AMPL in the Cloud

Using Online Services to Develop and Deploy Optimization Applications through Algebraic Modeling

Robert Fourer, Filipe Brandão, Martin Laskowski

AMPL Optimization Inc.

{4er,fdabrandao,martin}@ampl.com

INFORMS Conference on Business Analytics and Operations Research

Baltimore, April 15-17, 2018

Track 10 — Technology Tutorials — Monday, 3:40-4:30 pm

AMPL in the Cloud

Using Online Services to Develop and Deploy Optimization Applications through Algebraic Modeling

Cloud services promising "optimization on demand" have become steadily more numerous and more powerful in recent years. This presentation offers a user-oriented survey, with a focus on the role of the AMPL modeling language in streamlining development and deployment of optimization models using online tools. Starting with the pioneering free NEOS Server, we compare more recent commercial offerings such as Gurobi Instant Cloud and the Satalia SolveEngine; the benefits of these solver services are enhanced through their use with AMPL's algebraic modeling facilities. We conclude by introducing QuanDec, which turns AMPL models into web-based collaborative decision-making tools.

Outline

AMPL Introduction

- Concept of a modeling language
- Example in the AMPL language

AMPL in the cloud

- Optimization on demand
  - NEOS Server
  - Satalia SolveEngine
  - IBM Decision Optimization on Cloud
- Optimization to order
  - Gurobi Instant Cloud
- Collaborative optimization
  - QuanDec

The Optimization Modeling Cycle

Steps

- Communicate with problem owner
- Build model
- Prepare data
- Generate optimization problem
- Submit problem to solver
  - Gurobi, Knitro, CPLEX, Xpress, CONOPT, MINOS, . . .
- Report & analyze results
- ***Repeat** until you get it right**!***

Goals for optimization software

- Do this quickly and reliably
- Get results before client loses interest
- Deploy for application

Optimization Modeling Languages

Two forms of an optimization problem

- Modeler's form
  - Mathematical description, easy for people to work with
- Solver's form
  - Explicit data structure, easy for solvers to compute with

Idea of a modeling language

- ***A computer-readable modeler's form***
  - You write optimization problems in a modeling language
  - Computers translate to algorithm's form for solution

Advantages of a modeling language

- Faster modeling cycles
- More reliable modeling
- More maintainable applications

Algebraic Modeling Languages

Formulation concept

- Define data in terms of sets & parameters
  - Analogous to database keys & records
- Define decision variables
- Minimize or maximize a function of decision variables
- Subject to equations or inequalities
  that constrain the values of the variables

Advantages

- Familiar
- Powerful
- Proven

AMPL

Features

- Algebraic modeling language
- Built specially for optimization
- Designed to support many solvers

Design goals

- Powerful, general expressions
- Natural, easy-to-learn modeling principles
- Efficient processing that scales well with problem size

Many ways to use . . .

Many Ways to Use AMPL

Command language

- Browse results & debug model interactively
- Make changes and re-run

Scripting language

- Bring the programmer to the modeling language

Programming interface (API)

- Bring the modeling language to the programmer

Deployment tool

- Turn models into collaboration environments

Example (1st cycle)

Multicommodity transportation . . .

- Products available at factories
- Products needed at stores
- Plan shipments at lowest cost

Mathematical Formulation

Given

O Set of origins (factories)

D Set of destinations (stores)

P Set of products

and

a_{ip} Amount available, for each $i \in O$ and $p \in P$

b_{jp} Amount required, for each $j \in D$ and $p \in P$

l_{ij} Limit on total shipments, for each $i \in O$ and $j \in D$

c_{ijp} Shipping cost per unit, for each $i \in O, j \in D, p \in P$

Multicommodity Transportation

Mathematical Formulation

Determine

X_{ijp} Amount of each product $p \in P$
to be shipped from origin $i \in O$ to destination $j \in D$

to minimize

$\sum_{i \in O} \sum_{j \in D} \sum_{p \in P} c_{ijp} X_{ijp}$

Total shipping cost

Mathematical Formulation

Subject to

$\sum_{j \in D} X_{ijp} \leq a_{ip}$ for all $i \in O, p \in P$

Total shipments of product p out of origin i must not exceed availability

$\sum_{i \in O} X_{ijp} = b_{jp}$ for all $j \in D, p \in P$

Total shipments of product p into destination j must satisfy requirements

$\sum_{p \in P} X_{ijp} \leq l_{ij}$ for all $i \in O, j \in D$

Total shipments from origin i to destination j must not exceed the limit

Multicommodity Transportation

AMPL Formulation

Symbolic data

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

param supply {ORIG,PROD} >= 0;  # availabilities at origins
param demand {DEST,PROD} >= 0;  # requirements at destinations
param limit {ORIG,DEST} >= 0;   # capacities of links

param cost {ORIG,DEST,PROD} >= 0;  # shipment cost
```

AMPL Formulation

Symbolic model: variables and objective

```
var Trans {ORIG,DEST,PROD} >= 0;   # actual units to be shipped

minimize Total_Cost:
   sum {i in ORIG, j in DEST, p in PROD} cost[i,j,p] * Trans[i,j,p];
```

$$\sum_{i \in O} \sum_{j \in D} \sum_{p \in P} c_{ijp} \, X_{ijp}$$

AMPL Formulation

Symbolic model: constraint

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

$$\sum_{j \in D} X_{ijp} \leq a_{ip}, \text{ for all } i \in O, p \in P$$

AMPL Formulation

Symbolic model: constraints

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

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

subject to Multi {i in ORIG, j in DEST}:
   sum {p in PROD} Trans[i,j,p] <= limit[i,j];
```

Multicommodity Transportation

AMPL Formulation

Explicit data independent of symbolic model

```
set ORIG := GARY CLEV PITT ;
set DEST := FRA DET LAN WIN STL FRE LAF ;
set PROD := bands coils plate ;

param supply (tr):  GARY   CLEV   PITT :=
            bands    400    700    800
            coils    800   1600   1800
            plate    200    300    300 ;

param demand (tr):
            FRA   DET   LAN   WIN   STL   FRE   LAF :=
   bands    300   300   100    75   650   225   250
   coils    500   750   400   250   950   850   500
   plate    100   100     0    50   200   100   250 ;

param limit default 625 ;
```

AMPL Formulation

Explicit data (continued)

```
param cost :=

 [*,*,bands]:  FRA  DET  LAN  WIN  STL  FRE  LAF :=
        GARY    30   10    8   10   11   71    6
        CLEV    22    7   10    7   21   82   13
        PITT    19   11   12   10   25   83   15

 [*,*,coils]:  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

 [*,*,plate]:  FRA  DET  LAN  WIN  STL  FRE  LAF :=
        GARY    41   15   12   16   17   86    8
        CLEV    29    9   13    9   28   99   18
        PITT    26   14   17   13   31  104   20 ;
```

AMPL Solution

Model + data = problem instance to be solved

```
ampl: model multi.mod;
ampl: data multi.dat;
ampl: option solver minos;

ampl: solve;
MINOS 5.51: optimal solution found.
53 iterations, objective 199500

ampl: display Trans;
Trans [CLEV,*,*] (tr)
:        DET   FRA   FRE   LAF   LAN   STL  WIN  :=
bands      0   225     0   150     0   250   75
coils    525     0   225    75   400   300   75
plate    100    50   100     0     0     0   50
...
```

AMPL Solution

Solver choice independent of model and data

```
ampl: model multi.mod;
ampl: data multi.dat;
ampl: option solver snopt;

ampl: solve;
SNOPT 7.5-1.2 : Optimal solution found.
60 iterations, objective 199500

ampl: display Trans;
Trans [CLEV,*,*] (tr)
:        DET   FRA   FRE   LAF   LAN   STL  WIN  :=
bands    150   225     0     0     0   250   75
coils    375     0   225   225   400   300   75
plate    100    50   100     0     0     0   50
...
```

Example (4th cycle)

Multicommodity transportation . . .

- Products available at factories
- Products needed at stores
- Plan shipments at lowest cost

. . . revised in light of practical considerations

- Using an origin-destination pair incurs some fixed cost
- Shipments cannot be too small
- Factories cannot serve too many stores

Multicommodity Transportation

Mathematical Formulation

Given

O Set of origins (factories)

D Set of destinations (stores)

P Set of products

and

a_{ip} Amount available, for each $i \in O$ and $p \in P$

b_{jp} Amount required, for each $j \in D$ and $p \in P$

l_{ij} Limit on total shipments, for each $i \in O$ and $j \in D$

c_{ijp} Shipping cost per unit, for each $i \in O, j \in D, p \in P$

d_{ij} Fixed cost for shipping any amount from $i \in O$ to $j \in D$

s Minimum total size of any shipment

n Maximum number of destinations served by any origin

Multicommodity Transportation

Mathematical Formulation

Determine

X_{ijp} Amount of each $p \in P$ to be shipped from $i \in O$ to $j \in D$

Y_{ij} 1 if any product is shipped from $i \in O$ to $j \in D$
0 otherwise

to minimize

$$\sum_{i \in O} \sum_{j \in D} \sum_{p \in P} c_{ijp} X_{ijp} + \sum_{i \in O} \sum_{j \in D} d_{ij} Y_{ij}$$

Total variable cost plus total fixed cost

Mathematical Formulation

Subject to

$\sum_{j \in D} X_{ijp} \leq a_{ip}$ for all $i \in O, p \in P$

Total shipments of product p out of origin i must not exceed availability

$\sum_{i \in O} X_{ijp} = b_{jp}$ for all $j \in D, p \in P$

Total shipments of product p into destination j must satisfy requirements

$\sum_{p \in P} X_{ijp} \leq l_{ij} Y_{ij}$ for all $i \in O, j \in D$

When there are shipments from origin i to destination j, the total may not exceed the limit, and Y_{ij} must be 1

Mathematical Formulation

Subject to

$\sum_{p \in P} X_{ijp} \geq s Y_{ij}$ for all $i \in O, j \in D$

When there are shipments from origin i to destination j, the total amount of shipments must be at least s

$\sum_{j \in D} Y_{ij} \leq n$ for all $i \in O$

Number of destinations served by origin i must be as most n

Multicommodity Transportation

AMPL Formulation

Symbolic data

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

param supply {ORIG,PROD} >= 0;  # availabilities at origins
param demand {DEST,PROD} >= 0;  # requirements at destinations
param limit {ORIG,DEST} >= 0;   # capacities of links

param vcost {ORIG,DEST,PROD} >= 0;  # variable shipment cost
param fcost {ORIG,DEST} > 0;        # fixed usage cost

param minload >= 0;                 # minimum shipment size
param maxserve integer > 0;         # maximum destinations served
```

Multicommodity Transportation

AMPL Formulation

Symbolic model: variables and objective

```
var Trans {ORIG,DEST,PROD} >= 0;    # actual units to be shipped
var Use {ORIG, DEST} binary;        # 1 if link used, 0 otherwise

minimize Total_Cost:
   sum {i in ORIG, j in DEST, p in PROD} vcost[i,j,p] * Trans[i,j,p]
 + sum {i in ORIG, j in DEST} fcost[i,j] * Use[i,j];
```

$$\sum_{i \in O} \sum_{j \in D} \sum_{p \in P} c_{ijp} X_{ijp} + \sum_{i \in O} \sum_{j \in D} d_{ij} Y_{ij}$$

AMPL Formulation

Symbolic model: constraints

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

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

subject to Multi {i in ORIG, j in DEST}:
   sum {p in PROD} Trans[i,j,p] <= limit[i,j] * Use[i,j];

subject to Min_Ship {i in ORIG, j in DEST}:
   sum {p in PROD} Trans[i,j,p] >= minload * Use[i,j];

subject to Max_Serve {i in ORIG}:
   sum {j in DEST} Use[i,j] <= maxserve;
```

Multicommodity Transportation

AMPL Formulation

Explicit data independent of symbolic model

```
set ORIG := GARY CLEV PITT ;
set DEST := FRA DET LAN WIN STL FRE LAF ;
set PROD := bands coils plate ;

param supply (tr):  GARY   CLEV   PITT :=
            bands    400    700    800
            coils    800   1600   1800
            plate    200    300    300 ;

param demand (tr):
            FRA   DET   LAN   WIN   STL   FRE   LAF :=
   bands    300   300   100    75   650   225   250
   coils    500   750   400   250   950   850   500
   plate    100   100     0    50   200   100   250 ;

param limit default 625 ;

param minload := 375 ;
param maxserve := 5 ;
```

AMPL Formulation

Explicit data (continued)

```
param vcost :=

 [*,*,bands]:  FRA  DET  LAN  WIN  STL  FRE  LAF :=
        GARY    30   10    8   10   11   71    6
        CLEV    22    7   10    7   21   82   13
        PITT    19   11   12   10   25   83   15

 [*,*,coils]:  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

 [*,*,plate]:  FRA  DET  LAN  WIN  STL  FRE  LAF :=
        GARY    41   15   12   16   17   86    8
        CLEV    29    9   13    9   28   99   18
        PITT    26   14   17   13   31  104   20 ;

param fcost:   FRA  DET  LAN  WIN  STL  FRE  LAF :=
        GARY  3000 1200 1200 1200 2500 3500 2500
        CLEV  2000 1000 1500 1200 2500 3000 2200
        PITT  2000 1200 1500 1500 2500 3500 2200 ;
```

AMPL Solution

Model + data = problem instance to be solved

```
ampl: model multmip3.mod;
ampl: data multmip3.dat;
ampl: option solver gurobi;
ampl: solve;
Gurobi 7.0.0: optimal solution; objective 235625
332 simplex iterations
23 branch-and-cut nodes
ampl: display Use;
Use [*,*]
:    DET FRA FRE LAF LAN STL WIN  :=
CLEV   1   1   1   0   1   1   0
GARY   0   0   0   1   0   1   1
PITT   1   1   1   1   0   1   0
;
```

AMPL Solution

Solver choice independent of model and data

```
ampl: model multmip3.mod;
ampl: data multmip3.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 12.7.0.0: optimal integer solution; objective 235625
135 MIP simplex iterations
0 branch-and-bound nodes
ampl: display Use;
Use [*,*]
:    DET FRA FRE LAF LAN STL WIN  :=
CLEV   1   1   1   0   1   1   0
GARY   0   0   0   1   0   1   1
PITT   1   1   1   1   0   1   0
;
```

AMPL Solution

Solver choice independent of model and data

```
ampl: model multmip3.mod;
ampl: data multmip3.dat;

ampl: option solver xpress;

ampl: solve;

XPRESS 29.01: Global search complete
Best integer solution found 235625
4 integer solutions have been found, 7 branch and bound nodes

ampl: display Use;

Use [*,*]

:    DET FRA FRE LAF LAN STL WIN  :=
CLEV   1   1   1   0   1   1   0
GARY   0   0   0   1   0   1   1
PITT   1   1   1   1   0   1   0
;
```

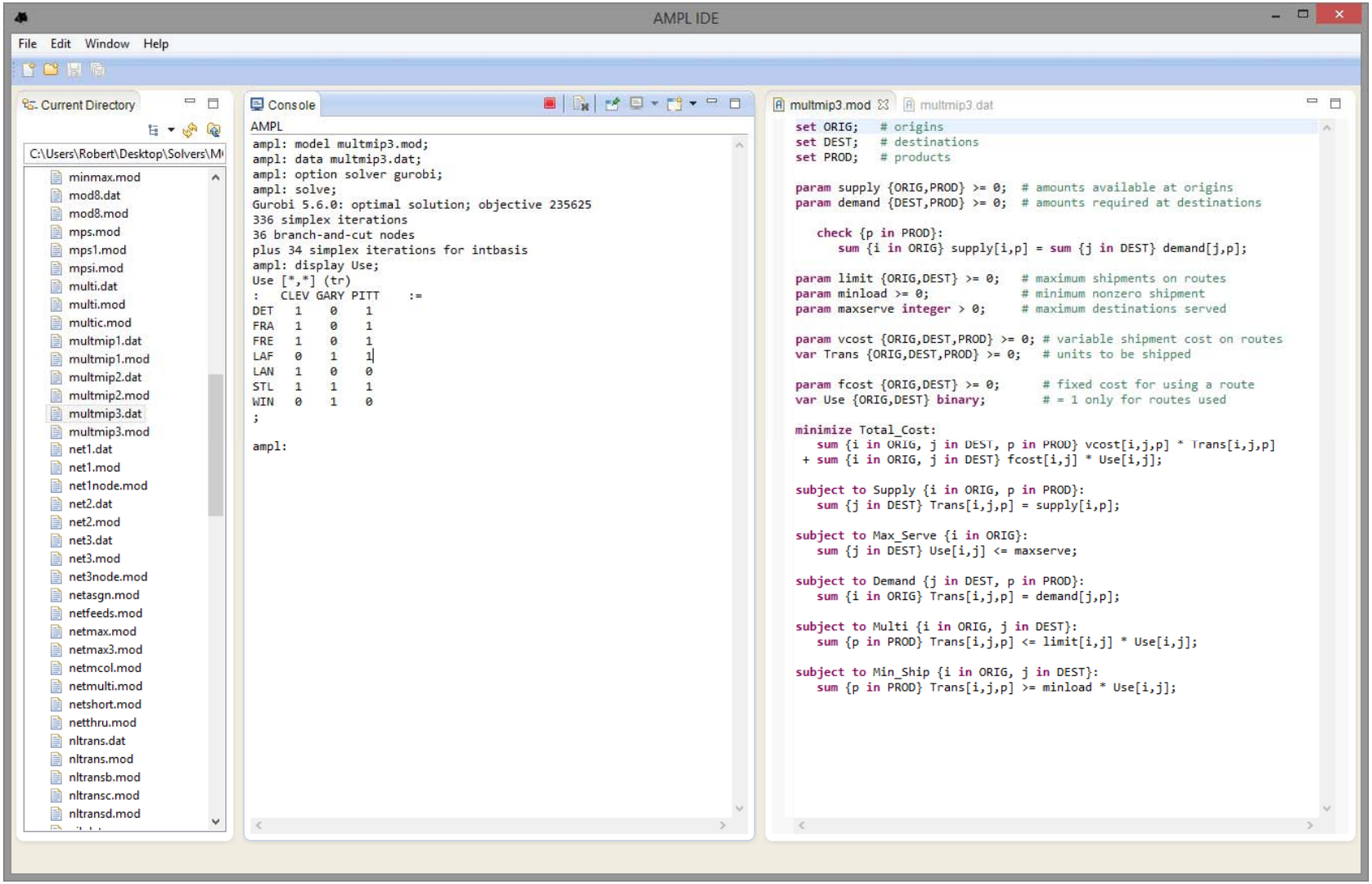
AMPL Solution

Examine results

```
ampl: display {i in ORIG, j in DEST}
ampl?    sum {p in PROD} Trans[i,j,p] / limit[i,j];
:       DET     FRA     FRE     LAF     LAN     STL     WIN     :=
CLEV    1       0.6     0.88    0       0.8     0.88    0
GARY    0       0       0       0.64    0       1       0.6
PITT    0.84    0.84    1       0.96    0       1       0
;

ampl: display Max_Serve.body;
CLEV  5
GARY  3
PITT  5
;

ampl: display TotalCost,
ampl?    sum {i in ORIG, j in DEST} fcost[i,j] * Use[i,j];
TotalCost = 235625
sum {i in ORIG, j in DEST} fcost[i,j]*Use[i,j] = 27600
```

Multicommodity Transportation

AMPL IDE



Scripting

Extend modeling language syntax . . .

- Algebraic expressions
- Set indexing expressions
- Interactive commands

. . . with programming concepts

- Loops of various kinds
- If-then and If-then-else conditionals
- Assignments

APIs

Run AMPL from your programs

- Transfer data to and from AMPL
- Execute AMPL commands within your application
  - Access all available solvers

Use popular programming languages

- C++, C#, Java, MATLAB, Python, R
  - Use Python dictionaries, R dataframes, MATLAB matrices, etc.

AMPL

Optimization in the Cloud

Optimization on demand

- NEOS Server
- Satalia SolveEngine
- IBM Decision Optimization on Cloud

Optimization to order

- Gurobi Instant Cloud

Collaborative optimization

- QuanDec

NEOS Server www.neos-server.org

Network Enabled Optimization System

- Originated 1995 at Argonne National Laboratory and Northwestern University
  - U.S. Department of Energy
  - National Science Foundation
- Since 2011 at University of Wisconsin, Madison
  - Wisconsin Institutes for Discovery

Free "optimization on demand"

- Over 40 solvers
- Several optimization modeling languages

NEOS Server

Architecture

Distributed workstations

- Offer varied inputs & solvers
- Process submissions on demand
- Contributed by varied organizations

Central scheduler

- Receives and queues submissions
- Sends submissions to appropriate workstations
- Returns results

Minimal hands-on management

- *Distributed:* Install NEOS software on workstations
- *Central:* Update server database of workstation locations and abilities

Original Facilities

Local submission clients

- Email
- Website
- NEOS submission tool

Problem description formats

- Linear: MPS and other solver files
- Nonlinear: Fortran or C programs
  - automatic differentiation of programs

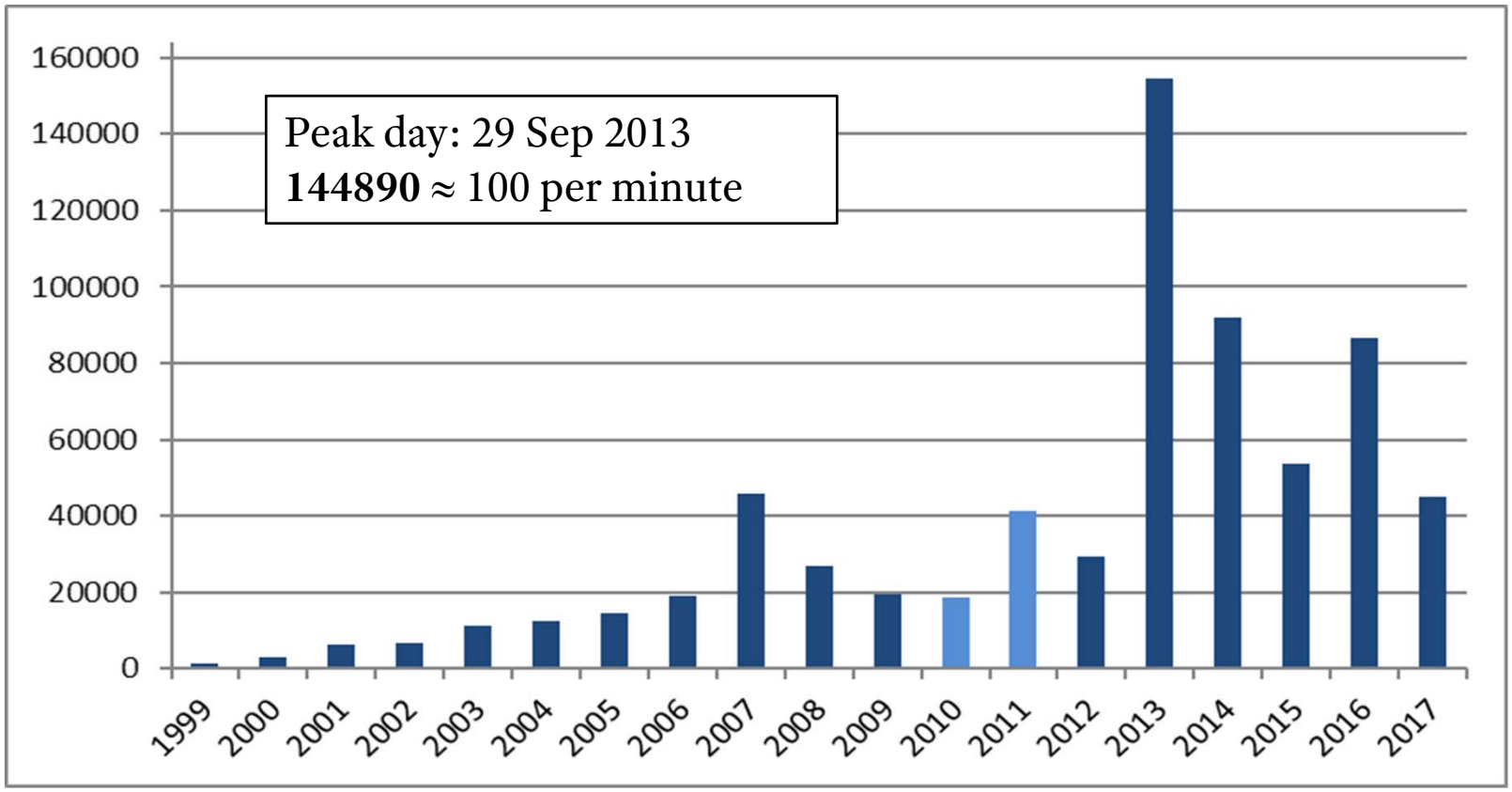
W. Gropp and J.J. Moré, 1997. **Optimization Environments and the NEOS Server**. *Approximation Theory and Optimization,* M. D. Buhmann and A. Iserles, eds., Cambridge University Press, 167-182.

J. Czyzyk, M.P. Mesnier and J.J. Moré, 1998. **The NEOS Server**. *IEEE Journal on Computational Science and Engineering* **5**(3), 68-75.

NEOS Server

Impact: Total Submissions

Monthly rates since 1999

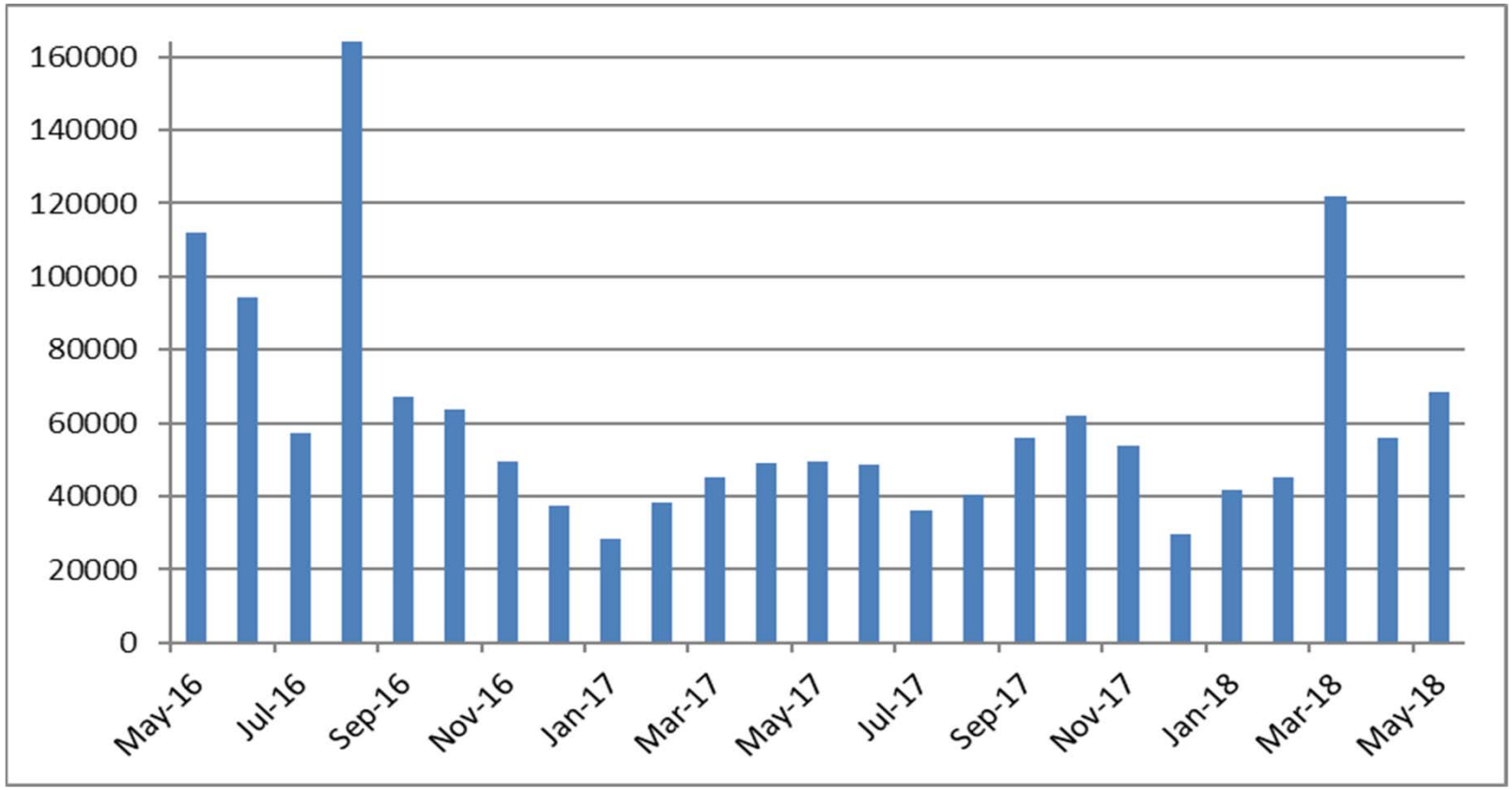


45000/month ≈ one per minute

NEOS Server

Impact: Recent Submissions

Monthly rates for past two years



45000/month ≈ one per minute

Assessment

Strengths

- Free
- Choice of solvers
  - Every popular solver available
- Easy to use
  - No account setup
  - No advance scheduling

Weaknesses

- Stand-alone focus: submission of "solve jobs"
- Non-profit management
  - Limited support & development
  - No guarantee of confidentiality
  - No guarantee of performance

Modeling Languages in NEOS

Modeling language inputs

- AMPL model, data, commands files
- GAMS model, options, gdx files

Modeling language operation

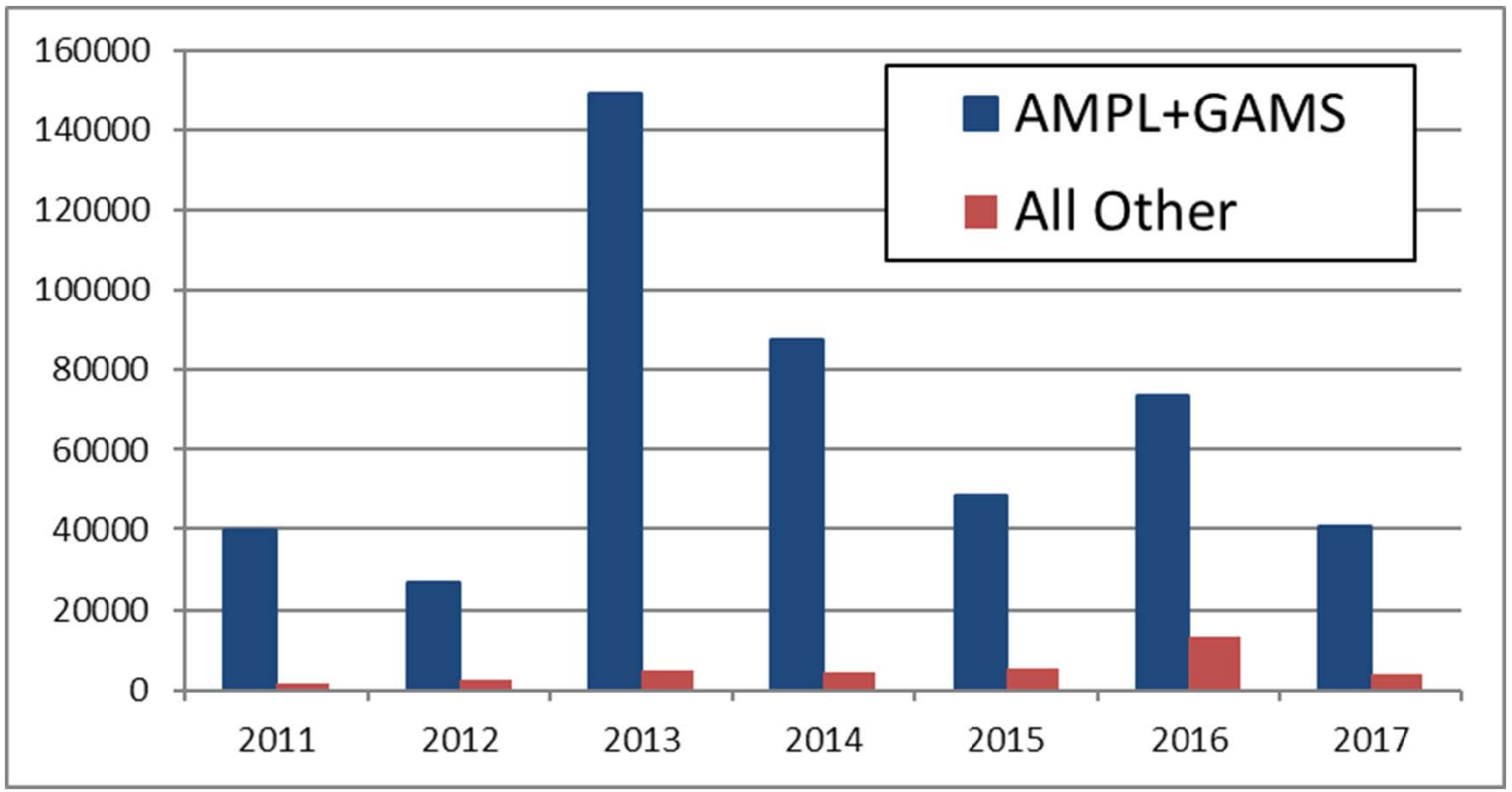
- User chooses a solver and a language
- NEOS scheduler finds a compatible workstation
- NEOS workstation invokes modeling language system with given inputs
- Modeling language system invokes solver

E.D. Dolan, R. Fourer, J.J. Moré and T.S. Munson,
Optimization on the NEOS Server. *SIAM News* **35:**6
(July/August 2002) 4, 8–9. www.siam.org/pdf/news/457.pdf

NEOS Server

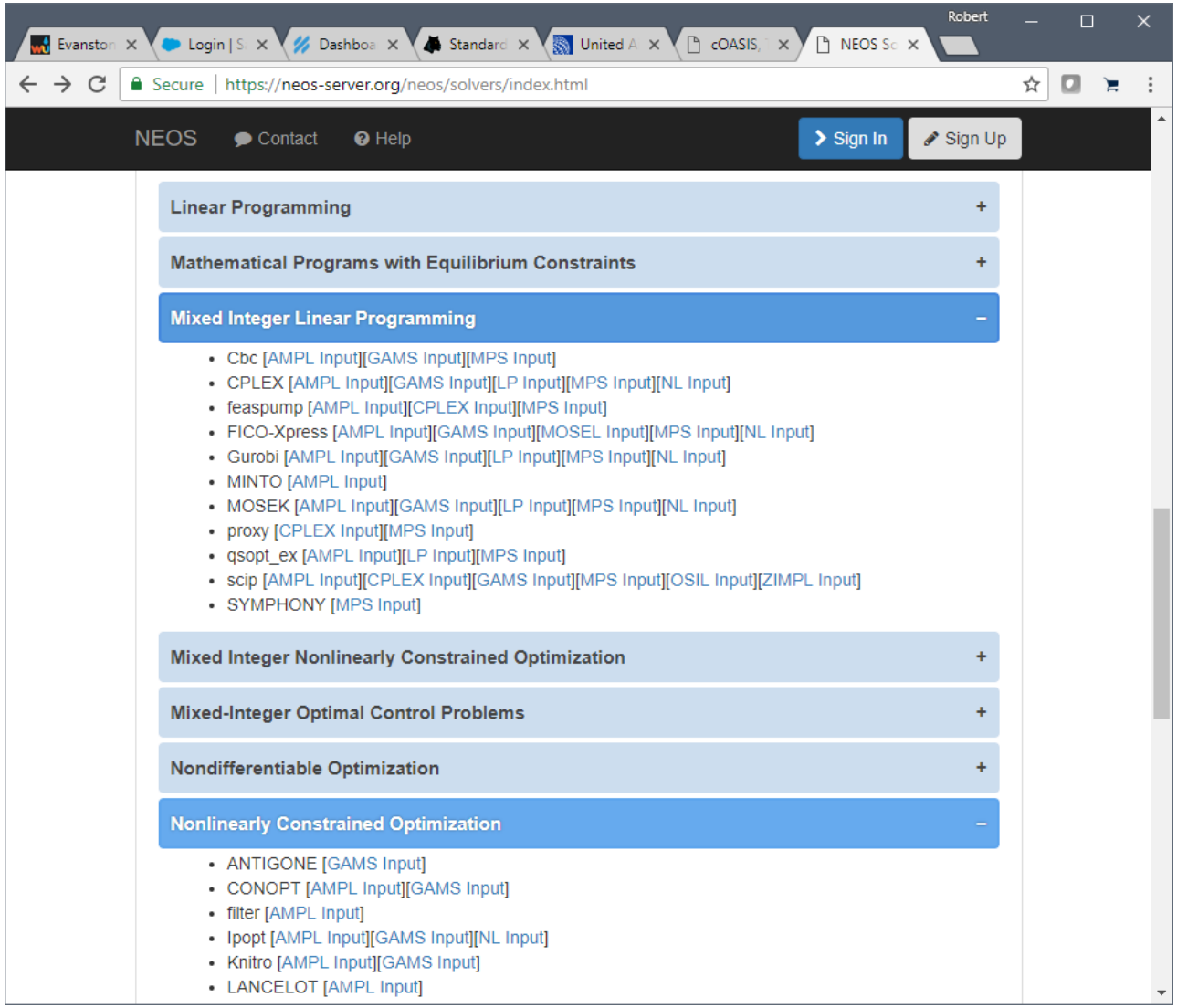
Impact: Modeling Languages

Monthly rates since 2011



NEOS Server

Solver & Language Listing

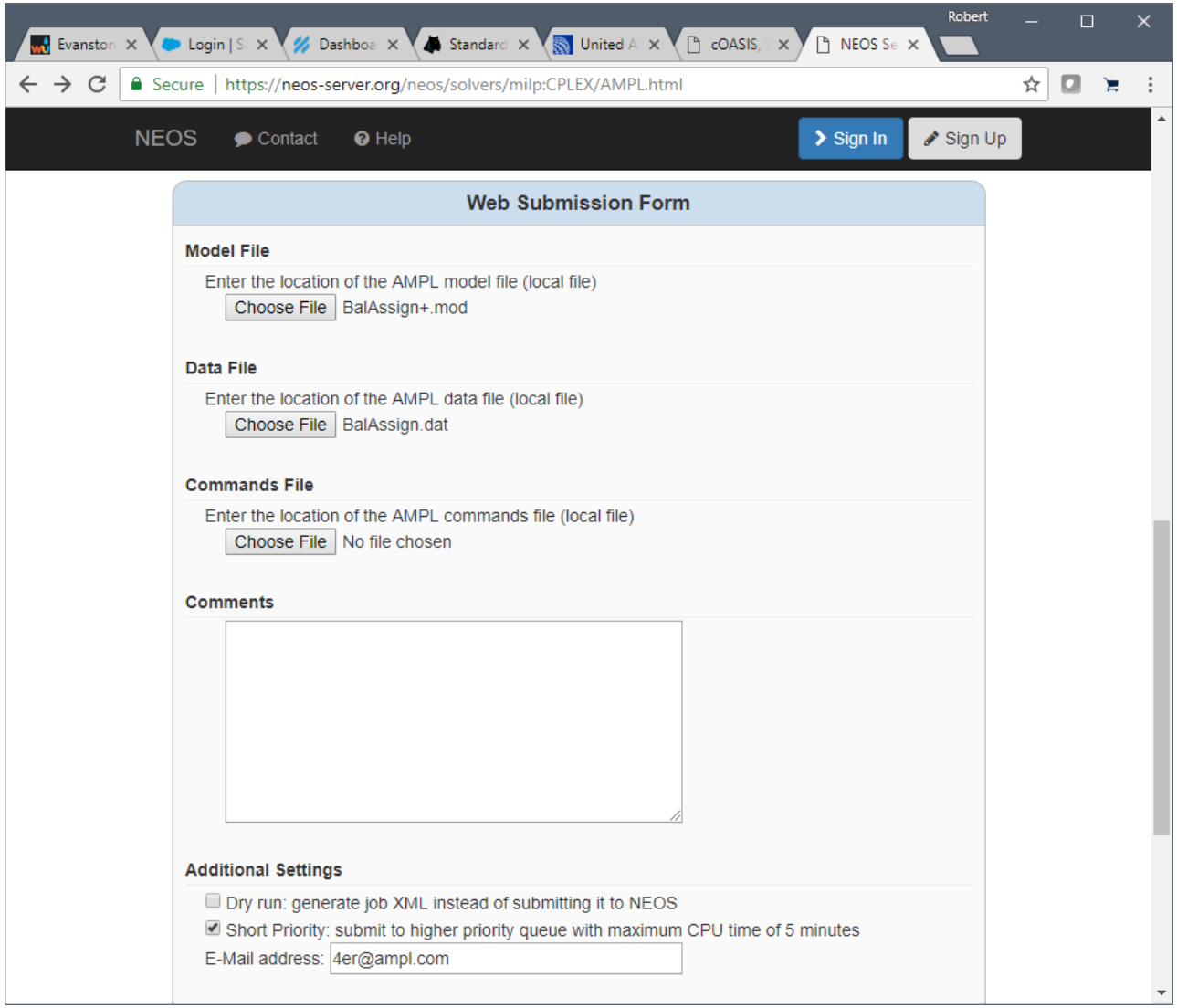


NEOS Server

Input Page for CPLEX using AMPL

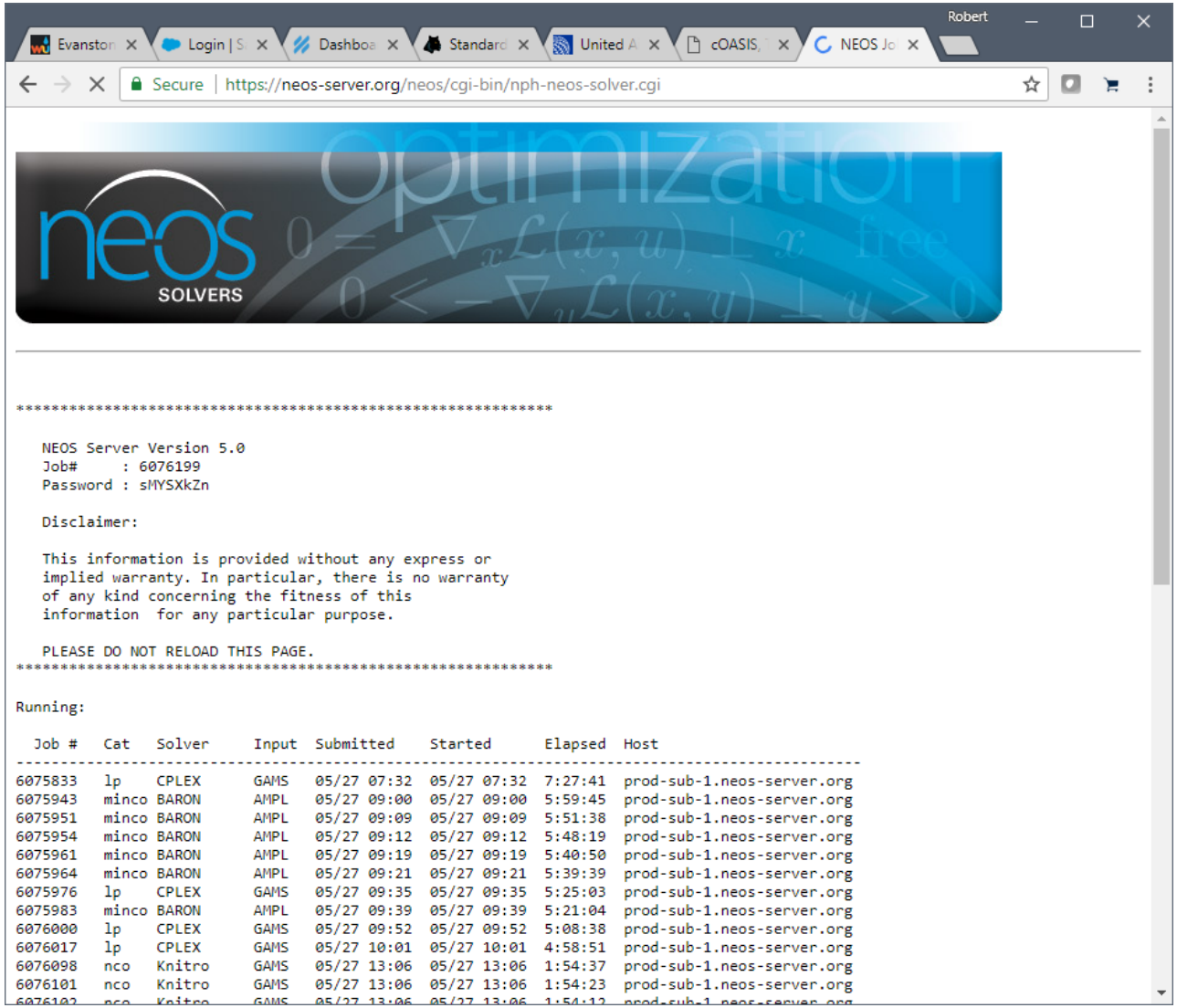
NEOS Server

Input Page *(cont'd)*



NEOS Server

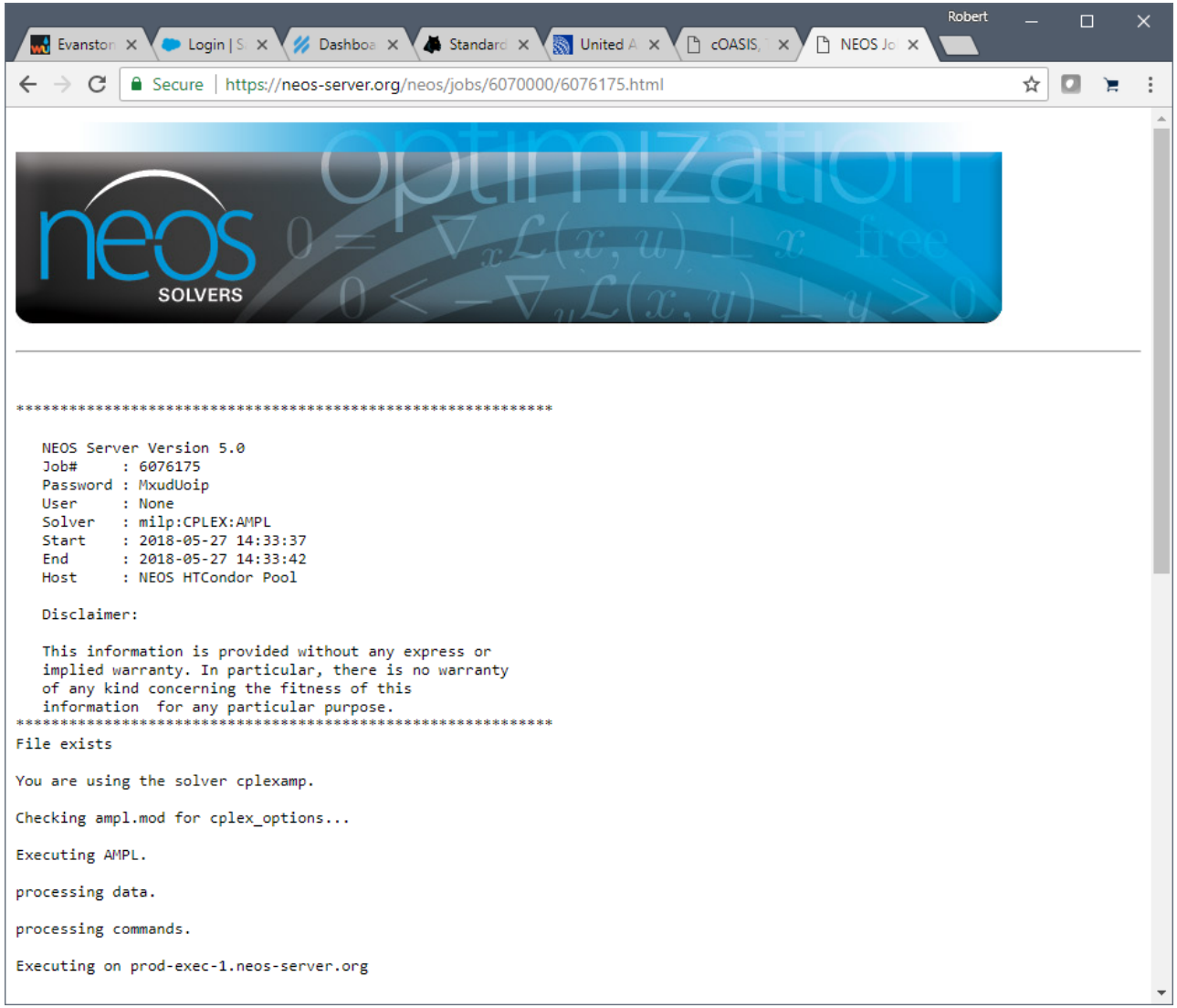
Queue Page



Job #	Cat	Solver	Input	Submitted	Started	Elapsed	Host
6075833	lp	CPLEX	GAMS	05/27 07:32	05/27 07:32	7:27:41	prod-sub-1.neos-server.org
6075943	minco	BARON	AMPL	05/27 09:00	05/27 09:00	5:59:45	prod-sub-1.neos-server.org
6075951	minco	BARON	AMPL	05/27 09:09	05/27 09:09	5:51:38	prod-sub-1.neos-server.org
6075954	minco	BARON	AMPL	05/27 09:12	05/27 09:12	5:48:19	prod-sub-1.neos-server.org
6075961	minco	BARON	AMPL	05/27 09:19	05/27 09:19	5:40:50	prod-sub-1.neos-server.org
6075964	minco	BARON	AMPL	05/27 09:21	05/27 09:21	5:39:39	prod-sub-1.neos-server.org
6075976	lp	CPLEX	GAMS	05/27 09:35	05/27 09:35	5:25:03	prod-sub-1.neos-server.org
6075983	minco	BARON	AMPL	05/27 09:39	05/27 09:39	5:21:04	prod-sub-1.neos-server.org
6076000	lp	CPLEX	GAMS	05/27 09:52	05/27 09:52	5:08:38	prod-sub-1.neos-server.org
6076017	lp	CPLEX	GAMS	05/27 10:01	05/27 10:01	4:58:51	prod-sub-1.neos-server.org
6076098	nco	Knitro	GAMS	05/27 13:06	05/27 13:06	1:54:37	prod-sub-1.neos-server.org
6076101	nco	Knitro	GAMS	05/27 13:06	05/27 13:06	1:54:23	prod-sub-1.neos-server.org
6076102	nco	Knitro	GAMS	05/27 13:06	05/27 13:06	1:54:12	prod-sub-1.neos-server.org

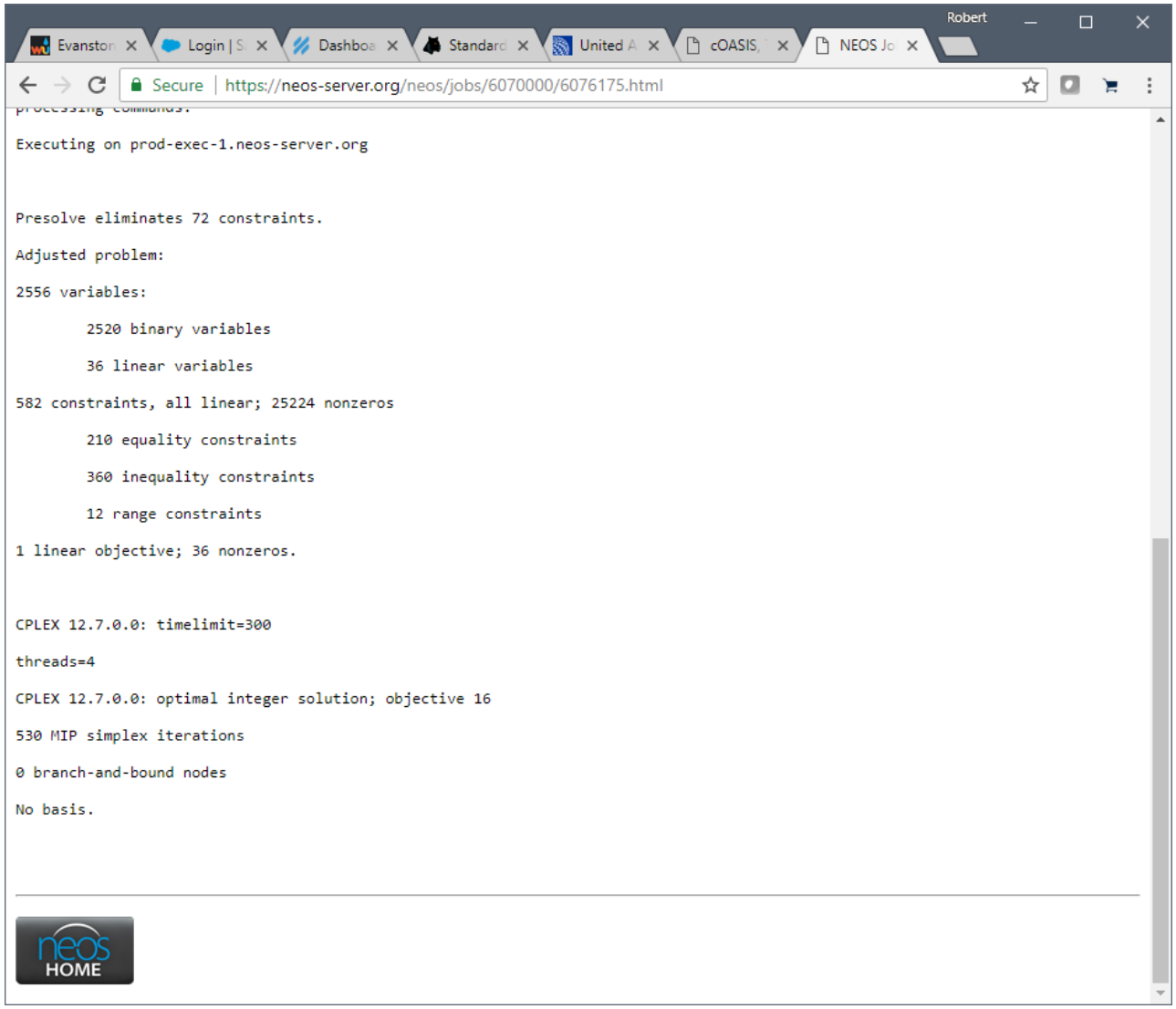
NEOS Server

Output Page



NEOS Server

Output Page *(cont'd)*



APIs

Application programming interfaces

- Access NEOS from a local program

Implementations

- Version 1: XML-RPC remote procedure call
- Version 5: full Python API

Uses

- NEOS submission tool
- NEOS option in Solver Studio for Excel
- **NEOS as a "solver" for modeling systems**

Modeling Systems as NEOS Clients

New "solvers"

- Kestrel for AMPL
- Kestrel for GAMS

Familiar operation

- Choose Kestrel as the local "solver"
- Set an option to choose a real solver on NEOS
- Initiate a solve and wait for results

E.D. Dolan, R. Fourer, J.-P. Goux, T.S. Munson and J. 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

NEOS Server

AMPL Interactive Session

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

ampl: let least_assign := 16;

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

ampl: solve;

Connecting to: neos-server.org:3332
Job 4679195 submitted to NEOS, password='JMNRQoTD'

Check the following URL for progress report :

http://neos-server.org/neos/cgi-bin/nph-neos-
solver.cgi?admin=results&jobnumber=4679195&pass=JMNRQoTD

Job 4679195 dispatched
password: JMNRQoTD

---------- Begin Solver Output -----------

Job submitted to NEOS HTCondor pool.
```

NEOS Server

AMPL Interactive Session

```
---------- Begin Solver Output -----------

Job submitted to NEOS HTCondor pool.

CPLEX 12.6.2.0: optimal integer solution; objective 265.9999999999943
135348 MIP simplex iterations
17430 branch-and-bound nodes

ampl: option omit_zero_rows 1, display_1col 0;

ampl: display Work;

Work [*] :=

  1 16    11 16    36 19    72 20    82 20   106 16   114 20   125 20
  3 16    29 16    66 17    79 19   104 19   112 16   121 16
;

ampl:
```

NEOS Server

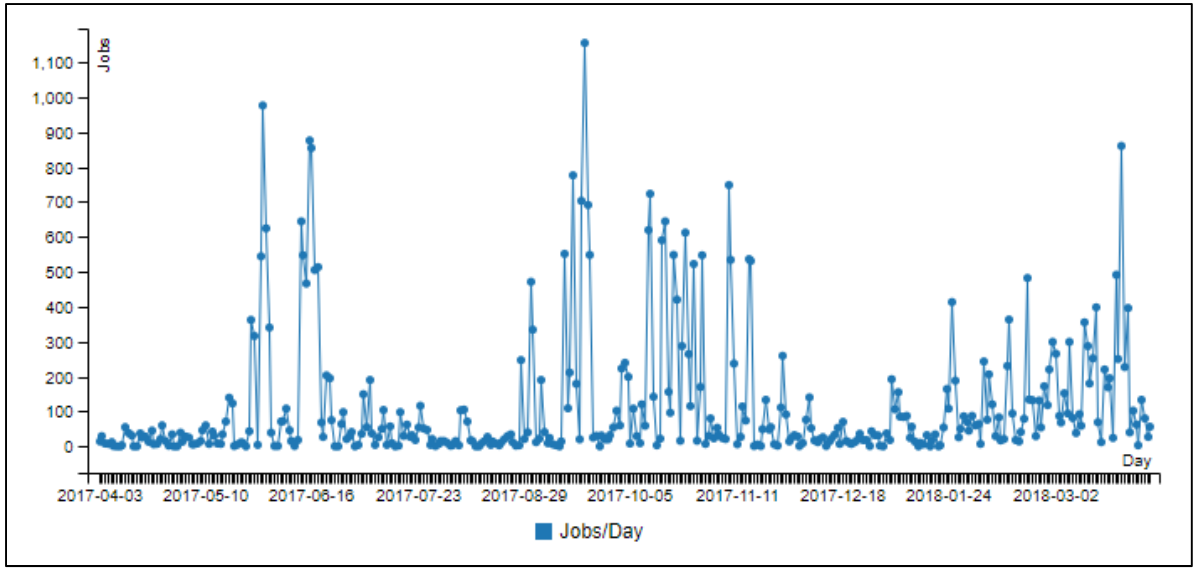
Kestrel Impact

Intensive use in short bursts

- Peaks of 10,000-60,000 per day

Modest use on average

- Average of 3,570 per month over past year
- Mostly AMPL/CPLEX



Kestrel Assessment

Strengths

- Powerful local client for modeling
- NEOS facilities for solving

Weaknesses

- Not all NEOS solvers available
- Local solver software is strong competition . . .
  - Bundled with modeling languages
  - Free for trial use
  - Free for course and academic use
- Limited support & development
  - Solver logs not currently returned

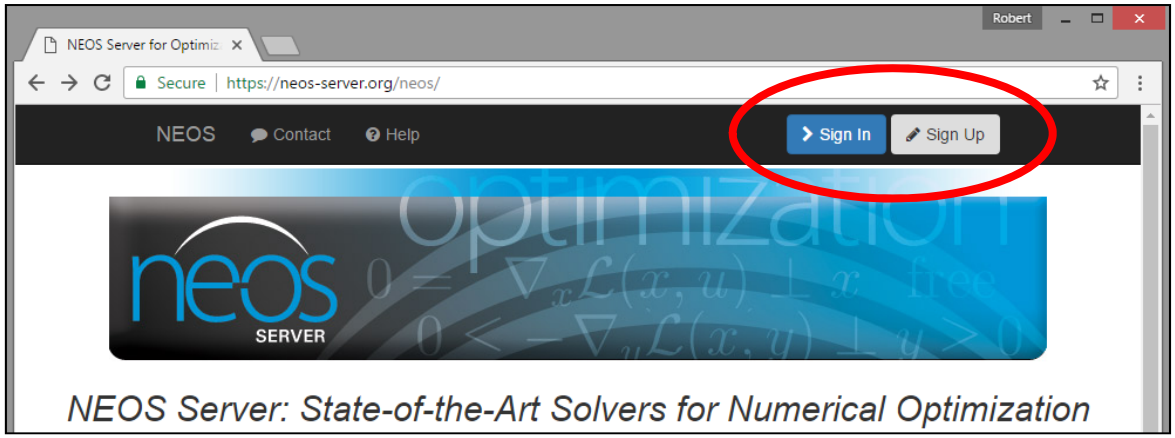
. . . we provide support for Kestrel/AMPL

Recent Developments

Intensified support

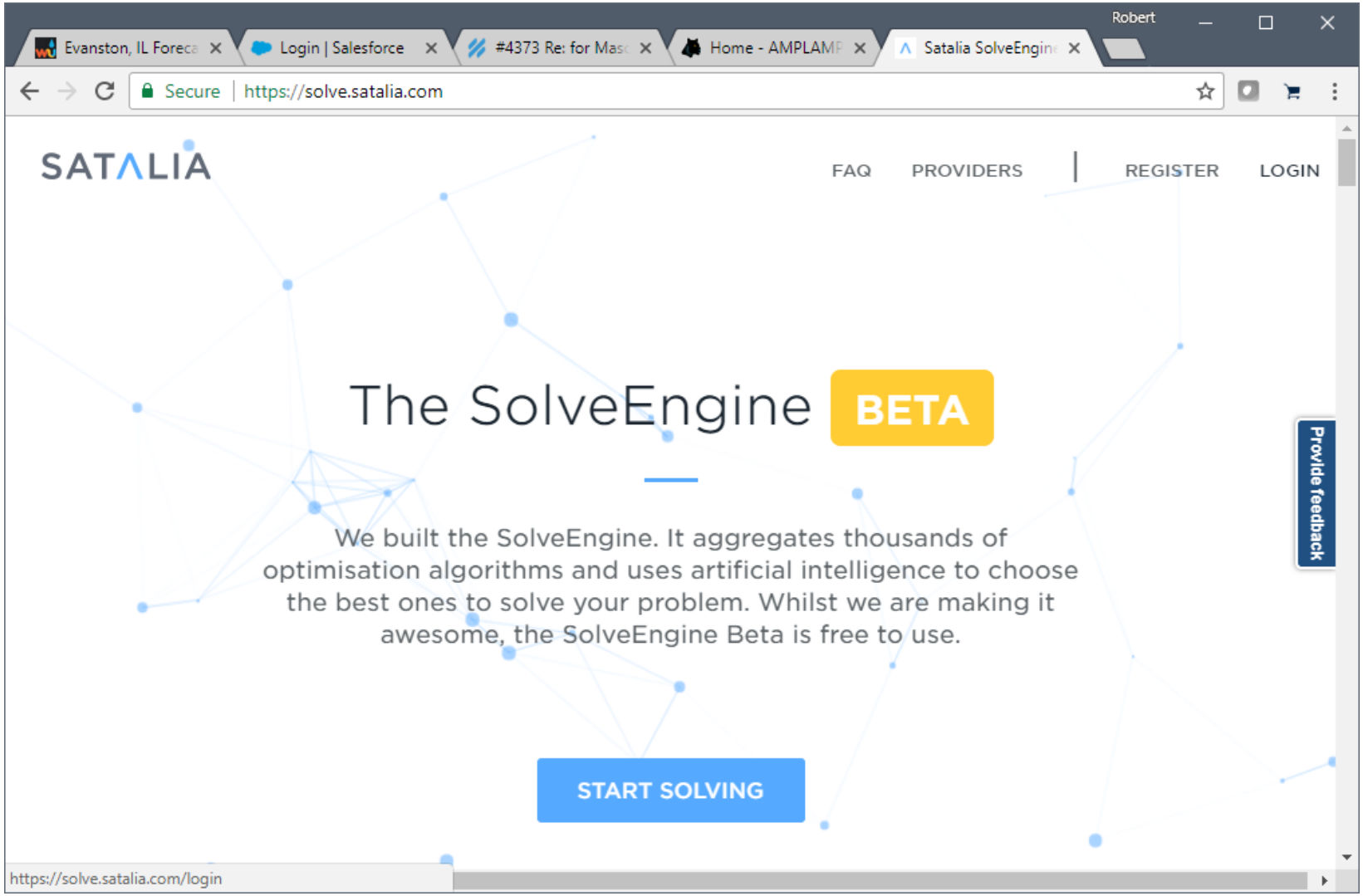
- Shift to HTCondor "high-throughput" platforms
- Updated Kestrel client
- Updated solver offerings

User accounts

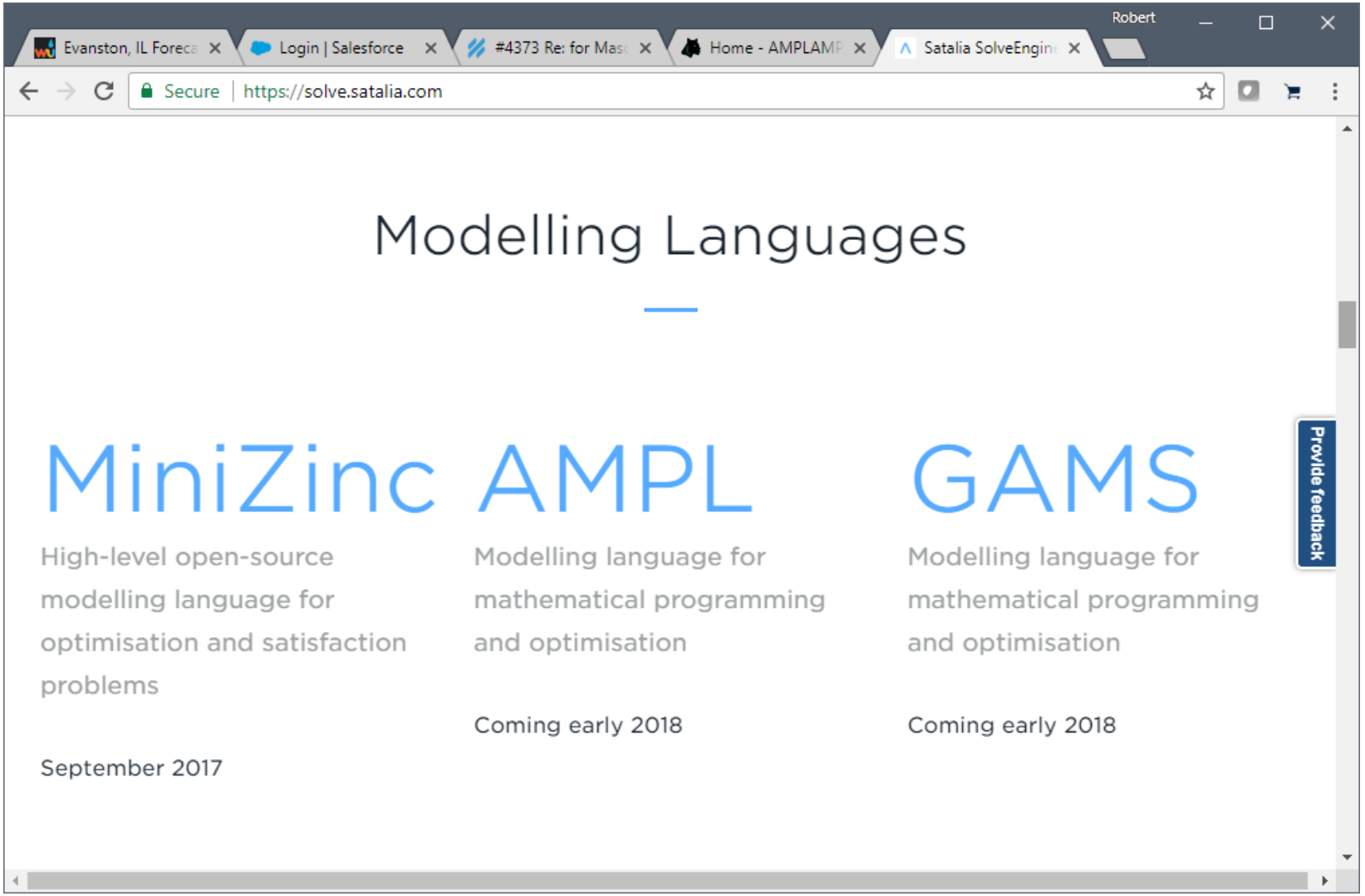


- Higher priority for job scheduling
- "My Jobs" tab listing recent jobs & links to results

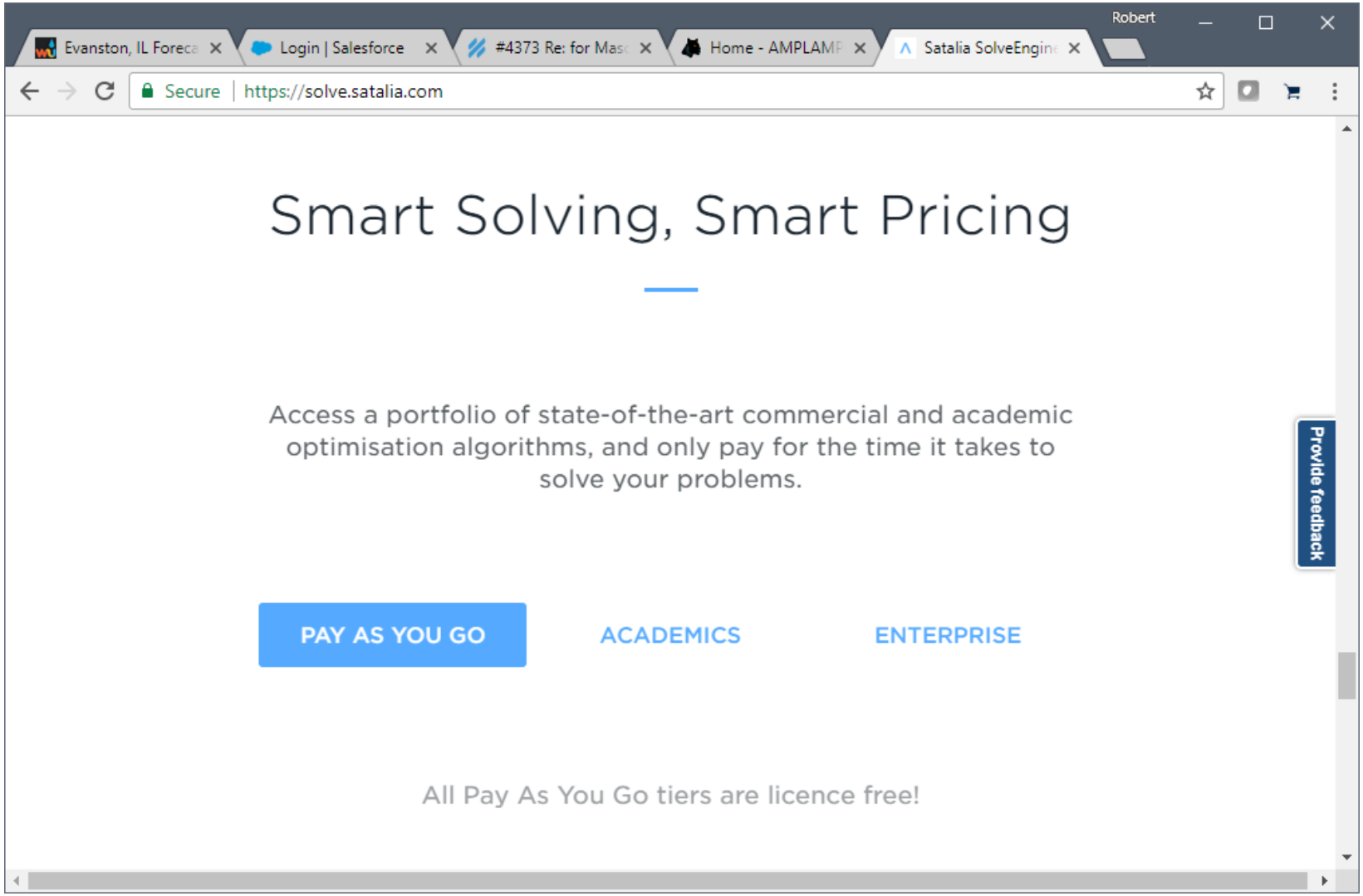
A Commercial NEOS?



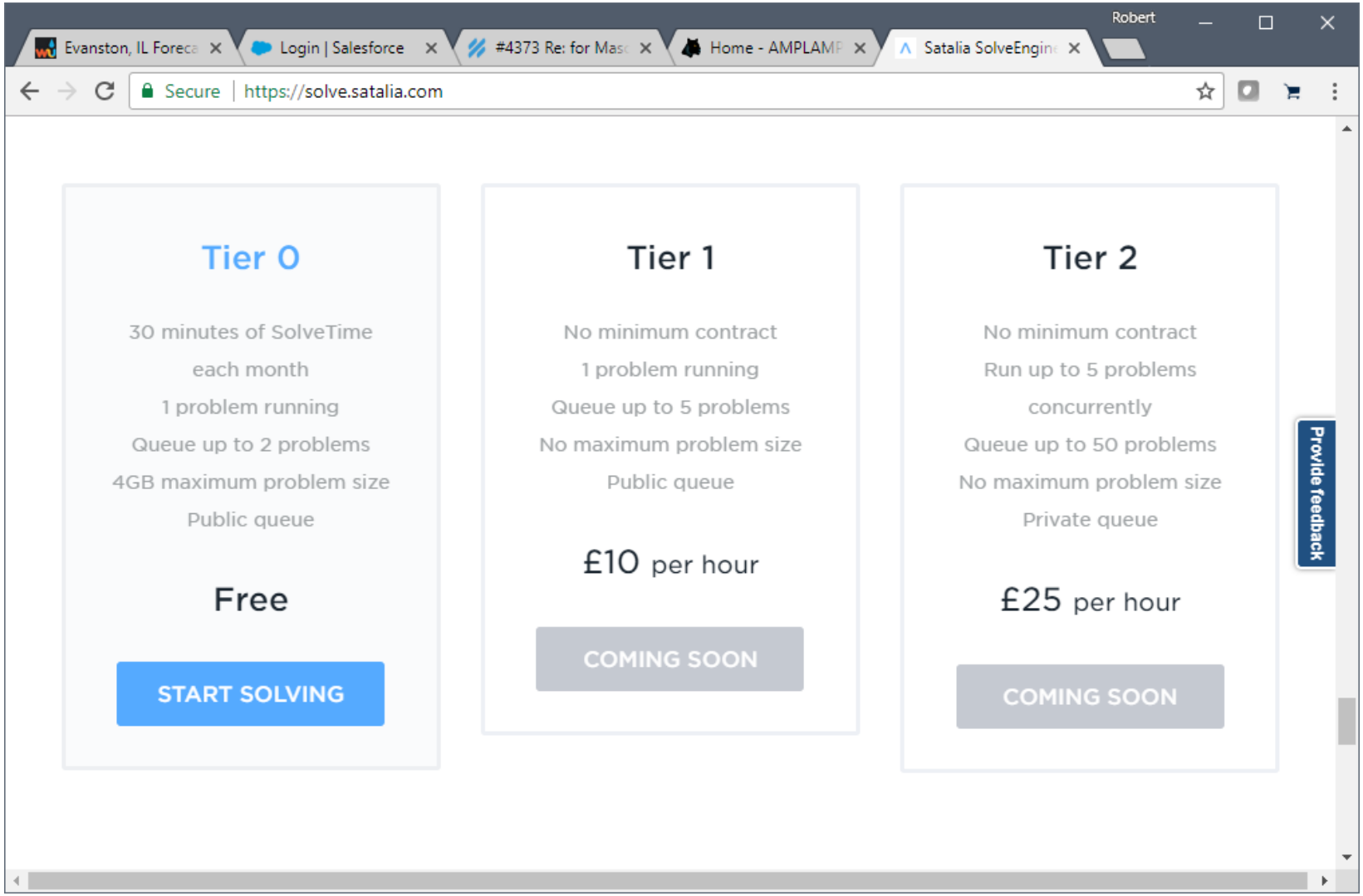
Satalia SolveEngine



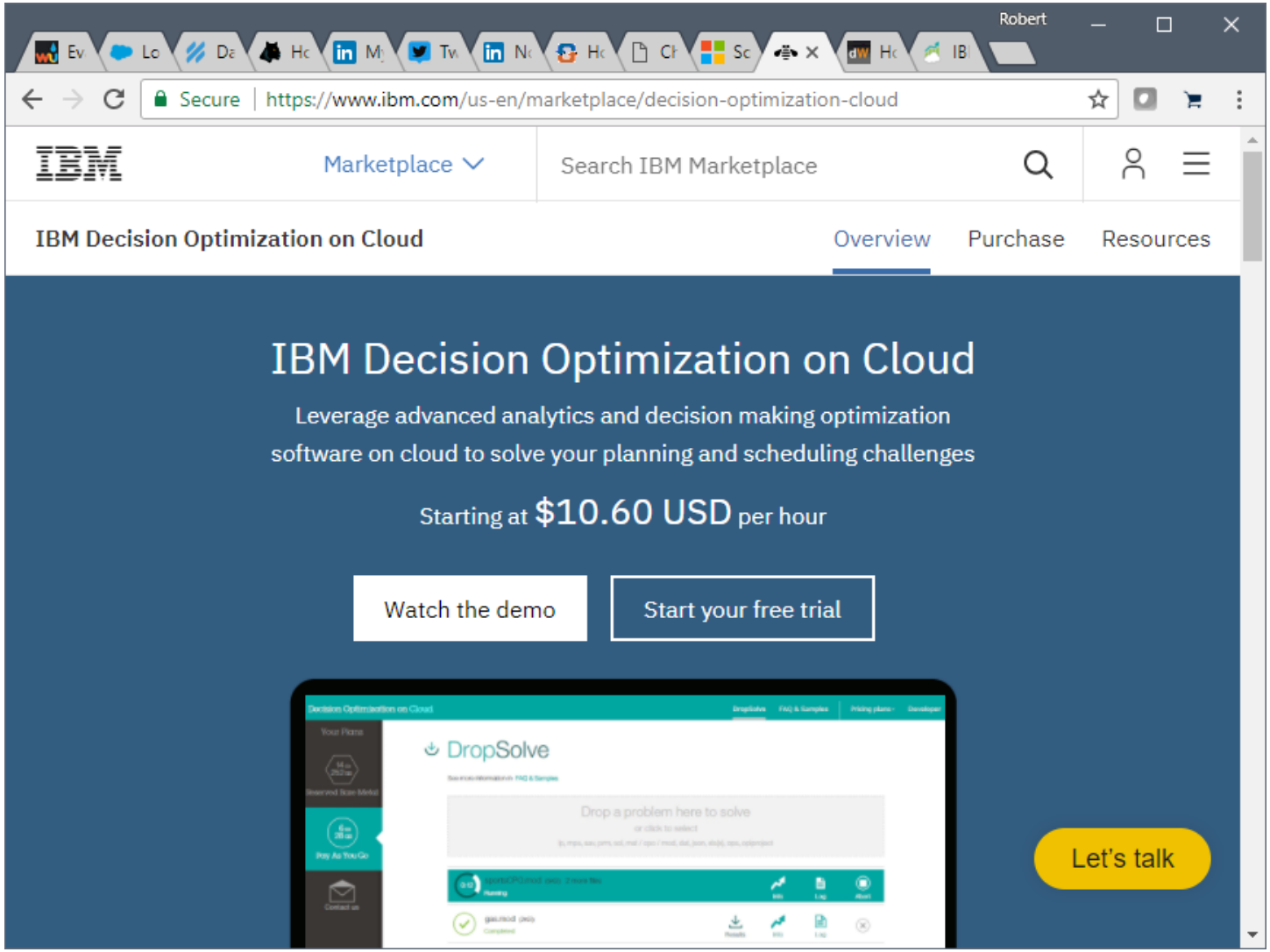
Satalia SolveEngine



Satalia SolveEngine



A Solver-Specific NEOS?



IBM Decision Optimization on Cloud

DOcplexcloud API

- Analogous to NEOS Python API

DropSolve service

- *Similar to NEOS web services*
- Matrix formats: LP, SAV, and MPS files
- OPL modeling language: MOD and DAT files
- Python programs using the API

IBM solvers

- CPLEX
- CP optimizer

See also FICO Analytic Cloud for Xpress solver

Gurobi Instant Cloud cloud.gurobi.com

Client side

- Standard Gurobi installation
- Cloud license

Server side

- Compute server for Gurobi solver
  - Single-machine solves
  - Distributed MIP solves
  - Distributed tuning
- Server pools with load balancing

. . . hosted on Amazon Web Services

"Cloud computing technology is changing quickly.
Please check these documents periodically to ensure
you have the latest instructions for the Gurobi Cloud."

Gurobi Instant Cloud for AMPL

Client side

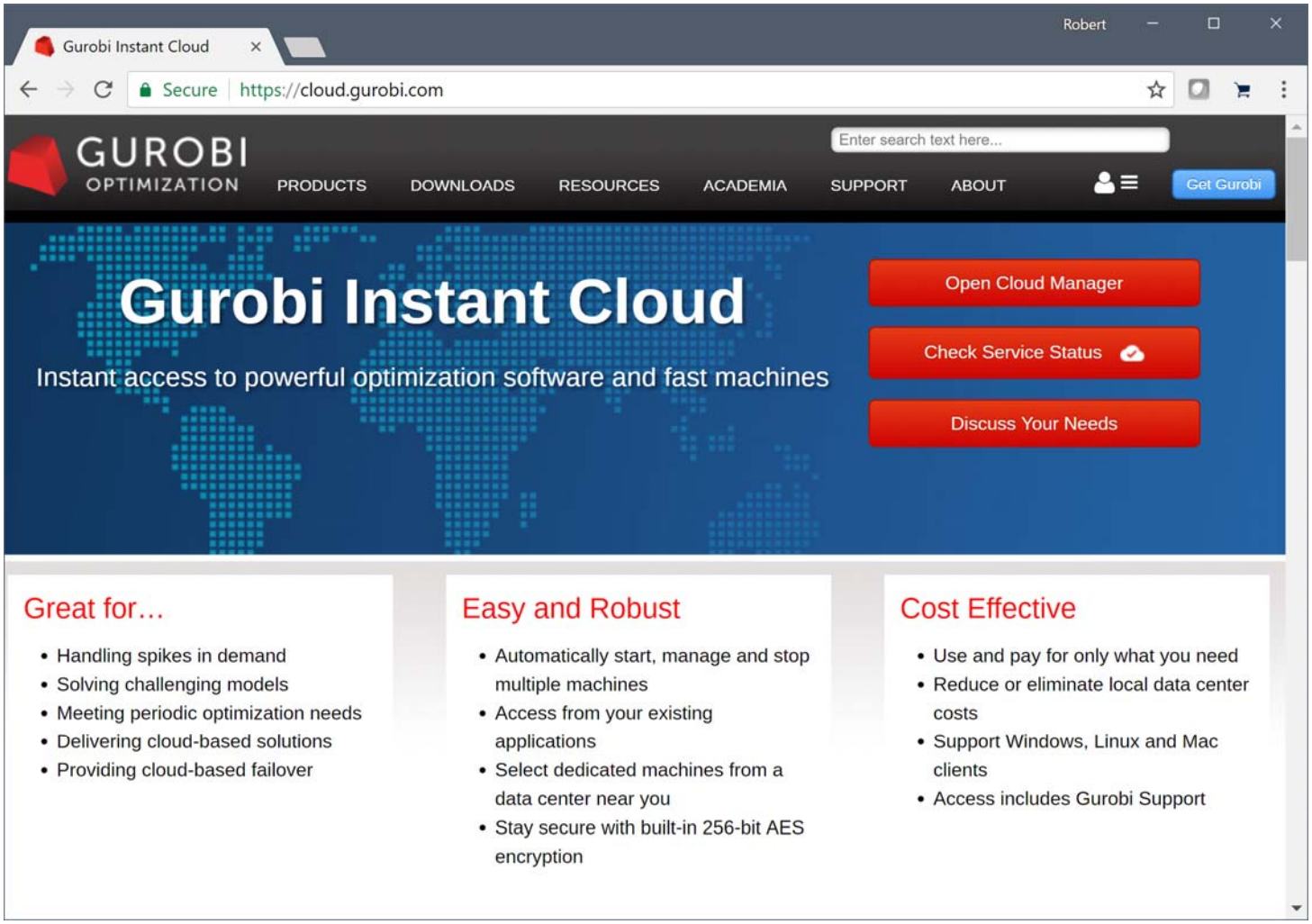
- AMPL installation (command-line or IDE)
- Standard Gurobi-for-AMPL installation

Server side

- Gurobi compute server
- Gurobi optimizer

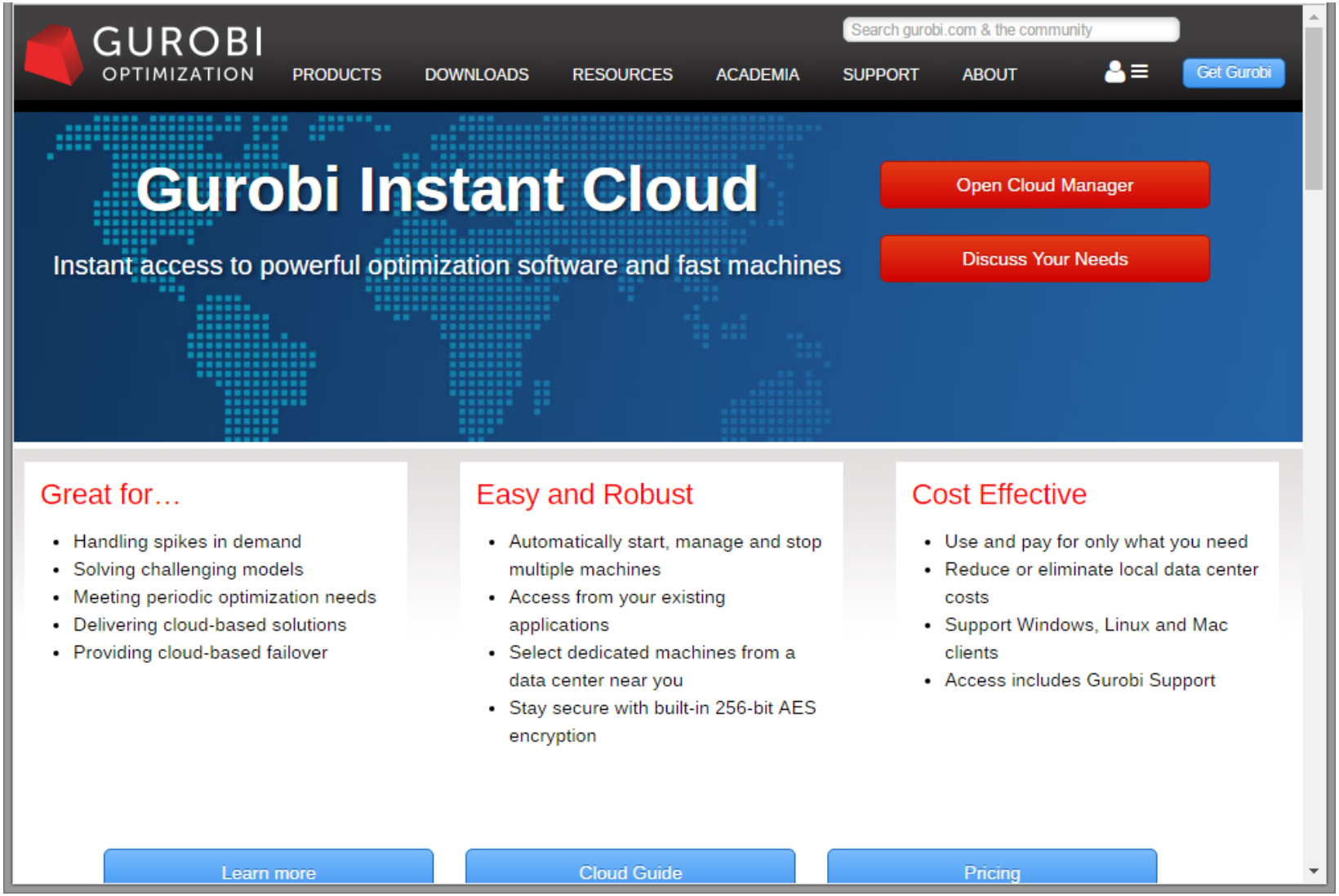
Gurobi Instant Cloud

cloud.gurobi.com



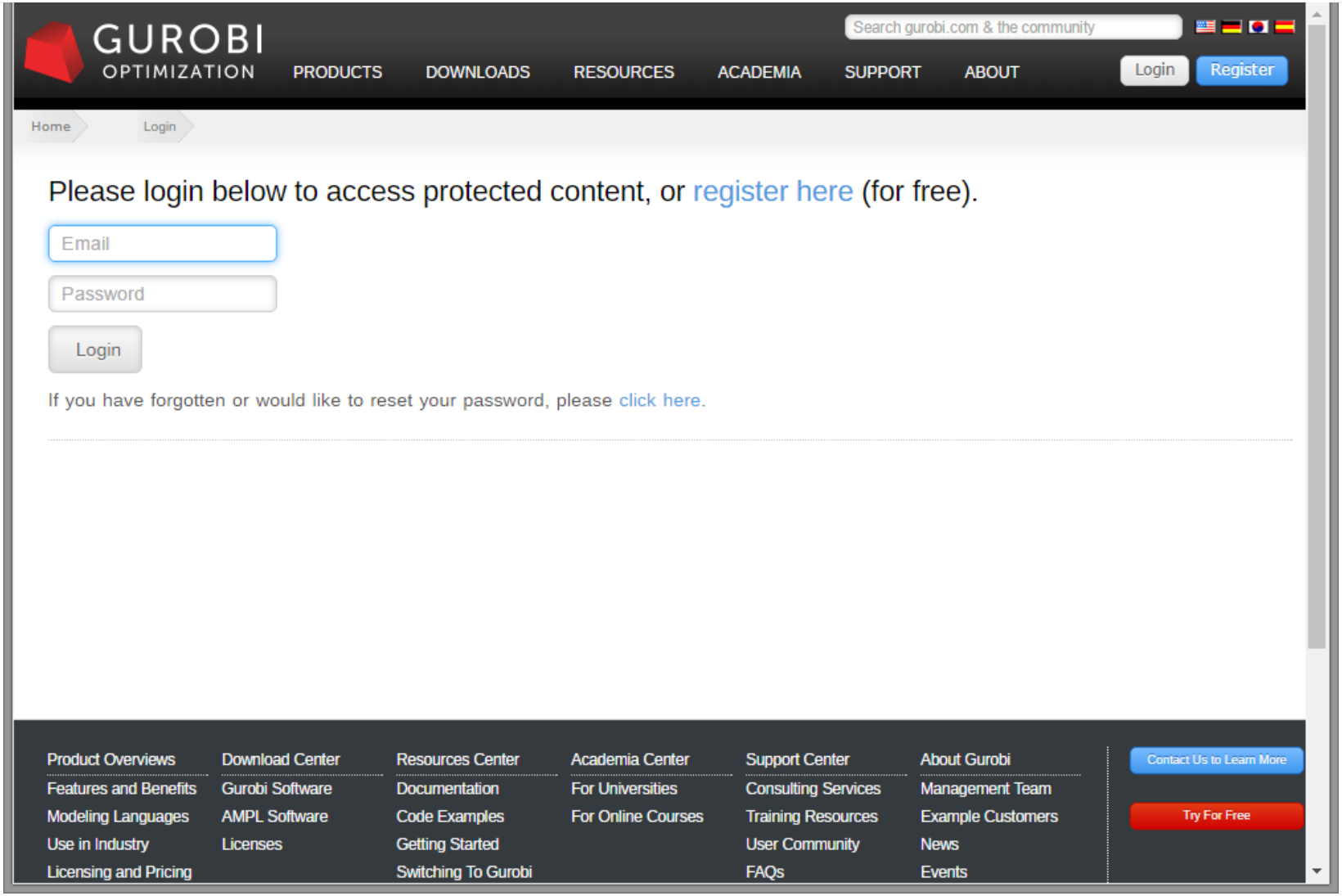
Gurobi Instant Cloud

cloud.gurobi.com



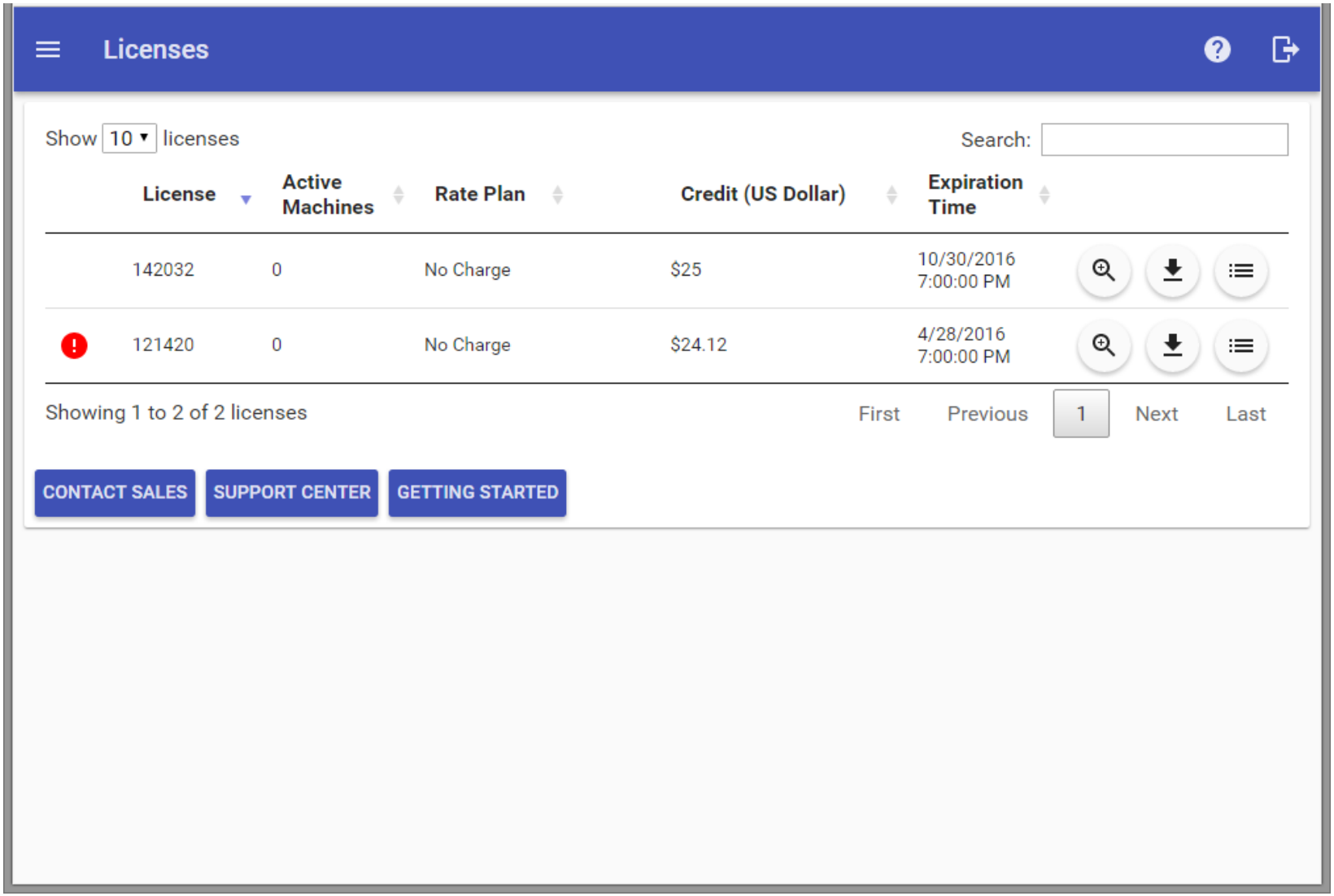
Gurobi Instant Cloud

www.gurobi.com



Gurobi Instant Cloud

View Available Licenses



Gurobi Instant Cloud

Get Gurobi License File

```
# This is a license file created by the Gurobi Instant Cloud
# Created on Mon, 17 Oct 2016 20:46:26 GMT
# License Id: 142032
# Place this file in your home directory or one of the following
# locations where XXX is the Gurobi Optimizer version you are using:
#    * C:\gurobi\ or C:\gurobiXXX\ on Windows
#    * /opt/gurobi/ or /opt/gurobiXXX/ on Linux
#    * /Library/gurobi/ or /Library/gurobiXXX/ on Mac OS X
# Or set environment variable GRB_LICENSE_FILE to point to this file
# Do not share this license file because it contains your secret key

CLOUDACCESSID=fedf3901-04f1-44d7-9725-e36c1c3f70f6
CLOUDKEY=0v9XdWrDQLiE3EiAAEKtFw
CLOUDHOST=ngcloud.gurobi.com
```

Gurobi Instant Cloud

Use with AMPL: *Setup*

```
ampl: model multmip3.mod;
ampl: data multmip3.dat;

ampl: option solver gurobi;

ampl: option gurobi_options
ampl?   'cloudid=fedf3901-04f1-44d7-9725-e36c1c3f70f6 \
ampl?    cloudkey=0v9XdWrDQLiE3EiAAEKtFw';

ampl:
```

Gurobi Instant Cloud

Use with AMPL: *Startup*

```
ampl: model multmip3.mod;
ampl: data multmip3.dat;

ampl: option solver gurobi;

ampl: option gurobi_options
ampl?   'cloudid=fedf3901-04f1-44d7-9725-e36c1c3f70f6 \
ampl?    cloudkey=0v9XdWrDQLiE3EiAAEKtFw';

ampl: solve;

Gurobi 7.0.0: cloudid=fedf3901-04f1-44d7-9725-e36c1c3f70f6
cloudkey=0v9XdWrDQLiE3EiAAEKtFw

Waiting for cloud server to start.........
```

Gurobi Instant Cloud

Use with AMPL: *Solve*

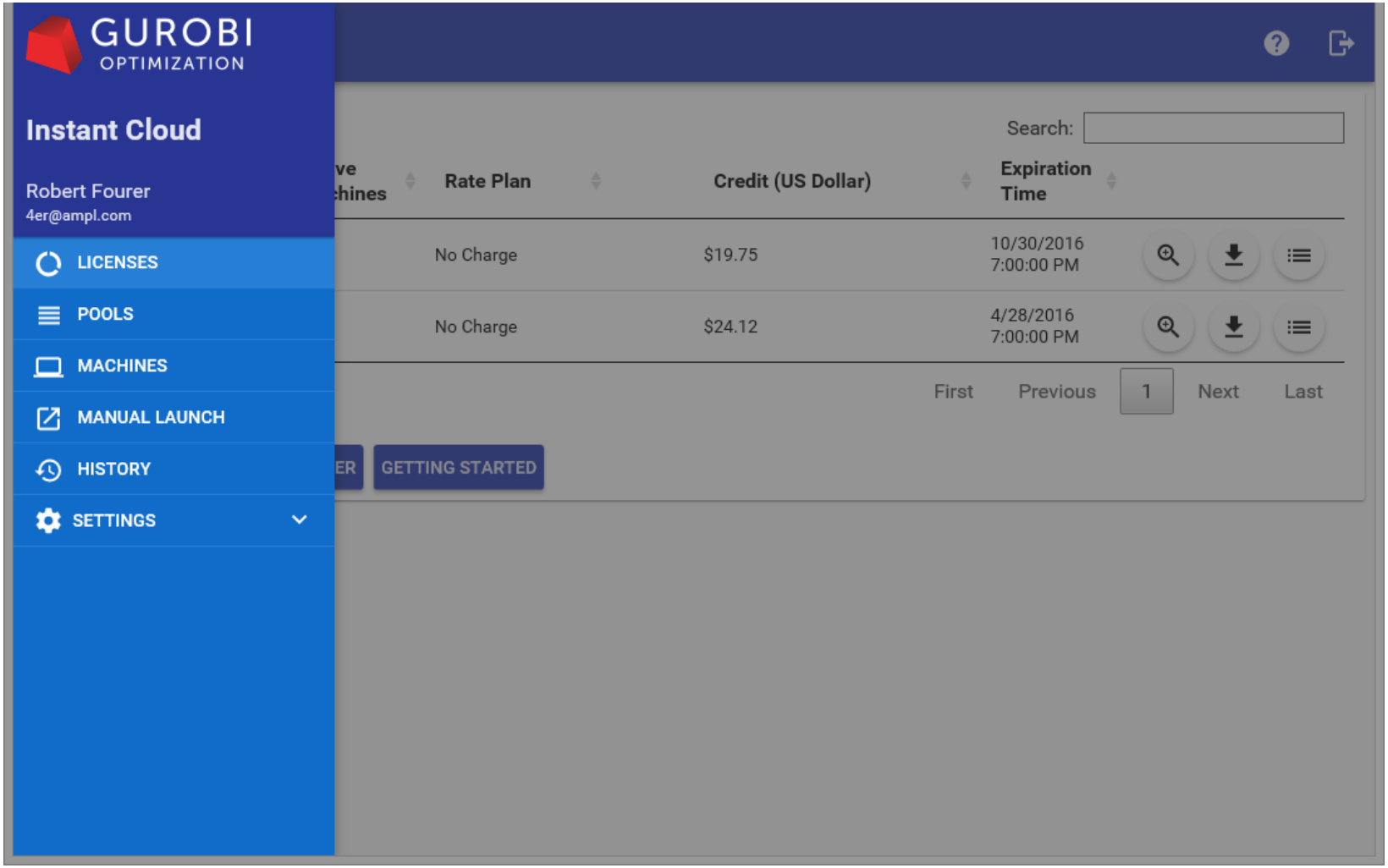
```
ampl: model multmip3.mod;
ampl: data multmip3.dat;

ampl: option solver gurobi;

ampl: option gurobi_options
ampl?   'cloudid=fedf3901-04f1-44d7-9725-e36c1c3f70f6 \
ampl?    cloudkey=0v9XdWrDQLiE3EiAAEKtFw';

ampl: solve;

Gurobi 7.0.0: cloudid=fedf3901-04f1-44d7-9725-e36c1c3f70f6
cloudkey=0v9XdWrDQLiE3EiAAEKtFw

Waiting for cloud server to start.............
Capacity available on 'default' cloud pool - connecting...
Established 256-bit AES encrypted connection

Gurobi 7.0.0: optimal solution; objective 235625
289 simplex iterations
25 branch-and-cut nodes
plus 35 simplex iterations for intbasis

ampl:
```

Use with AMPL: *Continue*

```
ampl: display {i in ORIG, j in DEST} sum {p in PROD} Trans[i,j,p];

:      DET   FRA   FRE   LAF   LAN   STL   WIN    :=
CLEV   625   375   550     0   500   550     0
GARY     0     0     0   400     0   625   375
PITT   525   525   625   600     0   625     0
;

ampl: reset data;
ampl: data multmip3a.dat;

ampl: solve;

Gurobi 7.0.0: cloudid=fedf3901-04f1-44d7-9725-e36c1c3f70f6
cloudkey=0v9XdWrDQLiE3EiAAEKtFw

Capacity available on 'default' cloud pool - connecting...
Established 256-bit AES encrypted connection

Gurobi 7.0.0: optimal solution; objective 238450
163 simplex iterations
plus 33 simplex iterations for intbasis

ampl:
```

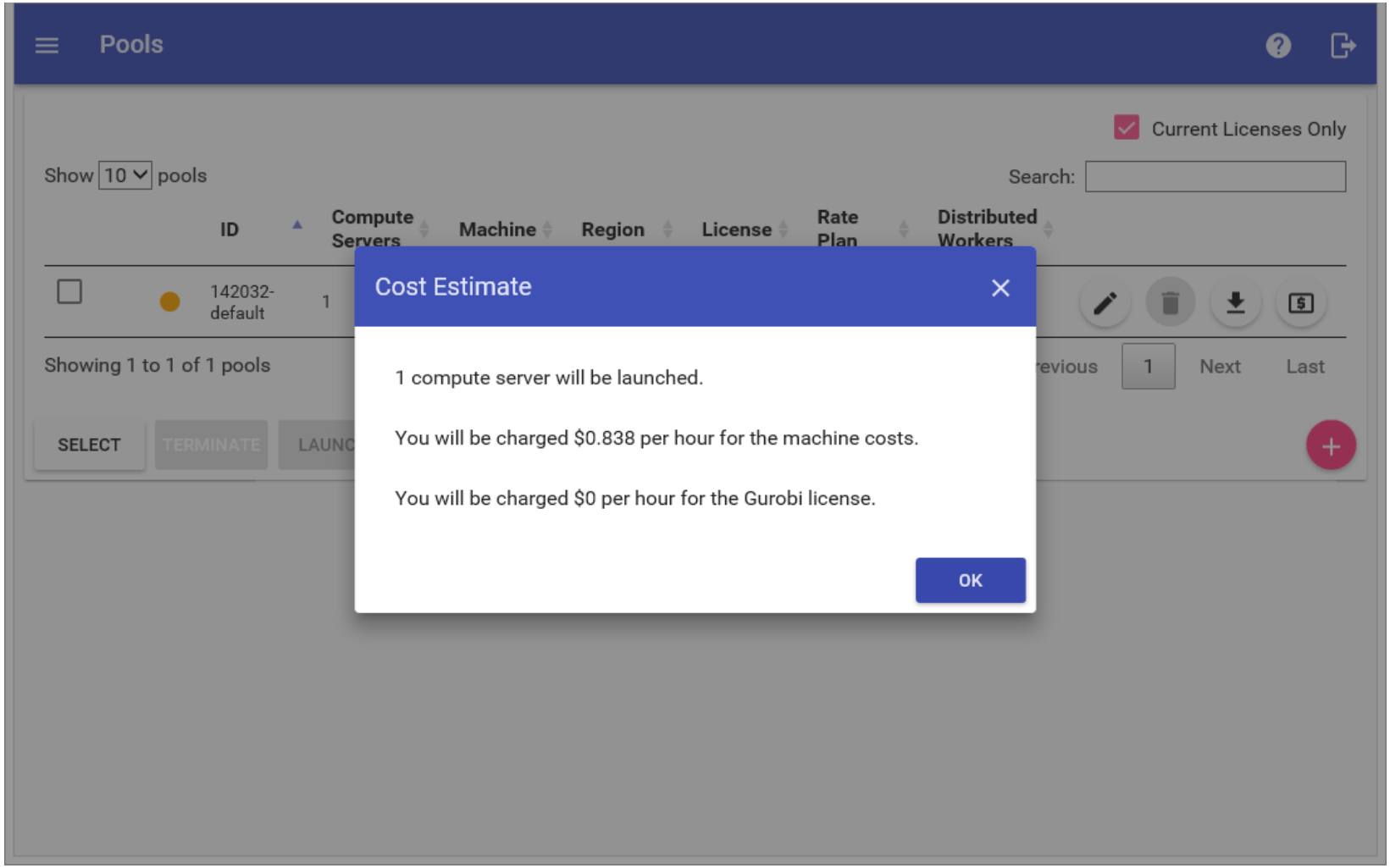
Gurobi Instant Cloud

Manage Server Configuration



Gurobi Instant Cloud

Check Costs



Gurobi Cloud Costs

Commercial plans

- Annual subscription fee, ***plus***
- Hourly rates for use:
  - Gurobi rate for compute servers
  - Amazon rate for distributed workers

Pick the Gurobi Instant Cloud plan that best meets your needs:

Plan	Cost	Best For...
Starter	$1,500 for three months of access, incl. Support, + $20/hr. per machine	...situations where you want a short-term introduction to the Instant Cloud. Includes a Compute Server license. Note that the Starter Plan is not renewable.
Bronze	$4,000 for one year of access, incl. Support, + $10/hr. per machine	...situations where you just want to solve one model at a time and don't need the features or flexiblity of Compute Server. Includes a Single-Use license.
Silver	$10,000 for one year of access, incl. Support, + $8/hr. per machine	...more active development and deployment situations where the hourly charge is more of a factor. Includes a Compute Server license.
Gold	$20,000 for one year of access, incl. Support, + $5/hr. per machine	...heavy usage situations where the features of Compute Server and getting the lowest hourly charge are important.

We recommend starting with our default instance size, c4.4xlarge, and then experimenting from there depending on your needs and budget. Instances with more than 16 vCPUs (such as the c4.8xlarge and r3.8xlarge instances) are charged at 2X the stated hourly price.

Gurobi Cloud for AMPL: Assessment

Strengths

- Security
- Reliability (via Amazon)
- Support for multi-server and/or multi-worker pools
- Support for local modeling clients

Drawbacks (compared to NEOS)

- Not free
  - Budgeting can be complicated
- Solver-specific
- Not quite “optimization on demand”

QuanDec

Server side

- AMPL model and data
- Standard AMPL-solver installations

Client side

- Interactive tool for collaboration & decision-making
- Runs on any recent web browser
- Java-based implementation
  - AMPL API for Java
  - Eclipse Remote Application Platform

. . . developed / supported by Cassotis Consulting

Initialization

Prepare the model and data

- Add reporting variables to the model
- Select initial data in AMPL .dat format

Import to QuanDec

- Install on a server
- Read zipfile of model and data
- Create new application and first master problem

Configure displays

- Create data tables
- Adjust views

. . . mostly done automatically

QUANDEC

Quantify your decisions

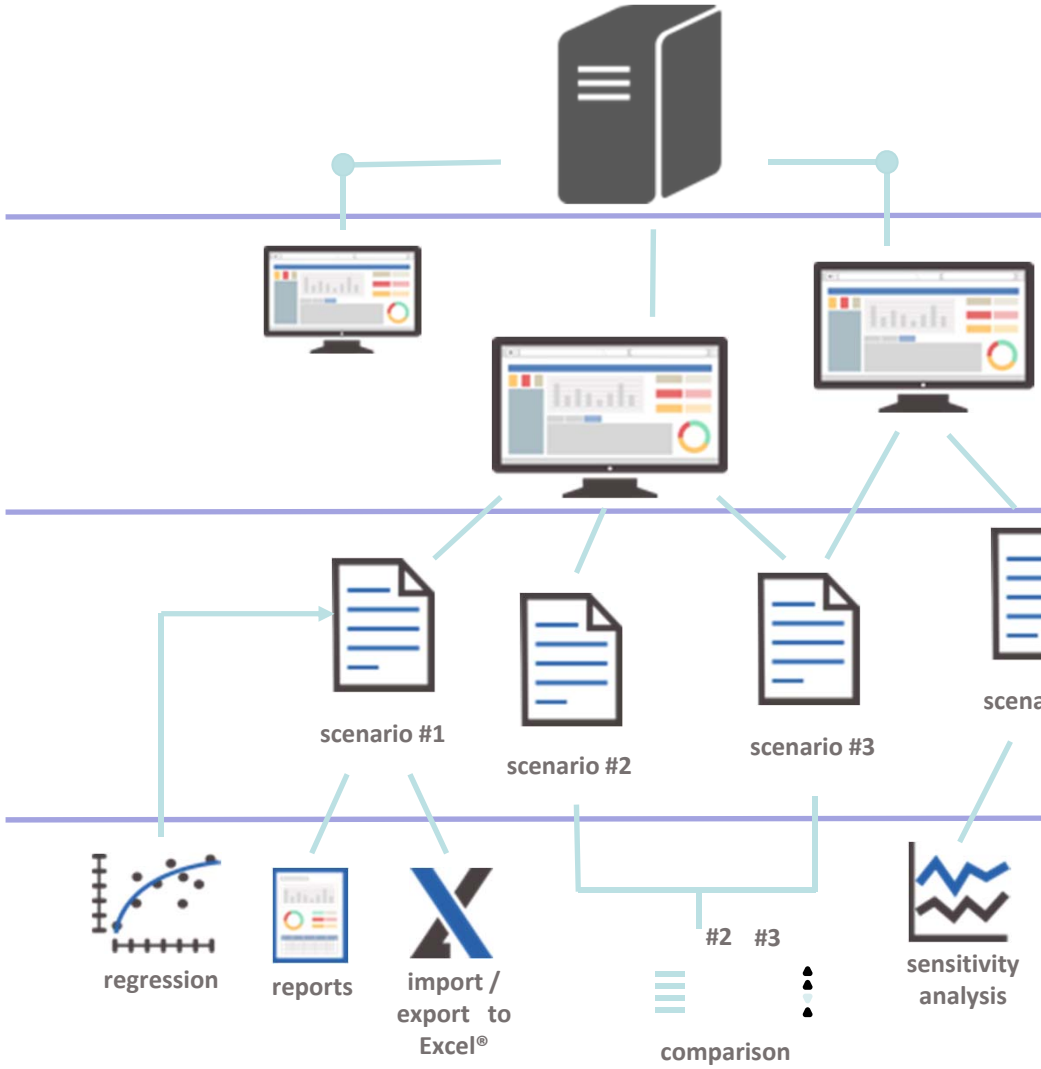
CASSOTIS

CONSULTING & SOLUTIONS IN OPTIMIZATION

The web-based graphical interface that turns optimization models written in AMPL into decision-making tools

AMPL

Features



Server application

Centralized data

Several apps on a single instance

Web-based

Multi-users

Concurrent access

Secure access

Scenario-based

Sharing between users

Sharing rights (edit / comment/ view)

And much more…

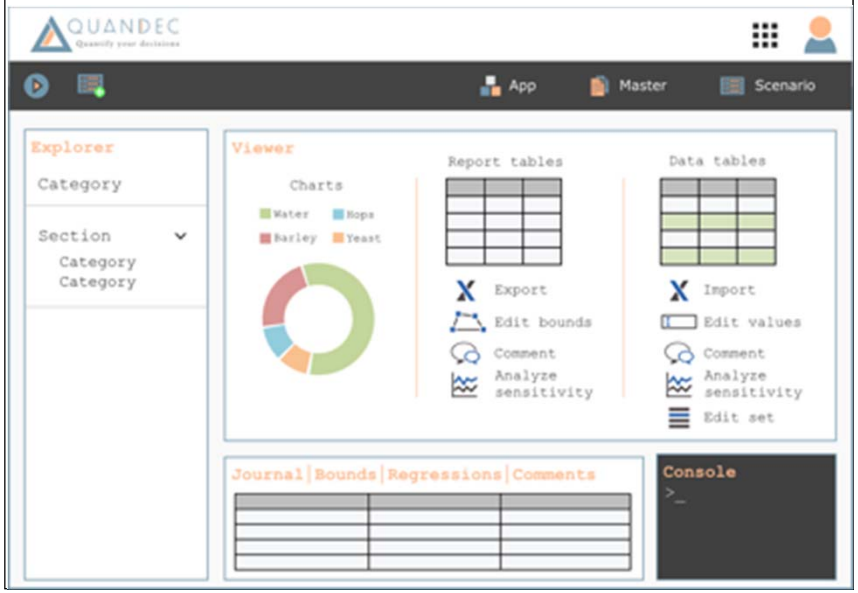
Getting started

Your
AMPL model

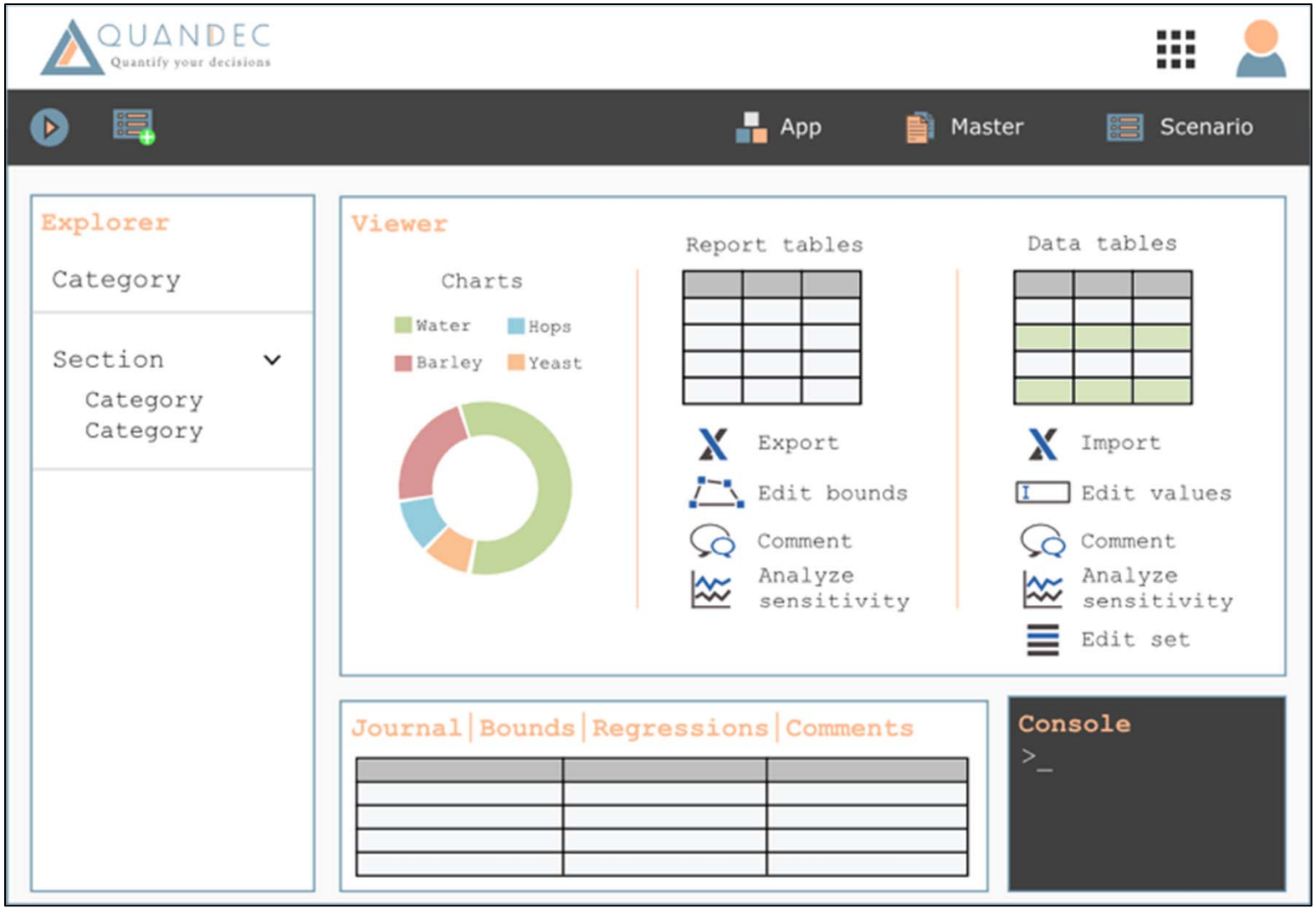
Configure how you want to display your parameters and variables (many options of tables and charts)

Zip and upload configuration and AMPL model files into

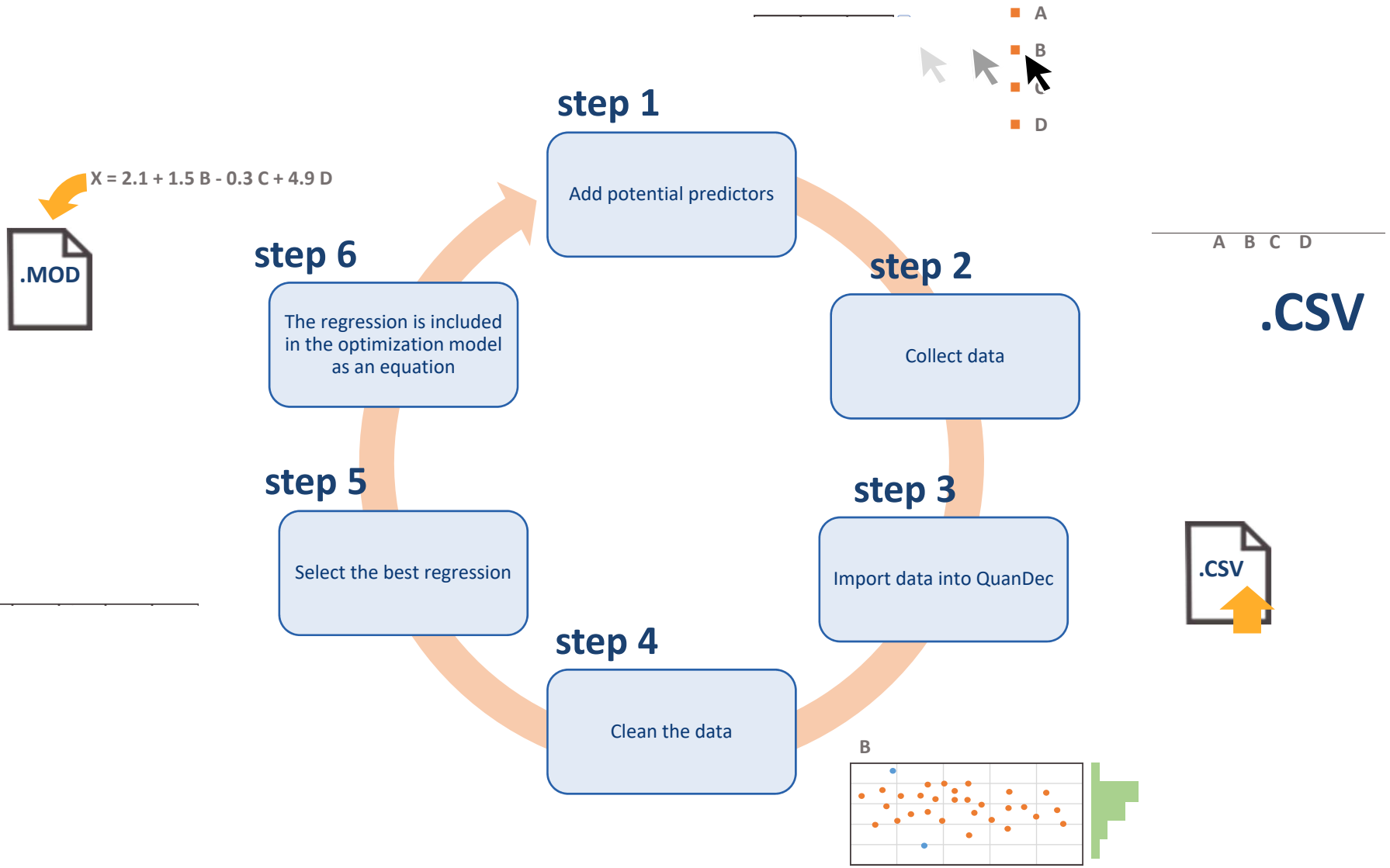
Quantify your decisions!



Workbench


QUANDEC
Quantify your decisions
App
Master
Scenario
Explorer
Category
Section
Category
Category
Viewer
Charts
Water
Hops
Barley
Yeast
Report tables
Export
Edit bounds
Comment
Analyze sensitivity
Data tables
Import
Edit values
Comment
Analyze sensitivity
Edit set
Journal | Bounds | Regressions | Comments
Console
>_

Regression tool



Quandec's value proposition

GUI creation in less than 2 days, without any programming

No need of knowledge in Operations Research to be an end user

No proprietary format: use your preferred editor and use any good versioning system

Run optimization in various conditions (scenarios)

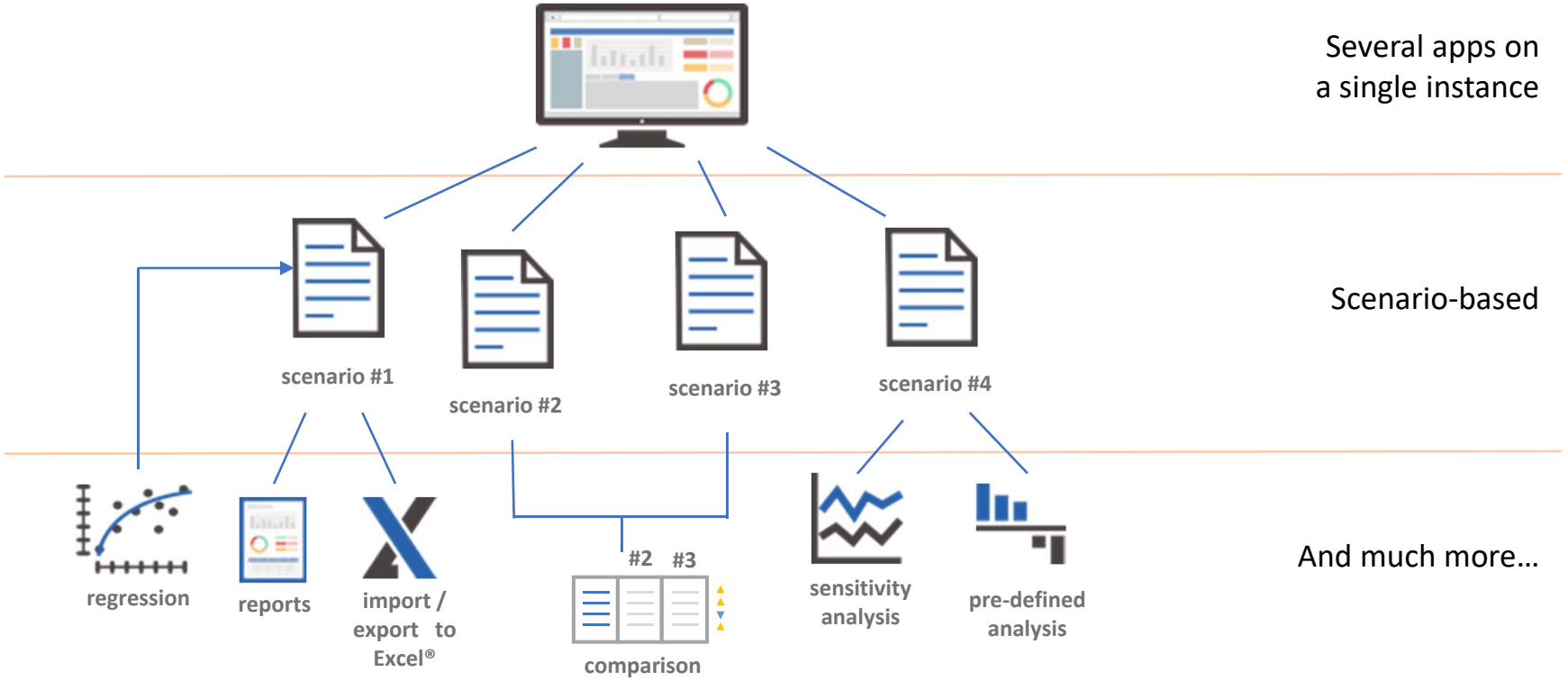
Intuitive & user-friendly

Benefit from all AMPL language features

Recurrent decisions based on the result of analyses (scripts)

Standalone Quandec Version

Available in 2018 Q3



QuanDec Availability

Contact sales@ampl.com

- Free trials available
- Pricing keyed to number of models & users

First year's support included

- Tailored setup support from Cassotis Consulting
- Customizations possible