Modeling and Solving Nontraditional Optimization Problems

Session 1a: Background

Robert Fourer

Industrial Engineering & Management Sciences
Northwestern University

AMPL Optimization LLC

4er@northwestern.edu — 4er@ampl.com

Chiang Mai University International Conference
Workshop
Chiang Mai, Thailand — 4-5 January 2011

Motivation

General-purpose optimization

- Minimize/maximize a function of decision variables
- Subject to equalities/inequalities constraining the values of the variables

General-purpose optimization software

- Solvers
  - apply algorithms to optimization problems
  - specialized to mathematical problem types
- Modeling systems
  - describe models to solvers using representations familiar to people
  - extended to problems & solvers of many types

Traditional Paradigms

Continuous optimization

- Interval domains for decision variables
- Smooth objective and constraint functions
- Locally optimal solutions

Discrete optimization

- Integer decision variables
  - Often zero-one decision variables
  - Often mixed with continuous decision variables
- Linear objective and constraint functions

. . . diverse problem types converted to these forms

Non-Traditional Paradigms

Alternative problem types involving . . .

- Logic operators
- Complementarity conditions
- Conic constraints
- Globally optimal points

Varied solver strategies

- Automated conversions
- Extended traditional algorithms
- New non-traditional algorithms

Sessions

1. *Introduction*
   - *a*. Background
   - *b*. Current features
2. *Nontraditional specialized optimization*
   - *a*. Second-order conic constraints
   - *b*. Complementarity conditions
3. *Nontraditional discrete optimization*
   - *a*. Modeling support
   - *b*. Solver support
4. *Interfacing wih nontraditional solvers*
   - *a*. Solver interfaces
   - *b*. Solver selection

Session 1a: Background

Focus

- Survey of traditional systems
- Availability of software

Topics

- Solvers
- Modeling languages & systems
- Free solver sources
  - NEOS Server
  - COIN-OR

Solving: Problems & Algorithms

Linear programming

Integer programming

Quadratic programming

Nonlinear programming

Solving

Linear Programming

Algorithms

- Scope
  - Linear objective
  - Linear equations and inequalities
- Methods
  - Primal simplex
  - Dual simplex
  - Barrier (interior-point)

Solvers

- Module within nonlinear
  - MINOS
- Module within mixed-integer
  - *see next slide . . .*

Mixed-Integer Programming

Algorithms

- Scope: linear programming with
  - integer variables
  - zero-one variables
- Method: branch-and-bound
  - solve fractional subproblems by dual simplex
  - improve fractional solutions by cut generation
  - seek integer solutions by branching & heuristic search

Solvers

- Commercial
  - CPLEX, Gurobi, MOSEK, XA, Xpress
- Open source
  - CBC, GLPK, lp_solve

PSD Quadratic Programming

Algorithms

- Scope: linear with
  - objective minimizing $x^T Q x$
  - constraints of the form $x^T Q x \leq b$
  - . . . where Q is *positive semi-definite* $(x^T Q x \geq 0 \text{ for all } x)$
- Methods: generalizations of linear algorithms
  - Simplex methods
  - Barrier methods
  - Branch-and-bound procedures

Solvers

- Commercial
  - CPLEX, Gurobi, MOSEK, Xpress

Solving

Nonlinear Programming

Algorithms

- Scope
  - Smooth nonlinear objective and constraints
  - First-order necessary conditions for local optimums
- Methods
  - Generalized reduced gradient
  - Sequential quadratic
  - Interior-point (barrier)

Solvers

- Commercial
  - CONOPT (GRG), KNITRO (IP/SQ), LOQO (IP), MOSEK (IP), MINOS (GRG), SNOPT (SQ)
- Open source
  - Ipopt (IP)

Modeling: Languages & Systems

Goals

- Express models in forms that are familiar and convenient to people
- Support the entire modeling process
  - Formulate
  - Solve
  - Analyze
  - Revise
  - Deploy

Alternatives

- Program your own specialized system
- Adapt an existing language or program
- Use a language & system designed for optimization

Alternatives

Adaptations for optimization modeling

- Spreadsheets
  - Frontline Excel Solver, What's Best
- Math modeling systems
  - MATLAB, Mathematica
- Object-oriented programming languages
  - C++: FLOPC++
  - Python: Pyomo, POAMS, CVXMOD

Algebraic modeling languages for optimization

- Captive
  - OPL (CPLEX), Mosel (Xpress), OPTMODEL (SAS)
- General-purpose
  - AIMMS, **AMPL,** GAMS, MPL

AMPL

Algebraic modeling language: symbolic data

```
set SHIFTS;                 # shifts

param Nsched;               # number of schedules;
set SCHEDS = 1..Nsched;     # set of schedules

set SHIFT_LIST {SCHEDS} within SHIFTS;

param rate {SCHEDS} >= 0;       # pay rates
param required {SHIFTS} >= 0;   # staffing requirements

param least_assign >= 0;        # min workers on any schedule used
```

AMPL

Algebraic modeling language: symbolic model

```
var Work {SCHEDS} >= 0 integer;
var Use  {SCHEDS} >= 0 binary;

minimize Total_Cost:
   sum {j in SCHEDS} rate[j] * Work[j];

subject to Shift_Needs {i in SHIFTS}:
   sum {j in SCHEDS: i in SHIFT_LIST[j]} Work[j] >= required[i];

subject to Least_Use1 {j in SCHEDS}:
   least_assign * Use[j] <= Work[j];

subject to Least_Use2 {j in SCHEDS}:
   Work[j] <= (max {i in SHIFT_LIST[j]} required[i]) * Use[j];
```

AMPL

Explicit data independent of symbolic model

```
set SHIFTS := Mon1 Tue1 Wed1 Thu1 Fri1 Sat1
              Mon2 Tue2 Wed2 Thu2 Fri2 Sat2
              Mon3 Tue3 Wed3 Thu3 Fri3 ;

param Nsched := 126 ;

set SHIFT_LIST[1] := Mon1 Tue1 Wed1 Thu1 Fri1 ;
set SHIFT_LIST[2] := Mon1 Tue1 Wed1 Thu1 Fri2 ;
set SHIFT_LIST[3] := Mon1 Tue1 Wed1 Thu1 Fri3 ;
set SHIFT_LIST[4] := Mon1 Tue1 Wed1 Thu1 Sat1 ;
set SHIFT_LIST[5] := Mon1 Tue1 Wed1 Thu1 Sat2 ;  .......

param required :=  Mon1 100  Mon2 78  Mon3 52
                   Tue1 100  Tue2 78  Tue3 52
                   Wed1 100  Wed2 78  Wed3 52
                   Thu1 100  Thu2 78  Thu3 52
                   Fri1 100  Fri2 78  Fri3 52
                   Sat1 100  Sat2 78 ;
```

AMPL

Solver independent of model & data

```
ampl: model sched1.mod;
ampl: data sched.dat;

ampl: let least_assign := 7;

ampl: option solver cplex;
ampl: solve;

CPLEX 12.2.0.0: optimal integer solution; objective 266
419 MIP simplex iterations
39 branch-and-bound nodes

ampl: option omit_zero_rows 1, display_1col 0;
ampl: display Work;

Work [*] :=
  3  7    18  9    37  7    66  7    82 16   112 23   124 15
  6 21    20  7    41  9    72 13    91 20   118 29
 16 13    29  7    53 13    78 20    94  9   122 21
;
```

AMPL

Language independent of solver

```
ampl: option solver gurobi;
ampl: solve;

Gurobi 4.0.0: optimal solution; objective 266
857 simplex iterations
29 branch-and-cut nodes

ampl: display Work;

Work [*] :=
  1 21    21 36    52  7    89 29    94  7   109 16   124 36
  3  7    37 29    71 13    91 16    95 13   116 36
;
```

AMPL

Nonlinear network example: symbolic data

```
set INTERS;

param EN symbolic;
param EX symbolic;

   check {EN,EX} not within INTERS;

set ROADS within {INTERS union {EN}, INTERS union {EX}};

param time {ROADS} > 0;
param cap {ROADS} > 0;
param sens {ROADS} > 0;

param through > 0;
```

AMPL

Algebraic modeling language: symbolic model

```
var Flow {(i,j) in ROADS} >= 0, <= .9999 * cap[i,j];
var Time {ROADS} >= 0;

minimize Avg_Time:
  (sum {(i,j) in ROADS} Time[i,j] * Flow[i,j]) / through;

subject to Travel_Time {(i,j) in ROADS}:
  Time[i,j] = base[i,j] + (sens[i,j]*Flow[i,j]) / (1-Flow[i,j]/cap[i,j]);

subject to Balance_Node {i in INTERS}:
  sum{(i,j) in ROADS} Flow[i,j] = sum{(j,i) in ROADS} Flow[j,i];

subject to Balance_Flow:
  sum{(EN,j) in ROADS} Flow[EN,j] = through;
```

AMPL

Explicit data independent of symbolic model

```
set INTERS := b c ;

param EN := a;
param EX := d;

param: ROADS: base cap sens :=
         a b    5   10   .1
         a c    1   30   .9
         c b    2   10   .9
         b d    1   30   .9
         c d    5   10   .1 ;

param through := 4;
```

AMPL

Solver independent of model & data

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

ampl: option solver minos;
ampl: solve;

MINOS 5.51: optimal solution found.
7 iterations, objective 8.178571429
Nonlin evals: obj = 16, grad = 15, constrs = 16, Jac = 15.

ampl: display Flow, Time;

:   Flow    Time      :=
a b   2    5.25
a c   2    2.92857
b d   2    2.92857
c b   0    2
c d   2    5.25
;
```

AMPL

Language independent of solver

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

ampl: option solver knitro;
ampl: solve;

KNITRO 6.0.0: Locally optimal solution.
objective 8.178571522; feasibility error 3.73e-07
4 iterations; 5 function evaluations

ampl: display Flow, Time;

:        Flow        Time      :=
a b   2             5.25
a c   2             2.92857
b d   2             2.92857
c b   4.13927e-07   2
c d   2             5.25
;
```

NEOS www.neos-server.org

Network Enabled Optimization System

- Guide
  - tutorials, case studies, test problems, FAQs
- Server
  - free Internet access to solvers

NEOS Server

Since 1995 . . .

- Hosted at Argonne National Laboratory (Illinois, USA)
- Developed through 5 major releases
  - many contributors @ Argonne, Northwestern & elsewhere
  - increasingly sophisticated as Web has matured
- 10-20,000 server submissions in a typical month

. . . has handled over 100,000

A research project

- Currently free of charge
- Supported by grants & volunteer efforts
- ***Moved in December . . .***

NEOS @ WID

Wisconsin Institutes for Discovery **(discovery.wisc.edu)**

- Wisconsin Institute for Discovery (public)
- Morgridge Institute for Research (private)

Key participants

- Michael Ferris
  - research theme leader, *optimization in biology & medicine*
  - coordinator of NEOS move
- Miron Livny
  - founder of the Condor distributed-computing project
  - coordinator of computing technology for WID

Design

Flexible architecture

- Central controller and scheduler machine
- Distributed solver sites

Standard formats

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

Varied submission options

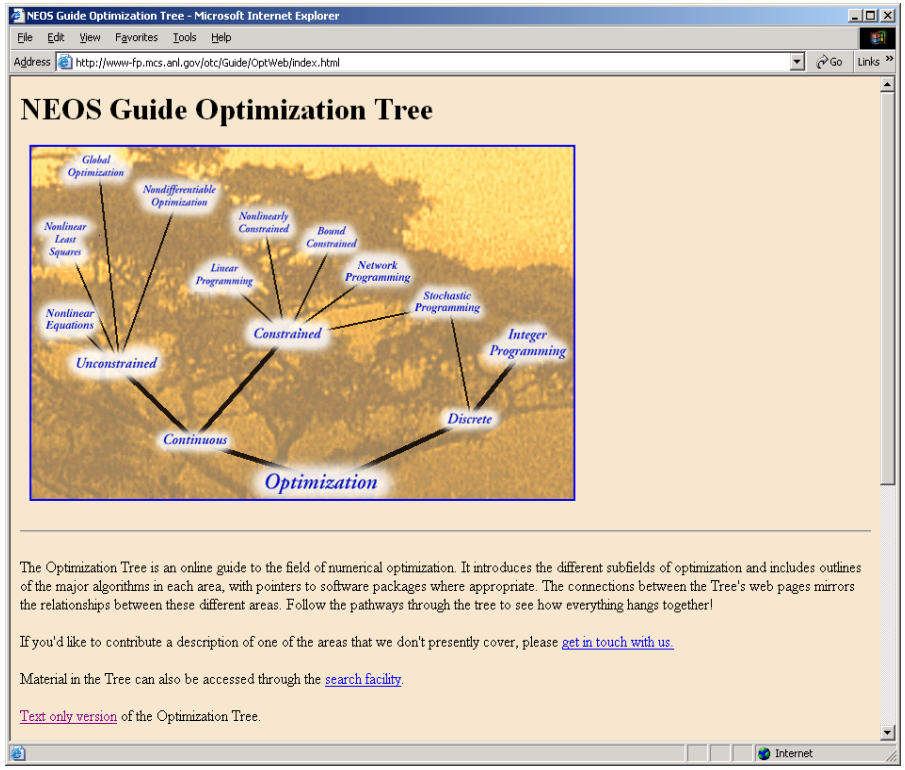
- E-mail
- Web form
- Direct call via XML-RPC
  - from AMPL or GAMS client ***(Kestrel)***
  - from user's client program using NEOS's API

. . . server processes submissions of new solvers, too

Learn About Your Problem

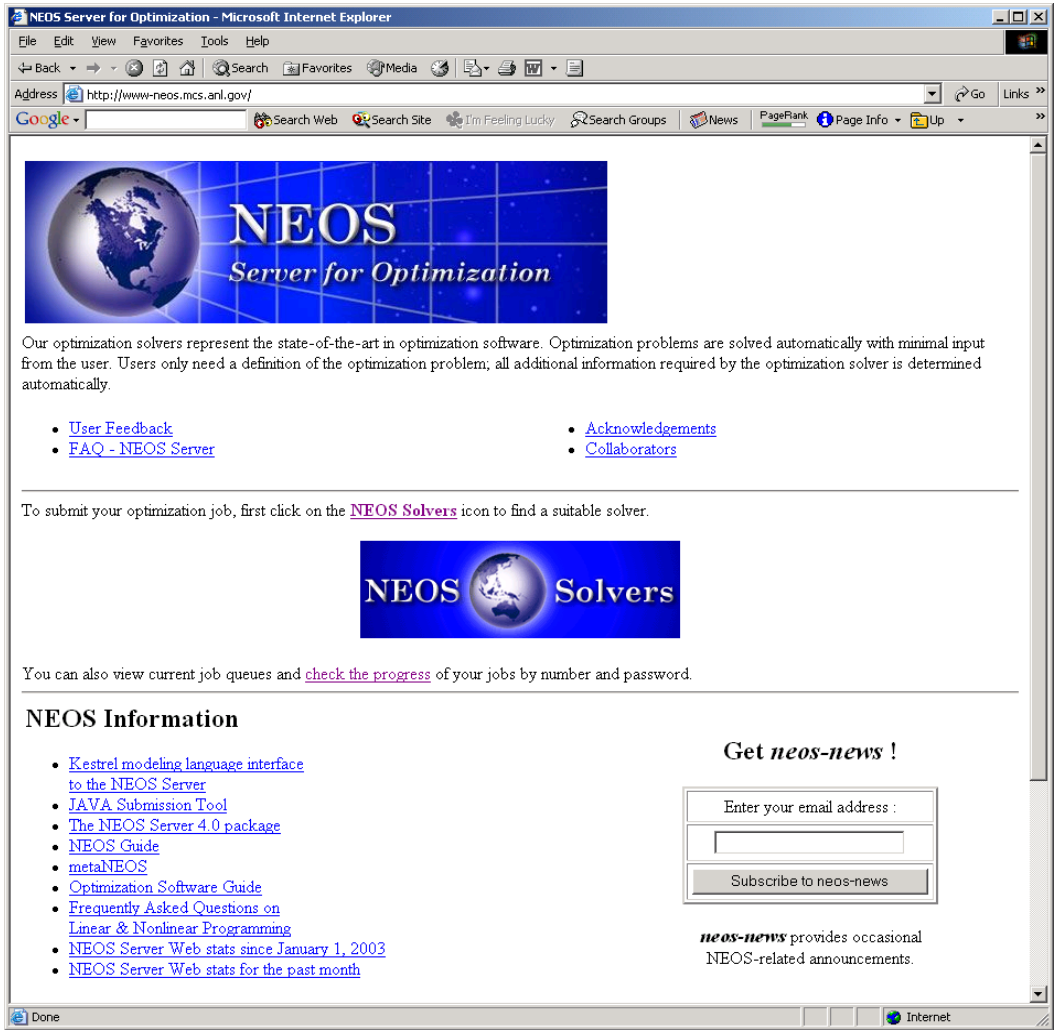
The NEOS Guide

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



Investigate Solvers

NEOS Server home page

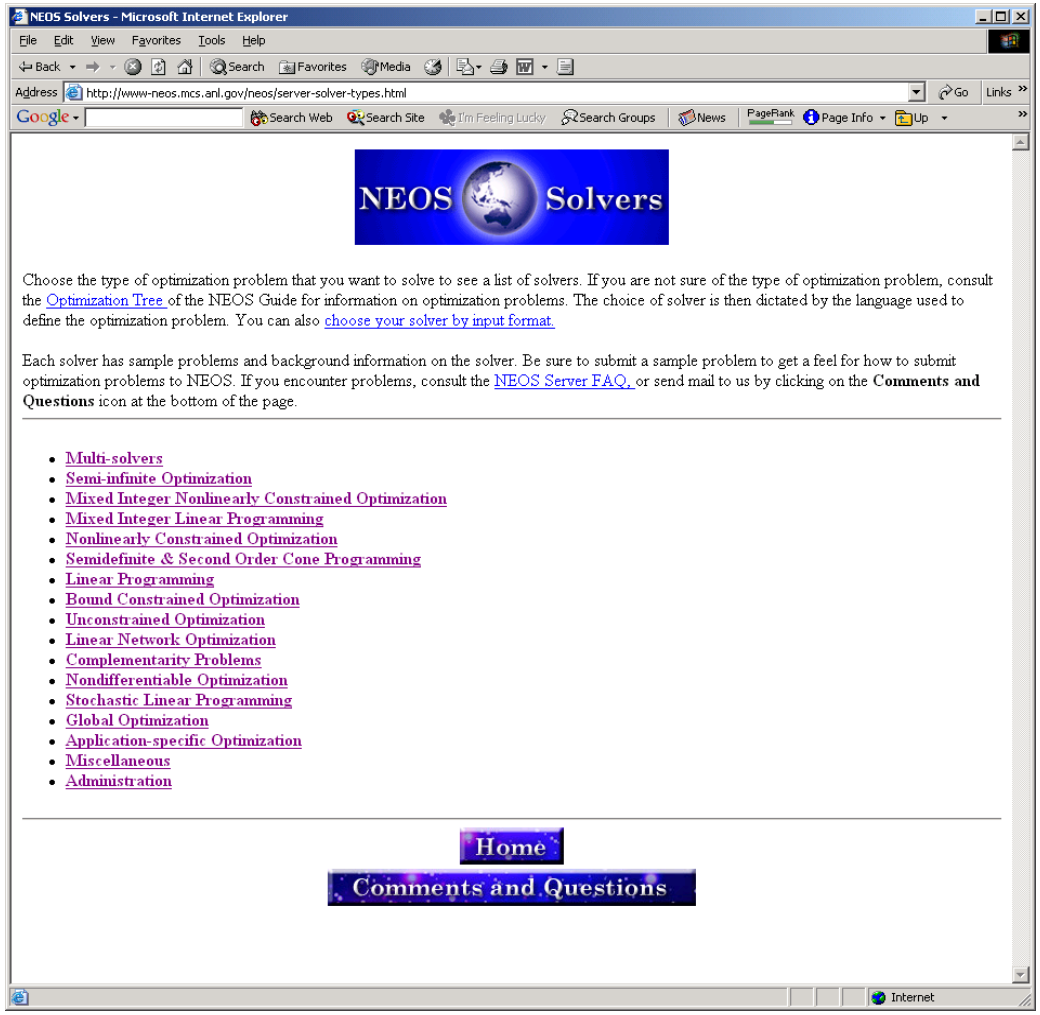


Investigate Solvers

NEOS Server home page (new @ WID)

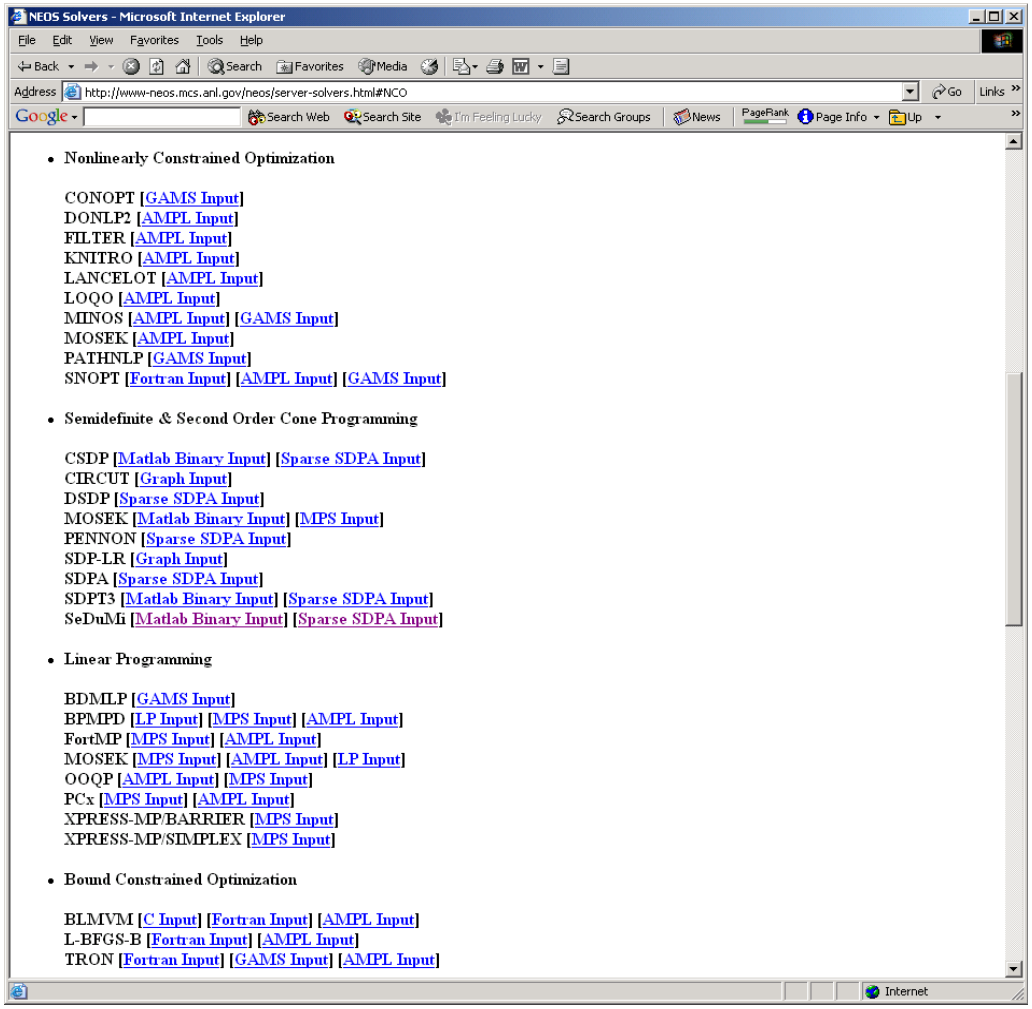
Investigate Solvers

NEOS Server solver type listing



Investigate Solvers

NEOS Server solver listing



- **Nonlinearly Constrained Optimization**

  CONOPT [GAMS Input]
  DONLP2 [AMPL Input]
  FILTER [AMPL Input]
  KNITRO [AMPL Input]
  LANCELOT [AMPL Input]
  LOQO [AMPL Input]
  MINOS [AMPL Input] [GAMS Input]
  MOSEK [AMPL Input]
  PATHNLP [GAMS Input]
  SNOPT [Fortran Input] [AMPL Input] [GAMS Input]

- **Semidefinite & Second Order Cone Programming**

  CSDP [Matlab Binary Input] [Sparse SDPA Input]
  CIRCUT [Graph Input]
  DSDP [Sparse SDPA Input]
  MOSEK [Matlab Binary Input] [MPS Input]
  PENNON [Sparse SDPA Input]
  SDP-LR [Graph Input]
  SDPA [Sparse SDPA Input]
  SDPT3 [Matlab Binary Input] [Sparse SDPA Input]
  SeDuMi [Matlab Binary Input] [Sparse SDPA Input]

- **Linear Programming**

  BDMLP [GAMS Input]
  BPMPD [LP Input] [MPS Input] [AMPL Input]
  FortMP [MPS Input] [AMPL Input]
  MOSEK [MPS Input] [AMPL Input] [LP Input]
  OOQP [AMPL Input] [MPS Input]
  PCx [MPS Input] [AMPL Input]
  XPRESS-MP/BARRIER [MPS Input]
  XPRESS-MP/SIMPLEX [MPS Input]

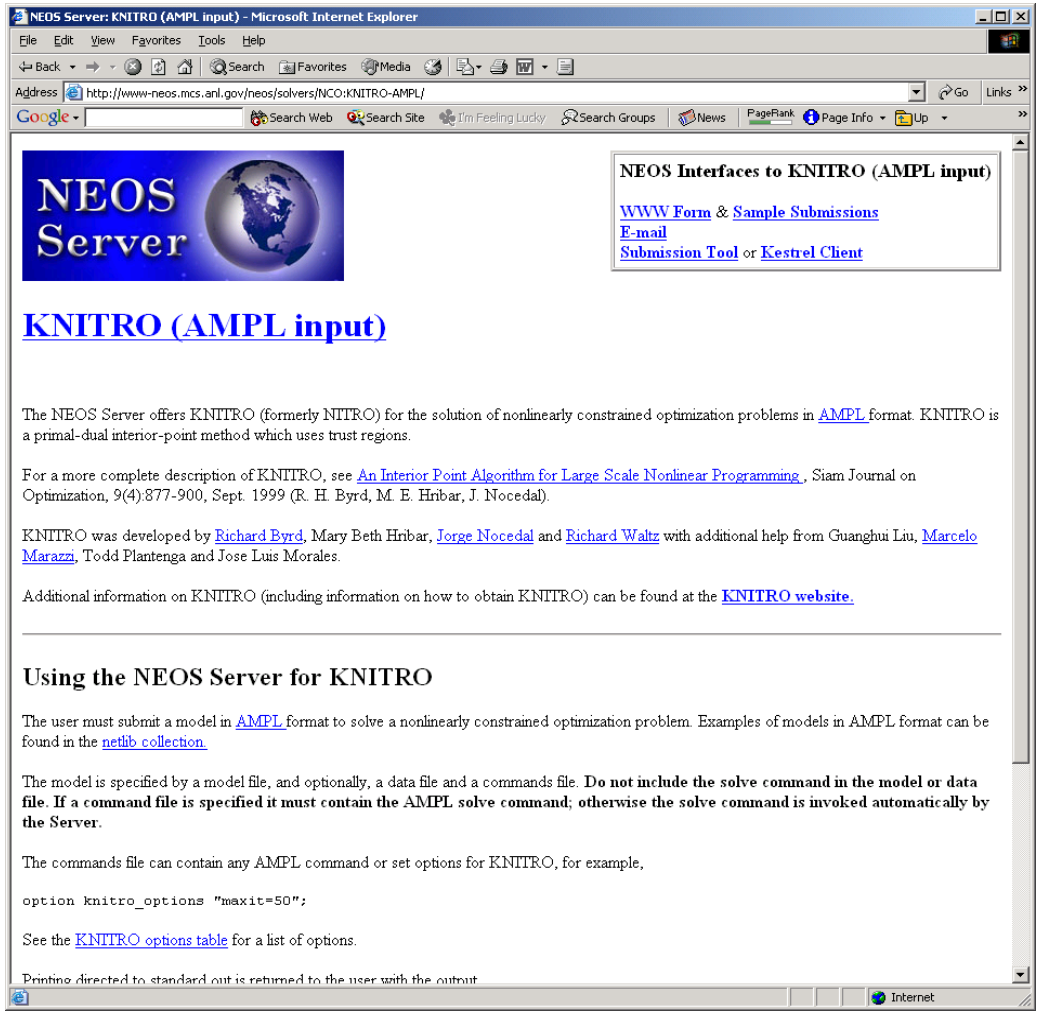
- **Bound Constrained Optimization**

  BLMVM [C Input] [Fortran Input] [AMPL Input]
  L-BFGS-B [Fortran Input] [AMPL Input]
  TRON [Fortran Input] [GAMS Input] [AMPL Input]

Using NEOS

Investigate Solvers

Individual solver listing



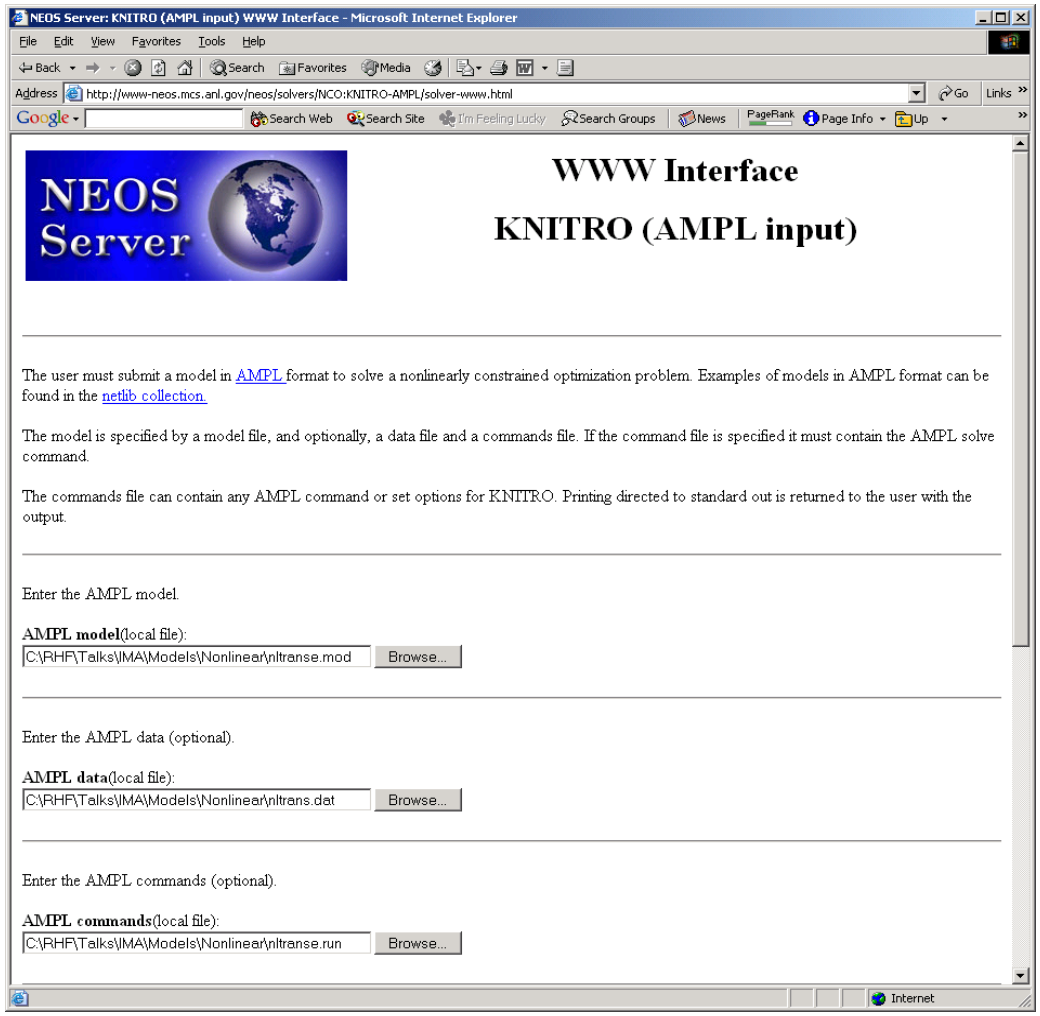
Try a Solver: Web Interface

Sample submission form

- "Comments and Questions" button on every page

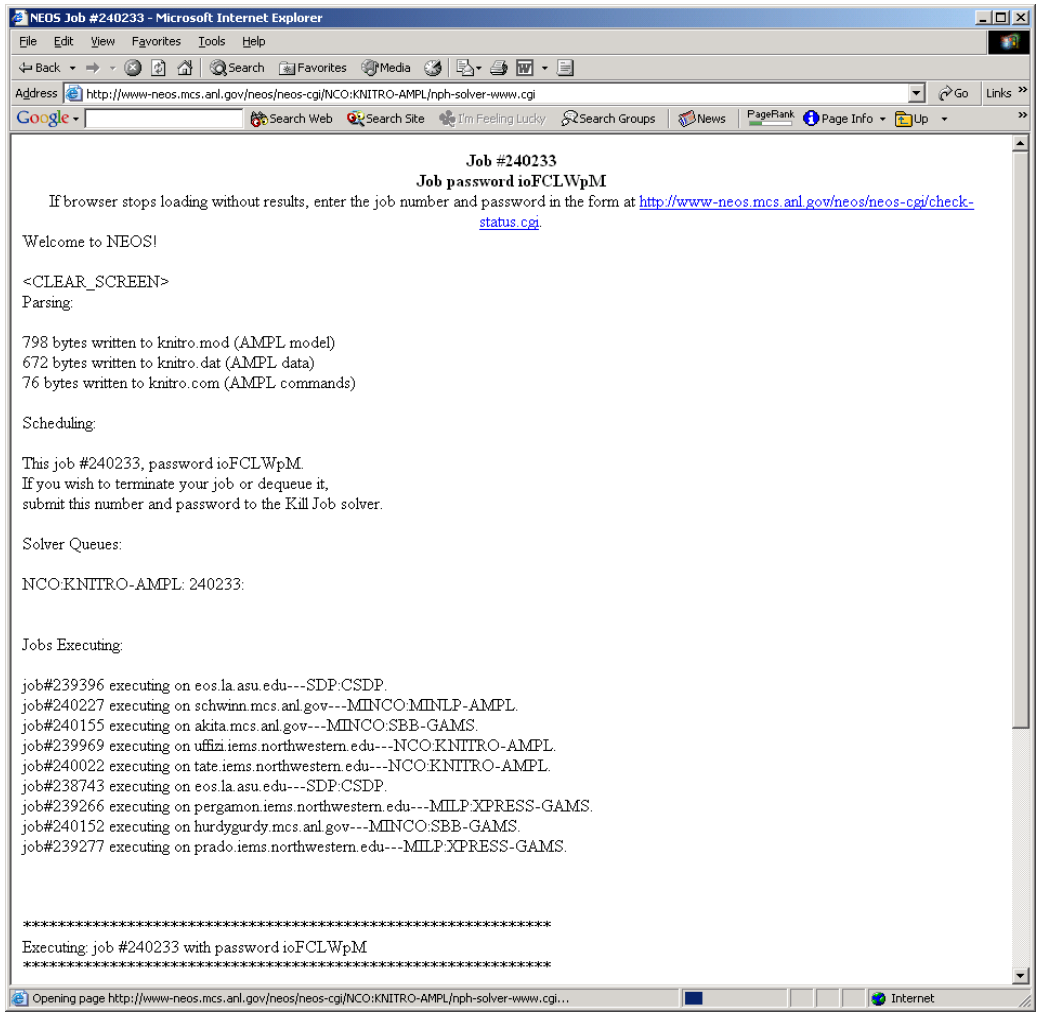
Try a Solver: Web Interface

Submission form for your problem



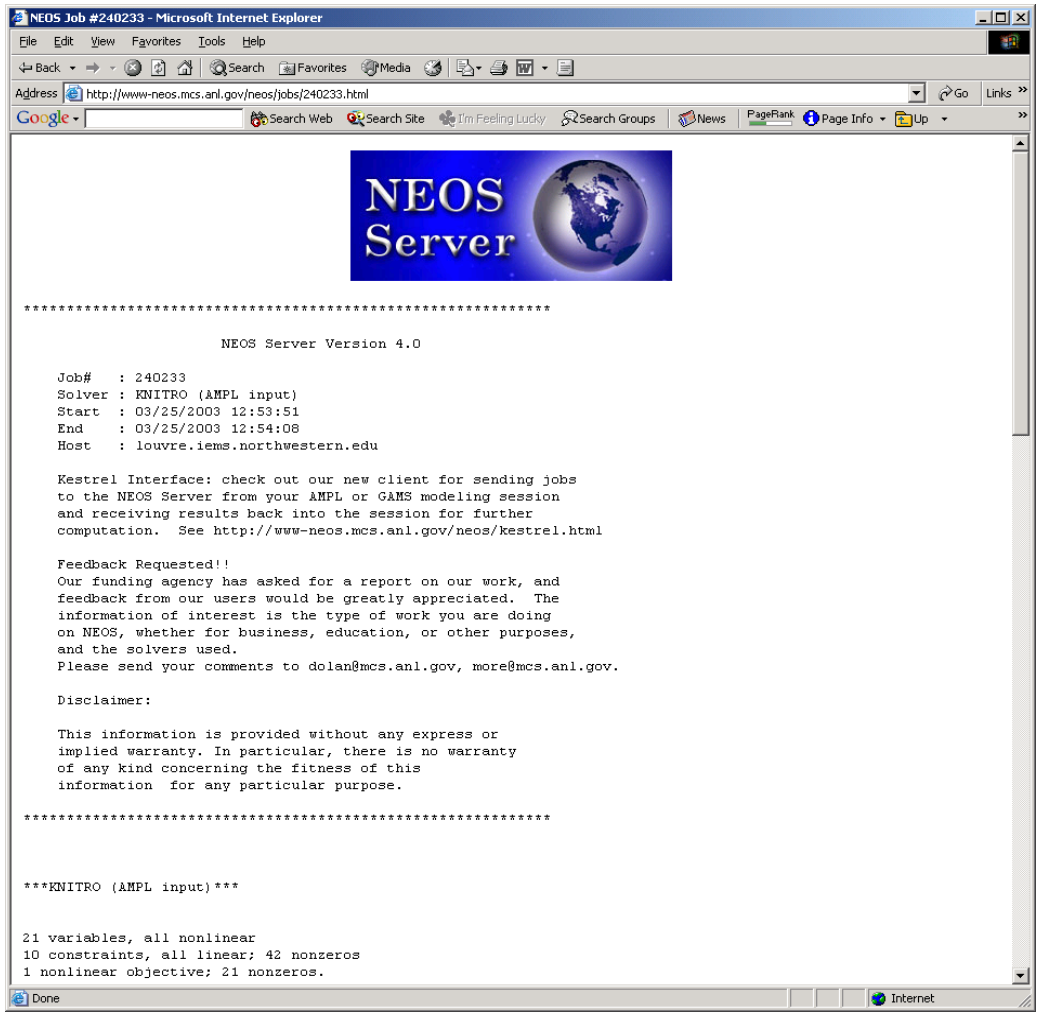
Try a Solver: Web Interface

Start of your run



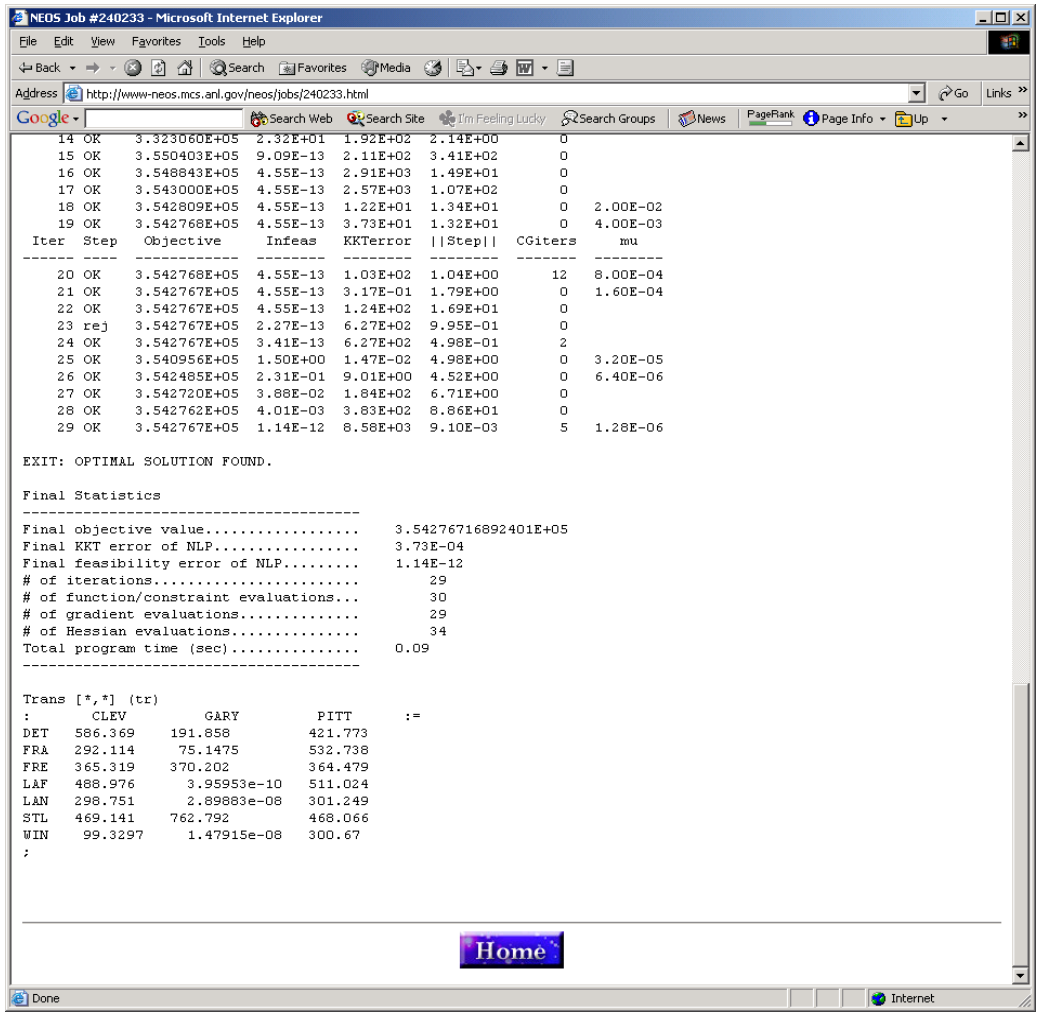
Try a Solver: Web Interface

Beginning of your solution listing



Try a Solver: Web Interface

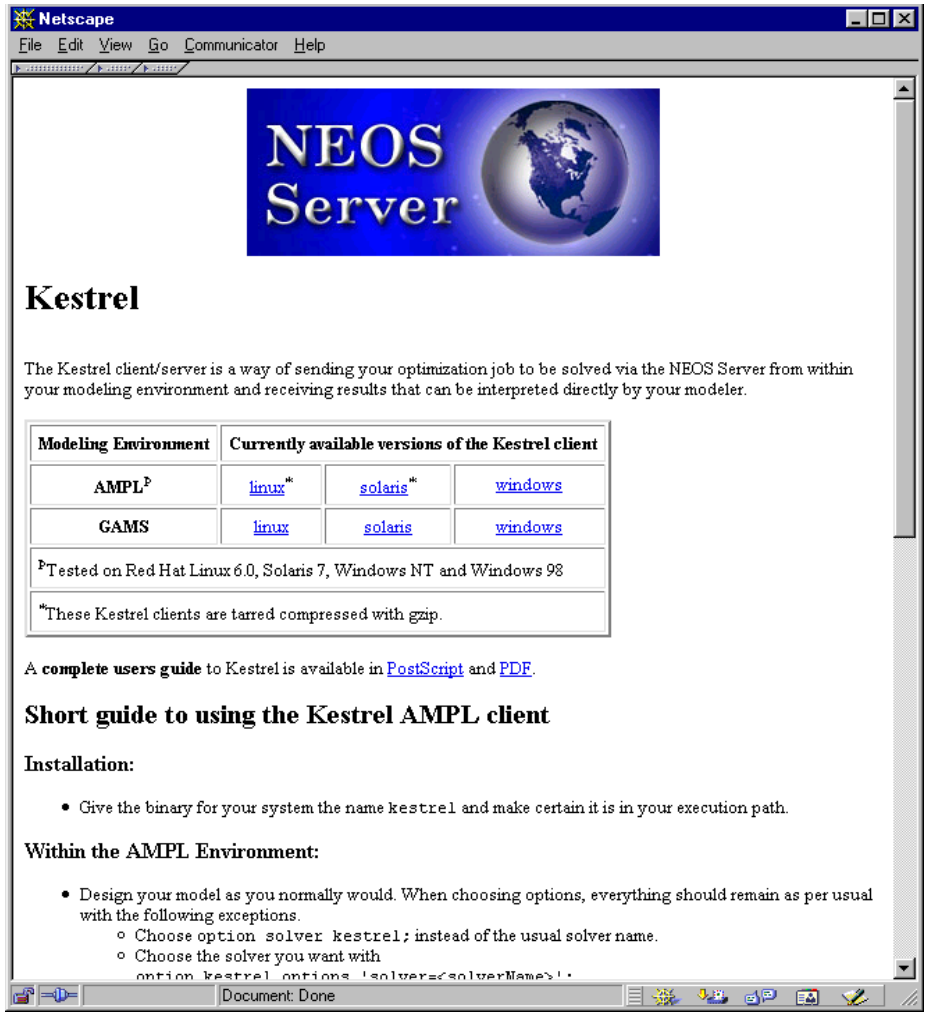
End of your solution listing




```
    14 OK    3.323060E+05  2.32E+01  1.92E+02  2.14E+00       0
    15 OK    3.550403E+05  9.09E-13  2.11E+02  3.41E+02       0
    16 OK    3.548843E+05  4.55E-13  2.91E+03  1.49E+01       0
    17 OK    3.543000E+05  4.55E-13  2.57E+03  1.07E+02       0
    18 OK    3.542809E+05  4.55E-13  1.22E+01  1.34E+01       0   2.00E-02
    19 OK    3.542768E+05  4.55E-13  3.73E+01  1.32E+01       0   4.00E-03
 Iter  Step   Objective     Infeas   KKTerror  ||Step||  CGiters     mu
------ ----  ------------  --------  --------  --------  -------  --------
    20 OK    3.542768E+05  4.55E-13  1.03E+02  1.04E+00      12   8.00E-04
    21 OK    3.542767E+05  4.55E-13  3.17E-01  1.79E+00       0   1.60E-04
    22 OK    3.542767E+05  4.55E-13  1.24E+02  1.69E+01       0
    23 rej   3.542767E+05  2.27E-13  6.27E+02  9.95E-01       0
    24 OK    3.542767E+05  3.41E-13  6.27E+02  4.98E-01       2
    25 OK    3.540956E+05  1.50E+00  1.47E-02  4.98E+00       0   3.20E-05
    26 OK    3.542485E+05  2.31E-01  9.01E+00  4.52E+00       0   6.40E-06
    27 OK    3.542720E+05  3.88E-02  1.84E+02  6.71E+00       0
    28 OK    3.542762E+05  4.01E-03  3.83E+02  8.86E+01       0
    29 OK    3.542767E+05  1.14E-12  8.58E+03  9.10E-03       5   1.28E-06

EXIT: OPTIMAL SOLUTION FOUND.

Final Statistics
----------------------------------------
Final objective value.................    3.54276716892401E+05
Final KKT error of NLP................    3.73E-04
Final feasibility error of NLP........    1.14E-12
# of iterations.....................        29
# of function/constraint evaluations...     30
# of gradient evaluations.............        29
# of Hessian evaluations..............        34
Total program time (sec)..............    0.09
----------------------------------------

Trans [*,*] (tr)
:       CLEV          GARY          PITT      :=
DET    586.369     191.858          421.773
FRA    292.114      75.1475         532.738
FRE    365.319     370.202          364.479
LAF    488.976       3.95953e-10    511.024
LAN    298.751       2.89883e-08    301.249
STL    469.141     762.792          468.066
WIN     99.3297      1.47915e-08    300.67
;
```

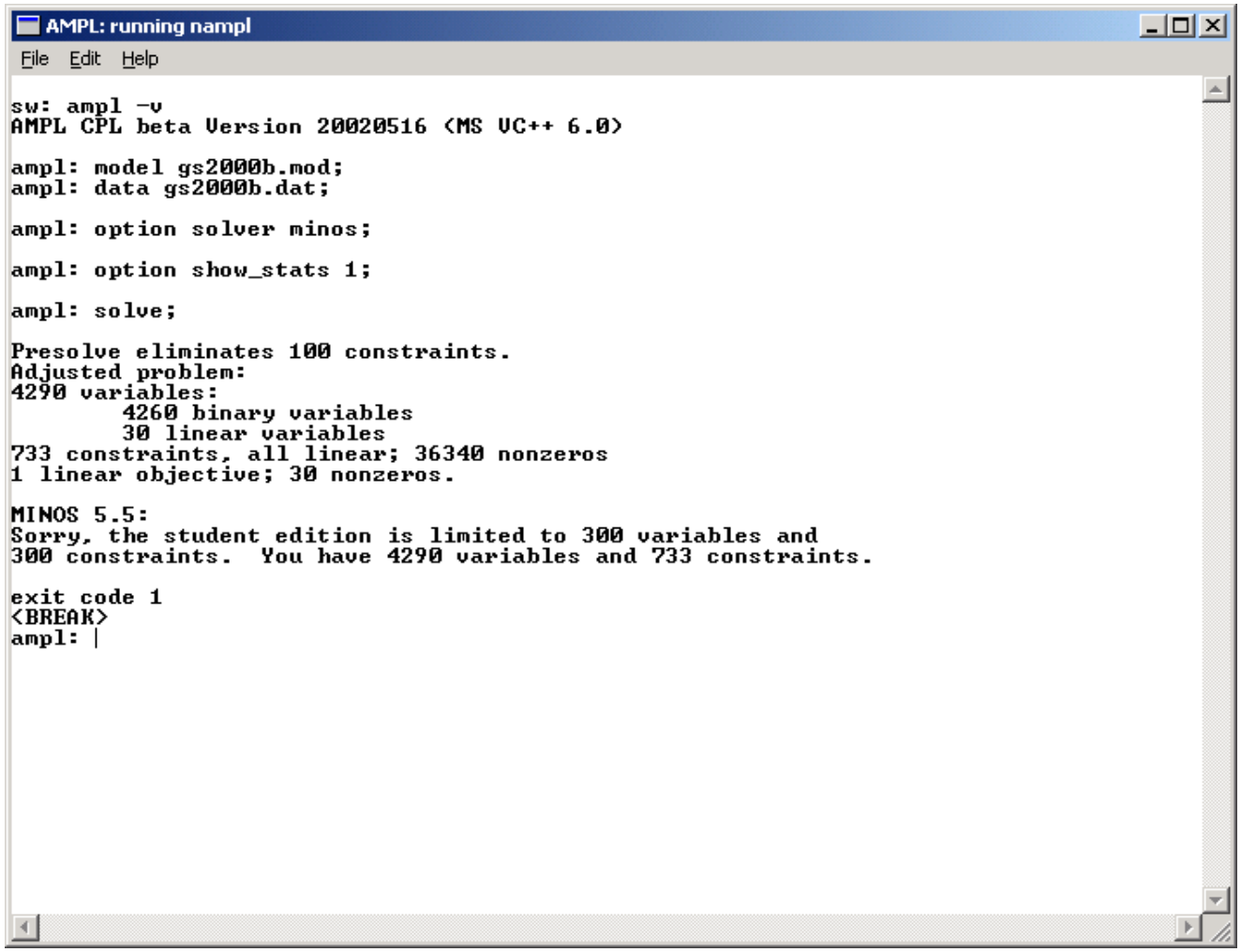
Try a Solver: Kestrel Interface

Kestrel client download page



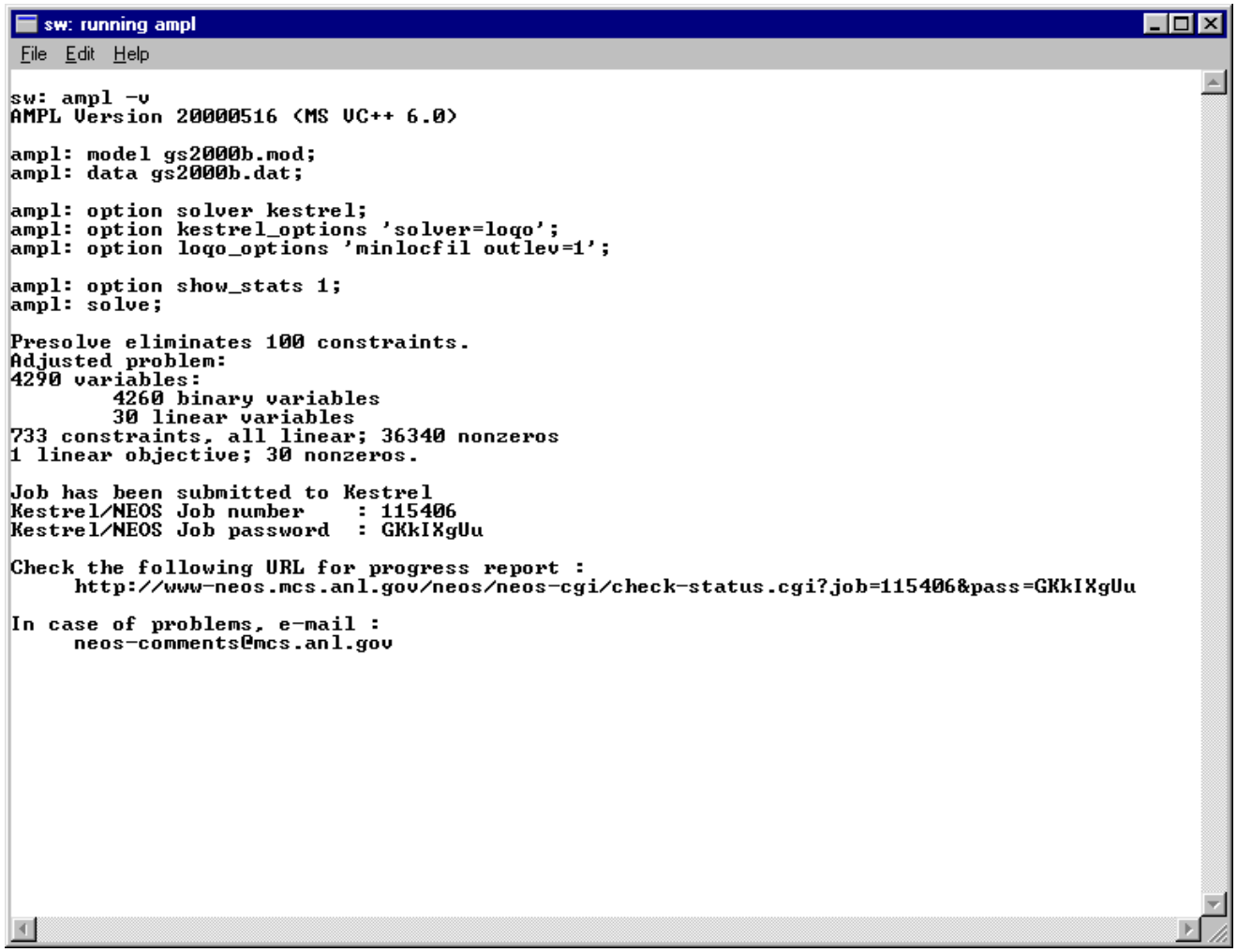
Try a Solver: Kestrel Interface

Applying a local solver to an AMPL model



Try a Solver: Kestrel Interface

Applying a NEOS solver to an AMPL model . . .



Try a Solver: Kestrel Interface

. . . and receiving a solution from the NEOS

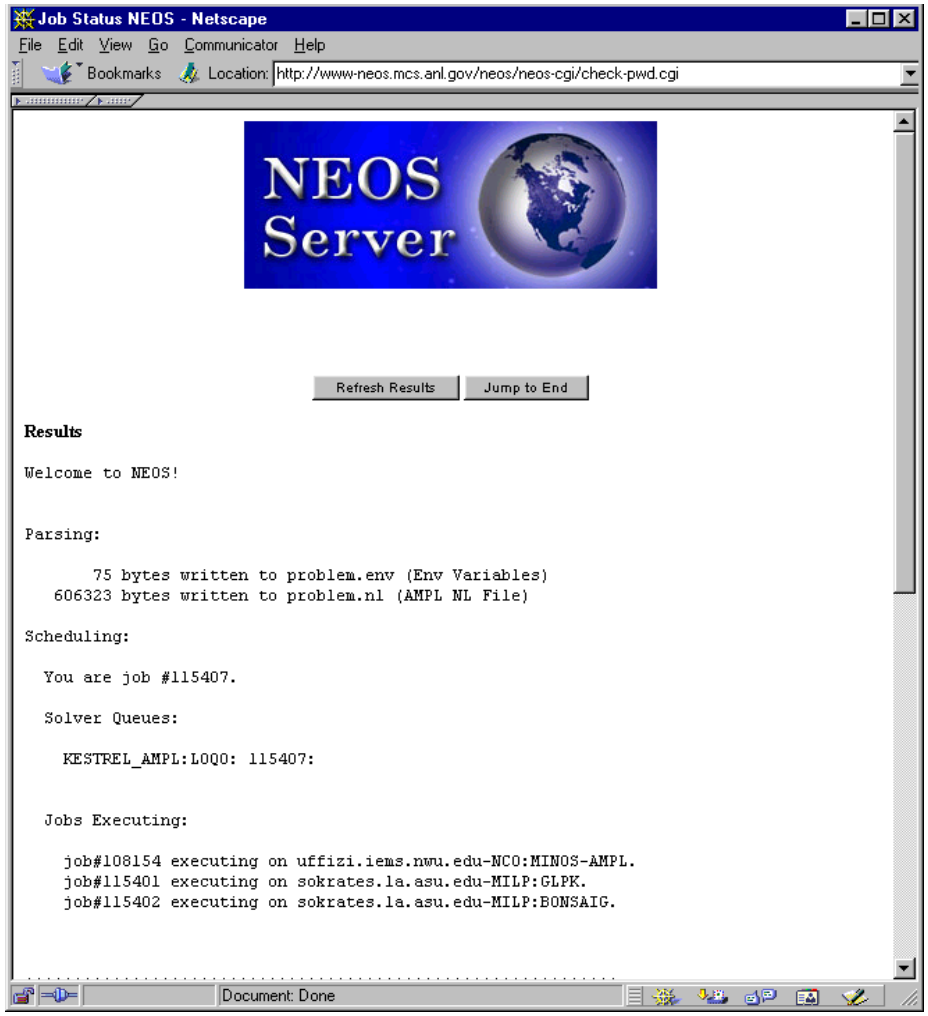
sw: running ampl

File Edit Help

```
Check the following URL for progress report :
     http://www-neos.mcs.anl.gov/neos/neos-cgi/check-status.cgi?job=115406&pass=GKkIXgUu

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

Intermediate Solver Output:
Checking the AMPL files
Executing algorithm...

LOQO 6.00: minlocfil
outlev=1

It's a QP.
ignoring integrality of 4260 variables

  1    0.000000e+00   2.1e+02    -4.263593e+05   1.7e+03
  2    2.839512e+03   1.1e+01    -4.206438e+05   8.8e+01
  3    2.803962e+03   5.8e-01    -3.084425e+05   3.7e+00
  4    1.804909e+03   7.0e-02    -2.965997e+04   1.4e-13            DF
  5    3.154594e+02   1.1e-02    -3.913235e+03   1.7e-13            DF
  6    3.771029e+01   1.2e-03    -2.201994e+02   4.6e-14            DF
  7    2.235023e+01   6.4e-04    -1.072541e+01   3.6e-14            DF
  8    1.700808e+01   3.1e-04     2.429905e+00   2.9e-14            DF
  9    1.536949e+01   1.4e-04     9.410120e+00   4.6e-14            DF
 10    1.446494e+01   4.4e-05     1.271394e+01   3.8e-14    1       DF
 11    1.405838e+01   2.4e-06     1.326864e+01   3.0e-14    1       DF
 12    1.400320e+01   1.4e-07     1.396308e+01   3.5e-14    3    PF DF
 13    1.400016e+01   7.3e-09     1.399815e+01   3.9e-14    4    PF DF
 14    1.400001e+01   3.6e-10     1.399991e+01   3.5e-14    5    PF DF
 15    1.400000e+01   1.8e-11     1.400000e+01   4.1e-14    6    PF DF
 16    1.400000e+01   9.1e-13     1.400000e+01   3.5e-14    8    PF DF
Finished call

LOQO 6.00: optimal solution (16 QP iterations, 31 evaluations)
primal objective 14.00000002
  dual objective 13.99999977

ampl: display MinNotDom, MaxNotDom;

:               MinNotDom MaxNotDom    :=
Office Americas       3         4
;

ampl:
```

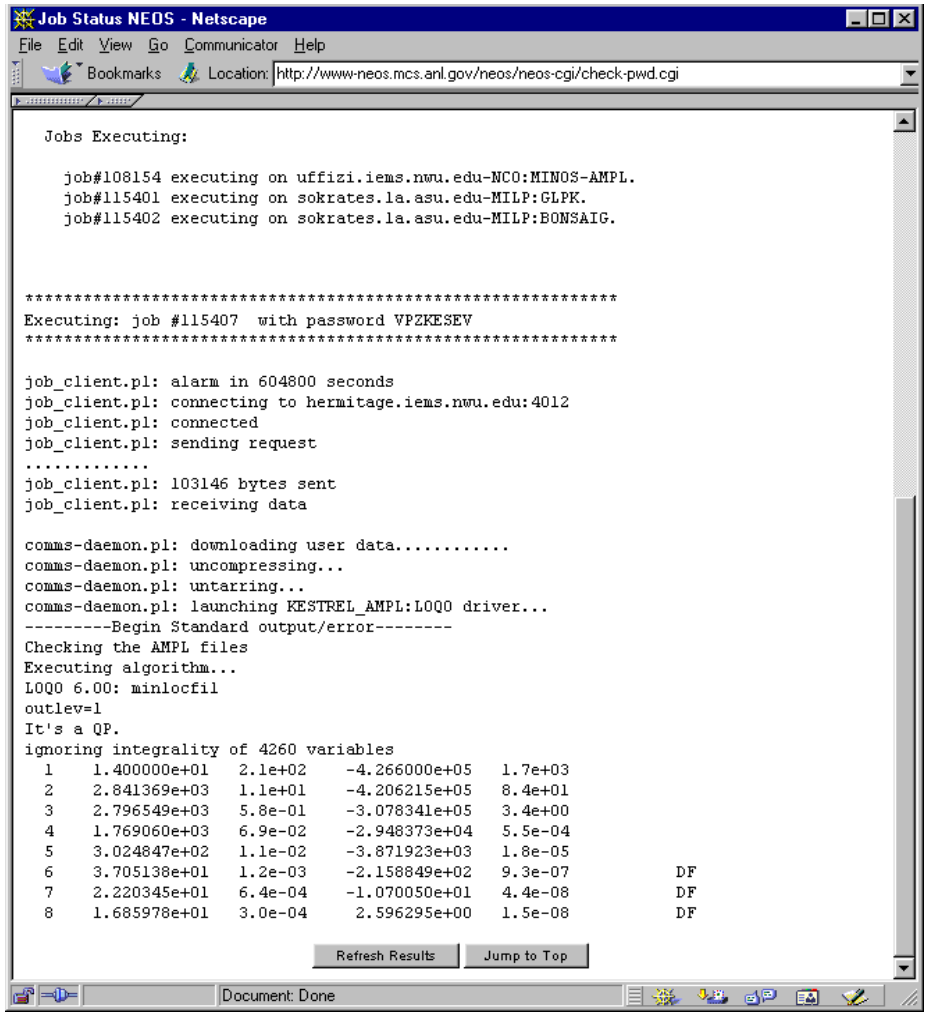
Try a Solver: Kestrel Interface

Web form for checking your run's status



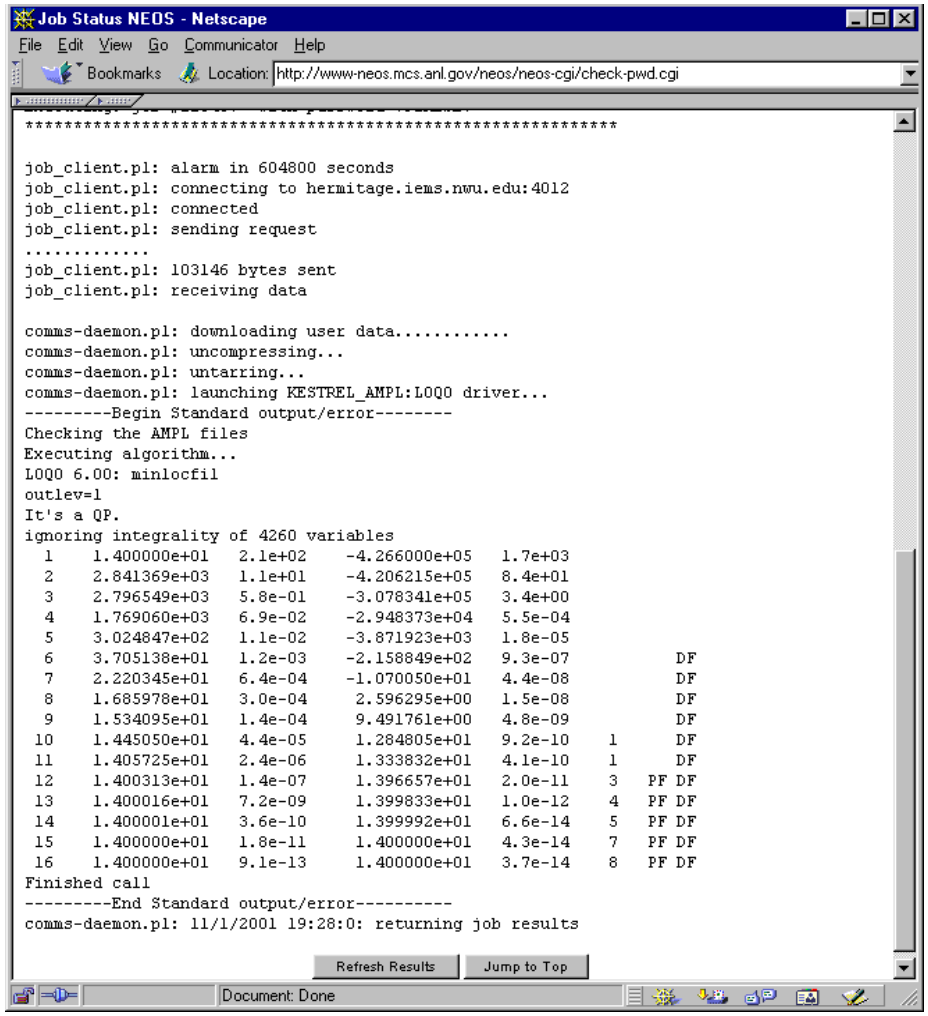
Try a Solver: Kestrel Interface

Intermediate status listing



Try a Solver: Kestrel Interface

Final result listing



NEOS Frequently Asked Questions

Who uses it?

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

What kinds of solvers does it offer?

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

How is it 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)*

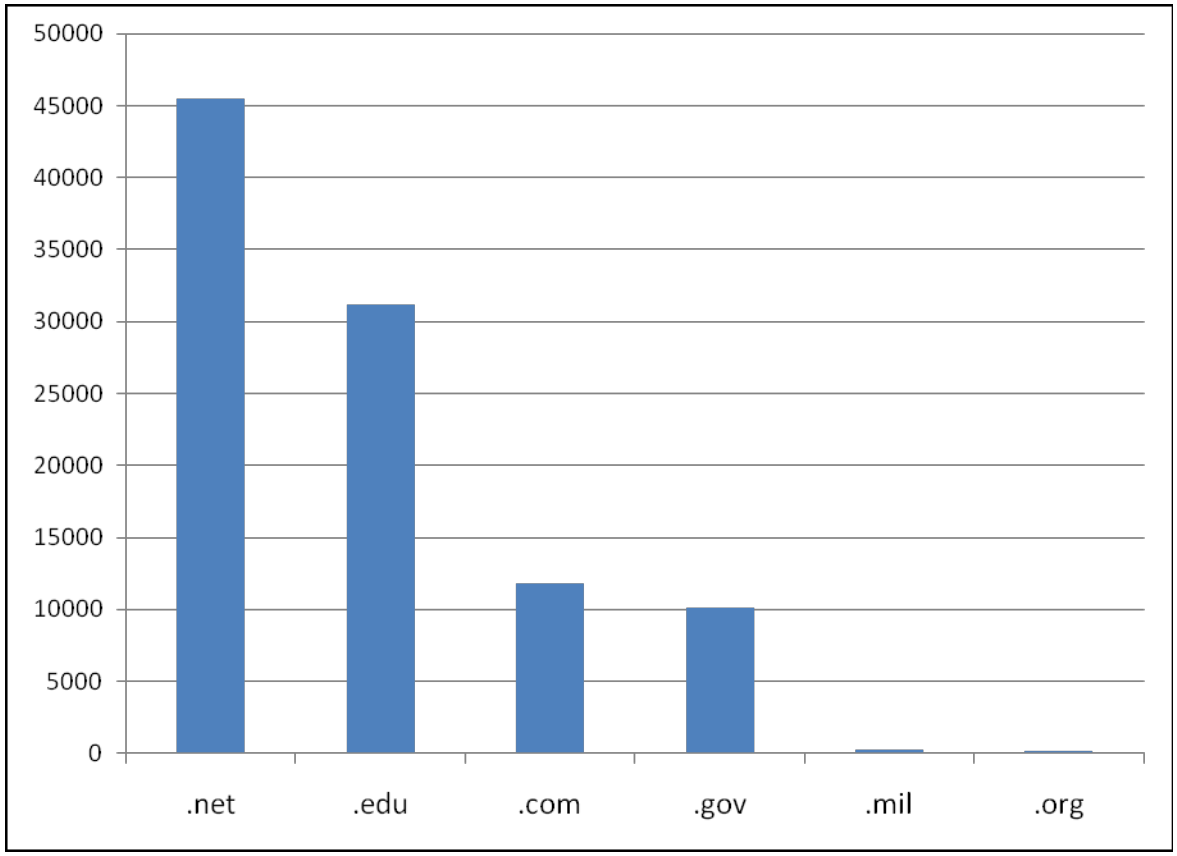
- I have been able to build and solve a prototype combinatorial auction MIP model using AMPL and NEOS in a fraction of the time it would have required me to do this had I needed to requisition a solver and install it locally.
- Our idea is trying to design antennas by using the computer. . . . We have 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.

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

NEOS Users

Where are They From?

Standard domains

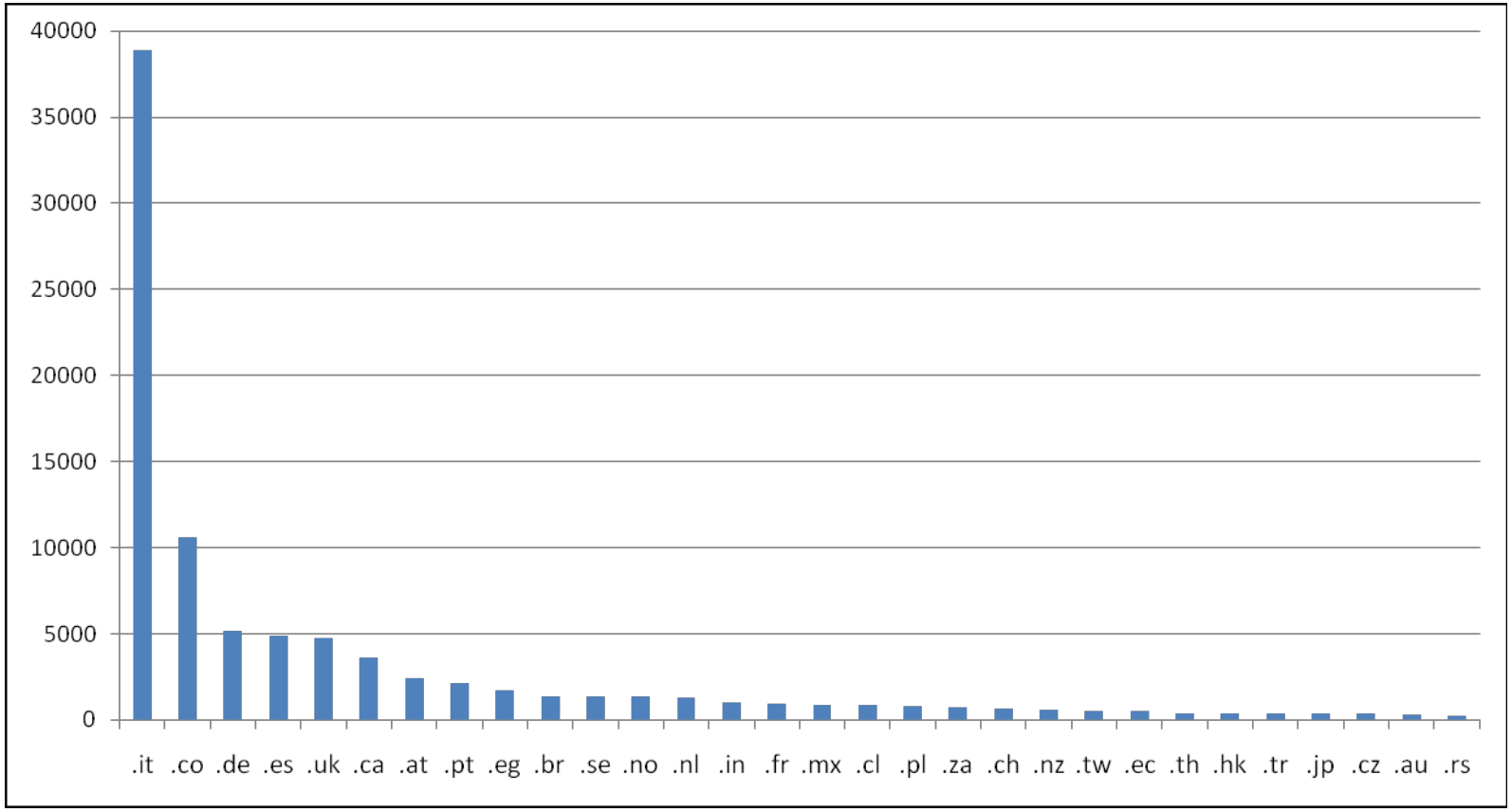


(2010 through October)

NEOS Users

Where are They From?

Country domains (< 40000)

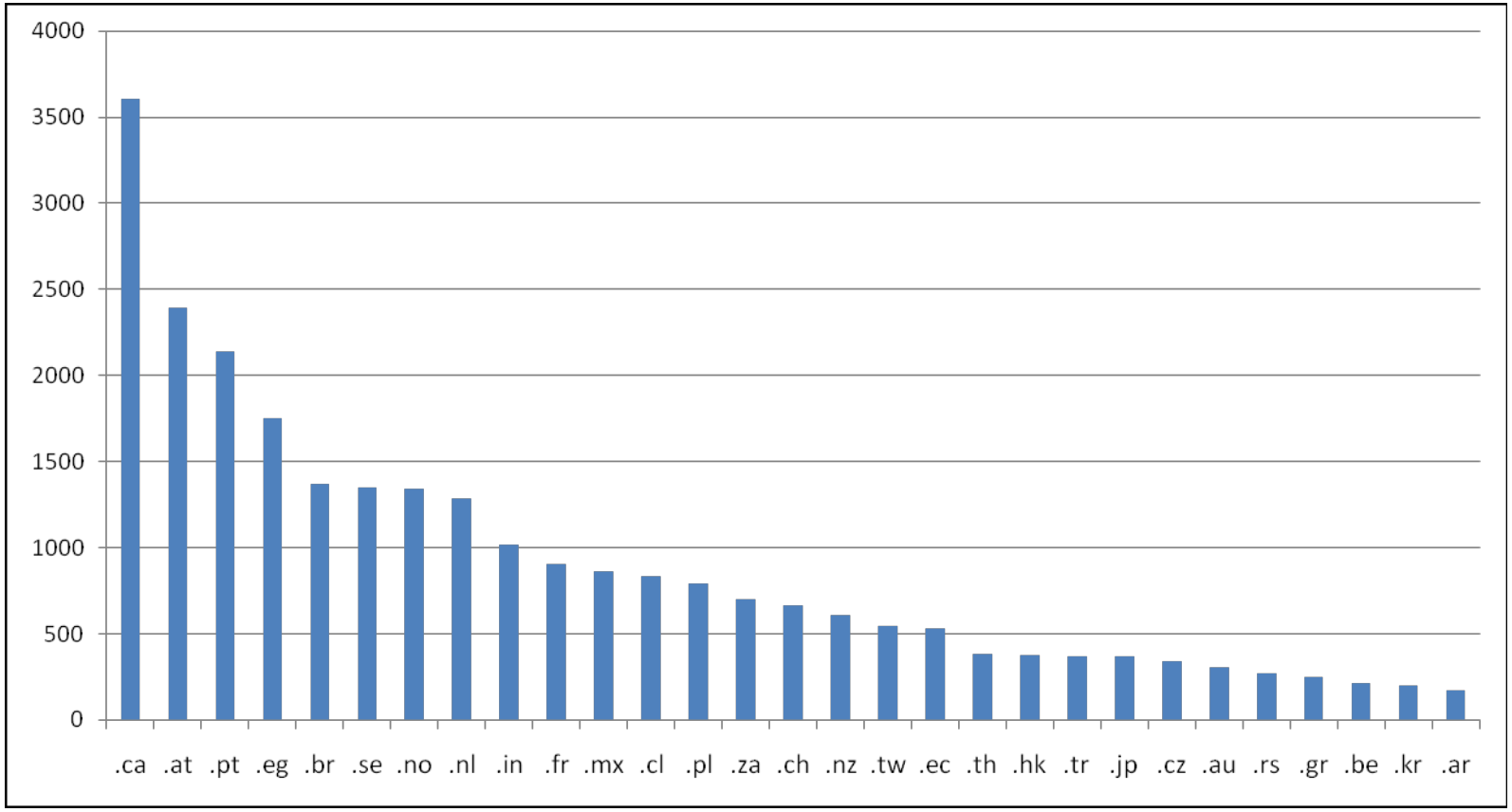


(2010 through October)

NEOS Users

Where are They From?

Country domains (< 4000)

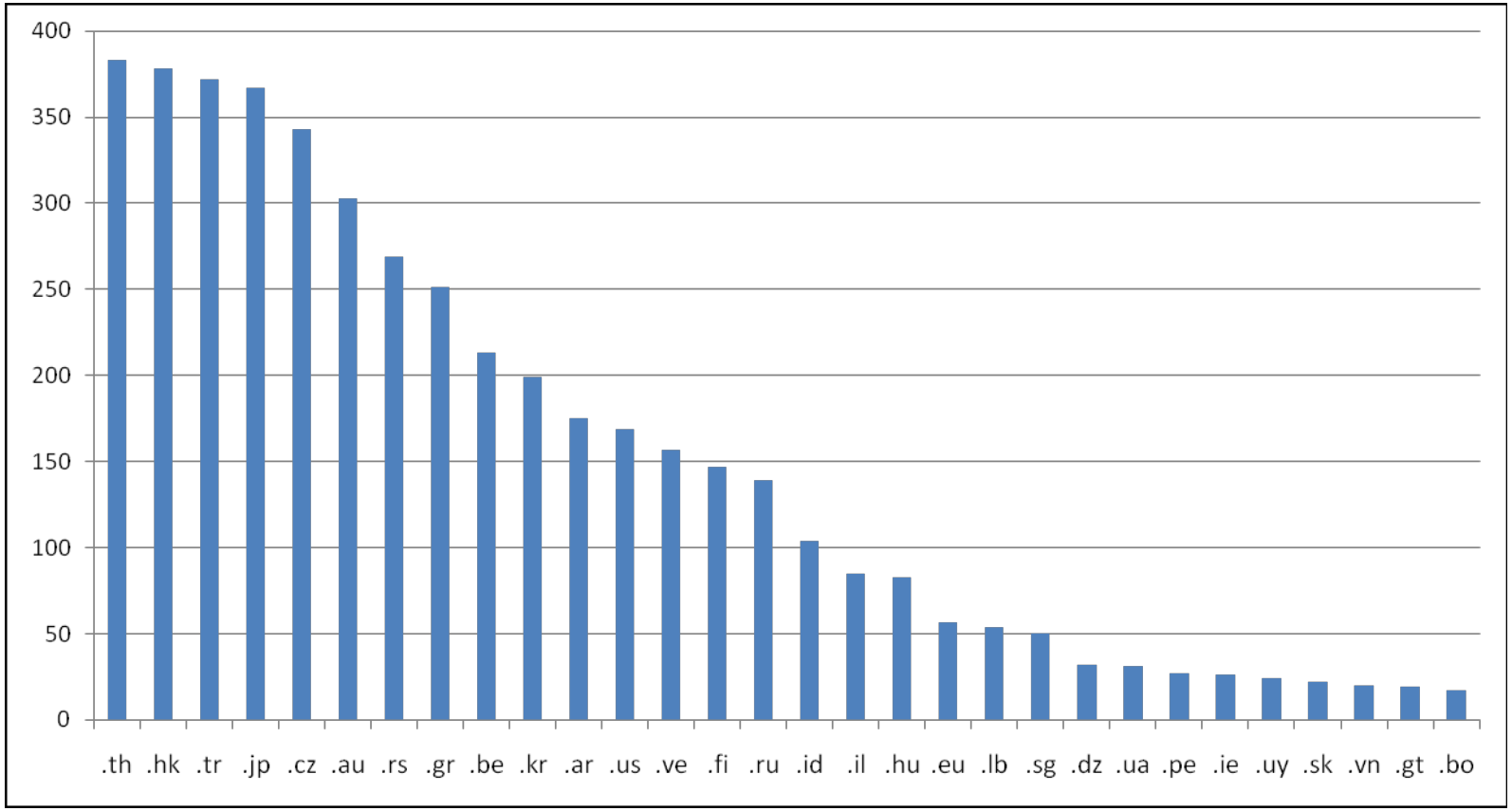


(2010 through October)

NEOS Users

Where are They From?

Country domains (< 400)

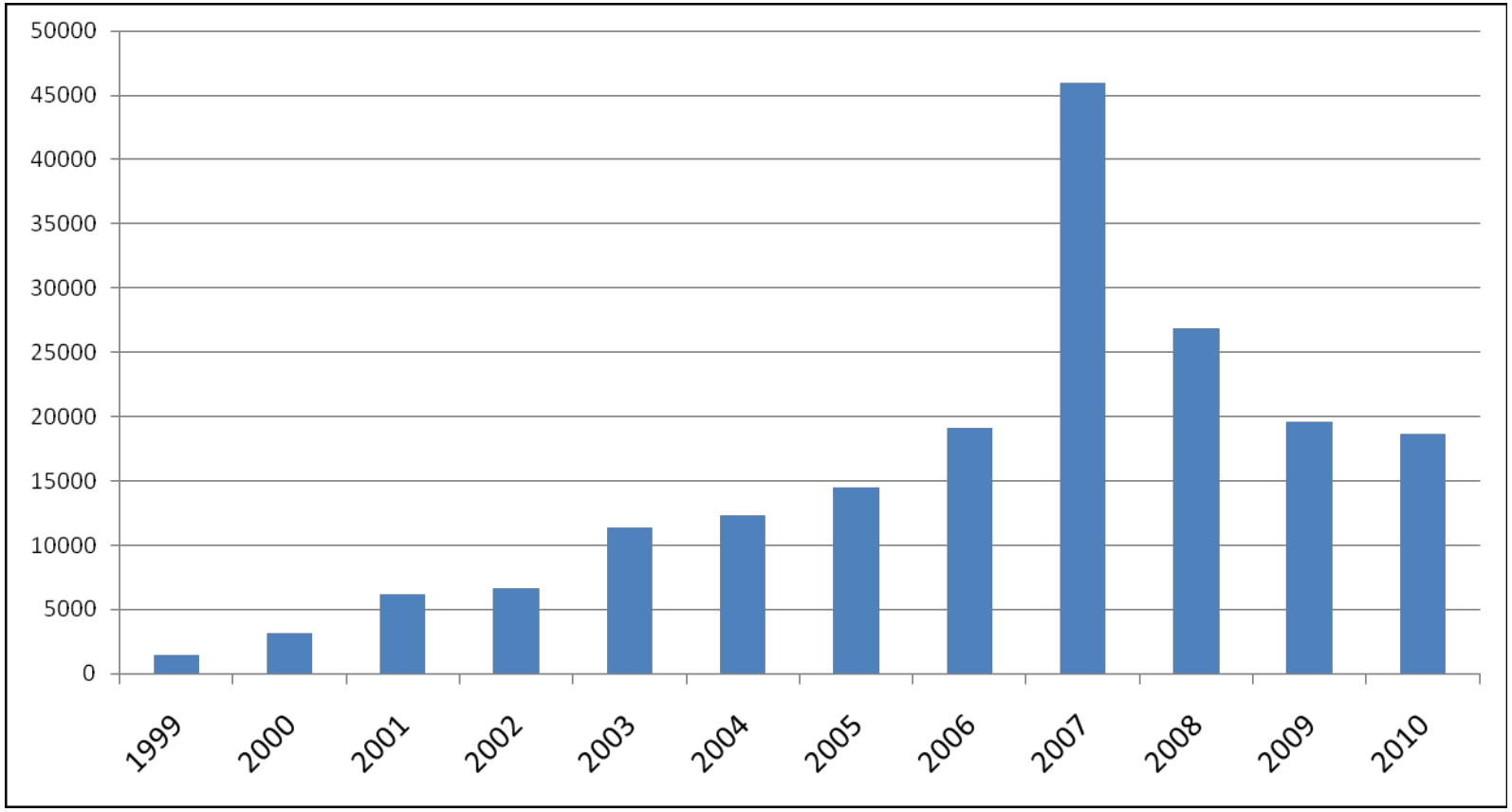


(2010 through October)

NEOS Users

How Much Do They Use It?

Monthly rates since 1999

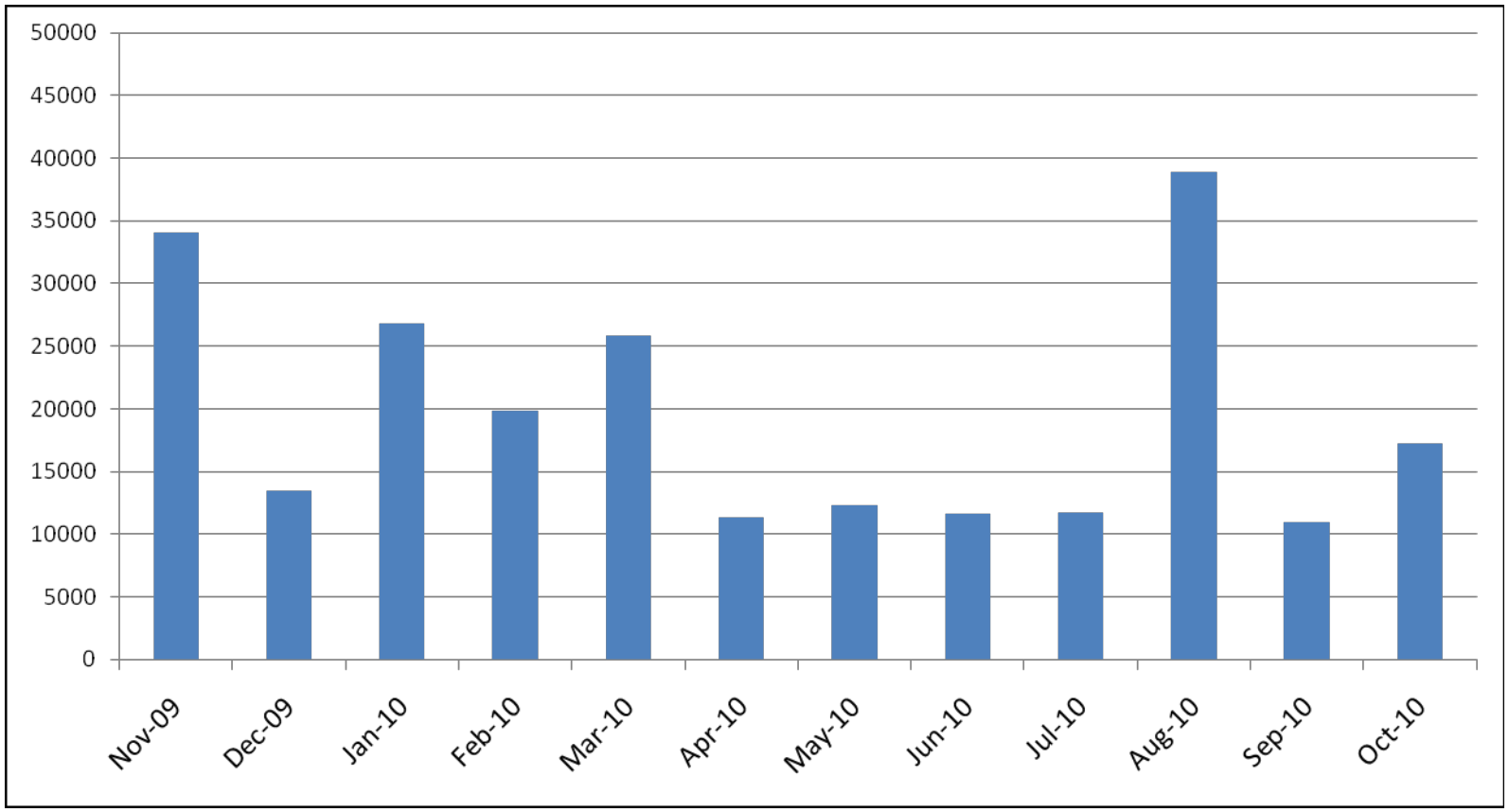


20000/month ≈ 25/hour

NEOS Users

How Much Do They Use It?

Monthly rates for past year



20000/month ≈ 25/hour

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

Who Supplies Them?

Some commercial solver vendors

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

Universities and their researchers

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

Open-Source Enthusiasts

- GLPK, CBC, Bonmin (mixed integer)

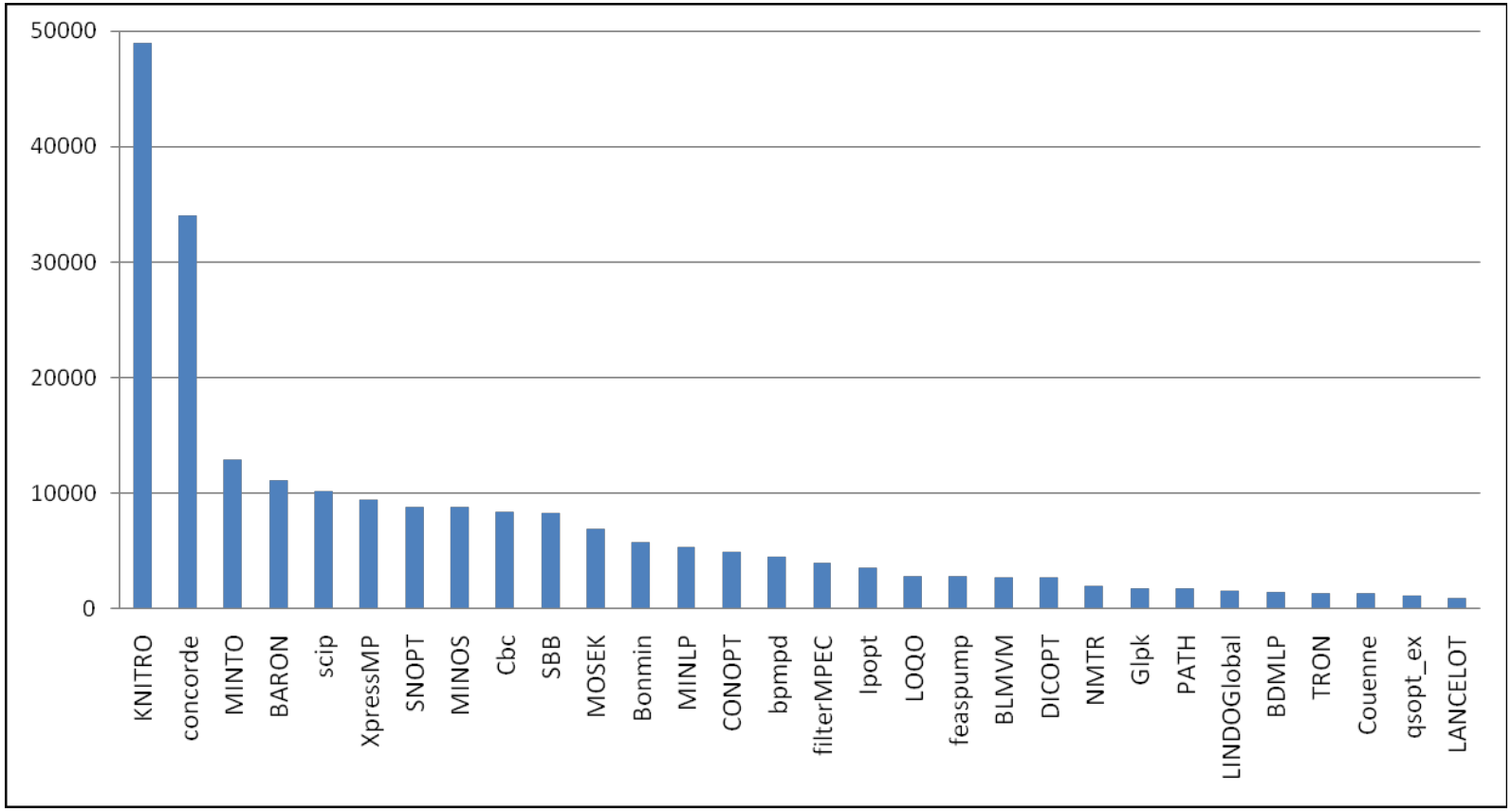
with thanks to . . .

- ***AMPL*** and ***GAMS*** developers
- ***Hans Mittelmann,*** Arizona State

NEOS Solvers

Which are Most Heavily Used?

Solver submissions (< 50000)

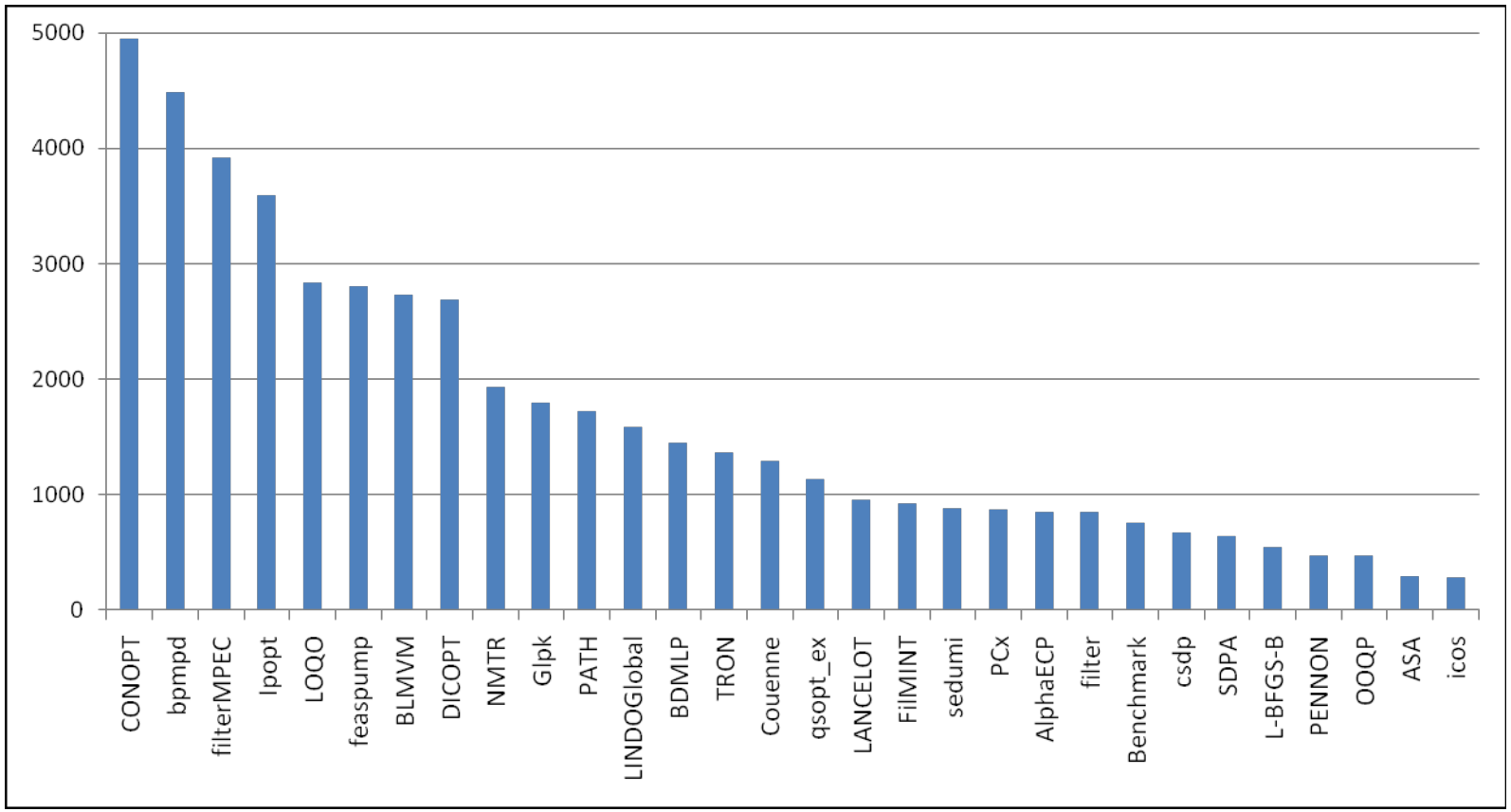


(2010 through October)

NEOS Solvers

Which are Most Heavily Used?

Solver submissions (< 5000)

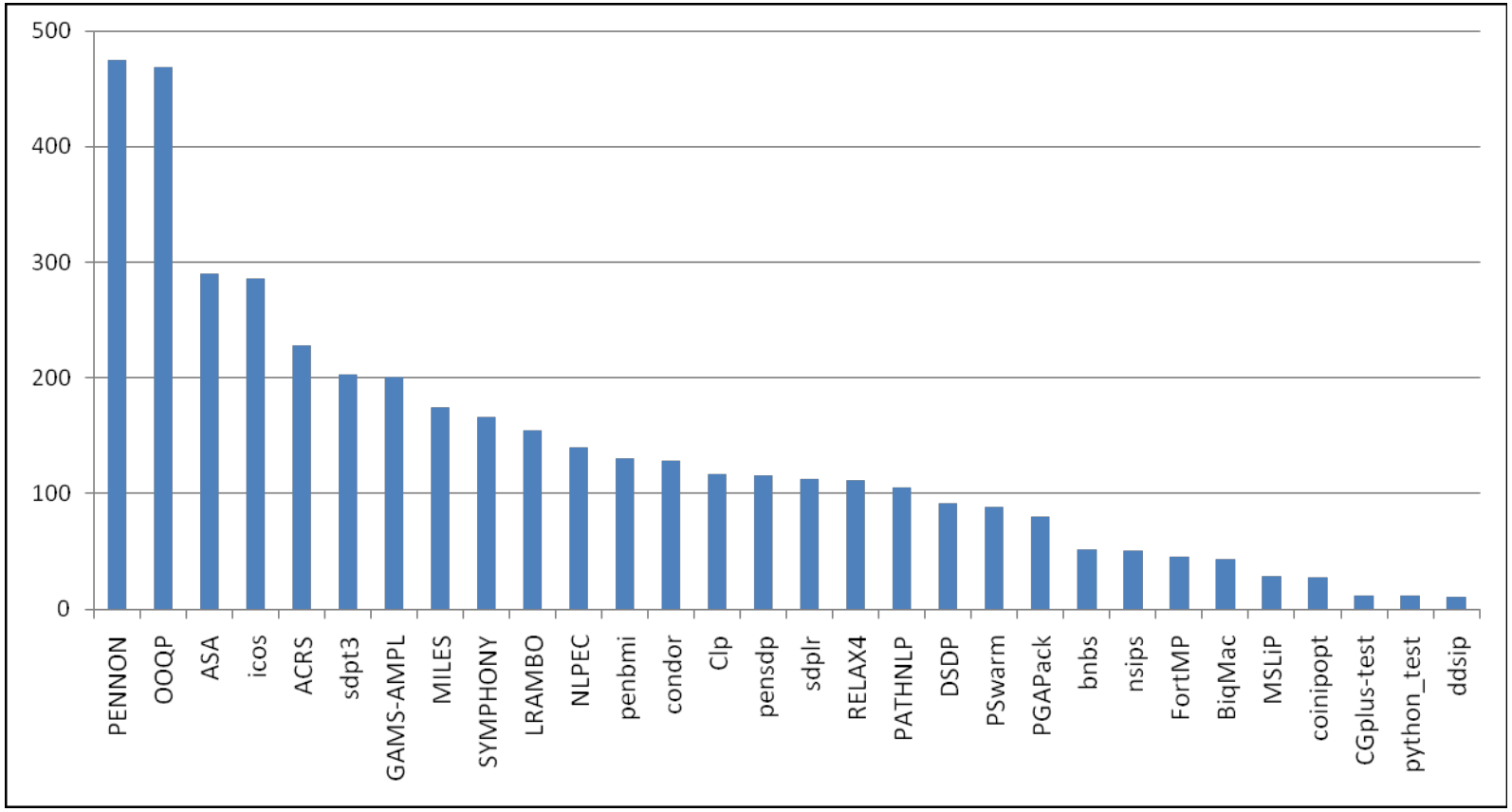


(2010 through October)

NEOS Solvers

Which are Most Heavily Used?

Solver submissions (< 500)



(2010 through October)

Where are They Hosted?

Varied workstations at

- Aachen University of Technology, Germany
- Argonne National Laboratory
- Arizona State University
- Lehigh University
- National Taiwan University
- Universidade do Minho, Portugal
- University of Wisconsin at Madison

. . . new hosts 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

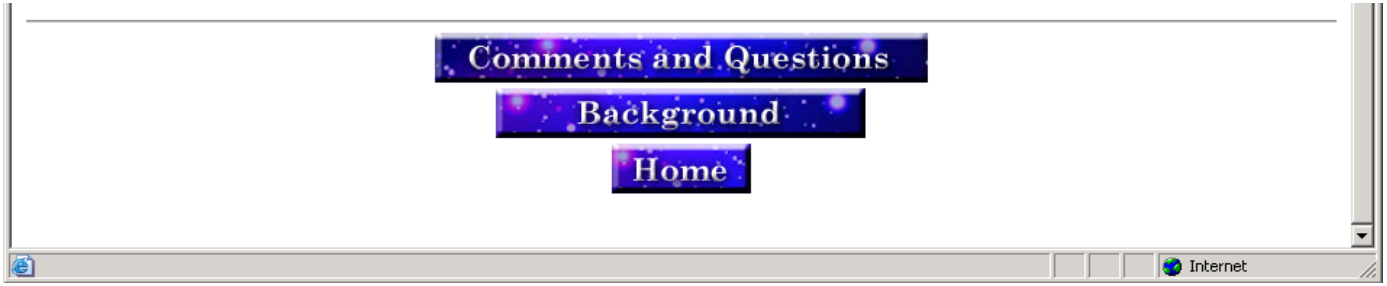
- Processor cycles
- Many people's time

Who Answers Users' Questions?

Large mailing list for support questions

- NEOS developers
- Solver developers

Support request buttons on every page



NEOS Limitations

Limited choices for MIP

- ***But now offers Gurobi solver***

Limited input standardization

- Some AMPL, some GAMS
- Varied low-level formats

Limited support

- Maintenance
- Computing power

Limited funding model

- Grants?
- User fees?

. . . recent move may change things!

To Learn More . . .

Websites

- **`www.neos-server.org`**

Overview

- Elizabeth D. Dolan, Robert Fourer, Jorge J. Moré, and Todd S. Munson, "Optimization on the NEOS Server." SIA*M News* **35:**6 (July/August 2002) 4, 8–9. `www.siam.org/pdf/news/457.pdf`

Modeling system interfaces

- Elizabeth D. Dolan, Robert Fourer, Jean-Pierre Goux, Todd S. Munson and Jason Sarich, "Kestrel: An Interface from Optimization Modeling Systems to the NEOS Server." *INFORMS Journal on Computing* **20** (2008) 525–538. `dx.doi.org/10.1287/ijoc.1080.0264`

COIN-OR www.coin-or.org

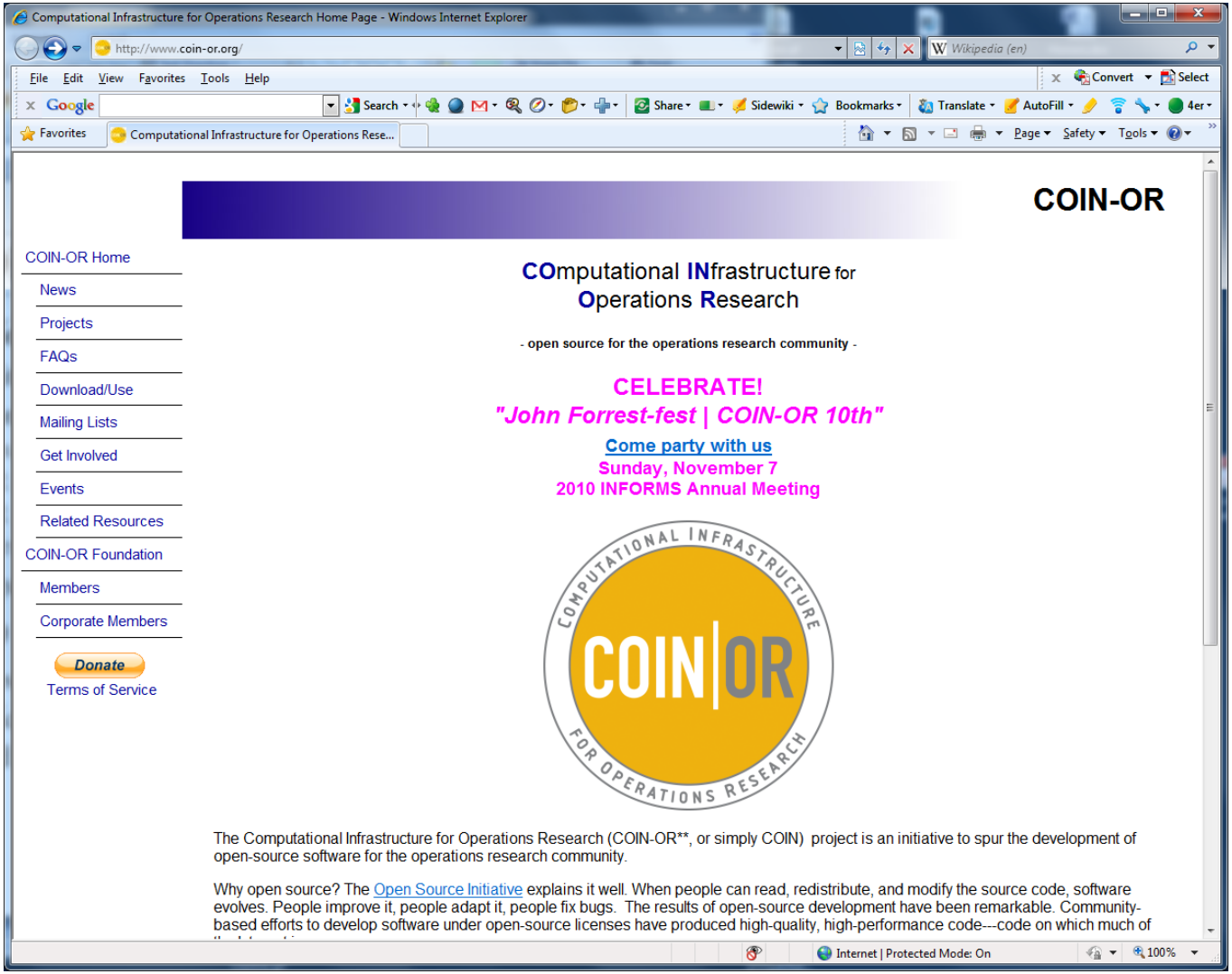
Computational Infrastructure for Operations Research

- Repository for open-source software for optimization
- Mission:
  - Develop, manage & distribute
  - OR software, models, and data so that
  - OR professionals can benefit from
  - peer-reviewed, archived, openly-disseminated software
- Strongest in optimization

Since 2000 . . .

- Origins at IBM
- Transferred to nonprofit COIN-OR Foundation
- Hosted by INFORMS

COIN-OR


Computational Infrastructure for Operations Research Home Page - Windows Internet Explorer
http://www.coin-or.org/
COIN-OR
COIN-OR Home
News
Projects
FAQs
Download/Use
Mailing Lists
Get Involved
Events
Related Resources
COIN-OR Foundation
Members
Corporate Members
Donate
Terms of Service
COmputational INfrastructure for Operations Research
- open source for the operations research community -
CELEBRATE!
"John Forrest-fest | COIN-OR 10th"
Come party with us
Sunday, November 7
2010 INFORMS Annual Meeting
COMPUTATIONAL INFRASTRUCTURE FOR OPERATIONS RESEARCH
COIN|OR
The Computational Infrastructure for Operations Research (COIN-OR**, or simply COIN) project is an initiative to spur the development of open-source software for the operations research community.
Why open source? The Open Source Initiative explains it well. When people can read, redistribute, and modify the source code, software evolves. People improve it, people adapt it, people fix bugs. The results of open-source development have been remarkable. Community-based efforts to develop software under open-source licenses have produced high-quality, high-performance code---code on which much of

Open-Source Software

Things to know

- Free, but subject to licensing restrictions
- Licenses vary considerably
- Equally available to all user classes
- Possibly owned (in part) by
  - Co-authors
  - Employer
  - Granting Agencies
  - Owner of the machines it was developed on

Examples

- GNU Public License, GNU Library Public License
- Common Public License, Eclipse Public License
- Mozilla Public License
- Apache License
- BSD Licenses

Open-Source Software at COIN-OR

Solvers

- Linear continuous & discrete
- Nonlinear continuous & discrete
- Semidefinite
- Stochastic

. . . source & binary

Infrastructures

- Developer tools
- Optimization utilities
- Interfaces
- Modeling systems and environments

Open-Source Software at COIN-OR