COIN-OR and the COIN-OR Optimization Suite

Ted Ralphs

COIN fORgery: Developing Open Source Tools for OR

Institute for Mathematics and Its Applications, Minneapolis, MN
First, thanks to the IMA for their generous support of this workshop!
And thank you for being here!!
This will be a wide-ranging workshop with an equally wide-ranging audience.
The approach is not like an academic conference, we’re really going for engagement, active feedback, and spirited discussion.
We have tried to focus the material for the audience, but feel free to let us know if this is not the case—there is still time to adjust.
The schedule includes
- Structured talks in the mornings
- Lunch discussion groups
- Hands-on sessions in the afternoon
- (Mostly) informal socializing/networking in the evening

Let’s Go!
Goals of the Workshop

There are several goals underlying this workshop.

- Provide tutorials and hands-on coding sessions for developers interested in using our tools and contributing.
- Recruit some new volunteers for a range of important tasks that need to be to move things forward.
- Discuss how the COIN-OR Foundation itself can grow and move forward.
- Discuss how the COIN-OR Optimization Suite and its aging code base can be effectively maintained and developed going forward.
- Plan for the coming year of coding sprints!

Our hope is that there will be many questions, diversions, discussions, etc.

Please jump in at any time!

We can even add/delete topics and adjust the schedule as appropriate.
1. The COIN-OR Foundation

2. Overview of Projects

3. Overview of the Optimization Suite
Many of the papers being written in OR today are **computational** in nature or have a computational component.

Historically, the pace of computational research has been relatively slow and the transfer of knowledge to practitioners has been even slower.

Results of computational research are still generally **not reproducible**.

Research codes tend to be buggy, narrowly focused, and lacking robustness/generality.

There have been **few rewards for publishing software** outside of archival journals.

There has been **no peer review process** for software and referees of computational papers have generally had little to go on.

Building on previous results is difficult and time-consuming.

**Interoperating** with other software libraries (such as LP solvers) is difficult.

The paradigm encouraged by archival journals does not work well for computational research.
The Common Optimization Interface for Operations Research Initiative was an initiative launched by IBM at ISMP in 2000.

IBM seeded an open source repository with four initial projects and created a Web site.

The goal was to develop the project and then hand it over to the community.

The project grew to be self-sustaining and was spun off as a nonprofit educational foundation in the U.S. more than a decade ago.

The name was also changed to the Computational Infrastructure for Operations Research to reflect a broader mission.
What is COIN-OR Today?

The COIN-OR Foundation

- A **non-profit foundation** promoting the development and use of interoperable, open-source software for operations research.
- A **consortium** of researchers in both industry and academia dedicated to improving the state of computational research in OR.
- A **venue** for developing and maintaining standards.
- A **forum** for discussion and interaction between practitioners and researchers.

The COIN-OR Repository

- A **collection** of interoperable software tools for building optimization codes, as well as a few stand alone packages.
- A **venue for peer review** of OR software tools.
- A **development platform** for open source projects, including a wide range of project management tools.
The COIN Boards

The COIN-OR Foundation is governed by two boards.

<table>
<thead>
<tr>
<th>Strategic Leadership Board</th>
<th>Technical Leadership Council</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robert Fourer</td>
<td>Ted Ralphs (Chair)</td>
</tr>
<tr>
<td>Kevin Furman</td>
<td>Haroldo Santos</td>
</tr>
<tr>
<td>Horand (Gus) Gassmann (Treasurer)</td>
<td>John Siirola</td>
</tr>
<tr>
<td>Bill Hart</td>
<td>Mike Steglich</td>
</tr>
<tr>
<td>Alan King (Treasurer)</td>
<td>Stefan Vigerske</td>
</tr>
<tr>
<td>Andrew Mason</td>
<td></td>
</tr>
<tr>
<td>Giacomo Nannicini (Secretary)</td>
<td></td>
</tr>
<tr>
<td>Ted Ralphs (TLC Rep)</td>
<td></td>
</tr>
<tr>
<td>Matt Saltzman (President)</td>
<td></td>
</tr>
</tbody>
</table>

- The SLB sets the overall strategic direction and manages the business operations: budgeting, fund-raising, legal, etc.
- The TLC focuses on technical issues: build system, versioning system, bug reporting, interoperability, etc.
A Little Historical Perspective

- **Early years**
  - Most projects were offshoots of or integrated with the original IBM codes.
  - These codes formed the basis for what is now the Optimization Suite.
  - There were a large number of developers, including some that were full-time developers at IBM.
  - IBM provided both in-kind and monetary support.
  - Developer effort was focused mainly on low-level tools.

- **Recent years**
  - Shift to development of high-level tools (modeling, etc.).
  - Most active projects are stand-alone and don’t use the technical infrastructure (build tools, etc.)
  - The number of active developers of the Optimization Suite has dropped.
  - We are struggling to support these tools on which many people and organizations depend.

- **We would like to hear your opinion about this situation throughout the week.**
Evolution of the Foundation’s Role

In the early years, the foundation promised to provide many free tools to developers that were not readily available on-line.

- Centralized version control (CVS/SVN)
- Integrated Wiki, issue tracking, source browser (TRAC)
- Mailing lists
- Web space
- Continuous integration and testing (Jenkins)
- Distribution of binaries

All of these things are now readily available for free and the role of the foundation is not as clear.

Possible future roles

- Publisher of peer-reviewed software and a promoter of the importance of software publication.
- Educational resource for new developers and users.
- Developer of standards, best practices, etc.
- Focal point for combined efforts to obtain funding/support for development of open source tools?
- Developer/maintainer of aging but still heavily used codes?
An Evolving Discussion

- The current trends have been evident for some time.
- The future direction of COIN-OR has been an on-going topic of discussion for many years already.
- What should the future of the foundation be?
- Is there a role for supporting on-going development and maintainance for existing codes?
- Should we re-organize and re-focus?
- We would like to hear your opinions in the coming days.
How is COIN-OR Supported Today?

[Image of a PayPal receipt]

You received a payment

Hello COIN-OR Foundation, Inc,
You received a payment from [Redacted] for Donation

Customer details

Customer name: [Redacted]
Customer email: [Redacted]
Profile ID: [Redacted]
Profile status: Active

Subscription details

Amount received: $0.01 USD
For: Donation
Amount paid each time: $0.01 USD
Maximum amount you can bill: $0.01 USD
Billing cycle: Monthly
Next payment due: Jun 2, 2013
We currently have 50+ projects and more are being added all the time.

Most projects are now licensed under the EPL (very permissive).

COIN-OR has solvers for most common optimization problem classes.
- Linear programming
- Nonlinear programming
- Mixed integer linear programming
- Mixed integer nonlinear programming (convex and nonconvex)
- Stochastic linear programming
- Semidefinite programming
- Graph problems
- Combinatorial problems (VRP, TSP, SPP, etc.)

COIN-OR has various utilities for reading/building/manipulating/preprocessing optimization models and getting them into solvers.

COIN-OR has overarching frameworks that support implementation of broad algorithm classes.
- Parallel search
- Branch and cut (and price)
- Decomposition-based algorithms
What You Can Do With COIN-OR: High-level Tools

One of the most exciting developments of recent years is the number of high-level tools available to access COIN-OR solvers.

- Python-based modeling languages
- Spreadsheet modeling (!)
- Commercial modeling languages
- Mathematica
- Matlab
- R
- Sage
- Julia
- ...

COIN-OR isn’t just for breakfast anymore!
COIN-OR Projects: Linear Optimization

- **Clp**: COIN LP Solver
  
  **Project Manager**: Julian Hall

- **DyLP**: An implementation of the dynamic simplex method
  
  **Project Manager**: Lou Hafer

- **Cbc**: COIN Branch and Cut
  
  **Project Manager**: Ted Ralphs

- **SYMPHONY**: MILP solver framework that supports shared and distributed memory parallel processing, biobjective optimization, warm starting, sensitivity analysis, application development, etc.
  
  **Project Manager**: Ted Ralphs

- **BLIS**: Parallel IP solver built to test the scalability of the CHiPPS framework.
  
  **Project Manager**: Ted Ralphs

- **Cgl**: A library of cut generators
  
  **Project Manager**: Robin Lougee
COIN-OR Projects: Nonlinear Optimization

- Ipopt: Interior Point OPTimizer for nonlinear optimization problems.
  
  **Project Manager**: Andreas Wächter

- DFO: An algorithm for derivative free optimization.
  
  **Project Manager**: Katya Scheinberg

- CSDP: A solver for semi-definite programs
  
  **Project Manager**: Brian Borchers

- OBOE: Oracle based optimization engine
  
  **Project Manager**: Nidhi Sawhney

- FilterSD: Package for linearly constrained non-linear optimization problems.
  
  **Project Manager**: Frank Curtis

- OptiML: Optimization for machine learning, interior point, active set method.
  
  **Project Manager**: Katya Scheinberg

- qpOASES: QP solver using the active online set strategy.
  
  **Project Manager**: Joachim Ferreau
- **oBB**: Parallel global optimization of Hessian Lipschitz continuous functions.
  
  **Project Manager**: Jaroslav Fowkes

- **RBFOpt**: A global derivative-free solver.

  **Project Manager**: Giacomo Nannicini
Bonmin: Basic Open-source Nonlinear Mixed INteger programming is for (convex) nonlinear integer programming.
  Project Manager: Pierre Bonami

Couenne: Solver for nonconvex nonlinear integer programming problems.
  Project Manager: Pietro Belotti

LaGO: Lagrangian Global Optimizer, for the global optimization of nonconvex mixed-integer nonlinear programs.
  Project Manager: Stefan Vigerske

DisCO: Discrete Conic Optimization, a solver and framework for solving mixed integer second-order conic optimization problems.
  Project Manager: Aykut Bulut

MibS: A solver for mixed integer bilevel optimization problems.
  Project Manager: Ted Ralphs

SHOT: A deterministic convex MINLP solver based on polyhedral outer approximation and primal heuristics.
  Project Manager: Andreas Lundell
COIN-OR Projects: Modeling

- **FLOPC++**: An open-source modeling system.
  - Project Manager: Tim Hultberg
- **Pyomo**: A repository of python-based modeling tools.
  - Project Manager: Bill Hart
- **PuLP**: Another python-based modeling language.
  - Project Manager: Stu Mitchell
- **DipPy**: A python-based modeling language for decomposition-based solvers.
  - Project Manager: Mike O’Sullivan
- **CMPL**: An algebraic modeling language
  - Project Manager: Mike Stieglich
- **SMI**: Stochastic Modeling Interface, for optimization under uncertainty.
  - Project Manager: Alan King
- **yaposib**: Yet Another Python OSI Binding.
  - Project Manager: Ted Ralphs
- **CyLP**: Python interface to Cbc and Clp.
  - Project Manager: Ted Ralphs (?)

Project Manager: Hassan Hijazi


Project Manager: Juan F. Perez

Rehearse: An algebraic modeling library in C++.

Project Manager: Onur Celebi
COIN-OR Projects: Interfaces and Solver Links

- **Osi**: Open solver interface is a generic API for linear and mixed integer linear programs.
  
  **Project Manager**: Matthew Saltzman

- **GAMSlinks**: Allows you to use the GAMS algebraic modeling language and call COIN-OR solvers.
  
  **Project Manager**: Stefan Vigerske

- **AIMMSlinks**: Allows you to use the AIMMS modeling system and call COIN-OR solvers.
  
  **Project Manager**: Marcel Hunting

- **MSFlinks**: Allows you to call COIN-OR solvers through Microsoft Solver Foundation.
  
  **Project Manager**: Lou Hafer

- **CoinMP**: A callable library that wraps around CLP and CBC, providing an API similar to CPLEX, XPRESS, Gurobi, etc.
  
  **Project Manager**: Bjarni Kristjansson

- **Optimization Services**: Framework providing data interchange formats and tools for calling solvers locally and remotely through Web services.
  
  **Project Managers**: Jun Ma, Gus Gassmann, and Kipp Martin
COIN-OR Projects: Frameworks

- **ABACUS**: An LP-based branch-and-cut framework.
  - **Project Manager**: Frank Baumann, Mark Sprenger

- **Bcp**: A generic framework for implementing branch, cut, and price algorithms.
  - **Project Manager**: Laci Ladanyi

- **CHiPPS**: A framework for developing parallel tree search algorithms.
  - **Project Manager**: Ted Ralphs

- **DIP**: A framework for implementing decomposition-based algorithms for integer programming, including Dantzig-Wolfe, Lagrangian relaxation, cutting plane, and combinations.
  - **Project Manager**: Ted Ralphs
COIN-OR Projects: Automatic Differentiation

- **ADOL-C**: Package for the automatic differentiation of C and C++ programs.
  
  **Project Manager**: Andrea Walther

- **CppAD**: A tool for differentiation of C++ functions.
  
  **Project Manager**: Brad Bell
COIN-OR Projects: Graphs

- **GiMPy and GrUMPy**: Python packages for visualizing algorithms  
  Project Manager: Ted Ralphs

- **Cgc**: Coin graph class utilities, etc.  
  Project Manager: Phil Walton

- **LEMON**: Library of Efficient Models and Optimization in Networks  
  Project Manager: Alpar Juttner
**Djinni**: C++ framework with Python bindings for heuristic search  
  *Project Manager*: Justin Goodson

**METSlib**: An object oriented metaheuristics optimization framework and toolkit in C++  
  *Project Manager*: Mirko Maischberger

**CoinBazaar**: A collection of examples, application codes, utilities, etc.  
  *Project Manager*: Bill Hart

**PFunc**: Parallel Functions, a lightweight and portable library that provides C and C++ APIs to express task parallelism  
  *Project Manager*: Prabhanjan Kambadur

**ROSE**: Reformulation-Optimization Software Engine, software for performing symbolic reformulations to Mathematical Programs (MP)  
  *Project Manager*: David Savourey

**MOCHA**: Matroid Optimization: Combinatorial Heuristics and Algorithms, heuristics and algorithms for multicriteria matroid optimization  
  *Project Manager*: David Hawes
COIN-OR Projects: Miscellaneous

- **Créme**: Randomized thermal relaxation for finding a feasible solution of the Maximum Feasible Subsystem problem.
  - **Project Manager**: Pietro Belotti

- **jORLib**: Java library that provides algorithmic implementations and frameworks for optimization problems in the area of Operations Research.
  - **Project Manager**: Joris Kinable

- **MC++**: Toolkit for bounding factorable functions.
  - **Project Manager**: Benoit Chachuat

- **Paver**: Python scripts to do comparisons of solver performance.
  - **Project Manager**: Stefan Vigerske

- **QAPSolver**: Solver for quadratic assignment problems
  - **Project Manager**: Peter Hahn
1. The COIN-OR Foundation

2. Overview of Projects

3. Overview of the Optimization Suite
Many of the tools in the repository that are focused on solution of mathematical optimization models are inter-related.

They are built from a common underlying set of tools in a hierarchical fashion using a common build harness.

The COIN-OR Optimization Suite is an umbrella project that consists of compatible version of all these mutually interoperable projects.

This suite will be the focus of the remainder of the tutorial.
One of the hallmarks of good open source tools is **modularity**.

The suite is made up of building blocks with well-defined interfaces that allow construction of higher level tools.

There have been 75 authors over time and most have never coordinated directly with each other!

This is the open source model of development.
The CoinUtils project contains a wide range of low-level utilities used in almost every project in suite.

- Factorization
- File parsing
- Sparse matrix and array storage
- Presolve
- Memory management
- Model building
- Parameter parsing
- Timing
- Basic data structures
Basic Building Blocks: Open Solver Interface (OSI)

Uniform API for a variety of solvers:

- Cbc
- Clp
- CPLEX
- DyLP
- FortMP
- Xpress
- GLPK
- Mosek
- OSL
- Soplex
- SYMPHONY
- Volume

- Read input from MPS or LP format files or construct instances using COIN-OR data structures.
- Manipulate instances and output to MPS or LP file.
- Set solver parameters.
- Calls LP solver for LP or MIP LP relaxation.
- Manages interaction with dynamic cut and column generators.
- Calls MIP solver.
- Returns solution and status information.
Building Blocks: Cut Generator Library (CGL)

- A collection of cutting-plane generators and management utilities.
- Interacts with OSI to inspect problem instance and solution information and get violated cuts.
- Cuts include:
  - Combinatorial cuts: AllDifferent, Clique, KnapsackCover, OddHole
  - Flow cover cuts
  - Lift-and-project cuts
  - Mixed integer rounding cuts
  - General strengthening: DuplicateRows, Preprocessing, Probing, SimpleRounding
Building Blocks: Frameworks

- **CHiPPS (COIN High Performance Parallel Search)** is a library hierarchy for implementing parallel tree search.
  - **ALPS** (Abstract Library for Parallel Search) is a collection of abstract base classes for implementing basic (parallel) tree search.
  - **BiCePS (Branch, Constrain, and Price Software)** implements a range of abstract base classes needed for implementing relaxation-based branch-and-bound.

- **Bcp** is a low-level framework for implementing branch, cut, and price algorithms.

- **Dip (Decomposition for Integer Programming)** is a framework for automatic decomposition and decomposition-based algorithms (Lagrangian relaxation, Dantzig-Wolfe decomposition) built on ALPS.
**Base Solvers**

- **Clp** is the LP solver on which the rest of the Optimization Suite is built.
- **Cbc** is the MILP solver built on **Clp** that underpins many other projects.
- **Ipopt** is the core NLP solver that is also

Anecdotally, these three projects are very widely used in both government, academia, and industry.

- These projects are also embedded in many other applications, such as OpenOffice,
- They are distributed with modeling system, such as AMPL, MPL, AIMMS, etc.
- They can be called from R, Matlab, Sage, etc.

- **SYMPHONY** and **DyLP** are experimental LP and MIP solvers, respectively.
Higher Level Projects

- Solvers built on **CHiPPS**
  - **BLIS** (MILPs)
  - **DisCO** (MISOCP)
  - **MibS** (MIBLP)
  - **Dip** (MILP)

- Solvers built on **Cbc** and **Ipopt**.
  - **Couenne** (MINLP)
  - **Bonmin** (MINLP)
Modeling Tools and Interface

- **OS** (Optimization Services)
- **OSI** (Open Solver Interface)
- **SMI** (Stochastic Modeling Interface)

Python interfaces

- Pyomo
- PuLP/DipPy
- yaposib
- CyLP