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

Lindo System Inc. 05/18/2005
OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. 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, OSmL)

2. Solver
   (Too many, e.g. Lindo)

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
   (Lindo Black-Box Interface, COIN-OSI, CPLEX-Concert, AMPL/GAMS-Kestrel, etc.)

7. Low Level Instance Representation
   (Lindo’s Instruction List, and others on 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 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

M \times N drivers

M + N drivers
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. Demonstration

3. Optimization Services and Optimization Services Protocol

4. Optimization System Background

5. Computing and Distributed Background

6. Optimization Services Protocol - Representation

7. Optimization Services Protocol - Communication

8. Optimization Services Protocol - Registry

9. Optimization Services modeling Language (OSmL)

10. Future and Derived Research
XML-based standard

Optimization Services (OS)
What is happening behind?

Modeler

Model/Data

Parse to OSiL

OSmL

AMPL

Agent

OSP -- OSHL(OSiL)

Solver

Web address
OS Server

Web page

OS Server

location

Web Server

OS Server

Data in HTML Form

Analyzer

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(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)

Registry

Google

html form

socket

http/html

CGI

browser

Data in HTML Form

HTML Checker

Raw Data

Solver

Simulation

OS Server

OS Server

Database/App Service

Jun Ma, Optimization Services, May 18, 2005
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
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

HTTP

TCP

IP

Ethernet

Application

Presentation

Session

Transport

Network

Link

Physical

SOAP

OSP

GET /xt/services/ColorRequest HTTP/1.0
Content-Length: 442
Host: localhost
Content-type: text/xml; charset=utf-8
SOAPAction: " getColor"

<soap:Envelope>
  <soap:Body>
    String solve(String instance)
    input string instance follow OSIIL
    -- output string follow OSrIL
  </soap:Body>
</soap:Envelope>

SOAP is usually wrapped under HTTP
Optimization Services Protocol (OSP)

What does the protocol involve? – 20+ OSxL languages

*OSmL*: a modeling language and NOT an Optimization Services Protocol

*Letters not currently used*: w, z

OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research
Optimization System Background

What does an optimization system look like?

users

developers

modelers

minimize $cx$

subject to $Ax = b$

$x \geq 0$
Optimization System Background

What is the difference between a model and an instance?

**Model:**
- High-level, user-friendly
- Symbolic, general
- Concise, understandable

**Instance:**
- Low-level, computer-friendly
- Explicit, specific
- Redundant, convenient

### Model
```
set  MUTE  ordered;
set FOOD  ordered;

param cost (FOOD) >= 0;
param f_min (MUTE) >= 0, default 0;
param f_max (in  FOOD) := f_max[i], default Infinity;
param n_min (MUTE) := n_min, default Infinity;
param n_max (MUTE, FOOD) >= 0;

var  Buy  (j in  FOOD) := x_min[j], >= f_min[j];
```

```
  minimize Totnal Cost: sum (j in  FOOD) cost[j] * Buy[j];
```

```
subject to Dist (i in MUTE):
  n_min[i] <= sum (j in  FOOD) x[j] * Buy[j] <= n_max[i];
```

```
minimize  c'x
subject to  Ax = b
x >= 0
```

### Instance
```
```
```
```
```
```
```
```
Optimization System Background

What's the difference between local interfacing and communication agent
Optimization System Background

Why is analyzer important?
Optimization System Background

What’s the difference between a server and a registry
Optimization System Background

What’s a simulation?

minimize  \( x_1^2 + 2x_2^2 \)
subject to  \( 2x_1 + 3x_2 \geq 9 \)
\( x_1 \geq 0, x_2 \geq 0 \)

\[ \text{mySimulation}\{ \]
\[ \text{address} = \text{http://somesite.com/mySimulation} \]
\[ \text{input} : \]
\[ a \]
\[ b \]
\[ c \]
\[ \text{output} : \]
\[ \text{value} + \text{confidence} * 0 \]
\[ \} \]
Optimization System Background

AMPL, NEOS and Kestrel

AMPL: option optimizationservices on

AMPL: solve;
Optimization System Background

Motorola Optimization System
OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research
Computing and Distributed Background

What we used in our implementation

1. Java, Open Source Libraries, Object-oriented Programming (OS library)

2. Networking Protocols: HTTP, SOAP, OSP
   (OS server: Tomcat, Axis, OS library)

3. Eclipse IDE for JAVA development

4. XML Spy for XML Schema design
Computing and Distributed Background

XML and XML Dialect (e.g. MathML, OSiL)

MathML

\[(2X_1 + 3X_2)^2\]

OSiL

\[(2X_1 + 3X_2)^2\]
Computing and Distributed Background

**XML Schema**

\[
\begin{align*}
\text{minimize} & \quad 100(x_1 - x_0^2)^2 + (1 - x_0)^2 + 7x_1 \\
\text{subject to} & \quad x_0 + 7x_1 \leq 10 \\
& \quad \ln(x_0x_1) + 7x_0 + 5x_1 \leq 10 \\
& \quad x_0, x_1 \geq 0
\end{align*}
\]

```xml
<xs:element name="variables">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="var" type="var" minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:complexType name="var">
    <xs:attribute name="name" type="xs:string" use="optional"/>
    <xs:attribute name="init" type="xs:string" use="optional"/>
    <xs:attribute name="type" use="optional" default="C">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="C"/>
                <xs:enumeration value="B"/>
                <xs:enumeration value="I"/>
                <xs:enumeration value="S"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="lb" type="xs:double" use="optional" default="0"/>
    <xs:attribute name="ub" type="xs:double" use="optional" default="INF"/>
    <xs:attribute name="objCoef" type="xs:double" use="optional" default="0.0"/>
    <xs:attribute name="mul" type="xs:positiveInteger" use="optional" default="1"/>
</xs:complexType>
```

"sequence of"
Computing and Distributed Background

Other XML Technologies

1. Parsing: SAX and DOM models

2. Transformation: XSL style sheet

3. Lookup: XPath and XQuery

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/">
    <html>
      <body>
        <hr/>
        <h1>Stocks</h1>
        <p/>
        <xsl:for-each select="stocks/stock">
          <p>stock: <xsl:value-of select="@name"/>
        </xsl:for-each>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```
Computing and Distributed Background

Web services

- **Platform and implementation independent** components
- **Described** using a service description language (WSDL)
- **Published** to a registry of services (UDDI, OS Registry)
- **Discovered** through a standard mechanism (UDDI, OS Registry)
- **Invoked** through a declared API (SOAP)
- **Composed** with other services (SOAP)
Computing and Distributed Background

Web services and SOAP

Architecture View

Web Service

HTTP

TCP/IP

Protocol View

SOAP

HTTP

SOAP

Contains call and response information

SOAP Envelope

SOAP Header

SOAP Body

Payload Document(s)

SOAP Fault

Java application

ANY client!

Java-Structure

SOAP Server

SOAP

VB application

VB-Structure

SOAP client

SOAP Message

POST /services/VersionRequest HTTP/1.0
Content-Length: 123
Host: http://user.ims.nwu.edu/
Content-type: text/xml; charset=utf-8
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope ...>
  <soap:Body>
    <m:SolverVersionRequestMsg
      xmlns:m="http://www.optimizationservices.org/soap-methods">
      <question xsi:type="xsd:string">
        What is the version of the IMPACT MINLP solver?
      </question>
    </m:SolverVersionRequestMsg>
  </soap:Body>
</soap:Envelope>
Computing and Distributed Background

Web services and WSDL

Operations

String getJobID()

String solve (String, String)

OSiL  OSiL, OSoL

String solve (String)

OSiL  OSiL

String retrieveResult (String)

OSiL  JobID

String analyze(String)

OSaL  OSiL
Computing and Distributed Background

Web services and WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
  
  <wsdl:message name="solve">
    <wsdl:part name="result" element="http://www.optimizationservices.org/OS/optimization/"/>
  </wsdl:message>

  <wsdl:portType name="OptimizationSolverServiceSoap">
    <wsdl:operation name="solve">  
      <wsdl:input>
        <wsdl:element name="input" type="http://www.optimizationservices.org/OS/optimization/"/>
      </wsdl:input>
      <wsdl:output>
        <wsdl:element name="output" type="http://www.optimizationservices.org/OS/optimization/"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="OptimizationSolverServiceSoap" type="os:OptimizationSolverService">
    <wsdl:operation name="solve">  
      <wsdl:input>
        <wsdl:element name="input" type="http://www.optimizationservices.org/OS/optimization/"/>
      </wsdl:input>
      <wsdl:output>
        <wsdl:element name="output" type="http://www.optimizationservices.org/OS/optimization/"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  
  <wsdl:service name="OptimizationSolverService">
    <wsdl:port name="OptimizationSolverService" binding="os:OptimizationSolverServiceSoapBinding">  
      <wsdl:operation name="solve">  
        <wsdl:input>
          <wsdl:element name="input" type="http://www.optimizationservices.org/OS/optimization/"/>
        </wsdl:input>
        <wsdl:output>
          <wsdl:element name="output" type="http://www.optimizationservices.org/OS/optimization/"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

"solve" operation is wrapped in a soap envelope over the http protocol and using rpc style.

The element should be empty.
Read the comments in <!-- comments -->.
OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research
Optimization Services Representation

Who else did this before?

• Many “standards”
• All limited to problem input
• Highly fragmented and no general format
• Fourer, Lopes, and Martin’s LPFML (OSIL)
• Kristjánsson’s OptML
• Bradley’s NaGML
• We are the first in designing
  - Systematic representation of major optimization types
  - All major instance types (result, analysis, input, query, etc.)
  - Web services (SOAP) based communication standards
  - Optimization registry
  - A universal framework
Optimization Services Representation
Optimization Services general Language (OSgL)
General data structures; Included in other schemas

Optimization Services instance Language (OSiL)
- Linear
- Mixed integer
- Bound constrained optimization
- General quadratic optimization
- Nonlinear unconstrained/constrained
- General mixed integer nonlinear
- General nonlinear with user-defined functions
- Global optimization
- General nonlinear with simulations (black-box functions)
- Optimization over simulation/nondifferentiable optimization
- General nonlinear with xml data (either within OSiL or remotely located)
- General nonlinear with data look up (XPath)
- Network and graph definition
- Network programming
- Constraint programming
- Semidefinite programming
- Semi-infinite programming
- Cone programming
- Complementarity problems
- Stochastic linear/nonlinear (distribution based recourse problem, scenario based recourse problem, chance constrained)
- Combinatorial optimization/Heuristic Optimization (TSP, MST, SP, MF, MCF, VRP, Set Covering, Coloring etc. etc.)
OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research
Optimization Services Communication

Optimization Services hookup Language (OShL)

Hookup to solvers, and analyzers

Operations
OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background

6. Optimization Services Protocol - Representation

7. Optimization Services Protocol - Communication

8. Optimization Services Protocol - Registry

9. Optimization Services modeling Language (OSmL)

10. Future and Derived Research
Optimization Services Registry

- **OSyL** XML database; native XML; A sequence of [OSeL, OSpL, OSbL] triplets
- **OSjL** Join the OS registry (manual and automatic process) with entity (OSeL) information.
- **OSkL** "Knock" at the services for runtime dynamic process (OSpL) information
Optimization Services Registry

Optimization Services Query Language (OSqL, representation)
Like SQL for relational database; can use XQuery, OSaL (analysis), predefined

Optimization Services Discover Language (OSdL, communication)
Send the query to the OS registry to discover services

Optimization Services URI Language (OSuL, representation)
A sequence of URI (URL) addresses for service locations with degree of fitness

Optimization Services Validate Language (OSvL, validate)
A validation service provided by the OS registry that validates all OSxL instances

---

<?xml version="1.0" encoding="UTF-8"?>
<OSuL xmlns="os.optimizationservices.org" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="os.optimizationservices.org http://www.optimizationservices.org/schemas/OSuL.xsd">
  <uri>http://www.abc.com/lp/solver.jws</uri>
  <uri match="exact">http://www.pdf.net/other/service.web</uri>
  <uri match="moreCommon">http://www.pdf.net/other/service.web</uri>
  <uri match="approx">http://www.pdf.net/other/service.web</uri>
  <uri match="guess">http://www.pdf.net/other/service.web</uri>
</OSuL>

---

(String)
validate (String)
error Message
OSxL

---

<standard>
  <entity>
    <service>
      <key>Keywords</key>
      <value>interior point method</value>
      <key>key</key>
      <value>convex programming</value>
    </service>
    <optimizationType>
      <variableType>mixedInteger</variableType>
      <optimizationType>
    </entity>
  </standard>
</OSxL>
OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry

9. Optimization Services modeling Language (OSmL)

10. Future and Derived Research
Optimization Services modeling Language (OSmL)

A derived research

Open source and general purpose
Standard based (XQuery input; OSiL output)
Suitable for distributed optimization

XML data is ubiquitous

```
return
<mathProgram>
    <obj maxOrMin="min" name="Rosenbrock">
        100*(x2 - x1^2)^2 + (1 - x1)^2
    </obj>
    <constraints>
        <con>
            x1 + x2 <= 100
        </con>
    </constraints>
</mathProgram>
```
# Optimization Services modeling Language

**XML Representation**

```xml
<AMPL>
  <set name="PROD"> set PROD; </set>
  <set name="LINKS"> set LINKS = {PROD, 1..T}; </set>
  <param name="HC"> param HC {PROD} ; </param>
  <param name="FXC"> param FXC {PROD} ; </param>
  <param name="CAP"> param CAP {1..T} ; </param>
  <param name="DEM"> param DEM {LINKS} ; </param>
  <param name="PCOST"> param PCOST {PROD, 1..T} ; </param>

  <variable name="x"> var x {PROD, 1..T} >= 0; </variable>
  <variable name="I"> var I {PROD, 0..T} >=0; </variable>
  <variable name="y"> var y {PROD, 1..T}binary; </variable>

  <objective name="Total_Cost"> minimize Total_Cost: </objective>
  <subject to="Init_Inv"> subject to Init_Inv {i in PROD}: I[i,0] = 0.0; </subject>
  <subject to="Balance"> subject to Balance {i in PROD, t in 1..T}: x[i, t] + I[i, t - 1] - I[i, t] = DEM[i, t]; </subject>
  <subject to="Fixed_Charge"> subject to Fixed_Charge {i in PROD, t in 1..T}: x[i, t] <= CAP[t]*y[i, t]; </subject>
  <subject to="Capacity"> subject to Capacity {t in 1..T}: sum {i in PROD} x[i, t] <= CAP[t]; </subject>
</AMPL>
```

**OSmL Representation**

```
<mathProgram>
  (: VARIABLE DECLARATION :) <variables>{ for $i in (1 to $N),  $t in (1 to $T) return (<var name="X[{$i},{$t}]"/>, <var name="I[{$i},{$t}]"/>, <var name="Y[{$i},{$t}]" type="B"/>) }
  (: OBJECTIVE FUNCTION   :) <obj maxOrMin="min" name="Total_Cost"> SUM(for $i in (1 to $N), $t in (1 to $T) return {$PCOST[{$i}]>*X[{$i},{$t}]+{$FXC[{$i}]>*Y[{$i},{$t}]+{$HC[{$i}]>*I[{$i},{$t}]})</obj> <constraints> (: INITIAL INVENTORY CONSTRAINTS :) {for $i in $PROD return <con name="inventory[{$i}]"> I[{$i},0]  = 0 </con>  }
  (: DEMAND CONSTRAINTS :) {for $i in $PROD,  $t in (1 to $T) let $demand := ($products/$i[@periodID=$t]/demand/text()) return <con name="demand[{$i},{$t }]">  X[{$i},{$t}] + I[{$ i},{$t - 1}] - I[{$i},{$t}] = {$demand} </con>  }
  (: FIXED CHARGE CONSTRAINTS :) {for $t in (1 to $T),  $i in (1 to $N) return <con name="fixed_charge[{$i},{$t }]" > X[{$i},{$t}]-{$CAP[{$t}]}*Y[{$i},{$t}] <= 0</con>  }
  (:  CAPACITY CONSTRAINTS  :) {for $t in (1 to $T) return <con name="capacity[{$t}]">  SUM(for $i in (1 to $N) return  X[{$i},{$t}])<= {$CAP[{$t}]} </con>}
</constraints> </mathProgram>
```
Optimization Services modeling Language

4 ways of combining XML with optimization

1. Use XML to represent the instance of a mathematical program
2. Develop an XML modeling language dialect
3. Enhance modeling languages with XML features such as XPath
4. Use XML technologies to transform XML data into a problem instance
OUTLINE

1. Motivations

2. Demonstration

3. Optimization Services and Optimization Services Protocol

4. Optimization System Background

5. Computing and Distributed Background

6. Optimization Services Protocol - Representation

7. Optimization Services Protocol - Communication

8. Optimization Services Protocol - Registry

9. Optimization Services modeling Language (OSmL)

10. Future and Derived Research
Derived Research and Business Models

- 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, OSrl, OSol?)
- 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

Library developers, registry/server developers, and other auxiliary developers
Computing on demand and “result on demand”
End

http://www.optimizationservices.org (.net)