Interval Analysis on DAGs for Global Optimization

Hermann Schichl
Arnold Neumaier

Universität Wien

as part of the COCONUT project, an European Community initiative for an adoption of global optimization techniques.

www.ac.at/sicm/aco/ & www.cactus.at/concur/
Motivation

Global optimization w.r.t. guarantees

- constrained propagation
- interval decision type methods
- outer approximation
- bound & bound

Outer approximation replaces "infinite" (unbounded) constraints by finite, "approximated" constraints to create a bounded problem whose solution sets are the same (same decoder or decoding lower bound.)

- convex underestimation via LP relaxations
- interval constraint by LP method
- convex quadratic underestimation via SCC relaxations
- outer bounds by interval point methods
- polyhedral underestimation
- polyhedral relaxation
- outer bounds by removal methods (singlepoint - bisection)

Q. How to find linear/quadratic underestimator?
Linear under-estimation

(1) Using slopes

The slope defined at a particular point is
containing the function.

The linear under-estimation of the curve can define as
under-estimating due to
a point at a tangent line
by the formula:

\[ f(x) + f'(x)(x - x_0) = f(x_0) \]

where \( x_0 \) is the point of tangency.

- Use standard techniques
- Use geometric interpretation

- Additional cases in simple cases
- Use quadratic approximation

Critical point: slope at a corner
- Optimal if slope optimal
Example: Consider the system of equations

\[\begin{align*}
\mathbf{A}\mathbf{x} &= \mathbf{b} \\
\mathbf{x}(0) &= \mathbf{x}_0 \\
\mathbf{A} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{x}_0 &= \begin{bmatrix} x_{01} \\ x_{02} \end{bmatrix}
\end{align*}\]

To understand the system, we expected to get good results from the initial conditions. However, the solution did not match the expected values.

The results did not meet our expectations.

The errors in the calculations are as follows:

\[\begin{align*}
\text{Error in } a_{11} &= 0.05 \\
\text{Error in } a_{12} &= 0.03 \\
\text{Error in } a_{21} &= 0.02 \\
\text{Error in } a_{22} &= 0.01
\end{align*}\]
Linear underestimation

(N) Recursive underestimation

- Based on the DAE
- All intermediate results are labeled \( x_i \)
- Propagate inequality of the form

\[
0 \leq L(x) \leq U(x)
\]

- At each node, eliminate the intermediate \( x_i \) using linear bounds from the operation

\[
\begin{align*}
x_i &\geq g_0(x) \\
g_i(x) &> f_i(x) \\
g_i(x) &> f_i(x) \\
g_i(x) &> f_i(x) \\
g_i(x) &> f_i(x)
\end{align*}
\]

- Result:

\[
0 \leq L(x) \leq U(x)
\]
**Quadratic underestimator**

(i) from interval Hessians

$$f(x) = (x_1 - x_2)^2 + (x_1 - x_2)^2$$

for some $x \in \mathbb{R}^n$. 

- Interval Hessians can be computed on the fly as in automatic differentiation.

$$\nabla f(x) \cdot (x' - x)$$

where $x'$ is arbitrary

$$\frac{\partial f}{\partial x_j}(x) \cdot (x' - x)$$

subject to $x_j = \max(x_j, x'_j)$

$$x_j = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$$x'_j = \begin{pmatrix} x'_1 \\ \vdots \\ x'_n \end{pmatrix}$$

$$\nabla f(x) \cdot (x' - x) = \begin{pmatrix} \sum_{i=1}^n (x_i - x'_i)^2 \\ \vdots \\ \sum_{i=1}^n (x_i - x'_i)^2 \end{pmatrix}$$

+ Assumes local quadratic convexity.

- Chooses last row if exact convexity is unknown.
Quadratic underestimators

\( \phi(x) \) from 2nd order slopes

\[
\phi(x) = ax^2 + bx + c
\]

Rewritten using: \( \phi(x) = f(x) \) in horizontal mode, modified theorem conditions.

For \( \phi(x) \), the behavior changes at points:

- \( f(x) = \phi(x) \) for points \( x = a, b, c \) (within)
Quadratic underestimators

(1) direct backward underestimation

- Propagating squares of the quadratic form
  \[ Q(x) = x^T H x + c^T x + d \]
- at each node, construct the underestimator to from the superset point only
  \[ Q(x) = \min_{i} (x_i^2 + 2x_i z_i + z_i^2) \]
  \[ = \min_{i} (x_i^2 + 2x_i z_i + z_i^2) \quad \text{(simplified notation only)} \]
  \[ = x_i^2 + 2x_i z_i + z_i^2 \quad \text{(quadratic underestimator for } Q(x)) \]

- mãs the quadratic constraint \( x \in \mathbb{R}^n \)
  - reduce \( x_i^2 + 2x_i z_i + z_i^2 \) to quadratic form
    - with linear equality

- x-x is already linear in \( x \)
- x-x is already linear in \( x \)
- x-x is already linear in \( x \)
- x-x is already linear in \( x \)

- the dual objective variable is in the final quadratic form
  - finds linear equality constraints to the original variables
  - finally, subtracts some linear expressions and more linear constraints into the quadratic form