Verified global optimization with Gloptlab

Ferenc Domes, Arnold Neumaier

University of Vienna

19/07/2007
A global optimization problem can be posed as

$$\min \ f(x)$$
$$\text{s.t. } G(x) \in F$$
$$x \in x.$$ 

The function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is called the **objective function**, the $G_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ are called **general constraints** with bounds $F_i$ and the $x_i \in x_i$ are called **bound constraints**.
Basics

- An \( x \in \mathbf{x} \) is called a \textit{feasible point} if \( G(x) \in \mathbf{F} \) is satisfied.

- The problem is called \textit{infeasible} if there are no feasible points.

- A constraint satisfaction problem is an optimization problem with a constant objective function. In this case the task is to find all or one feasible point.
Introduction

- Gloptlab is a global optimization environment, currently capable of handling *constraint satisfaction problems*.

- It is implemented in Matlab and is meant to be a *testing and development platform*.

- The promising algorithms will be integrated in the COCONUT environment, which was developed by Hermann Schichl and Arnold Neumaier.
The SAMPL (M. Markot) input and the internal inequality form is

$$\begin{align*}
\min & \quad A_i:q(x) \\
\text{s.t.} & \quad Aq(x) \in F \quad \text{for some } A \in A, \\
& \quad x \in x, \ x_k := \gamma_k(x_j).
\end{align*}$$

with

$$q(x) = (x_1, \ldots, x_n, x_1^2, \ldots, x_1x_n, \ldots, x_nx_1, \ldots, x_n^2)^T$$

$$i, j \in \{1, \ldots, n_o\}, \ k = n_{o+1}, \ldots, n_x \quad \text{and}$$

$$\gamma_k - \text{nonquadratic univariate functions.}$$
Gloptlab uses various *rigorous* methods to bound the feasible domain.

Using the internal form, rigorous means that each method \( \Gamma : (x, F) \rightarrow (\tilde{x}, \tilde{F}) \) has the property

\[
\{ x \in x \mid Aq(x) \in F \} \subseteq \{ x \in \tilde{x} \mid Aq(x) \in \tilde{F} \}.
\]

- Rigorousity guarantees that *no feasible points are lost*.
- Serious *safety problems* could *arise from losing feasible points*.
- Sometimes having a good approximative solution is not good enough!
Method Selection

The following classes of rigorous methods are currently implemented:

- Problem Simplification
- Linear Relaxations
- Constraint Propagation
- Strict Convex Enclosure
- Conic Methods
- Splitting, Clustering and Hull
Gloptlab GUI

Verified global optimization with Gloptlab

Ferenc Domes, Arnold Neumaier

Basics
Introduction
Problem Definition
Verified Computing
Method Selection
Gloptlab Structure
Demonstration
Implementation
Method details
The end
Toolboxes

- IntLab, developed by Siegfried Rump, allows an easy and sophisticated usage of interval arithmetics.

- The toolbox SeDuMi is an optimization tool over symmetric cones developed by Jos F. Sturm.

- Alternatively SDPT3 from Kim-Chuan Toh, Michael J. Todd, and Reha H. Tutuncu.

- Linear programs are solved with LPSolve by Michel Berkelaar, Jeroen Dirks, Kjell Eikland and Peter Notebaert. SeDuMi and SDPT3 can also be used for LP solving.

- Projected BFGS and conjugate gradient methods from C. T. Kelley.

- AMPL modeling language: Robert Fourer, David Gay and Brian Kernighan.
Features

- Script based execution.
- Easy extension feature.
- Integrated TestEnvironment output.
- Problem database.
- Automatic proof generation.
Working on

- Solving general optimization problems.
- Integrating the non algebraic part.
- New scaling algorithm.
- Comparison with other solvers (Baron, GlobSol, Lingo etc.)
- Decreasing the solution time of the conic programs.
- Automatic strategy selection.
Problem Simplification

- Remove unused variables
- Remove unbounded constraints
- Remove redundant constraints
- Transform into the equality form
- Find additional structural characteristics.
Linear Relaxations

Let $x$ be a finite box with $x \in \mathbf{x}$ and let

$$p(x) := \sum_{k,j=1}^{n} (q_{k,k}x_k^2 + q_{k,j}x_kx_j) + \sum_{k=1}^{n} c_k x_k + d$$

be a constraint of the constraint satisfaction problem.

- We find a linear enclosure of the constraint $p(x)$ at $z \in \mathbf{x}$ such that for all $x \in \mathbf{x}$, $p(x) \in v^T x + \mathbf{w}$ holds.
- Using different linear techniques (Linear CP, solving LPs) we try to obtain better bounds on $x$. 
Constraint Propagation

For each constraint $A_i:q(x) \in F_i$

$$\sum_{i=1}^{n} (a_i x_i + b_i x_i^2) + \sum_{i=1,j=1,i \neq j}^{n} c_{ij} x_i x_j + \sum_{i=1}^{n_u} d_i \gamma_i(x_{J_i}) \in e$$

- Eliminate the bilinear and non-quadratic terms.
- Enclose each quadratic, univariate expression $a_i x_i + b_i x_i^2 \in u_i$ (*forward propagation*).
- Check the consistency of the constraints.
- Find new bounds on the constraint by bounding the sum of the enclosures.
- Get new bound constraints by finding the set of all $x_i$ with $a x_i^2 + b x_i \in u_j$ (*backward propagation*).
Strict Convex Enclosure

By factoring the Hessian $G$ of a strictly convex quadratic multivariate inequality constraint, the constraint can always be brought into the form

$$\|Rx\|_2^2 + 2a^T x \leq \alpha$$

with a nonsingular, lower triangular matrix $R$.

The factorization is done with a directed Cholesky factorization resulting in $G - R^T R$, positive semidefinite and tiny.

This can be translated into $w$ and $\delta$ such that

$$\|Rx\|_2 \leq \delta \quad \text{and} \quad |x| \leq \delta w.$$
**Conic Methods**

Since \( Eq(x) = 0 \) for all feasible \( x \in x \), the equation

\[
\begin{pmatrix} 1 \\ x \end{pmatrix}^T G(y) \begin{pmatrix} 1 \\ x \end{pmatrix} = \begin{pmatrix} 1 \\ \bar{x} - x \end{pmatrix}^T Z \begin{pmatrix} 1 \\ \bar{x} - x \end{pmatrix} + z^T Eq(x) + p(x)
\]

implies \( 0 \leq p(x) \), if \( G(y) \) is positive semidefinite and \( Z \leq 0 \).

Choosing one of

- \( p(x) = \pm x_i + \zeta \) and minimizing \( \zeta \),
- \( p(x) = -\sum_{i=1}^{n} x_i^2 + \zeta \) and minimizing \( \zeta \),
- \( p(x) = -1 \) and minimizing 0,
- \( p(x) = -||\omega \circ x||^2 + 2\xi^T (\omega \circ x) + \delta \) with \( ||\xi|| \leq \zeta \) and minimizing \( \zeta + \delta \),

results in interesting enclosures of the feasible domain.
Branch and Bound

Branch and Bound means that we partition the current box \( x \) of the bound constraints into \( s \) subboxes

\[
x = \{x^1 \cup \cdots \cup x^s\}
\]

and then use rigorous methods \( \Gamma_i(x^k, F) \) on each \( x^k \).

- Recursive splitting results in a finite cover of the feasible domain by nonempty subboxes of a given maximum size.
- The *interval hull* of these subboxes results in new bound constraints.
- Connected components of the union of the subboxes define clusters, which can be separately bounded by their interval hull.
Contact

Please check:
http://www.mat.univie.ac.at/~dferi/
http://www.mat.univie.ac.at/~neum/

For questions please contact me at:
ferenc.domes@univie.ac.at

Thank You!