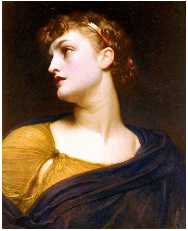
MINLP – Global Solvers

Stefan Vigerske
stefan@gams.com

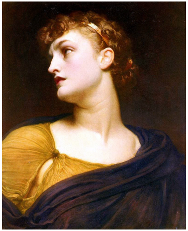
October 8th, 2015, CO@Work, Berlin

ANTIGONE (**A**lgorithms for co**NT**inuous / **I**nteger **G**lobal **O**ptimization of **N**onlinear **E**quations)



- by R. Misener (Imperial) and C.A. Floudas (Princeton)
- originating from a solver for pooling problems
- available as commercial solver in GAMS
- Misener and Floudas [2012a,b, 2014], Misener [2012]

ANTIGONE (**A**lgorithms for co**NT**inuous / **I**nteger **G**lobal **O**ptimization of **N**onlinear **E**quations)



- by R. Misener (Imperial) and C.A. Floudas (Princeton)
- originating from a solver for pooling problems
- available as commercial solver in GAMS
- Misener and Floudas [2012a,b, 2014], Misener [2012]

BARON (**B**ranch **A**nd **R**educe **O**ptimization **N**avigator)

- by N. Sahinidis (CMU) and M. Tawarmalani (Purdue)
- one of the first general purpose codes
- available as commercial solver in AIMMS, AMPL, GAMS
- Tawarmalani and Sahinidis [2002, 2004, 2005]

Couenne (**C**onvex **O**ver and **U**nder **EN**velopes for **N**onlinear **E**stimation)

- by P. Belotti (CMU, Clemson, now FICO)
- COIN-OR open source solver based on Bonmin (based on CBC and Ipopt)
- supports also trigonometric functions (sin, cos)
- available for AMPL and in GAMS and OS
- Belotti, Lee, Liberti, Margot, and Wächter [2009]

Couenne (**C**onvex **O**ver and **U**nder **EN**velopes for **N**onlinear **E**stimation)

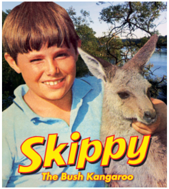
- by P. Belotti (CMU, Clemson, now FICO)
- COIN-OR open source solver based on Bonmin (based on CBC and Ipopt)
- supports also trigonometric functions (sin, cos)
- available for AMPL and in GAMS and OS
- Belotti, Lee, Liberti, Margot, and Wächter [2009]

LindoAPI

- by Y. Lin and L. Schrage (LINDO Systems, Inc.)
- supports many functions, incl. trigonometric (sin, cos)
- available as commercial solver within LINDO and GAMS
- Lin and Schrage [2009]

SCIP (**S**olving **C**onstraint **I**nteger **P**rograms)

- by Zuse Institute Berlin, TU Darmstadt, ...
- part of a constraint integer programming framework
- free for academic use, available for AMPL and in GAMS
- Berthold, Gleixner, Heinz, and Vigerske [2012], Vigerske [2013]



Upcoming Deterministic Global Solvers for MINLP

COCONUT (**CO**ntinuous **CON**straints – **U**pdating the **T**echnology)

- by A. Neumaier, H. Schichl, E. Monfroy (Vienna), et.al.
- rigorous calculations via interval arithmetics, thus avoiding floating point roundoff errors
- still in development, no stable release so far
- Neumaier [2004], Bliek et al. [2001]

Upcoming Deterministic Global Solvers for MINLP

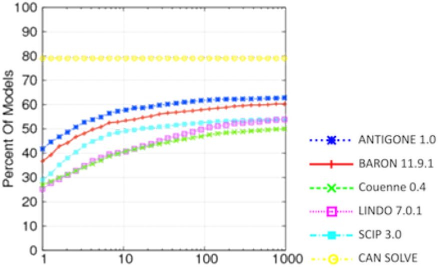
COCONUT (**CO**ntinuous **CON**straints – **U**pdating the **T**echnology)

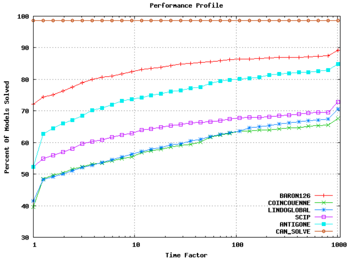
- by A. Neumaier, H. Schichl, E. Monfroy (Vienna), et.al.
- rigorous calculations via interval arithmetics, thus avoiding floating point roundoff errors
- still in development, no stable release so far
- Neumaier [2004], Bliek et al. [2001]

MINOTAUR (**M**ixed-**I**nteger **N**onconvex **O**ptimization **T**oolbox – **A**lgorithms, **U**nderestimators, **R**elaxations)

- by A. Mahajan, S. Leyffer, J. Linderoth, J. Luedtke, T. Munson, et.al. (Argonne, Wisconsin-Madison, IIT Bombay)
- open source with AMPL interface
- branch-and-bound with NLP relaxation (or its QP approximation); facilities to handle and manipulate algebraic expression are in place
- Mahajan and Munson [2010], Mahajan et al. [2012]

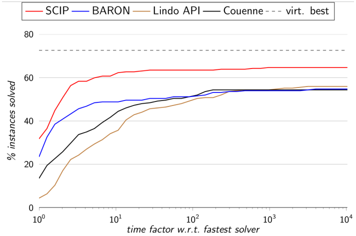
“Best Solver”?





http://helios.princeton.edu/ANTIGONE/
1705 instances, NLPs and MINLPs

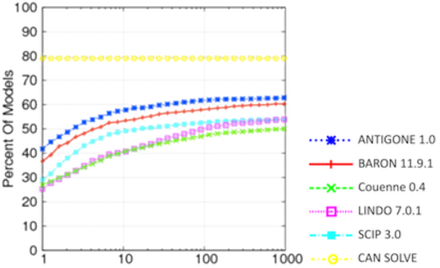
http://archimedes.cheme.cmu.edu/?q=baron
1599 instances, NLPs and MINLPs



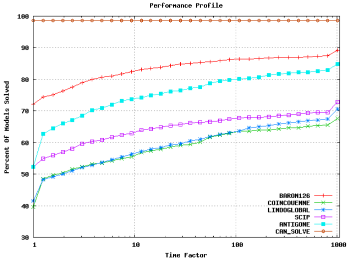
http://www.math.hu-berlin.de/~stefan/SCIP_
ISMP12.pdf; 252 instances, MINLPs

“Best Solver”?

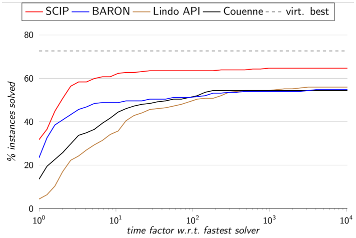
GAMS



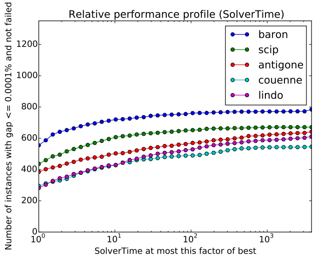
http://helios.princeton.edu/ANTIGONE/
1705 instances, NLPs and MINLPs



http://archimedes.cheme.cmu.edu/?q=baron
1599 instances, NLPs and MINLPs



http://www.math.hu-berlin.de/~stefan/SCIP_ISMP12.pdf; 252 instances, MINLPs



www.gams.com/presentations/2015_ismp.pdf
≈ 1360 instances, NLPs and MINLPs

Where do the differences come from?

Different benchmark sets:

- So far, there exists no test set that the major solver vendors agree on.
- Test sets may be unbalanced and contain randomly created instances, trivial or extremely hard instances.

Recall also **Thorsten's talk** "Notes on MIPLIB and Benchmarking"

Where do the differences come from?

Different benchmark sets:

- So far, there exists no test set that the major solver vendors agree on.
- Test sets may be unbalanced and contain randomly created instances, trivial or extremely hard instances.

Different performance metrics:

- When to count an instance as **solved**?
  - If found a solution and proven optimality: gap $= 0$?
  - If found a solution and proven that it is close to optimality: gap $\leq 10^{-6}$?
  - If found the best solution among all solvers in the comparison ?
- Mean solve time, number of solved instances, (extended) (inverted) (filtered) performance profiles, primal/dual/primal-dual integrals, ...
- Handling of crashes, timeouts, memouts, ...

Recall also **Thorsten's talk** "Notes on MIPLIB and Benchmarking"

Where do the differences come from?

GAMS

Different benchmark sets:

- So far, there exists no test set that the major solver vendors agree on.
- Test sets may be unbalanced and contain randomly created instances, trivial or extremely hard instances.

Different performance metrics:

- When to count an instance as **solved**?
  - If found a solution and proven optimality: gap $= 0$?
  - If found a solution and proven that it is close to optimality: gap $\leq 10^{-6}$?
  - If found the best solution among all solvers in the comparison ?
- Mean solve time, number of solved instances, (extended) (inverted) (filtered) performance profiles, primal/dual/primal-dual integrals, ...
- Handling of crashes, timeouts, memouts, ...

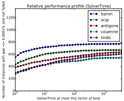
Numerics:

- Relative vs. absolute **feasibility tolerances**.
- Solvers are tuned to they default tolerance.
- A strictly-feasible (tolerance $= 0$) solution may exist neither in floating-point nor in rational arithmetics ($\sqrt{2}, \pi, \ldots$)
- In general no simple proofs for **infeasibilty or unboundedness**.

Recall also **Thorsten's talk** "Notes on MIPLIB and Benchmarking"

Model instance collections

GAMS

For the performance profile 

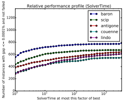
, MINLPLib 2 was used.

However, collecting optimization problems has been popular for a 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
2015	QPLIB (β)	(Mixed-Integer) Quadratic Programming

Model instance collections

GAMS

For the performance profile 

, MINLPLib 2 was used.

However, collecting optimization problems has been popular for a long time, e.g.,

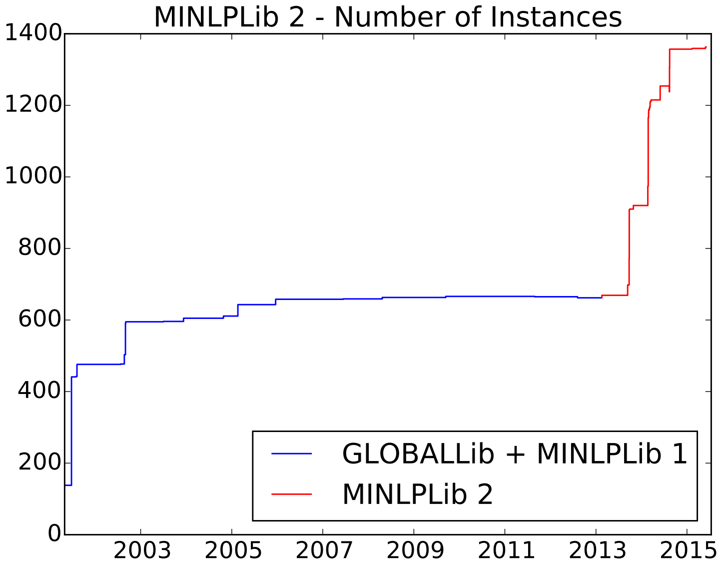
first release	library	problem types
1985	Netlib	Linear Programming
1992	MIPLIB	Mixed-Integer Programming
1993	CUTE	**Nonlinear Programming**
1998	SDPLib	Semidefinite Programming
1999	CSPLib	Constraint Satisfaction Programming
199x	MacMINLP	**Mixed-Integer Nonlinear Programming**
2001	GAMS World	LP, MIP, **NLP**, **MINLP**, SOCP, MPEC
2003	COCONUT	**Nonlinear and Constraint Satisfaction Programm**
2008	mintOC	Mixed-Integer Optimal Control
2009	minlp.org	**MINLP**, General Disjunctive Programming
2011	POLIP	**Mixed-Integer Polynomial Programming**
2014	CBLIB	**Conic Programming**
2015	QPLIB (β)	**(Mixed-Integer) Quadratic Programming**

- **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)
- 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.)
- until 2009 maintained by Michael Bussieck, then I got "volunteered"
- frequently used for testing, but also benchmarking

- **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)
- 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.)
- until 2009 maintained by Michael Bussieck, then I got "volunteered"
- frequently used for testing, but also benchmarking

Since 2013:

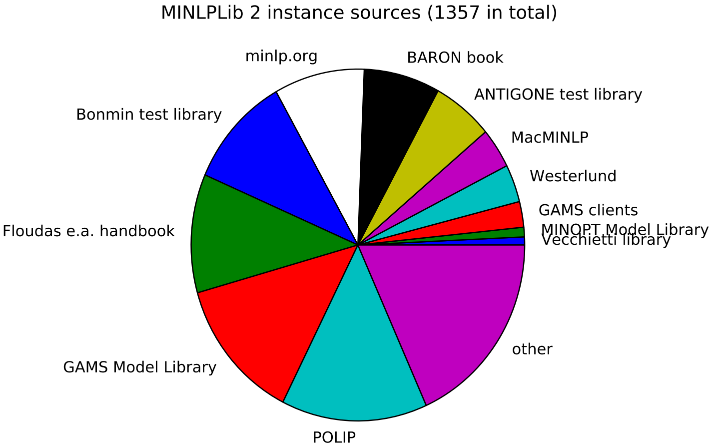
- **Extension and Renovation** (work in progress)
- Merged GlobalLib and MINPLLib 1 into MINLPLib 2.
- Current version:
  `http://www.gamsworld.org/minlp/minlplib2/html/index.html`.


MINLPLib 2 - Number of Instances
1400
1200
1000
800
600
400
200
0
2003
2005
2007
2009
2011
2013
2015
GLOBALLib + MINLPLib 1
MINLPLib 2

Sources of newly added instances

Harvesting mainly from

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



MINLPLib 2

Several formats:

- AIMMS, AMPL `.mod`/`.nl`, GAMS, OSiL (XML), CPLEX LP, PIP

MINLPLib 2

GAMS

Several formats:

- AIMMS, AMPL `.mod`/`.nl`, GAMS, OSiL (XML), CPLEX LP, PIP

Classification of instances:

- binary, general integer, SOS, ...
- quadratic, polynomial, signomial, other-nonlinear
- convex, concave, indefinite, unknown

MINLPLib 2

Several formats:

- AIMMS, AMPL `.mod`/`.nl`, GAMS, OSiL (XML), CPLEX LP, PIP

Classification of instances:

- binary, general integer, SOS, ...
- quadratic, polynomial, signomial, other-nonlinear
- convex, concave, indefinite, unknown

Solution Points:

- record maximal absolute constraint violation

MINLPLib 2

Several formats:

- AIMMS, AMPL `.mod`/`.nl`, GAMS, OSiL (XML), CPLEX LP, PIP

Classification of instances:

- binary, general integer, SOS, ...
- quadratic, polynomial, signomial, other-nonlinear
- convex, concave, indefinite, unknown

Solution Points:

- record maximal absolute constraint violation

Dual Bounds:

- record dual bounds from various solvers
- trust only bounds that were reported by several (e.g., 3) solvers

MINLPLib 2

Several formats:

- AIMMS, AMPL `.mod`/`.nl`, GAMS, OSiL (XML), CPLEX LP, PIP

Classification of instances:

- binary, general integer, SOS, ...
- quadratic, polynomial, signomial, other-nonlinear
- convex, concave, indefinite, unknown

Solution Points:

- record maximal absolute constraint violation

Dual Bounds:

- record dual bounds from various solvers
- trust only bounds that were reported by several (e.g., 3) solvers

Query Script:

```
$ ./query.py "(nvars > 4242) & (convex == True)" -c nvars -c ncons
$ ./query.py "(opsin == True) or (opcos == True)"
$ ./query.py "gap > 0.1" -c gap -z open.zip
```

What to do with all these instances?

What to do with all these instances?

GAMS

Quantify Improvements of global MINLP solvers over the last 4 years!

date	GAMS	ANTIGONE	BARON	COUENNE	LINDO	SCIP
08/11	23.7.3	–	**9.3**.1	**0.3**	**6.1**.1.588	–
04/12	23.8.2	–	**10.2**.0	**0.4**	**7.0**.1.421	**2.1**.1
11/12	23.9.5	–	**11.5**.2	0.4	7.0.1.497	2.1.2
02/13	24.0.2	–	**11.9**.1	0.4	7.0.1.497	**3.0**
07/13	24.1.3	**1.1**	**12.3**.3	0.4	**8.0**.1283.385	3.0
05/14	24.2.3	1.1	**12.7**.7	0.4	8.0.1694.498	3.0
09/14	24.3.3	1.1	**14.0**.3	0.4	8.0.1694.550	**3.1**
06/15	24.4.6	1.1	**14.4**.0	0.4	**9.0**.1983.157	3.1
07/15	24.5α	1.1	**15.6**.5	**0.5**	9.0.1983.157	**3.2**.0

More Detailed Slides: `http://www.gams.com/presentations/2015_ismp.pdf`

Which instances to run?

MINLPLib 2 with its **1363 instances**?

Which instances to run?

GAMS

MINLPLib 2 with its **1363 instances**?

✓ large number of instances
$\Rightarrow$ **40** solver versions $\times$ **1363** instances = **54520** runs **(!)**

Which instances to run?

MINLPLib 2 with its **1363 instances**?

- ✓ large number of instances
  $\Rightarrow$ **40** solver versions $\times$ **1363** instances = **54520** runs **(!)**
- ✓ wide variety of applications

Which instances to run?

GAMS

MINLPLib 2 with its **1363 instances**?

- ✓ large number of instances
  $\Rightarrow$ **40** solver versions $\times$ **1363** instances = **54520** runs **(!)**
- ✓ wide variety of applications
- × dominance of certain models, e.g.,
  - 32 block layout design problems
  - 60 small investor portfolio optimization instances
  - ...

Which instances to run?

GAMS

MINLPLib 2 with its **1363 instances**?

- ✓ large number of instances
  $\Rightarrow$ **40** solver versions $\times$ **1363** instances = **54520** runs **(!)**
- ✓ 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

Which instances to run?

MINLPLib 2 with its **1363 instances**?

- ✓ large number of instances
  $\Rightarrow$ **40** solver versions $\times$ **1363** instances = **54520** runs **(!)**
- ✓ 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

Thus, need to **select a reasonable subset** of (e.g., 87) instances.

Which instances to run?

GAMS

MINLPLib 2 with its **1363 instances**?

- ✓ large number of instances
  ⇒ **40** solver versions × **1363** instances = **54520** runs **(!)**
- ✓ 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

Thus, need to **select a reasonable subset** of (e.g., 87) instances.

With 15 co-authors and plenty of time, this would be no problem.

MIPLIB 2010

Mixed Integer Programming Library version 5

Thorsten Koch · Tobias Achterberg · Erling Andersen · Oliver Bastert · Timo Berthold · Robert E. Bixby · Emilie Danna · Gerald Gamrath · Ambros M. Gleixner · Stefan Heinz · Andrea Lodi · Hans Mittelmann · Ted Ralphs · Domenico Salvagnin · Daniel E. Steffy · Kati Wolter

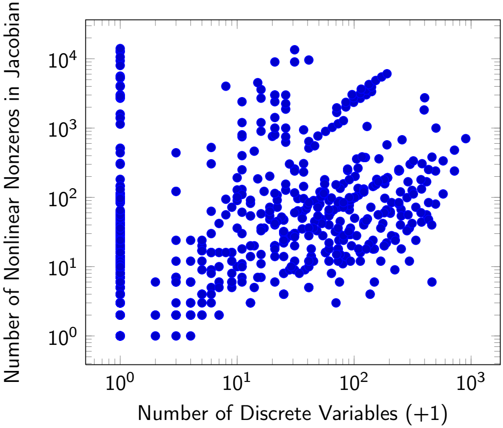
But when only 3 weeks left until ISMP: Apply the **P.I.T.T. heuristic**.

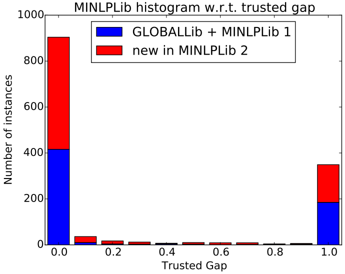
Prune Instances by Tractability and Triviality Heur.

GAMS

1. Remove intractable instances

- consider only the 881 instances that are **marked as solved** in MINLPLib 2



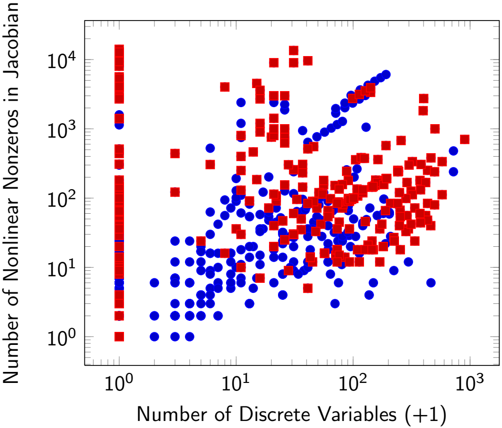


Prune Instances by Tractability and Triviality Heur.

GAMS

2. For each solver separately:

- Remove instances that are **solved within 60 seconds** by the oldest solver version (e.g., as in GAMS 23.7).
- Remove instances that the solver **cannot handle** (due to trigonometric functions, SOS, ...).



In case of SCIP:

GAMS

For SCIP, this leaves 312 instances:

alkylation	clay0304h	ex6_1_1	fo9_ar5_1	house
arki0003	clay0305h	ex6_1_3	gasnet	jbearing25
arki0005	crudeoil_lee2_10	ex6_2_12	genpooling_meyer04	jbearing75
arki0006	crudeoil_lee3_07	ex6_2_14	ghg_1veh	johnall
arki0019	crudeoil_lee3_08	ex6_2_8	ghg_3veh	kall_circles_c6a
arki0024	crudeoil_lee3_09	ex6_2_9	glider100	kall_circles_c6b
autocorr_bern20-10	crudeoil_lee3_10	ex7_2_4	graphpart_2g-0066-0066	kall_circles_c7a
autocorr_bern20-15	crudeoil_li06	ex8_1_7	graphpart_2g-0077-0077	kall_circles_c8a
autocorr_bern25-06	csched1a	ex8_2_1b	graphpart_2g-0088-0088	kall_circlespolygons_c1p
autocorr_bern25-13	edgecross10-060	ex8_2_4b	graphpart_2g-0099-9211	kall_circlespolygons_c1p
autocorr_bern30-04	edgecross10-070	ex8_4_1	graphpart_2pm-0066-0066	kall_circlesrectangles_c
autocorr_bern35-04	edgecross10-080	ex8_4_3	graphpart_2pm-0077-0777	kall_circlesrectangles_c
batch0812_nc	edgecross14-039	ex8_4_4	graphpart_2pm-0088-0888	kall_congruentcircles_c7
batchs201210m	edgecross14-058	ex8_4_5	graphpart_2pm-0099-0999	kall_diffcircles_10
bayes2_50	edgecross14-078	ex8_4_8_bnd	graphpart_3g-0334-0334	kall_diffcircles_5b
blend480	edgecross14-176	filter	graphpart_3g-0344-0344	kall_diffcircles_7
blend531	edgecross20-040	fin2bb	graphpart_3g-0444-0444	kall_diffcircles_8
blend718	edgecross22-048	flay05h	graphpart_3pm-0244-0244	kall_diffcircles_9
blend852	emfl050_5_5	fo7	graphpart_3pm-0333-0333	launch
carton7	emfl100_5_5	fo8	graphpart_3pm-0334-0334	lop97icx
casctanks	ethanolh	fo8_ar25_1	graphpart_3pm-0344-0344	mathopt5_7
cecil_13	ethanolm	fo8_ar2_1	graphpart_3pm-0444-0444	mathopt5_8
chem	ex1252a	fo9	graphpart_clique-30	mhw4d
clay0203h	ex14_1_1	fo9_ar25_1	graphpart_clique-40	milinfract
clay0204h	ex14_1_7	fo9_ar2_1	gsg_0001	minlphix
clay0205h	ex4_1_5	fo9_ar3_1	hda	minsurf100
clay0303h	ex4_1_6	fo9_ar4_1	heatexch_trigen	...

Prune Instances by Tractability and Triviality Heur.

For SCIP, this leaves 312 instances – **obvious dominance by some models**

alkylation
arki0003
arki0005
arki0006
arki0019
arki0024
autocorr_bern20-10
autocorr_bern20-15
autocorr_bern25-06
autocorr_bern25-13
autocorr_bern30-04
autocorr_bern35-04
batch0812_nc
batchs201210m
bayes2_50
blend480
blend531
blend718
blend852
carton7
casctanks
cecil_13
chem
clay0203h
clay0204h
clay0205h
clay0303h
clay0304h
clay0305h
crudeoil_lee2_10
crudeoil_lee3_07
crudeoil_lee3_08
crudeoil_lee3_09
crudeoil_lee3_10
crudeoil_li06
csched1a
edgecross10-060
edgecross10-070
edgecross10-080
edgecross14-039
edgecross14-058
edgecross14-078
edgecross14-176
edgecross20-040
edgecross22-048
emfl050_5_5
emfl100_5_5
ethanolh
ethanolm
ex1252a
ex14_1_1
ex14_1_7
ex4_1_5
ex4_1_6
ex6_1_1
ex6_1_3
ex6_2_12
ex6_2_14
ex6_2_8
ex6_2_9
ex7_2_4
ex8_1_7
ex8_2_1b
ex8_2_4b
ex8_4_1
ex8_4_3
ex8_4_4
ex8_4_5
ex8_4_8_bnd
filter
fin2bb
flay05h
fo7
fo8
fo8_ar25_1
fo8_ar2_1
fo9
fo9_ar25_1
fo9_ar2_1
fo9_ar3_1
fo9_ar4_1
fo9_ar5_1
gasnet
genpooling_meyer04
ghg_1veh
ghg_3veh
glider100
graphpart_2g-0066-0066
graphpart_2g-0077-0077
graphpart_2g-0088-0088
graphpart_2g-0099-9211
graphpart_2pm-0066-0066
graphpart_2pm-0077-0777
graphpart_2pm-0088-0888
graphpart_2pm-0099-0999
graphpart_3g-0334-0334
graphpart_3g-0344-0344
graphpart_3g-0444-0444
graphpart_3pm-0244-0244
graphpart_3pm-0333-0333
graphpart_3pm-0334-0334
graphpart_3pm-0344-0344
graphpart_3pm-0444-0444
graphpart_clique-30
graphpart_clique-40
gsg_0001
hda
heatexch_trigen
house
jbearing25
jbearing75
johnall
kall_circles_c6a
kall_circles_c6b
kall_circles_c7a
kall_circles_c8a
kall_circlespolygons_c1p
kall_circlespolygons_c1p
kall_circlesrectangles_c
kall_circlesrectangles_c
kall_congruentcircles_c7
kall_diffcircles_10
kall_diffcircles_5b
kall_diffcircles_7
kall_diffcircles_8
kall_diffcircles_9
launch
lop97icx
mathopt5_7
mathopt5_8
mhw4d
milinfract
minlphix
minsurf100
...

P.I.T.T.E.D. Heur.: P.I.T.T. with **E**ased **D**ominance

GAMS

3. Ensure **uniqueness of 6-char.-prefix** of instances names.

P.I.T.T.E.D. Heur.: P.I.T.T. with **E**ased **D**ominance

3. Ensure **uniqueness of 6-char.-prefix** of instances names.

alkylation
arki0003
~~arki0005~~
~~arki0006~~
~~arki0019~~
~~arki0024~~
autocorr_bern20-10
~~autocorr_bern20-15~~
~~autocorr_bern25-06~~
~~autocorr_bern25-13~~
~~autocorr_bern30-04~~
~~autocorr_bern35-04~~
batch0812_nc
batchs201210m
bayes2_50
blend480
blend531
blend718
blend852
carton7
casctanks
cecil_13
chem
clay0203h
~~clay0204h~~
~~clay0205h~~
clay0303h
~~clay0304h~~
~~clay0305h~~
crudeoil_lee2_10
~~crudeoil_lee3_07~~
~~crudeoil_lee3_08~~
~~crudeoil_lee3_09~~
~~crudeoil_lee3_10~~
~~crudeoil_li06~~
csched1a
edgecross10-060
~~edgecross10-070~~
~~edgecross10-080~~
~~edgecross14-039~~
~~edgecross14-058~~
~~edgecross14-078~~
~~edgecross14-176~~
~~edgecross20-040~~
~~edgecross22-048~~
emfl050_5_5
~~emfl100_5_5~~
ethanolh
~~ethanolm~~
ex1252a
ex14_1_1
~~ex14_1_7~~
ex4_1_5
~~ex4_1_6~~
ex6_1_1
~~ex6_1_3~~
ex6_2_12
~~ex6_2_14~~
~~ex6_2_8~~
~~ex6_2_9~~
ex7_2_4
ex8_1_7
ex8_2_1b
~~ex8_2_4b~~
ex8_4_1
~~ex8_4_3~~
~~ex8_4_4~~
~~ex8_4_5~~
~~ex8_4_8_bnd~~
filter
fin2bb
flay05h
fo7
fo8
fo8_ar25_1
~~fo8_ar2_1~~
fo9
fo9_ar25_1
~~fo9_ar2_1~~
~~fo9_ar3_1~~
~~fo9_ar4_1~~
~~fo9_ar5_1~~
gasnet
genpooling_meyer04
ghg_1veh
ghg_3veh
glider100
graphpart_2g-0066-0066
~~graphpart_2g-0077-0077~~
~~graphpart_2g-0088-0088~~
~~graphpart_2g-0099-9211~~
~~graphpart_2pm-0066-0066~~
~~graphpart_2pm-0077-0777~~
~~graphpart_2pm-0088-0888~~
~~graphpart_2pm-0099-0999~~
~~graphpart_3g-0334-0334~~
~~graphpart_3g-0344-0344~~
~~graphpart_3g-0444-0444~~
~~graphpart_3pm-0244-0244~~
~~graphpart_3pm-0333-0333~~
~~graphpart_3pm-0334-0334~~
~~graphpart_3pm-0344-0344~~
~~graphpart_3pm-0444-0444~~
~~graphpart_clique-30~~
~~graphpart_clique-40~~
gsg_0001
hda
heatexch_trigen
house
jbearing25
~~jbearing75~~
johnall
kall_circles_c6a
~~kall_circles_c6b~~
~~kall_circles_c7a~~
~~kall_circles_c8a~~
~~kall_circlespolygons_c1p~~
~~kall_circlespolygons_c1p~~
~~kall_circlesrectangles_c~~
~~kall_circlesrectangles_c~~
~~kall_congruentcircles_c7~~
kall_diffcircles_10
~~kall_diffcircles_5b~~
~~kall_diffcircles_7~~
~~kall_diffcircles_8~~
~~kall_diffcircles_9~~
launch
lop97icx
mathopt5_7
~~mathopt5_8~~
mhw4d
milinfract
minlphix
minsurf100
...

P.I.T.T.E.D. SCIP testset

In summary:

1. Keep only instances that are marked as **solved** in MINLPLib 2.
2. Keep only instances that take $\geq$ **60s with oldest version** of solver and that can be **handled** by solver.
3. Reduce instances with **similar names**.

For SCIP, this reduces from 1363 to 881 to 123 instances:

alkylation
arki0003
autocorr_bern20-10
batch0812_nc
batchs201210m
bayes2_50
blend480
blend531
blend718
blend852
carton7
casctanks
cecil_13
chem
clay0203h
clay0303h
crudeoil_lee2_10
csched1a
edgecross10-060
emfl050_5_5
emfl100_5_5
ethanolh
ex1252a
ex14_1_1
ex4_1_5
ex6_1_1
ex6_2_12
ex7_2_4
ex8_1_7
ex8_2_1b
ex8_4_1
filter
fin2bb
flay05h
fo7
fo8
fo8_ar25_1
fo9
fo9_ar25_1
gasnet
genpooling_meyer04
ghg_1veh
ghg_3veh
glider100
graphpart_2g-0066-0066
gsg_0001
hda
heatexch_trigen
house
jbearing25
johnall
kall_circles_c6a
kall_diffcircles_10
launch
lop97icx
mathopt5_7
mhw4d
milinfract
minlphix
minsurf100
multiplants_mtg1a
no7_ar3_1
nous1
nvs09
nvs22
o7
o7_2
o7_ar25_1
o7_ar3_1
o7_ar4_1
o7_ar5_1
o8_ar4_1
o9_ar4_1
oil
oil2
parallel
pinene50
pointpack08
pooling_epa1
prob07
process
procsyn
prolog
qp3
routingdelay_bigm
rsyn0805m02h
sepasequ_convent
sfacloc2_2_80
slay07h
slay09h
slay10h
smallinvDAXr1b020-022
sporttournament14
squfl010-025
sssd08-04
sssd12-05
sssd15-04
sssd16-07
sssd18-06
sssd20-04
sssd25-04
st_e35
stockcycle
supplychai
syn10m03h
syn15m02h
syn20m02h
syn30h
syn30m02h
syn40h
syn40m02h

Run jobs

GAMS

date	GAMS	ANTIGONE	BARON	COUENNE	LINDO	SCIP
08/11	23.7.3	–	**9.3**.1	**0.3**	**6.1**.1.588	–
04/12	23.8.2	–	**10.2**.0	**0.4**	**7.0**.1.421	**2.1**.1
11/12	23.9.5	–	**11.5**.2	0.4	7.0.1.497	2.1.2
02/13	24.0.2	–	**11.9**.1	0.4	7.0.1.497	**3.0**
07/13	24.1.3	**1.1**	**12.3**.3	0.4	**8.0**.1283.385	3.0
05/14	24.2.3	1.1	**12.7**.7	0.4	8.0.1694.498	3.0
09/14	24.3.3	1.1	**14.0**.3	0.4	8.0.1694.550	**3.1**
06/15	24.4.6	1.1	**14.4**.0	0.4	**9.0**.1983.157	3.1
07/15	24.5α	1.1	**15.6**.5	**0.5**	9.0.1983.157	**3.2**.0

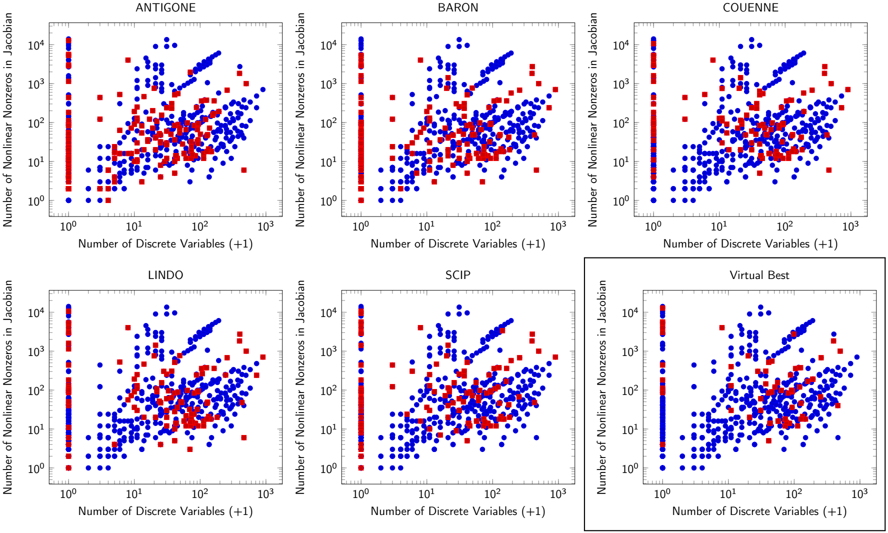
```
for GAMS in $GAMSS ; do
 for SOLVER in $SOLVERS($GAMS) ; do
  for INSTANCE in $TESTSET($SOLVER) ; do
   sbatch --exclusive --time=0:1800 $GAMS $INSTANCE SOLVER=$SOLVER
  done
 done
done
```

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

“Virtual Best” Solver

GAMS

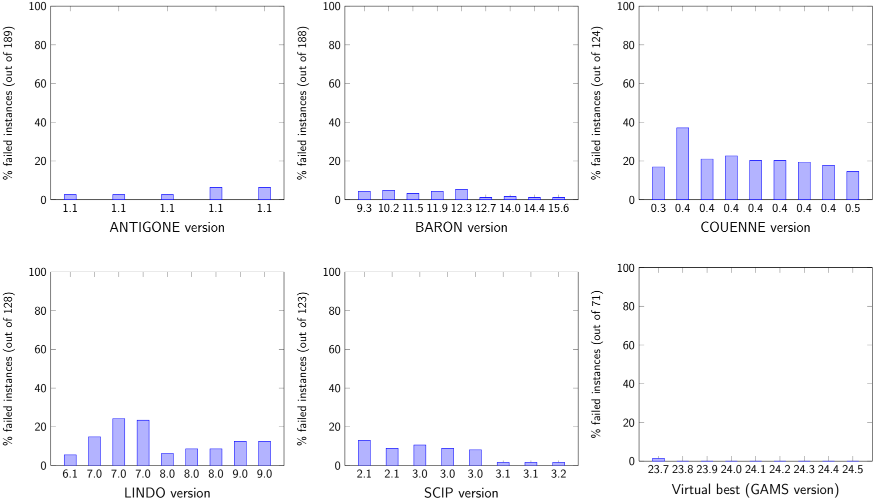
- common subset of instances (71 many)
- for each instance and GAMS version, pick **best results among all solvers**



Fails

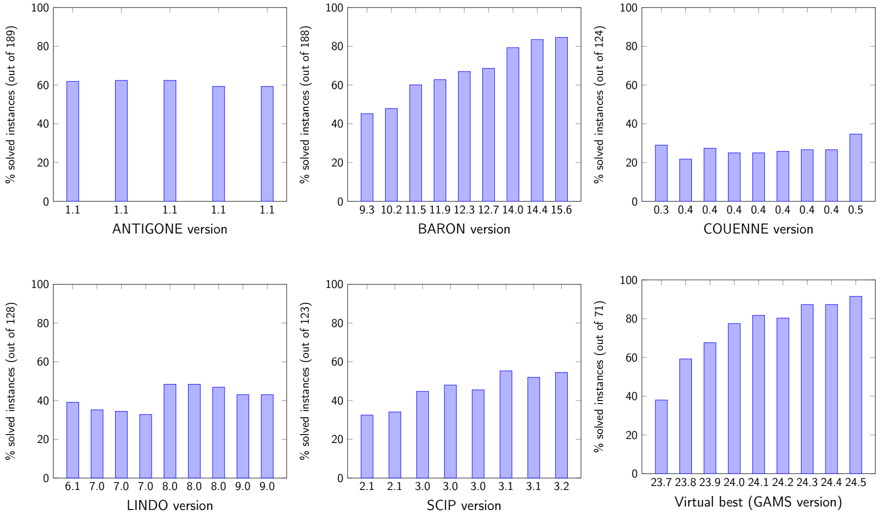
A solver **failed**, if it

- crashed, or
- reported an infeasible point as feasible (tolerance: 10^{-4}), or
- reported a suboptimal solution as optimal (tolerance: 10^{-4})



Solved

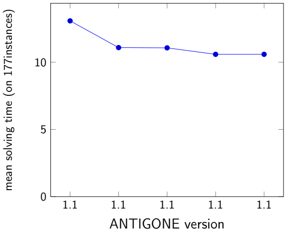
Solved: solver did **not fail** and reports a relative optimality **gap** $\leq 10^{-4}$



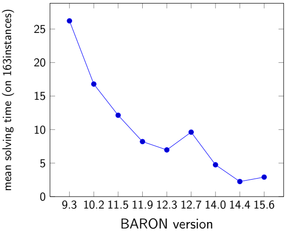
Mean solve time on instances that never failed

GAMS

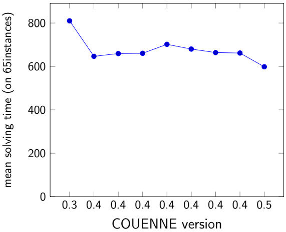
Overall Speedup: 1.18



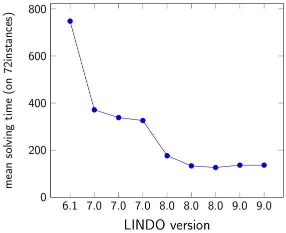
Overall Speedup: 9.01



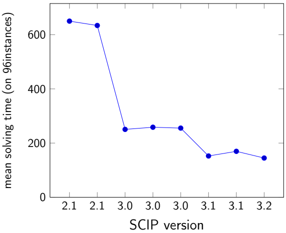
Overall speedup: 1.35



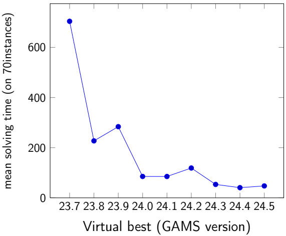
Overall speedup: 5.48



Overall speedup: 4.49



Overall speedup: 14.84



Literature I

GAMS

Pietro Belotti, Jon Lee, Leo Liberti, F. Margot, and Andreas Wächter. Branching and bounds tightening techniques for non-convex MINLP. Optimization Methods and Software, 24(4-5): 597–634, 2009. doi: 10.1080/10556780903087124.

Timo Berthold, Ambros M. Gleixner, Stefan Heinz, and Stefan Vigerske. Analyzing the computational impact of MIQCP solver components. Numerical Algebra, Control and Optimization, 2(4):739–748, 2012. doi: 10.3934/naco.2012.2.739.

Christian Bliek, Peter Spellucci, Luis N. Vicente, Arnold Neumaier, Laurent Granvilliers, Eric Monfroy, Frederic Benhamou, Etienne Huens, Pascal Van Hentenryck, Djamila Sam-Haroud, and Boi Faltings. Algorithms for solving nonlinear constrained and optimization problems: The state of the art. Technical report, Universität Wien, 2001. URL `http://www.mat.univie.ac.at/~neum/glopt/coconut/StArt.html`.

Youdong Lin and Linus Schrage. The global solver in the LINDO API. Optimization Methods & Software, 24(4–5):657–668, 2009. doi: 10.1080/10556780902753221.

Ashutosh Mahajan and Todd Munson. Exploiting second-order cone structure for global optimization. Technical Report ANL/MCS-P1801-1010, Argonne National Laboratory, 2010. URL `http://www.optimization-online.org/DB_HTML/2010/10/2780.html`.

Ashutosh Mahajan, Sven Leyffer, and Christian Kirches. Solving mixed-integer nonlinear programs by QP-diving. Preprint ANL/MCS-P2071-0312, Argonne National Laboratory, 2012. URL `http://www.optimization-online.org/DB_HTML/2012/03/3409.html`.

Literature II

GAMS

Ruth Misener. Novel Global Optimization Methods: Theoretical and Computational Studies on Pooling Problems with Environmental Constraints. PhD thesis, Princeton University, 2012. URL http://arks.princeton.edu/ark:/88435/dsp015q47rn787.

Ruth Misener and Christodoulos A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. Mathematical Programming, 136(1):155–182, 2012a. doi: 10.1007/s10107-012-0555-6.

Ruth Misener and Christodoulos A. Floudas. GloMIQO: Global mixed-integer quadratic optimizer. Journal of Global Optimization, 57(1):3–50, 2012b. doi: 10.1007/s10898-012-9874-7.

Ruth Misener and Christodoulos A. Floudas. ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. Journal of Global Optimization, 59(2-3): 503–526, 2014. doi: 10.1007/s10898-014-0166-2.

Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. In Acta Numerica, volume 13, chapter 4, pages 271–369. Cambridge University Press, 2004. doi: 10.1017/S0962492904000194.

Mohit Tawarmalani and Nikolaos V. Sahinidis. Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications, volume 65 of Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, 2002. doi: 10.1007/978-1-4757-3532-1.

Literature III

GAMS

Mohit Tawarmalani and Nikolaos V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Mathematical Programming, 99(3): 563–591, 2004. doi: 10.1007/s10107-003-0467-6.

Mohit Tawarmalani and Nikolaos V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. Mathematical Programming, 103(2):225–249, 2005. doi: 10.1007/s10107-005-0581-8.

Stefan Vigerske. Decomposition in Multistage Stochastic Programming and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming. PhD thesis, Humboldt-Universität zu Berlin, 2013. urn:nbn:de:kobv:11-100208240.