New Developments in the Design and Operation of the NEOS Server

Robert Fourer

Industrial Engineering & Management Sciences
Northwestern University, Evanston, IL, USA

4er@iems.northwestern.edu

Jorge Moré, Todd Munson, Jason Sarich

Mathematics and Computer Science Division
Argonne National Laboratory, Argonne, IL, USA

[more,tmunson,sarich]@mcs.anl.gov

APMOD 2004
Brunel University, London — Tuesday, June 22, 2004 — TB31

1

The NEOS Server

Server basics

- Use through a web browsers
- Use within a modeling environment

Frequently asked questions

- How is NEOS supported?
- Who uses it? How much?
- What solvers are there?
- How does a user know which solver to choose?

Recent and forthcoming developments

- XML-standard formats
- Problem analysis and solver choice
- Web services
- Benchmarking and verification

Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 6

NEOS www-neos.mcs.anl.gov/neos/

A general-purpose optimization server

- Several dozen solvers in all
- Commercial as well as experimental solvers
- Central scheduler with distributed solver sites

A research project

- Currently free of charge
- Supported through the Optimization Technology Center of Northwestern University & Argonne National Laboratory

. . . 4109 submissions last week
. . . as many as 6094 submissions per week

Using NEOS

Varied submission options

- E-mail
- Web forms
- TCP/IP socket-based submission tool: Java or tcl/tk
- Direct from optimization modeling environments

Numerous formats

- Low-level formats: MPS, SIF, SDPA
- Programming languages: C/ADOL-C, Fortran/ADIFOR
- High-level modeling languages: AMPL, GAMS

Example . . .

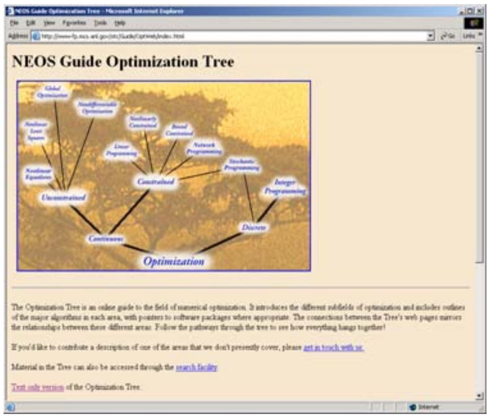
- Investigating solvers
- Using the Web interface
- Using AMPL via the Kestrel interface

Using NEOS

Learn About Your Problem

The NEOS Guide

- Optimization tree: Problem types
- Optimization software guide: Individual solvers
- Frequently asked questions: Varied listings & advice

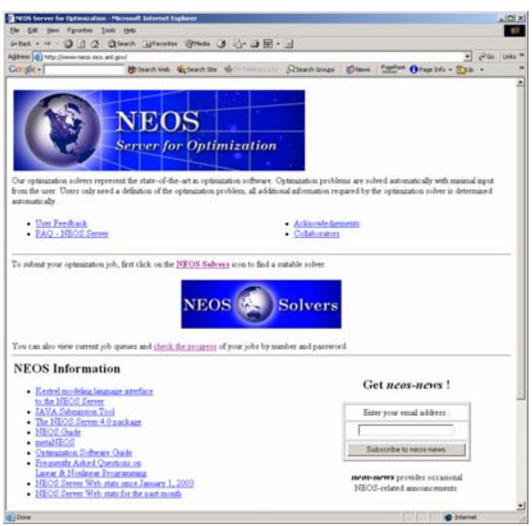


Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 9

Using NEOS

Investigate Solvers

NEOS Server home page

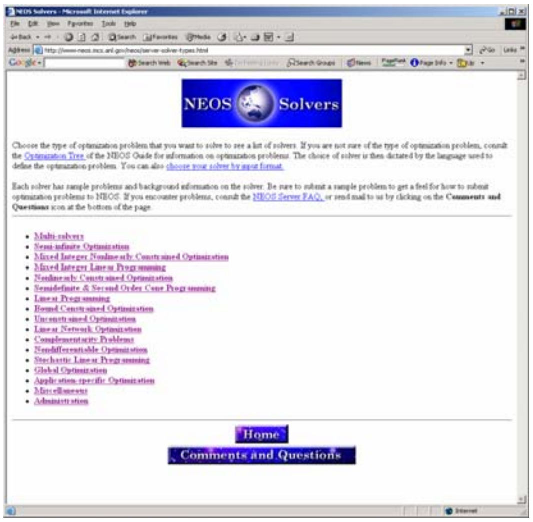


Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 10

Using NEOS

Investigate Solvers

NEOS Server solver type listing

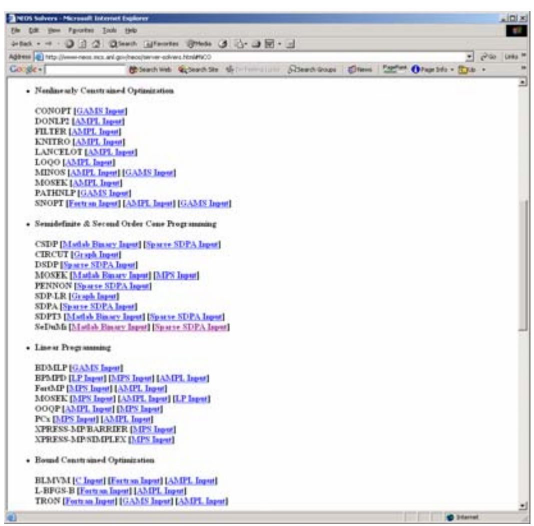


Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 11

Using NEOS

Investigate Solvers

NEOS Server solver listing



Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 12

Using NEOS

Investigate Solvers

Individual solver listing

Using NEOS

Try a Solver: Web Interface

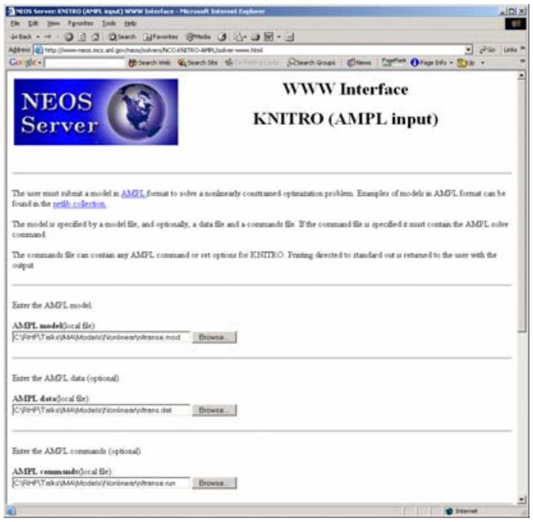
Sample submission form

- “Comments and Questions” button on every page

Using NEOS

Try a Solver: Web Interface

Submission form for your problem

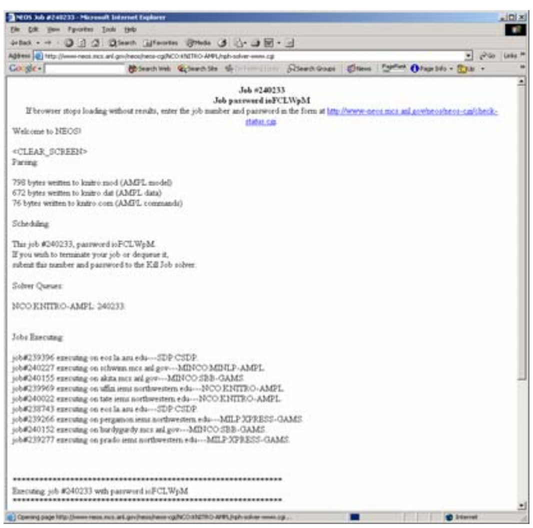


Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 15

Using NEOS

Try a Solver: Web Interface

Start of your run

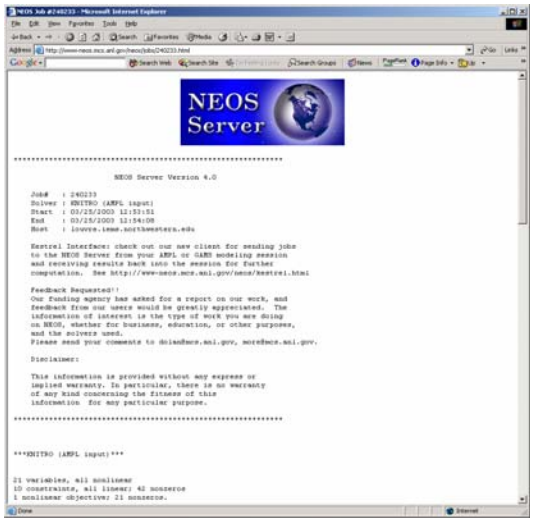


Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 16

Using NEOS

Try a Solver: Web Interface

Beginning of your solution listing

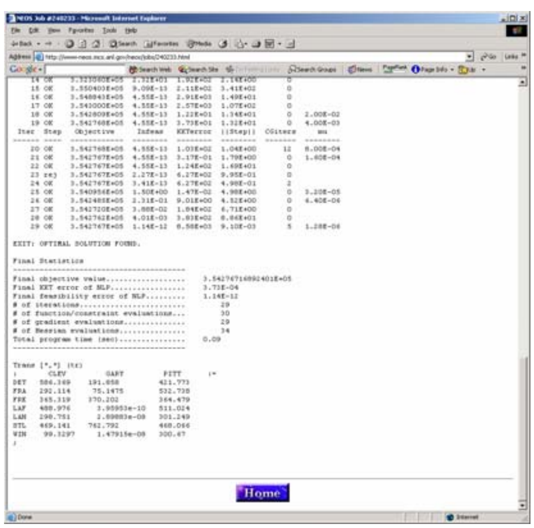


Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 17

Using NEOS

Try a Solver: Web Interface

End of your solution listing

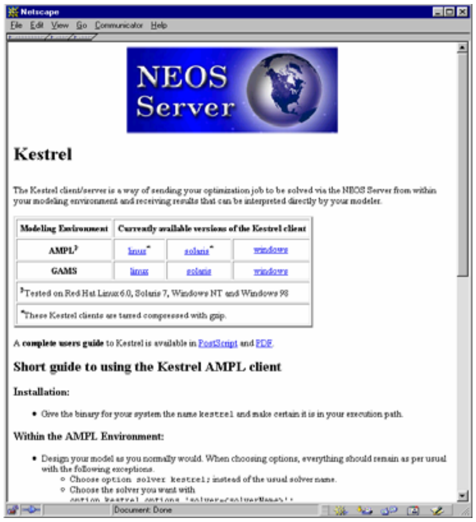


Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 18

Using NEOS

Try a Solver: Kestrel Interface

Kestrel client download page



Using NEOS

Try a Solver: Kestrel Interface

Applying a local solver to an AMPL model

```
AMPL Version 20040202 (MS VC++ 6.0)

ampl: model gs2000b.mod; data gs2000b.dat;
ampl: option solver minos;
ampl: solve;

Presolve eliminates 100 constraints.
Adjusted problem:
4290 variables:
        4260 binary variables
        30 linear variables
733 constraints, all linear; 36340 nonzeros
1 linear objective; 30 nonzeros.

MINOS 5.5:

Sorry, the student edition is limited to 300 variables and
300 constraints.  You have 4290 variables and 733 constraints.

exit code 1
<BREAK>
```

Using NEOS

Try a Solver: Kestrel Interface

Applying a NEOS solver to an AMPL model . . .

```
ampl: option solver kestrel;
ampl: option kestrel_options 'solver=loqo';
ampl: option loqo_options 'minlocfil outlev=1';
ampl: solve;

Job has been submitted to Kestrel

Kestrel/NEOS Job number    : 368607
Kestrel/NEOS Job password  : OxBpVYMb

Check the following URL for progress report :

http://www-neos.mcs.anl.gov/neos/
neos-cgi/check-status.cgi?job=368607&pass=OxBpVYMb

In case of problems, e-mail neos-comments@mcs.anl.gov
```


Using NEOS

Try a Solver: Kestrel Interface

. . . and receiving a solution from NEOS . . .

```
LOQO 6.06: minlocfil

 1    0.000000e+00   2.1e-01   -4.266000e+05   3.1e+02
 2    2.840664e+03   9.8e-04   -4.206215e+05   1.5e+01
 3    2.796511e+03   4.8e-04   -3.078341e+05   6.2e-01
 4    1.769055e+03   4.2e-04   -2.948373e+04   0.0e+00            DF
 5    3.024840e+02   5.2e-05   -3.871922e+03   0.0e+00            DF
 6    3.705130e+01   5.9e-06   -2.158849e+02   0.0e+00            DF
 7    2.220340e+01   3.1e-06   -1.070050e+01   0.0e+00            DF
 8    1.685976e+01   1.4e-06    2.596295e+00   0.0e+00            DF
 9    1.534094e+01   6.7e-07    9.491761e+00   0.0e+00         PF DF
10    1.445050e+01   2.1e-07    1.284805e+01   0.0e+00    1    PF DF
11    1.405725e+01   8.3e-09    1.333832e+01   0.0e+00    1    PF DF
12    1.400313e+01   4.9e-10    1.396657e+01   0.0e+00    3    PF DF
13    1.400016e+01   2.5e-11    1.399833e+01   0.0e+00    4    PF DF
14    1.400001e+01   1.2e-12    1.399992e+01   0.0e+00    5    PF DF

LOQO 6.06: optimal solution (14 QP iterations, 14 evaluations)

primal objective 14.00000783
  dual objective 13.99991642
```

Using NEOS

Try a Solver: Kestrel Interface

. . . where it can be browsed interactively

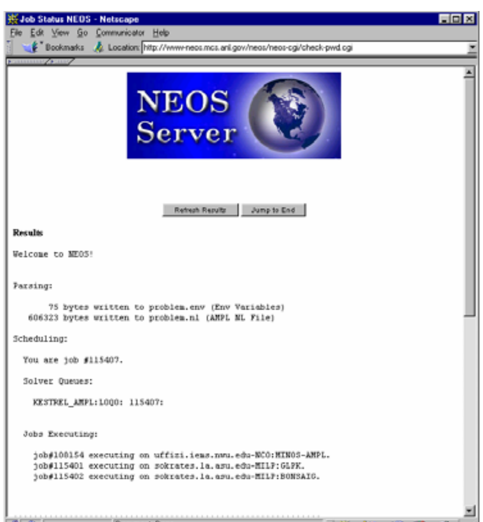
```
ampl: option display_eps .000001;

ampl: display MinType, MaxType;
:                                         MinType  MaxType :=
Division Shipping                            0        1
Division Logistics_and_Supply_Chain          1        2
Division Information_Technology              1        1
Division Production                          0        1
Division Production_Scheduling               1        2
Division Production_Scheduling_Research      1        2
Division Operations_Management               0        1
Division Finance                             3        4
Division Support                             0        1
Office   Americas                            7        8
Office   EMEA                                1        2
Office   Far_East                            1        2
Gender   F                                   3        4
Gender   M                                   7        8
;
```


Using NEOS

Try a Solver: Kestrel Interface

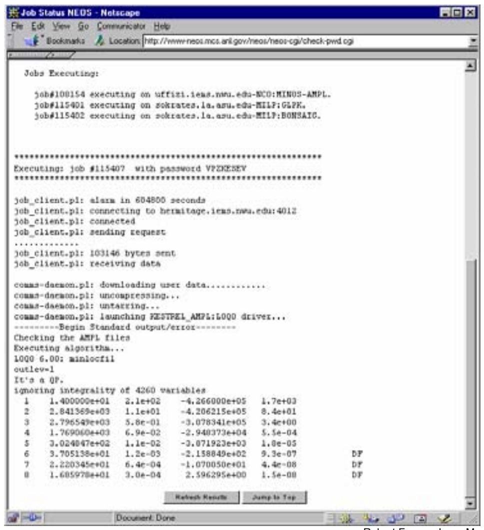
Web form for checking your run's status



Using NEOS

Try a Solver: Kestrel Interface

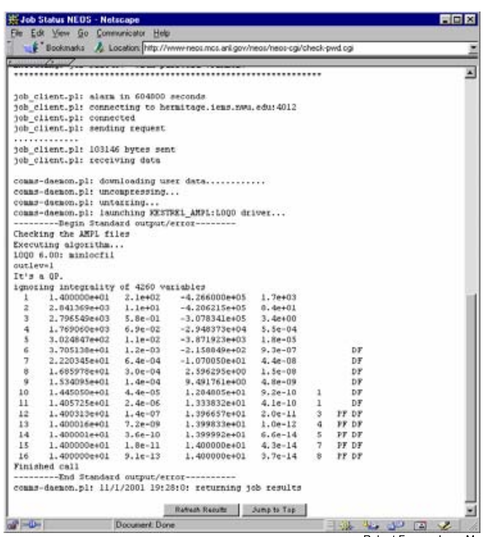
Intermediate status listing



Using NEOS

Try a Solver: Kestrel Interface

Final result listing



Frequently Asked Questions

Who uses NEOS?

- Where are its users from?
- How much is it used?

What kinds of solvers does NEOS offer?

- Who supplies them?
- Which are most heavily used?
- Where are they hosted?

How is NEOS supported?

- Who answers user questions?

Who Uses NEOS? *(a sample)*

- We are using NEOS services for duty-scheduling for ground handling activities in a regional airport environment.
- We used NEOS to solve nonlinear optimization problems associated with models of physical properties in chemistry.
- Our company is working with various projects concerning R&D of internal combustion engines for cars and brakes for heavy vehicles.
- We are working on bi-dimensional modeling of earth's conductivity distribution.
- I am dealing with ultimate limit-state analyses of large dams by means of a non-standard approach (“direct method”); this requires solving problems of linear and non-linear programming. The NEOS server is an extraordinary tool to perform parametric tests on small models, in order to choose the best suited solver.
- I have used NEOS with LOQO solver to optimize an interpolator. . . . My domain is digital receivers where the receiver clock is not changed to match the transmitter clock.

Who Uses NEOS? *(more)*

- I have been able to build and solve a prototype combinatorial auction MIP model using AMPL and NEOS in a fraction of the time it would have required me to do this had I needed to requisition a solver and install it locally.
- Our idea is trying to design antennas by using the computer. . . . We have tried various solvers on NEOS to see if this is possible at all.
- I am using the LOQO solver and code written in AMPL to perform numerical optimization of a spinor Bose-Einstein condensate.
- We are using the NEOS Server for solving linear and nonlinear complementarity problems in engineering mechanics and in robotics.
- I have been working on a system for protein structure prediction. . . . I had need to incorporate a nonlinear solver to handle packing of sidechain atoms in the protein.

Who Uses NEOS? *(academic)*

- I am regularly suggesting my students to use NEOS as soon as their projects in AMPL cannot be solved with the student edition. **So they debug their AMPL models locally . . . and then they run their real-life projects thanks to NEOS.**
- I didn't even know what nonlinear programming was and after I discovered what it was, it became clear how enormous a task it would be to solve the problems assigned to me. . . . I had extremely complicated objective functions, both convex and nonconvex, an armload of variables, and an armload of convex, nonconvex, equality and inequality constraints, but when I sent off the information via the web submission form, within seconds I received extremely accurate and consistent results. **The results were used for verifying a certain theory in my professor's research** and so accuracy was extremely important.
- NEOS has been a very valuable tool in the two graduate optimization courses that I teach. **NEOS allows students to see a broader variety of solvers than we have available . . .**

. . . more at **`www-neos.mcs.anl.gov/neos/stories.html`**

NEOS Users

Where are They From?

2004 through 24 April:
Identifiable domain and >= 20 submissions

(com)	3502
(edu)	13610
(gov)	2510
(net)	12310
(mil)	90

Northwestern	1338
Argonne	2194

Belgium(be)	4918
Brazil(br)	2558
Canada(ca)	4514
Switzerland(ch)	1098
Chile(cl)	158
Colombia(co)	218
Cyprus(cy)	20
Czech Republic(cz)	554
Germany(de)	1834
Spain(es)	1548
Finland(fi)	270
Micronesia(fm)	36
France(fr)	2730
Greece(gr)	776
Hong Kong(hk)	...

NEOS Users

What Countries are They From?

2004 through 24 April

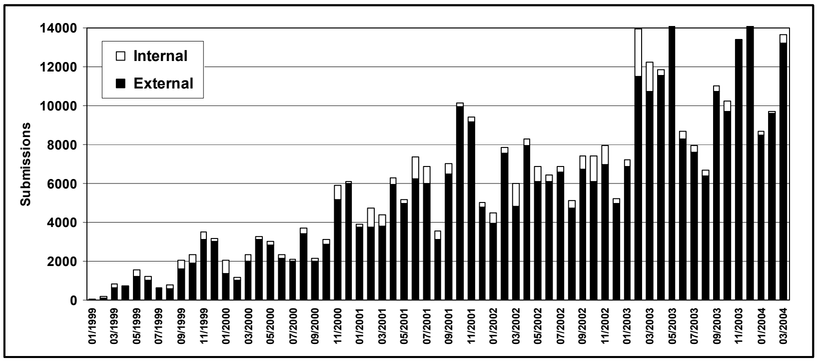
Belgium(be)	4918
Brazil(br)	2558
Canada(ca)	4514
Switzerland(ch)	1098
Chile(cl)	158
Colombia(co)	218
Cyprus(cy)	20
Czech Republic(cz)	554
Germany(de)	1834
Spain(es)	1548
Finland(fi)	270
Micronesia(fm)	36
France(fr)	2730
Greece(gr)	776
Hong Kong(hk)	466
Hungary(hu)	182

Israel(il)	1022
Italy(it)	1646
Japan(jp)	32
Luxembourg(lu)	40
Mexico(mx)	252
Malaysia(my)	912
Netherlands(nl)	4254
Norway(no)	82
New Zealand(nz)	20
Poland(pl)	302
Sweden(se)	3562
Singapore(sg)	236
USSR (former)(su)	834
Turkey(tr)	526
Taiwan(tw)	62
United Kingdom(uk)	4870
Venezuela(ve)	2042

NEOS Users

How Much Do They Use It?

Submissions by month, 1/1999 through 3/2004



. . . 15 / hour over past year
. . . 20 / hour in peak months

What Solvers Does NEOS Offer?

For familiar problem types

- Linear programming
- Linear network optimization
- Linear integer programming
- Nonlinear programming
- Stochastic linear programming
- Complementarity problems

For emerging problem types

- Nondifferentiable optimization
- Semi-infinite optimization
- Global optimization
- Nonlinear integer programming
- Semidefinite & 2nd-order cone programming

. . . virtually every published semidefinite programming code

NEOS Solvers

Who Supplies Them?

Some commercial solver vendors

- Xpress, FortMP (mixed integer)
- CONOPT, KNITRO, MOSEK (nonlinear)

Universities and their researchers

- BonsaiG (mixed integer)
- DONLP2, FILTER, LANCELOT, LOQO, MINOS, SNOPT (nonlinear)

Open-Source Enthusiasts

- GLPK (mixed integer)

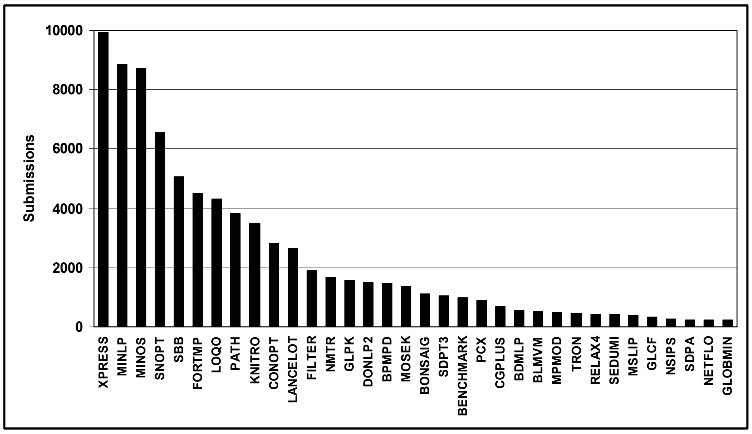
with thanks to . . .

- Modeling language vendors (AMPL, GAMS)
- ***Hans Mittelmann,*** University of Arizona

NEOS Solvers

Which are Most Heavily Used?

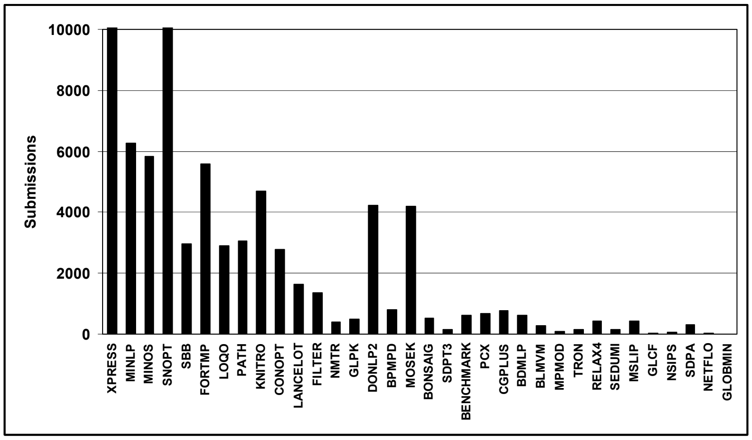
Totals for 2002 . . .



NEOS Solvers

Which are Most Heavily Used?

. . . totals since January 1st, 2003



NEOS Solvers

Where are They Hosted?

Varied workstations at

- Aachen University of Technology
- Argonne National Laboratory
- Arizona State University
- Lehigh University
- National Taiwan University
- Northwestern University *(with support from Sun Microsystems)*
- University of Wisconsin at Madison

. . . new hosts are readily added anywhere on the Internet

How is NEOS Supported?

Grants

- National Science Foundation, Operations Research Program, grant DMI-0322580
- National Science Foundation, Information Technology Research Program, grant CCR-0082807
- U.S. Department of Energy, Office of Advanced Scientific Computing, Mathematical, Information, and Computational Sciences Division subprogram, Contract W-31-109-Eng-38
- National Science Foundation, Challenges in Computational Science Program, grant CDA-9726385

Donations

- Processor cycles
- Many people's time

. . . no user charges as yet

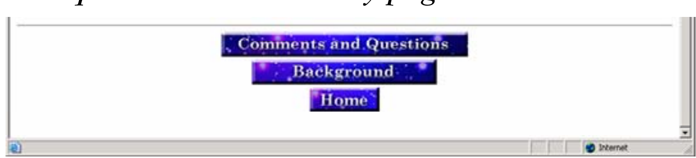
NEOS Support

Who Answers Users' Questions?

Large mailing list for support questions

- NEOS developers
- Solver developers

Support request buttons on every page



Recent & Forthcoming Developments

Problem analysis and solver choice

Web services

Benchmarking and verification

XML-standard formats

- see next talk

Problem Analysis & Solver Choice

Motivation

- Confirm problem type
- Choose appropriate solver

Problem analysis

- Information included with problem instance
- Characteristics determined by analyzer
- Convexity

Solver choice

- Relational database
- Database queries

Problem analysis

Example 1

Transportation with nonlinear costs

```
set ORIG;   # origins
set DEST;   # destinations

param supply {ORIG} >= 0;   # amounts available at origins
param demand {DEST} >= 0;   # amounts required at destinations

param rate {ORIG,DEST} >= 0;   # base shipment costs per unit
param limit {ORIG,DEST} > 0;   # limit on units shipped

var Trans {i in ORIG, j in DEST}
   >= 1e-10, <= .9999 * limit[i,j], := limit[i,j]/2;

minimize Total_Cost:
   sum {i in ORIG, j in DEST}
      rate[i,j] * Trans[i,j]^0.8 / (1 - Trans[i,j]/limit[i,j]);

subject to Supply {i in ORIG}:
   sum {j in DEST} Trans[i,j] = supply[i];

subject to Demand {j in DEST}:
   sum {i in ORIG} Trans[i,j] = demand[j];
```


Problem analysis

Example 1 *(cont'd)*

Transportation data

```
param: ORIG:  supply :=
        GARY   1400
        CLEV   2600
        PITT   2900 ;

param: DEST:  demand :=
        FRA     900    STL    1700
        DET    1200    FRE    1100
        LAN     600    LAF    1000
        WIN     400 ;

param rate :  FRA  DET  LAN  WIN  STL  FRE  LAF :=
        GARY   39   14   11   14   16   82    8
        CLEV   27    9   12    9   26   95   17
        PITT   24   14   17   13   28   99   20 ;

param limit :  FRA  DET  LAN  WIN  STL  FRE  LAF :=
        GARY   500 1000 1000 1000  800  500 1000
        CLEV   500  800  800  800  500  500 1000
        PITT   800  600  600  600  500  500  900 ;
```

Problem analysis

Example 1 *(cont'd)*

AMPL's .nl file: Summary information

```
0 1        # nonlinear constraints, objectives
0 0        # network constraints: nonlinear, linear
0 21 0     # nonlinear vars in constraints, objectives, both
0 0 0 1    # linear network vars; functions; arith, flags
0 0 0 0 0  # discrete vars: binary, integer, nonlinear (b,c,o)
42 21      # nonzeros in Jacobian, gradients
0 0        # max name lengths: constraints, variables
0 0 0 0 0  # common exprs: b,c,o,c1,o1
```

. . . AMPL does all the work here

Problem analysis

Example 1 *(cont'd)*

AMPL's .nl file: Nonlinear expressions

```
O0 0  #Total_Cost
o54   #sumlist
21
o3    #/
o2    #*
n39
o5    #^
v0    #Trans['GARY','FRA']
n0.8
o1    # -
n1
o3    #/
v0    #Trans['GARY','FRA']
n500
o3    #/
o2    #*
n14
o5    #^
........
```

Problem analysis

Example 2

Torsion model (parameters and variables)

```
param nx > 0, integer;     # grid points in 1st direction
param ny > 0, integer;     # grid points in 2nd direction

param c;                   # constant

param hx := 1/(nx+1);      # grid spacing
param hy := 1/(ny+1);      # grid spacing

param area := 0.5*hx*hy;   # area of triangle

param D {i in 0..nx+1,j in 0..ny+1} =

  min( min(i,nx-i+1)*hx, min(j,ny-j+1)*hy );

                           # distance to the boundary

var v {i in 0..nx+1,j in 0..ny+1};

                           # definition of the
                           # finite element approximation
```

Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 60

Problem analysis

Example 2 *(cont'd)*

Torsion model (objective and constraints)

```
var linLower = sum {i in 0..nx, j in 0..ny}
   (v[i+1,j] + v[i,j] + v[i,j+1]);

var linUpper = sum {i in 1..nx+1, j in 1..ny+1}
   (v[i,j] + v[i-1,j] + v[i,j-1]);

var quadLower = sum {i in 0..nx,j in 0..ny} (
   ((v[i+1,j] - v[i,j])/hx)**2 + ((v[i,j+1] - v[i,j])/hy)**2 );

var quadUpper = sum {i in 1..nx+1, j in 1..ny+1} (
   ((v[i,j] - v[i-1,j])/hx)**2 + ((v[i,j] - v[i,j-1])/hy)**2 );

minimize Stress:
  area * ((quadLower+quadUpper)/2 - c*(linLower+linUpper)/3);

subject to distanceBound {i in 0..nx+1, j in 0..ny+1}:
  -D[i,j] <= v[i,j] <= D[i,j];
```

Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 61

Problem analysis

Example 2 *(cont'd)*

Output from AMPL's presolver

```
Presolve eliminates 2704 constraints and 204 variables.
Substitution eliminates 4 variables.

Adjusted problem:
2500 variables, all nonlinear
0 constraints
1 nonlinear objective; 2500 nonzeros.
```


Problem analysis

Example 2 *(cont'd)*

Output from Dr. AMPL prototype (analysis)

```
Problem type
-------------
 -Problem has bounded variables
 -Problem has no constraints

Analyzing problem using only objective
--------------------------------------
 -This objective is quadratic
 -Problem is a QP with bounds

 -0.833013 <= objective <= 0.8359

Problem convexity
------------------
 Nonlinear objective looks convex on its domain.

 Detected 0/0 nonlinear convex  constraints,
          0/0 nonlinear concave constraints.
```

Solver choice

Example 2 *(cont'd)*

Output from Dr. AMPL (solver recommendations)

```
### Specialized solvers, based on all properties ###

        MOSEK
        OOQP

### Specialized solvers, excluding "hard" properties ###

        BLMVM
        FortMP
        L-BFGS-B
        MINLP
        MOSEK
        OOQP
        PathNLP
        SBB
        TRON

 ### General-purpose solvers ###

        KNITRO
        LANCELOT
        LOQO
```


Solver choice

Example 2 *(cont'd)*

Output from MOSEK solver run

```
ampl: model torsion.mod;
ampl: data torsion.dat;

ampl: option solver kestrel;
ampl: option kestrel_options 'solver=mosek';

ampl: solve;

Job has been submitted to Kestrel
Kestrel/NEOS Job number    : 280313
Kestrel/NEOS Job password  : ExPXrRcP

MOSEK finished.
(interior-point iterations - 11, simplex iterations - 0)

Problem status     : PRIMAL_AND_DUAL_FEASIBLE
Solution status    : OPTIMAL

Primal objective  : -0.4180876313
Dual objective    : -0.4180876333
```

Problem analysis

Problem Instance Characteristics

Read from .nl file header

- Size
- Differentiability
- Linearity
- Sparsity

Deduce from expression trees

- Quadraticity
- Smoothness
- Convexity . . .

Problem analysis

Convexity

Significance

- For an optimization problem of the form

$$\begin{array}{ll} \text{Minimize} & f(x_1,\ldots,x_n) \\ \text{Subject to} & g_i(x_1,\ldots,x_n) \ge 0, \quad i=1,\ldots,r \\ & h_i(x_1,\ldots,x_n) = 0, \quad i=1,\ldots,s \end{array}$$

  a local minimum is global provided
  - ⋆ f is convex
  - ⋆ each g_i is convex
  - ⋆ each h_i is linear
- Many physical problems are naturally convex if formulated properly

Analyses . . .

- Disproof of convexity
- Proof of convexity

Problem analysis

Disproof of Convexity

Find any counterexample

- Sample in feasible region
- Test any characterization of convex functions

Sampling along lines

- Look for $f(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) > \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2)$
- See implementation in mProbe
(John Chinneck, www.sce.carleton.ca/faculty/chinneck.html)

Sampling at points

- Look for $\nabla^2 f(\mathbf{x})$ not positive semi-definite
- Implemented in Dr. AMPL . . .

Problem analysis

Disproof of Convexity *(cont'd)*

Sampling

- Choose points $\mathbf{x}_0$
  such that $x_{01}, \ldots, x_{0n}$ are within inferred bounds

Testing

- Apply GLTR (galahad.rl.ac.uk/galahad-www/doc/gltr.pdf) to

$$\min_{\mathbf{d}} \ \nabla f(\mathbf{x}_0)\mathbf{d} + \tfrac{1}{2}\mathbf{d}\nabla^2 f(\mathbf{x}_0)\mathbf{d}$$
$$\text{s.t.} \quad \|\mathbf{d}\|_2 \le \max\{10, \|\nabla f(x_0)\|/10\}$$

- Declare ***nonconvex*** if GLTR's Lanczos method
  finds a direction of negative curvature
- Declare ***inconclusive*** if GLTR reaches the trust region boundary
  without finding a direction of negative curvature

Problem analysis

Proof of Convexity

Recursively assess each expression tree node for

- Bounds
- Monotonicity
- Convexity / Concavity

Apply properties of functions

- $\|\mathbf{x}\|_p$ is convex, ≥ 0 everywhere
- x^α is convex for $\alpha \leq 0$, $\alpha \geq 1$; $-x^\alpha$ is convex for $0 \leq \alpha \leq 1$
- x^p for even $p > 0$ is convex everywhere,
  decreasing on $x \leq 0$, increasing on $x \geq 0$, *etc.*
- $-\log x$ and $x \log x$ are convex and increasing on $x > 0$
- $\sin x$ is concave on $0 \leq x \leq \pi$, convex on $\pi \leq x \leq 2\pi$,
  increasing on $0 \leq x \leq \pi/2$ and $3\pi/2 \leq x \leq 2\pi$, decreasing . . .
  ≥ -1 and ≤ 1 everywhere
- $\mathbf{x}^T\mathbf{M}\mathbf{x}$ is convex if $\mathbf{M}$ is positive semidefinite
- $e^{\alpha x}$ is convex, increasing everywhere for $\alpha > 0$, *etc.*
- $-(\Pi_i\, x_i)^{1/n}$ is convex where all $x_i > 0$ . . . ***etc., etc.***

Problem analysis

Proof of Convexity *(cont'd)*

Apply properties of convexity

- Certain expressions are convex:
  - $-f(\mathbf{x})$ for any concave f
  - $\alpha f(\mathbf{x})$ for any convex f and $\alpha > 0$
  - $f(\mathbf{x}) + g(\mathbf{x})$ for any convex f and g
  - $f(\mathbf{Ax} + \mathbf{b})$ for any convex f
  - $f(g(\mathbf{x}))$ for any convex nondecreasing f and convex g
  - $f(g(\mathbf{x}))$ for any convex nonincreasing f and concave g
- Use these recursively to show that
  nodes in the expression tree are convex

Apply properties of monotonicity & bounds

- Work recursively as for convexity
- Determine all properties in one recursive pass

Problem analysis

Testing Convexity Analyzers

Principles

- Disprovers can establish nonconvexity, suggest convexity
- Provers can establish convexity, suggest nonconvexity

Test problems

- Established test sets:
  COPS (17), CUTE (734), Hock & Schittkowski (119),
  Netlib (40), Schittkowski (195), Vanderbei (29 groups)
- Submissions to NEOS Server

Design of experiments

- Run a prover and a disprover on each test problem
- Check results for consistency
- Collect and characterize problems found to be convex
- Inspect functions not proved or disproved convex,
  to suggest possible enhancements to analyzers

Solver choice

Problem / Solver Data

Relational database

- Table of identifiable *problem* characteristics
- Table of *solvers* and general information about them
- Table of all valid problem-solver pairs

Database queries

- Most specialized solvers
- Moderately specialized solvers:
  "hard" criteria such as convexity not used
- General-purpose solvers

Room for enhancement

- Add data from NEOS Server runs
- Automatically apply "best" solver (or solvers)

Web Services

NEOS currently uses

- E-mail
- Web forms
- TCP/IP sockets (for submission tool)
- CORBA (for Kestrel calls from modeling environments)
- ***Ad hoc* central server design**

NEOS of the future could use

- More general, flexible design
  - SOAP
  - UDDI
  - WSDL
- Less centralized architecture

Web services

New Web Service Standards

SOAP

- Simple Object Access Protocol
- Access to remote objects' methods and data
- Via standard web servers & HTTP protocols
- Using XML description of the interface

. . . more general & flexible than CORBA, DCOM
. . . less client-server coordination required

Web services

New Web Service Standards *(cont'd)*

UDDI

- Universal Description, Discovery, and Integration
- Specification for online web services registry
- Providers list their services
- Users search listings in a standard way

. . . decentralized listing of solvers?

WSDL

- Define XML tags used in accessing a remote service
- Make available through UDDI listings

. . . incorporate optimization XML standard
. . . add XML tags for solver-specific directives

Web services

Issues for NEOS Redesign

Special nature of optimization services

- Potentially huge resource requirements
- Hard-to-predict relations between size and requirements
- Dependence on mathematical form of problem

Client vs. server, local vs. remote

- Scheduling requests
- Analysis of problems
- Databases of problem and solver characteristics
- Choice of solver

. . . prefer to decentralize as much as possible

Web services

Issues for NEOS Scheduler

Many jobs, one processor

- Allow 1 large + several small jobs to share a processor?
- How does the scheduler tell what's "large"?

One job, many processors

- Generalize metaNEOS experiments?

Many jobs, many processors

- Use CONDOR to run on workstations when they are idle

Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 80

Benchmarking

Fundamentals

- Collection of test problems
- Collection of solvers & solver parameter settings

Performance measures

- Number of iterations
- Number of function evaluations
- Number of conjugate gradient iterations
- Computing time

Issues

- Running the solvers
- Verifying the results
- Comparing performance

Robert Fourer, Jorge Moré, Todd Munson & Jason Sarich, APMOD 2004, 22 June 2004, TB31 81

Benchmarking

Running Solvers

NEOS benchmarking tools

- User submits one problem to NEOS benchmark "solver"
- User selects solvers to be compared
- NEOS tries all solvers, using the same computer
- NEOS verifies reported solutions
- NEOS returns listing of results

Other current benchmarking resources

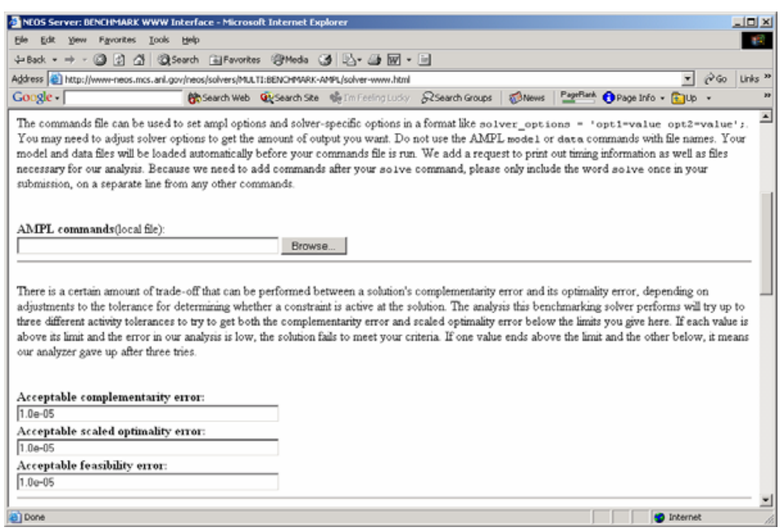
- Hans Mittelmann's benchmark pages, `plato.la.asu.edu/bench.html`
- PAVER performance analysis tool, **`www.gamsworld.org/performance/paver/`**

. . . access to numerous solvers is essential

Benchmarking

NEOS Tools

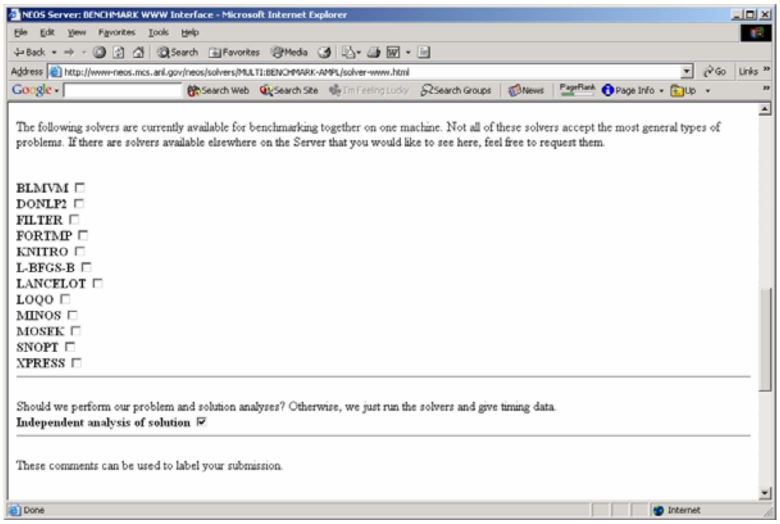
Benchmarking web page (instructions)



Benchmarking

NEOS Tools *(cont'd)*

Benchmarking web page (solver choice)



Benchmarking

Verifying Results

Comparable running environments

- Same computer and operating system
- User's choice of solver parameters
- User's choice of tolerances for
  feasibility, optimality, complementarity

Independent assessment of solutions

- Based only on solution returned

E.D. Dolan, J.J. Moré and T.S. Munson,
"Optimality Measures for Performance Profiles"
(available from www.optimization-online.org)

Benchmarking

NEOS Tools *(cont'd)*

Benchmark verification results

```
Solver lbfgsb.

feasibility error                       0.000000e+00
complementarity error                   0.000000e+00
optimality error                        1.923416e-07
scaled optimality error                 3.827304e-06

Solver solution optimality and complementarity found
acceptable.

Solver loqo.

feasibility error                       0.000000e+00
complementarity error                   7.554012e-05
optimality error                        6.588373e-06
scaled optimality error                 1.311233e-04

Solver solution not acceptable by this analysis because the
scaled optimality error is greater than your limit of 1.0e-05
and the complementarity error is greater than your limit of
1.0e-05.
```


Benchmarking

Comparing Performance

Average or cumulative totals of metric

- Sensitive to results on a small number of problems

Medians or quartiles of a metric

- Information between quartiles is lost

Number of k-th place entries

- No information on the size of improvement

Number of wins by a fixed amount or percentage

- Dependent on the subjective choice of a parameter

Benchmarking

Performance Profiles

Quantities to compute

- For each solver s on each test problem p:
  ratio r_{ps} of that solver's metric to best solver's metric
- For each solver s:
  fraction $\rho_s(\tau)$ of test problems that have $\log_2 r_{ps} \le \tau$

Values to display

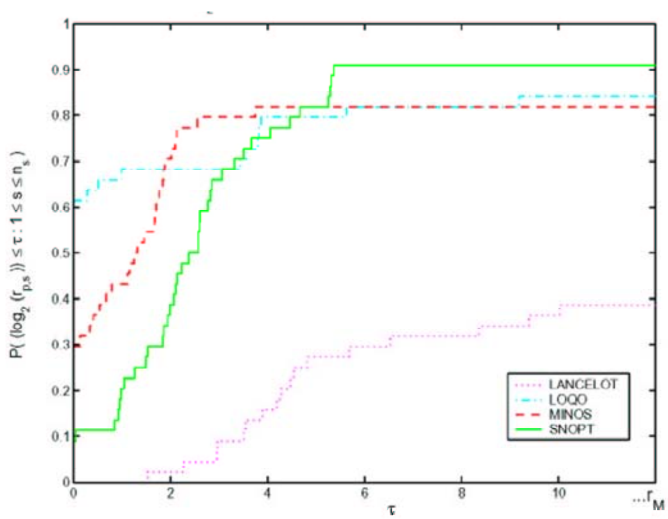
- Plot $\rho_s(\tau)$ vs. τ for each solver s
- $\rho_s : \Re \rightarrow [0,1]$ is a non-decreasing, piecewise constant function
- $\rho_s(0)$ is the fraction of problems on which solver s was best
  $\rho_s(\infty)$ is the fraction of problems on which solver s did not fail
- Emphasis goes from performance to reliability
  as you go from left to right in the plot

E.D. Dolan and J.J. Moré,
"Benchmarking Optimization Software with Performance Profiles."
***Mathematical Programming* 91 (2002) 201–213.**

Benchmarking

Performance Profiles *(cont'd)*

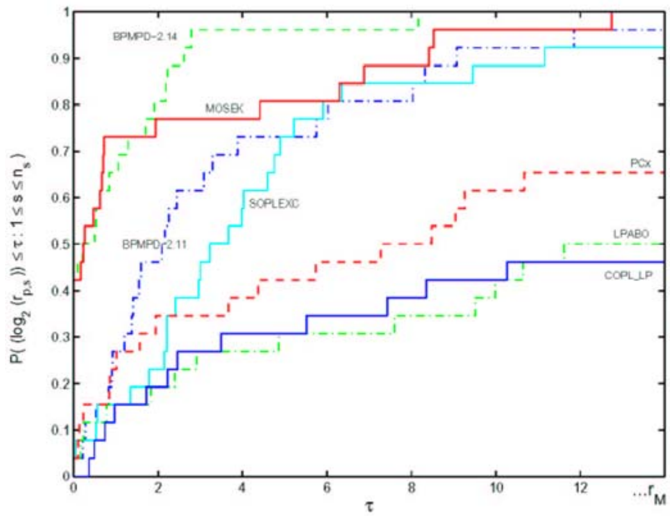
COPS optimal control problems



Benchmarking

Performance Profiles *(cont'd)*

Mittelman test problems



Benchmarking

Performance Profiles *(cont'd)*

Advantages

- Not sensitive to the data on a small number of problems
- Not sensitive to small changes in the data
- Information on the size of improvement is provided
- Does not depend on the subjective choice of a parameter
- Can be used to compare more than two solvers

Further research interests

- Multi-problem NEOS benchmarking tool
- Automated benchmark runs
- Automated generation of result tables & performance profiles

For more information . . .

List of published papers on NEOS

- **www-neos.mcs.anl.gov/neos/faq.html**

Recent writings on NEOS

- Optimization on the NEOS Server,
  SIAM News **35** (2002), 5, 8–9.
- The NEOS Server: Version 4 and Beyond,
  www-neos.mcs.anl.gov/neos/ftp/v4.pdf
- Kestrel: An Interface
  from Modeling Systems to the NEOS Server,
  www-neos.mcs.anl.gov/neos/ftp/kestrel2.pdf

Server websites

- NEOS, **www-neos.mcs.anl.gov**
- OSP, **www.osp-craft.com**
  WEBOPT, **www.webopt.org** — *see TB33, this session*