- 線形行列不等式 $\{(X,Y) \in \mathcal{S}^n \times \mathcal{S}^\ell : \ AXA^T + Y \succeq O\}$

- 凸 2 次行列不等式

  $\{(X,Y) \in \mathcal{S}^n \times \mathcal{S}^\ell : \ (AXB)(AXB)^T + CXD + (CXD)^T + E \preceq Y\}$

$$\begin{pmatrix} I & (AXB)^T \\ AXB & Y - E - CXD - (CXD)^T \end{pmatrix} \succeq O$$

- 分数 2 次行列不等式 $\{(X,Y) \in \mathcal{S}^n \times \mathcal{S}^\ell : \ XV^{-1}X^T \preceq Y\}$

$$\begin{pmatrix} V & X^T \\ X & Y \end{pmatrix} \succeq O$$

5.3.2 多項式の 2 乗和分解

- 偶数次数の多項式が 2 乗和分解できるかどうかを調べる

- 例： $2x^4 + 2x^3y - x^2y^2 + 5y^4$

$u = (1, x, y, x^2, y^2, xy)^T$ とすると

$$2x^4 + 2x^3y - x^2y^2 + 5y^4 =$$

$$\langle C, uu^T \rangle = \left\langle \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & c_{45} & c_{46} \\ 0 & 0 & 0 & c_{45} & c_{55} & 0 \\ 0 & 0 & 0 & c_{46} & 0 & c_{66} \end{pmatrix}, \begin{pmatrix} 1 & x & y & x^2 & y^2 & xy \\ x & x^2 & xy & x^3 & xy^2 & x^2y \\ y & xy & y^2 & x^2y & y^3 & xy^2 \\ x^2 & x^3 & x^2y & x^4 & x^2y^2 & x^3y \\ y^2 & xy^2 & y^3 & x^2y^2 & y^4 & xy^3 \\ xy & x^2y & xy^2 & x^3y & xy^3 & x^2y^2 \end{pmatrix} \right\rangle$$

と

$$c_{44} = 2,\ 2c_{45} + c_{66} = -1,\ 2c_{46} = 2,\ c_{55} = 5 \tag{5.4}$$

を満たす行列が求められる.

上記の行列で 0 に固定されている要素は必ずしも 0 でなくてもよい.

- [BEN-TAL2001] などに書かれてある結果から例の多項式が 2 乗和で書ける為の必要十分条件が (5.4) を満たす半正定値行列 $C \succeq O$ が存在することがいえる.

- その一つの解として

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & -3 & 1 \\ 0 & 0 & 0 & -3 & 5 & 0 \\ 0 & 0 & 0 & 1 & 0 & 5 \end{pmatrix}$$

があり，非ゼロ要素からなる右下の 3×3 行列の分解を考えると

$$\begin{pmatrix} 2 & -3 & 1 \\ -3 & 5 & 0 \\ 1 & 0 & 5 \end{pmatrix} = \left[\frac{1}{\sqrt{2}} \begin{pmatrix} 2 & 0 \\ -3 & 1 \\ 1 & 3 \end{pmatrix} \right] \left[\frac{1}{\sqrt{2}} \begin{pmatrix} 2 & 0 \\ -3 & 1 \\ 1 & 3 \end{pmatrix} \right]^T = L^T L$$

となるので，

$$\begin{aligned} & 2x^4 + 2x^3y - x^2y^2 + 5y^4 \\ = \ & \langle L(x^2, y^2, xy)^T, L(x^2, y^2, xy)^T \rangle \\ = \ & \frac{1}{2}(2x^2 - 3y^2 + xy)^2 + \frac{1}{2}(y^2 + 3xy)^2 \end{aligned}$$

2 乗和として分解できる．

5.3.3 半正定値計画問題

半正定値計画問題

$$\begin{cases} \text{最小化：} & \langle C, X \rangle \\ \text{条件：} & \langle A_i, X \rangle = b_i \quad i = 1, 2, \ldots, m \\ & X \succeq O \qquad (\Leftrightarrow X \in \mathcal{S}^n_+) \end{cases}$$

$\Updownarrow$ 行列 $X \in \mathcal{S}^n$ をベクトル $x \in \mathbb{R}^{n(n+1)/2}$ とみなす

半正定値計画問題

$$\begin{cases} \text{最小化：} & c^T x \\ \text{条件：} & Ax = b \\ & x \in \mathcal{S}^n_+ \end{cases}$$

- 半正定値計画問題は 2 次錐計画問題を特殊ケースとして持つ
- 主双対内点法が一番効率の良い解法
- 小・中規模の問題ではかなり高速であるが，大規模な問題ではかなりアルゴリズムの工夫が必要
- Euclid 的 Jordan 代数を用いて LP, SOCP, SDP 等をまとめて議論ができる (複素数も含む)

応用例

- システムと制御
- 組み合わせ最適化問題の緩和
- 連続最適化問題の緩和 (多項式最適化問題, 非凸 2 次計画問題)
- データマイニング
- センサーネットワーク
- 構造計算
- 量子化学
- 量子力学から派生する情報理論の問題

5.4 Copositive計画問題(Copositive Programming, CP)

次のような閉凸錐を定義する.

$$\boxed{\mathcal{C}^n = \{X \in \mathcal{S}^n \ : \ v^T X v \geq, \ \forall v \geq 0\}}$$

Copositive 行列錐

$$\boxed{(\mathcal{C}^n)^* = \left\{ \sum_{i=1}^{k} z_i z_i^T \in \mathcal{S}^n \ : \ \exists z_i \geq 0, \ i = 1, 2, \ldots, k \ (\leq n) \right\}}$$

Completely Positive 行列錐

- 一般に双対錐とは

$$(\mathcal{C}^n)^* = \{X \in \mathbb{R}^{n \times n} \ : \langle X, U \rangle \geq 0, \ \forall U \in \mathcal{C}^n\}$$

- 線形計画問題，２次錐計画問題や半正定値計画問題に登場してくる錐の双対はその錐そのものになる.

$$(\mathbb{R}^n_+)^* = \mathbb{R}^n_+, \quad (\mathcal{K}^n)^* = \mathcal{K}^n, \quad (\mathcal{S}^n_+)^* = \mathcal{S}^n_+$$

Copositive 計画問題

$$\begin{cases} \text{最小化：} & \langle C, X \rangle \\ \text{条件：} & \langle A_i, X \rangle = b_i, \ i = 1, 2, \ldots, m \\ & X \in \mathcal{C}^n \end{cases}$$

Completely Posivite 計画問題

$$\begin{cases} \text{最小化：} & \displaystyle\sum_{i=1}^{m} b_i y_i \\ \text{条件：} & \displaystyle\sum_{i=1}^{m} A_i y_i + S = C \\ & S \in (\mathcal{C}^n)^* \end{cases}$$

- 次の自明な包含関係が成り立つ：

$$(\mathcal{C}^n)^* \subseteq (\mathcal{S}_+^n)^* = \mathcal{S}_+^n \subseteq \mathcal{C}^n$$

- さらに $\mathcal{N}^n = \{X \in \mathcal{S}^n \ : \ X_{ij} \geq 0, \ i, j = 1, 2, \ldots, n\}$ と定義すると以下が成り立つ：

$$(\mathcal{C}^n)^* \subseteq (\mathcal{S}_+^n + \mathcal{N}^n)^* = \mathcal{S}_+^n \cap \mathcal{N}^n \subseteq (\mathcal{S}_+^n)^* = \mathcal{S}_+^n \subseteq (\mathcal{S}_+^n + \mathcal{N}^n) \subseteq \mathcal{C}^n$$

- その他、上記の包含関係の間に挟まる凸閉集合が幾つも知られている
- **Copositive** 計画問題と **Completely Posivite** 計画問題は凸集合上で凸関数（線形関数）を最小化するという観点からは凸計画問題である．
- $\mathbf{0-1}$ 整数非凸２次計画問題，２次割当て問題，最大安定集合問題，標準２次計画問題など一般に難しい問題をこの両者で定式化できることが知られている．
- しかし，与えられた対称行列が $\mathcal{C}^n$ もしくは $(\mathcal{C}^n)^*$ に含まれているかどうかということをチェックすることは $\mathcal{NP}$ 完全である．

5.5　演習問題

$\boxed{\text{問題 1}}$

　次の最小２乗問題

$$\text{最小化：}\ \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|^2$$

を凸２次計画問題として定式化せよ．

$\boxed{\text{問題 2}}$

　$\boldsymbol{x}_0 \in \mathbb{R}^n$ から超平面 $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}\}$ への最小距離を求める問題は次のような凸２次計画問題を解くことによって求まる．

$$\begin{cases} \text{最小化：} & \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_0)^T(\boldsymbol{x}-\boldsymbol{x}_0) \\ \text{条件：} & \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b} \end{cases}$$

但し，行列 $\boldsymbol{A}$ の行は線形独立であると仮定する．

　この時，最適解が $\boldsymbol{x}^* = \boldsymbol{x}_0 + \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0)$ であることを示せ．また特に $\boldsymbol{A}$ が１行しかない場合，最小距離は $|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0|/\|\boldsymbol{A}\|_2$ になることを示せ．

$\boxed{\text{問題 3}}$

　次の凸２次計画問題

$$\begin{cases} \text{最小化：} & \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{c}^T\boldsymbol{x} \\ \text{条件：} & \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}, \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{cases}$$

（ただし，Qは半正定値行列とする）を２次錐計画問題として定式化せよ．

$\boxed{\text{問題}4}$ ２次錐計画問題とはどのような問題か．線形計画問題と比較しながら説明せよ．またロバスト 最適化問題への応用について説明せよ．

$\boxed{\text{問題}5}$ 半正定値計画問題とはどのような問題か．線形計画問題と比較しながら説明せよ．どのような分野で応用されているか．

$\boxed{\text{問題}6}$

2次錐$\mathcal{K}^n$，半正定値行列錐$\mathcal{S}^n$，**copositive**行列錐$\mathcal{C}^n$，**completely positive**行列錐$(\mathcal{C}^n)^*$がそれぞれ凸閉錐であることを示せ．

Chapter 6

実践最適化：ソフトウェア，モデリング言語

6.1 Online ソルバー，ソフトウェア

6.1.1 NEOS server

- 最近ではわざわざ自前の最適化ソフトウェアを作成して問題を解く必要はなくなった
- インタネットを介して最適化問題が解ける

http://neos-server.org

Welcome to the NEOS Server.

Configuration changes today between 12PM and 1PM US Central. No downtime expected.

Our optimization solvers represent the state-of-the-art in optimization software. Optimization problems are solved automatically with minimal input from the user. Users only need a definition of the optimization problem; all additional information required by the optimization solver is determined automatically.

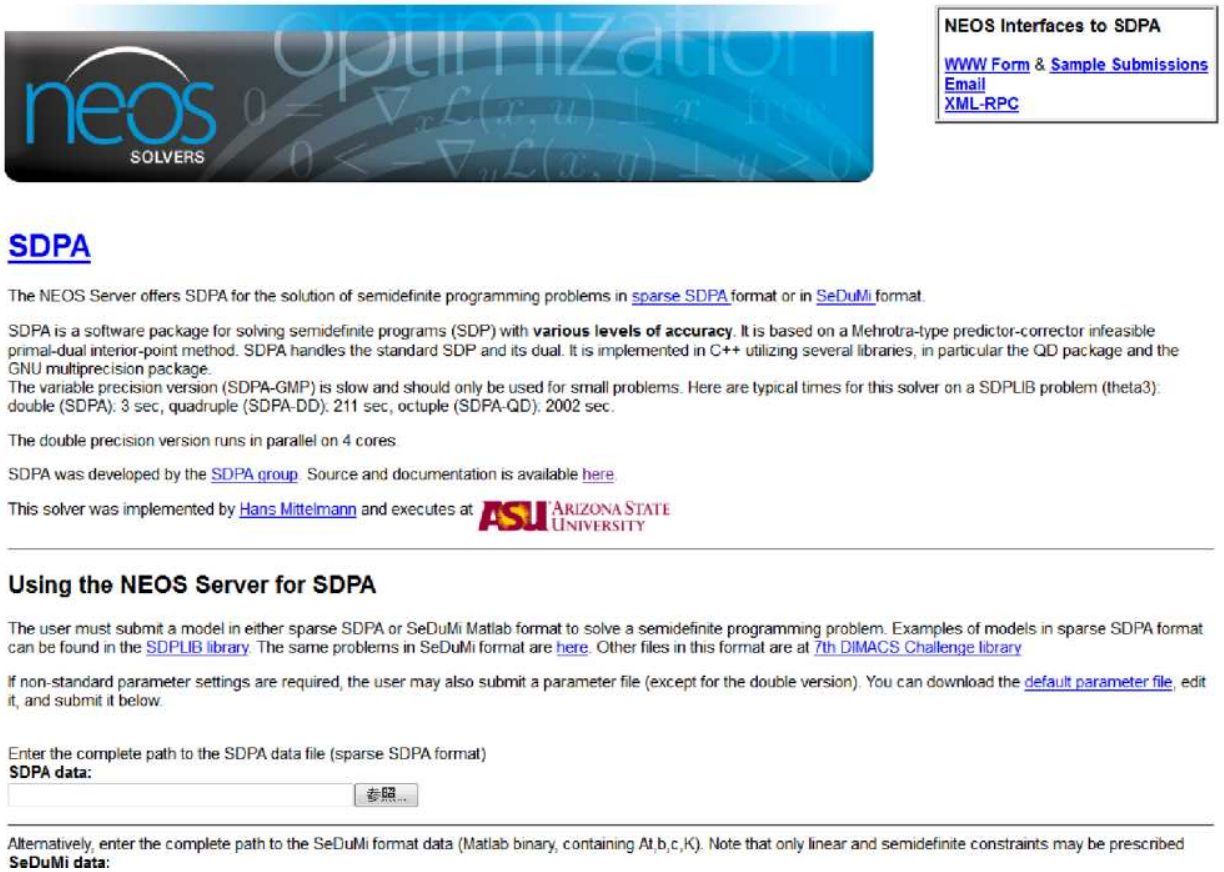
- User Feedback
- FAQ - NEOS Server
- Acknowledgements
- Collaborators

To submit your optimization job, first click on the **NEOS Solvers** icon to find a suitable solver.

You can also view current job queues and check job progress by number and password.

NEOS Information

- The NEOS Guide provides information on a variety of optimization techniques and solver programs.
- Optimization Software Guide
- The NEOS Application Programming Interface
- Statistics
- Web Site Statistics
- Cluster Statistics
- Job Archive
- Downloads



NEOS Interfaces to SDPA

WWW Form & Sample Submissions
Email
XML-RPC

SDPA

The NEOS Server offers SDPA for the solution of semidefinite programming problems in sparse SDPA format or in SeDuMi format.

SDPA is a software package for solving semidefinite programs (SDP) with **various levels of accuracy**. It is based on a Mehrotra-type predictor-corrector infeasible primal-dual interior-point method. SDPA handles the standard SDP and its dual. It is implemented in C++ utilizing several libraries, in particular the QD package and the GNU multiprecision package.
The variable precision version (SDPA-GMP) is slow and should only be used for small problems. Here are typical times for this solver on a SDPLIB problem (theta3): double (SDPA): 3 sec, quadruple (SDPA-DD): 211 sec, octuple (SDPA-QD): 2002 sec.

The double precision version runs in parallel on 4 cores.

SDPA was developed by the SDPA group. Source and documentation is available here.

This solver was implemented by Hans Mittelmann and executes at ASU Arizona State University

Using the NEOS Server for SDPA

The user must submit a model in either sparse SDPA or SeDuMi Matlab format to solve a semidefinite programming problem. Examples of models in sparse SDPA format can be found in the SDPLIB library. The same problems in SeDuMi format are here. Other files in this format are at 7th DIMACS Challenge library

If non-standard parameter settings are required, the user may also submit a parameter file (except for the double version). You can download the default parameter file, edit it, and submit it below.

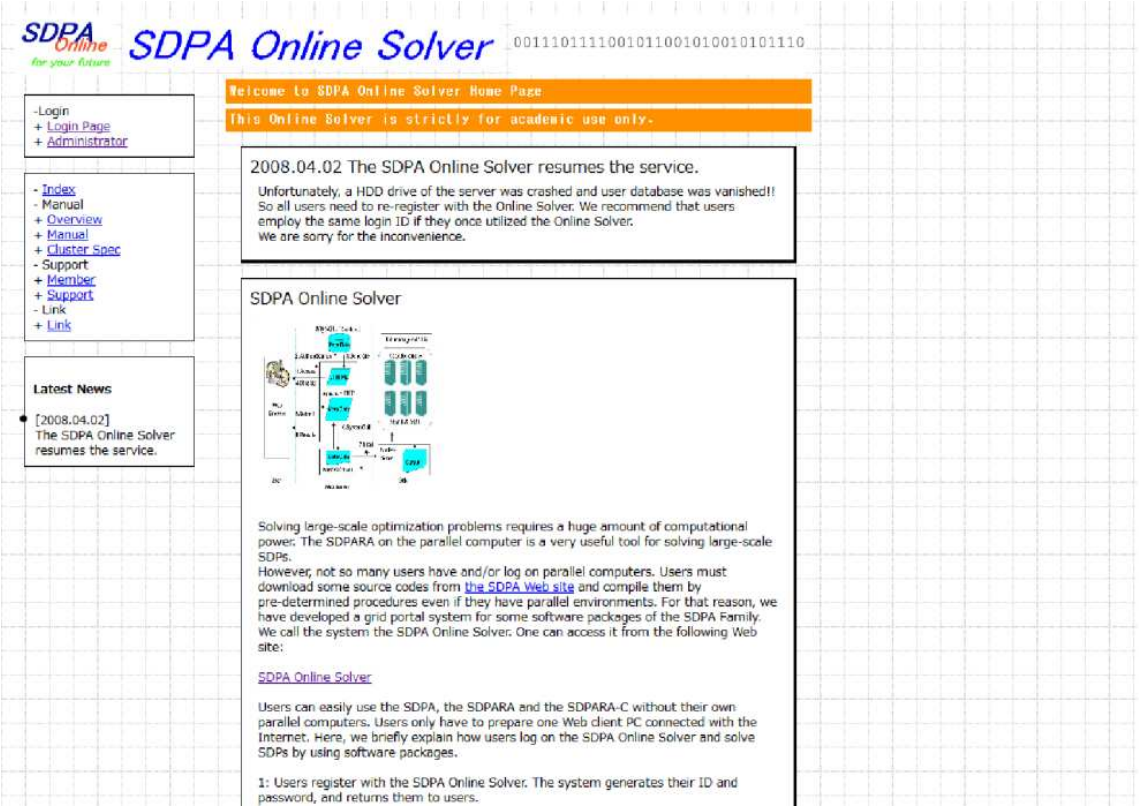
Enter the complete path to the SDPA data file (sparse SDPA format)
SDPA data:
参照...

Alternatively, enter the complete path to the SeDuMi format data (Matlab binary, containing At,b,c,K). Note that only linear and semidefinite constraints may be prescribed
SeDuMi data:

- **NEOS server** とは規定の問題定式化に従ってファイルを作成し，ウェブブラウザーを通して実際に問題を送信し，無料で最適化問題を解いてくれるサービスである
- 世界中の研究者が競って開発した最適化ソフトウェアを採用している
- それぞれ異なったインタフェイスや解ける問題の制限がある

6.1.2 SDPA Online Solver

http://sdpa.indsys.chuo-u.ac.jp/portal

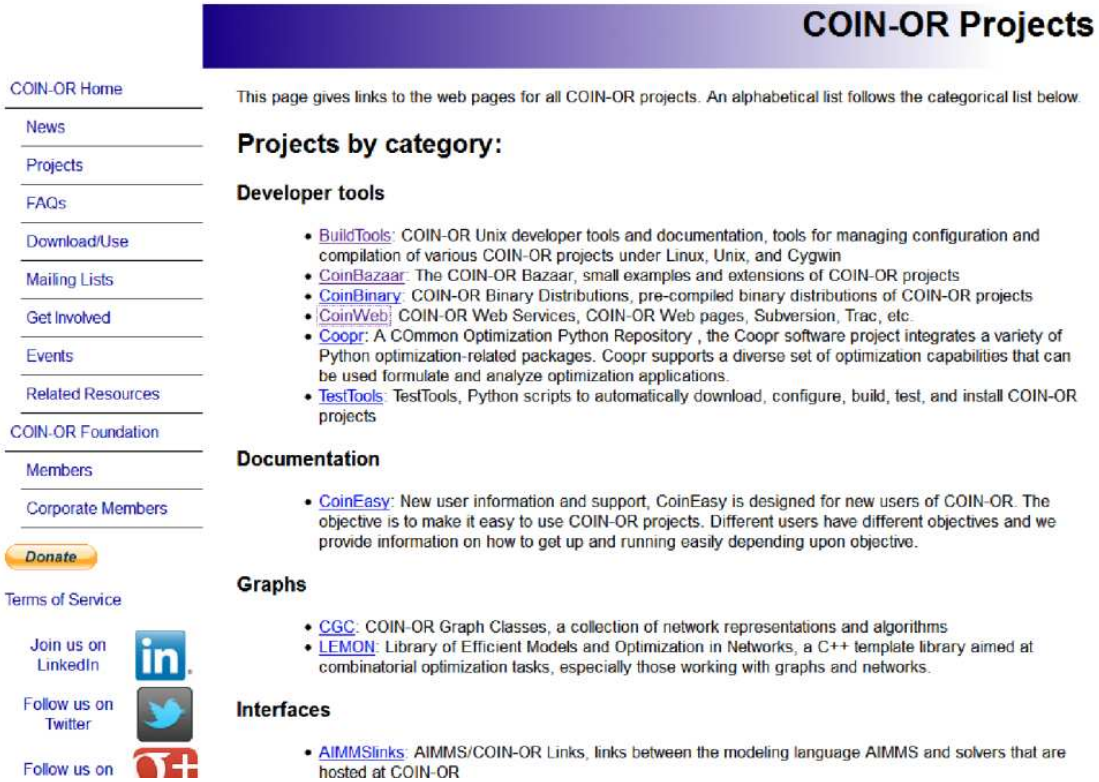


- **SDPA Online Solver**は中央大学の藤澤克樹先生を中心に開発されているシステム
- PCクラスター上で半正定値計画問題を並列で解いてくれる
- 登録さえすれば無料で利用できる

6.1.3 COIN-OR

- 余り凝りたくはないが，自分でも少し最適化ソフトウェアを開発したい
- オープンソースの最適化ソフトウェアを開発するプロジェクト

http://www.coin-or.org



COIN-OR Projects

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
Join us on LinkedIn
Follow us on Twitter
Follow us on

This page gives links to the web pages for all COIN-OR projects. An alphabetical list follows the categorical list below.

Projects by category:

Developer tools

- BuildTools: COIN-OR Unix developer tools and documentation, tools for managing configuration and compilation of various COIN-OR projects under Linux, Unix, and Cygwin
- CoinBazaar: The COIN-OR Bazaar, small examples and extensions of COIN-OR projects
- CoinBinary: COIN-OR Binary Distributions, pre-compiled binary distributions of COIN-OR projects
- CoinWeb: COIN-OR Web Services, COIN-OR Web pages, Subversion, Trac, etc.
- Coopr: A COmmon Optimization Python Repository , the Coopr software project integrates a variety of Python optimization-related packages. Coopr supports a diverse set of optimization capabilities that can be used formulate and analyze optimization applications.
- TestTools: TestTools, Python scripts to automatically download, configure, build, test, and install COIN-OR projects

Documentation

- CoinEasy: New user information and support, CoinEasy is designed for new users of COIN-OR. The objective is to make it easy to use COIN-OR projects. Different users have different objectives and we provide information on how to get up and running easily depending upon objective.

Graphs

- CGC: COIN-OR Graph Classes, a collection of network representations and algorithms
- LEMON: Library of Efficient Models and Optimization in Networks, a C++ template library aimed at combinatorial optimization tasks, especially those working with graphs and networks.

Interfaces

- AIMMSlinks: AIMMS/COIN-OR Links, links between the modeling language AIMMS and solvers that are hosted at COIN-OR

- COIN-OR は IBM が主体として始まったプロジェクトである
- 最適化ソフトウェアのダウンロードをし，それをベースに自作のコードや他のソフトウェアと組み合わせることが可能
- 柔軟性がある

6.1.4 Decision Tree for Optimization Software

http://plato.asu.edu/guide.html

Decision Tree for Optimization Software

Navigation Menu
- Home
- Problems & Software
- Benchmarks
- Testcases
- Books & Tutorials
- Tools
- Websubmission
- Other Sources

Search the decison tree

Webstatistics for server plato

Welcome! This site aims at helping you identify ready to use solutions for your optimization problem, or at least to find some way to build such a solution using work done by others. If you know of useful sources not listed here, please let us know. If something is found to be erroneous, please let us know, too. Where possible, public domain software is listed here.

In any case, observe the expressed or implied LICENSE conditions ! In most cases, these accompany the source code. As a rule, most codes are free for research. This means free for academic research and teaching or for trying whether it serves your needs. Commercial uses (either direct or indirect) require licensing, as a rule.

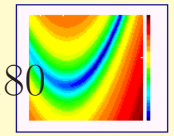
We do not aim at giving an overview over existing commercial products and recommend one of the other guides for that. We have structured the information in the way you can see on the left. Clicking on the corresponding part takes you there. The contents are as follows:

- problems/software: software sorted by problem to be solved
- benchmarks: collection of testresults and performance tests, made by us or others
- testcases: example files ready to use with existing software, in different formats
- books/tutorials: a short list of introductory texts, some online
- tools: software which helps formulating an optimization problem or simplifying its solution
- websubmission: some software can be used directly via the net (mainly for trying it out), thanks to implementors who make their computing facilities available to you
- other sources: for more information provided by others

For an explanation of terms used in optimization consult the -> Mathematical Programming Glossary

Hans D. Mittelmann,
Dept of Math&Stats
Arizona State University

@ mittelmann at math.asu.edu



This guide was started and maintained jointly for several years with Peter Spellucci, TU Darmstadt. You may cite it formally as:
H.D. Mittelmann, Decision Tree for Optimization Software.

- SAL-ORLib Pure 0/1, IP solvers, mixed integer solver (Basic, f77, C, Pascal, Java)

PURE LP SOLVERS: most allow input in MPS format:

- CLP — the COIN LP code, simplex (C++)
- PyCOIN — Python interface to CLP (maybe more COIN solvers)
- mexclp — Matlab interface to CLP
- QSopt — Simplex based; executables, library for Windows/Unix, free for research&education, Concorde interface, Matlab interface
- SOPLEX — Simplex based, C++
- BPMPD — interior point method, Win32-version, Unix-binaries (version 2.21), Unix-libraries (version 2.21), BPMPD_MEX , netlib-source (version 2.11, f77), C-version (version 2.11, f2c), AMPL interface
- HOPDM — local version of code (interior point method using predictor-corrector-steps of higher order), HOPDM_C (f2c-converted version, includes f2c-library), IIASA-archive (C++/f77, dynamic memory allocation, binaries for Solaris and win95 only)
- PCx — interior point method, f77/C, binaries for major platforms
- LOQO — interior point method, C, student binaries, executables for major platforms, work for two months w/o license, AMPL interface, Matlab interface
- COPL-LP — interior point method, includes crossover, DOS executable, HP binary, Linux binary
- CDDMEX — Matlab interface to polyhedral C/C++ code CDD
- LIPSOL — interior point method, also MAT or LPP formats (Matlab)
- IPO/SIMPO — interior and simplex LP-solver published in conjunction

- **Decision Tree for Optimization Software**は Arizona州立大学の Hans D. Mittelmann 教授によって始められた最適化ソフトウェアのガイド
- 最適化ソフトウェアの分類，リンク集
- ベンチマーク集
- 各ソフトウェアのパーフォマンスの比較 (ベンチマーク)

6.1.5 最近ホットなソフトウェア

- **Gurobi (http://www.gurobi.com)**
  - 線形計画問題や混合整数計画問題を解くソルバー
- **SCIP (http://scip.zib.de)**
  - 混合整数計画問題を解くソルバー
  - 整数計画問題を解く手法と constraint programming を解く手法を実装している

6.2 モデリング言語

- 既存のソフトウェアのインターフェイスにはモデリング言語が使われることが多い
- 素人でも分かり易い記述でデータの受け渡しが可能

- **AMPL (Bell研が開発→AMPL Optimization LLC)**
- **MPL (Maximal Software)**
- **GAMS (GAMS Development)**
- **AIMMS (Paragon Decision Technology)**
- **SIMPLE (数理システム)**

6.2.1 AMPLの例題

```
set NUTR;
set FOOD;

param cost {FOOD} > 0;
param f_min {FOOD} >= 0;
param f_max {j in FOOD} >= f_min[j];

param n_min {NUTR} >= 0;
param n_max {i in NUTR} >= n_min[i];

param amt {NUTR,FOOD} >= 0;

var Buy {j in FOOD} >= f_min[j], <= f_max[j];

minimize Total_Cost:  sum {j in FOOD} cost[j] * Buy[j];

subject to Diet {i in NUTR}:
   n_min[i] <= sum {j in FOOD} amt[i,j] * Buy[j] <= n_max[i];
```

```
data;

set NUTR := A B1 B2 C ;
set FOOD := BEEF CHK FISH HAM MCH MTL SPG TUR ;

param:   cost  f_min  f_max :=
  BEEF   3.19    0     100
  CHK    2.59    0     100
  FISH   2.29    0     100
  HAM    2.89    0     100
  MCH    1.89    0     100
  MTL    1.99    0     100
  SPG    1.99    0     100
  TUR    2.49    0     100 ;

param:   n_min  n_max :=
   A      700   10000
   C      700   10000
   B1     700   10000
   B2     700   10000 ;
```

```
param amt (tr):
           A    C   B1   B2 :=
   BEEF   60   20   10   15
   CHK     8    0   20   20
   FISH    8   10   15   10
   HAM    40   40   35   10
   MCH    15   35   15   15
   MTL    70   30   15   15
   SPG    25   50   25   15
   TUR    60   20   15   10 ;
```