Optimization Services (OS)

-- A Framework for Optimization Software

-- A Computational Infrastructure

-- The Next Generation NEOS

-- The OR Internet

Jun Ma

Industrial Engineering and Management Sciences
Northwestern University
IFORS, Hawaii, 07/14/2005
1. Motivations

2. Optimization Services and Optimization Services Protocol

3. Future and Derived Research
Motivation

Future of Computing
Motivation

But how... with so many type of components

1. Modeling Language Environment (MLE)
   (AIMMS, AMPL, GAMS, LINGO, LPL, MOSEL, MPL, OPL, MathProg, PuLP, POAMS, OSmL)

2. Solver
   (Too many)

3. Analyzer/Preprocessor
   (Analyzer, MProbe, Dr. AMPL)

4. Simulation
   (Software that does heavy computation, deterministic or stochastic)

5. Server/Registry
   (NEOS, BARON, HIRON, NIMBUS, LPL, AMPL, etc.)

6. Interface/Communication Agent
   (COIN-OSI, CPLEX-Concert, AMPL/GAMS-Kestrel, etc.)

7. Low Level Instance Representation
   (Next page)
## Motivation

But how… with so many optimization types and representation formats

<table>
<thead>
<tr>
<th>Optimization Type</th>
<th>Representation Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Programming</td>
<td>MPS, xMPS, LP, CPLEX, GMP, GLP, PuLP, LPFML, MLE instances</td>
</tr>
<tr>
<td>Quadratic Programming</td>
<td></td>
</tr>
<tr>
<td>Mixed Integer Linear Programming</td>
<td></td>
</tr>
<tr>
<td>Nonlinearly Constrained Optimization</td>
<td>MLE instances</td>
</tr>
<tr>
<td>Bounded Constrained Optimization</td>
<td>SIF (only for Lancelot solver)</td>
</tr>
<tr>
<td>Mixed Integer Nonlinearly Constrained Optimization</td>
<td></td>
</tr>
<tr>
<td>Complementarity Problems</td>
<td></td>
</tr>
<tr>
<td>Nondifferentiable Optimization</td>
<td></td>
</tr>
<tr>
<td>Global Optimization</td>
<td></td>
</tr>
<tr>
<td>Semidefinite &amp; Second Order Cone Programming</td>
<td>Sparse SDPA, SDPL</td>
</tr>
<tr>
<td>Linear Network Optimization</td>
<td>NETGEN, NETFLO, DIMACS, RELAX4</td>
</tr>
<tr>
<td>Stochastic Linear Programming</td>
<td>sMPS</td>
</tr>
<tr>
<td>Stochastic Nonlinear Programming</td>
<td>None</td>
</tr>
<tr>
<td>Combinatorial Optimization</td>
<td>None (except for TSP input, only intended for solving</td>
</tr>
<tr>
<td></td>
<td>Traveling Sales Person problems.</td>
</tr>
<tr>
<td>Constraint and Logic Programming</td>
<td>None</td>
</tr>
<tr>
<td>Optimization with Distributed Data</td>
<td>None</td>
</tr>
<tr>
<td>Optimization via Simulation</td>
<td>None</td>
</tr>
</tbody>
</table>
Motivation
Look at the NEOS server Web site
Motivation
As if it’s not bad enough …

1. Tightly-coupled implementation (OOP? Why not!)

2. Various operating systems

3. Various communication/interfacing mechanisms

4. Various programming languages

5. Various benchmarking standards
Motivation

Now...

• The key issue is communication, not solution!

• ... and Optimization Services is intended to solve all the above issues.
OUTLINE

1. Motivations

2. Optimization Services and Optimization Services Protocol

3. Future and Derived Research
Optimization Services (OS)
What is happening behind?

XML-based standard

Modeler

Parse to OSiL

Agent

OSmL

Solver

Model/Data

AMPL

OS Server

Solver

Web address

Web page

OS

location

OS Server

Solver

OSP -- OShL (OSiL)

CGI

http/html

Socket

browser

html form

Data in HTML Form

Data in HTML Form

Registry

Analyser

Max f(x)

s.t. lb_1 <= g_1(x) <= ub_2

lb_2 <= g_2(x) <= ub_2

f(x) can be sin(x(1)) + x(x(2))

g_1(x) can be if(x(1) > 0) then x(2) else cost(x(2))

g_2(x) can be a metric from a finite element simulation (non-closed form black box function evaluator)
Optimization Services
What is it? – A framework for optimization software
Optimization Services
What is it? – A computational infrastructure
• The NEOS server and its connected solvers uses the OS framework.
• NEOS accepts the OSiL and other related OSP for problem submissions
• NEOS becomes an OS compatible meta-solver on the OS network
• NEOS hosts the OS registry
Optimization Services
What is it? – The OR Internet

Diagram:
- Internet User
  - Internet Browser
    - Socket
      - Search Engine
        - http
          - html
    - Web Server
      - Web Pages
    - Modeling Language Environment (MLE) or GUI
      - Communication agent
        - OSP
          - OSxL
            - Optimization Services registry
              - Solvers/ OR software
- OS Server
Optimization Services Protocol (OSP)

What is it? – Application level networking protocol
– Interdisciplinary protocol between CS and OR

The 7-layer OSI Model  The 4-layer Internet model

Get /xt/services/colorRequest HTTP/1.0
Content-typename: text/xml; charset=utf-8
SOAPAction: "get/color"

<soap:Envelope>
  <soap:Body>
    <string solve(String instance)>
      input string instance follow OSiL
    </string solve(String instance)>
    -- output string follow OSiL
    <soap:Body>
  </soap:Envelope>
Optimization Services Protocol (OSP)
What does the protocol involve? – 20+ OSxL languages

*OSmL*: a modeling language and NOT an Optimization Services Protocol
*Leters not currently used*: w, z
Optimization System Background
What does an optimization system look like?

\[
\begin{align*}
\text{minimize} & \quad cx \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]
OUTLINE

1. Motivations

2. Optimization Services and Optimization Services Protocol

3. Future and Derived Research
Future and Derived Research

- The Optimization Services project
- Standardization
- Problem repository building
- OS server software, library enhancement
- Derived research in distributed systems (coordination, scheduling and congestion control)
- Derived research in decentralized systems (registration, discovery, analysis, control)
- Derived research in local systems (OSI? OSII, OSIII, OSIV?)
- Derived research in optimization servers (NEOS)
- Derived research in computational software (AMPL, Knitro, Lindo/Lingo, IMPACT, OSmL, MProbe, Dr. AMPL, etc.)
- Derived research in computational algorithm
  - Parallel computing

- Derived business model
  - Modeling language developers, solver developers, and analyzer developers
  - Library developers, registry/server developers, and other auxiliary developers
  - Computing on demand and “result on demand”
http://www.optimizationservices.org