GloptLab
A Configurable Framework for Global Optimization

Ferenc Domes, Arnold Neumaier
University of Vienna

25.08.2009, Chicago
Introduction
Problem Specification

Optimization problems

\[ \begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad G(x) \in \hat{F}, x \in \hat{x},
\end{align*} \]

with uncertain constraint coefficients can be represented as

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

where \( q(x) = (x, \text{vec}(xx^T)) \) is a quadratic monomial vector and the \( \gamma_k \) are nonquadratic univariate functions.

\textbf{GloptLab}, the Configurable Framework for Global Optimization is designed to solve such problems.
GloptLab

GloptLab is an easy-to-use testing and development platform for solving quadratic optimization problems, written in Matlab.

Various new and state-of-the-art algorithms implemented in GloptLab are used to reduce the search space.

Other techniques, such as finding and verifying feasible points, enable to find the global minimum of the objective function.

All methods in GloptLab are rigorous, hence it is guaranteed that no feasible point is lost.

From the method repertoire custom made strategies can be built, with a user-friendly graphical interface.
Methods
 Verified Computing

**GloptLab** uses various *rigorous* methods to bound the feasible domain.

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

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

- Rigorous methods reduce the search space while guarantee that *no feasible points are lost*.

- In the applications, serious *safety problems* could *arise from losing feasible points* (Gough platform).
Method Features

- Rigorous methods estimate the error for each step in their algorithms and use \textit{directed rounding} or \textit{interval arithmetic}.

- Another way is to find approximate solutions and then verify the results.

- Rigorous computations slow down the solution process, and often require more theoretical work.

- But sometimes having a good approximative solution is not good enough (e.g. computer assisted proofs)!
Method Selection

The following classes of methods are used to rigorously reduce the search space:

- Problem Transformation/Simplification
- Constraint Propagation
- Linear Methods
- Strict Convex Enclosure
- Conic Methods
- Branch and Bound
- Probing, Slicing.

Finding and verifying feasible points combined with a special branch and bound method, enable to find the global minimum of the objective function.
We make use of several external toolboxes to compute approximative solutions of linear, semidefinite or conic programs:

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

IntLab, by Siegfried Rump is used for interval computation while the AMPL modeling language by Robert Fourer, David Gay and Brian Kernighan is used for problem input.
Method References

More details on the implemented methods as well as their mathematical background can be found in various papers on the official GloptLab homepage:

http://www.mat.univie.ac.at/~dferi/gloptlab.html
The Features of GloptLab
Summary of the features

- General and well structured input format.
- Implemented in a completely modular way, allowing easy portability of individual methods.
- Easy to use for prototyping and for development of new techniques in the context of other methods.
- The strategy builder allows us to test different strategies for different problem classes.
- Interactive solution of a particular problem: stop the execution of the strategy, remove and add new tasks, then resume the solution process.
- Contributors can add their own method with only minimal knowledge of the other parts of the software.
- Graphical user interface for building strategies and visualization of the solution process.
- Batch execution mode, Test Environment compatibility.
In order to solve a problem or a list of problems we need a *strategy*.

A strategy is a list of *tasks* used to solve a problem.

A task could be one of the methods listed above, or a *control task*.

The *control tasks* like loops, conditions and breaks extend the functionality and ensure the versatility of a strategy.

Strategies are built comfortably by using the graphical strategy builder.

New methods and solvers are automatically recognized by the strategy builder.
Simple Sample Strategy

1: Read Problem
2: Simplify
3: Feasibility
4: Begin Condition
5: Break
6: End Condition
7: Begin While
8: Propagate
9: Feasibility
10: Begin Condition
11: Break
12: End Condition
13: End While
14: Begin Split
15: Propagate
16: Feasibility
17: Begin Condition
18: Break
19: End Condition
20: End Split
21: Merge
22: Begin Postprocess
23: Merge
24: Feasibility
25: End Postprocess
26: Pause
27: Finish
Complex Sample Strategy

1: Read Problem
2: Simplify
3: Ehull
4: Linear
5: Feasibility
6: Begin Condition
7: Break
8: End Condition
9: Conic
10: Begin While
11: Propagate
12: Linear
13: Feasibility
14: Begin Condition
15: Break
16: End Condition
17: End While
18: Begin Split
19: Propagate
20: Linear
21: Feasibility
22: Begin Condition
23: Break
24: End Condition
25: End Split
26: Merge
27: Begin Split
28: Propagate
29: Linear
30: Feasibility
31: Begin Condition
32: Break
33: End Condition
34: End Split
35: Begin Postprocess
36: Merge
37: Feasibility
38: End Postprocess
39: Pause
40: Finish
Problem solving

When a strategy has been built it can be used to solve a specific problem or a list of problems.

Solving can be started either by using the batch solution mode or directly in the *graphical user interface* of GloptLab:
The graphical user interface consists of areas for entering problems, for defining strategies, for displaying the solver progress and for configuring GloptLab.

The interactive solution of a particular problem in the graphical user interface: it is possible to stop the execution of the strategy, remove and add new tasks to it and then resume the solution process.

Manipulating the method parameters, experimenting with different combinations of tasks can greatly improve the solution results and lead to more knowledge about building effective strategies.
Demonstration
We used the **Test Environment** to test and compare some **GloptLab** strategies.

Library **Lib3** of the **Coconut** Environment Testset containing 308 constraint satisfaction problems has used, 63 of them was classified as hard problems the other as easy ones.

The two sample strategies have been configured to accept only problems with less than 100 variables and used to solve the library.

The maximal time allowed for the solution of a single problem was 120 seconds.
### Test Results of the Simple Sample Strategy

<table>
<thead>
<tr>
<th>stat</th>
<th>all</th>
<th>wr</th>
<th>easy loc.</th>
<th>hard loc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>+G</td>
<td>-G</td>
</tr>
<tr>
<td>all</td>
<td>308</td>
<td>0</td>
<td>121</td>
<td>124</td>
</tr>
<tr>
<td>G</td>
<td>125</td>
<td>0</td>
<td>111</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>76</td>
<td>0</td>
<td>0</td>
<td>57</td>
</tr>
<tr>
<td>TU</td>
<td>95</td>
<td>0</td>
<td>8</td>
<td>59</td>
</tr>
<tr>
<td>U</td>
<td>12</td>
<td>0</td>
<td>2</td>
<td>8</td>
</tr>
</tbody>
</table>

#### GloptLab summary statistics

<table>
<thead>
<tr>
<th>lib</th>
<th>all</th>
<th>accept</th>
<th>+G</th>
<th>G!</th>
<th>G?</th>
<th>I?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib3</td>
<td>308</td>
<td>232</td>
<td>135</td>
<td>125</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
# Test Results of the Complex Sample Strategy

**GloptLab on Lib3 (Complex Sample Strategy)**

<table>
<thead>
<tr>
<th>stat</th>
<th>all</th>
<th>wr</th>
<th>easy loc.</th>
<th>hard loc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>+G</td>
<td>-G</td>
</tr>
<tr>
<td>all</td>
<td>308</td>
<td>0</td>
<td>130</td>
<td>115</td>
</tr>
<tr>
<td>G</td>
<td>139</td>
<td>0</td>
<td>120</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>76</td>
<td>0</td>
<td>0</td>
<td>57</td>
</tr>
<tr>
<td>TU</td>
<td>85</td>
<td>0</td>
<td>10</td>
<td>52</td>
</tr>
<tr>
<td>U</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

**GloptLab summary statistics**

<table>
<thead>
<tr>
<th>lib</th>
<th>all</th>
<th>accept</th>
<th>+G</th>
<th>G!</th>
<th>G?</th>
<th>I?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib3</td>
<td>308</td>
<td>232</td>
<td>149</td>
<td>139</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Evaluation of the Test Results

We have found 135 correct solutions (125 of them was claimed as correct) by using the first strategy and 149 correct solutions (139 of them was claimed as correct) by using the second one.

Within the same allowed solution time we solved approximately 10 percent more problems with the second strategy than with the first one.

35 percent more hard problems was solved by using the second strategy!

The significant difference was caused by the more sophisticated methods and the clever structure of the second strategy.
Perspectives and Conclusions
Perspectives

- Integrating the non algebraic, univariate functions.
- Testing and improving the optimization part.
- Enhancing the existing methods and developing new ones.
- Implementing promising methods in the Coconut Environment.
- Comparison with other solvers (ICOS, Realpaver, Baron, GlobSol, etc.)
- Automatic, intelligent strategy selection.
Conclusions

External contributors are welcome to join the project by implementing and testing their own user-defined methods. User-defined methods submitted to us will be permanently added to the method repertoire of future versions of GloptLab if they are promising enough.

I would like to thank Arnold Neumaier for his help and support.

This research was founded by the grant FSP 506/003 of the University of Vienna.
References

GL: http://www.mat.univie.ac.at/~dferi/gloptlab.html
TEST ENVIRONMENT: http://www.mat.univie.ac.at/~dferi/
COCO: http://www.mat.univie.ac.at/coconut-environment
iCOS: http://ylebbah.googlepages.com/icos
INTLAB: http://www.ti3.tu-harburg.de/~rump/intlab/
LpSOLVE: http://lpsolve.sourceforge.net/5.5/
SeDuMI: http://sedumi.ie.lehigh.edu/
AMPL: http://www.ampl.com/

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

Thank You for your attention!