Quality Assurance, Performance Analysis, and The GAMS/COIN-OR Solvers

Steven P. Dirkse
sDirkse@gams.com

GAMS Development Corp.
http://www.gams.com

INFORMS International
Puerto Rico 10 July 2007

Welcome/Agenda

- Benchmarking & QA tools
- GAMS/COIN-OR Solvers
- GAMS/Examiner
- Computational Results & Summary

Agenda

- Benchmarking & QA tools
- GAMS/COIN-OR Solvers
- GAMS/Examiner
- Computational Results & Summary

Challenges in Benchmarking and QA

- QA is not glamorous – where's the novelty & publications?
  - *Make the tools used public - "open-source" them*
  - *Make it a group project with high priority*
- QA & Benchmarking is time-consuming
  - *Create standard libraries of test problems, categorized for convenient access*
  - *Automate the creation of test scripts, the collection of data, and the creation and display of statistics*
- Benchmark results are can be subjective, misleading, or hard to believe – conclusions can be hard to draw.
  - *Test libraries and automation reduce subjective element and make benchmarks reproducible, hence believable*
  - *Automate the creation and display of statistics*

Performance World

Performance World

Welcome to the Performance World!

Performance World is a forum for discussion and dissemination of information and tools about all aspects of performance testing of solvers for mathematical programming problems. This world has been established in response to user demands for independent and reproducible performance results.

Overall performance highly depends on problem formulation, solver, and tuning parameters. Our performance tools are designed to serve the different needs of our user community. One user may be interested in finding the most reliable way to solve a proprietary or classified model. On the other hand, an academic researcher may be interested in testing a new algorithm against a set of existing test problems and competing approaches. The main features are:

- Uniform access to a comprehensive set of established and new test problems
- Automation tools for collecting performance measurements
- Tools for analyzing and visualizing test results

Editorial Board
PerformanceLib
Performance Tools
Performance List
Related Links
Search
Contact

What's New:

- Try our online PAVER Server for automated performance analysis and visualization, batch file creation and model translation
- New tools for analyzing non-convex or discrete models
- MINLP type models from the MINLP World have been added to the PerformanceLib

Performance Libraries

- Performance tests require public test libraries
  - Creating models for this is not feasible
  - Shared test libraries allow reproducible results
- PerformLIB contains multiple libraries
  - GLOBALLib - *NLP*
  - LINLib – *LP*, MIP, QCP
  - MPLLib – *LP, NLP*
  - Several others not used here

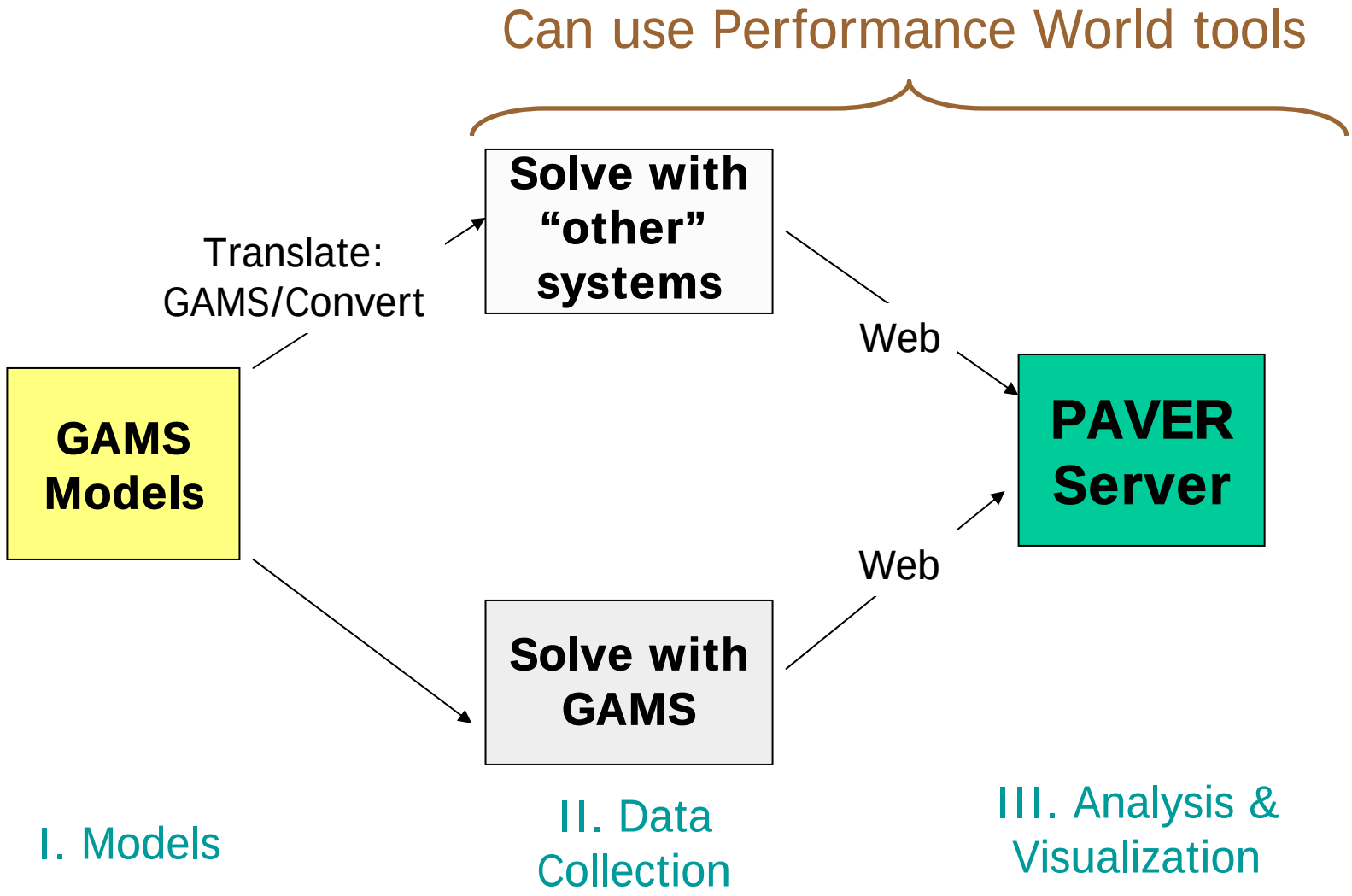
Tools: Performance Analysis

- Different objectives:
  - Solver robustness and correctness
  - Solver efficiency
  - Quality of solution (nonconvex and discrete models)

→Tools are GAMS independent

→Results in HTML format: platform independent

Open Testing Architecture



PAVER Server

- PAVER server (**P**erformance **A**nalysis and **V**isualization for **E**ffortless **R**eproducibility)

  www.gamsworld.org/performance/paver

- Online server to facilitate performance testing and analysis/visualization

- Results sent via e-mail in HTML format
  – System independent

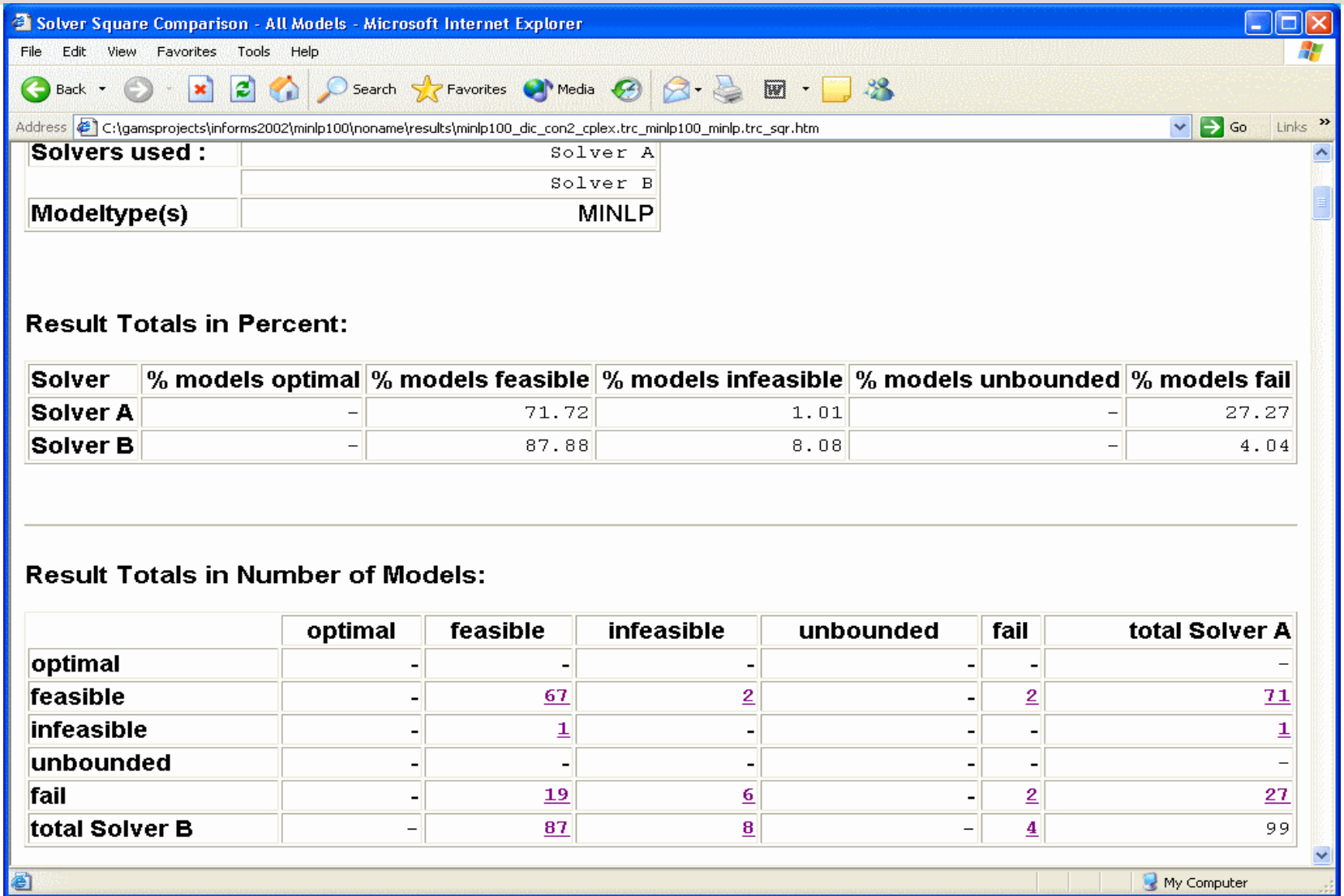
Tools: Robustness

Solver Square Utility:

- Cross comparison of solver outcomes of two solvers:
  - Optimal, integer, infeasible, unbounded, fail
- Compact tabular form for results
- Shows resource time and objective value information

→ Can use online using PAVER

PAVER: Solver Square



Solvers used :	Solver A
	Solver B
Modeltype(s)	MINLP

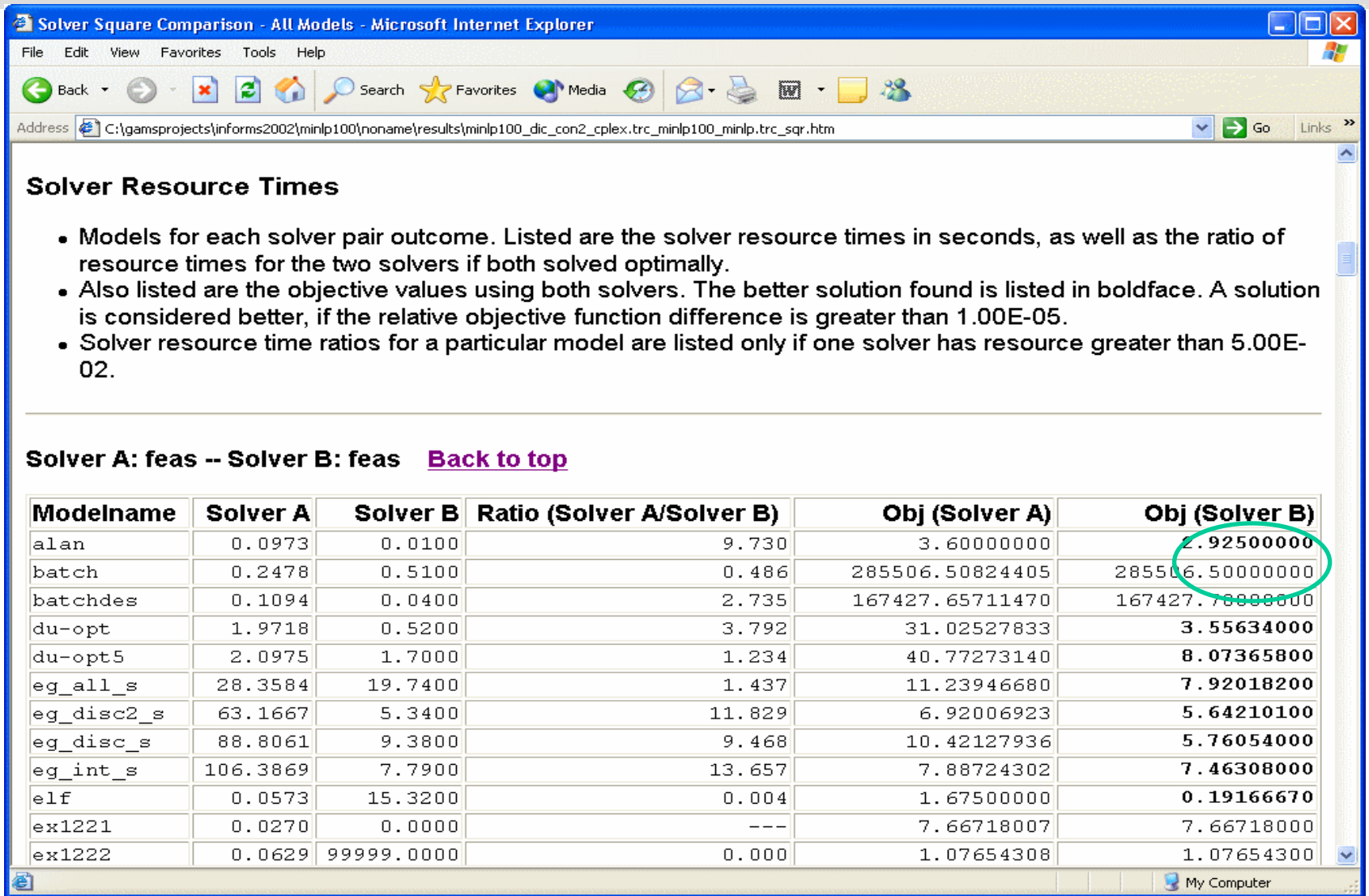
Result Totals in Percent:

Solver	% models optimal	% models feasible	% models infeasible	% models unbounded	% models fail
Solver A	-	71.72	1.01	-	27.27
Solver B	-	87.88	8.08	-	4.04

Result Totals in Number of Models:

	optimal	feasible	infeasible	unbounded	fail	total Solver A
optimal	-	-	-	-	-	-
feasible	-	67	2	-	2	71
infeasible	-	1	-	-	-	1
unbounded	-	-	-	-	-	-
fail	-	19	6	-	2	27
total Solver B	-	87	8	-	4	99

PAVER: Square (cont.)



Solver Resource Times

- Models for each solver pair outcome. Listed are the solver resource times in seconds, as well as the ratio of resource times for the two solvers if both solved optimally.
- Also listed are the objective values using both solvers. The better solution found is listed in boldface. A solution is considered better, if the relative objective function difference is greater than 1.00E-05.
- Solver resource time ratios for a particular model are listed only if one solver has resource greater than 5.00E-02.

Solver A: feas -- Solver B: feas Back to top

Modelname	Solver A	Solver B	Ratio (Solver A/Solver B)	Obj (Solver A)	Obj (Solver B)
alan	0.0973	0.0100	9.730	3.60000000	**2.92500000**
batch	0.2478	0.5100	0.486	285506.50824405	285506.50000000
batchdes	0.1094	0.0400	2.735	167427.65711470	167427.78888000
du-opt	1.9718	0.5200	3.792	31.02527833	**3.55634000**
du-opt5	2.0975	1.7000	1.234	40.77273140	**8.07365800**
eg_all_s	28.3584	19.7400	1.437	11.23946680	**7.92018200**
eg_disc2_s	63.1667	5.3400	11.829	6.92006923	**5.64210100**
eg_disc_s	88.8061	9.3800	9.468	10.42127936	**5.76054000**
eg_int_s	106.3869	7.7900	13.657	7.88724302	**7.46308000**
elf	0.0573	15.3200	0.004	1.67500000	**0.19166670**
ex1221	0.0270	0.0000	---	7.66718007	7.66718000
ex1222	0.0629	99999.0000	0.000	1.07654308	1.07654300

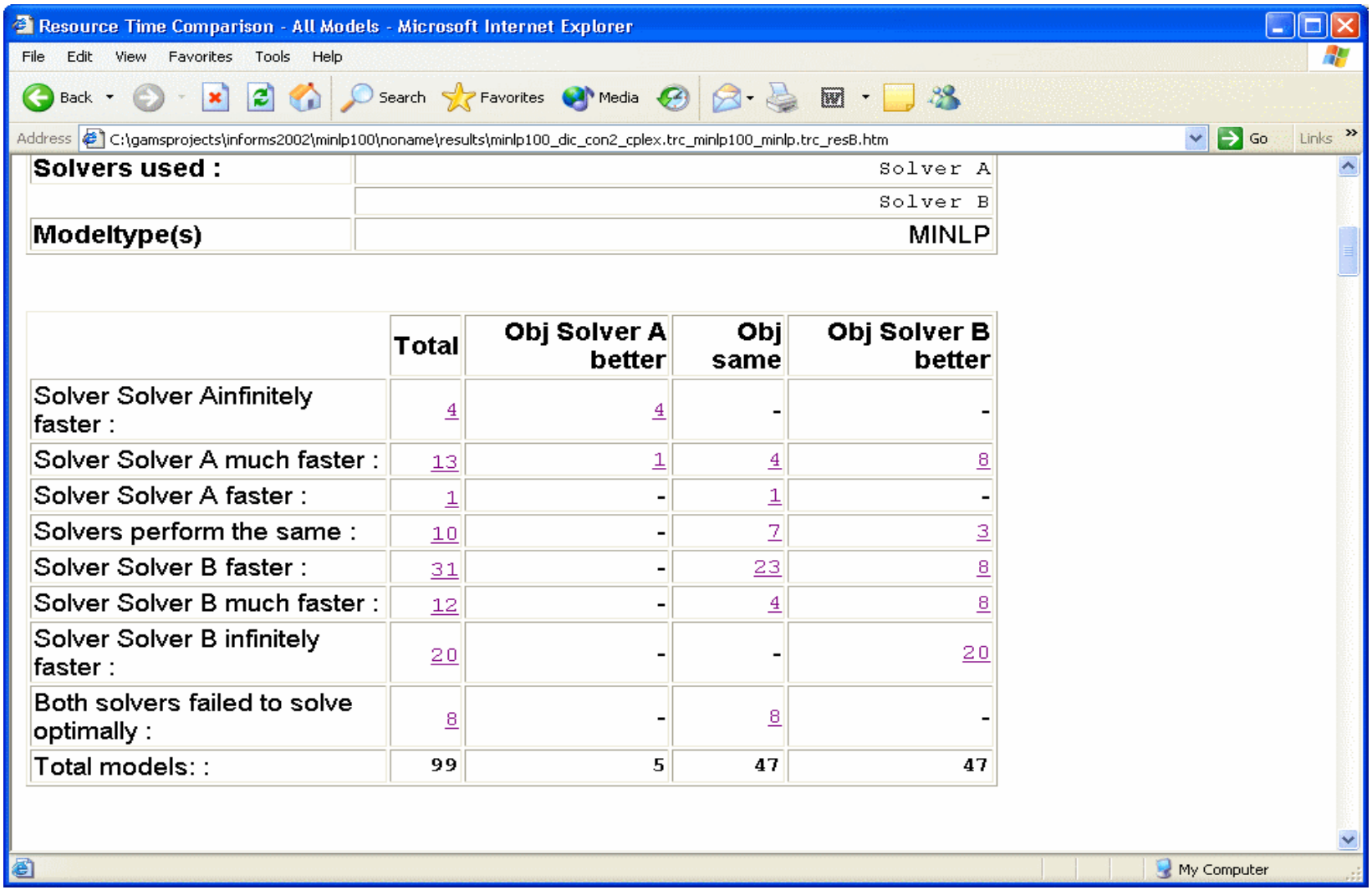
Tools: Efficiency

Resource Time Utility:

- Cross comparison of solver resource times of two solvers
- Ratios of resource times

→ Can use online using PAVER

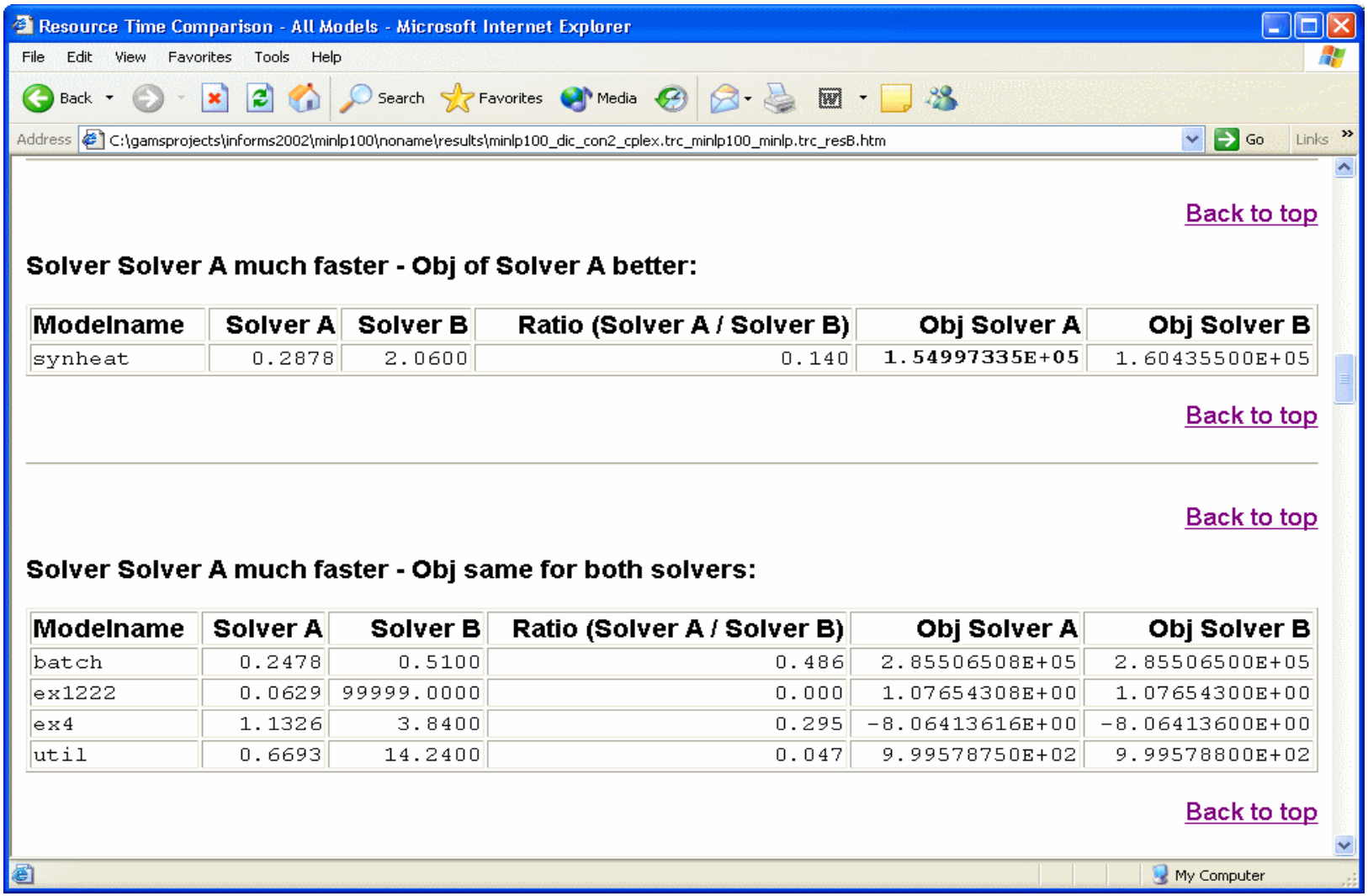
PAVER: Solver Resource Time



Solvers used :	Solver A
	Solver B
Modeltype(s)	MINLP

	Total	Obj Solver A better	Obj same	Obj Solver B better
Solver Solver Ainfinitely faster :	4	4	-	-
Solver Solver A much faster :	13	1	4	8
Solver Solver A faster :	1	-	1	-
Solvers perform the same :	10	-	7	3
Solver Solver B faster :	31	-	23	8
Solver Solver B much faster :	12	-	4	8
Solver Solver B infinitely faster :	20	-	-	20
Both solvers failed to solve optimally :	8	-	8	-
Total models: :	99	5	47	47

PAVER: Resource Time (cont.)

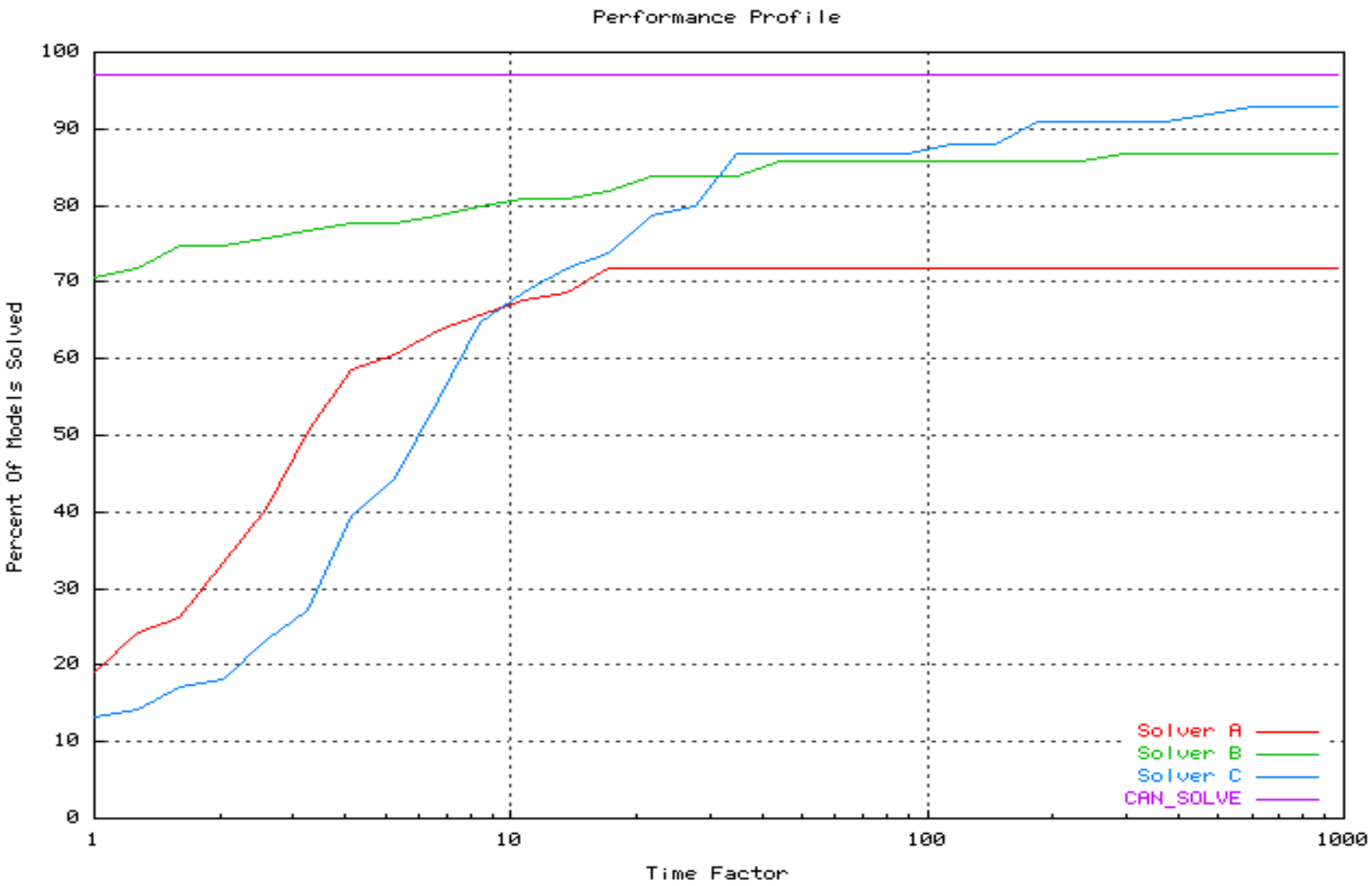


Tools: Visualization

Performance Profiles (Dolan and More, 2002):

- Cumulative distribution function for a performance metric
- Performance metric: ratio of current solver time over best time of all solvers
- Intuitively: probability of success if given τ times fastest time (τ=ratio)

Profiles (best resource time)



Agenda

- Benchmarking & QA tools
- **GAMS/COIN-OR Solvers**
- GAMS/Examiner
- Computational Results & Summary

GAMS/COIN-OR Timeline

- 2004 - Bussieck links to COIN-OR via OSI, bridge library
- 2004 (Dist 21.4) – ship GLPK and CBC as free solvers
- 2006 – Dirkse links to IPOPT via SMAGLIB
- Jan. ‘07 – Vigerske starts the COIN-OR/GAMSlinks project
- Feb. ‘07 (Dist 22.4) - ship GLPK, CBC, and Ipopt
- Feb. ’07 – corporate sponsorship of COIN-OR Foundation
- Jun. ‘07 (Dist 22.5) - ship GLPK, CBC, Ipopt, and Bonmin

The Evolution of GAMS/COIN-OR

- Initial links were more prototype than production
  - Find out what is needed on our side to make a link
  - Gain some experience working with open-source code
  - Enable experimentation with COIN solvers
  - Shipping solvers a useful by-product, included ***as-is***
- Expectations have grown (support, quality, performance)
  - Growing user community for these solvers
  - Sustained development at COIN-OR
  - GAMSlinks project catalyst for pushing forward
- QA study is now possible and desirable
  - No crashes and sign errors in returned solution
  - Provide some solid, believable evidence on quality
  - Stefan will fix things that aren’t right

Agenda

- Benchmarking & QA tools
- GAMS/COIN-OR Solvers
- **GAMS/Examiner**
- Computational Results & Summary

Benchmarking pitfalls

- Solvers may contain bugs – really!
  - Wrong solution returned
  - Wrong objective returned
  - False claims of feasibility/optimality
- Solvers will use different termination checks/tolerances
  - Difficult to compare “quality” of solutions
  - Common standard of comparison is lacking
- PAVER does not check validity of input data
  - Garbage in, garbage out
  - A “good” PAVER solver would return 0 immediately!

GAMS/Examiner

- Purpose: to make an unbiased, independent report on the merit of points
- Points may come from GAMS or a solver
  - GAMS passes the previous solution as initial iterate
  - Solvers pass solutions back to GAMS
- Useful during solver debugging – helps pinpoint problems
  - Most checks are obvious – almost insultingly so
  - Updated as new ways to pass bad points are discovered
- Does checks on the scaled and unscaled (orginal) model
- All solution tolerances can be adjusted, default is tight
- Different points can be checked – no obvious right choice
- Examiner only *reports*, it doesn't *fix*. It's not a healer.

Examiner: What point to examine?

- Assume the solver returns `x.l, x.m, f.l, f.m`
  - `x.l` are variable levels
  - `f.m` are equation marginals or shadow prices
  - `x.m` are the reduced costs: e.g. c – A'y for LP
  - `f.l` are row activity levels: e.g. Ax – b for LP
- GAMS/Base accepts this *solvpoint* point as given, so it should be checked
  - Check the signs (e.g. reduced costs positive)
  - Check complementary slackness
- What if the solver cheated or got it wrong for `x.m` and `f.l`?
  - Take `x.l` and `f.m` as solution, recompute `x.m` and `f.l`
  - This *solupoint* checked as above
  - Better indicator of solution quality

Examiner: scaled or unscaled?

- Typically, the solvers scale a model prior to solving
- The GAMS user does not have access to this scaled model
- Examine the unscaled (original) solution/model?
  - ☺ Easily accessed
  - ☺ The user is probably interested in this model
  - ☹ Not the model that was solved
  - ☹ Unscaling may throw solution out of tolerances
- Examine the scaled solution/model?
  - ☺ This was the model that was solved
  - ☺ Most likely to satisfy Examiner's tolerances
  - ☹ We don't know what this model is
  - ☹ The user may not be interested in this model

Examiner: scaled checks (Chinneck CC)

- Examiner computes the scaled model by doing row scaling
- The true row scale s(i) is computed as
  s(i) = max(|RHS(i)|, max {j, |A(i,j)| * max(1,|x(j)|)})
- The true row scale s(i) is projected onto the scale bounds before being applied
  - The default lower bound is 1 – rows are scaled down, not up.
- Scaling info is printed but not used for unscaled checks – it's useful to see when & where your maximum row scale is 1e13!

Examiner: tolerances

- Four tolerances used to check feasibility & optimality
  - PrimalFeasTol (1e-6): feasibility of x.l, f.l
  - PrimalCSTol (1e-7): complementary slackness, x.l x.m
  - DualFeasTol (1e-6): feasibility of f.m, x.l
  - DualCSTol (1e-7): complementary slackness, f.l f.m
- Default tolerances are tight
  - Favor false negatives over false positives
  - If it passes on defaults, you're typically satisfied
  - If Examiner reports problems, you have a chance to think about the right tolerance for your model

Agenda

- Benchmarking & QA tools
- GAMS/COIN-OR Solvers
- GAMS/Examiner
- Computational Results & Summary

Computational Details

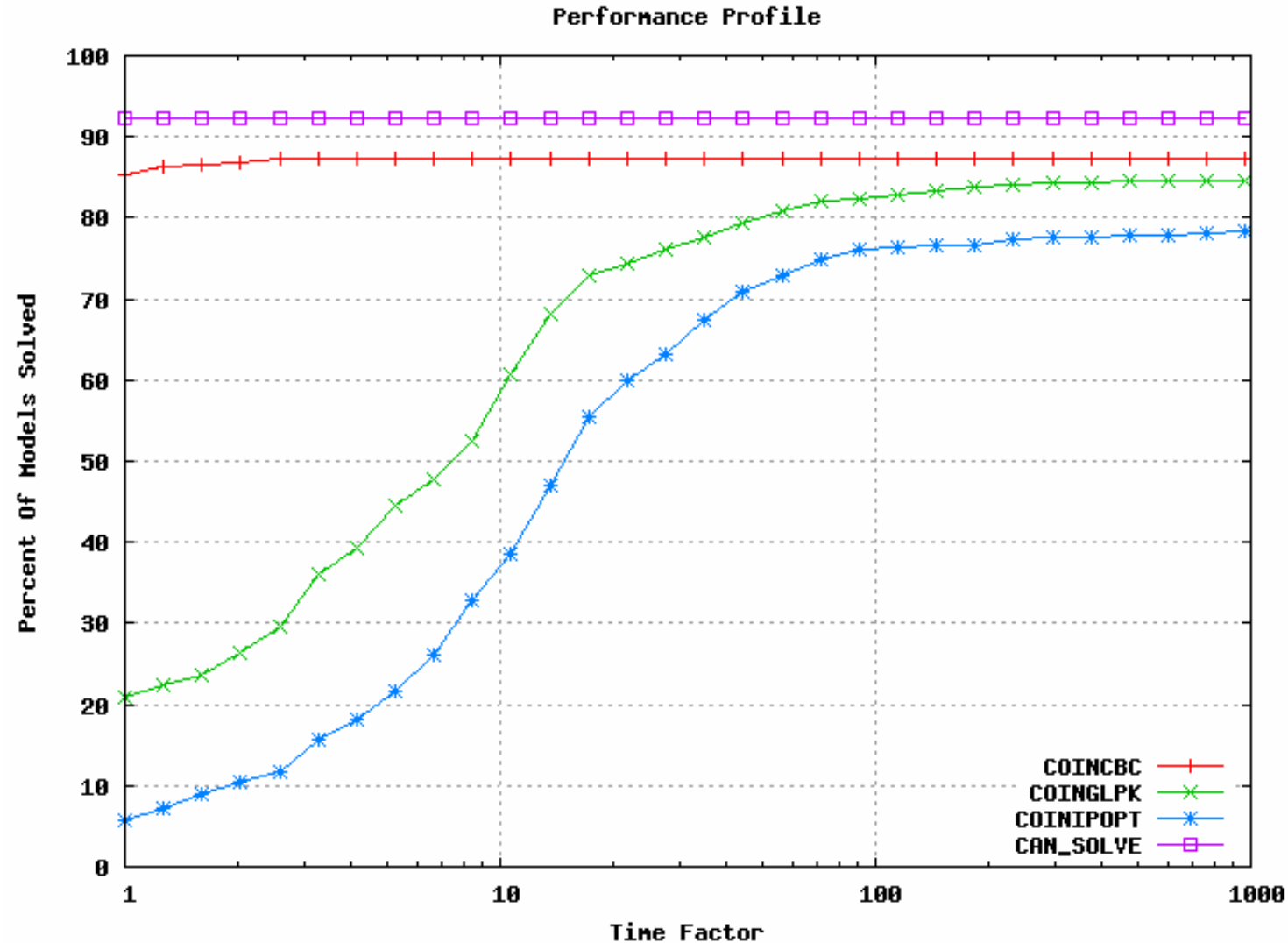
- Test Platform – my newish Laptop
  - Lenovo Thinkpad T60, Windows Vista Business 32-bit
  - 2GB RAM, CoreDuo CPU T7200 @2GHz
- Software Version: GAMS 22.5+
  - Updated GAMS/Examiner, otherwise used 22.5 as-is
  - COIN/CBC uses 1.2pre libraries
  - COIN/GLPK uses 4.15 libraries
  - COIN/IPOPT uses 3.3pre
- No need to use schultz, all solvers stopped gracefully

LINLib – LP Test

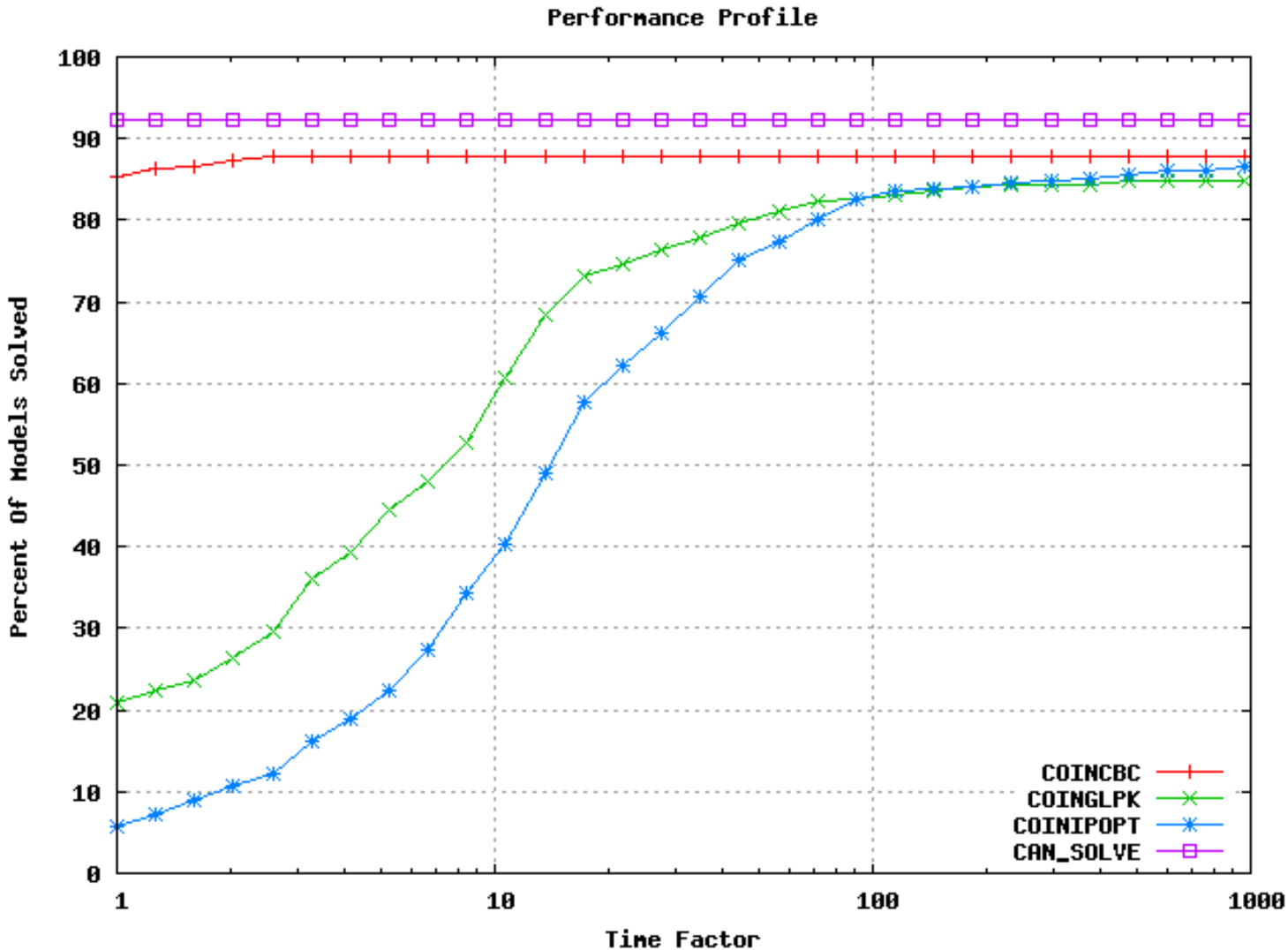
- Test on LP models from LINLib
  - Remove 6 models that GAMS/Base called infeasible
  - Remaining LP models for test: 520
- Time limit set to 2 hrs
- Iteration limit set very high (not active)
- Solvers tested: CBC, GLPK, IPOPT
- see PAVER output for complete results

	Optimal	Feasible	Infeas	Unbnd	Fail
CBC	456		40		24
GLPK	444		40	1	35
IPOPT	445	11	35		29

LINLib – LP Perf Profile (1e-5 obj tol)



LINLib – LP Perf Profile (Infinite obj tol)

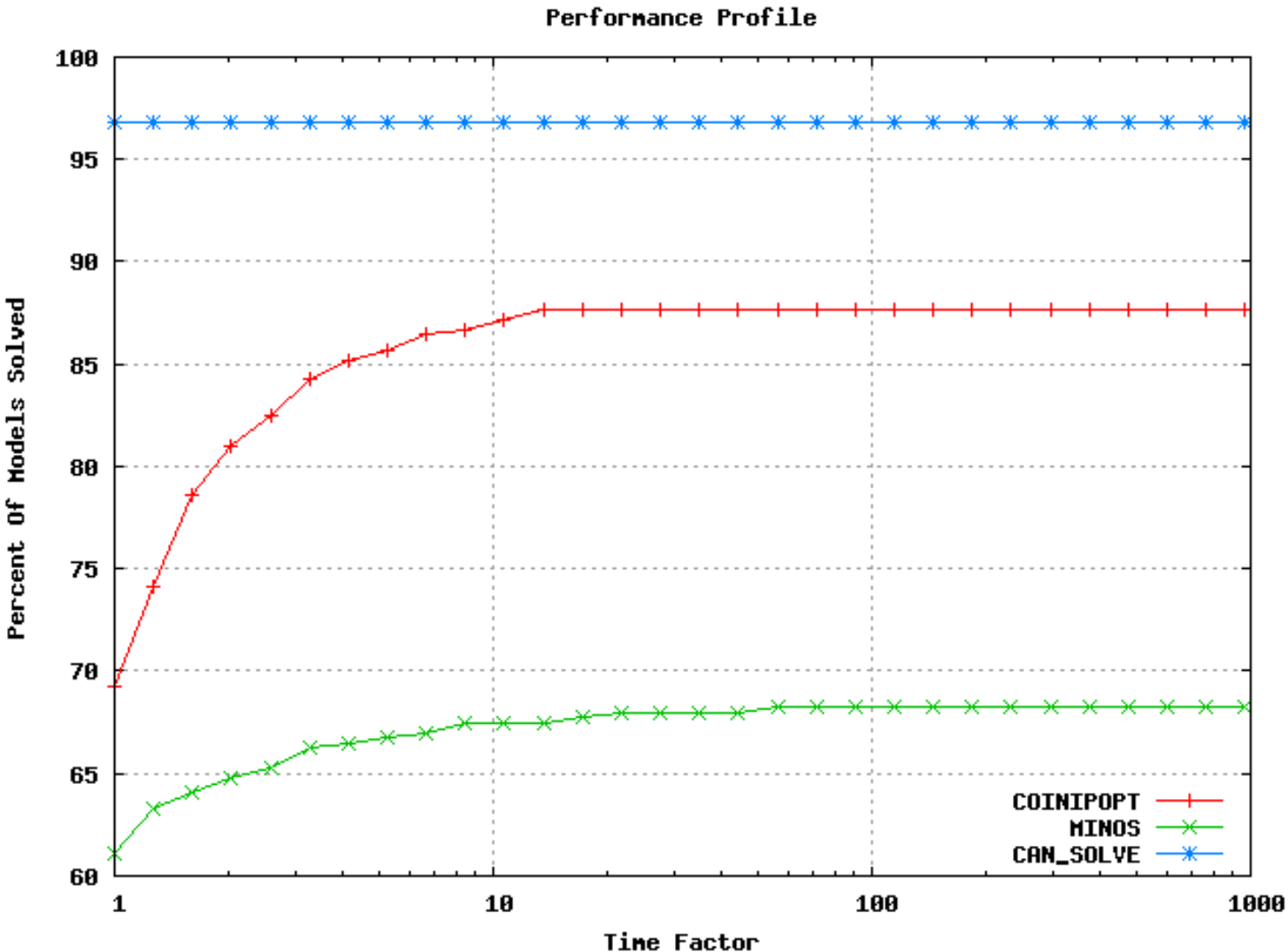


GLOBALLib – NLP Test

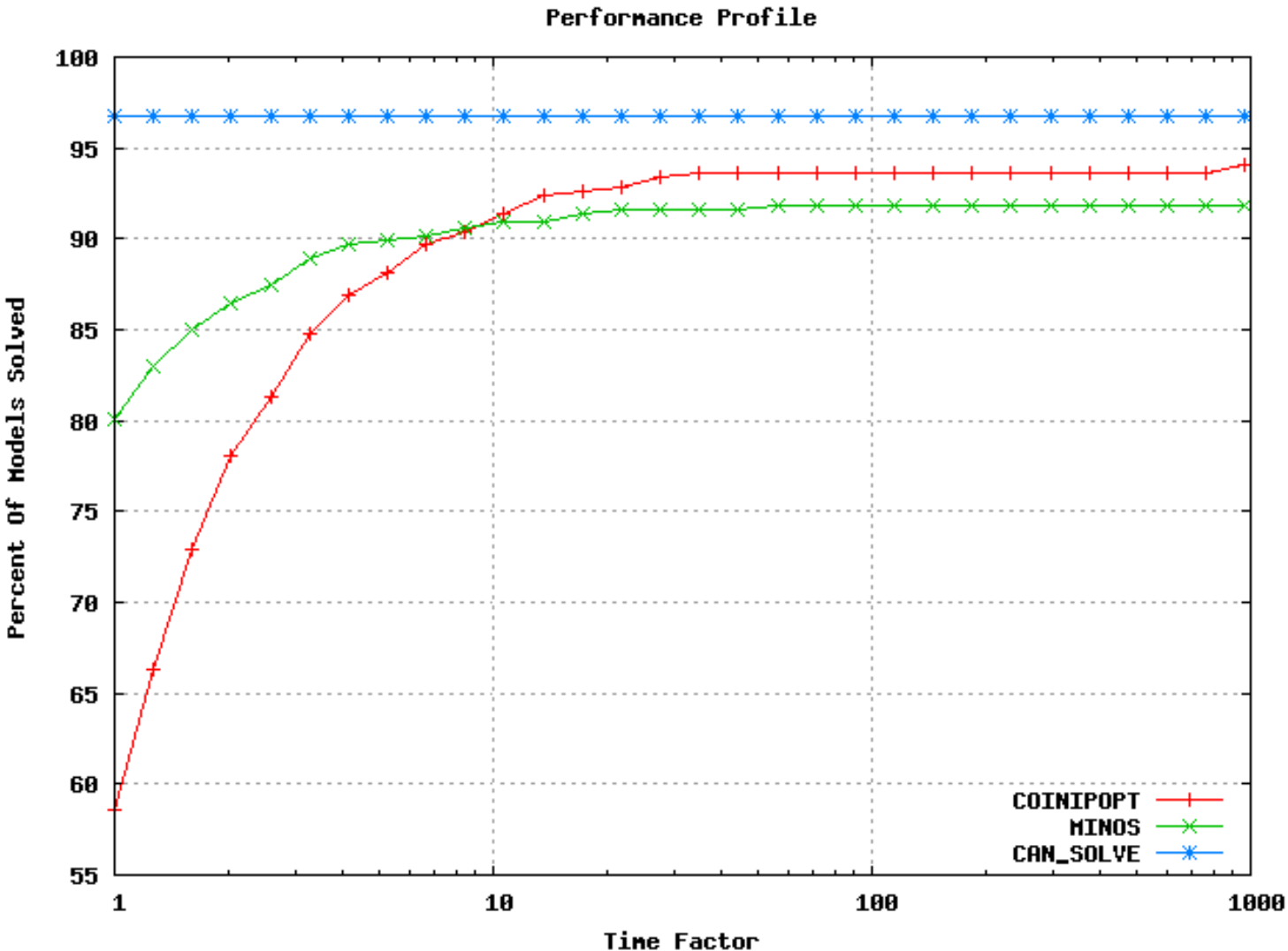
- Test on all 406 NLP models from GLOBALLib
- Time limit set to 1 hr
- Iteration limit set very high (not active)
- Solvers tested: IPOPT, MINOS
- see PAVER output for complete results

	Optimal	Feasible	Infeas	Unbnd	Fail
IPOPT	378	4	7	3	14
MINOS	337	36	30	3	

GLOBALLib – NLP Perf Profile (1e-5 obj tol)



GLOBALLib – NLP Perf Profile (Infinite obj tol)

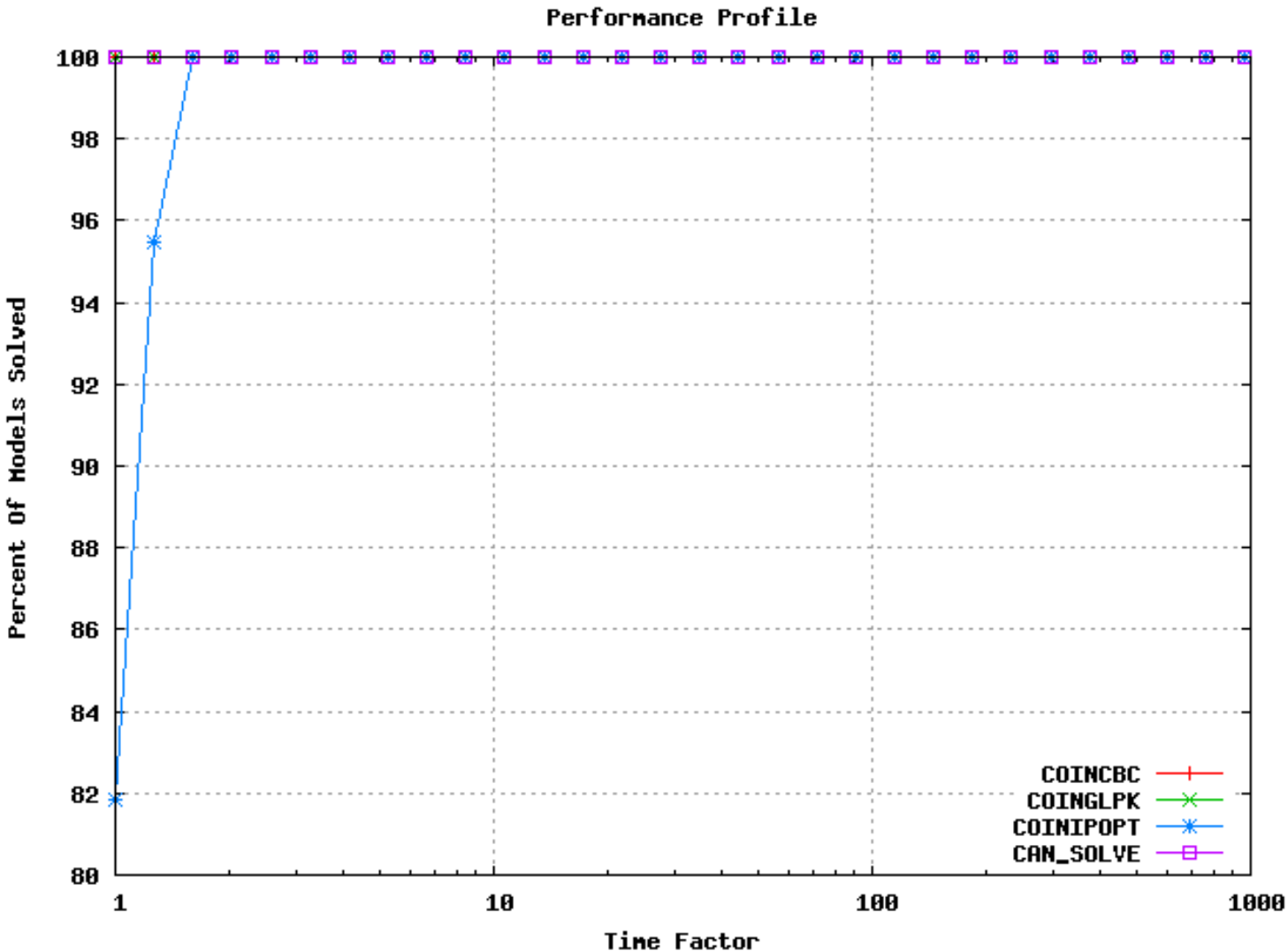


MPLLib – LP Test

- Test on 22 LP models from MPLLib
- Solvers tested: CBC, GLPK, IPOPT
- All solvers solve all models to Examiner tolerances
- see PAVER output for complete results

	Optimal	Feasible	Infeas	Unbnd	Fail
CBC	22				
GLPK	22				
IPOPT	22				

MPLLib – LP Perf Profile (1e-5 obj tol)



Summary

- COIN solvers are quite robust overall
- CBC and IPOPT are looking very good
- *Results will go to COIN developers*
- Performlib, PAVER are very useful
  - PAVER could fit well at COIN
  - Developers can do their own studies
  - Is LINLib too hard a test set?
- Examiner-ish tool to COIN as well?
- “If you build it, they will come”