to obtain candidate solution $x^k \in X$.
3. **Solve** (LLP) at $x = x^k$ to obtain solution $y^k \in Y$ and $G(x^k)$.
4. **If** $G(x^k) > 0$: $Y_k \leftarrow Y_{k-1} \cup \{y^k\}$, $k \leftarrow k + 1$ goto 2.
   Else: terminate with x^k as a solution to (SIP).





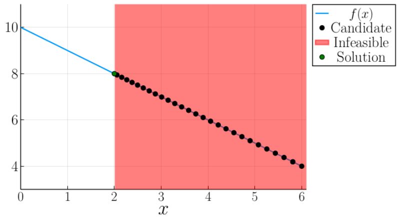
PROBLEM DP from Mitsos (2009)

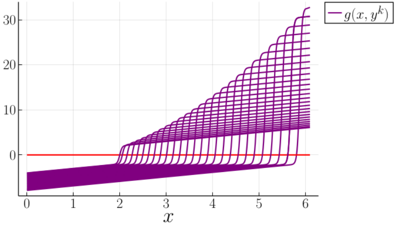
$$X = [0, 6],\ Y = [2, 6],\ f(x) = 10 - x,\ g(x, y) = \frac{y^2}{1 + \exp(-40(x - y))} + x - y - 2$$

The Blankenship and Falk (1976) Algorithm

A Feasibility-Based Discretization Method

1. **Initialize**: $Y_0 = \emptyset$, $k = 1$.
2. **Solve** (LBP) to obtain candidate solution $x^k \in X$.
3. **Solve** (LLP) at $x = x^k$ to obtain solution $y^k \in Y$ and $G(x^k)$.
4. **If** $G(x^k) > 0$: $Y_k \leftarrow Y_{k-1} \cup \{y^k\}$, $k \leftarrow k+1$ goto 2.
   Else: terminate with x^k as a solution to (SIP).





`PROBLEM DP` from Mitsos (2009)

$$X = [0,6],\ Y = [2,6],\ f(x) = 10 - x,\ g(x,y) = \frac{y^2}{1+\exp(-40(x-y))} + x - y - 2$$

Optimality-Based Discretization Methods

- BF: add $y^k \in Y$ corresponding to greatest constraint violation
- Our Idea: add $y^k \in Y$ that results in the highest lower bound

Optimality-Based Discretization Methods

- BF: add $y^k \in Y$ corresponding to greatest constraint violation
- Our Idea: add $y^k \in Y$ that results in the highest lower bound
- At the first iteration we solve the max-min problem:

$$\bar{y}^1 \in \arg\max_{y \in Y} \overbrace{\min_{x \in X} \ f(x)}^{\phi_1(y)} \qquad \text{(max-min)}$$
$$\text{s.t. } g(x, y) \leq 0$$

Optimality-Based Discretization Methods

- BF: add $y^k \in Y$ corresponding to greatest constraint violation
- Our Idea: add $y^k \in Y$ that results in the highest lower bound
- At the first iteration we solve the max-min problem:

$$\bar{y}^1 \in \arg\max_{y \in Y} \overbrace{\min_{x \in X} \; f(x)}^{\phi_1(y)} \qquad \text{(max-min)}$$
$$\text{s.t. } g(x, y) \leq 0$$

- Solving to local optimality is sufficient for valid discretization

Optimality-Based Discretization Methods

- BF: add $y^k \in Y$ corresponding to greatest constraint violation
- Our Idea: add $y^k \in Y$ that results in the highest lower bound
- At the first iteration we solve the max-min problem:

$$\bar{y}^1 \in \arg\max_{y \in Y} \overbrace{\min_{x \in X} \; f(x)}^{\phi_1(y)} \quad \text{s.t. } g(x,y) \le 0 \qquad \text{(max-min)}$$

- Solving to local optimality is sufficient for valid discretization
- Compute $\nabla \phi_1$ using sensitivity theory

Basic sensitivity theory

- Parametric NLP: f, g twice differentiable in x, once in p

$$\nu^*(p) = \min_x \quad f(x)$$
$$g(w,p) = 0$$

- $x^*(p)$ and $\lambda^*(p)$ implicitly given by KKT conditions:

$$\nabla_x \mathcal{L}(x, \lambda, p) = 0$$
$$g(x,p) = 0$$

Basic sensitivity theory

- Parametric NLP: f, g twice differentiable in x, once in p

$$\nu^*(p) = \min_x \quad f(x)$$
$$g(w,p) = 0$$

- $x^*(p)$ and $\lambda^*(p)$ implicitly given by KKT conditions:

$$\nabla_x \mathcal{L}(x,\lambda,p) = 0$$
$$g(x,p) = 0$$

$$\begin{bmatrix} \nabla_x^2 L(x^*,\lambda^*,p) & \nabla_x g(x^*,p) \\ \nabla_x g(x^*,p)^T & 0 \end{bmatrix} \begin{bmatrix} \nabla_p x^* \\ \nabla_p \lambda^* \end{bmatrix} = \begin{bmatrix} \nabla_{xp} L(x^*,\lambda^*,p) \\ \nabla_p g(x^*,p) \end{bmatrix}$$

Basic sensitivity theory

- Parametric NLP: f, g twice differentiable in x, once in p

$$\nu^*(p) = \min_x \quad f(x)$$
$$g(w,p) = 0$$

- $x^*(p)$ and $\lambda^*(p)$ implicitly given by KKT conditions:

$$\nabla_x \mathcal{L}(x,\lambda,p) = 0$$
$$g(x,p) = 0$$

$$\begin{bmatrix} \nabla_x^2 L(x^*,\lambda^*,p) & \nabla_x g(x^*,p) \\ \nabla_x g(x^*,p)^T & 0 \end{bmatrix} \begin{bmatrix} \nabla_p x^* \\ \nabla_p \lambda^* \end{bmatrix} = \begin{bmatrix} \nabla_{xp} L(x^*,\lambda^*,p) \\ \nabla_p g(x^*,p) \end{bmatrix}$$

- $x^*(p)$ is well defined and $\mathcal{C}^1$ in a neighbourhood of p if solution is LICQ and SOSC (Fiacco, 1983).

Basic sensitivity theory

- Parametric NLP: f, g twice differentiable in x, once in p

$$\nu^*(p) = \min_x \quad f(x)$$
$$g(w, p) = 0$$

- $x^*(p)$ and $\lambda^*(p)$ implicitly given by KKT conditions:

$$\nabla_x \mathcal{L}(x, \lambda, p) = 0$$
$$g(x, p) = 0$$

$$\begin{bmatrix} \nabla_x^2 L(x^*, \lambda^*, p) & \nabla_x g(x^*, p) \\ \nabla_x g(x^*, p)^T & 0 \end{bmatrix} \begin{bmatrix} \nabla_p x^* \\ \nabla_p \lambda^* \end{bmatrix} = \begin{bmatrix} \nabla_{xp} L(x^*, \lambda^*, p) \\ \nabla_p g(x^*, p) \end{bmatrix}$$

- $x^*(p)$ is well defined and $\mathcal{C}^1$ in a neighbourhood of p if solution is LICQ and SOSC (Fiacco, 1983).
- kinks in $x^*(p)$ if weakly active inequalities
- results for generalized derivatives, continuity of $\nu^*(p)$, ...

Optimality-Based Discretization Methods

- BF: add $y^k \in Y$ corresponding to greatest constraint violation
- Our Idea: add $y^k \in Y$ that results in the highest lower bound
- At the first iteration we solve the max-min problem:

$$\bar{y}^1 \in \arg\max_{y \in Y} \overbrace{\min_{x \in X} \; f(x)}^{\phi_1(y)} \qquad \text{(max-min)}$$
$$\text{s.t. } g(x, y) \leq 0$$

- Solving to local optimality is sufficient for valid discretization
- Compute $\nabla \phi_1$ using sensitivity theory (treat y as parameters of the inner min problem) assuming (Fiacco, 1983; Still, 2018)
  - f and g are twice differentiable in x and once differentiable in y
  - LBD yields a KKT point satisfying constraint qualifications; if not, update discretization with the BF point y^1

Optimality-Based Discretization Methods

- At iteration 1 we solve the max-min problem

$$\bar{y}^1 \in \arg\max_{y^1 \in Y} \overbrace{\begin{aligned} \min_{x \in X} \quad & f(x) \\ \text{s.t.} \quad & g(x, y^1) \leq 0. \end{aligned}}^{\phi_1(y^1)} \qquad \text{(max-min)}$$

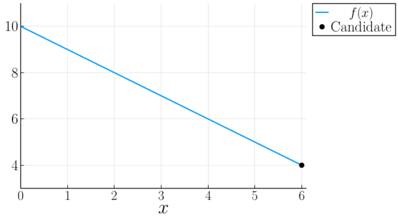
- Use $\nabla \phi_1$ within a bundle method for nonconvex optimization
- Initialize $\bar{y}^1$ with the BF point y^1

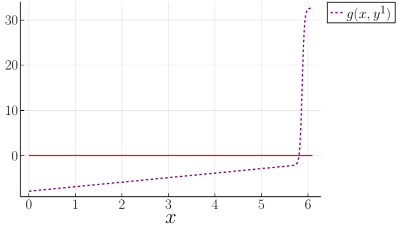
Optimality-Based Discretization Methods

- At iteration 1 we solve the max-min problem

$$\bar{y}^1 \in \arg\max_{y^1 \in Y} \overbrace{\min_{x \in X} \; f(x)}^{\phi_1(y^1)} \qquad \text{(max-min)}$$
$$\text{s.t. } g(x, y^1) \leq 0.$$

- Use $\nabla \phi_1$ within a bundle method for nonconvex optimization
- Initialize $\bar{y}^1$ with the BF point y^1



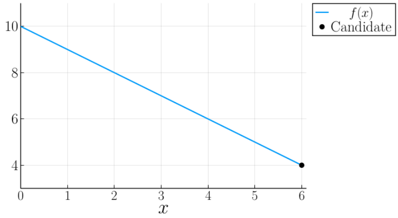


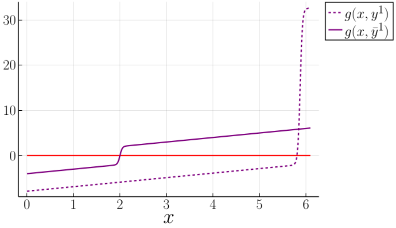
Optimality-Based Discretization Methods

- At iteration 1 we solve the max-min problem

$$\bar{y}^1 \in \arg\max_{y^1 \in Y} \overbrace{\min_{x \in X} \; f(x)}^{\phi_1(y^1)} \quad \text{s.t. } g(x, y^1) \leq 0. \qquad \text{(max-min)}$$

- Use $\nabla \phi_1$ within a bundle method for nonconvex optimization
- Initialize $\bar{y}^1$ with the BF point y^1



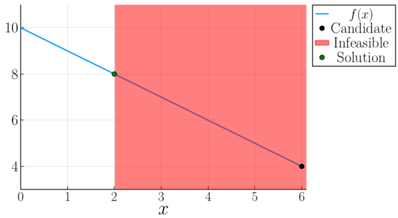


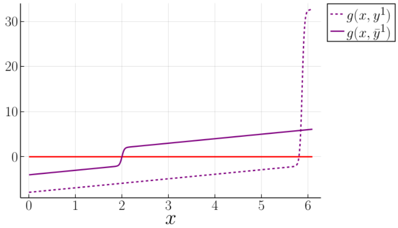
Optimality-Based Discretization Methods

- At iteration 1 we solve the max-min problem

$$\bar{y}^1 \in \arg\max_{y^1 \in Y} \overbrace{\min_{x \in X} \; f(x)}^{\phi_1(y^1)} \qquad \text{(max-min)}$$
$$\text{s.t. } g(x, y^1) \leq 0.$$

- Use $\nabla \phi_1$ within a bundle method for nonconvex optimization
- Initialize $\bar{y}^1$ with the BF point y^1
- PROBLEM DP now solves with a single discretization point!



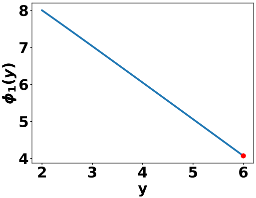


Optimality-Based Discretization Methods

- At iteration 1 we solve the max-min problem

$$\bar{y}^1 \in \arg\max_{y^1 \in Y} \overbrace{\min_{x \in X} \; f(x)}^{\phi_1(y^1)} \quad \text{s.t. } g(x, y^1) \le 0. \tag{max-min}$$

- Use $\nabla\phi_1$ within a bundle method for nonconvex optimization
- Initialize $\bar{y}^1$ with the BF point y^1
- `PROBLEM DP` now solves with a single discretization point!

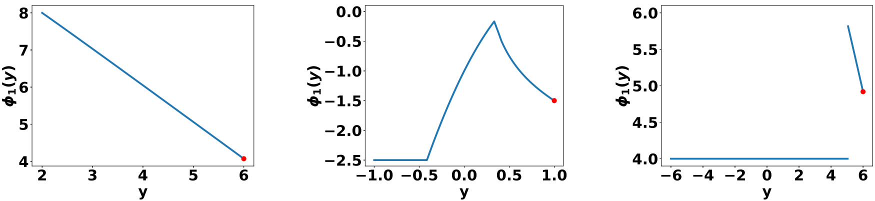


Optimality-Based Discretization Methods

- At iteration 1 we solve the max-min problem

$$\bar{y}^1 \in \arg\max_{y^1 \in Y} \overbrace{\min_{x \in X} \; f(x)}^{\phi_1(y^1)} \quad \text{s.t. } g(x, y^1) \leq 0. \qquad \text{(max-min)}$$

- Use $\nabla \phi_1$ within a bundle method for nonconvex optimization
- Initialize $\bar{y}^1$ with the BF point y^1
- PROBLEM DP now solves with a single discretization point!

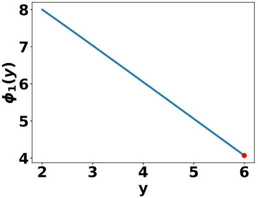


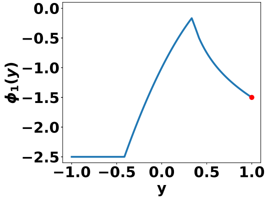
Optimality-Based Discretization Methods

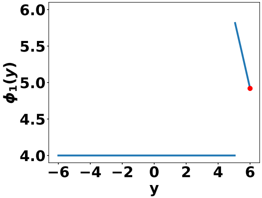
- At iteration k we solve the max-min problem (Alg: `GREEDY`)

$$\bar{y}^k \in \arg\max_{y^k \in Y} \overbrace{\min_{x \in X} \; f(x)}^{\phi_k(y^k)} \quad \text{s.t. } g(x,y) \le 0, \quad \forall y \in Y_{k-1}, \quad g(x,y^k) \le 0. \tag{max-min}$$

- Use $\nabla \phi_k$ within a bundle method for nonconvex optimization
- Initialize $\bar{y}^1$ with the BF point y^1
- `PROBLEM DP` now solves with a single discretization point!







Variants of the GREEDY Algorithm

- 2GREEDY: add two discretization points per iteration
  - First, add the BF point y^k to the discretization
  - Next, solve a max-min problem to find an additional discretization point $\bar{y}^k$
  - Motivation: might help reduce the number of global solves

Variants of the GREEDY Algorithm

- 2GREEDY: add two discretization points per iteration
  - First, add the BF point y^k to the discretization
  - Next, solve a max-min problem to find an additional discretization point $\bar{y}^k$
  - Motivation: might help reduce the number of global solves

- OPT: recompute entire discretization at each iteration

$$(\bar{y}^1, \bar{y}^2, \ldots, \bar{y}^k) \in \underset{(y^1, y^2, \ldots, y^k) \in Y^k}{\arg\max} \; \min_{x \in X} \; f(x) \quad \text{s.t. } g(x, y^i) \leq 0, \quad \forall i \in [k]$$

  - Can switch to GREEDY/2GREEDY after K iterations (HYBRID)

Variants of the GREEDY Algorithm

- 2GREEDY: add two discretization points per iteration
  - First, add the BF point y^k to the discretization
  - Next, solve a max-min problem to find an additional discretization point $\bar{y}^k$
  - Motivation: might help reduce the number of global solves
- OPT: recompute entire discretization at each iteration

$$(\bar{y}^1, \bar{y}^2, \ldots, \bar{y}^k) \in \underset{(y^1,y^2,\ldots,y^k) \in Y^k}{\arg\max} \; \min_{x \in X} \; f(x) \quad \text{s.t. } g(x, y^i) \leq 0, \quad \forall i \in [k]$$

  - Can switch to GREEDY/2GREEDY after K iterations (HYBRID)
- Similar assumptions and routines for computing sensitivities

Theoretical Results

Theorem (Convergence of LBD^k)

Suppose that a candidate discretization Y_d is determined using Algorithm `OPT`, `GREEDY`, `2GREEDY`, or `HYBRID`, and accepted if it improves LBD^k by $\delta > 0$. Then $\lim_{k\to\infty} LBD^k = v^*$.

Theoretical Results

Theorem (Convergence of LBD^k)

Suppose that a candidate discretization Y_d is determined using Algorithm `OPT`, `GREEDY`, `2GREEDY`, or `HYBRID`, and accepted if it improves LBD^k by $\delta > 0$. Then $\lim_{k \to \infty} LBD^k = v^*$.

Theorem (Convergence of `OPT` - convex SIP)

Suppose that (SIP) *is convex with respect to x, the max-min problem is solved to global optimality, and $\exists \bar{x} \in X$ such that $G(\bar{x}) < 0$. Then Algorithm `OPT` converges to v^* in at most d_x iterations.*

Theoretical Results

Theorem (Convergence of LBD^k)

Suppose that a candidate discretization Y_d is determined using Algorithm `OPT`, `GREEDY`, `2GREEDY`, or `HYBRID`, and accepted if it improves LBD^k by $\delta > 0$. Then $\lim_{k \to \infty} LBD^k = v^*$.

Theorem (Convergence of `OPT`)

Suppose $\{g(x, \cdot)\}_{x \in X}$ is uniformly Lipschitz continuous on Y with Lipschitz constant $L_{g,y} > 0$. If the max-min problem is solved to global optimality, then Algorithm `OPT` terminates with an ε_f-feasible point in at most $\left(\frac{\operatorname{diam}(Y) L_{g,y}}{2\varepsilon_f}\right)^{d_y}$ iterations.

Furthermore, if $\{g(\cdot, y)\}_{y \in Y}$ is uniformly Lipschitz continuous on X with Lipschitz constant $L_{g,x} > 0$, then Algorithm `OPT` terminates in at most $\min\left\{ \left(\frac{\operatorname{diam}(Y) L_{g,y}}{2\varepsilon_f}\right)^{d_y}, \left(\frac{\operatorname{diam}(X) L_{g,x}}{2\varepsilon_f} + 1\right)^{d_x} \right\}$ iterations.

Numerical Results: Small-Scale Instances

- Implemented in Julia 1.7.3, JuMP 1.3.1
- Global Solver: Baron 21.1.13, NLP Solver: Knitro 13.1.0
  LP Solvers: Gurobi 9.1.2 and CPLEX 22.1.0
  Bundle Solver: MPBNGC 2.0
- Absolute and Relative Optimality Tolerance: 10^{-3}

Numerical Results: Small-Scale Instances

- Implemented in Julia 1.7.3, JuMP 1.3.1
- Global Solver: Baron 21.1.13, NLP Solver: Knitro 13.1.0
  LP Solvers: Gurobi 9.1.2 and CPLEX 22.1.0
  Bundle Solver: MPBNGC 2.0
- Absolute and Relative Optimality Tolerance: 10^{-3}

Problem	n_x	n_y	BF # iter.	GREEDY	2GREEDY	HYBRID
				# iter. relative to BF		
Watson 2	2	1	2	+0%	+0%	+0%
Watson 5	3	1	5	-20%	-60%	-40%
Watson 6	2	1	3	-33%	-33%	-33%
Watson 7	3	2	2	+0%	+0%	+0%
Watson 8	6	2	15	-60%	-73%	-40%
Watson 9	6	2	9	-11%	-44%	-44%
Watson h	2	1	18	+61%	+6%	+61%
Watson n	2	1	3	+0%	-33%	+0%

- Instances from Watson (1983); all methods take $\sim$ 1 second

Numerical Results: Small-Scale Instances

- Implemented in Julia 1.7.3, JuMP 1.3.1
- Global Solver: Baron 21.1.13, NLP Solver: Knitro 13.1.0
  LP Solvers: Gurobi 9.1.2 and CPLEX 22.1.0
  Bundle Solver: MPBNGC 2.0
- Absolute and Relative Optimality Tolerance: 10^{-3}

Problem	n_x	n_y	BF # iter.	`GREEDY`	`2GREEDY`	`HYBRID`
				# iter. relative to BF		
Watson 2	2	1	2	+0%	+0%	+0%
Watson 5	3	1	5	-20%	-60%	-40%
Watson 6	2	1	3	-33%	-33%	-33%
Watson 7	3	2	2	+0%	+0%	+0%
Watson 8	6	2	15	-60%	-73%	-40%
Watson 9	6	2	9	-11%	-44%	-44%
Watson h	2	1	18	+61%	+6%	+61%
Watson n	2	1	3	+0%	-33%	+0%

- Instances from Watson (1983); all methods take $\sim$ 1 second
- `HYBRID` doesn't always outperform `GREEDY` in practice as max-min problem may get stuck at local maxima

Numerical Results: Small-Scale Instances

Problem	n_x	n_y	BF	GREEDY	2GREEDY	HYBRID
			# iter.	# iter. relative to BF		
Seidel	2	1	8	-75%	-75%	-75%
Tsoukalas	1	1	8	-38%	-25%	-38%
Mitsos 4_3	3	1	5	-20%	-60%	+20%
Mitsos 4_6	6	1	7	+14%	-14%	-29%
Mitsos DP	1	1	28	-93%	-93%	-93%

- Cases from Mitsos (2009); Tsoukalas and Rustem (2011); Seidel and Küfer (2022); all methods solve in ~ 1 second

Numerical Results: Large-Scale Instances

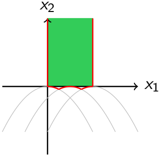
Problem	n_x	n_y	BF # iter.	GREEDY	2GREEDY	HYBRID
				# iter. relative to BF		
Cerulli PSD 1	21	5	2	+0%	+0%	+0%
Cerulli PSD 2	21	5	2	+0%	+0%	+0%
Cerulli PSD 3	21	5	2	+0%	+0%	+0%
Cerulli PSD 4	21	5	2	+0%	+0%	+0%
Cerulli PSD 5	66	10	5	-60%	+0%	-60%
Cerulli PSD 6	66	10	6	-67%	-67%	-67%
Cerulli PSD 7	105	13	7	-71%	-43%	-71%
Cerulli PSD 8	105	13	5	-60%	+0%	-60%

- QCQP problems from Cerulli et al. (2022) for constrained quadratic regression;
- Quadratic matrices are not necessarily PSD during intermediate iterations

Optimality-Based Generalized Discretization Methods

Motivation: *all* discretization methods perform poorly on `Watson h`

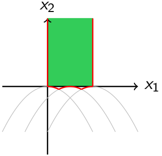
$$\min_{x_1,x_2} \ x_2$$
$$\text{s.t. } x_2 \geq -(x_1 - y)^2, \quad \forall y \in [-1, 1],$$
$$0 \leq x_1 \leq 1, \ -1000 \leq x_2 \leq 1000$$



Optimality-Based Generalized Discretization Methods

Motivation: *all* discretization methods perform poorly on `Watson h`

$$\min_{x_1,x_2} \; x_2$$
$$\text{s.t. } x_2 \geq -(x_1 - y)^2, \quad \forall y \in [-1, 1],$$
$$0 \leq x_1 \leq 1, \; -1000 \leq x_2 \leq 1000$$



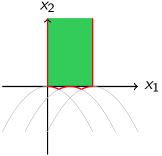
- The optimal solution mapping for the (LLP) is

$$y^*(x) \in \arg\max_{y \in [-1,1]} -(x_1 - y)^2 - x_2 \implies y^*(x) = x_1$$

Optimality-Based Generalized Discretization Methods

Motivation: *all* discretization methods perform poorly on `Watson h`

$$\min_{x_1,x_2} \; x_2$$
$$\text{s.t. } x_2 \geq -(x_1 - y)^2, \quad \forall y \in [-1,1],$$
$$0 \leq x_1 \leq 1, \;\; -1000 \leq x_2 \leq 1000$$



- The optimal solution mapping for the (LLP) is

$$y^*(x) \in \arg\max_{y\in[-1,1]} -(x_1 - y)^2 - x_2 \implies y^*(x) = x_1$$

- The BF algorithm approximates

$$g(x, y^*(x)) \leq 0$$

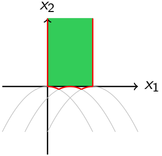
using the zeroth-order approximation $y^*(x) \approx y(x^k)$, i.e.,

$$g(x, y^*(x^k)) := g(x, y^k) \leq 0$$

Optimality-Based Generalized Discretization Methods

Motivation: *all* discretization methods perform poorly on `Watson h`

$$\begin{aligned} \min_{x_1, x_2} \;& x_2 \\ \text{s.t. } & x_2 \geq -(x_1 - y)^2, \quad \forall y \in [-1, 1], \\ & 0 \leq x_1 \leq 1, \; -1000 \leq x_2 \leq 1000 \end{aligned}$$



- The optimal solution mapping for the (LLP) is

$$y^*(x) \in \arg\max_{y \in [-1,1]} -(x_1 - y)^2 - x_2 \implies y^*(x) = x_1$$

- The BF algorithm approximates

$$g(x, y^*(x)) \leq 0$$

using the zeroth-order approximation $y^*(x) \approx y(x^k)$, i.e.,

$$g(x, y^*(x^k)) := g(x, y^k) \leq 0$$

- To use a first-order approximation (Seidel and Küfer, 2022):

$$y^*(x) \approx \texttt{proj}_Y\left(y^*(x^k) + \left(\frac{\partial y^*}{\partial x}(x^k)\right)(x - x^k)\right)$$

Optimality-Based Generalized Discretization Methods

Motivation: Hijazi et al. (2014)

$$\begin{aligned} \min_{x} \quad & -\|x\|^2 \\ \text{s.t.} \quad & \sum_{i=1}^{d_x} (x_i - y_i) y_i \leq 0, \quad \forall \{y \in Y\} \\ & X = [-1,1]^{d_x}, \quad Y = \{y \in \mathbb{R}^{d_x} : \|y\|^2 = d_x - 1\} \end{aligned}$$

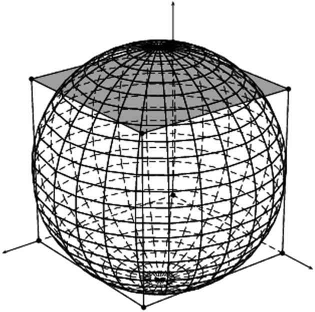
Optimality-Based Generalized Discretization Methods

Motivation: Hijazi et al. (2014)

$$\min_{x} \; -\|x\|^2$$

$$\text{s.t.} \; \sum_{i=1}^{d_x}(x_i - y_i)y_i \leq 0, \quad \forall\{y \in Y\}$$

$$X = [-1,1]^{d_x}, \quad Y = \{y \in \mathbb{R}^{d_x} : \|y\|^2 = d_x - 1\}$$

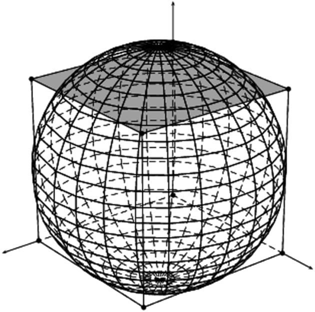


- Reformulation: $\|x\| \leq \sqrt{dx - 1}$

Optimality-Based Generalized Discretization Methods

Motivation: Hijazi et al. (2014)

$$
\begin{aligned}
\min_{x} \quad & -\|x\|^2 \\
\text{s.t.} \quad & \sum_{i=1}^{d_x} (x_i - y_i) y_i \leq 0, \quad \forall \{y \in Y\} \\
& X = [-1, 1]^{d_x}, \quad Y = \{y \in \mathbb{R}^{d_x} : \|y\|^2 = d_x - 1\}
\end{aligned}
$$

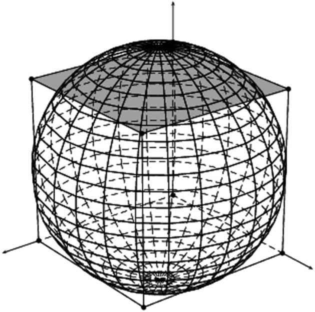


- Reformulation: $\|x\| \leq \sqrt{dx - 1}$
- every vertex of X is a solution of (LBP)
- Any discretization point $y \in Y$ can exclude at most one vertex of X

Optimality-Based Generalized Discretization Methods

Motivation: Hijazi et al. (2014)

$$
\begin{aligned}
\min_{x} \quad & -\|x\|^2 \\
\text{s.t.} \quad & \sum_{i=1}^{d_x} (x_i - y_i) y_i \leq 0, \quad \forall \{y \in Y\} \\
& X = [-1, 1]^{d_x}, \quad Y = \{y \in \mathbb{R}^{d_x} : \|y\|^2 = d_x - 1\}
\end{aligned}
$$



- Reformulation: $\|x\| \leq \sqrt{dx - 1}$
- every vertex of X is a solution of (LBP)
- Any discretization point $y \in Y$ can exclude at most one vertex of X
- but $\text{proj}_Y(x) = y^*(x)$, $\forall x \in X$

Optimality-Based Generalized Discretization Methods

- Our Idea: find a linear surrogate for y^* that yields the highest lower bound

$$(\bar{A}^k, \bar{b}^k) \in \underset{\substack{A^k \in \mathbb{R}^{d_y \times d_x} \\ b^k \in \mathbb{R}^{d_y}}}{\arg\max} \; \min_{x \in X} \; f(x)$$

$$\text{s.t. } g(x, \texttt{proj}_Y(A^k x + b^k)) \leq 0$$

$$g(x, \texttt{proj}_Y(A^i x + b^i)) \leq 0, \; \forall i \in [k-1]$$

Optimality-Based Generalized Discretization Methods

- Our Idea: find a linear surrogate for y^* that yields the highest lower bound

$$
\begin{aligned}
(\bar{A}^k, \bar{b}^k) \in \underset{\substack{A^k \in \mathbb{R}^{d_y \times d_x} \\ b^k \in \mathbb{R}^{d_y}}}{\arg\max} \ \min_{x \in X} \ & f(x) \\
\text{s.t. } & g(x, \texttt{proj}_Y(A^k x + b^k)) \leq 0 \\
& g(x, \texttt{proj}_Y(A^i x + b^i)) \leq 0, \ \forall i \in [k-1]
\end{aligned}
$$

- Initialise (A^k, b^k) using parametric sensitivity of the (LLP) — existence requires additional assumptions
- Similar assumptions and routines for computing sensitivities (use smooth approximation of $\texttt{proj}_Y$ in max-min)
- Can form generalized equivalents of the previous problems, e.g. `G-GREEDY`

Optimality-Based Generalized Discretization Methods

- Our Idea: find a linear surrogate for y^* that yields the highest lower bound

$$
\begin{aligned}
(\bar{A}^k, \bar{b}^k) \in \underset{\substack{A^k \in \mathbb{R}^{d_y \times d_x} \\ b^k \in \mathbb{R}^{d_y}}}{\arg\max} \ \min_{x \in X} \ & f(x) \\
\text{s.t. } & g(x, \mathtt{proj}_Y(A^k x + b^k)) \le 0 \\
& g(x, \mathtt{proj}_Y(A^i x + b^i)) \le 0, \ \forall i \in [k-1]
\end{aligned}
$$

- Initialise (A^k, b^k) using parametric sensitivity of the (LLP) — existence requires additional assumptions
- Similar assumptions and routines for computing sensitivities (use smooth approximation of $\mathtt{proj}_Y$ in max-min)
- Can form generalized equivalents of the previous problems, e.g. `G-GREEDY`
- (LBP) becomes a (manageable) MINLP!

Selected Numerical Results

- Implemented in Julia 1.7.3, JuMP 1.3.1
- Global Solver: Baron 21.1.13
  NLP Solver: Knitro 13.1.0
  LP Solvers: Gurobi 9.1.2 and CPLEX 22.1.0
  Bundle Solver: MPBNGC 2.0
- Absolute and Relative Optimality Tolerance: 10^{-3}

Problem	n_x	n_y	BF	GREEDY	G-GREEDY	G-2GREEDY
			# iter.	# iter. relative to BF		
Watson 2	2	1	2	+0%	+0%	+0%
Watson 5	3	1	5	-20%	-20%	-40%
Watson 6	2	1	3	-33%	-33%	-33%
Watson 8	6	2	15	-60%	TLE	-40%
Watson h	2	1	18	+61%	-89%	-89%
Tsoukalas	1	1	8	-38%	-75%	-63%
Mitsos 4_3	3	1	5	-20%	+0%	+0%

Conclusion

- Optimality-based discretization methods significantly reduces # iterations for convergence on problems from the literature
- `GREEDY` and `2GREEDY` perform the best in practice

Conclusion

- Optimality-based discretization methods significantly reduces # iterations for convergence on problems from the literature
- `GREEDY` and `2GREEDY` perform the best in practice
- The generalized methods have the potential to improve convergence rate significantly, but harder (LBP)
- Ongoing Work
  - More efficient, reliable methods for solving max-min problems
  - Application of max-min idea to other problem classes
  - Machine Learning for optimal discretization of SIPs

References I

Jerry W Blankenship and James E Falk. Infinitely constrained optimization problems. *Journal of Optimization Theory and Applications*, 19(2):261–281, 1976.

Martina Cerulli, Antoine Oustry, Claudia d'Ambrosio, and Leo Liberti. Convergent algorithms for a class of convex semi-infinite programs. *SIAM Journal on Optimization*, 32(4):2493–2526, 2022.

Hatim Djelassi, Alexander Mitsos, and Oliver Stein. Recent advances in nonconvex semi-infinite programming: Applications and algorithms. *EURO Journal on Computational Optimization*, 9:100006, 2021.

Anthony V Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*, volume 165. Academic press, 1983.

Hassan Hijazi, Pierre Bonami, and Adam Ouorou. An outer-inner approximation for separable mixed-integer nonlinear programs. *INFORMS Journal on Computing*, 26 (1):31–44, 2014.

James E Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

A Mitsos. Test set of semi-infinite programs. In *Technical Report*. Massachusetts Institute of Technology, 2009.

Tobias Seidel and Karl-Heinz Küfer. An adaptive discretization method solving semi-infinite optimization problems with quadratic rate of convergence. *Optimization*, 71(8):2211–2239, 2022.

References II

Georg Still. *Lectures on parametric optimization: An introduction*. Optimization Online, 2018.

Angelos Tsoukalas and Berç Rustem. A feasible point adaptation of the blankenship and falk algorithm for semi-infinite programming. *Optimization Letters*, 5(4): 705–716, 2011.

G Alistair Watson. Numerical experiments with globally convergent methods for semi-infinite programming problems. In *Semi-infinite programming and applications*, pages 193–205. Springer, 1983.