The Optimization Test Environment

Martin Fuchs

CERFACS, Toulouse, France

in collaboration with Ferenc Domes, Hermann Schichl,
University of Vienna, Austria

September 3, 2010

Motivation

Testing is crucial for users who seek to

1. adjust solver parameters
   → improve the configuration of a particular solver
2. compare solvers on single problems
   → find a good solution of a specific problem
3. compare solvers on suitable test sets
   → solver benchmarks with large test sets

The Test Environment automatizes many of these (possibly little exciting) procedures.

Study background

Comparisons of global optimization solvers:

- typically for black box problems only
- many interfaces of test environments are not very user friendly
- extensive study in 2005, based on the COCONUT Environment benchmark

! Generally, much effort has to be spent to conduct a solver comparison

Basic testing procedure

1. organize (possibly large) test libraries
2. solve selected problems with selected solvers
3. analyze, check and compare the results
4. create summary tables (e.g., in LaTeX) and performance profiles

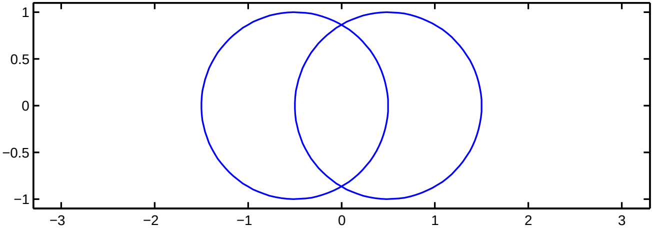
Notation for optimization problems

$$\left.\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathbf{x}, \\ & F(x) \in \mathbf{F}, \\ & x_i \in \mathbb{Z} \text{ for } i \in I. \end{array}\right\} \qquad (1)$$

- $\mathbf{x} = [\underline{x}, \overline{x}] = \{x \in \mathbb{R}^n \mid \underline{x} \leq x \leq \overline{x}\}$ is a **box** in $\mathbb{R}^n$
- $f : \mathbf{x} \to \mathbb{R}$ is the **objective function**
- $F : \mathbf{x} \to \mathbb{R}^m$ is a vector of **constraint functions**
- $\mathbf{F}$ is a box in $\mathbb{R}^m$
- $I \subseteq \{1, \ldots, n\}$ defines the **integer components** of x

Test problem examples

$$\left.\begin{aligned} \min_x \;& x_2 \\ \text{s.t.} \;& (x_1-0.5)^2+x_2^2=1, \\ & (x_1+0.5)^2+x_2^2=1, \\ & x_1\in[-3,3], x_2\in[-3,3]. \end{aligned}\right\} \quad (2)$$



$$\rightarrow \widehat{x} = (0,-\sqrt{3}/2)^T \approx (0,-0.8660)^T$$

AMPL formulation

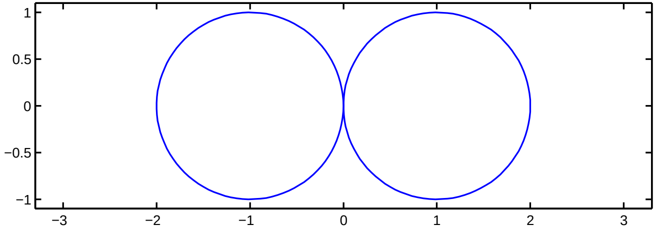
- ```
  var x1 >=-3, <=3;
  var x2 >=-3, <=3;

  minimize obj:  x2;

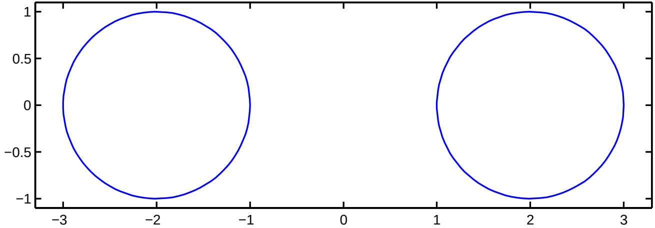
  s.t.  c1:  (x1-0.5)^2+x2^2=1;
  s.t.  c2:  (x1+0.5)^2+x2^2=1;
  ```

- our problem representations are DAGs (COCONUT Environment)
- converters to DAGs are available online for popular modeling languages (e.g., AMPL, GAMS)
  ```

Test problem examples ctd.



$\longrightarrow$ $x = (0,0)^T$ is the only feasible point



$\longrightarrow$ no feasible solution

Interactive library management

[demo]

Solver interface

1. solver selection
2. solver configuration
   → setup paths and scripts
3. test problem subset selection
   → criteria editor
4. solve command
   → solve selected problems with selected solvers

→ [demo]

Analyze

- translate solver results to a unified format

$\longrightarrow$ .res files

`modelstatus`	solver model status
`x(i)`	solver output for $\widehat{x}_i$, $i = 1, \ldots, n$
`obj`	solver output for $f(\widehat{x})$
`infeas`	feasibility distance provided by the solver
`nonopt`	0 if $\widehat{x}$ claimed to be at least locally optimal, 1 otherwise
`time`	used CPU time to solve the problem
`splits`	number of splits made, e.g., in branching algorithms

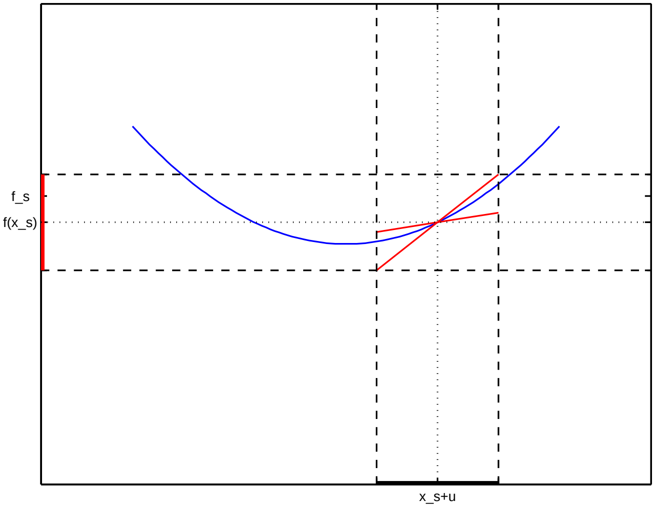
Solution check

- let $C := \{x \in \mathbf{x} \mid x_I \in \mathbb{Z}, F(x) \in \mathbf{F}\}$
- compute a feasibility distance d_{feas}
- intuitively: $\min_{y \in C} \|x - y\|_p$
- computationally: componentwise feasibility
- account for scaling issues via the following interval computations

Feasibility distance

- parameters ε $(:= 10^{-6})$, κ $(:= 1)$, α $(:= 0)$
- solver result x_s, f_s
- $\mathbf{u} = [-\varepsilon \max(\|x_s\|_\infty, \kappa), \varepsilon \max(\|x_s\|_\infty, \kappa)]$
- $\mathbf{x}_s = x_s + \mathbf{u}$
- objective violation
  $v_o(x_s, f_s) := \langle f(x_s) + f'(\mathbf{x}_s)(\mathbf{x}_s - x_s) - f_s \rangle$
- constraint violations
  $v_c(x_s, f_s) := \langle F(x_s) + F'(\mathbf{x}_s)(\mathbf{x}_s - x_s) - \mathbf{F} \rangle$
- box constraint violations
  $v_b(x_s, f_s) := \langle \mathbf{x}_s - \mathbf{x} \rangle$
- **feasibility distance**
  $\mathrm{d}_{\mathrm{feas,p}}(x_s, f_s) := \|(v_o(x_s, f_s), v_c(x_s, f_s), v_b(x_s, f_s))^T\|_p$
  (we use $p = \infty$)
- if $\mathrm{d}_{\mathrm{feas,p}}(x_s, f_s) \le \alpha \Rightarrow x_s$ **feasible**

Illustration of objective violation



Comparison of results

- $J := J_x \times J_f$, with
  $J_x := \{x_1, \ldots, x_N\}$, $J_f := \{f_1, \ldots, f_N\}$:
  solver results that passed the check

- **global numerical solution** $(x_{\text{opt}}, f_{\text{opt}})$:
  $(x_{\text{opt}}, f_{\text{opt}}) \in J$ and $f_{\text{opt}} \leq f_j$ for all $f_j \in J_f$

- $(\widetilde{x}, \widetilde{f}) \in J$, $\widetilde{x} \neq x_{\text{opt}}$ is also considered global if $\widetilde{f}$ is sufficiently close to f_{opt}, i.e.,

$$\widetilde{f} \leq \begin{cases} f_{\text{opt}} + \beta & \text{if } |f_{\text{opt}}| \leq \kappa, \\ f_{\text{opt}} + \beta\,|f_{\text{opt}}| & \text{otherwise} \end{cases},$$

  with tolerance β ($:= 10^{-6}$).

- otherwise $\widetilde{x}$ is a **local numerical solution**

Comparison of results ctd.

- **best point**:

$$(x_{\text{best}}, f_{\text{best}}) = \begin{cases} (x_{\text{opt}}, f_{\text{opt}}) & \text{if } J \neq \emptyset, \\ \arg\min_{(x,f)} \mathrm{d}_{\text{feas,p}}(x, f) & \text{if } J = \emptyset \end{cases}$$

- if the best point was found by a default local solver $\rightarrow$ **easy location**
- otherwise $\rightarrow$ **hard location**

Performance assessment

Count for each solver

- the number of global numerical solutions found
- the number of wrong and correct solver claims

Further performance measures for rigorous or black box solvers will be implemented in the future.

Summary tables and performance profiles

- automated creation of summary tables of the results in LaTeX and performance profiles in JPG

→ just copy paste them to result sections of scientific publications

→ [demo]

Benchmark

COCONUT Environment benchmark:

- three libraries of global constrained optimization and constraint satisfaction problems
- over 1000 test problems in up to 20000 variables
- time limits depending on the problem size:

size	n	**timeout** (s)
1	1-9	180
2	10-99	900
3	100-999	1800
4	≥ 1000	.

We collect solver benchmark results online.

Benchmark results

baron summary statistics										
library	all	acc	wr	G+	G!	I!	W	G?	L?	I?
Lib1	264	228	38	146	82	1	34	2	0	2
Lib2	715	635	43	414	232	2	23	16	1	3
Lib3	307	274	11	252	252	5	8	0	0	3
total	1286	1137	92	812	566	8	65	18	1	8

coin summary statistics										
library	all	acc	wr	G+	G!	I!	W	G?	L?	I?
Lib1	264	239	6	138	2	0	6	0	0	0
Lib2	715	674	46	439	20	0	35	5	6	0
Lib3	307	298	3	215	3	0	2	0	1	0
total	1286	1211	55	792	25	0	43	5	7	0

Reliability analysis

solver	G+/aF+	G!/G+	wr/acc
baron	73%	69%	8%
cocos	60%	76%	12%
coin	66%	3%	4%
conopt	56%	4%	8%
knitro	49%	0%	21%
lindoglobal	67%	79%	15%
minos	58%	4%	3%
pathnlp	57%	3%	3%

Conclusions

Easy-to-use management of

- test libraries and problem subset selection
- solver configurations

Automatized

- solution check and comparison
- LaTeX code creation
- performance profiles

Final remarks

The benchmark results show

- BARON performed best as a global solver
- COIN performed best as a local solver

Visit my website for updates

http://www.martin-fuchs.net