Mixed-Integer Nonlinear Optimisation: Outer Approximation

Ruth Misener
r.misener@imperial.ac.uk

Computational Optimisation Group
Centre for Process Systems Engineering
Department of Computing

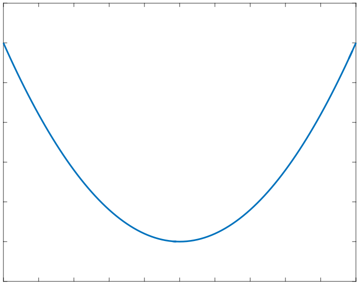
10 May 2017

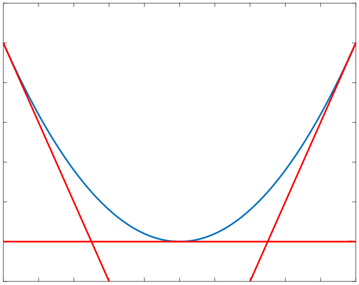
1. Multi-Variable Relaxations of Convex Functions
   - Trade-Offs in Outer-Approximation Cuts
   - Convexity Detection

Linear Underestimators for Convex Terms

Sanity Check.

Why may we want underestimators for convex terms?



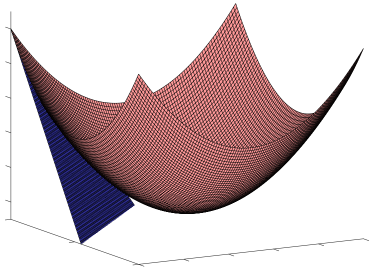


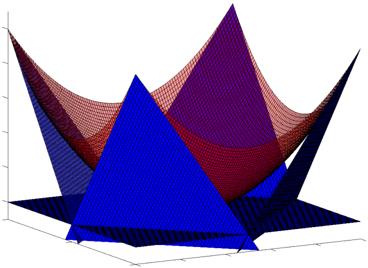
$$\nabla g_i\left(\hat{\mathbf{x}}\right)^T\left(\mathbf{x}-\hat{\mathbf{x}}\right)+g_i\left(\hat{\mathbf{x}}\right)\leq 0$$

Linear Underestimators for Convex Terms

Sanity Check.

Why may we want underestimators for convex terms?





$$\nabla g_i\left(\hat{\mathbf{x}}\right)^T\left(\mathbf{x}-\hat{\mathbf{x}}\right)+g_i\left(\hat{\mathbf{x}}\right)\leq 0$$

How to solve convex MINLP using Outer Approximation?

Key Idea

We're pretty good at solving MILP with state-of-the-art solvers. Solve a series of MILP converging to the solution of the convex MINLP.

1. Use R previous MILP relaxations $\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots \hat{\mathbf{x}}_r$. Solve the new MILP:

$$\begin{array}{lll} \min_{\mathbf{x}} & \mathbf{c}^T \mathbf{x} & \\ \text{s.t.} & \nabla g_i\left(\hat{\mathbf{x}}_r\right)^T\left(\mathbf{x}-\hat{\mathbf{x}}_r\right)+g_i\left(\hat{\mathbf{x}}_r\right) \leq 0 & \forall\, i, r=1, \ldots R \\ & x_j \in \mathbb{Z} & \forall\, j \in I \end{array}$$

2. Solve convex NLP with integer values fixed. Get a new feasible point.
3. Set $R = R + 1$ and repeat the MILP solve in Step 1.
4. Converge when the solution of the MILP $\mathbf{c}^T \hat{\mathbf{x}}$ is within a pre-determined tolerance of the convex NLP $\mathbf{c}^T \mathbf{x}^\star$.

How to solve convex MINLP using Outer Approximation?

Key Idea

We're pretty good at solving MILP with state-of-the-art solvers. Solve a series of MILP converging to the solution of the convex MINLP.

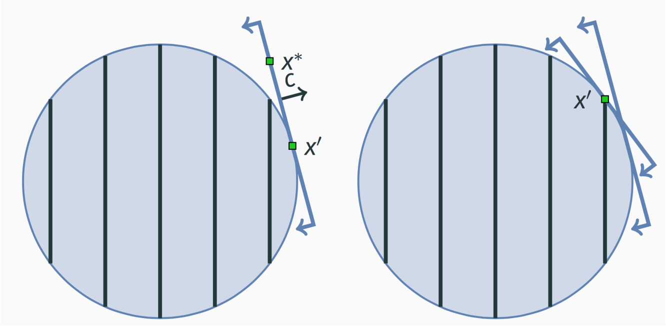
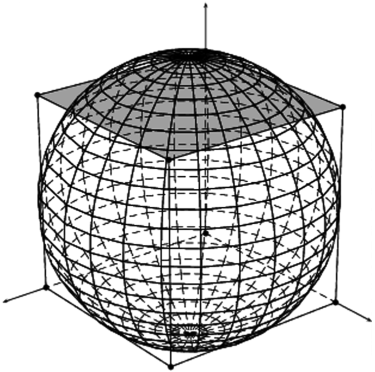


Image credit: Miles Lubin

What could go wrong with outer approximation?

Solve the feasibility problem.

$$\mathbb{B}^3 = \left\{ \mathbf{x} \in \{0,1\}^3 \middle| \sum_{j=1}^{3} \left(x_j - \frac{1}{2} \right)^2 \leq \frac{1}{2} \right\}$$



Worst case scenario?

How many *gradient linearisations* are necessary in the worst case?

Answer

The $\mathbb{B}^3$ example requires 3 variables and $2^3 = 8$ linearisations. In general, for a ball with n dimensions, we may need 2^n constraints.

Hijazi, Bonami, Ouorou. *INFORMS J Computing*, 2014.

How address a possibly exponential number of constraints?

Key Idea

Introduce new variables to construct an *extended formulation*! Given $\mathbb{B}^n$:

$$\mathbb{B}^n = \left\{ \mathbf{x} \in \{0,1\}^n \;\middle|\; \sum_{j=1}^{n} \left(x_j - \frac{1}{2} \right)^2 \leq \frac{n-1}{4} \right\},$$

construct a new $\hat{\mathbb{B}}^n$ such that $\mathbb{B}^n = \mathrm{proj}_{\mathbf{x}} \hat{\mathbb{B}}^n$:

$$\mathbb{B}^n = \left\{ \mathbf{x} \in \{0,1\}^n, \mathbf{z} \in \mathbb{R}^n \;\middle|\; \sum_{j=1}^{n} z_j \leq \frac{n-1}{4}, \left(x_j - \frac{1}{2} \right)^2 \leq z_j \; \forall j \right\}.$$

Outcome?

We now have increased the number of variables from 3 to $2 \times 3 = 6$, but now we only need $2 \times 3 = 6$ hyperplanes in $\mathbb{R}^6$ to exclude all integer points. The outer approximation algorithm now converges in 2 iterations.

Separable Quadratic Facility Location Problems (SQFL)

$$\begin{aligned}
\min \quad & \left\{ \sum_{i \in I} f_i z_i + \sum_{i \in I, j \in J} q_{ij} x_{ij}^2 \right\} \\
\text{s.t.} \quad & x_{ij} \le z_i \quad \forall i \in I,\ j \in J, \\
& \sum_{i \in I} x_{ij} = 1 \quad \forall j \in J, \\
& z_i \in \{0, 1\},\ x_{ij} \in [0, 1] \quad \forall i \in I,\ j \in J.
\end{aligned}$$

Table 1 CPU Times for SQFL Instances

			Initial formulation		Univariate formulation			
Inst.	$\|I\|$	$\|J\|$	CPLEX	B − OA	B − OA	B − OA + Ref	Inner sol (%)	Inner time
1	8	30	0.91	70.01	7.36	2.20	0.00	0.35
2	20	100	25.36	[16.59%]	153.66	25.06	0.00	6.47
3	20	150	141.95	[64.96%]	3,075.63	464.40	1.21	71.40
4	30	150	249.31	[48.79%]	2,496.28	338.32	2.13	49.26
5	35	300	[12.42%]	[72.58%]	[54.62%]	[4.57%]	9.73	180.00
6	50	500	[77.91%]	[91.72%]	[64.51%]	[51.78%]	[∞]	180.00

Hijazi, Bonami, Ouorou. *INFORMS J Computing*, 2014.

What is the justification for extended formulations?

Key Idea

Introduce extra variables for a formulation with $2n$ variables and $2n+1$ constraints rather than n variables and 2^n constraints. Small polyhedron in higher dimension can have exponentially many facets in lower dimension.

Common trick in linear programming

Given:

$$\mathcal{B}_1 = \{\mathbf{x} | \, \|\mathbf{x}\|_1 \leq 1\} = \left\{ \mathbf{x} \,\middle|\, \sum_{j=1}^{n} s_j x_j \leq 1, s \in \{-1,+1\}^n \right\},$$

Reformulate:

$$\mathcal{B}_1 = \left\{ \mathbf{x} \,\middle|\, \exists y \in \mathbb{R}^n \text{ s.t. } \sum_{j}^{n} y_j \leq 1, x_j \leq y_j, x_j \geq -y_j, \; \forall j = 1, \ldots, n \right\}.$$

General Principle[1]

Proposition

Considerable a separable function: $f(\mathbf{x}) = f_1(x_1) + \cdots + f_n(x_n)$, where each f_i is strictly convex. The outer approximation formed by:

$$S_1(f) = \left\{ (\gamma, \mathbf{x}) \middle| \gamma \geq f\left(\hat{\mathbf{x}}^{(k)}\right) + \nabla f\left(\hat{\mathbf{x}}^{(k)}\right)\left(\mathbf{x} - \hat{\mathbf{x}}^{(k)}\right), \hat{\mathbf{x}}^{(k)} \in \mathbb{P} \right\},$$

is a subset of:

$$S_2(f) = \left\{ (\gamma, \mathbf{x}) \middle| \begin{array}{l} \gamma \geq \sum_i \phi_i \\ \phi_i \geq f_i\left(\hat{x}_i^{(k)}\right) + \nabla f_i\left(\hat{x}_i^{(k)}\right)\left(x_i - \hat{x}_i^{(k)}\right), \hat{\mathbf{x}}^{(k)} \in \mathbb{P} \end{array} \right\},$$

and if there are r points in $\mathbb{P}$, then we may need as many as r^n points to approximate $S_1(f)$ as tightly as $S_2(f)$.

[1] Tawarmalani & Sahinidis, *Math Program*, 2005

Automate extended formulations in general MINLP?

Convexity detection is a hard problem

$$\begin{array}{lll} \min_{\mathbf{x}} & \mathbf{c}^T\mathbf{x} & \\ \text{s.t.} & g_i(\mathbf{x}) \le 0 & \forall\ i \\ & x_j \in \mathbb{Z} & \forall\ j \in I \end{array}$$

For example, consider convex function $g_i(\mathbf{x}) = g_{i1}(\mathbf{x}) + g_{i2}(\mathbf{x})$. The subfunctions g_{i1} and g_{i1} are not necessarily convex:

$$\begin{aligned} g_{i1}(\mathbf{x}) &= \ \ 2x_1^2 - \ x_2^2 \\ g_{i2}(\mathbf{x}) &= -\ x_1^2 + 2x_2^2 \end{aligned}$$

Implementation Advances

Hijazi, Bonami, Ouorou (2014), Vielma, Dunning, Huchette, Lubin (2015)

Completed implementations

Hijazi, Bonami, Ouorou (2014) $\Rightarrow$ CPLEX

Trick with $\mathbb{B}^n$

Vielma, Dunning, Huchette, Lubin (2015) $\Rightarrow$ CPLEX & Gurobi

Second-order cone:

$$\mathrm{SOC}_n = \left\{(t, \mathbf{x}) \in \mathbb{R}^{n+1} \,|\, \|\mathbf{x}\|_2 \leq t\right\},$$

Rewrite:

$$\sum_i x_i^2 \leq t^2 \Rightarrow \sum_i z_i \leq t, \text{ where } x_i^2/t \leq z_i \ \forall i.$$

So that:

$$\mathrm{SOC}_n = \left\{(t, \mathbf{x}, \mathbf{z}) \in \mathbb{R}^{2n+1} \,\middle|\, \sum_i z_i \leq t, \left\{\begin{array}{ll} x_i^2/t \leq z_i & t > 0 \\ 0 \leq z_i & t = 0 \\ \infty \leq z_i & t < 0 \end{array}\right\} \ \forall i \right\},$$

What else could we implement?

Convex quadratic

Assume that we know that matrix $\mathbf{Q}$ is positive definite and therefore $g(\mathbf{x})$ is convex:

$$g(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} \leq 0.$$

Recall from linear algebra that *spectral decomposition* $\mathbf{Q} = \mathbf{D} \boldsymbol{\Lambda} \mathbf{D}^T$ of real symmetric matrix $\mathbf{Q}$ has diagonal matrix $\boldsymbol{\Lambda}$. Then the constraint can be written with $2n$ variables:

$$g(\mathbf{z}) = \mathbf{z}^T \boldsymbol{\Lambda} \mathbf{z} \leq 0,$$
$$\mathbf{z} = \mathbf{x}^T \mathbf{D}.$$

Why can we assume that $\mathbf{Q}$ is symmetric?

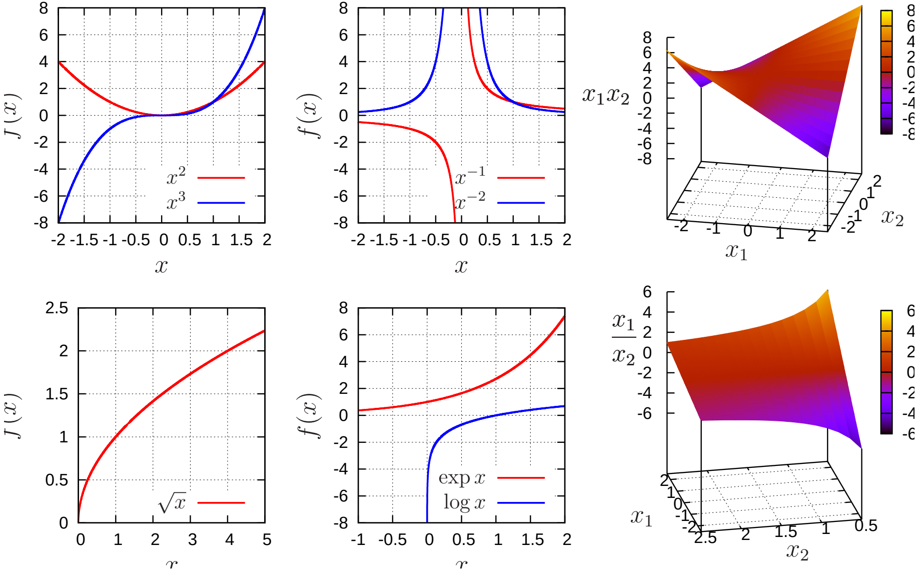
What if we had full convexity knowledge?

Disciplined convex programming (Grant, Boyd, and Ye, Lubin, Yamangil, Bent, Vielma, Coey)

1. Multi-Variable Relaxations of Convex Functions
   - Trade-Offs in Outer-Approximation Cuts
   - Convexity Detection

Common Functions in Modeling

Adapted from B Chachuat



Detecting Convexity

Adapted from B Chachuat

Operations That Preserve Convexity

- Positive combination of convex functions: Why positive?

$$\left.\begin{array}{l} f, g : S \subseteq \mathbb{R}^n \to \mathbb{R}^m \text{ convex,} \\ \alpha, \beta \geq 0 \end{array}\right\} \Longrightarrow \mathbf{x} \mapsto \alpha f(\mathbf{x}) + \beta g(\mathbf{x}) \text{ convex on } S$$

- Inner composition of a convex function with an affine function:

$$\left.\begin{array}{l} f : \mathbb{R}^n \to \mathbb{R}^m \text{ convex,} \\ \mathbf{A} \in \mathbb{R}^{n \times p}, \mathbf{b} \in \mathbb{R}^n \end{array}\right\} \Longrightarrow \mathbf{x} \mapsto f(\mathbf{A}\mathbf{x} + \mathbf{b}) \text{ convex on } \mathbb{R}^p$$

- Maximum over a set of convex functions: Try and represent it!

$$f^k : \mathbb{R}^n \to \mathbb{R}^m \text{ convex, } k \in K \Longrightarrow \mathbf{x} \mapsto \max_{k \in K} f^k(\mathbf{x}) \text{ convex on } \mathbb{R}^p$$

Operations That Do Not Preserve Convexity

- Product and composition of univariate/multivariate convex functions
E.g., $x \mapsto x \times x^2$, $x \mapsto [-\log(x)]^2$

Detecting Convexity

Adapted from B Chachuat

Given a multivariable function $f : \mathbb{R}^n \to \mathbb{R}$ and a point $\hat{\mathbf{x}} \in \mathbb{R}^n$,

- the gradient, $\boldsymbol{\nabla} f(\hat{\mathbf{x}})$, is the vector of first partial derivatives at $\hat{\mathbf{x}}$
- the Hessian, $\mathbf{H}(\hat{\mathbf{x}})$, the matrix of second partial derivatives

$$\boldsymbol{\nabla} f(\hat{\mathbf{x}}) \triangleq \begin{pmatrix} \frac{\partial f}{\partial x_1}(\hat{\mathbf{x}}) \\ \frac{\partial f}{\partial x_2}(\hat{\mathbf{x}}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\hat{\mathbf{x}}) \end{pmatrix} \quad \mathbf{H}(\hat{\mathbf{x}}) \triangleq \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(\hat{\mathbf{x}}) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\hat{\mathbf{x}}) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\hat{\mathbf{x}}) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\hat{\mathbf{x}}) & \frac{\partial^2 f}{\partial x_2^2}(\hat{\mathbf{x}}) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(\hat{\mathbf{x}}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\hat{\mathbf{x}}) & \frac{\partial^2 f}{\partial x_n \partial x_2}(\hat{\mathbf{x}}) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(\hat{\mathbf{x}}) \end{pmatrix}$$

Gradient and Hessian **Sufficient** Conditions for Convexity

A function $f : S \subset \mathbb{R}^n \to \mathbb{R}$ in $\mathcal{C}^2$ is convex on S if, at each $\hat{\mathbf{x}} \in S$,

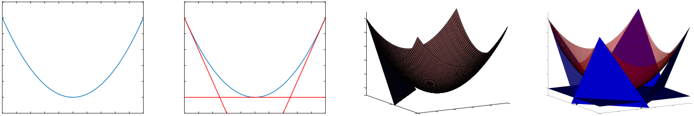
- **Gradient Test:** $f(\mathbf{x}) \geq f(\hat{\mathbf{x}}) + \boldsymbol{\nabla} f(\hat{\mathbf{x}})^T (\mathbf{x} - \hat{\mathbf{x}}), \quad \forall \mathbf{x} \in S$
- **Hessian Test:** $\mathbf{H}(\hat{\mathbf{x}}) \succeq 0$ (positive semi-definite)

Detecting Convexity – Workshop

Workshop. Determine whether or not the following functions are convex:

- $x \mapsto x \log(x)$, for $x > 0$
- $(x_1, x_2) \mapsto x_1^2 + x_1 x_2 + 2x_2 + 4$, for $(x_1, x_2) \in \mathbb{R}^2$

Tests for Convexity in High-Dimension Quadratics? [1/2]



$\mathbf{x}^T\mathbf{Q}\mathbf{x} \Rightarrow$ Numeric Test for Convexity (LAPACK)

```
dsyev(cJobz, cUpLo, iNumVars, pdA, iLDA, pdEigenVal,
      pdWorkspace, iWorkLength, iExitInfo);
```

Where:

`cJobz`	V to get the eigenvalues & eigenvectors
`cUpLo`	U to store the matrix $\mathbf{Q}$ in upper-triangular form
`iNumVars`	Dimension n of the matrix $\mathbf{Q}$
`pdQ`	The matrix $\mathbf{Q}$. At function exit, it holds the eigenvectors
`iLDA`	Leading dimension n of the array.
`pdEigenVal`	When the function exits, this holds the eigenvalues.

Reasonable for small n. How big can n be? What is greater than 0?