(MI)NLPLib 2

Stefan Vigerske

GAMS

4th September 2014

MAGO 2014, Málaga

Model instance collections

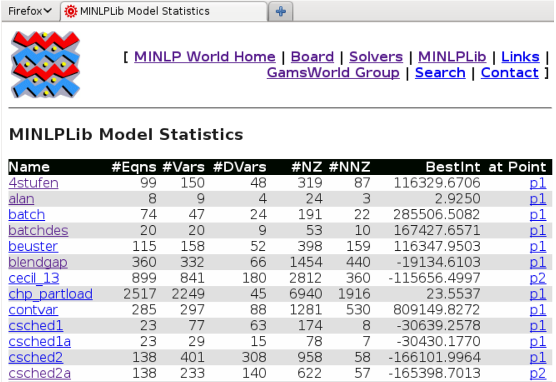
Collecting optimization problems has been a popular "hobby" for long time, e.g.,

first release	library	problem types
1985	Netlib	Linear Programming
1992	MIPLIB	Mixed-Integer Programming
1993	CUTE	Nonlinear Programming
1998	SDPLib	Semidefinite Programming
1999	CSPLib	Constraint Satisfaction Programming
199x	MacMINLP	Mixed-Integer Nonlinear Programming
2001	GAMS World	LP, MIP, NLP, MINLP, SOCP, MPEC
2003	COCONUT	Nonlinear and Constraint Satisfaction Programming
2008	mintOC	Mixed-Integer Optimal Control
2009	minlp.org	MINLP, General Disjunctive Programming
2011	POLIP	Mixed-Integer Polynomial Programming
2014	CBLIB	Conic Programming

- for solver developers, access to a wide set of interesting problem instances with different characteristics has always been important
- commercial solver vendors test their solver on thousands of test problems before releasing a new software version
- the evaluation of algorithmic improvements (w.r.t. robustness and efficiency) requires well-balanced test sets of significantly many real-world instances

MINLPLib and GLOBALLib http://www.gamsworld.org

- Initiated in 2001 (as part of GamsWorld/MinlpWorld/GlobalWorld):
  M. Bussieck, A. Drud, and A. Meeraus
  MINLPLib – A Collection of Test Models for Mixed-Integer Nonlinear Programming
  INFORMS Journal on Computing 15, 114–119 (2003)
- "white-box" NLPs (GLOBALLib) and MINLPs (MINLPLib)


Firefox
MINLPLib Model Statistics
[MINLP World Home | Board | Solvers | MINLPLib | Links |
GamsWorld Group | Search | Contact]

MINLPLib Model Statistics

Name	#Eqns	#Vars	#DVars	#NZ	#NNZ	BestInt	at Point
4stufen	99	150	48	319	87	116329.6706	p1
alan	8	9	4	24	3	2.9250	p1
batch	74	47	24	191	22	285506.5082	p1
batchdes	20	20	9	53	10	167427.6571	p1
beuster	115	158	52	398	159	116347.9503	p1
blendgap	360	332	66	1454	440	-19134.6103	p1
cecil_13	899	841	180	2812	360	-115656.4997	p2
chp_partload	2517	2249	45	6940	1916	23.5537	p1
contvar	285	297	88	1281	530	809149.8272	p1
csched1	23	77	63	174	8	-30639.2578	p1
csched1a	23	29	15	78	7	-30430.1770	p1
csched2	138	401	308	958	58	-166101.9964	p1
csched2a	138	233	140	622	57	-165398.7013	p2

- frequently used for testing, but also benchmarking

MINLPLib and GLOBALLib Instances

- scalar GAMS format

```
Variables  x1,x2,b3,b4,b5,objvar;
Positive Variables x1,x2;
Binary Variables b3,b4,b5;
Equations  e1,e2,e3,e4,e5,e6;
e1..  - 2*x1 - 3*x2 - 1.5*b3 - 2*b4 + 0.5*b5 + objvar =E= 0;
e2.. sqr(x1) + b3 =E= 1.25;
e3.. x2**1.5 + 1.5*b4 =E= 3;
e4..    x1 + b3 =L= 1.6;
e5..    1.333*x2 + b4 =L= 3;
e6..  - b3 - b4 + b5 =L= 0;
```

MINLPLib and GLOBALLib Instances

- scalar GAMS format

```
Variables  x1,x2,b3,b4,b5,objvar;
Positive Variables x1,x2;
Binary Variables b3,b4,b5;
Equations  e1,e2,e3,e4,e5,e6;
e1..  - 2*x1 - 3*x2 - 1.5*b3 - 2*b4 + 0.5*b5 + objvar =E= 0;
e2.. sqr(x1) + b3 =E= 1.25;
e3.. x2**1.5 + 1.5*b4 =E= 3;
e4..    x1 + b3 =L= 1.6;
e5..    1.333*x2 + b4 =L= 3;
e6..  - b3 - b4 + b5 =L= 0;
```

- varying from small scale (great for debugging!) to large scale real world instances (agricultural economics, chemical-, civil-, and electrical engineering, finance, management, OR)
- intentionally including instances from badly formulated models or different formulations of the same problem

MINLPLib and GLOBALLib Instances

- scalar GAMS format

```
Variables  x1,x2,b3,b4,b5,objvar;
Positive Variables x1,x2;
Binary Variables b3,b4,b5;
Equations  e1,e2,e3,e4,e5,e6;
e1..  - 2*x1 - 3*x2 - 1.5*b3 - 2*b4 + 0.5*b5 + objvar =E= 0;
e2.. sqr(x1) + b3 =E= 1.25;
e3.. x2**1.5 + 1.5*b4 =E= 3;
e4..    x1 + b3 =L= 1.6;
e5..    1.333*x2 + b4 =L= 3;
e6..  - b3 - b4 + b5 =L= 0;
```

- varying from small scale (great for debugging!) to large scale real world instances (agricultural economics, chemical-, civil-, and electrical engineering, finance, management, OR)
- intentionally including instances from badly formulated models or different formulations of the same problem
- including solution points for many instances

MINLPLib and GLOBALLib Instances

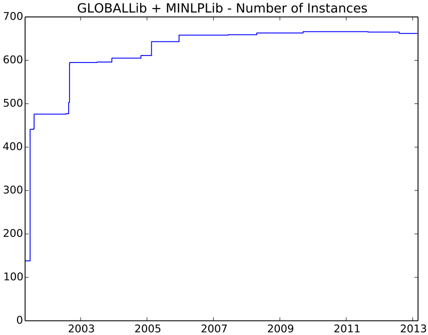
- scalar GAMS format

```
Variables  x1,x2,b3,b4,b5,objvar;
Positive Variables x1,x2;
Binary Variables b3,b4,b5;
Equations  e1,e2,e3,e4,e5,e6;
e1..  - 2*x1 - 3*x2 - 1.5*b3 - 2*b4 + 0.5*b5 + objvar =E= 0;
e2.. sqr(x1) + b3 =E= 1.25;
e3.. x2**1.5 + 1.5*b4 =E= 3;
e4..    x1 + b3 =L= 1.6;
e5..    1.333*x2 + b4 =L= 3;
e6..  - b3 - b4 + b5 =L= 0;
```

- varying from small scale (great for debugging!) to large scale real world instances (agricultural economics, chemical-, civil-, and electrical engineering, finance, management, OR)
- intentionally including instances from badly formulated models or different formulations of the same problem
- including solution points for many instances
- solely an instance collection, i.e., consisting of instantiations of models by specific data sets

MINLPLib and GLOBALLib History

- instances were harvested from existing collections, initially:
  - GAMS Model Library
  - MacMINLP (Leyffer)
  - MINOPT library (Floudas)
  - Handbook of Test Problems in Local and Global Optimization (Floudas et.al.)
- 2001 – 2009: maintained by Michael Bussieck
- new instances were added
- new incumbent solutions were added
- in 2009: Michael “volunteered” me as maintainer



Letters to the Editor

So which requirements should the next generation MINLPLib meet?

Letters to the Editor

So which requirements should the next generation MINLPLib meet?

P.B. (2007)

Just a side question. I would be interested in having the list of the convex models in minlpLib since you seem to know them.

C.D. (2009)

Is it somewhere available the convexity informations on (part of) the MINLPLib instances? I found something in Nowak's book and for few more instances, but I think it would be useful...

Letters to the Editor

So which requirements should the next generation MINLPLib meet?

P.B. (2007)

Just a side question. I would be interested in having the list of the convex models in minlpLib since you seem to know them.

C.D. (2009)

Is it somewhere available the convexity informations on (part of) the MINLPLib instances? I found something in Nowak's book and for few more instances, but I think it would be useful...

T.L. (2011)

For 50 problems AlphaECP can find a better solution than reported in the MinlpLib if the absolute tolerance of 0.001 is used, however, in most cases it is only due to the lower tolerance. Do you think it would appropriate to specify a solution tolerance for an accepted solution for the MinlpLib problems and those solutions that does not fulfill the tolerance would be disregarded, like fo7_2?

T.L. (2011)

I would like to run a test on the convex MINLP problems in the MinlpLib. Do you happen to have a list of which of the problems are convex? I would greatly appreciate this information.

T.L. (2011)

I want to remind/inform you about the following link where convex MINLP problems can be found (problems from this page could be added to the MINLP library and, furthermore, classified as convex).

Letters to the Editor (cont.)

A.D. (2011)

> *I have been running some experiments with nuclear10a and nuclear10b from minlplib. I think that the formulation is pretty poor, but it is not possible to experiment with the scalar format in the library. Do you know where I can find the original (pre-Convert) version of these models?*

Letters to the Editor (cont.)

A.D. (2011)

I have been running some experiments with nuclear10a and nuclear10b from minlplib. I think that the formulation is pretty poor, but it is not possible to experiment with the scalar format in the library. Do you know where I can find the original (pre-Convert) version of these models?

R.M. (2012)

The attached document describes our sources for each of the 173 test cases that are not already in GLOBALLib or MINLPLib. The additional 173 test cases described in the PDF file are either already publicly available (e.g., on MINLP.org) or described in the open literature. Because none of the models are proprietary, they may be posted in any public forum such as GLOBALLib or MINLPLib.

Letters to the Editor (cont.)

A.D. (2011)

I have been running some experiments with nuclear10a and nuclear10b from minlplib. I think that the formulation is pretty poor, but it is not possible to experiment with the scalar format in the library. Do you know where I can find the original (pre-Convert) version of these models?

R.M. (2012)

The attached document describes our sources for each of the 173 test cases that are not already in GLOBALLib or MINLPLib. The additional 173 test cases described in the PDF file are either already publicly available (e.g., on MINLP.org) or described in the open literature. Because none of the models are proprietary, they may be posted in any public forum such as GLOBALLib or MINLPLib.

N.S. (2012)

The following two problems seem identical:

```
http://www.gamsworld.org/minlp/minlplib/detf1.htm
http://www.gamsworld.org/minlp/minlplib/saa_2.htm
```

N.S. (2013)

I found a number of problems that appear to be duplicates. These are listed below in terms of identical pairs for GlobalLib, MINLPLib and PrincetonLib

```
MINLPLib
nuclear14b.gms nuclear24b.gms
```

...

Letters to the Editor (cont.)

Y.S. (Dec. 2012)

> *By the way, do you have any MINLP instance which is good to solve on HLRN II supercomputer? If you have it, I can try to solve it because [...] I still have NPLs for this year.*

Letters to the Editor (cont.)

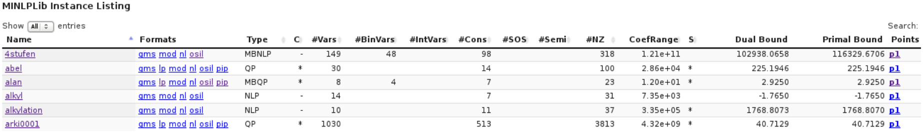
Y.S. (Dec. 2012)

By the way, do you have any MINLP instance which is good to solve on HLRN II supercomputer? If you have it, I can try to solve it because [...] I still have NPLs for this year.

A.P. (2013)

I am currently working on my thesis about nonconvex MINLPs. Looking for some instances I found the MINLPLib. But I do not know which of these instances are nonconvex and could not find the information on the webpage.

MINLPLib 2



MINLPLib Instance Listing

Show All entries

Search:

Name	Formats	Type	C	#Vars	#BinVars	#IntVars	#Cons	#SOS	#Semi	#NZ	CoefRange	S	Dual Bound	Primal Bound	Points
4stufen	gms mod nl osil	MBNLP	-	149	48		98			318	1.21e+11		102938.0658	116329.6706	p1
abel	gms lp mod nl osil pip	QP	*	30			14			100	2.86e+04	*	225.1946	225.1946	p1
alan	gms lp mod nl osil pip	MBQP	*	8	4		7			23	1.20e+01	*	2.9250	2.9250	p1
alkyl	gms mod nl osil	NLP	-	14			7			31	7.35e+03		-1.7650	-1.7650	p1
alkylation	gms mod nl osil	NLP	-	10			11			37	3.35e+05	*	1768.8073	1768.8070	p1
arki0001	gms lp mod nl osil pip	QP	*	1030			513			3813	4.32e+09	*	40.7129	40.7129	p1

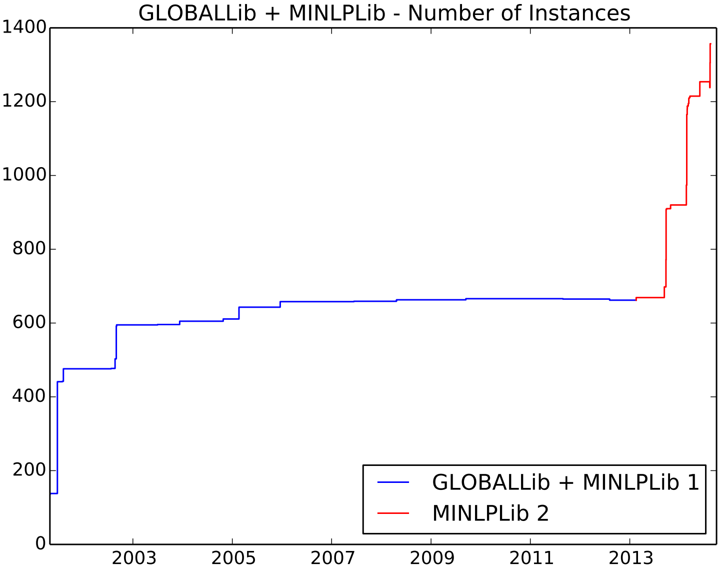
Tasks:

- Adding new problem instances:
  - both convex and nonconvex problems
  - (MI)QPs, (MI)QCQPs, and (MI)NLPs
  - easy solvable, solvable, difficult to solve, but not trivial
- Categorizing instances
  - convexity
  - problem type (quadratic, polynomial, general nonlinear)
  - function types (powers, exp/log, trigonometric, ...)
  - solved to global optimality?
- Providing feasible best known solutions

Work in progress, current version publicly available:
`http://www.gamsworld.org/minlp/minlplib2/html/index.html`.

Learn from MIPLIB2010 (Koch et.al., 2011) work.

New NLP and MINLP Instances

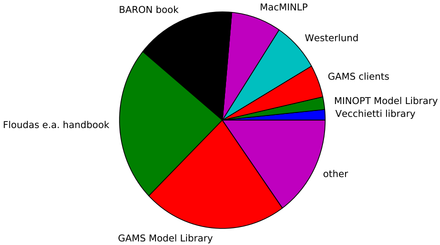


Sources of newly added instances

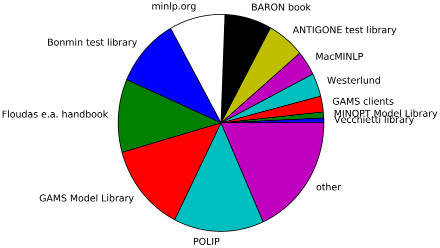
Harvesting mainly from

- CMU-IBM open source MINLP project (convex MINLPs)
- minlp.org
- POLIP (polynomial MINLPs)

GLOBALLib + MINLPLib 1 instance sources (637 in total)



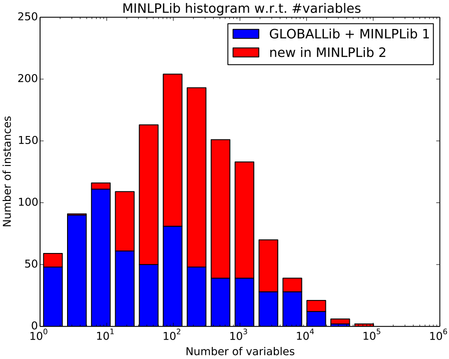
MINLPLib 2 instance sources (1357 in total)

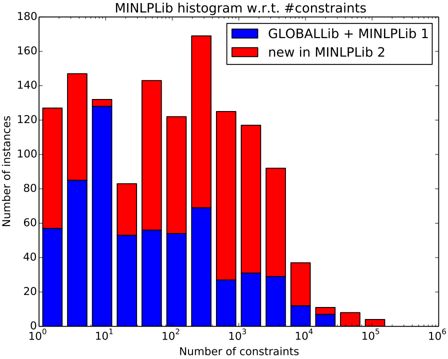


Instance Formats

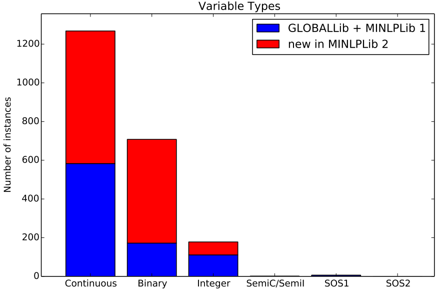
Format		#instances	
GAMS	`.gms`	**1357**	
AMPL	`.mod`	1331	(no errorf/signpower/...)
AMPL	`.nl`	1319	(no errorf/signpower/..., crash in parsing)
OSIL	XML	1336	(no signpower/...)
CPLEX LP	`.lp`	662	(limited to quadratics)
PIP	`.pip`	764	(limited to polynomial)

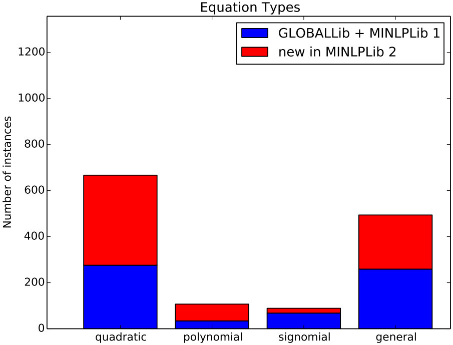
Instance Sizes

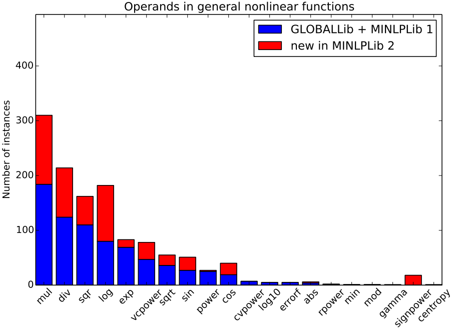




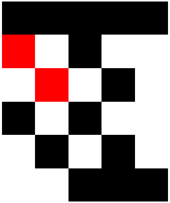
Problem types


Variable Types


Equation Types


Operands in general nonlinear functions

Sparsity Pattern



Objective Gradient and Jacobian:

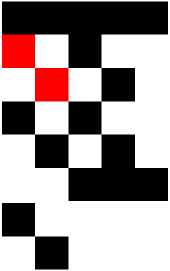
- first row objective gradient, below the Jacobian
- black pixel for linear coefficients (constant deriv.)
- red pixel for nonlinear coefficients (nonconstant deriv.)

```
Variables  x1,x2,b3,b4,b5,objvar;
Equations  e1,e2,e3,e4,e5,e6;
e1..  - 2*x1 - 3*x2 - 1.5*b3 - 2*b4 + 0.5*b5 + objvar =E= 0;
e2.. sqr(x1) + b3 =E= 1.25;
e3.. x2**1.5 + 1.5*b4 =E= 3;
e4..    x1 + b3 =L= 1.6;
e5..    1.333*x2 + b4 =L= 3;
e6..  - b3 - b4 + b5 =L= 0;
```

Sparsity Pattern

Objective Gradient and Jacobian:

- first row objective gradient, below the Jacobian
- black pixel for linear coefficients (constant deriv.)
- red pixel for nonlinear coefficients (nonconstant deriv.)

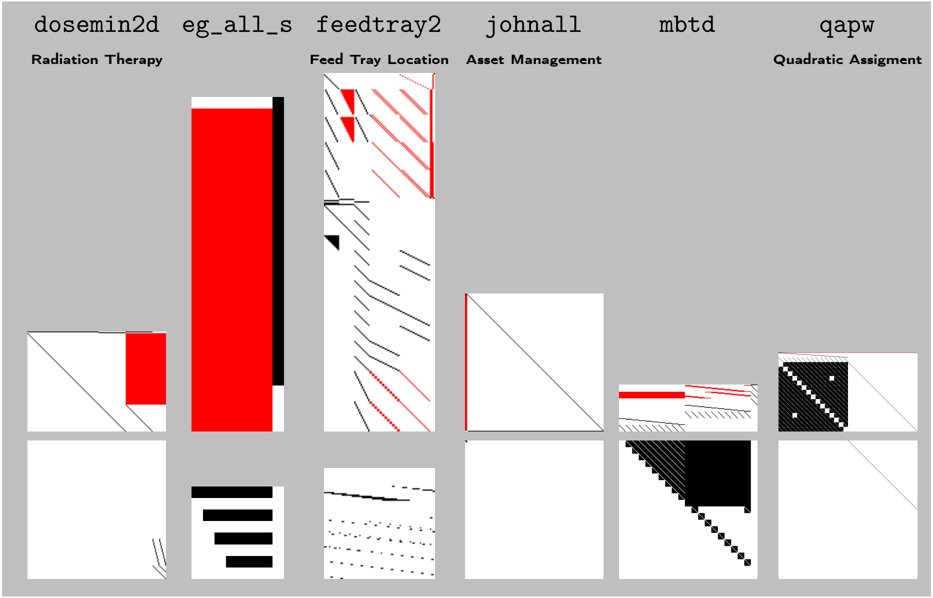


Hessian of Lagrangian:

- black pixel if two variables appear together in a nonlinear term

```
Variables  x1,x2,b3,b4,b5,objvar;
Equations  e1,e2,e3,e4,e5,e6;
e1..  - 2*x1 - 3*x2 - 1.5*b3 - 2*b4 + 0.5*b5 + objvar =E= 0;
e2.. sqr(x1) + b3 =E= 1.25;
e3.. x2**1.5 + 1.5*b4 =E= 3;
e4..    x1 + b3 =L= 1.6;
e5..    1.333*x2 + b4 =L= 3;
e6..  - b3 - b4 + b5 =L= 0;
```

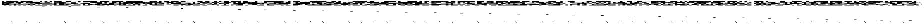
Sparsity Pattern – Examples



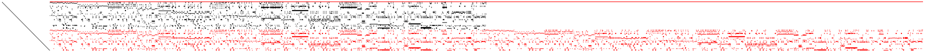
(top: Objective Gradient and Jacobian; bottom: Lagrangian Hessian)

Sparsity Pattern – Examples (cont.)

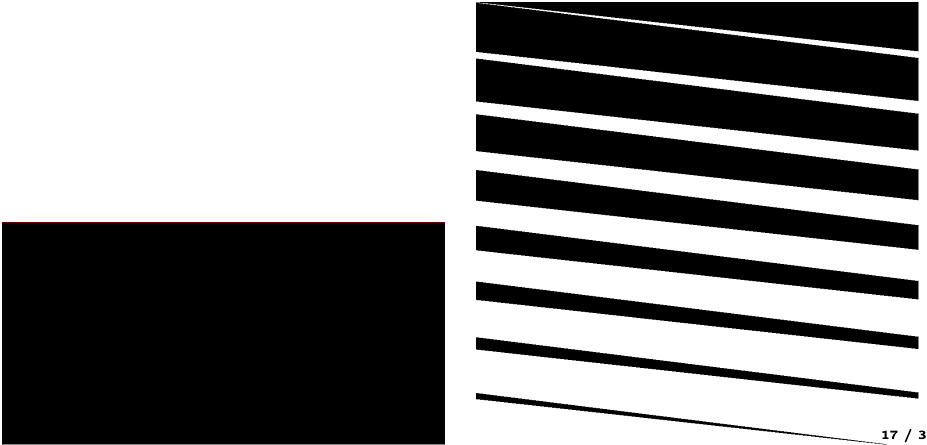
Jacobian `densitymod` **(Density modification based on single-crystal X-ray diffraction data; 23529 vars, 550 cons.)**



Jacobian `lop97ic` **(Rail Line Optimization, MIQCQP)**



`milinfract` **(Solving Mixed-Integer Linear Fractional Programming Problems with Dinkelbach's Algorithm)**



Objective Gradient + Jacobian

Lagrangian Hessian

(Non)Convexity Detection

- Convex (MI)NLPs are considered “easier” (though, still difficult) than nonconvex ones.
- Several solvers require convexity to guarantee global optimality.

(Non)Convexity Detection

- Convex (MI)NLPs are considered “easier” (though, still difficult) than nonconvex ones.
- Several solvers require convexity to guarantee global optimality.

- In the context of MINLPLib, we say a problem is convex if the objective function and each nonlinear constraint function is convex w.r.t. the bound constraints (easiest to verify, other notions exist).

(Non)Convexity Detection

- Convex (MI)NLPs are considered "easier" (though, still difficult) than nonconvex ones.
- Several solvers require convexity to guarantee global optimality.
- In the context of MINLPLib, we say a problem is convex if the objective function and each nonlinear constraint function is convex w.r.t. the bound constraints (easiest to verify, other notions exist).
- How to prove or disprove convexity?

(Non)Convexity Detection for Functions

Analyze the Hessian:

- Given twice differentiable function $h(x)$ and variable bounds $[\underline{x}, \bar{x}]$.
- Compute the spectrum of the Hessian in one random point and conclude
  - convexity/concavity/indefiniteness if $h(x)$ is quadratic
  - nonconvexity/nonconcavity if $h(x)$ is general nonlinear

(Non)Convexity Detection for Functions

Analyze the Hessian:

- Given twice differentiable function $h(x)$ and variable bounds $[\underline{x}, \bar{x}]$.
- Compute the spectrum of the Hessian in one random point and conclude
  - convexity/concavity/indefiniteness if $h(x)$ is quadratic
  - nonconvexity/nonconcavity if $h(x)$ is general nonlinear

Analyze the Algebraic Expression:

$$
\begin{aligned}
f(x) \text{ convex} &\Rightarrow\ a \cdot f(x) \begin{cases} \text{convex}, & a \geq 0 \\ \text{concave}, & a \leq 0 \end{cases} \\
f(x), g(x) \text{ convex} &\Rightarrow\ f(x) + g(x) \text{ convex} \\
f(x) \text{ concave} &\Rightarrow\ \log(f(x)) \text{ concave} \\
f(x) = \prod_i x_i^{e_i}, x_i \geq 0 &\Rightarrow\ f(x) \begin{cases} \text{convex}, & e_i \leq 0\ \forall i \\ \text{convex}, & \exists j : e_i \leq 0\ \forall i \neq j;\ \sum_i e_i \geq 1 \\ \text{concave}, & e_i \geq 0\ \forall i;\ \sum_i e_i \leq 1 \end{cases}
\end{aligned}
$$

(Non)Convexity Detection for Functions

Analyze the Hessian:

- Given twice differentiable function $h(x)$ and variable bounds $[\underline{x}, \bar{x}]$.
- Compute the spectrum of the Hessian in one random point and conclude
  - convexity/concavity/indefiniteness if $h(x)$ is quadratic
  - nonconvexity/nonconcavity if $h(x)$ is general nonlinear

Analyze the Algebraic Expression:

$$f(x) \text{ convex} \Rightarrow a \cdot f(x) \begin{cases} \text{convex}, & a \geq 0 \\ \text{concave}, & a \leq 0 \end{cases}$$

$$f(x), g(x) \text{ convex} \Rightarrow f(x) + g(x) \text{ convex}$$

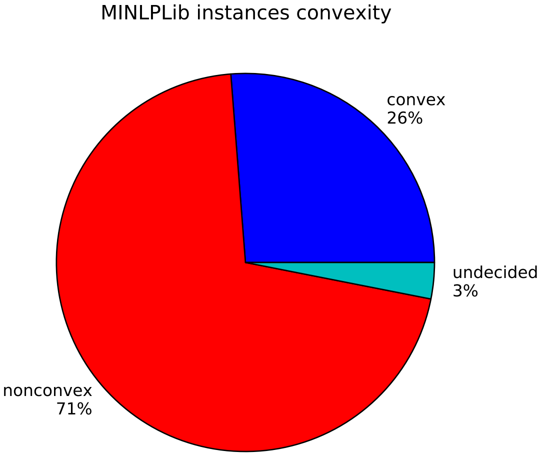
$$f(x) \text{ concave} \Rightarrow \log(f(x)) \text{ concave}$$

$$f(x) = \prod_i x_i^{e_i}, x_i \geq 0 \Rightarrow f(x) \begin{cases} \text{convex}, & e_i \leq 0 \; \forall i \\ \text{convex}, & \exists j : e_i \leq 0 \; \forall i \neq j; \; \sum_i e_i \geq 1 \\ \text{concave}, & e_i \geq 0 \; \forall i; \; \sum_i e_i \leq 1 \end{cases}$$

Analyze manually.

(Non)Convexity in MINLPLib

- Numerical Analysis of Hessians by LAPACK.
- Symbolic Analysis of Expressions by SCIP.
- Mark additional 71 instances (5%) as convex.



Solution Points

MINLPLib instances traditionally come with known feasible solution points.

Solution Points

MINLPLib instances traditionally come with known feasible solution points.

For MINLPLib 2, we added

Feasibility checking:

- compute maximal (unscaled) violation of constraints, variable bounds, and discreteness restrictions
- uses GAMS/EXAMINER2

Solution Points

MINLPLib instances traditionally come with known feasible solution points.

For MINLPLib 2, we added

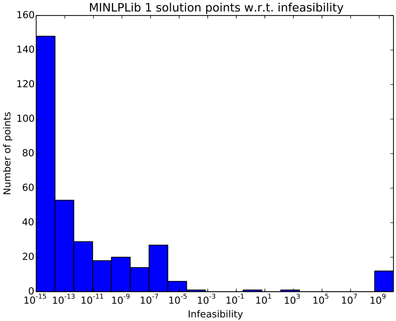
Feasibility checking:

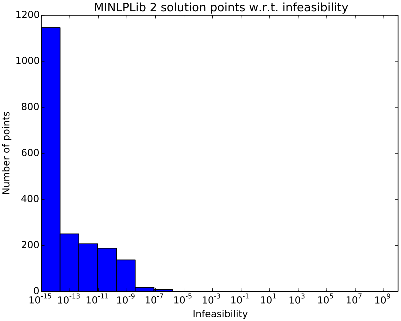
- compute maximal (unscaled) violation of constraints, variable bounds, and discreteness restrictions
- uses GAMS/EXAMINER2

Solution polishing: For a given point,

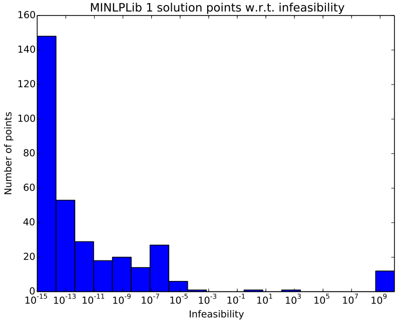
1. project onto variable bounds
2. round values for discrete variables to exact integers
3. ensure that semicontinuity/semiintegrality and special-ordered-set constraints are exactly satisfied
4. run CONOPT on MINLP with all binary/integer/semi*/SOS variables fixed, start from updated point, scaling disabled, feasibility tolerance 10^{-9}

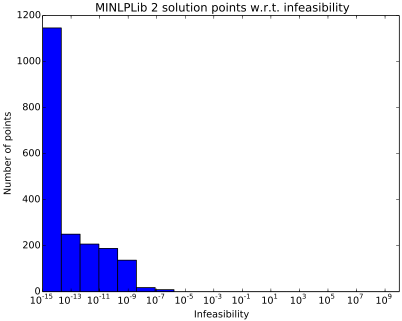
Polished Solution Points





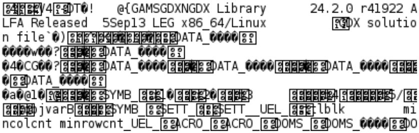
Polished Solution Points





Available in two formats:

GAMS Data Exchange (GDX)



```
����V4��DT�!   @{GAMSGDXNGDX Library      24.2.0 r41922 A
LFA Released  5Sep13 LEG x86_64/Linux        ��DX solutio
n file`�)����������������DATA_����
����w��?����DATA_����
�4�CG��?����DATA_��������DATA_��������DATA_��������
���DATA_����
�a�@1���������SYMB_���������������3     ��������4��������5/��
�����jvarB������SYMB_���SETT_���SETT__UEL_����tlblk        mi
ncolcnt minrowcnt_UEL_���ACRO_���ACRO_���DOMS_���DOMS_���������DO
```

ASCII (.sol)

```
x1      1.1180339887498901754
x2      1.3103706971044497739
b4      1.0000000000000000000
b5      1.0000000000000000000
objvar  7.6671800688131304113
```

Dual Bounds

$$\text{dual bound} = \begin{cases} \text{lower bound on optimal value,} & \text{if minimization} \\ \text{upper bound on optimal value,} & \text{if maximization} \end{cases}$$

Dual Bounds

$$\text{dual bound} = \begin{cases} \text{lower bound on optimal value,} & \text{if minimization} \\ \text{upper bound on optimal value,} & \text{if maximization} \end{cases}$$

Collected dual bounds from

- solvers for general (MI)NLP
  (ANTIGONE, BARON, Couenne, Lindo, SCIP)
- solvers for convex MINLP on proven convex MINLPs
  (AlphaECP, Bonmin BB, Bonmin Hyb)

Dual Bounds

$$\text{dual bound} = \begin{cases} \text{lower bound on optimal value,} & \text{if minimization} \\ \text{upper bound on optimal value,} & \text{if maximization} \end{cases}$$

Collected dual bounds from

- solvers for general (MI)NLP (ANTIGONE, BARON, Couenne, Lindo, SCIP)
- solvers for convex MINLP on proven convex MINLPs (AlphaECP, Bonmin BB, Bonmin Hyb)

But: No way to verify correctness of bound!

YOU MEAN IT´S NOT A GLOBAL OPTIMUM?
BUT I CHECKED ALL CONSTRAINT KOALIFICATIONS
memegenerator.net

Dual Bounds

$$\text{dual bound} = \begin{cases} \text{lower bound on optimal value,} & \text{if minimization} \\ \text{upper bound on optimal value,} & \text{if maximization} \end{cases}$$

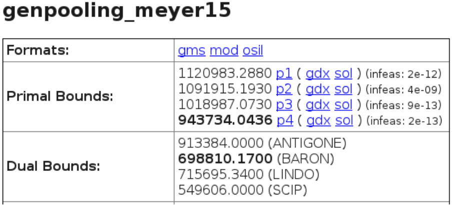
Collected dual bounds from

- solvers for general (MI)NLP (ANTIGONE, BARON, Couenne, Lindo, SCIP)
- solvers for convex MINLP on proven convex MINLPs (AlphaECP, Bonmin BB, Bonmin Hyb)

But: No way to verify correctness of bound!

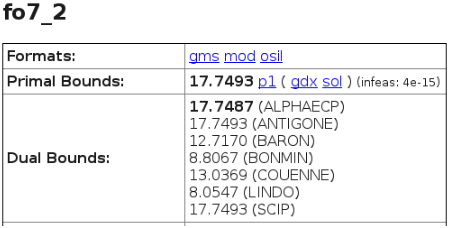
YOU MEAN IT´S NOT A GLOBAL OPTIMUM?
BUT I CHECKED ALL CONSTRAINT KOALIFICATIONS
memegenerator.net

Conservative approach: Only trust a solvers dual bound claim if it has been verified by at least 2 other solvers.



genpooling_meyer15

Formats:	gms mod osil
Primal Bounds:	1120983.2880 p1 (gdx sol) (infeas: 2e-12) 1091915.1930 p2 (gdx sol) (infeas: 4e-09) 1018987.0730 p3 (gdx sol) (infeas: 9e-13) **943734.0436** p4 (gdx sol) (infeas: 2e-13)
Dual Bounds:	913384.0000 (ANTIGONE) **698810.1700** (BARON) 715695.3400 (LINDO) 549606.0000 (SCIP)

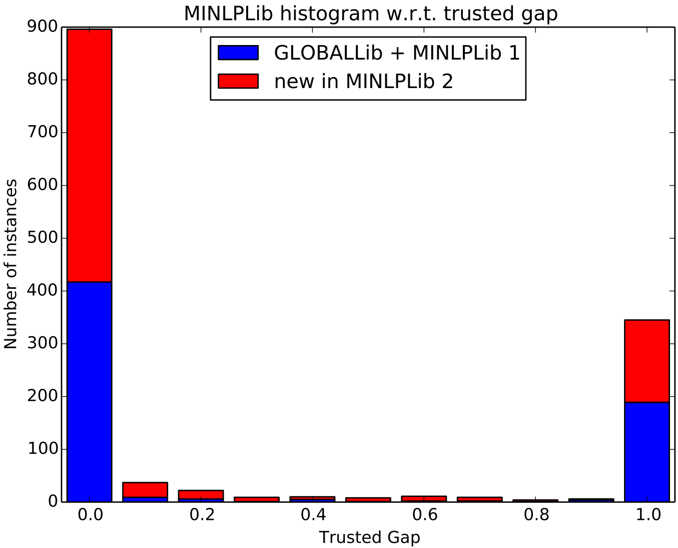


fo7_2

Formats:	gms mod osil
Primal Bounds:	**17.7493** p1 (gdx sol) (infeas: 4e-15)
Dual Bounds:	**17.7487** (ALPHAECP) 17.7493 (ANTIGONE) 12.7170 (BARON) 8.8067 (BONMIN) 13.0369 (COUENNE) 8.0547 (LINDO) 17.7493 (SCIP)

“Open” instances

Feasible solution points ⊕ trusted dual bounds ⇒ trusted gap



$0.0 \triangleq \leq 10^{-9} \qquad 1.0 \triangleq \geq 1.0$

Query the MINLPLib

Simple script to select instances by specific criteria, e.g.:

- all large convex instances, show # var. and # cons.:

  ```
  $ ./query.py "(nvars > 4242) & (convex == True)" -c nvars -c ncons
                           nvars   ncons
  jbearing100               5304       0
  squfl030-150              4530    4650
  watercontamination0202  106711  107209
  watercontamination0303  107222  108217
  ```

- all quadratic instances:

  ```
  ./query.py "npolynomfunc == 0 & nsignomfunc == 0 & ngennlfunc == 0"
  ```

- all instances with trigonometric functions:

  ```
  ./query.py "(opsin == True) or (opcos == True)"
  ```

- all separable instances, sorted by problem type:

  ```
  ./query.py "nlaghessiannz == nlaghessiandiagnz" -s probtype -c probtype
  ```

- all unsolved instances (w.r.t. “trusted” dual bounds), zipped up:

  ```
  ./query.py "gap > 0.1" -c gap -z open.zip
  ```

Solver Benchmark?

Can we use the current 1357 instances to make statements about the performance of general purpose solvers?

- ✓ large number of instances
- ✓ wide variety of applications
- × dominance of certain models, e.g.,
  - 32 block layout design problems
  - 60 small investor portfolio optimization instances
  - ...
- × many trivial, some hopeless, some numerically dubious instances

Solver Benchmark?

Can we use the current 1357 instances to make statements about the performance of general purpose solvers?

- ✓ large number of instances
- ✓ wide variety of applications
- × dominance of certain models, e.g.,
  - 32 block layout design problems
  - 60 small investor portfolio optimization instances
  - ...
- × many trivial, some hopeless, some numerically dubious instances

Let's try it anyway...

Solvers, etc.

Consider the 5 global solvers from GAMS 24.3.1:

- ANTIGONE 1.1
  by R. Misener (Imperial) and Ch. Floudas (Princeton)
- BARON 14.0.2
  by N. Sahinidis (CMU), M. Tawarmalani (Purdue), et.al.
- Couenne 0.4
  by P. Belotti (now FICO), et.al.; open-source (COIN-OR)
- Lindo API 8.0.1694.550
  by Lindo Systems Inc.
- SCIP 3.1 #695c979
  by Zuse Institute Berlin, et.al.; free for academic use

Timelimit: 15 min

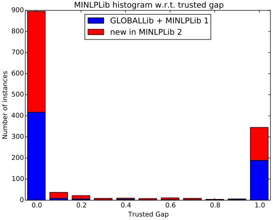
Hardware: Dell PowerEdge M1000e, 48GB RAM, Intel Xeon X5672@3.2GHz

Metric

Gold standard: time to proof of optimality

However:

- difficult to verify, as optimal value not known for every instance
- many timeouts when using only 15min timelimit

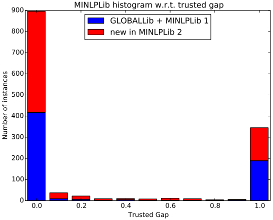


Metric

Gold standard: time to proof of optimality

However:

- difficult to verify, as optimal value not known for every instance
- many timeouts when using only 15min timelimit



Instead, look at primal bound (objective value of feasible solution) only . . .

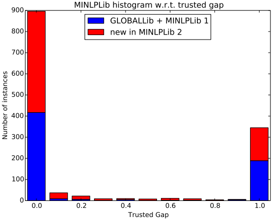
- gap to optimal value when terminating?
  - disregards time completely
- time to find an optimal solution?
  - disregards good previously found solutions

Metric

Gold standard: time to proof of optimality

However:

- difficult to verify, as optimal value not known for every instance
- many timeouts when using only 15min timelimit

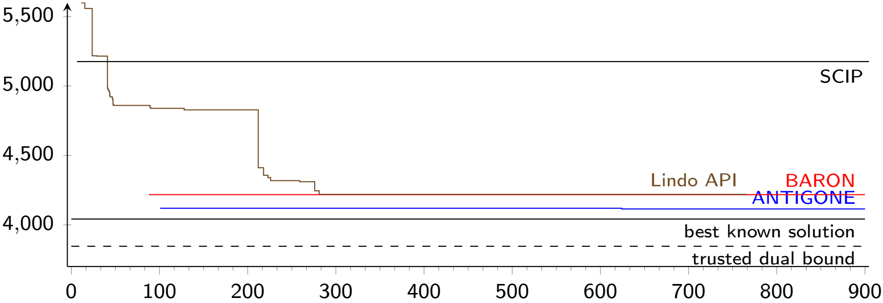


Instead, look at primal bound (objective value of feasible solution) only ...

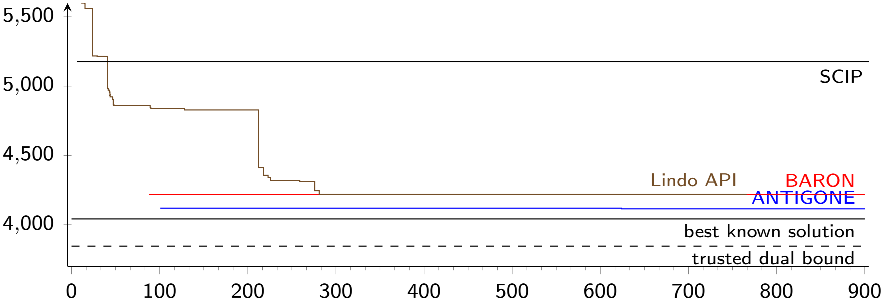
- gap to optimal value when terminating?
  - disregards time completely
- time to find an optimal solution?
  - disregards good previously found solutions

⇒ evaluate progress of primal bound over time

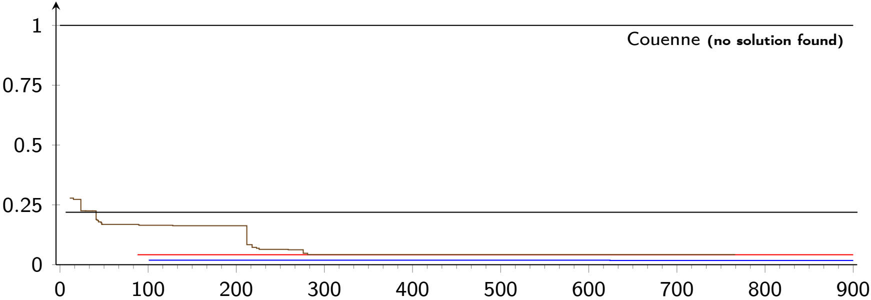
Primal Bound over Time (Instance: lop97ic)



Primal Bound over Time (Instance: lop97ic)



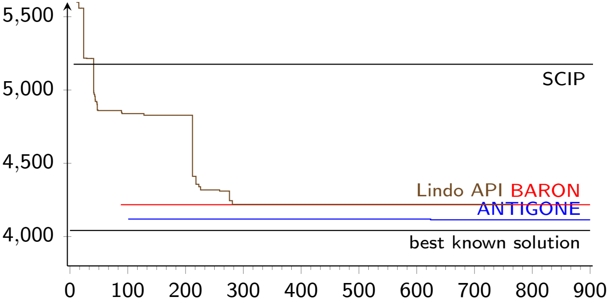
Scaling w.r.t. best known solution:



Primal Integral

Berthold [2013] propses: integrate relative improvement of primal bound over time

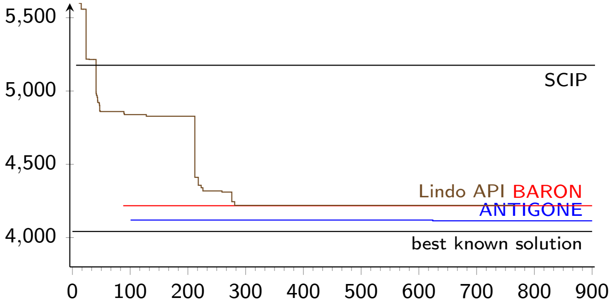
$$\sum_{i=1}^{n}(t_i - t_{i-1})\,\mathrm{gap}(\mathrm{pb}(t),\mathrm{opt}),\ \text{where } \mathrm{gap}(\mathrm{pb},\mathrm{opt}) = \begin{cases} 0, & \mathrm{pb} = \mathrm{opt} \\ 1, & \mathrm{pb}\cdot\mathrm{opt} < 0, \\ 1, & \mathrm{pb} = \infty, \\ \frac{|\mathrm{pb}-\mathrm{opt}|}{\max(|\mathrm{pb}|,|\mathrm{opt}|)} \end{cases}$$



Primal Integral

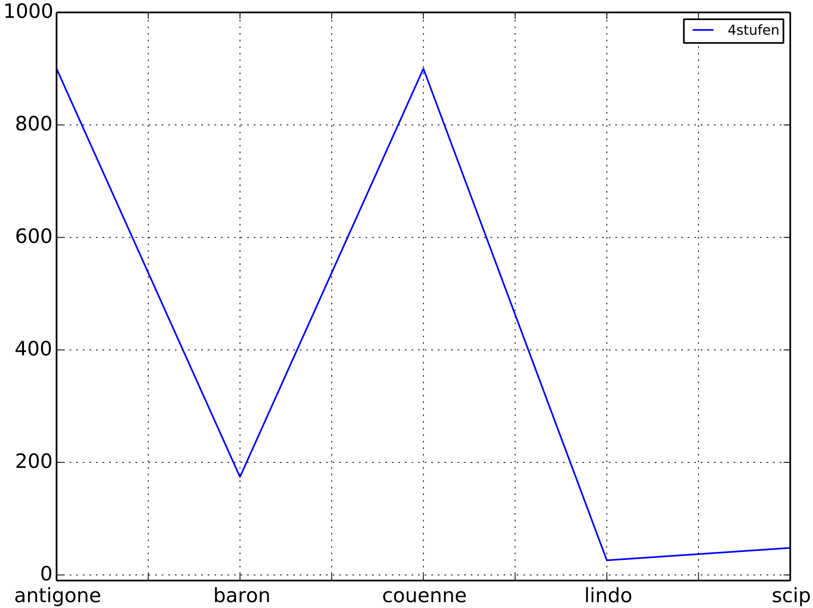
Berthold [2013] propses: integrate relative improvement of primal bound over time

$$\sum_{i=1}^{n}(t_i - t_{i-1})\,\mathrm{gap}(\mathrm{pb}(t),\mathrm{opt}),\ \text{where}\ \mathrm{gap}(\mathrm{pb},\mathrm{opt}) = \begin{cases} 0, & \mathrm{pb} = \mathrm{opt} \\ 1, & \mathrm{pb}\cdot\mathrm{opt} < 0, \\ 1, & \mathrm{pb} = \infty, \\ \frac{|\mathrm{pb}-\mathrm{opt}|}{\max(|\mathrm{pb}|,|\mathrm{opt}|)} & \end{cases}$$

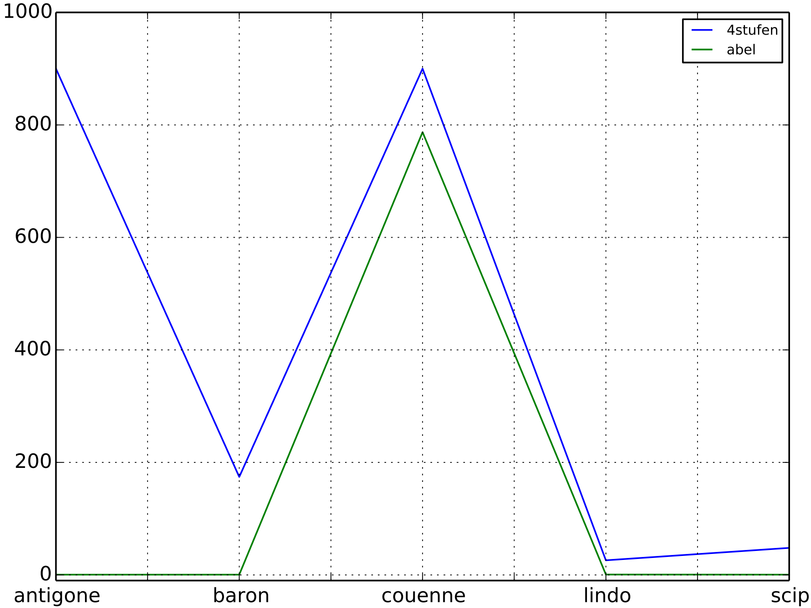


ANTIGONE	115.50
BARON	121.82
Couenne	900.00
Lindo API	77.32
SCIP	204.15

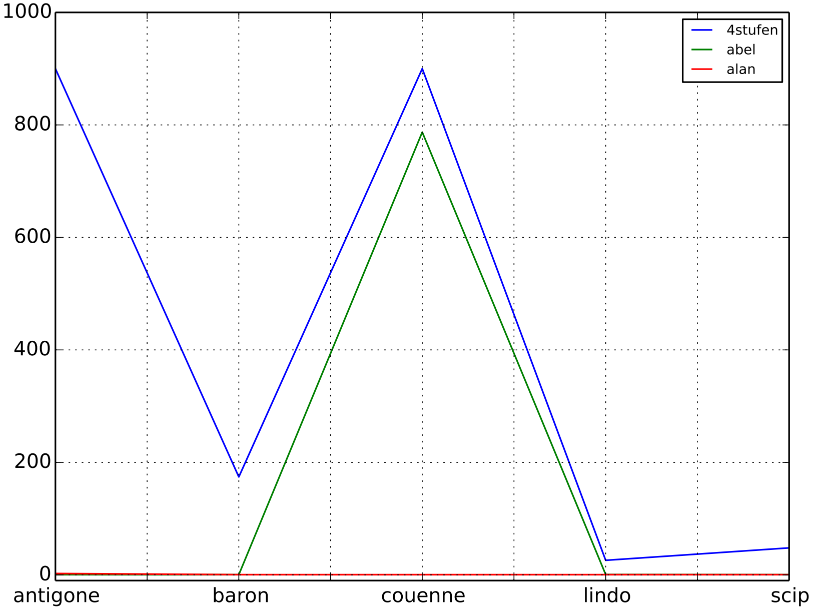
Primal Integrals on MINLPLib 2 (one line per instance)



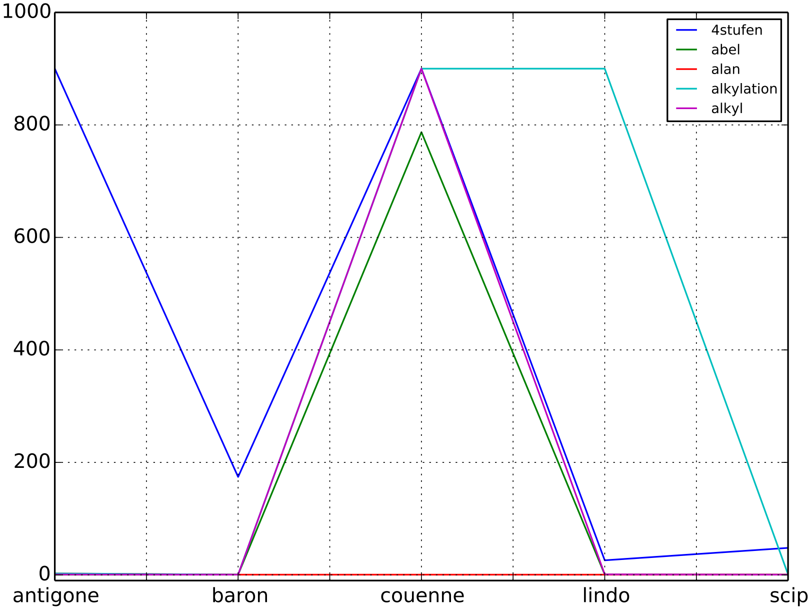
Primal Integrals on MINLPLib 2 (one line per instance)



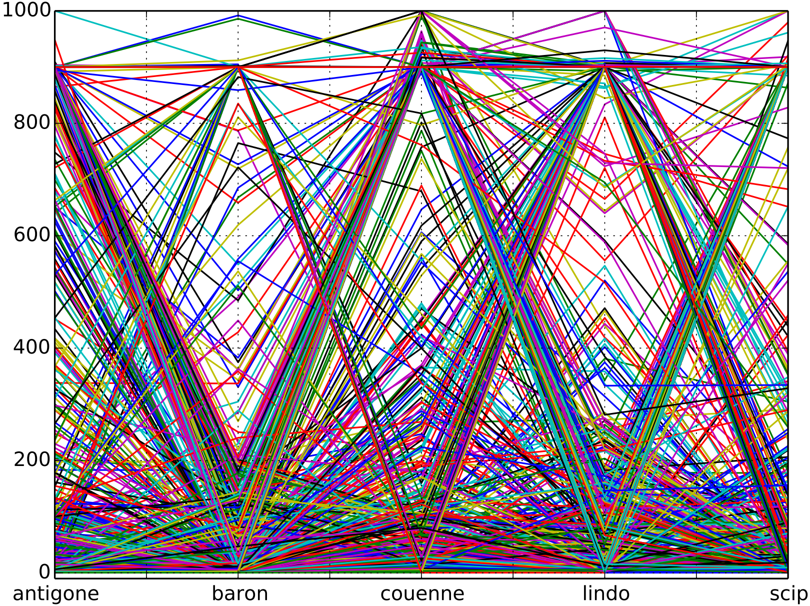
Primal Integrals on MINLPLib 2 (one line per instance)



Primal Integrals on MINLPLib 2 (one line per instance)



Primal Integrals on MINLPLib 2 (one line per instance)

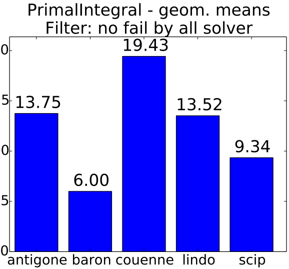


Average Primal Integrals on MINLPLib 2

Consider only instances

- that could be handled by every solver (no exotic operands or variable types)
- where no solver crashed
- where no solver reported an infeasible point as feasible (tolerance: 10^{-6})

all:



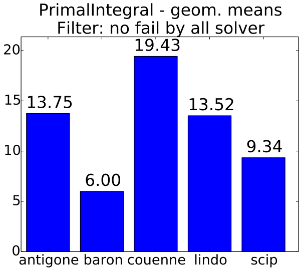
994 instances

Average Primal Integrals on MINLPLib 2

Consider only instances

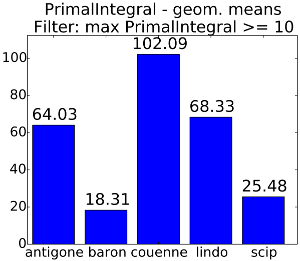
- that could be handled by every solver (no exotic operands or variable types)
- where no solver crashed
- where no solver reported an infeasible point as feasible (tolerance: 10^{-6})

all:



994 instances

exclude trivial instances:



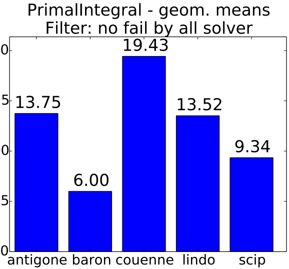
639 instances

Average Primal Integrals on MINLPLib 2

Consider only instances

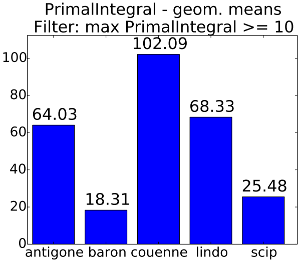
- that could be handled by every solver (no exotic operands or variable types)
- where no solver crashed
- where no solver reported an infeasible point as feasible (tolerance: 10^{-6})

all:



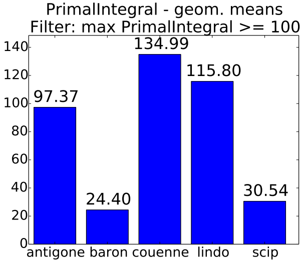
994 instances

exclude trivial instances:



639 instances

exclude trivial and easy:



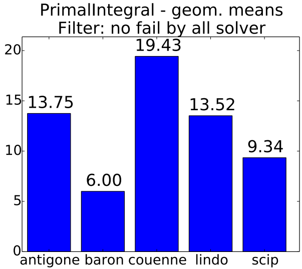
522 instances

Average Primal Integrals on MINLPLib 2

Consider only instances

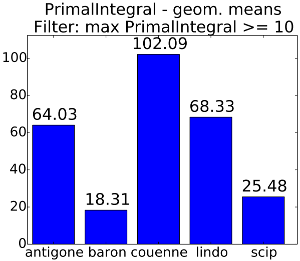
- that could be handled by every solver (no exotic operands or variable types)
- where no solver crashed
- where no solver reported an infeasible point as feasible (tolerance: 10^{-6})

all:



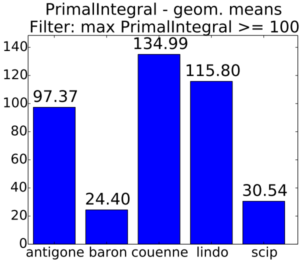
994 instances

exclude trivial instances:



639 instances

exclude trivial and easy:

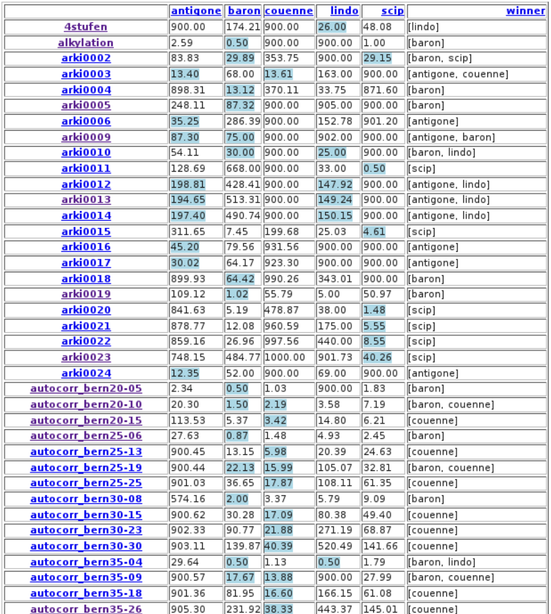


522 instances

However, keep in mind that the testset is unbalanced!

Does a certain solver outperform others on certain optimization problems?

Does a certain solver outperform others on certain optimization problems?



	antigone	baron	couenne	lindo	scip	winner
4stufen	900.00	174.21	900.00	26.00	48.08	[lindo]
alkylation	2.59	0.50	900.00	900.00	1.00	[baron]
arki0002	83.83	29.89	353.75	900.00	29.15	[baron, scip]
arki0003	13.40	68.00	13.61	163.00	900.00	[antigone, couenne]
arki0004	898.31	13.12	370.11	33.75	871.60	[baron]
arki0005	248.11	87.32	900.00	905.00	900.00	[baron]
arki0006	35.25	286.39	900.00	152.78	901.20	[antigone]
arki0009	87.30	75.00	900.00	902.00	900.00	[antigone, baron]
arki0010	54.11	30.00	900.00	25.00	900.00	[baron, lindo]
arki0011	128.69	668.00	900.00	33.00	0.50	[scip]
arki0012	198.81	428.41	900.00	147.92	900.00	[antigone, lindo]
arki0013	194.65	513.31	900.00	149.24	900.00	[antigone, lindo]
arki0014	197.40	490.74	900.00	150.15	900.00	[antigone, lindo]
arki0015	311.65	7.45	199.68	25.03	4.61	[scip]
arki0016	45.20	79.56	931.56	900.00	900.00	[antigone]
arki0017	30.02	64.17	923.30	900.00	900.00	[antigone]
arki0018	899.93	64.42	990.26	343.01	900.00	[baron]
arki0019	109.12	1.02	55.79	5.00	50.97	[baron]
arki0020	841.63	5.19	478.87	38.00	1.48	[scip]
arki0021	878.77	12.08	960.59	175.00	5.55	[scip]
arki0022	859.16	26.96	997.56	440.00	8.55	[scip]
arki0023	748.15	484.77	1000.00	901.73	40.26	[scip]
arki0024	12.35	52.00	900.00	69.00	900.00	[antigone]
autocorr_bern20-05	2.34	0.50	1.03	900.00	1.83	[baron]
autocorr_bern20-10	20.30	1.50	2.19	3.58	7.19	[baron, couenne]
autocorr_bern20-15	113.53	5.37	3.42	14.80	6.21	[couenne]
autocorr_bern25-06	27.63	0.87	1.48	4.93	2.45	[baron]
autocorr_bern25-13	900.45	13.15	5.98	20.39	24.63	[couenne]
autocorr_bern25-19	900.44	22.13	15.99	105.07	32.81	[baron, couenne]
autocorr_bern25-25	901.03	36.65	17.87	108.11	61.35	[couenne]
autocorr_bern30-08	574.16	2.00	3.37	5.79	9.09	[baron]
autocorr_bern30-15	900.62	30.28	17.09	80.38	49.40	[couenne]
autocorr_bern30-23	902.33	90.77	21.88	271.19	68.87	[couenne]
autocorr_bern30-30	903.11	139.87	40.39	520.49	141.66	[couenne]
autocorr_bern35-04	29.64	0.50	1.13	0.50	1.79	[baron, lindo]
autocorr_bern35-09	900.57	17.67	13.88	900.00	27.99	[baron, couenne]
autocorr_bern35-18	901.36	81.95	16.60	166.15	61.08	[couenne]
autocorr_bern35-26	905.30	231.92	38.33	443.37	145.01	[couenne]

Best Primal Integral ↔ Optimization Problem

ANTIGONE

- Multiperiod Blend Scheduling (blend*)
- Extended Pooling Problem (pooling_epa*)
- Generalized Pooling (genpooling*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (m6, m7*)
- Facility Location (sfacloc*)
- Supply Chain Design (supplychain*)
- Water Network Design (waterund*, watertreat*)

LINDO API

- Facility Location (emfl*)
- Portfolio Optimization DAX (smallinvDAX*)
- Synthesis of Space Truss (space25*)
- Uncapacitated Facility Location (squfl*)
- Telecom. Network Design (telecomsp*)
- Water Network Contamination (watercontamination*)

BARON

- Facility Layout (m6, m7*, o7,8,9*)
- Farm Layout (flay*)
- Block Layout (fo*, no*)
- Graph partitioning (graphpart*)
- Circle/Polygon/Rectangle Packing (kall_*)
- Point Packing (pointpack*)
- Product Portfolio Optimization (kport*)
- Retrofit Planning (rsyn*)
- Safety Layout (slay*)

SCIP

- Block Layout (no*)
- Cutting Stock (carton*)
- Cross-dock Door Assignment (crossdock*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (o7,8,9*)
- General Quadratic Assignment (pb*)
- Quadratic Linear Ordering: bipart. graphs edge crossing (edgecross*), Facility Layout (faclay*)
- Multicommodity network design (ndcc*)
- Portfolio Optimization S&P100 (smallinvSNP*)
- Stochastic Service System Design (sssd*)
- Trimloss (tln*, tls*)

Best Primal Integral ↔ Optimization Problem

ANTIGONE

- Multiperiod Blend Scheduling (blend*)
- Extended Pooling Problem (pooling_epa*)
- Generalized Pooling (genpooling*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (m6, m7*)
- Facility Location (sfacloc*)
- Supply Chain Design (supplychain*)
- Water Network Design (waterund*, watertreat*)

LINDO API

- Facility Location (emfl*)
- Portfolio Optimization DAX (smallinvDAX*)
- Synthesis of Space Truss (space25*)
- Uncapacitated Facility Location (squfl*)
- Telecom. Network Design (telecomsp*)
- Water Network Contamination (watercontamination*)

BARON

- Facility Layout (m6, m7*, o7,8,9*)
- Farm Layout (flay*)
- Block Layout (fo*, no*)
- Graph partitioning (graphpart*)
- Circle/Polygon/Rectangle Packing (kall_*)
- Point Packing (pointpack*)
- Product Portfolio Optimization (kport*)
- Retrofit Planning (rsyn*)
- Safety Layout (slay*)

SCIP

- Block Layout (no*)
- Cutting Stock (carton*)
- Cross-dock Door Assignment (crossdock*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (o7,8,9*)
- General Quadratic Assignment (pb*)
- Quadratic Linear Ordering: bipart. graphs edge crossing (edgecross*), Facility Layout (faclay*)
- Multicommodity network design (ndcc*)
- Portfolio Optimization S&P100 (smallinvSNP*)
- Stochastic Service System Design (sssd*)
- Trimloss (tln*, tls*)

Best Primal Integral ↔ Optimization Problem

ANTIGONE

- Multiperiod Blend Scheduling (blend*)
- Extended Pooling Problem (pooling_epa*)
- Generalized Pooling (genpooling*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (m6, m7*)
- Facility Location (sfacloc*)
- Supply Chain Design (supplychain*)
- Water Network Design (waterund*, watertreat*)

LINDO API

- Facility Location (emfl*)
- Portfolio Optimization DAX (smallinvDAX*)
- Synthesis of Space Truss (space25*)
- Uncapacitated Facility Location (squfl*)
- Telecom. Network Design (telecomsp*)
- Water Network Contamination (watercontamination*)

BARON

- Facility Layout (m6, m7*, o7,8,9*)
- Farm Layout (flay*)
- Block Layout (fo*, no*)
- Graph partitioning (graphpart*)
- Circle/Polygon/Rectangle Packing (kall_*)
- Point Packing (pointpack*)
- Product Portfolio Optimization (kport*)
- Retrofit Planning (rsyn*)
- Safety Layout (slay*)

SCIP

- Block Layout (no*)
- Cutting Stock (carton*)
- Cross-dock Door Assignment (crossdock*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (o7,8,9*)
- General Quadratic Assignment (pb*)
- Quadratic Linear Ordering: bipart. graphs edge crossing (edgecross*), Facility Layout (faclay*)
- Multicommodity network design (ndcc*)
- Portfolio Optimization S&P100 (smallinvSNP*)
- Stochastic Service System Design (sssd*)
- Trimloss (tln*, tls*)

Best Primal Integral ↔ Optimization Problem

ANTIGONE

- Multiperiod Blend Scheduling (blend*)
- Extended Pooling Problem (pooling_epa*)
- Generalized Pooling (genpooling*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (m6, m7*)
- Facility Location (sfacloc*)
- Supply Chain Design (supplychain*)
- Water Network Design (waterund*, watertreat*)

LINDO API

- Facility Location (emfl*)
- Portfolio Optimization DAX (smallinvDAX*)
- Synthesis of Space Truss (space25*)
- Uncapacitated Facility Location (squfl*)
- Telecom. Network Design (telecomsp*)
- Water Network Contamination (watercontamination*)

BARON

- Facility Layout (m6, m7*, o7,8,9*)
- Farm Layout (flay*)
- Block Layout (fo*, no*)
- Graph partitioning (graphpart*)
- Circle/Polygon/Rectangle Packing (kall_*)
- Point Packing (pointpack*)
- Product Portfolio Optimization (kport*)
- Retrofit Planning (rsyn*)
- Safety Layout (slay*)

SCIP

- Block Layout (no*)
- Cutting Stock (carton*)
- Cross-dock Door Assignment (crossdock*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (o7,8,9*)
- General Quadratic Assignment (pb*)
- Quadratic Linear Ordering: bipart. graphs edge crossing (edgecross*), Facility Layout (faclay*)
- Multicommodity network design (ndcc*)
- Portfolio Optimization S&P100 (smallinvSNP*)
- Stochastic Service System Design (sssd*)
- Trimloss (tln*, tls*)

Best Primal Integral ↔ Optimization Problem

ANTIGONE

- Multiperiod Blend Scheduling (blend*)
- Extended Pooling Problem (pooling_epa*)
- Generalized Pooling (genpooling*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (m6, m7*)
- Facility Location (sfacloc*)
- Supply Chain Design (supplychain*)
- Water Network Design (waterund*, watertreat*)

LINDO API

- Facility Location (emfl*)
- Portfolio Optimization DAX (smallinvDAX*)
- Synthesis of Space Truss (space25*)
- Uncapacitated Facility Location (squfl*)
- Telecom. Network Design (telecomsp*)
- Water Network Contamination (watercontamination*)

BARON

- Facility Layout (m6, m7*, o7,8,9*)
- Farm Layout (flay*)
- Block Layout (fo*, no*)
- Graph partitioning (graphpart*)
- Circle/Polygon/Rectangle Packing (kall_*)
- Point Packing (pointpack*)
- Product Portfolio Optimization (kport*)
- Retrofit Planning (rsyn*)
- Safety Layout (slay*)

SCIP

- Block Layout (no*)
- Cutting Stock (carton*)
- Cross-dock Door Assignment (crossdock*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (o7,8,9*)
- General Quadratic Assignment (pb*)
- Quadratic Linear Ordering: bipart. graphs edge crossing (edgecross*), Facility Layout (faclay*)
- Multicommodity network design (ndcc*)
- Portfolio Optimization S&P100 (smallinvSNP*)
- Stochastic Service System Design (sssd*)
- Trimloss (tln*, tls*)

Best Primal Integral ↔ Optimization Problem

ANTIGONE

- Multiperiod Blend Scheduling (blend*)
- Extended Pooling Problem (pooling_epa*)
- Generalized Pooling (genpooling*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (m6, m7*)
- Facility Location (sfacloc*)
- Supply Chain Design (supplychain*)
- Water Network Design (waterund*, watertreat*)

LINDO API

- Facility Location (emfl*)
- Portfolio Optimization DAX (smallinvDAX*)
- Synthesis of Space Truss (space25*)
- Uncapacitated Facility Location (squfl*)
- Telecom. Network Design (telecomsp*)
- Water Network Contamination (watercontamination*)

BARON

- Facility Layout (m6, m7*, o7,8,9*)
- Farm Layout (flay*)
- Block Layout (fo*, no*)
- Graph partitioning (graphpart*)
- Circle/Polygon/Rectangle Packing (kall_*)
- Point Packing (pointpack*)
- Product Portfolio Optimization (kport*)
- Retrofit Planning (rsyn*)
- Safety Layout (slay*)

SCIP

- Block Layout (no*)
- Cutting Stock (carton*)
- Cross-dock Door Assignment (crossdock*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (o7,8,9*)
- General Quadratic Assignment (pb*)
- Quadratic Linear Ordering: bipart. graphs edge crossing (edgecross*), Facility Layout (faclay*)
- Multicommodity network design (ndcc*)
- Portfolio Optimization S&P100 (smallinvSNP*)
- Stochastic Service System Design (sssd*)
- Trimloss (tln*, tls*)

Best Primal Integral ↔ Optimization Problem

ANTIGONE

- Multiperiod Blend Scheduling (blend*)
- Extended Pooling Problem (pooling_epa*)
- Generalized Pooling (genpooling*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (m6, m7*)
- Facility Location (sfacloc*)
- Supply Chain Design (supplychain*)
- Water Network Design (waterund*, watertreat*)

LINDO API

- Facility Location (emfl*)
- Portfolio Optimization DAX (smallinvDAX*)
- Synthesis of Space Truss (space25*)
- Uncapacitated Facility Location (squfl*)
- Telecom. Network Design (telecomsp*)
- Water Network Contamination (watercontamination*)

BARON

- Facility Layout (m6, m7*, o7,8,9*)
- Farm Layout (flay*)
- Block Layout (fo*, no*)
- Graph partitioning (graphpart*)
- Circle/Polygon/Rectangle Packing (kall_*)
- Point Packing (pointpack*)
- Product Portfolio Optimization (kport*)
- Retrofit Planning (rsyn*)
- Safety Layout (slay*)

SCIP

- Block Layout (no*)
- Cutting Stock (carton*)
- Cross-dock Door Assignment (crossdock*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (o7,8,9*)
- General Quadratic Assignment (pb*)
- Quadratic Linear Ordering: bipart. graphs edge crossing (edgecross*), Facility Layout (faclay*)
- Multicommodity network design (ndcc*)
- Portfolio Optimization S&P100 (smallinvSNP*)
- Stochastic Service System Design (sssd*)
- Trimloss (tln*, tls*)

Best Primal Integral ↔ Optimization Problem

ANTIGONE

- Multiperiod Blend Scheduling (blend*)
- Extended Pooling Problem (pooling_epa*)
- Generalized Pooling (genpooling*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (m6, m7*)
- Facility Location (sfacloc*)
- Supply Chain Design (supplychain*)
- Water Network Design (waterund*, watertreat*)

LINDO API

- Facility Location (emfl*)
- Portfolio Optimization DAX (smallinvDAX*)
- Synthesis of Space Truss (space25*)
- Uncapacitated Facility Location (squfl*)
- Telecom. Network Design (telecomsp*)
- Water Network Contamination (watercontamination*)

almost completely discrete problems

BARON

- Facility Layout (m6, m7*, o7,8,9*)
- Farm Layout (flay*)
- Block Layout (fo*, no*)
- Graph partitioning (graphpart*)
- Circle/Polygon/Rectangle Packing (kall_*)
- Point Packing (pointpack*)
- Product Portfolio Optimization (kport*)
- Retrofit Planning (rsyn*)
- Safety Layout (slay*)

SCIP

- Block Layout (no*)
- Cutting Stock (carton*)
- Cross-dock Door Assignment (crossdock*)
- Crude Oil Scheduling (Pooling) (crudeoil*)
- Facility Layout (o7,8,9*)
- General Quadratic Assignment (pb*)
- Quadratic Linear Ordering: bipart. graphs edge crossing (edgecross*), Facility Layout (faclay*)
- Multicommodity network design (ndcc*)
- Portfolio Optimization S&P100 (smallinvSNP*)
- Stochastic Service System Design (sssd*)
- Trimloss (tln*, tls*)

End: Lunch time!

`http://www.gamsworld.org/minlp/minlplib2/html/`

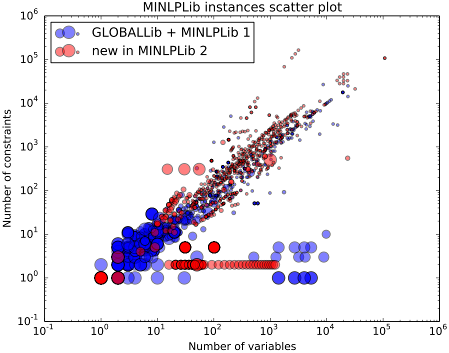
Future Work:

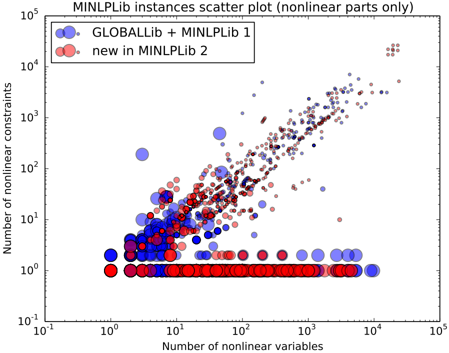
- add more NLPs (from PrincetonLib, COCONUT,...)
- semi-automatic identification of duplicates
- more structure recognition, e.g., second-order cones
- define interesting subsets, especially a benchmark set for global solvers

Call for contributions:

- Contribute your own (MI)NLP instances! (Or send your model to `minlp.org`!)
- Ideally from a model for a real life problem.
- Also infeasible instances are welcomed.
- Any (well-known) format is good (e.g., AMPL, GAMS, ZIMPL, BARON, CPLEX LP, MPS, PIP, OSiL).
- MINLPLib instances are anonymized (scalar format using generic names).
- Your benefit: Solver developers may test and tune their solver for your problem.

Instance Sparsity





Left: Area of bubble proportional to

$$\max\left(0.05, \frac{\#\text{nonzeros in objective grad. and jacobian}}{\#\text{vars}\cdot(\#\text{cons}+1)}\right).$$

Right: Area of bubble proportional to

$$\max\left(0.05, \frac{\#\text{nl. nonzeros in objective grad. and jacobian}}{\#\text{nl. vars}\cdot(\#\text{nl. cons}+1\text{ if obj. nl.})}\right).$$

Coefficient range

Let $P = \{\text{initial point}\} \cup \{\text{some known feasible points}\}$.

Minimal nonzero absolute coefficient:

$$m(f) := \min\{|(\nabla h(x))_i| \;:\; i \in \{1, \ldots, n\} \text{ with } |(\nabla h(\hat{x}))_i| > 10^{-12}, \hat{x} \in P\}$$

Maximal absolute coefficient:

$$M(f) := \max\{|\nabla h_i(x)| \;:\; i \in \{1, \ldots, n\}, \hat{x} \in P\}$$

Coefficient range

Let $P = \{\text{initial point}\} \cup \{\text{some known feasible points}\}$.
Minimal nonzero absolute coefficient:

$$m(f) := \min\{|(\nabla h(x))_i| \ : \ i \in \{1, \ldots, n\} \text{ with } |(\nabla h(\hat{x}))_i| > 10^{-12}, \hat{x} \in P\}$$

Maximal absolute coefficient:

$$M(f) := \max\{|\nabla h_i(x)| \ : \ i \in \{1, \ldots, n\}, \hat{x} \in P\}$$

For h ranging over the objective function and all constraint functions, consider

$$\frac{\max_h M(h)}{\min_h m(h)}$$

as coefficient range.

Coefficient range

Let $P = \{\text{initial point}\} \cup \{\text{some known feasible points}\}$.

Minimal nonzero absolute coefficient:

$$m(f) := \min\{|(\nabla h(x))_i| \ : \ i \in \{1, \dots, n\} \text{ with } |(\nabla h(\hat{x}))_i| > 10^{-12}, \hat{x} \in P\}$$

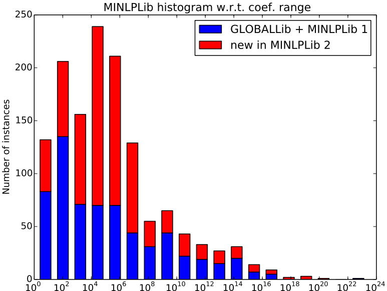
Maximal absolute coefficient:

$$M(f) := \max\{|\nabla h_i(x)| \ : \ i \in \{1, \dots, n\}, \hat{x} \in P\}$$

For h ranging over the objective function and all constraint functions, consider

$$\frac{\max_h M(h)}{\min_h m(h)}$$

as coefficient range.


MINLPLib histogram w.r.t. coef. range
Number of instances
🟦 GLOBALLib + MINLPLib 1
🟥 new in MINLPLib 2
10^0
10^2
10^4
10^6
10^8
10^{10}
10^{12}
10^{14}
10^{16}
10^{18}
10^{20}
10^{22}
10^{24}