MINLP Solvers – version to version

This is an updated version of the MINLP solvers benchmark from my talk at ISMP 2015.

Stefan Vigerske

11th December 2017

Which instances to run?

Does the current MINLPLib 2 with its 1363 instances qualify as a good test set?

Which instances to run?

Does the current MINLPLib 2 with its 1363 instances qualify as a good test set?

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

Which instances to run?

Does the current MINLPLib 2 with its 1363 instances qualify as a good test set?

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

Which instances to run?

Does the current MINLPLib 2 with its 1363 instances qualify as a good test set?

- ✓ 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?

Does the current MINLPLib 2 with its 1363 instances qualify as a good test set?

- ✓ 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?

Does the current MINLPLib 2 with its 1363 instances qualify as a good test set?

- ✓ 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?

Does the current MINLPLib 2 with its 1363 instances qualify as a good test set?

- ✓ 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

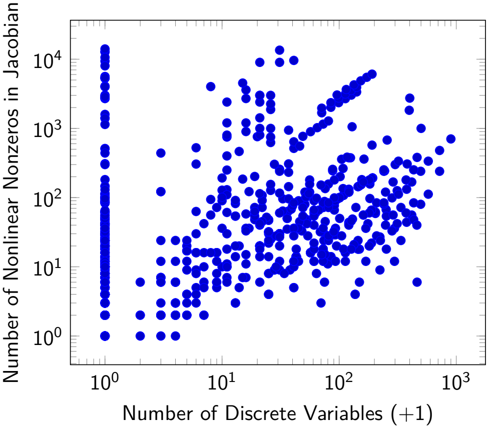
Thus, need to select a reasonable subset of (e.g., 87) instances.

Apply the P.I.T.T. heuristic.

Prune Instances by Tractability and Triviality Heuristic

1. Remove intractable instances

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

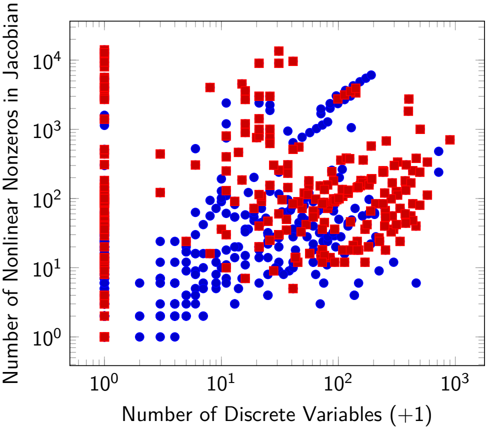


Prune Instances by Tractability and Triviality Heuristic

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:



Prune Instances by Tractability and Triviality Heuristic

For SCIP, this leaves 312 instances:

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
...

Prune Instances by Tractability and Triviality Heuristic

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. Heuristic: P.I.T.T. with Eased Dominance

3. Ensure uniqueness of 6-characters-prefix of instances names.

P.I.T.T.E.D. Heuristic: P.I.T.T. with Eased Dominance

3. Ensure uniqueness of 6-characters-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 ≥ 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

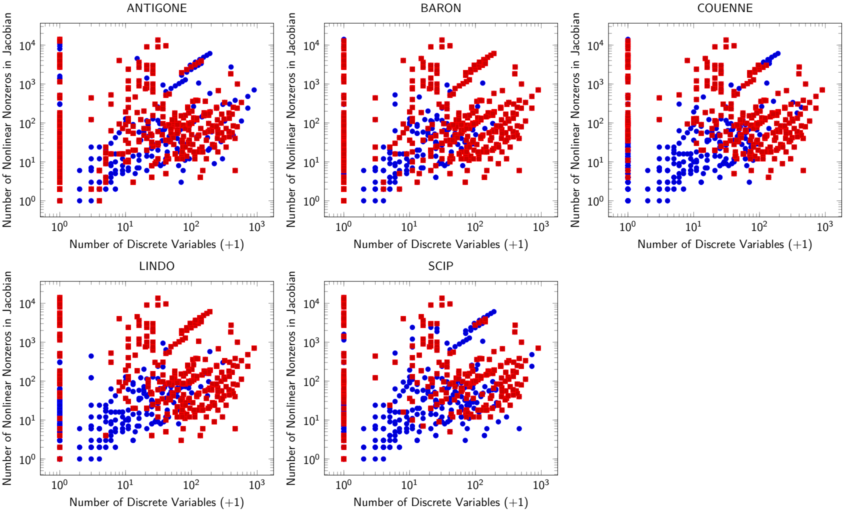
milinfract
minlphix
minsurf100
multiplants_mtg1a
no7_ar3_1
nous1
nvs09
nvs22
o7
o7_2
o7_ar25_1
o7_ar3_1
o7_ar4_1
o7_ar5_1
o8_ar4_1
o9_ar4_1
oil
oil2
parallel

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
stockcycl
supplychai
syn10m03h
syn15m02h
syn20m02h
syn30h
syn30m02h
syn40h
syn40m02h

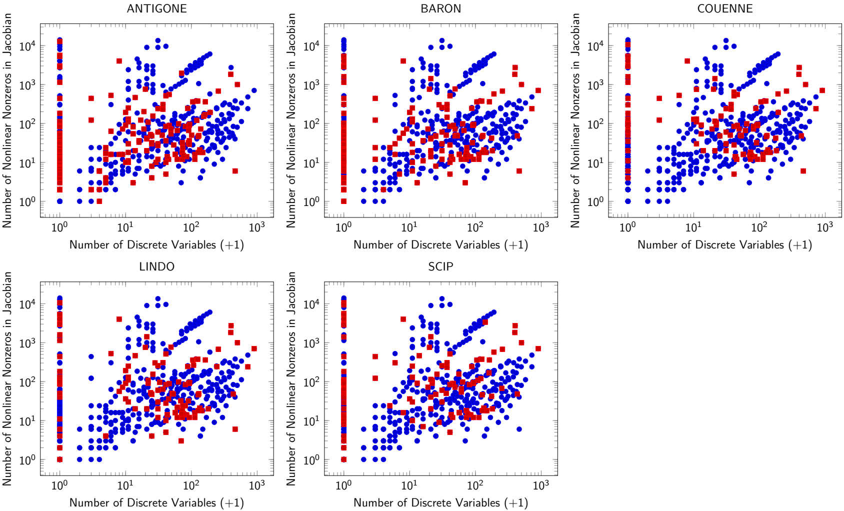
PITT test set for each solver

Removed easy and unsolvable instances:



PITTED test set for each solver

Removed easy and unsolvable instances, then filter by name:



Run jobs

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
11/15	24.5.6	1.1	**15.9**.22	**0.5**	9.0.2120.225	**3.2**
01/16	24.6.1	1.1	15.9.22	0.5	9.0.2120.225	3.2
09/16	24.7.4	1.1	**16.8**.24	0.5	9.0.2217.293	3.2
05/17	24.8.5	1.1	**17.4**.1	0.5	**10.0**.2539.182	3.2
11/17	24.9.2	1.1	**17.10**.16	0.5	**11.0**.3802.300	**4.0**

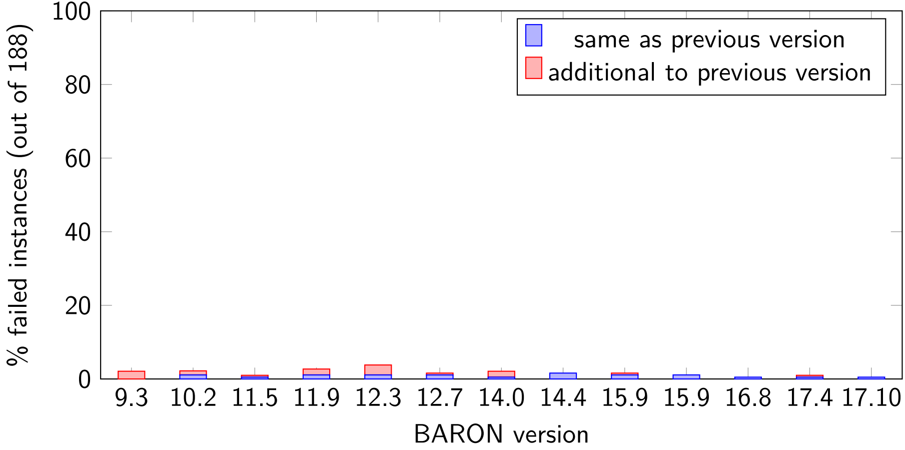
```
for GAMS in $GAMSS ; do
 for SOLVER in $SOLVERS($GAMS) ; do
  for INSTANCE in $TESTSET($SOLVER) ; do
   sbatch --exclusive --time=0:1800 $GAMS $INSTANCE SOLVER=$SOLVER
  done
 done
done
```

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

BARON: Fails

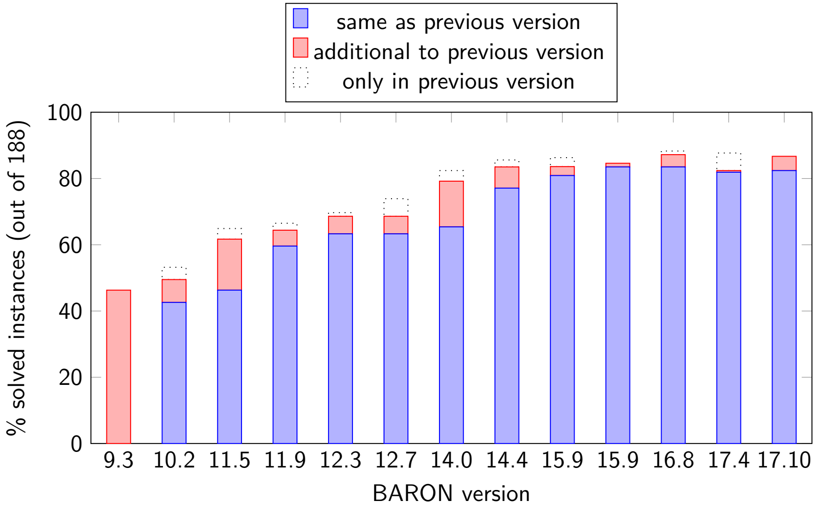
A solver failed, if it

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

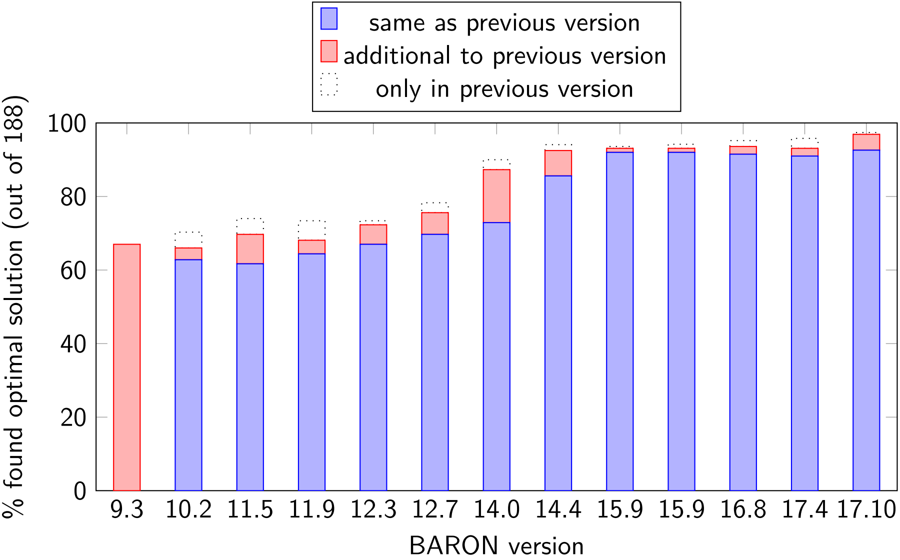


BARON: Solved

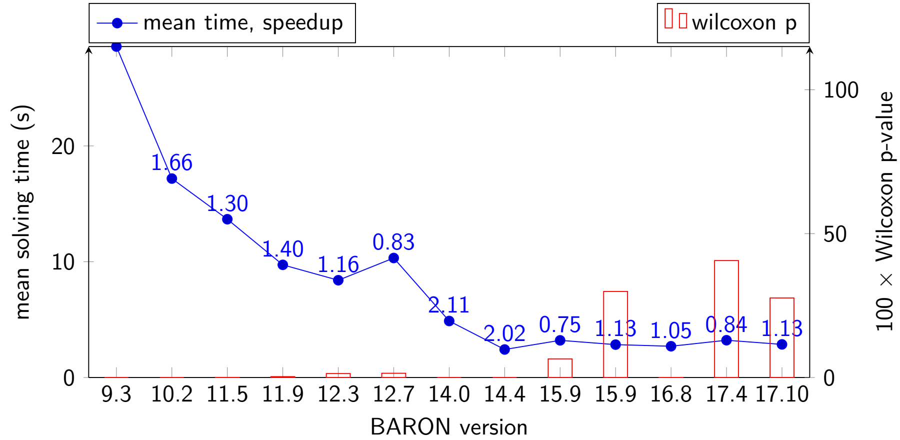
Solved: solver did not fail and reports a relative optimality gap $\leq 10^{-4}$



BARON: Found optimal solution



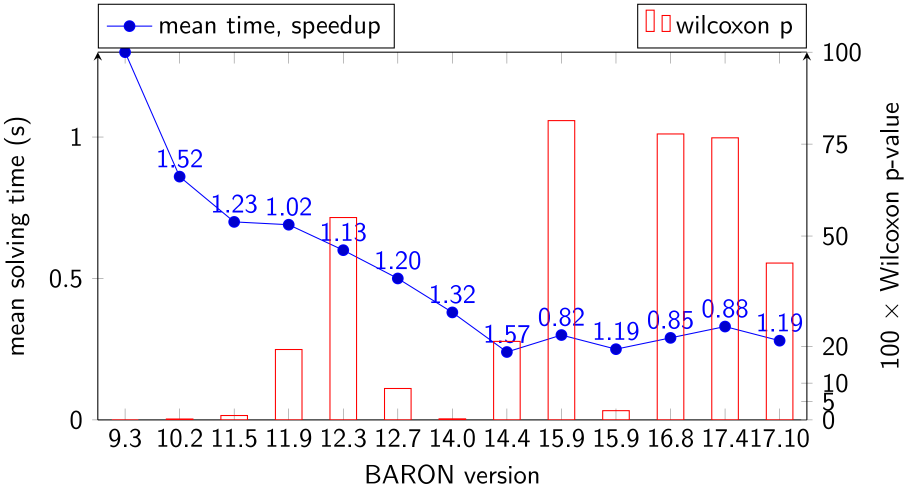
BARON: Solving time on instances that never failed (169)



Overall speedup: 10.1

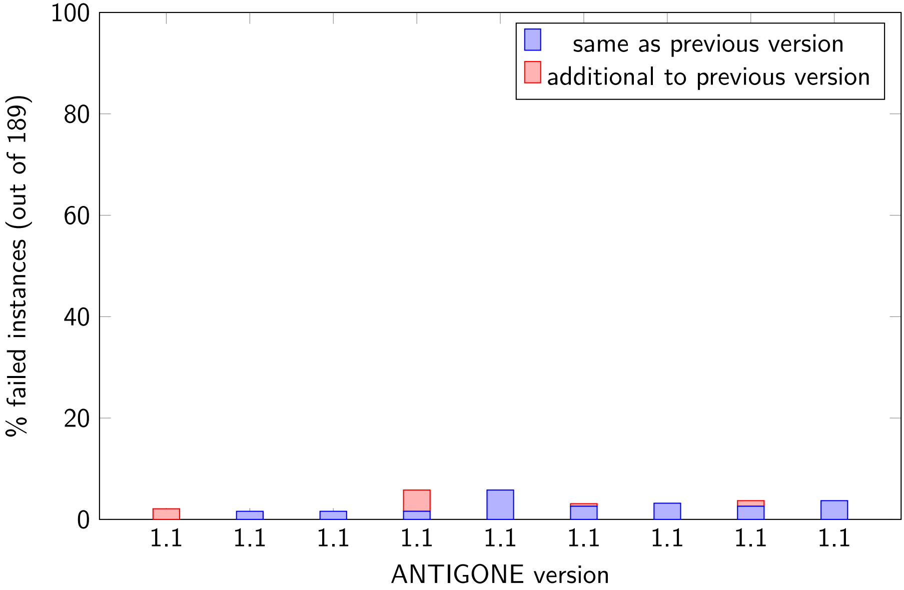
12.7: “Automatic setting of many options based on problem characteristics and learning algorithms.“

BARON: Solving time on instances solved by all vers. (71)

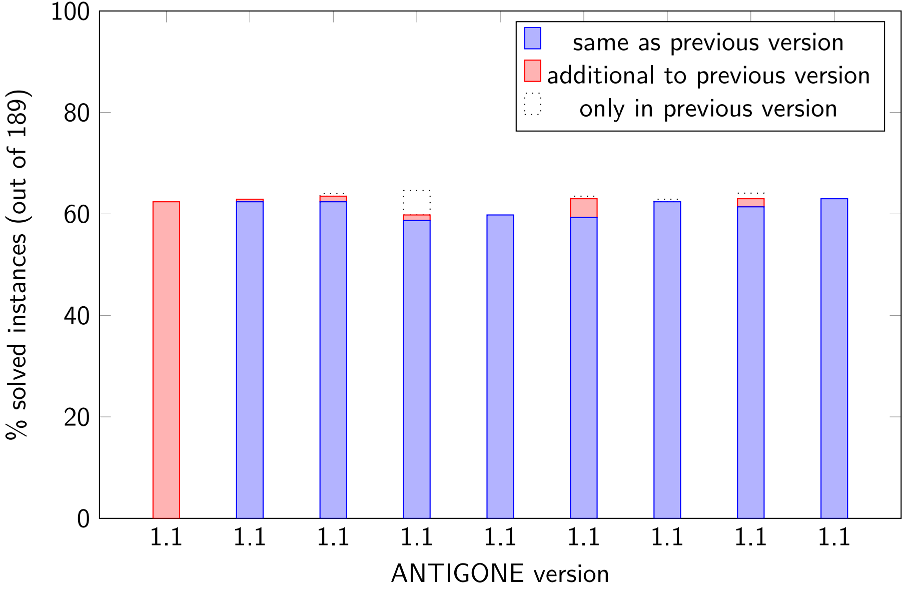


Overall speedup: 4.7

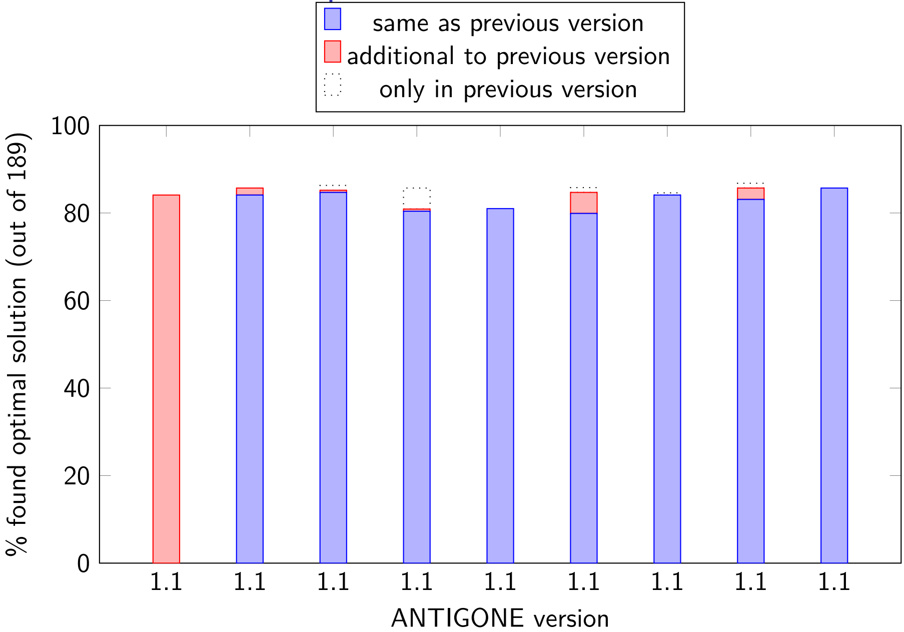
ANTIGONE: Fails



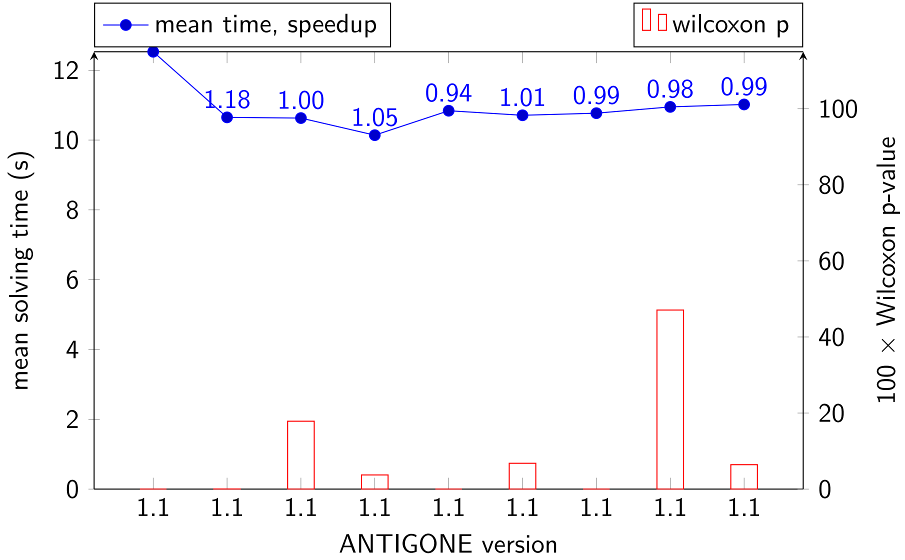
ANTIGONE: Solved



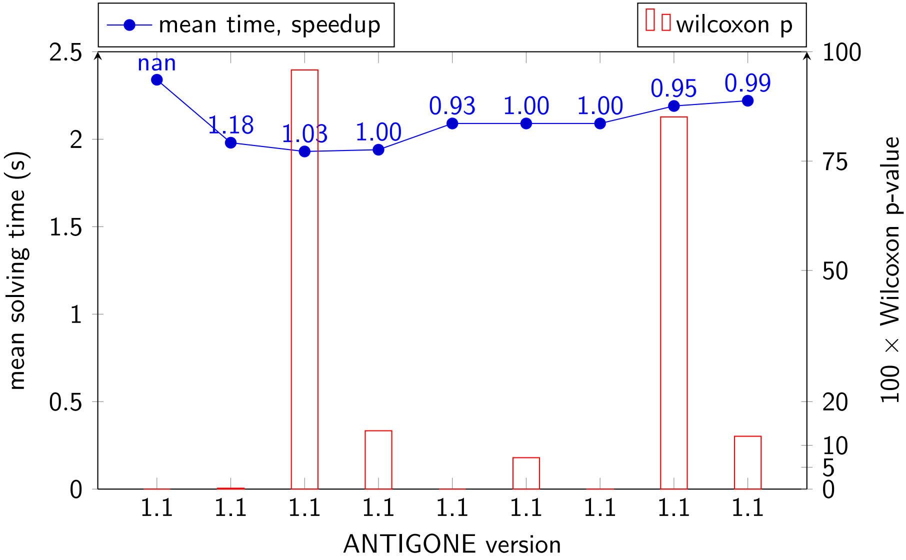
ANTIGONE: Found optimal solution



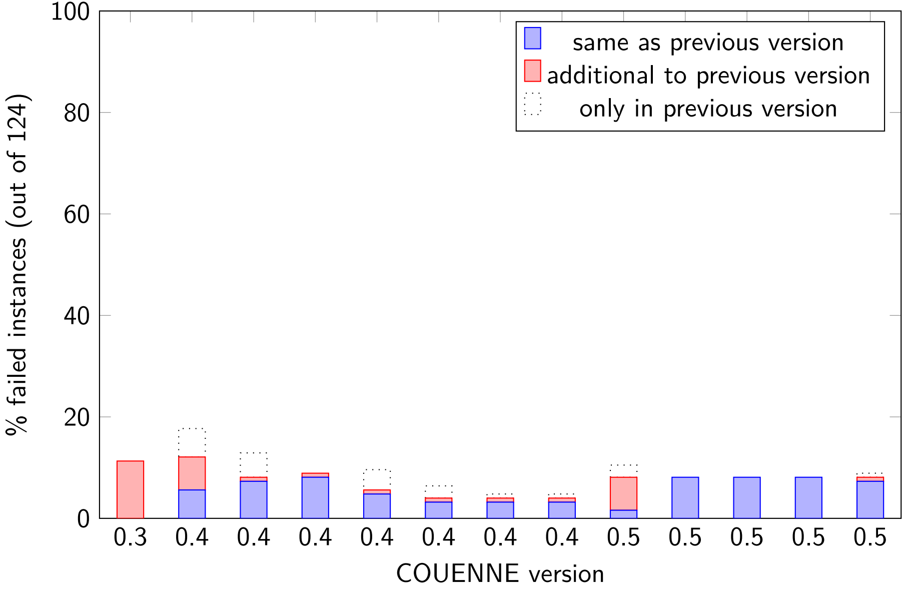
ANTIGONE: Solving time, instances that never failed (175)



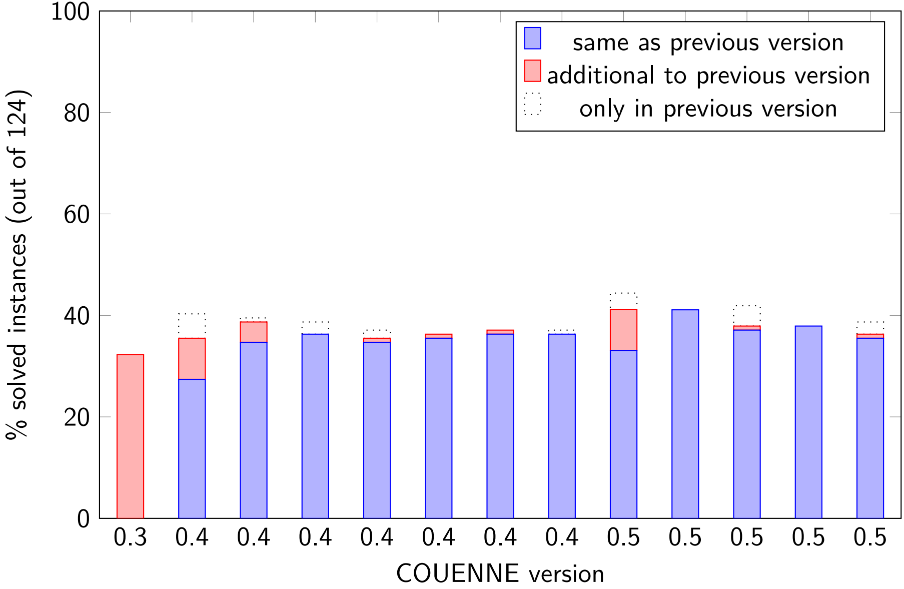
ANTIGONE: Solving time on instances solved by all (109)



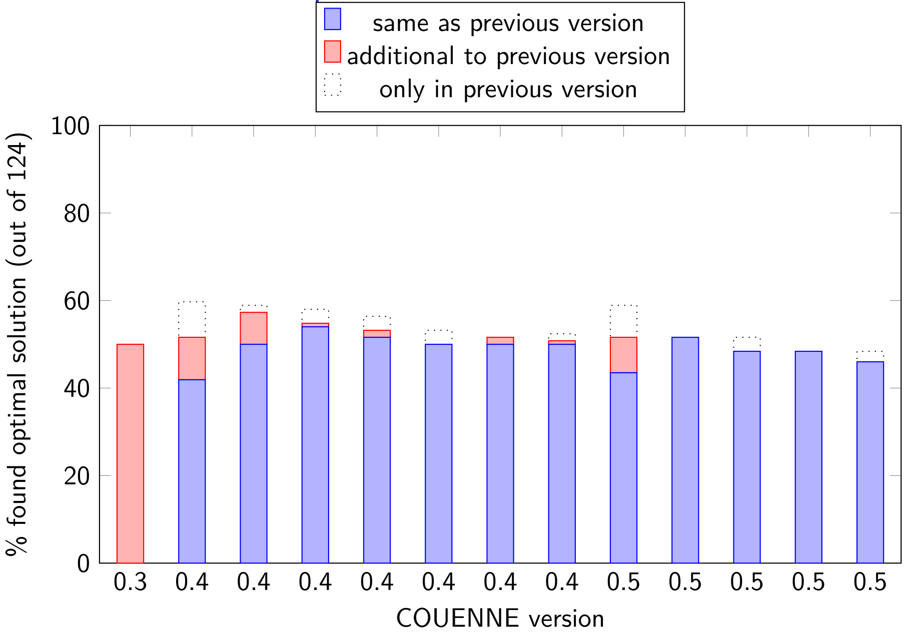
COUENNE: Fails



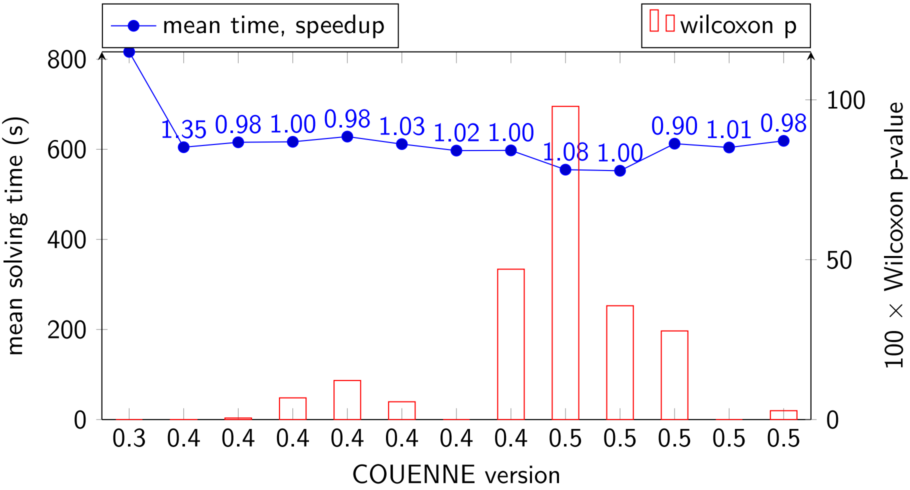
COUENNE: Solved



COUENNE: Found optimal solution

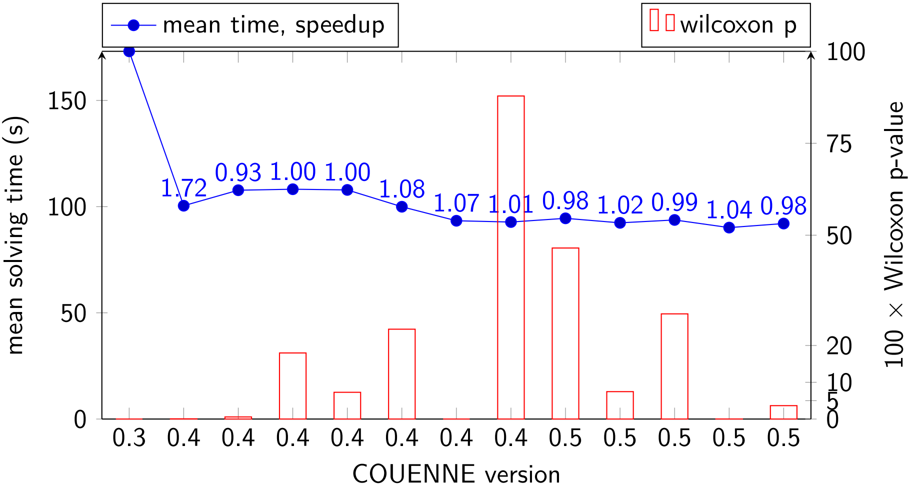


COUENNE: Solving time, instances that never failed (96)



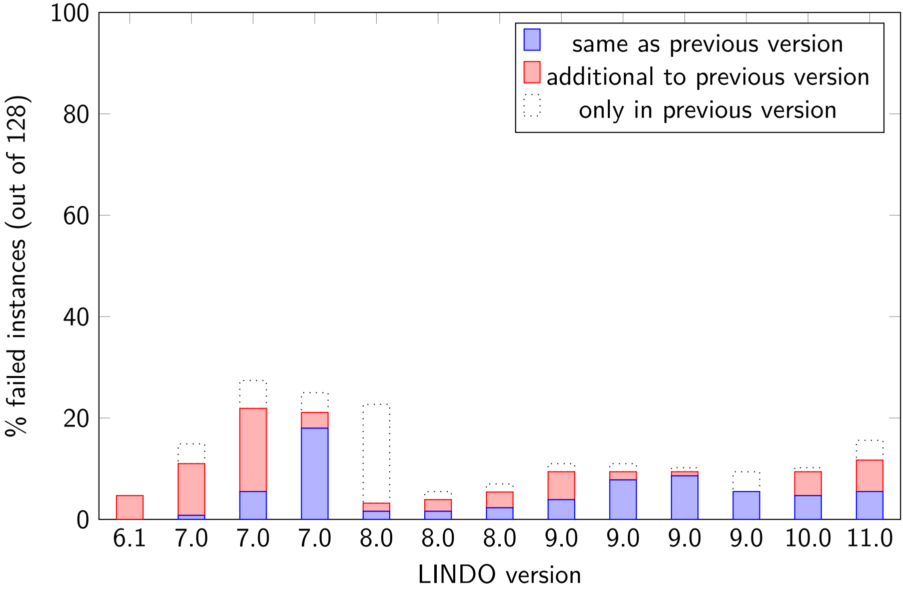
Overall speedup: 1.3

COUENNE: Solving time on instances solved by all (27)



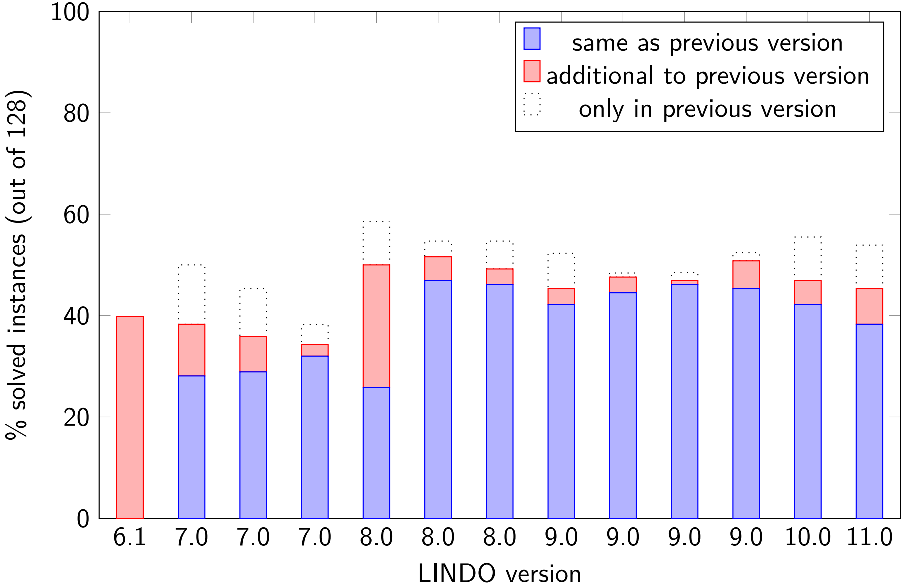
Overall speedup: 1.9

LINDO: Fails



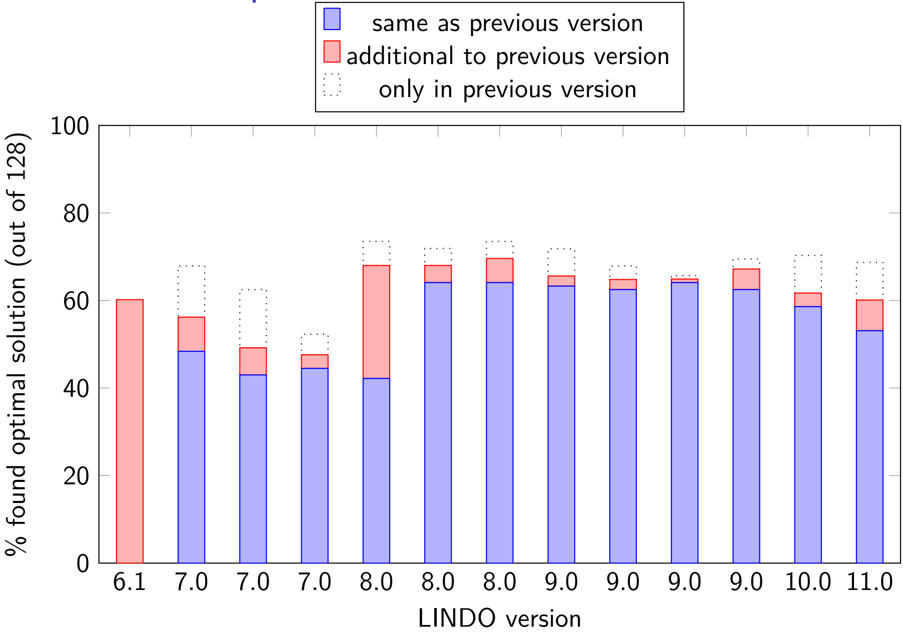
100
80
60
40
20
0
% failed instances (out of 128)
same as previous version
additional to previous version
only in previous version
6.1
7.0
7.0
7.0
8.0
8.0
8.0
9.0
9.0
9.0
9.0
10.0
11.0
LINDO version

LINDO: Solved

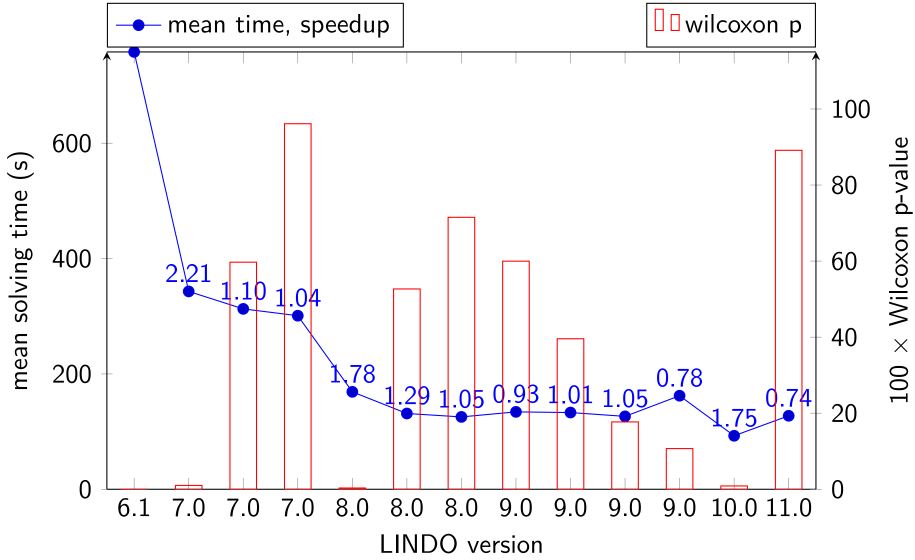


LINDO 8.0: improvements in primal heuristics for MIP (feas. pump) and

LINDO: Found optimal solution

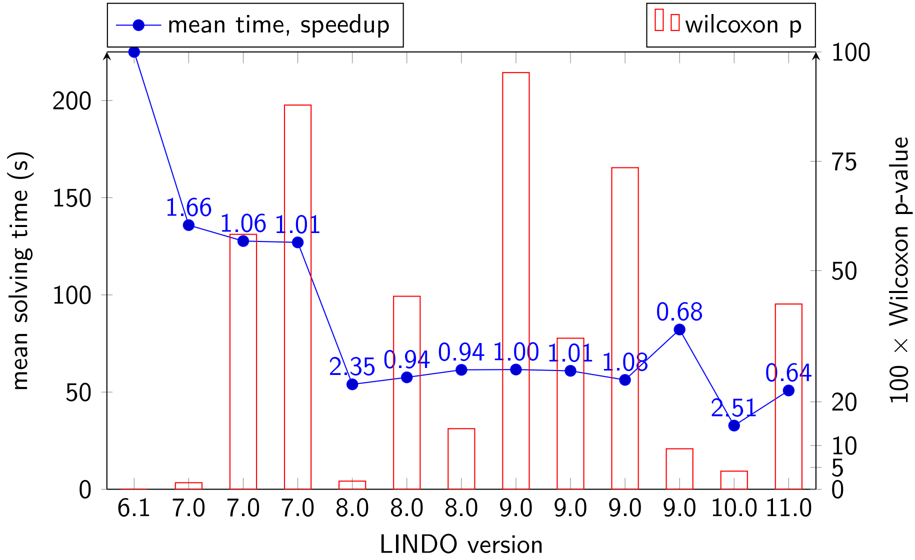


LINDO: Solving time on instances that never failed (76)



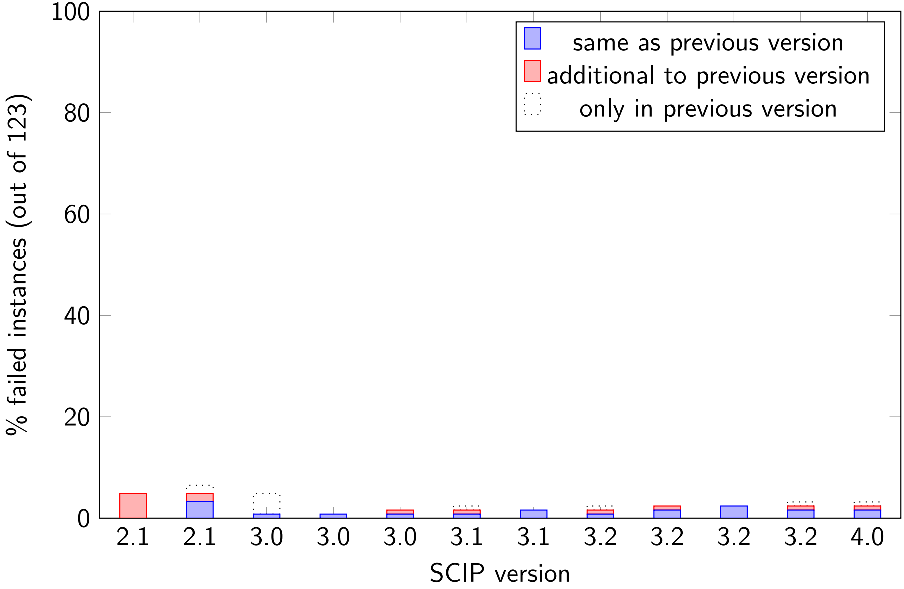
Overall speedup: 6.0

LINDO: Solving time on instances solved by all vers. (12)



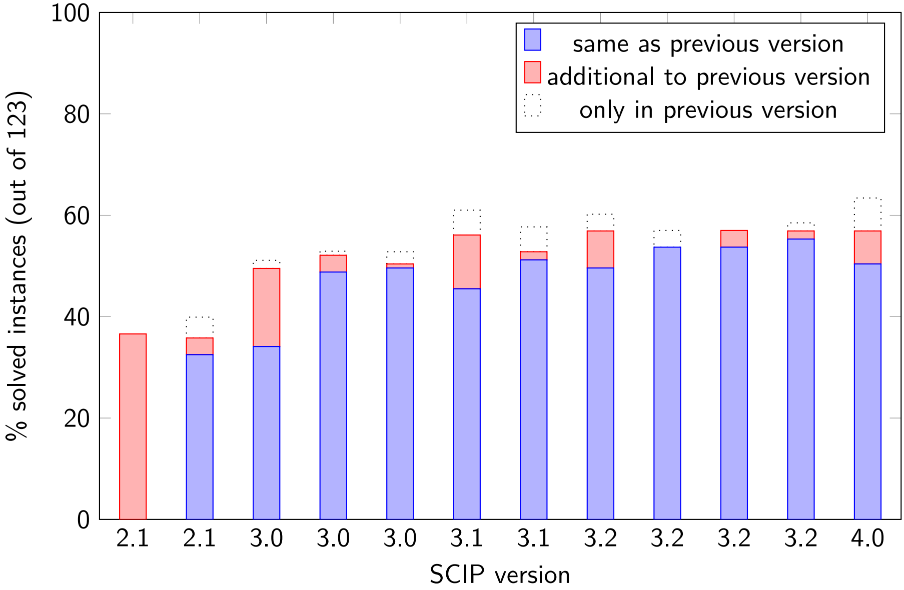
Overall speedup: 4.4

SCIP: Fails



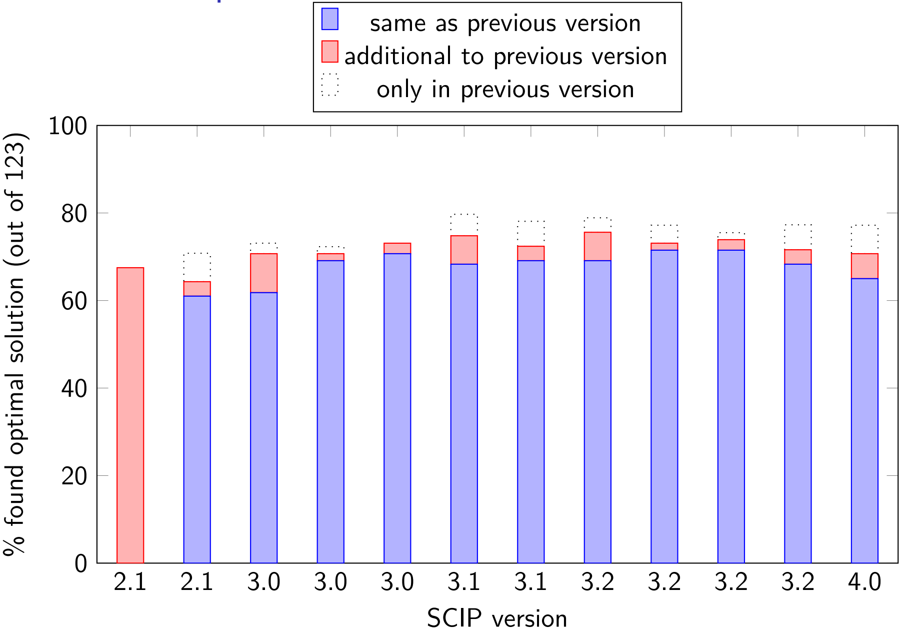
100
80
60
40
20
0
% failed instances (out of 123)
same as previous version
additional to previous version
only in previous version
2.1
2.1
3.0
3.0
3.0
3.1
3.1
3.2
3.2
3.2
3.2
4.0
SCIP version

SCIP: Solved

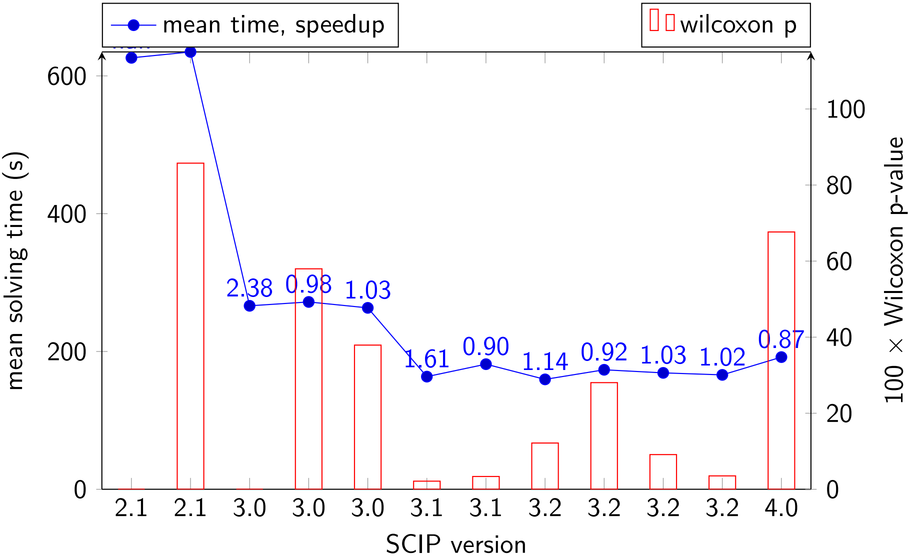


100
80
60
40
20
0
% solved instances (out of 123)
same as previous version
additional to previous version
only in previous version
2.1
2.1
3.0
3.0
3.0
3.1
3.1
3.2
3.2
3.2
3.2
4.0
SCIP version

SCIP: Found optimal solution

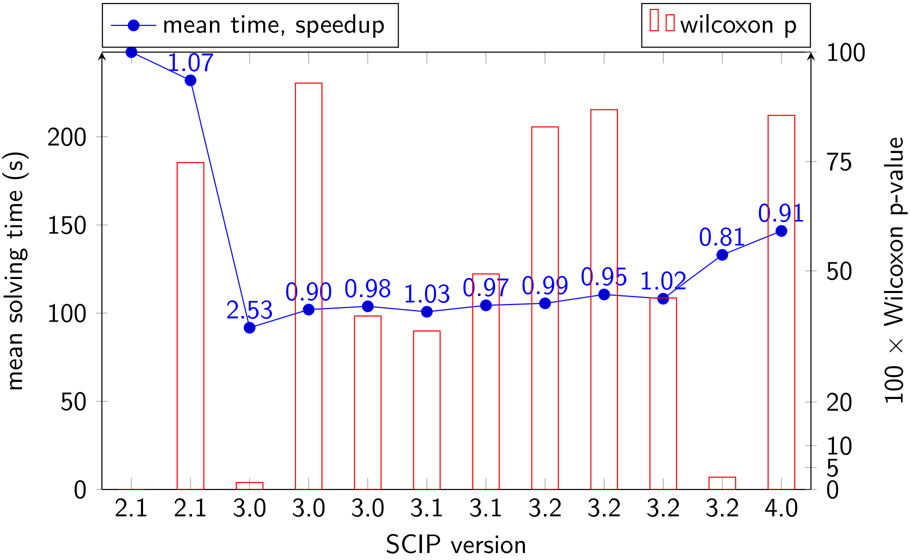


SCIP: Solving time on instances that never failed (111)



Overall speedup: 3.3

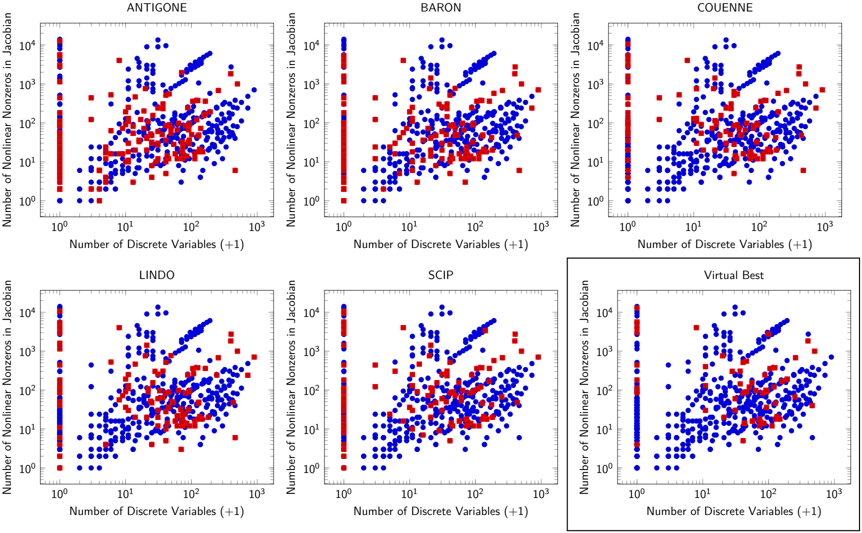
SCIP: Solving time on instances solved by all vers. (32)



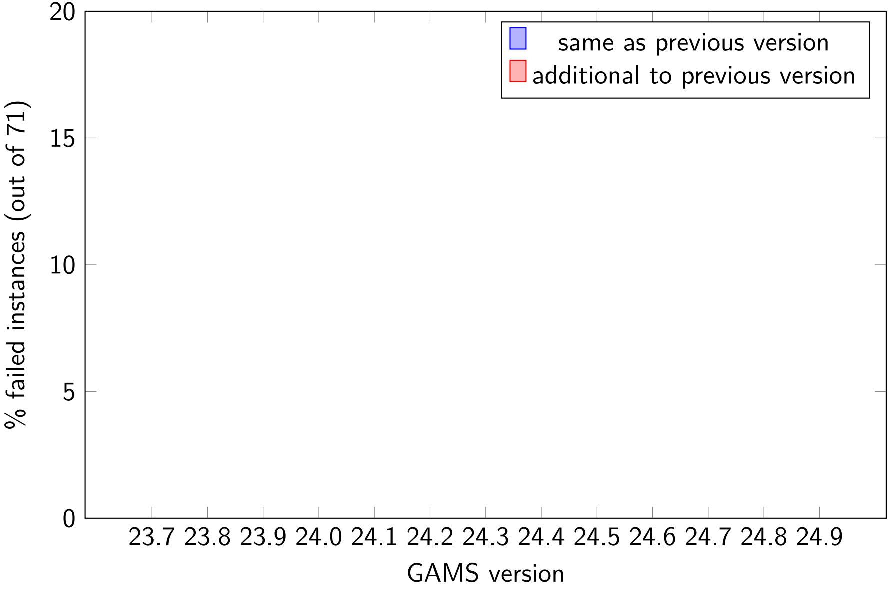
Overall speedup: 1.7

“Virtual Best” Solver

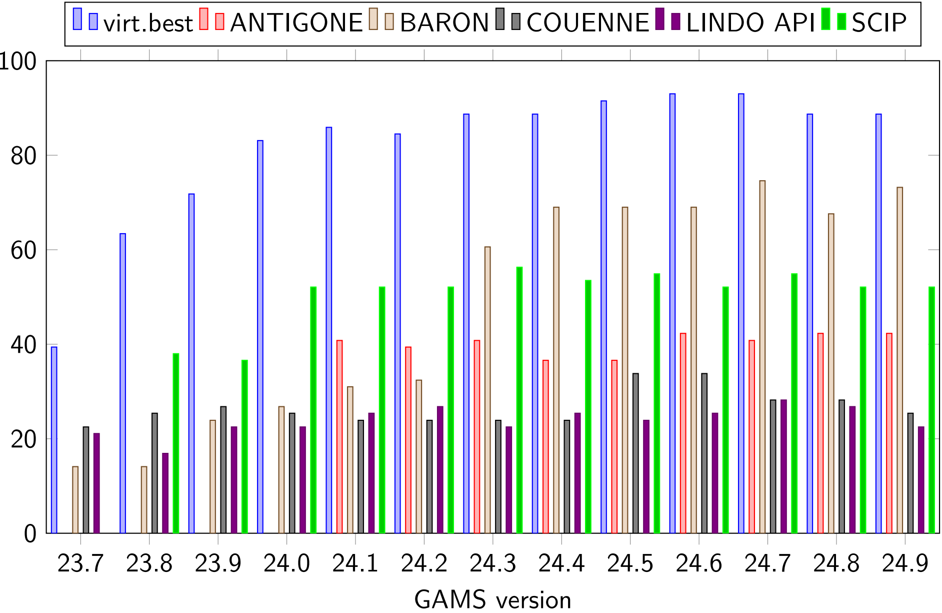
- common subset of instances
- for each instance and GAMS version, pick best results among all solvers



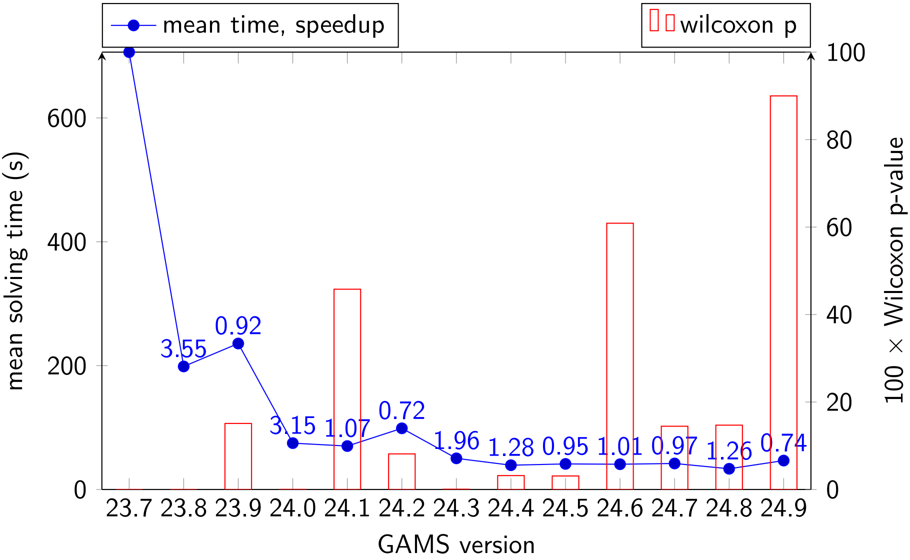
Virtual Best: Fails



Virtual Best: Solved (in %, out of 71)

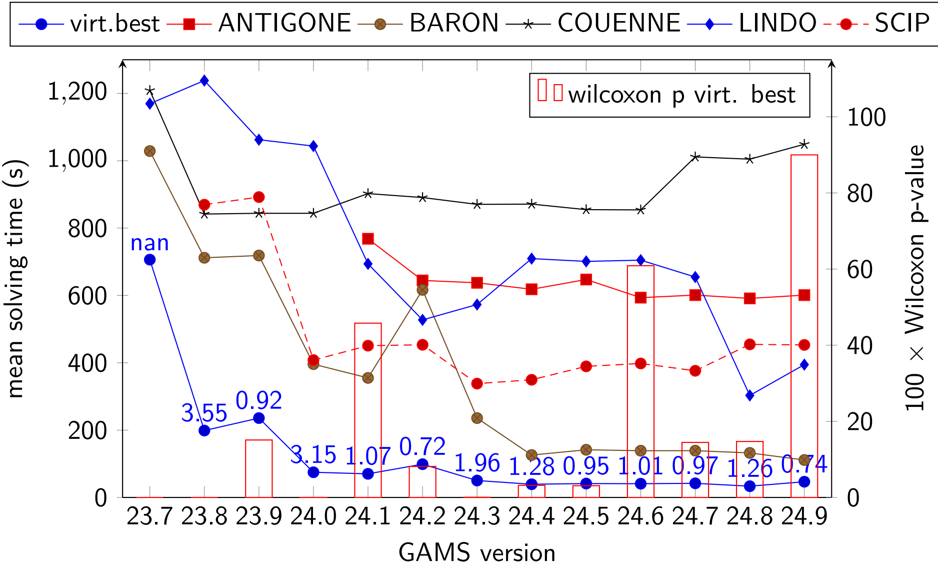


Virtual Best: Solving time, instances that never failed (71)



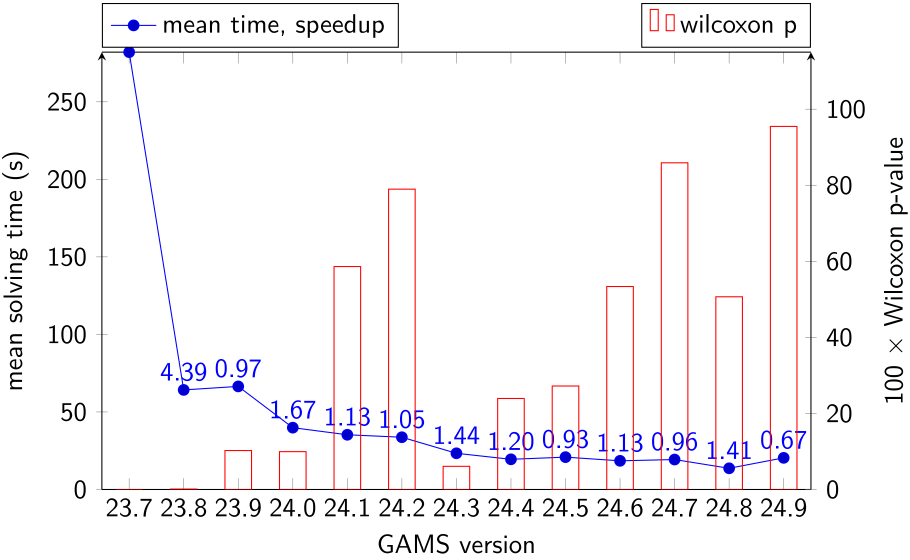
Overall speedup: 15.1

Virtual Best: Solving time, instances that never failed (71)



Overall speedup: 15.1

Virtual Best: Solving time on instances solved by all (20)



Overall speedup: 13.8