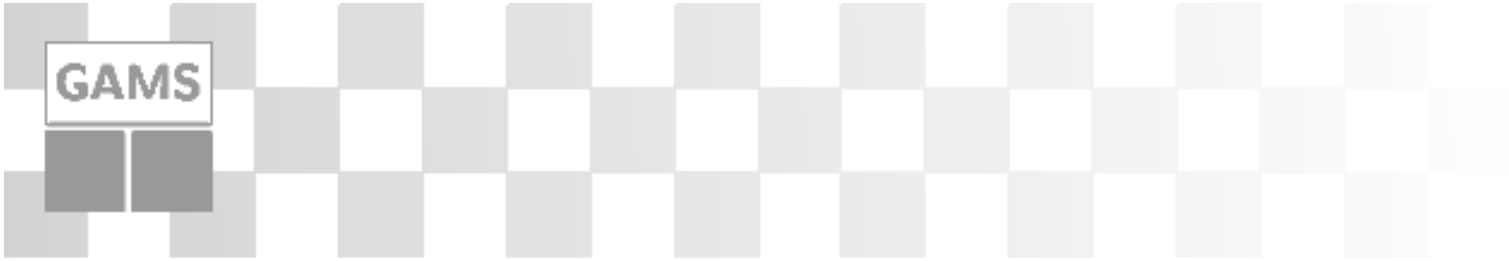


Open-source Quality Assurance and Performance Analysis Tools

Armin Pruessner, Michael Bussieck,
Steven Dirkse, Stefan Vigerske

GAMS Development Corporation
1217 Potomac Street NW
Washington, DC 20007

Informs Annual Meeting 2008
Washington DC

Agenda

- Motivation
- Definitions and Components
- Challenges
- Software Quality Assurance at GAMS
- Testing new Solver Links
- Performance Analysis
- Summary

Motivation

Quality Assurance

- Essential component in most industries
- Important in most software engineering sectors

Mathematical Programming

- Less attention to quality assurance (small community)
- Specific QA issues for modeling systems (initially expensive)
- Different focus for industry and academic

Focus on reliability

Focus on performance

Definitions

Quality: *The totality of features and characteristics of a product or service that bear on its ability to satisfy specified or implied needs* (ISO 8402)

Software Quality Assurance (SQA): "*Set of systematic activities providing evidence of the ability of the software process to produce a software product that is fit to use*" (Schulmeyer and McManus)

Definitions
cont'd

Key components of SQA (which includes monitoring of *products* and *processes)*:

- Software configuration management (SCM): All activities related to version control and change control
- Quality control and testing: monitoring the products
  - Focus on the quality of product ***within each phase*** of the software development lifecycle
  - Objective: identify and remove defects throughout the lifecycle, as ***early as possible***

Components

Software configuration management: All activities related to version control and change control

Goals:

- Identify and control changes
- Ensure that change is being properly implemented
- Report changes in the software to others who may need to know of them

Components

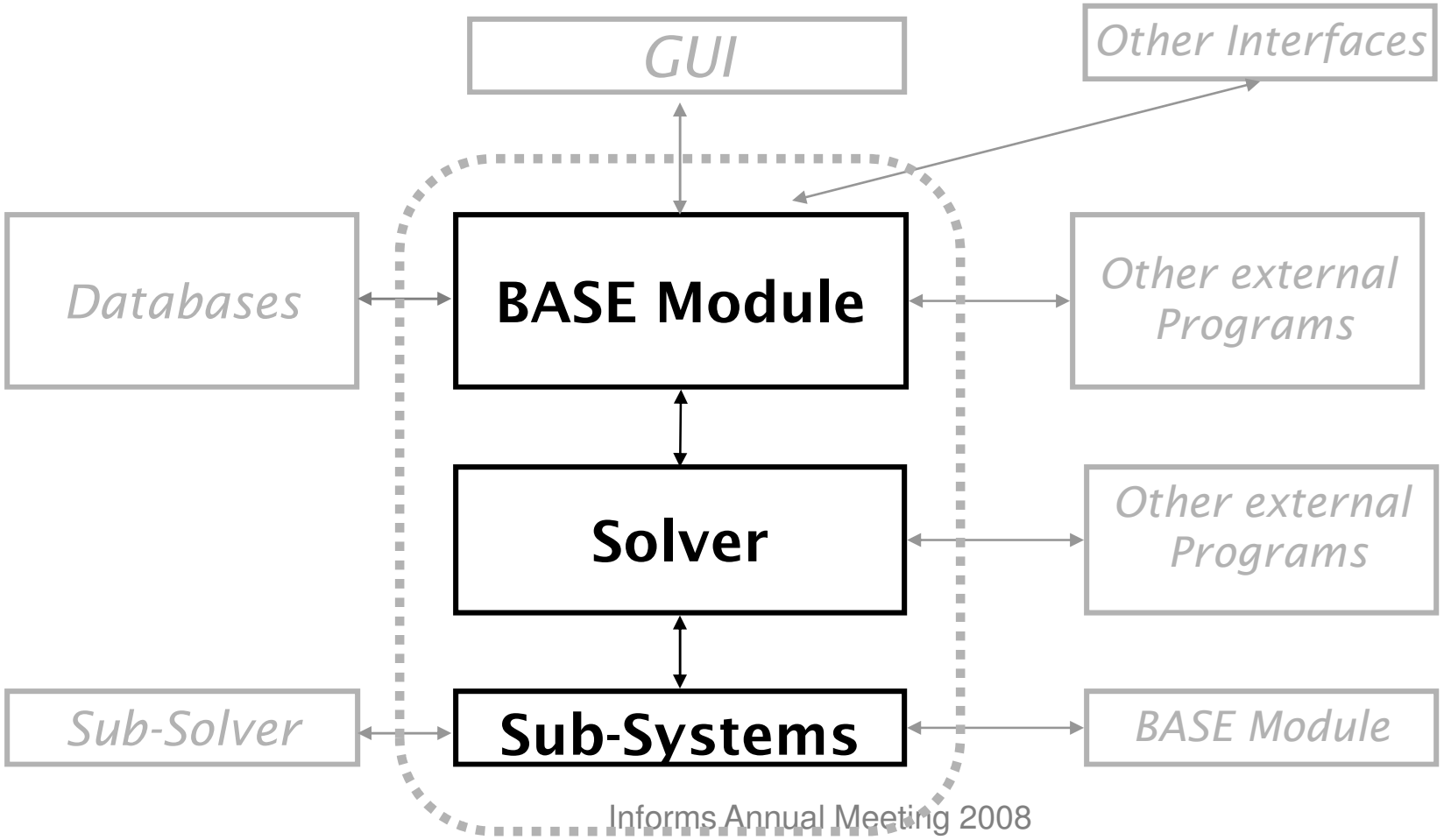
Quality control and testing of the product

- Unit testing (initial tests by developer)
- Regression testing (interaction with other modules)
- System Integration Testing (full scale testing)
- Metrics
- Bug tracking tools

Goal: Uncover defects during the complete life cycle of the software as *early as possible*

Algebraic Modeling Systems

Architecture



Algebraic Modeling Systems

Basic Principles

- Separation of model and solution methods
- Balanced mix of declarative and procedural approaches
- Computing platform independence
- Multiple model types, solvers and platforms

Algebraic Modeling Systems

Multiple Model Types

- LP Linear Programming
- MIP Mixed Integer Programming
- NLP Nonlinear Programming
- QCP/MIQCP Quadratic Programming
- Conic Programming
- MCP Mixed Complementarity Programming
- MINLP Mixed Integer Nonlinear Programming
- MPEC NLP with Complementarity Constraints
- MPSGE General Equilibrium Models
- Stochastic Optimization
- Global Optimization

GAMS

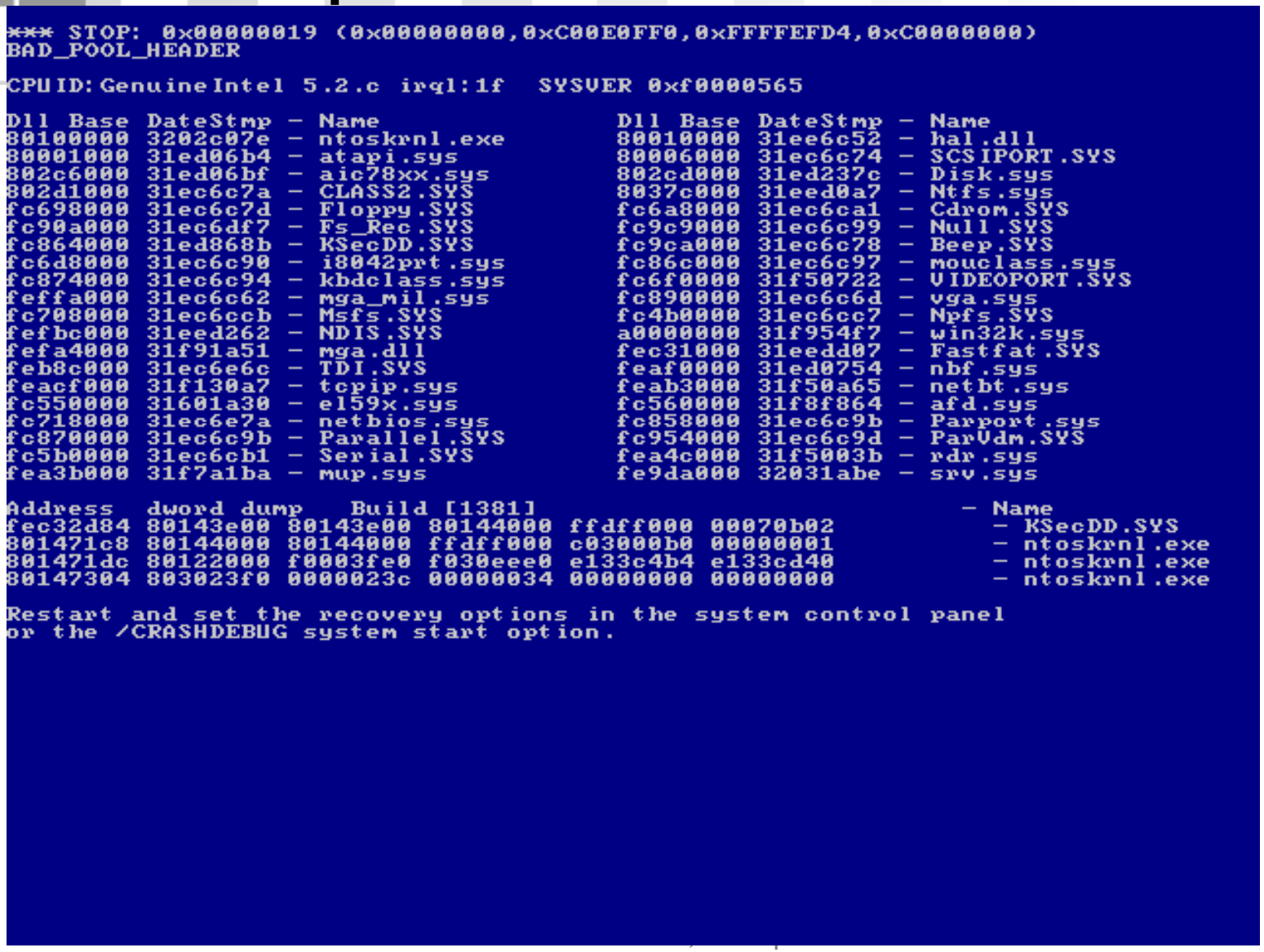
Challenges

- Expense: Rigorous SQA is initially expensive
  - MP industry is small
- Limited control about certain parts of the system
  - solvers are black box modules
- Distributed development of the system and various components

Challenges: Chance of Failure

QA issues specific to Mathematical Programming Software:

Implementation Defect



Blue Screen of Death in Microsoft Windows NT

Not acceptable

Solver Failure

Solver does not find a solution

Acceptable

```
               S O L V E      S U M M A R Y

     MODEL   one                    OBJECTIVE  output
     TYPE    NLP                    DIRECTION  MAXIMIZE

**** SOLVER STATUS     4 TERMINATED BY SOLVER
**** MODEL STATUS      6 INTERMEDIATE INFEASIBLE
**** OBJECTIVE VALUE              948403.4844

 RESOURCE USAGE, LIMIT          1.906      1000.000
 ITERATION COUNT, LIMIT         0         10000
 EVALUATION ERRORS              0             0
```

Challenges: Chance of Failure

QA issues specific to Modeling Systems:

- Additionally, must protect the user in case of solver failure
- Complex metric for return codes is necessary
- Complicates QA activities since this adds an additional level of complexity

SQA at GAMS

1. *Software configuration management*
2. **Quality control and tests of the product**
3. *Client model testing*
4. **Performance comparison tools**
5. *Solution verification tool: Examiner*
6. *Model converter and “encryption“ tool: Convert*

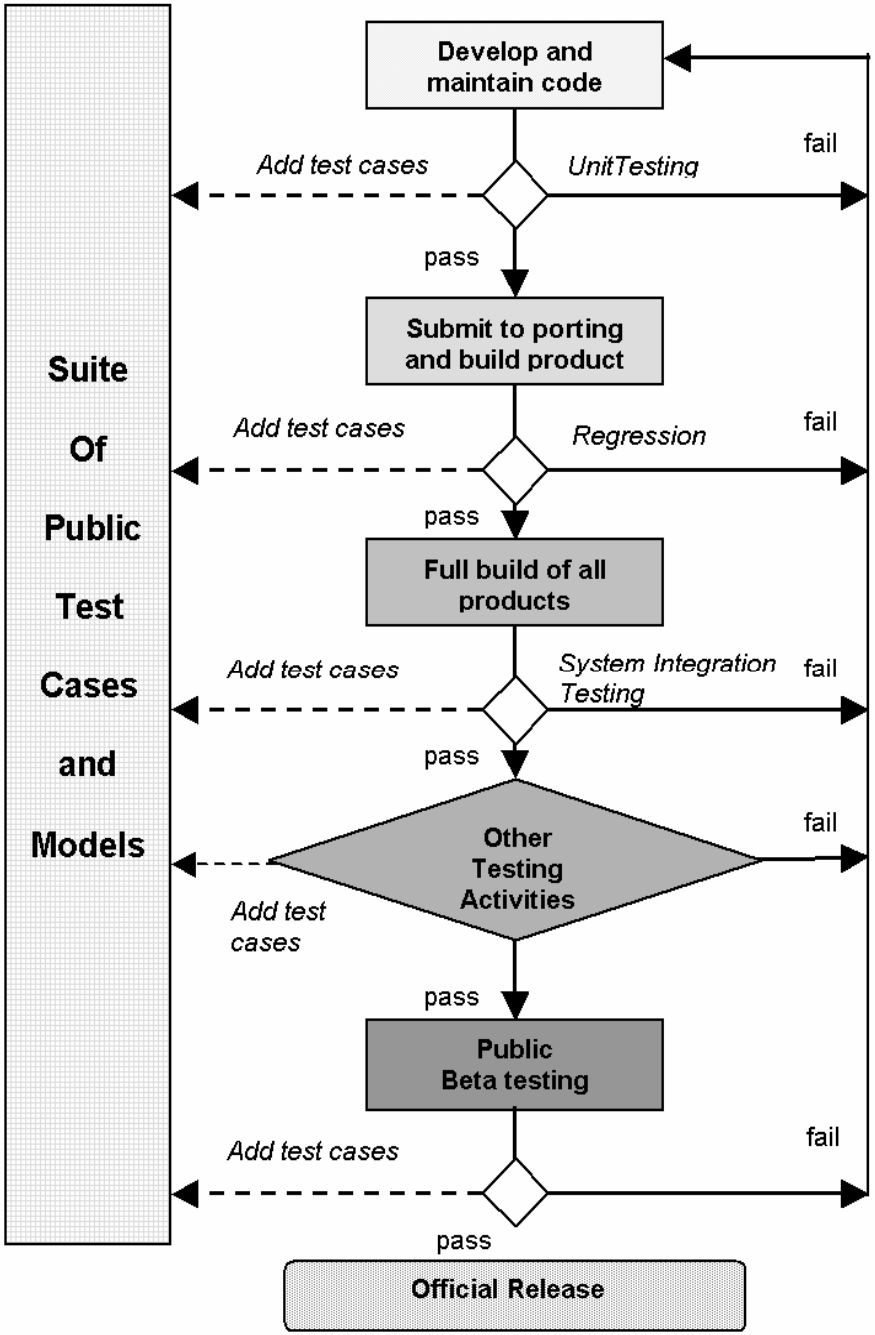
SQA at GAMS (Testing)

Quality control and tests of the product

- Goal: Continuous quality improvement using automated and reproducible tests
- Test libraries (available online):
  - GAMS Model Library (solver tests)
  - GAMS Quality Test Models Library (modeling system test)

- Continuous addition of new test models

General testing process



Continuous addition of new test models throughout life cycle

SQA at GAMS

Quality Test Models Library

- Include tests to verify proper behavior of the system
- More than 400 quality test models, each containing numerous pass/fail tests:

```
...
abort$card(delta) 'time routines have an error';
...
```

- Automatic generated test summaries with different level of information

SQA at GAMS

Summary of two quality runs

```
*** Status: Normal completion
--- quality.gms(284) 4 Mb
--- quality.gms(287) 4 Mb 1 Error
There were errors: 4 out of 408 tests failed.
See the file failures.gms to reproduce the failed runs
--- Putfile this D:\support\testlib\onetest.gms
--- quality.gms(287) 4 Mb 1 Error
*** Status: Execution error(s)

==============================================

*** Status: Normal completion
--- quality.gms(284) 4 Mb
--- quality.gms(295) 4 Mb
Congratulations!  All 408 tests passed.
See the file alltests.gms to reproduce all the runs
--- Putfile this D:\support\testlib\onetest.gms
*** Status: Normal completion
```

Testing New Solver Links

Solver developer has connected his solver to GAMS (e.g. a COIN solver)

- Automated tests to check basic functionality of the solver and the link to GAMS:

```
$title Simple level and sign test (LP02,SEQ=67)
* In this test series we status if a solver gets the levels
* and marginals right.
..
abort$( abs(cost.m-cost_m) > tol) 'bad cost.m';
```

- Gives developer and users assurance about the basic functionality of the link and the solver

Performance World

Performance World

Editorial Board

PerformanceLib

Performance Tools

Performance List

Related Links

Search

Contact

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

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

Motivation for Tools

Performance Tools driven by user needs:

- Finding the most reliable way to solve a proprietary model
- Testing a new algorithm against a set of existing test problems and competing approaches
- Reproducibility of performance results

Performance World

Default Solver Options

Performance Analysis & Visualization

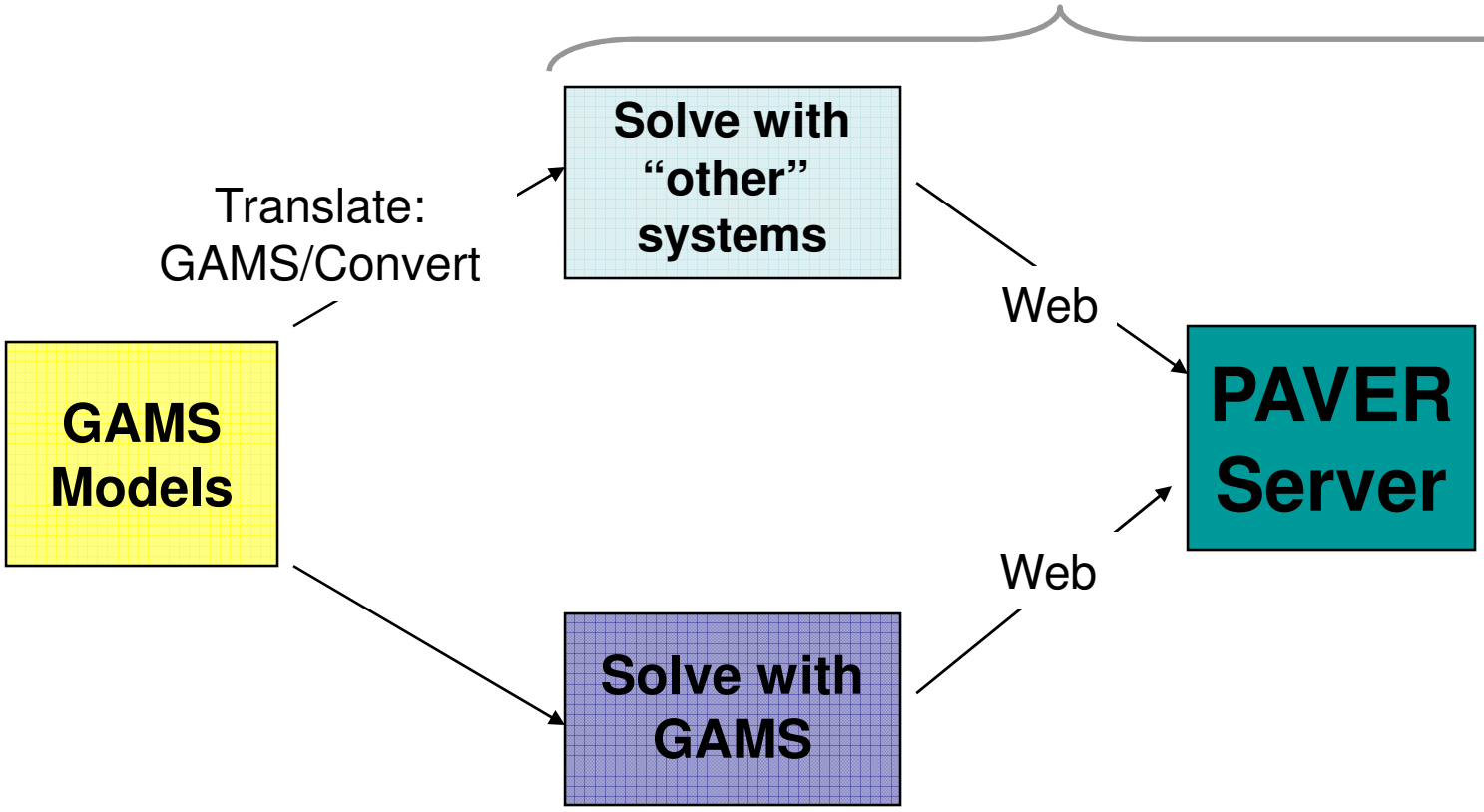
Custom Solver Options

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

Performance Framework

Can use Performance World tools



PAVER Server

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

  www.gamsworld.org/performance/paver

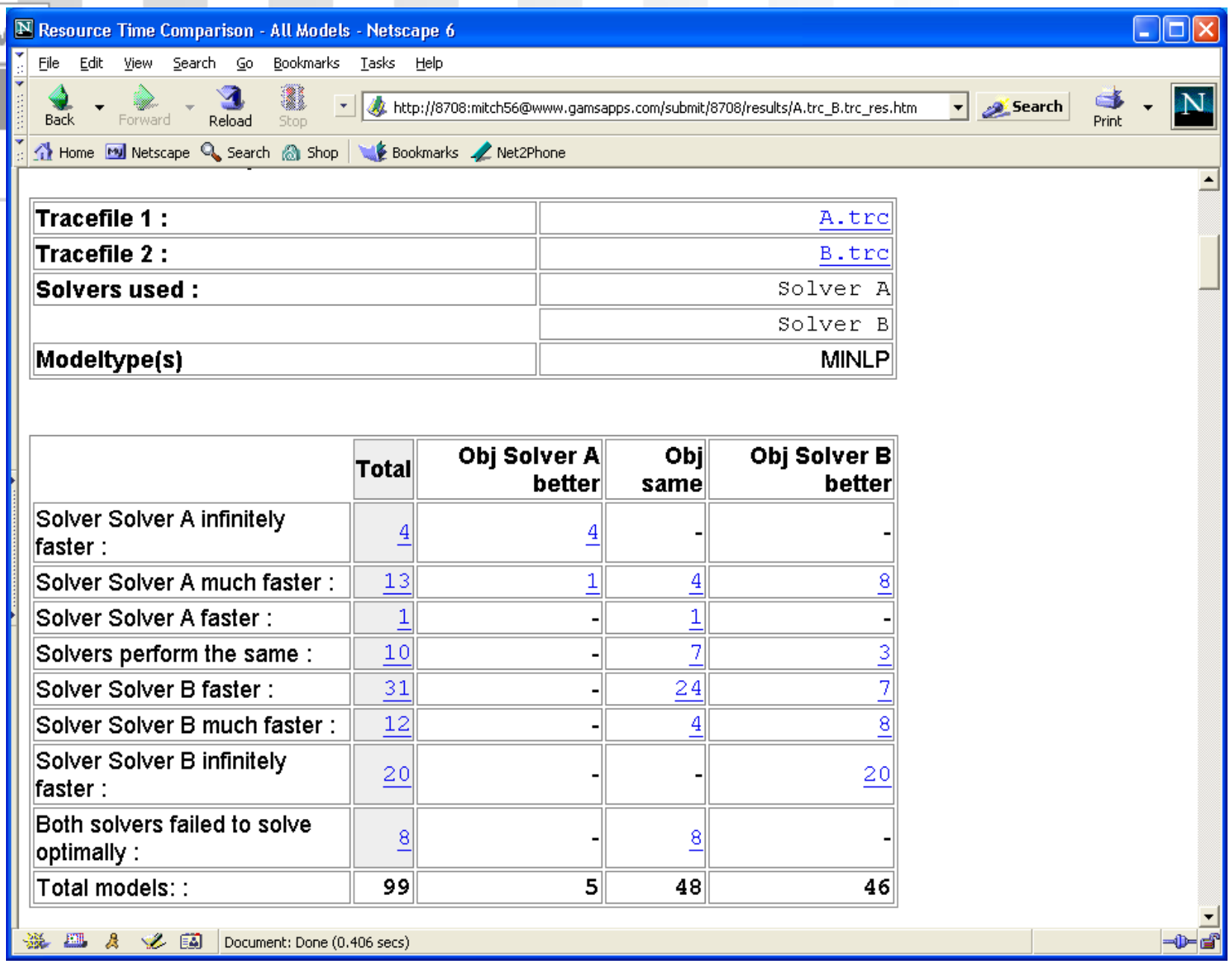
- Online server to facilitate performance analysis/visualization of data
- Results sent via e-mail in HTML format
- Rely on 3 tools: resource time, solver square, performance profiles

Tools: Robustness

Solver Square Utility:

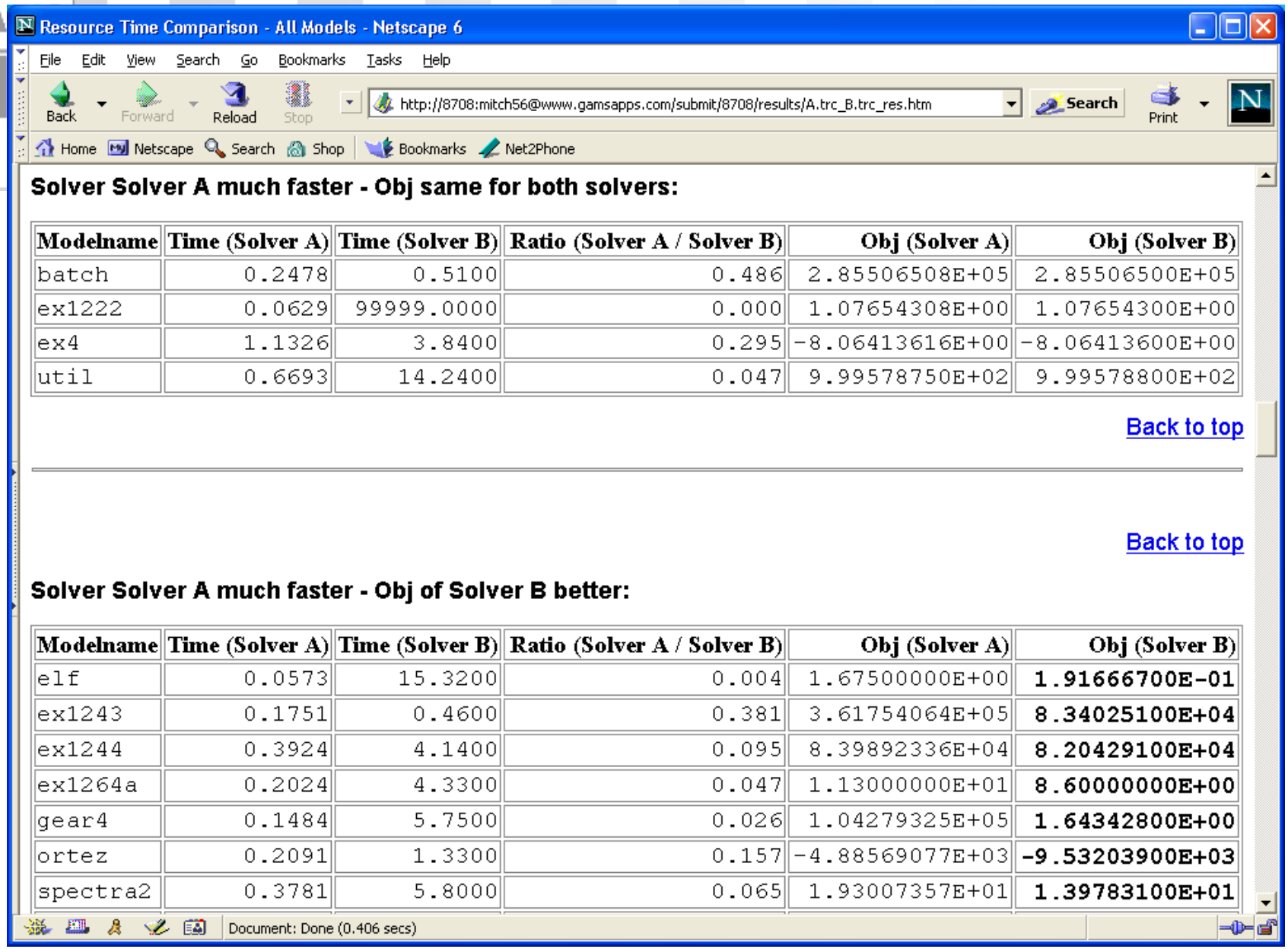
- Cross comparison of solver outcomes of two solvers:
  - Optimal, feasible, unbounded, infeasible, fail
- Compact tabular form for results
- Shows resource time and objective value information

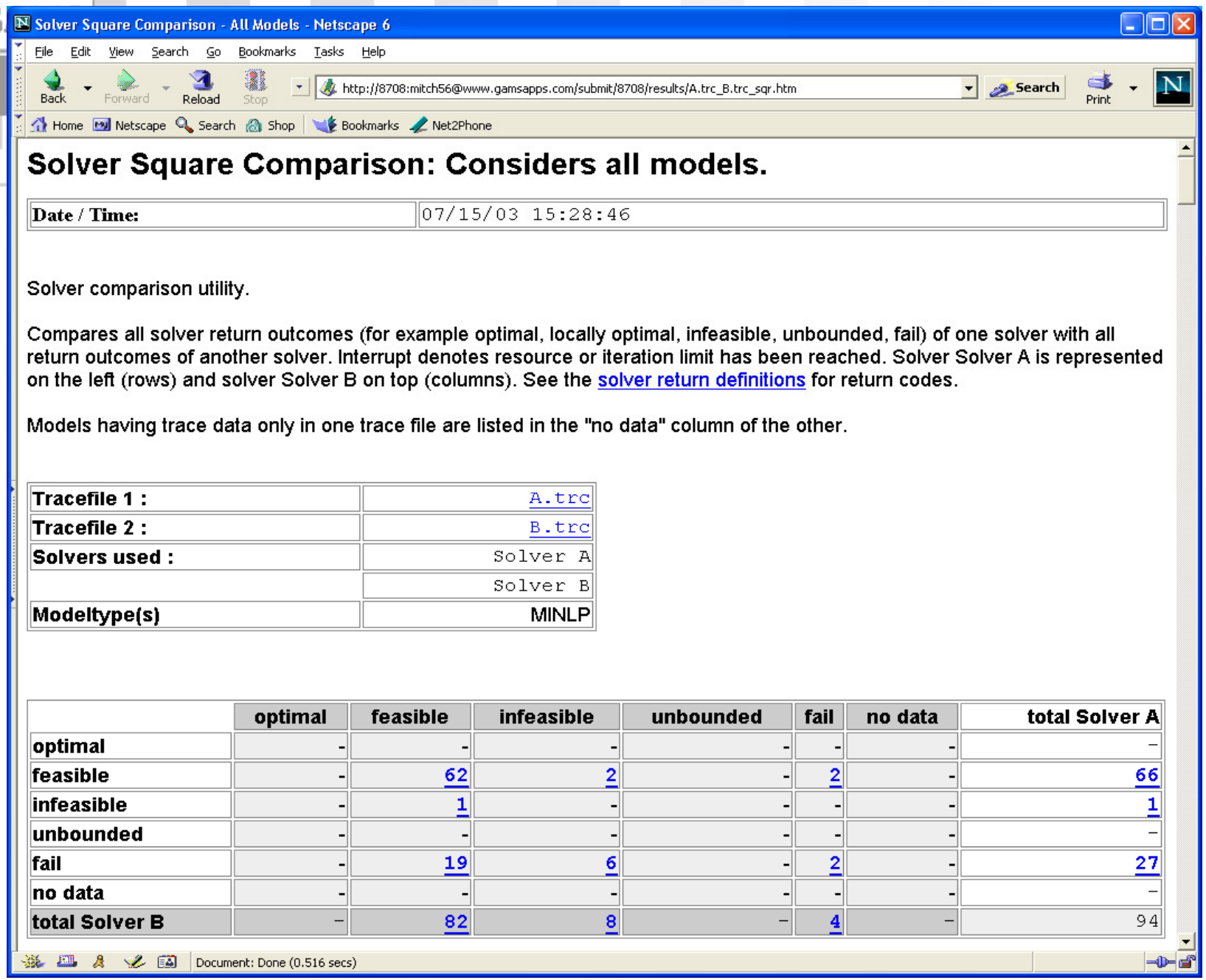
→ Can use online using PAVER



Tracefile 1 :	A.trc
Tracefile 2 :	B.trc
Solvers used :	Solver A
	Solver B
Modeltype(s)	MINLP

	Total	Obj Solver A better	Obj same	Obj Solver B better
Solver Solver A infinitely 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	-	24	7
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	48	46


Resource Time Comparison - All Models - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
Back Forward Reload Stop
http://8708:mitch56@www.gamsapps.com/submit/8708/results/A.trc_B.trc_res.htm
Search Print
Home Netscape Search Shop Bookmarks Net2Phone
Solver Solver A much faster - Obj same for both solvers:
Modelname | Time (Solver A) | Time (Solver B) | Ratio (Solver A / Solver B) | Obj (Solver A) | Obj (Solver B)
batch | 0.2478 | 0.5100 | 0.486 | 2.85506508E+05 | 2.85506500E+05
ex1222 | 0.0629 | 99999.0000 | 0.000 | 1.07654308E+00 | 1.07654300E+00
ex4 | 1.1326 | 3.8400 | 0.295 | -8.06413616E+00 | -8.06413600E+00
util | 0.6693 | 14.2400 | 0.047 | 9.99578750E+02 | 9.99578800E+02
Back to top
Back to top
Solver Solver A much faster - Obj of Solver B better:
Modelname | Time (Solver A) | Time (Solver B) | Ratio (Solver A / Solver B) | Obj (Solver A) | Obj (Solver B)
elf | 0.0573 | 15.3200 | 0.004 | 1.67500000E+00 | 1.91666700E-01
ex1243 | 0.1751 | 0.4600 | 0.381 | 3.61754064E+05 | 8.34025100E+04
ex1244 | 0.3924 | 4.1400 | 0.095 | 8.39892336E+04 | 8.20429100E+04
ex1264a | 0.2024 | 4.3300 | 0.047 | 1.13000000E+01 | 8.60000000E+00
gear4 | 0.1484 | 5.7500 | 0.026 | 1.04279325E+05 | 1.64342800E+00
ortez | 0.2091 | 1.3300 | 0.157 | -4.88569077E+03 | -9.53203900E+03
spectra2 | 0.3781 | 5.8000 | 0.065 | 1.93007357E+01 | 1.39783100E+01
Document: Done (0.406 secs)



Solver Square Comparison: Considers all models.

Date / Time:	07/15/03 15:28:46

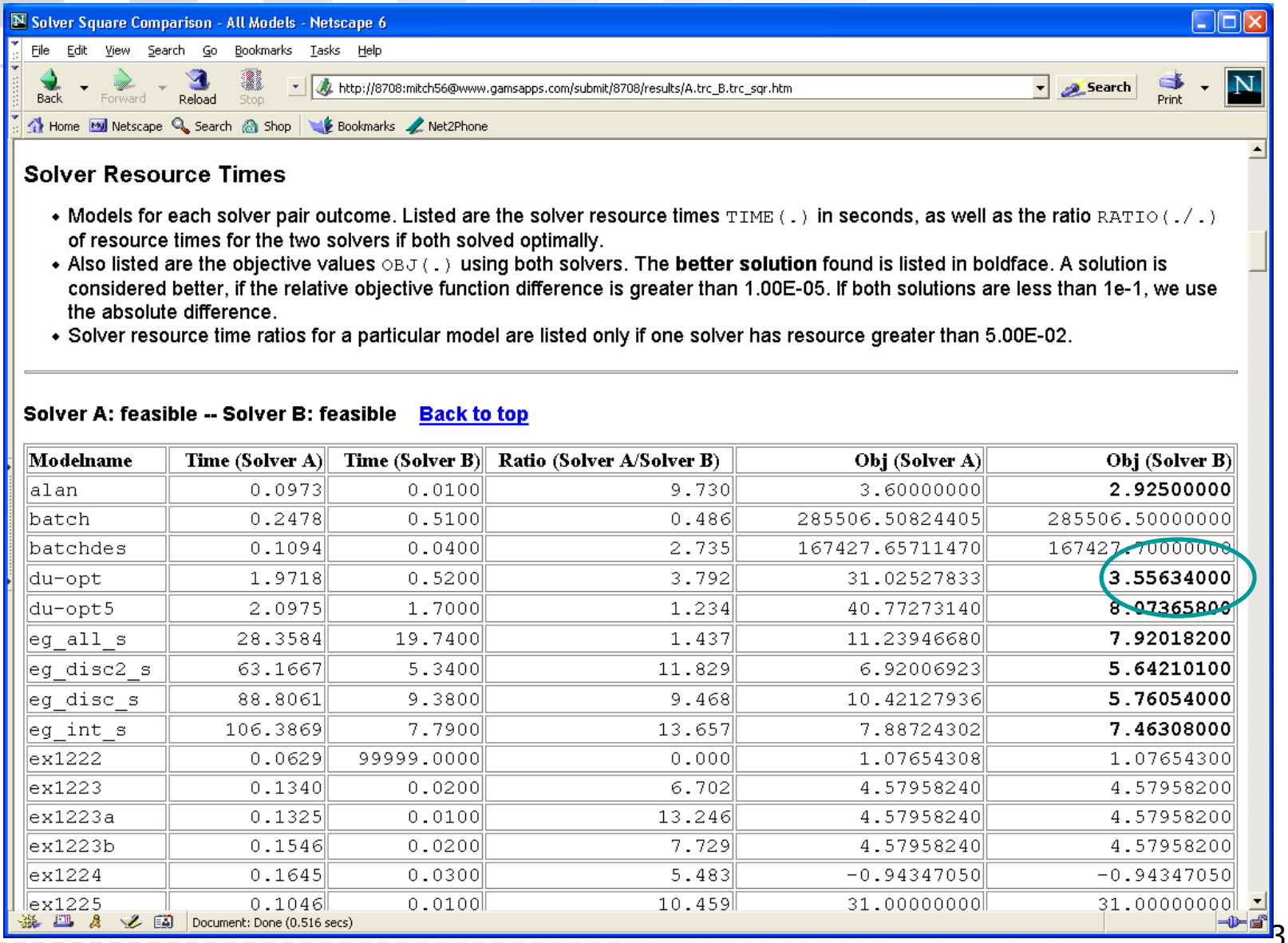
Solver comparison utility.

Compares all solver return outcomes (for example optimal, locally optimal, infeasible, unbounded, fail) of one solver with all return outcomes of another solver. Interrupt denotes resource or iteration limit has been reached. Solver Solver A is represented on the left (rows) and solver Solver B on top (columns). See the solver return definitions for return codes.

Models having trace data only in one trace file are listed in the "no data" column of the other.

Tracefile 1 :	A.trc
Tracefile 2 :	B.trc
Solvers used :	Solver A
	Solver B
Modeltype(s)	MINLP

	optimal	feasible	infeasible	unbounded	fail	no data	total Solver A
optimal	-	-	-	-	-	-	-
feasible	-	62	2	-	2	-	66
infeasible	-	1	-	-	-	-	1
unbounded	-	-	-	-	-	-	-
fail	-	19	6	-	2	-	27
no data	-	-	-	-	-	-	-
total Solver B	-	82	8	-	4	-	94

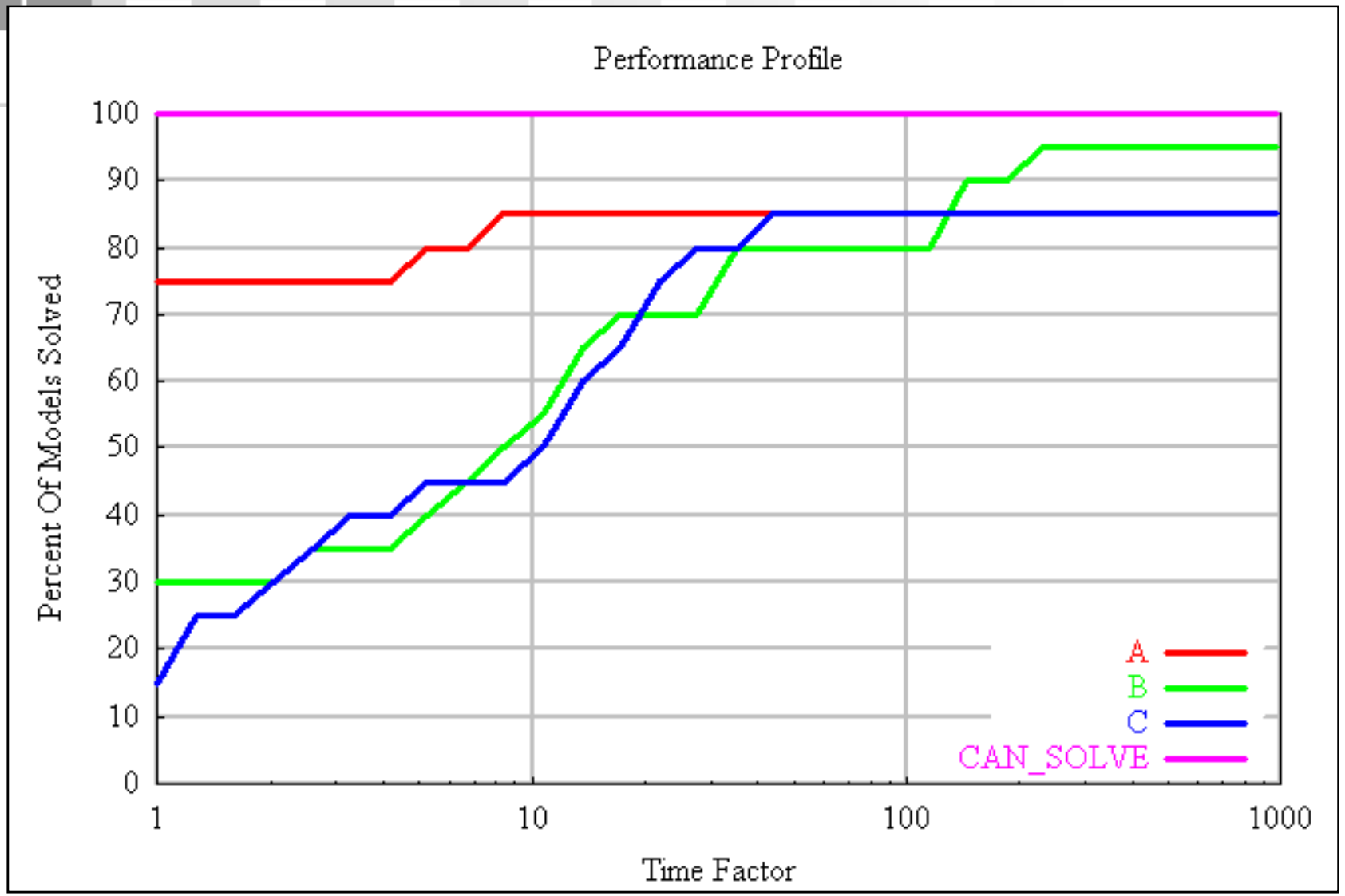


Solver Resource Times

- Models for each solver pair outcome. Listed are the solver resource times `TIME(.)` in seconds, as well as the ratio `RATIO(./.)` of resource times for the two solvers if both solved optimally.
- Also listed are the objective values `OBJ(.)` 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. If both solutions are less than 1e-1, we use the absolute difference.
- Solver resource time ratios for a particular model are listed only if one solver has resource greater than 5.00E-02.

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

Modelname	Time (Solver A)	Time (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	167427.70000000
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**
ex1222	0.0629	99999.0000	0.000	1.07654308	1.07654300
ex1223	0.1340	0.0200	6.702	4.57958240	4.57958200
ex1223a	0.1325	0.0100	13.246	4.57958240	4.57958200
ex1223b	0.1546	0.0200	7.729	4.57958240	4.57958200
ex1224	0.1645	0.0300	5.483	-0.94347050	-0.94347050
ex1225	0.1046	0.0100	10.459	31.00000000	31.00000000


Performance Profile
Percent Of Models Solved
100
90
80
70
60
50
40
30
20
10
0
1
10
100
1000
Time Factor
A
B
C
CAN_SOLVE

Lessons

- *Automate* the QA process and the certification
- Build early and *build often*
- *Incorporate* QA tools into the software and share the QA process
- Make the QA process *transparent and reproducible*
- *Involve solver developers and clients* into the QA process

Summary

- SQA becomes more and more important for MP industry
- Our focus is on automated testing (reproducible full life cycle testing)
- Most of the test components are available as transparent GAMS models
- Involvement of the clients in the QA process is essential
- Benefits both for solver developers, for clients and for GAMS
- Open Source!

References

- Bussieck, M.R., Drud, A.S., Meeraus, A. and Pruessner, A. (2002), Quality Assurance and Global Optimization, C. Bliek, C. Jermann, A. Neumaier, eds. Global Optimization and Constraint Satisfaction, First International Workshop on Global Constraint Optimization and Constraint Satisfaction, COCOS 2002, LNCS2861. Springer Verlag, Heidelberg Berlin, (223-238).
- Bussieck, M.R., Dirkse, S.D., Meeraus, A. and Pruessner, A. (2004), Software Quality Assurance for Mathematical Modeling Systems, The Next Wave in Computing, Optimization, and Decision Technologies, Proceedings of Ninth INFORMS Computing Society Conference, B. Golden, S. Raghavan, and E. Wasil (editors), January, 2005
- International Organization for Standardization (2004), online at:http://www.iso.org.
- PAVER Server (2004), online at http://www.gamsworld.org/performance/paver
- Pruessner, A. (2002), Automated Performance Testing and Analysis, online at http://www.gams.com/presentations/present_automation.pdf
- Schulmeyer, G.G. and McManus, J.I. (1998), Software Quality Handbook, Prentice Hall