Notes on MIPLIB and Benchmarking

Thorsten Koch

Why are we benchmarking?

- If you have a particular problem to solve, it typically belongs to exactly one class of instances!
- Benchmarking is mostly useful for those who develop solvers.
- It is a projection of a multidimensional vector into a 1-dimensional one. Something will be lost.

What is the MIPLIB

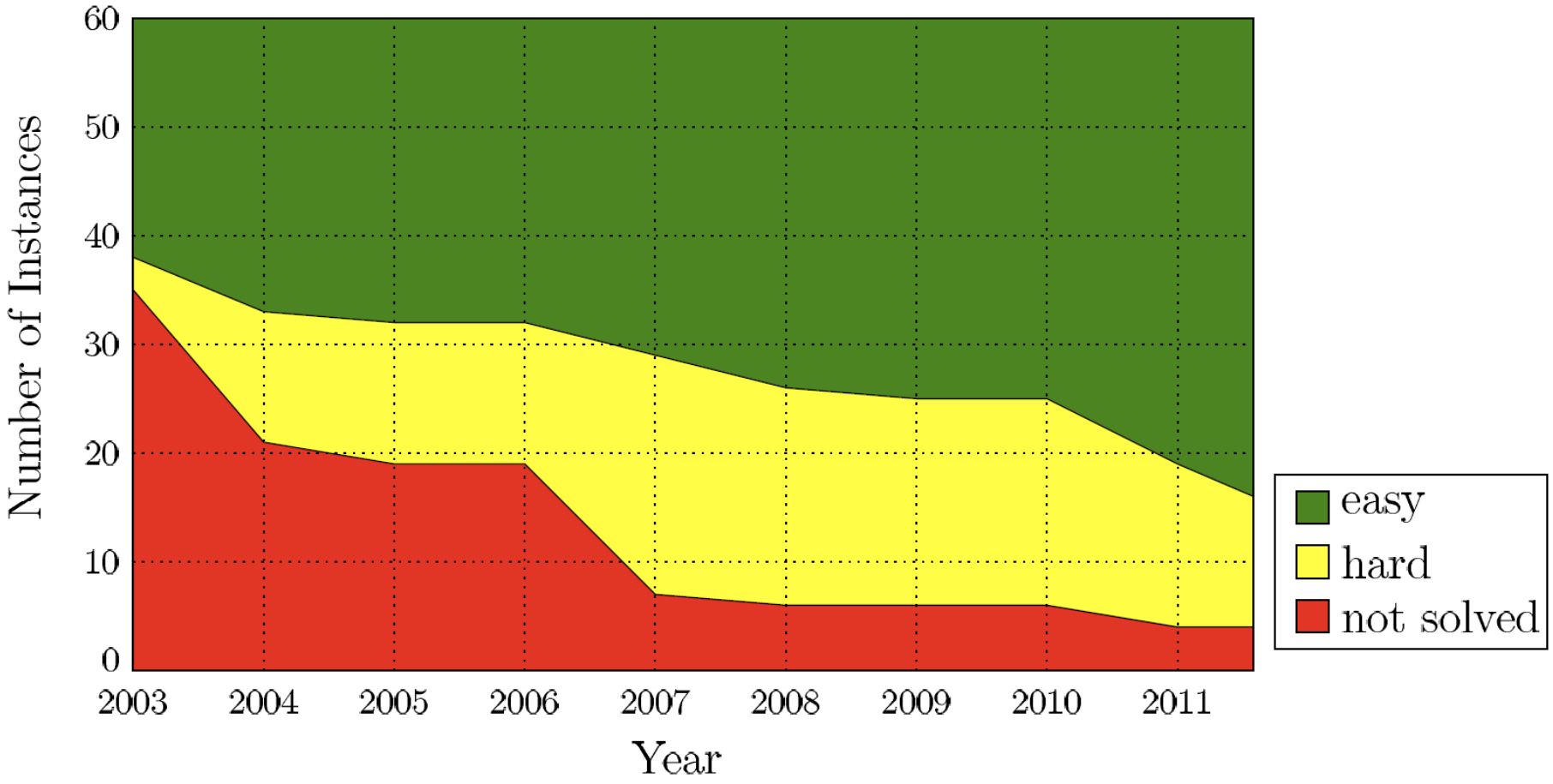
Version	Year	#	Who	Reference
1	1991	61	Bixby, Boyd, Indovina	
2	1992	61	Bixby, Boyd, Indovina	SIAM News 25, 15 (1992)
3	1996	65	Bixby, Ceria, McZeal, Savelsberg	Optima 58, 12-15 (1998)
4	2003	60	Achterberg, Koch, Martin	ORL 34(4) 361-372 (2006)
5	2010	87/361	Koch, Achterberg, Andersen, Bastert, Berthold, Bixby, Danna, Gamrath, Gleixner, Heinz, Lodi, Mittelmann, Ralphs, Salvagnin, Steffy, Wolter	MPC 3, 103-163 (2011)

The MIPLIB is a diverse collection of challenging real-world mixed integer programming (MIP) instances from various academic and industrial applications suited for benchmarking and testing of MIP solution algorithms.

The *Benchmark* set consist only of instances that could be solved in 2010 within two hours on a high-end personal computer by at least two MIP solvers.

MIPLIB 5 still contains 2 instances from Version 2, 7 instances from Version 3, and 17 instances from Version 4.

Development of MIPLIB 2003



easy can be solved within an hour on a contemporary pc with a state-of-the-art solver
hard are solvable but take a longer time or require specialized algorithms
open, *i.e. not solved* instances for which the optimal solution is not known

What is important in benchmarking?

- Testset (size and selection)
- Time limit
- How to deal with instances that hit the time limit? (One/some/all solvers?)
- What is the competition exactly? (mipgap=0)
- How to deal with wrong results?
- How to combine results from instances to overall result?
- Memory limit
- Computational environment (CPU, Cores, NUMA, etc.)

Example

Chose a convenient time limit.

LocalSolver wins on some of the hardest MIPLIB instances ...

Below are some results of LocalSolver on some of the hardest MIPLIB instances. As you will notice by looking to the results (marked by *), LocalSolver wins against both Gurobi and IBM ILOG Cplex, the fastest commercial mixed-integer linear programming (MIP) solvers.

- Lower objective is better
- 5 minutes of running time for each solver
- Standard computer: Intel Core i7-820QM (4 cores, 1.73 GHz, 6 GB RAM, 8 MB cache)
- Default settings for each solver
- MIP-oriented models: not suitable for LocalSolver

Select the 21 out of 361 instances you like

Instances	Status	Variables	LocalSolver 3.1	Gurobi 5.5	Cplex 12.4	Optimum
opm2-z10-s2	hard	6,250	* -25,719	-19,601	-18,539	-33,826
opm2-z11-s8	hard	8,019	* -33,028	-21,661	-18,883	-43,485
opm2-z12-s14	hard	10,800	* -46,957	-11,994	-36,469	-64,291
opm2-z12-s7	hard	10,800	* -46,034	-12,375	-30,887	-65,514
pb6	hard	462	-62	-62	-62	-63
queens-30	hard	900	-38	-36	-39	-40
dc11	open	37,297	11,100,000	21,300,000	1,840,402	unknown
ds-big	open	6,020	9,844	62,520	5,256	unknown
ex1010-pi	open	25,200	249	251	247	unknown
ivu06-big	open	1,812,044	* 479	9,416	678	unknown
ivu52	open	1,423,438	4,907	16,880	3,285	unknown
mining	open	753,404	* -65,720,600	902,969,000	no solution	unknown
ns-1853823	open	213,440	* 2,820,000	4,670,000	no solution	unknown
pb-simp-nonunif	open	23,848	* 90	140	94	unknown
ramos3	open	2,187	* 223	274	267	unknown
rmine14	open	32,205	* -3469	-170	-968	unknown
rmine21	open	162,547	* -3657	-184	no solution	unknown
rmine25	open	326,599	* -3052	-161	no solution	unknown
stein1	open	13,741	256,620,000	315,186,152	54,820,419	unknown
sts405	open	405	342	342	354	unknown
sts729	open	729	648	648	665	unknown

For many instances, the conclusion is that none of the fastest MIP solver is currently able to provide high-quality solutions quickly in short running times, as it is needed today in the practice of optimization and operations research.

More details on instances can be found on MIPLIB 2010.

From the classes you like take a few more

Example

LocalSolver wins on some of the hardest MIPLIB instances ...

Below are some results of LocalSolver on some of the hardest MIPLIB instances. As you will notice by looking to the results (marked by *), LocalSolver wins against both Gurobi and IBM ILOG Cplex, the fastest commercial mixed-integer linear programming (MIP) solvers.

- Lower objective is better
- 5 minutes of running time for each solver
- Standard computer: Intel Core i7-820QM (4 cores, 1.73 GHz, 6 GB RAM, 8 MB cache)
- Default settings for each solver
- MIP-oriented models: not suitable for LocalSolver

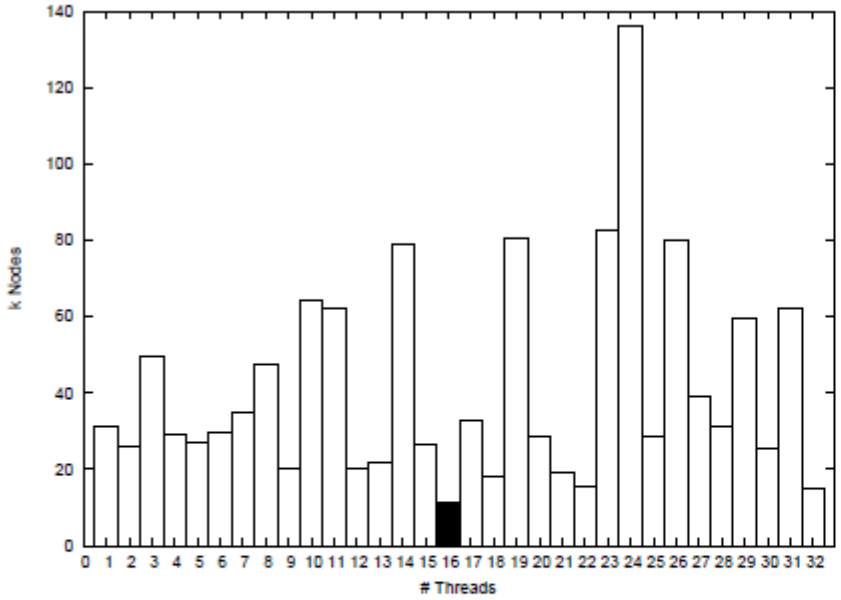
What exactly is the game to win?

Instances	Status	Variables	LocalSolver 3.1	Gurobi 5.5	Cplex 12.4	Optimum
opm2-z10-s2	hard	6,250	* -25,719	-19,601	-18,539	-33,826
opm2-z11-s8	hard	8,019	* -33,028	-21,661	-18,883	-43,485
opm2-z12-s14	hard	10,800	* -46,957	-11,994	-36,469	-64,291
opm2-z12-s7	hard	10,800	* -46,034	-12,375	-30,887	-65,514
pb6	hard	462	-62	-62	-62	-63
queens-30	hard	900	-38	-36	-39	-40
dc11	open	37,297	11,100,000	21,300,000	1,840,402	unknown
ds-big	open	6,020	9,844	62,520	5,256	unknown
ex1010-pi	open	25,200	249	251	247	unknown
ivu06-big	open	1,812,044	* 479	9,416	678	unknown
ivu52	open	1,423,438	4,907	16,880	3,285	unknown
mining	open	753,404	* -65,720,600	902,969,000	no solution	unknown
ns-1853823	open	213,440	* 2,820,000	4,670,000	no solution	unknown
pb-simp-nonunif	open	23,848	* 90	140	94	unknown
ramos3	open	2,187	* 223	274	267	unknown
rmine14	open	32,205	* -3469	-170	-968	unknown
rmine21	open	162,547	* -3657	-184	no solution	unknown
rmine25	open	326,599	* -3052	-161	no solution	unknown
siena1	open	13,741	256,620,000	315,186,152	54,820,419	unknown
sts405	open	405	342	342	354	unknown
sts729	open	709	648	648	665	unknown

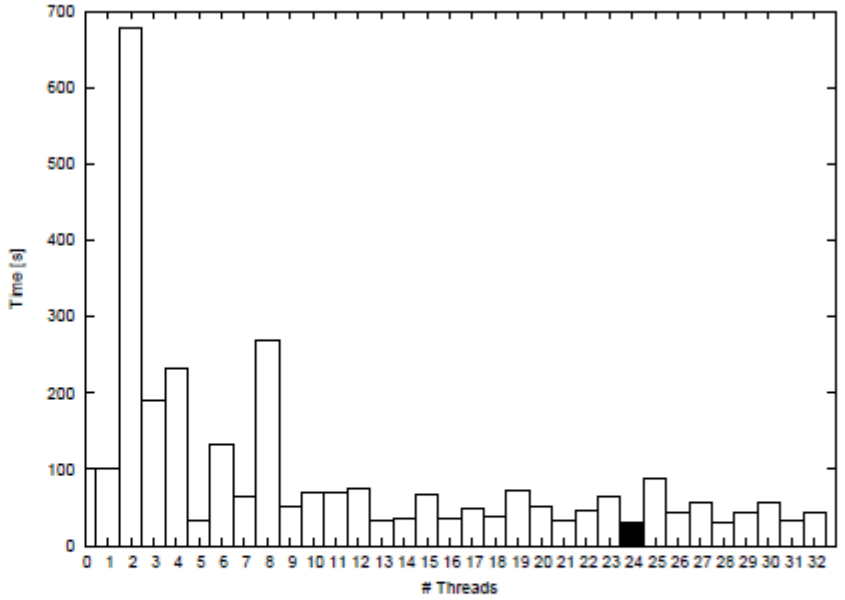
For many instances, the conclusion is that none of the fastest MIP solver is currently able to provide high-quality solutions quickly in short running times, as it is needed today in the practice of optimization and operations research.

More details on instances can be found on MIPLIB 2010.

Performance variation due to # of threads



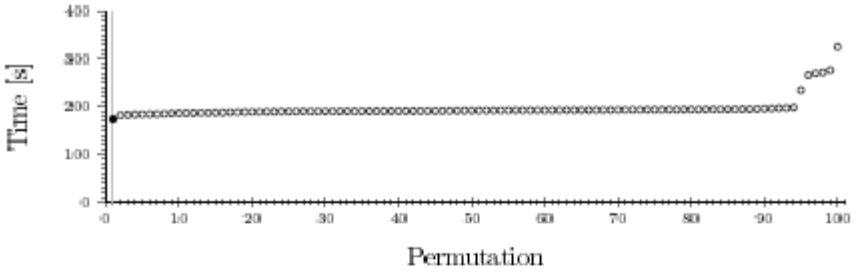
(a) Total number of nodes explored (CPLEX)



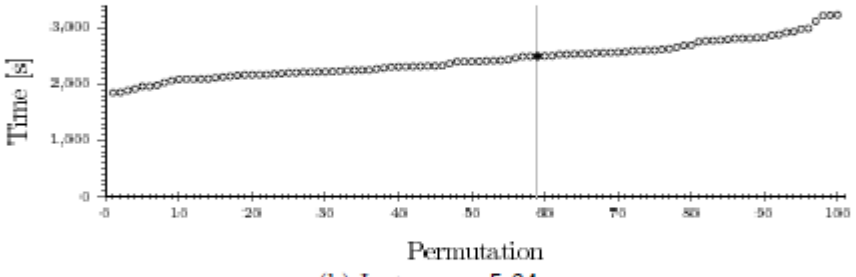
(b) Wall clock solution time (GUROBI)

Fig. 4: Example of performance variability depending on the number of threads. Instance roll3000 on a 32 core computer. Filled bar indicates minimum

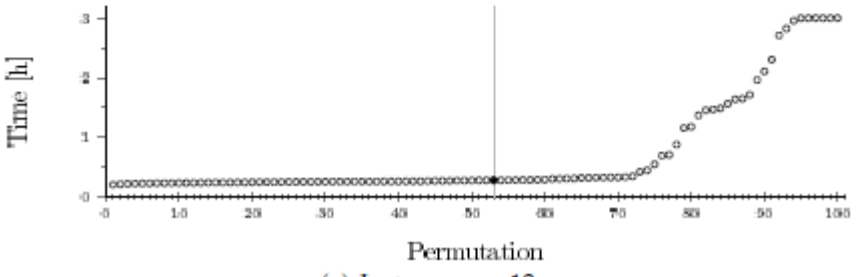
Performance variation due to permutation



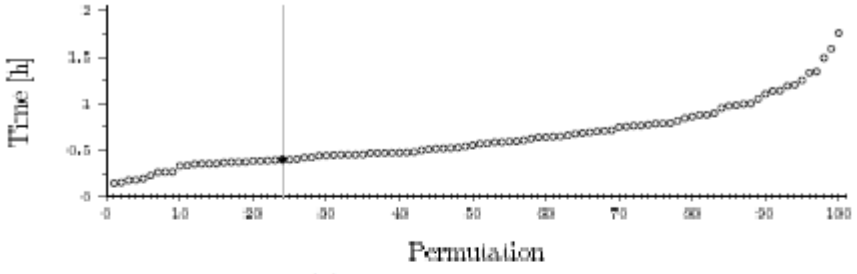
(a) Instance ex9



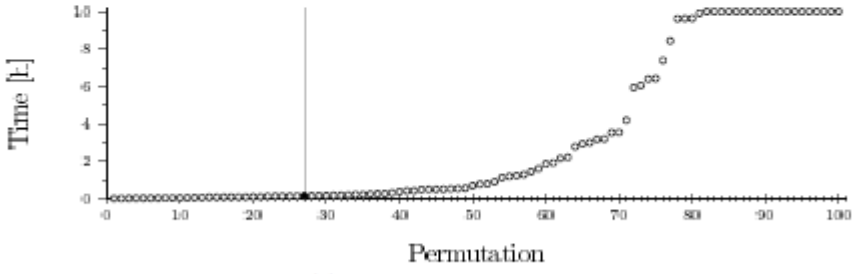
(b) Instance pg5_34



(c) Instance neos13



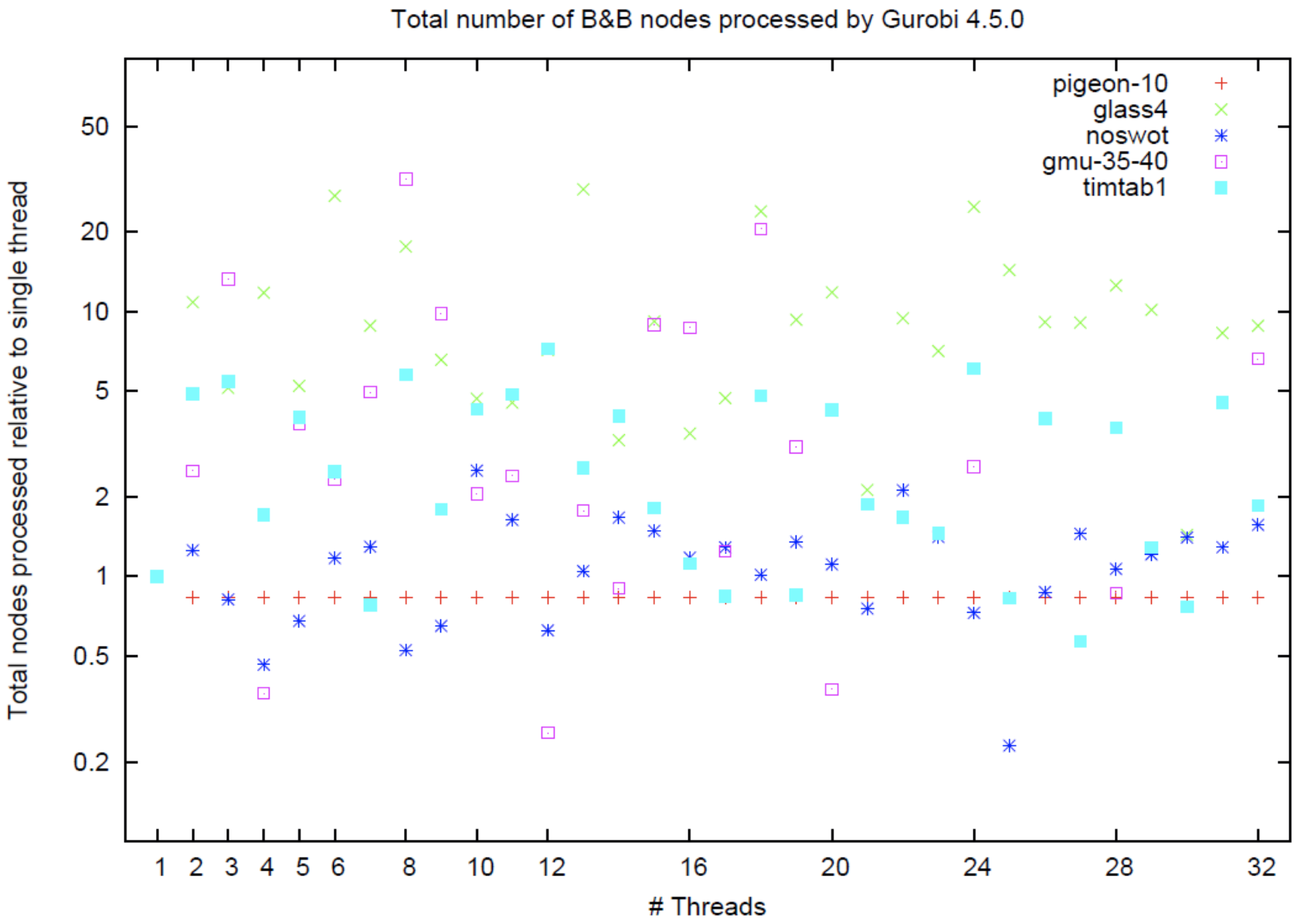
(d) Instance bnatt350



(e) Instance enlight13

Fig. 3: Solution times for 100 permutations

B&B nodes


Total number of B&B nodes processed by Gurobi 4.5.0
Total nodes processed relative to single thread
Threads
pigeon-10
glass4
noswot
gmu-35-40
timtab1
50
20
10
5
2
1
0.5
0.2
1
2
3
4
5
6
8
10
12
16
20
24
28
32

How to compare MIP solver performance?

There are two components to it:

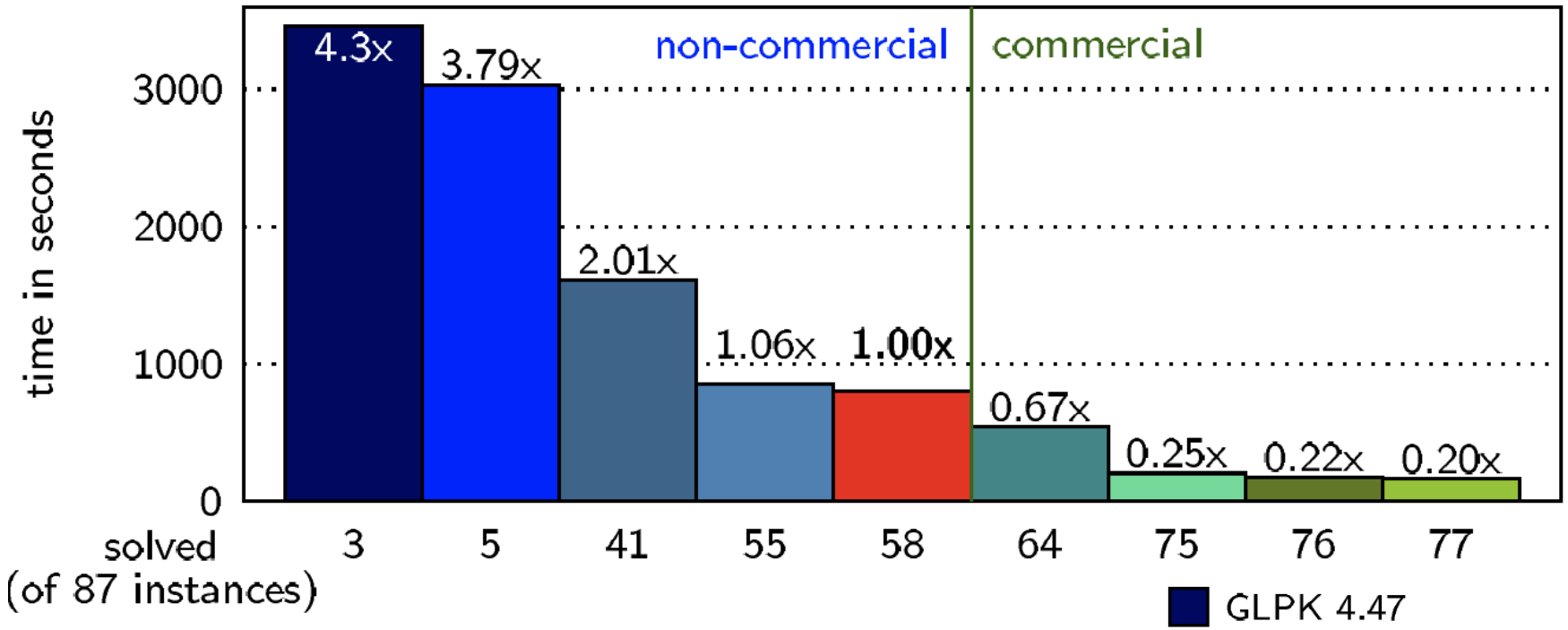
- Ability to solve an instance (within some timeframe)
- Time to solve

You have always to see both numbers!

If two solvers are compared there are two possibilities:

1. Both can solve all instances.
2. There are instances only one solver can solve
3. There are instances both cannot solve

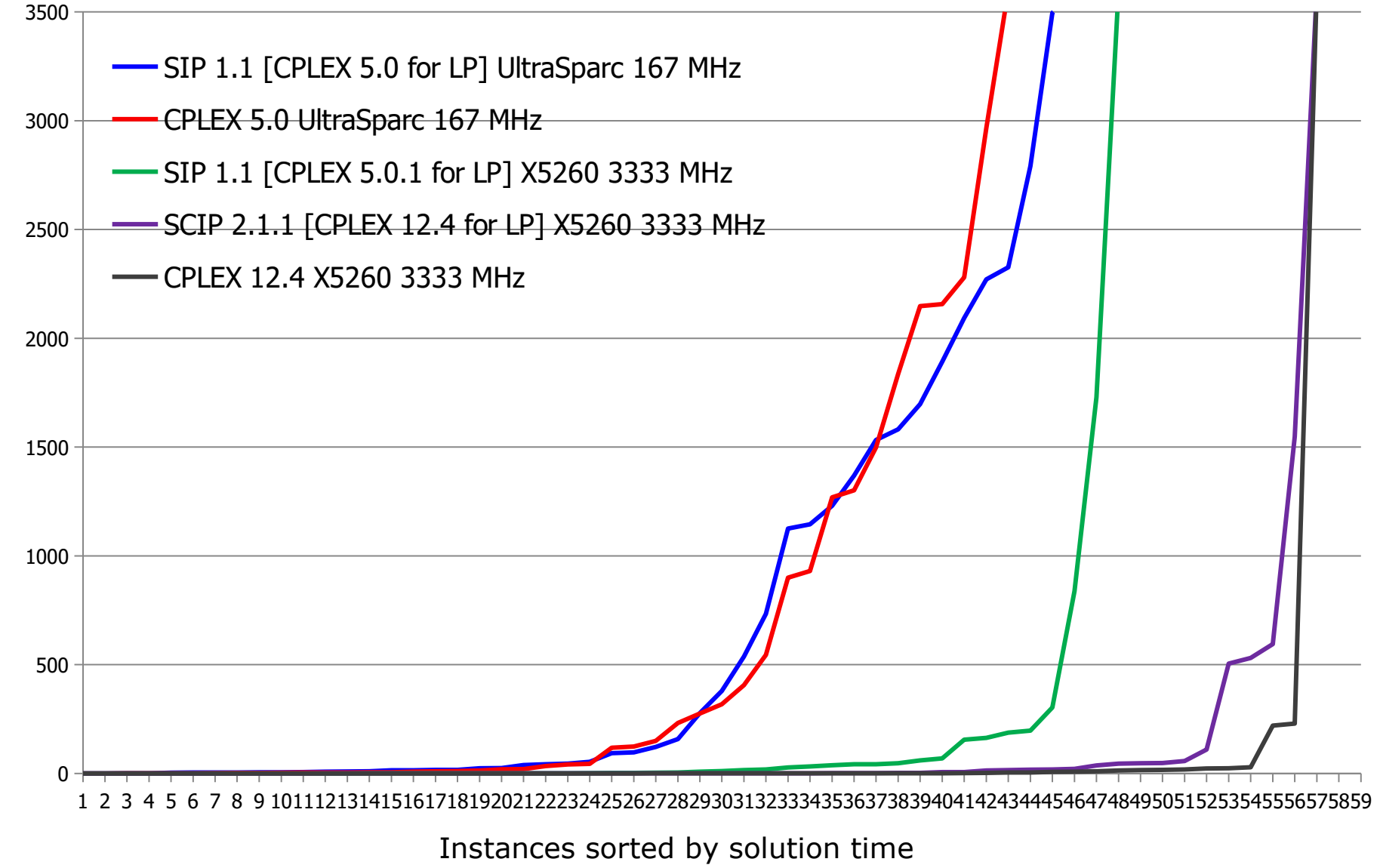
MIP Solver performance 2012, 1 thread



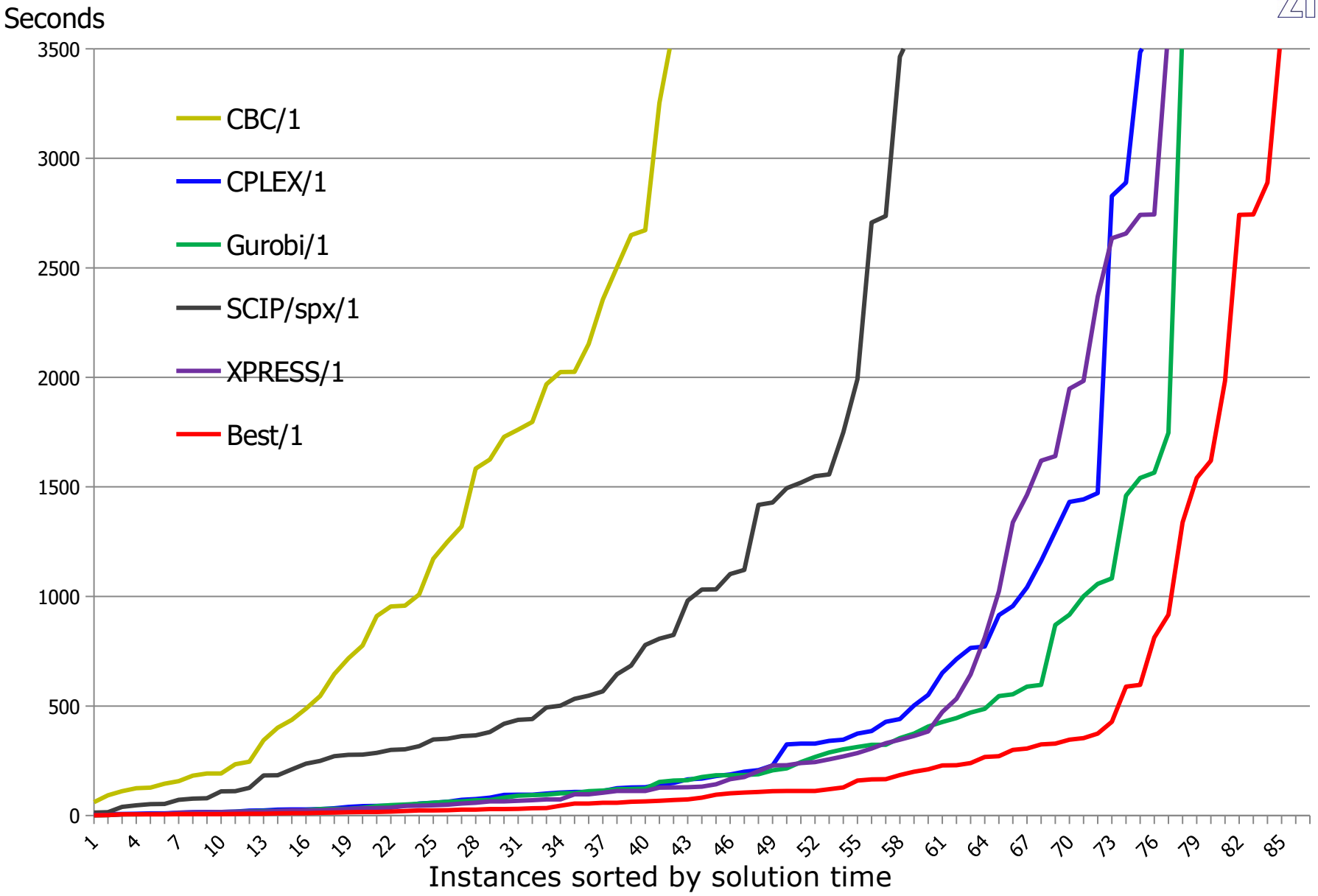
Geometric mean of results taken from the homepage of Hans Mittelmann (11. Aug 2012)
`http://plato.asu.edu/ftp/milpc.html`

Unsolved or failed instances are accounted for with the time limit of 1 hour.

1998 Benchmark Set



2012 Performance on MIPLIB 2010 / 1 thread

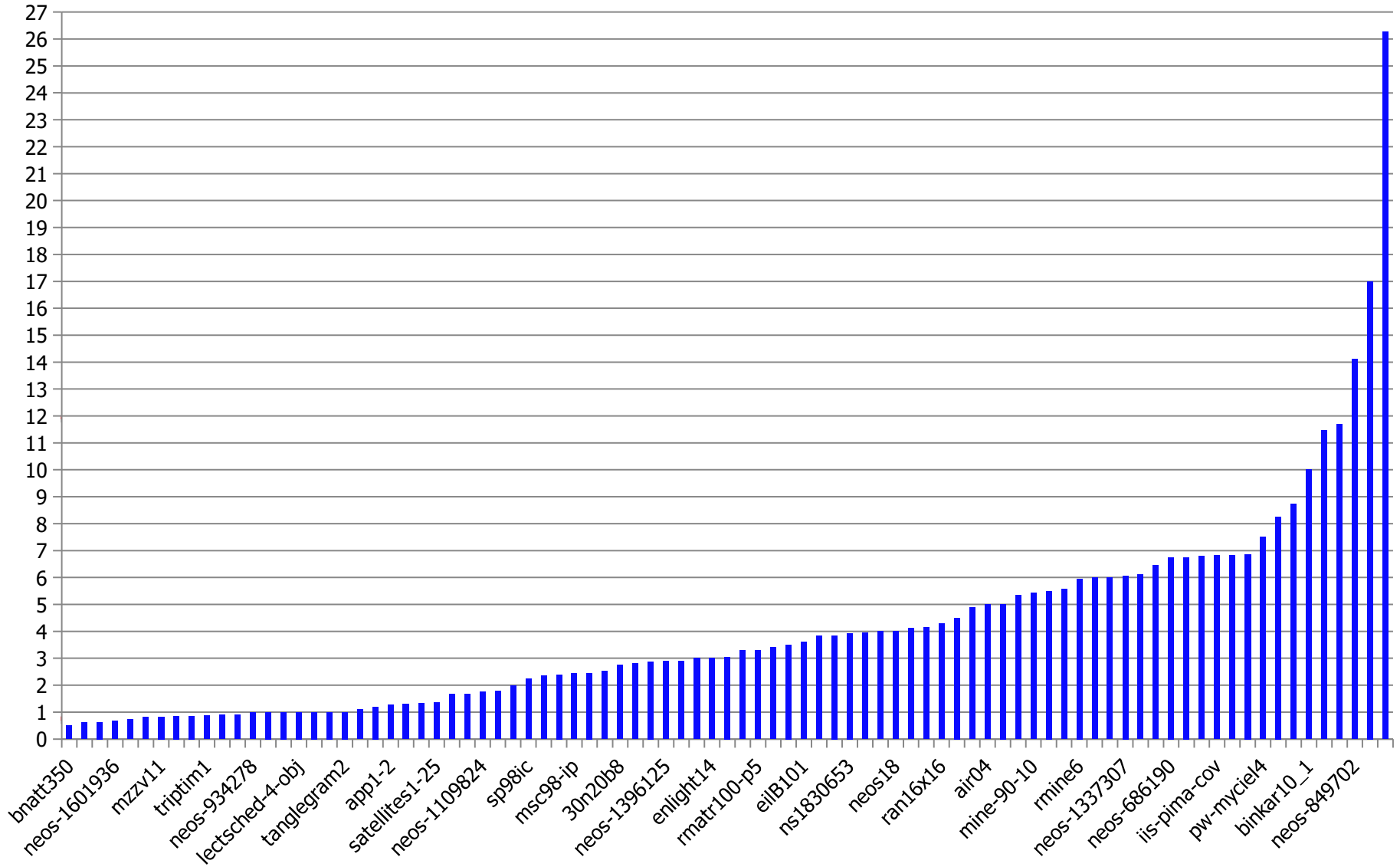


Progress in Numbers

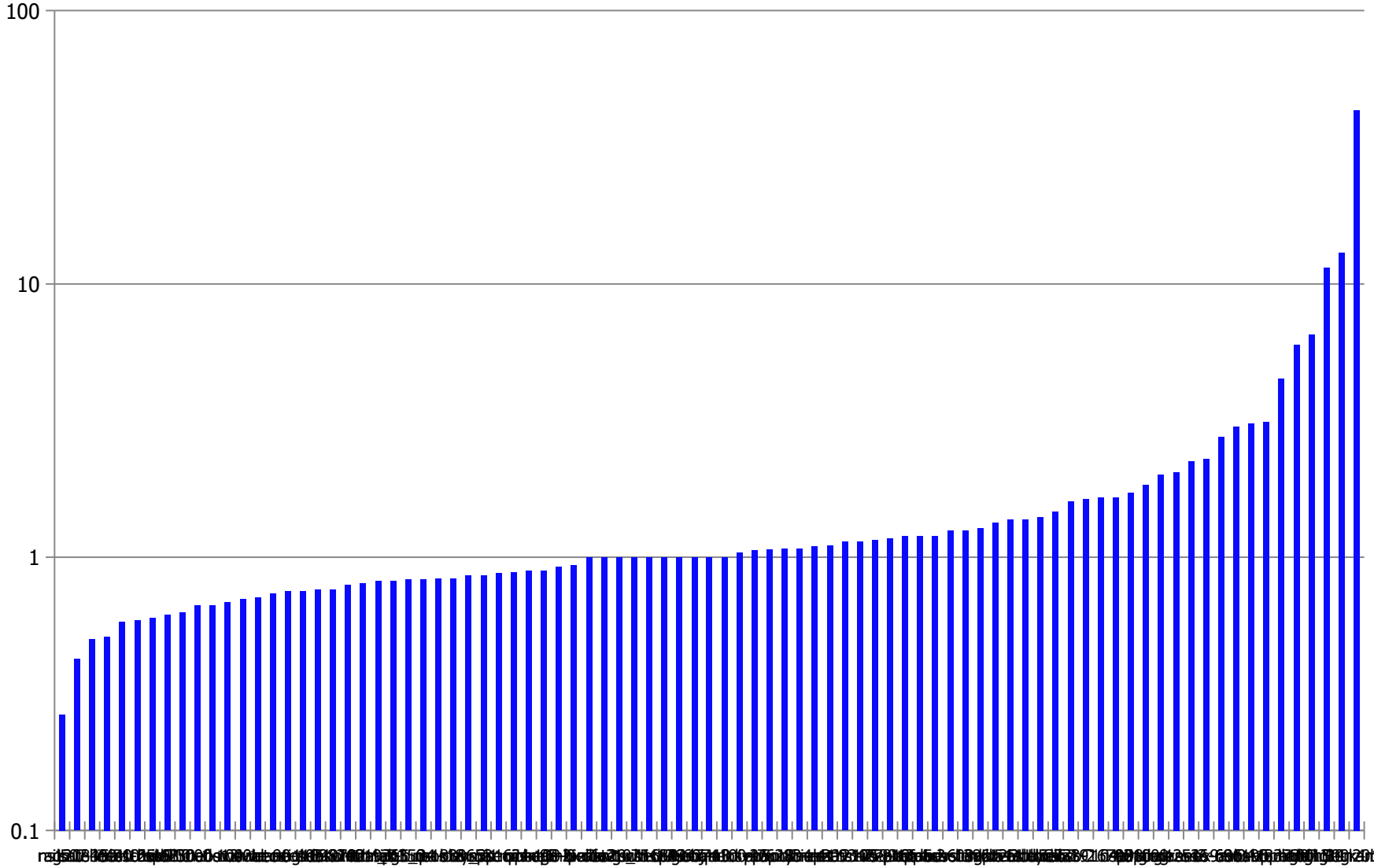
	2011	2012	Speed-up
Best/1 geom. mean time [s]	105	81	**23%**
Best/12 geom. mean time [s]	34	28	**17%**
Best/1 not optimal [#]	4	3	
Best/12 not optimal [#]	1	0	
Speedup 1 to 12 [×]	3.1	2.8	
Max(C,G,X)/Min(C,G,X) 1 thr.	600	259	
Max(C,G,X)/Min(C,G,X) 12 thr.	1138	107	
Best/1 geom. mean nodes [#]	2108	1958	8%
Best/12 geom. mean nodes [#]	2570	2406	7%
Max(C,G,X)/Min(C,G,X) 1 thr.	40,804	32,600	
Max(C,G,X)/Min(C,G,X) 12 thr.	10,499	887,849	

Single solvers have speed-ups up to 42%

2012 Parallel speedup Best 1/12 per instance



Speedup Best/12 2011:2012 per instance



What is important in benchmarking?

- Testset (size and selection) -> Large and diverse, permuted
- Time limit -> Find quiescence point
- How to deal with instances that hit the time limit?
  (One/some/all solvers?) -> Include all
- How to deal with wrong results? **Mark them!**
- How to combine results from instances to overall result?
  Geometric mean!
- Memory limit ->
  Should not make a big difference between the solvers.
- Computational environment (CPU, Cores, NUMA, etc.)
  Carefully check what was selected.

Thank you very much!