Building AD-compatible linear underestimators of convex functions by sampling

Yingkai Song  Huiyi Cao  Kamil A. Khan
kamilkhan@mcmaster.ca

Department of Chemical Engineering, McMaster University

24th Euro AD Workshop

November 2, 2021
Motivation: Global optimization

Several applications require deterministic global optimization of nonconvex systems:

▶ Thermodynamic equilibria: \( \min_x \) Gibbs free energy \((x)\)

▶ Worst-case uncertainty analysis: \( \max_{\theta \in P} \) Cost(\(\theta\))

▶ Model validation:

\[
\min_{p \in P} \text{Discrepancy(Model prediction}(p), \text{Data})
\]

Is the best fit good enough?

▶ Process optimization: \( \max_{p \in P} \) Net present value \((p)\)

Dynamic systems present unique challenges:

▶ Batch reactors and semibatch processes
▶ Closed-loop control systems
▶ Startup/shutdown
**Overarching problem formulation**

**Global optimization**

To **global optimality**, solve:

\[
\min_{p \in P} f(p)
\]

when \( f \) is nonconvex and \( P \) is a box.

(We will add in nontrivial constraints later.)

**Observations**

- Misleadingly easy for functions of one variable.
- The function \( f \) contains all of the problem’s information.
- Gradients don’t describe global behaviour.
Background: Global optimization methods

- Solvers (e.g. BARON, ANTIGONE) generate upper bounds and lower bounds on the unknown solution
- **Local minimization** yields upper bounds
- Local minimization of a **convex relaxation** yields lower bounds
- Tighter relaxations lead to fewer optimization iterations
- How can we compute better relaxations? How can we make better use of them?
Background: Global optimization methods

- Solvers (e.g. BARON, ANTIGONE) generate upper bounds and lower bounds on the unknown solution
- Local minimization yields upper bounds
- Local minimization of a convex relaxation yields lower bounds
- Tighter relaxations lead to fewer optimization iterations
- How can we compute better relaxations? How can we make better use of them?

\[ y = f(p) \]

\[ p^L \quad p^U \]
Background: Global optimization methods

- Solvers (e.g. BARON, ANTIGONE) generate upper bounds and lower bounds on the unknown solution
- **Local minimization** yields upper bounds
- Local minimization of a **convex relaxation** yields lower bounds
- Tighter relaxations lead to fewer optimization iterations
- How can we compute better relaxations? How can we make better use of them?
Interval arithmetic
Described by RE Moore (1979), A Neumaier (1990)

- Established, robust relaxation method based on an AD-ish approach
- Replaces each elemental function with constant interval bounds
- Bounding rules for elemental functions:
  - $x \in [x^L, x^U] \subset \mathbb{R}$, and $y := \exp(x)$
    - Conclusion: $y \in [\exp(x^L), \exp(x^U)]$
  - $a \in [a^L, a^U]$ and $b \in [b^L, b^U]$
    - if $a^L, b^L \geq 0$, then $ab \in [a^L b^L, a^U b^U]$
    - in general, $ab \in [\min\{a^L b^L, a^L b^U, a^U b^L, a^U b^U\}, \max\{a^L b^L, a^L b^U, a^U b^L, a^U b^U\}]$
Interval arithmetic
Described by RE Moore (1979), A Neumaier (1990)

Chaining these rules together, we may use a forward AD variant to obtain interval bounds of composite functions:

\[
f(x_1, x_2) := x_1 \exp x_2 + x_2^2
\]

\[
f(x_1, x_2) \leftarrow v_6
\]

\[
\begin{align*}
v_1 & \leftarrow x_1 \\
v_2 & \leftarrow x_2 \\
v_3 & \leftarrow \exp v_2 \\
v_4 & \leftarrow v_1 v_3 \\
v_5 & \leftarrow v_2^2 \\
v_6 & \leftarrow v_4 + v_5
\end{align*}
\]

\[
\begin{align*}
[v_1^L, v_1^U] & \leftarrow [x_1^L, x_1^U] =: X_1 \\
[v_2^L, v_2^U] & \leftarrow [x_2^L, x_2^U] =: X_2 \\
[v_3^L, v_3^U] & \leftarrow [\exp v_2^L, \exp v_2^U] \\
[v_4^L, v_4^U] & \leftarrow [\min\{v_1^L v_3^L, v_1^L v_3^U, v_1^U v_3^L, v_1^U v_3^U\}, \max\{\cdots\}] \\
[v_5^L, v_5^U] & \leftarrow [(\text{median}\{v_2^L, 0, v_2^U\})^2, \max\{(v_2^L)^2, (v_2^U)^2\}] \\
[v_6^L, v_6^U] & \leftarrow [v_4^L + v_5^L, v_4^U + v_5^U] \\
F(X_1, X_2) & \leftarrow [v_6^L, v_6^U]
\end{align*}
\]

Higher-order extensions have been studied
Tsoukalas-Mitsos relaxations

Generalization and improvement of an earlier method by McCormick (1976)

General Tsoukalas-Mitsos rule

- Suppose functions $\phi$ and $f$ have known convex/concave relaxations $\phi^{cv}/\phi^{cc}/f^{cv}/f^{cc}$, and consider $g(x) = \phi(f(x))$.
- Convex/concave relaxations for $g$ are described as optimal-value functions:

$$g^{cv}(x) := \min\{\phi^{cv}(z) : f^{cv}(x) \leq z \leq f^{cc}(x)\}$$
$$g^{cc}(x) := \max\{\phi^{cc}(z) : f^{cv}(x) \leq z \leq f^{cc}(x)\}$$

- Can again be implemented like forward AD (e.g. MC++, EAGO, MAiNGO)
- Smooth if we use smooth relaxations for $\phi$ and $f$ (KK and Barton (2017), KK (2018))
Computing subgradients of relaxations

▶ The forward AD mode works on the older McCormick relaxations (Mitsos et al. (2009))

▶ A reverse AD mode does too. Even though relaxations might be nonsmooth, convex functions nevertheless behave well enough. (Beckers and Naumann, AD2012)

▶ Both modes extend to the newer Tsoukalas-Mitsos rule (Yuan and KK (in prep.), EuroAD 2020)
Linear relaxations

- Computing lower bounds by minimizing convex relaxations requires an NLP solver and several evaluations.

\[ y = f(x) \]
\[ y = f^{cv}(x) \]

- Gradients/subgradients of relaxations yield subtangents.
- Subtangents provide useful lower bounding information without an NLP solver [Khan 2018; Cao, Song, and Khan 2019].
- In some cases no evaluation method is even known [Scott and Barton 2013].
Linear relaxations

- Computing lower bounds by minimizing convex relaxations requires an NLP solver and several evaluations

\[ y = f(x) \]

- Gradients/subgradients of relaxations yield subtangents
- Subtangents provide useful lower bounding information without an NLP solver [Khan 2018; Cao, Song, and Khan 2019]
- In some cases no evaluation method is even known [Scott and Barton 2013]
Can finite differences provide bounds?

- Established finite differences cannot guarantee a lower bound.
- e.g. Centered finite difference:

\[ y = f^{cv}(x) \]

- Invalid underestimators can cause global optimization methods to fail
Can finite differences provide bounds?

- Established finite differences cannot guarantee a lower bound.
- e.g. Centered finite difference:

$$y = f^{cv}(x)$$

- Invalid underestimators can cause global optimization methods to fail.
Can finite differences provide bounds?

- Established finite differences cannot guarantee a lower bound.
- e.g. Centered finite difference:

\[ y = f^{cv}(x) \]

- Invalid underestimators can cause global optimization methods to fail
Can finite differences provide bounds?

- Established finite differences cannot guarantee a lower bound.
- e.g. Centered finite difference:

\[ y = f^{cv}(x) \]

- Invalid underestimators can cause global optimization methods to fail
Finite difference example

- Finite differences may be especially poor for nonsmooth convex functions of $\geq 3$ variables. [Song et al., in press]

- Consider the following convex piecewise-linear function from [Khan and Yuan 2020]:

$$f(x_1, x_2, x_3) \equiv \max(x_1 + x_2 - x_3, x_2 + x_3 - x_1, x_3 + x_1 - x_2)$$

- All subgradients $s$ of $f$ at $(0, 0, 0)$ obey:

$$s_1 + s_2 + s_3 = 1.$$

- However, even with no numerical error in function evaluations,
  - forward finite differencing always yields $s \approx (1, 1, 1)$, and
  - centered finite differencing always yields $s \approx (0, 0, 0)$
  - ... for any finite differencing perturbation $\epsilon$.

- Smooth functions can behave similarly when the finite differencing perturbation $\epsilon$ is fixed.
Suppose we have a convex relaxation $f^{cv}$ of $n$ variables on a box.

We can tractably evaluate a useful guaranteed lower bound by:

1. performing $(2n + 1)$ black-box evaluations of $f^{cv}$, and
2. applying a new centered finite difference formula

Extends concepts from derivative-free optimization

No further assumptions; no need for gradients or NLP solvers

Uncertainty in function evaluations can be incorporated as well
Basic formulation
Song et al., *Comput. Chem. Eng.*, in press

- Suppose we have:
  - a box $X := \{ \xi \in \mathbb{R}^n : x^L \leq \xi \leq x^U \} \subset \mathbb{R}^n$,
  - a convex function $f^{cv} : X \rightarrow \mathbb{R}$
  - the midpoint $w^{(0)} := \frac{1}{2} (x^L + x^U)$ of $X$
  - for each coordinate $i$:
    - a step length $\alpha_i \in (0, 1]$ and
    - two vectors $w^{(\pm i)} := w^{(0)} \pm \frac{\alpha_i}{2} (x^U_i - x^L_i)e_i$,
  - and function values $y_0 := f(w^{(0)})$ and $y_{\pm i} := f(w^{(\pm i)})$ for each $i$. 
Basic formulation
Song et al., *Comput. Chem. Eng.*, in press

Construct:
- a vector \( \mathbf{b} \in \mathbb{R}^n \) for which:

\[
   b_i := \frac{y_{+i} - y_{-i}}{\|\mathbf{w}(+i) - \mathbf{w}(-i)\|_{\infty}}
\]

(already known as the centered simplex gradient)
- and two scalars:

\[
   c := y_0 - \frac{1}{2} \sum_{i=1}^{n} \left( \frac{y_{+i} + y_{-i} - 2y_0}{\alpha_i} \right),
\]

\[
   f^{L} := y_0 - \sum_{i=1}^{n} \left( \frac{\max(y_{+i}, y_{-i}) - y_0}{\alpha_i} \right).
\]

Then, for each \( \mathbf{x} \in X \), we obtain new bounds:

\[
   f^{cv}(\mathbf{x}) \geq c + \mathbf{b}^T(\mathbf{x} - \mathbf{w}^{(0)}) \geq f^{L}.
\]

These are finite difference formulas, but are not finite difference approximations. Still compatible with AD.
If $f^{cv}$ is univariate, then $c$ and $f^L$ can be increased further. In $\mathbb{R}$, $X$ lies on the same line as all three sampled points.

From top to bottom:
- Dotted purple: a generic convex function $f^{cv}$
- Dashed black: secants through sampled points
- Solid yellow: our new relaxation for univariate convex $f^{cv}$
- Dash-dotted green: our new relaxation for generic convex $f^{cv}$
Properties

- The original function $f^{cv}$ doesn’t need to be smooth
- Tractable; uses only $(2n + 1)$ function evaluations and arithmetic.
- Inherits rapid convergence: If we have a scheme of convex estimators $f^{cv,X}$ of a nonconvex smooth function $f$ with:

$$\sup_{x \in X} (f(x) - f^{cv,X}(x)) \leq \tau^{cv} (\text{diam } X)^2,$$

for each box $X$, then this second-order pointwise convergence (Bompadre and Mitsos, 2012) also holds for our sampling-based relaxations:

$$\sup_{x \in X} (f(x) - f^{aff,X}(x)) \leq \tau^{aff} (\text{diam } X)^2,$$

for each box $X$. 


Use in global optimization

- Our sampling-based affine relaxations are correct and converge rapidly, so they are amenable to use in global optimization.
- In exact arithmetic, branch-and-bound will definitely converge to the correct globally optimal value.
- Clustering is avoided in branch-and-bound.
- Particularly useful when gradients/subgradients are unknown or inconvenient to evaluate.
Use in global optimization

- Constraints can be handled similarly; sampling constraint relaxations as well yields a guaranteed LP outer approximation. So:

\[
\begin{align*}
\min_{x \in \Xi} \quad & f(x) \\
\text{subject to} \quad & g_k(x) \leq 0, \quad \text{for each } k \in \{1, \ldots, m\}
\end{align*}
\]

is relaxed to:

\[
\begin{align*}
\min_{x \in \Xi} \quad & f^{\text{aff}}(x) \\
\text{subject to} \quad & g_k^{\text{aff}}(x) \leq 0, \quad \text{for each } k \in \{1, \ldots, m\}
\end{align*}
\]
Variants

- We can obtain correct underestimators even if function evaluations are only computed to within an absolute tolerance:
  \[ |\tilde{f}^{cv}(x) - f^{cv}(x)| \leq \epsilon \]

- The sampled points don’t need to be centered within the box.

- So, we can obtain tighter LP outer approximation of constrained NLPs, using several sampled relaxations of each function:

  \[
  \min_{x \in \Xi, \, t \in \mathbb{R}} t \\
  \text{subject to} \quad f^{aff,j}(x) \leq t, \quad \text{for each } j \\
  g^{aff,j}_k(x) \leq 0, \quad \text{for each } j, k
  \]

- Compatible with the McCormick/Tsoukalas-Mitsos rules, and lets us include elemental functions whose subgradients aren’t known (e.g. optimal-value functions or discontinuous ODE solutions)
Examples

Song et al., CACE, in press

- Applied to convex relaxations and global optimization problems:
  - Right: McCormick relaxations of \( \phi(x, y_1, y_2) = (\sqrt{y_1} - y_2) \exp(-x) \)

- Performs comparably to subgradient-based bounds in global optimization, but without requiring gradients/subgradients

- Ongoing work: could we get by with \((n + 2)\) sampled points instead of \((2n + 1)\) points?
Conclusions

- We have developed the first tractable derivative-free method for lower-bounding convex relaxations in global optimization

- Uses a new finite difference formula that is (unusually) compatible with AD. Try it out!

- Proof-of-concept implementation in Julia via EAGO

- Funding: NSERC Discovery