Summer School on Numerical Optimization Software - University of Freiburg

Exercise 0: Solvers Description and Software Installation

Gianluca Frison, Hans Mittelmann, Moritz Diehl

1 Course website and repository

The course website is (as you probably already know)

`syscop.de/teaching/2016/summer-school-on-numerical-optimization-software`

During the course, teaching material will be updated on the Git repository

`http://gitlab.syscop.de/teaching/NOS_public`

Git is a powerful version control system. Even if you do not know Git, you can download all files you need directly from the website (using the Download zip button, the download by clicking on the single files does not work at the moment).

You can find this guide on the repository too.

2 Operating system

In the course description, it was indicated that the use of the Linux operating system was highly desirable. The main reason for this is that the course requires the installation of several software packages, and the teaching team is made up of experienced Unix (and mainly Linux) users, with very limited possibility to give support for Windows users.

Nevertheless, it is possible to attend the course and work on (most) of the exercises using Windows, even if it is not guaranteed that everything will work.

The remain of this Exercise 0 briefly describes the used solvers, and gives step-by-step instructions for software installation on a Linux 64-bit machine (and preferably a recent Ubuntu distribution like 16.04). Users of other operating systems must adapt the following instructions to their case.

As a final note, it is possible to use all solvers through submission to the NEOS server. However, this requires a working and reliable internet connections (that is not guaranteed). Furthermore, the submitted scripts may be queued on the server (and at the moment we have no idea how this will work out with about 50 participants). Finally, it may be much faster and easier to debug exercises locally rather than through successive submissions to the server. Therefore, it is highly recommended to use the solvers locally on your computer as much as possible (unless differently stated on the exercise text).

3 Solver list & brief description

In this course, the following solvers will be employed either locally or through submission to the NEOS server:

BARON Baron is a computational system for solving nonconvex optimization problems to global optimality. Purely continuous, purely integer and mixed-integer nonlinear problems can be solved with the software. The Branch and Reduce Optimization Navigator derives its name from combining constraint propagation, interval analysis and duality in its reduce arsenal with enhanced branch and bound concepts as it winds its way through the hills and valleys of complex optimization in search of global solutions.

Bonmin Basic Open-source Nonlinear Mixed INteger programming is an experimental open-source solver for general MINLP problems. Bonmin features several algorithms.

CONOPT CONOPT is a solver for large-scale nonlinear optimization. It is a feasible path solver based on the Generalized Reduced Gradient (GRG) method. It is oftern the preferable solver for very nonlinear models.

CPLEX CPLEX is a commercial optimization software package. It solves integer programming problems, very large LP using either primal or dual variants of the simplex method or the barrier IPM, convex and non-convex QP, convex QCP, SOCP.

COUENNE Convex Over and Under ENvelopes for Nonlinear Estimation is an open source library for solving global optimization problems. It can solve mixed-integer nonlinear optimization problems. Both the objective function and the constraints might be nonlinear and nonconvex. Couenne solves global optimization problems using a reformulation based branch-and-bound algorithm for a globally optimum solution.

Gurobi The Gurobi optimizer is a commercial optimization solver for LP, QP, QCP, MILP, MIQP and MIQCP. It is named after its founders.

IPOPT Interior Point OPTimizer is a software library for large scale nonlinear optimization of continuous systems. IPOPT implements a primal-dual interior point method and uses line search based on filter methods.

KNITRO Nonlinear Interior point Trust Region Optimization (K is silent) is a commercial software package for solving large scale mathematical optimization problems, specialized in nonlinear optimization. KNITRO offers four different optimization algorithms. Two algorithms are interior point methods (with direct or iterative (conjugate gradient) linear algebra), one algorithm is an active set method and one algorithms is a Sequential Quadratic Programming (SQP) method. KNITRO allows crossover during the solution process from one algorithm to another. It also provides multistart option for promoting the computation of the global minimum.

MOSEK MOSEK is a commercial software package for the solution of linear, mixed-integer linear, quadratic, mixed-integer quadratic, quadratically constraint, conic and convex nonlinear mathematical optimization problems. The emphasis in MOSEK is on solving large-scale sparse problems. Particularly the IPM for linear, conic quadratic and semi-definite problems es very efficient. This IPM is based on the homogeneous model with implies MOSEK can reliably detect primal and/or dual infeasible status.

MINOS Modular In-core Nonlinear Optimization System is a software package for solving linear and nonlinear optimization problems. For linear programs, a two-phase primal simplex method is used. For problems with linear constraints and nonlinear objective, a reduced-gradient method is used. For problems with nonlinear constraints, a linearly constrained Lagrangian method is used. MINOS is intended for large sparse problems.

SCIP Solving Constraint Integer Programs is a mixed integer programming solver and framework for branch and cut and branch and price. The solver for the LP relaxations is not a native component of SCIP, an open LP interface is provided instead. Run as a standalone solver, it is one of the fastest non-commercial solvers for mixed integer programs.

SeDuMi SeDuMi is an add-on for Matlab, which lets you solve optimization problems with linear, quadratic and semidefiniteness constraints.

SNOPT Sparse Nonlinear OPTimizer is a software package for solving large-scale nonlinear optimization problems. It employs a sparse SQP algorithm with limited-memory quasi-Newton approximation of the Hessian of the Lagrangian. The functions should be smooth but need not be convex.

Xpress The Fico Xpress-Optimizer is a commercial optimization solver for LP, MILP, convex QP, convex QCQP, SOCP and their mixed integer counterparts.

4 AMPL demo installation on a 64-bit Linux machine

Here it is described how to locally install and use the demo version of AMPL on a Linux machine, assuming that you do not have root privileges on your system.

The free demo version of AMPL can be downloaded from

`ampl.com/try-ampl/download-a-free-demo/`

The demo version can only solve problems of limited size. It comes with a number of solvers, some with even more restrictive limitations on the number of variables.

In the following, it is assumed that the 64-bit version for Linux is downloaded, and the file `amplide.linux64` is saved on your home directory `/home/your_name`. Once uncompressed the tgz file, you can navigate the main AMPL directory as

```
$ cd /home/your_name/amplide.linux64/
```

or simply

```
$ cd amplide.linux64/
```

if you already are in your home directory.

AMPL comes in two versions, a command line version (`ampl`), and an IDE version (`amplide`).

The easier way is probably the use of the IDE version. It can be launched by navigating in the AMPL installation directory and launching the `amplide` executable in the `amplide` directory, as

```
$ cd amplide
$ ./amplide
```

Once launched, the IDE version allows you to navigate to the folder where your model (`.mod`) and data (`.dat`) files are.

If you want to use the command line version, the main AMPL directory (where the `ampl` and the solvers executables are) has to be added to the execution path, such that you can use the `ampl` and the solvers executables from anywhere (e.g. from the folder where your model and data files are). An easy and safe way to do it is to type on a terminal

```
$ export PATH=~/amplide.linux64/:$PATH
```

and afterwards type `ampl` to launch the command line version. Note that the procedure is temporary, and has to be repeated every time you close and re-open the terminal.

If you want a permanent solution, you may add the above line at the end of the proper configuration file, that depending on you system may be e.g. `~/.bashrc`, `~/.profile` or `~/.bash_profile`. Depending on the configuration file and operative system used, yuo need to re-open your terminal or even log out and back into your system to make the changes applied to the configuration work.

5 Installation of additional open-source solvers for AMPL

Additional open-source solvers interfaced with AMPL can be downloaded from the page

`http://ampl.com/products/solvers/open-source/`

For most solvers, both source-code or binaries for several operating systems can be downloaded.

During the course, the following open-source solvers will be employed:

- Couenne
- Bonmin
- Ipopt

The easiest way to obtain them is to download the correct binary as zip file, unzip the folder and copy the executable to the main AMPL directory.

6 Installation of SCIP for AMPL

The SCIP solver has to be installed from source code. The procedure below is inspired by

`http://zverovich.net/2012/08/07/using-scip-with-ampl.html`

and has been tested on an Ubuntu 14.04 distribution. The process should be similar on other Linux distributions with `apt` replaced with the proper package manager. No hints can be done on the installation on other operating systems.

Download the archive containing the source code for the SCIP Optimization Suite at

`http://scip.zib.de/download.php?fname=scipoptsuite-3.2.1.tgz`

(tick the box for academic use licence) and extract its content:

```
$ tar xzf scipoptsuite-3.2.1.tgz
```

Install the dependencies:

```
$ sudo apt-get install g++ zlib1g-dev bison flex \
libgmp-dev libreadline-dev libncurses5-dev
```

Build SCIP:

```
$ cd scipoptsuite-3.2.1/
$ make
```

Download and build the AMPL solver library

```
$ cd scip-3.2.1/interfaces/ampl/
$ ./get.ASL
$ cd solvers/
$ sh configurehere
$ make
$ cd ..
```

Build the SCIP-AMPL interface and copy `scipampl` on the AMPL main directory (supposed to be `~/amplide.linux64/`)

```
$ make
$ cp bin/scipampl ~/amplide.linux64/
```

Optionally, the `scipampl` executable can be renamed into something shorter like `scip`:

```
$ mv ~/amplide.linux64/scipampl ~/amplide.linux64/scip
```

7 Installation of CVX

CVX is a Matlab-based modeling system for convex optimization. CVX turns Matlab in a modeling language, allowing objects and constraints to be specified using standard Matlab expression syntax.

Therefore, CVX requires Matlab. The exercises have been tested on Matlab R2015b on Linux Ubuntu 14.04 64-bit, but any version of Matlab for any OS should work.

The latest stable version of CVX is version 2.1, that can be downloaded here

`http://cvxr.com/cvx/download/`

choosing e.g. `cvx-rd.zip` (version with free solvers only, for all platforms).

Installation instructions can be found here

`http://cvxr.com/cvx/doc/install.html`

The combination of CVX and Octave is in alpha. In particular, Octave 4.0.0 or later is required (this version is installed e.g. in Ubuntu 16.04), and together with CVX 3.0 or later (that is in beta itself). The exercises in this course are made to work also in CVX 3.0 beta + Octave 4.0.0, but the solutions of the optimization problems are different. Therefore CVX 2.1 + Matlab is recommended.

If you still want to use Octave because e.g. you do not have a Matlab license, here is some hint on the installation process. First of all, it is recommended to install the following packages in e.g. Ubuntu 16.04 (to be sure to have version 4.0.0) by typing in a terminal:

```
$ sudo apt-get install octave liboctave-dev octave-image
```

(the latter is not needed to use with CVX, but it is required on another exercise during the course).

CVX 3.0 beta for Octave requires the header `f77blas.h` to be installed on the system. One way to do it is to install OpenBLAS, e.g. in Ubuntu typing in a terminal

```
$ sudo apt-get install libopenblas-base libopenblas-dev
```

Afterwards download CVX 3.0 beta, and follow the installation instructions. You will get an error, saying that the platform is not yet supported in CVX, with the hint to run `cvx_compile`: do so.

Afterwards you need to compile the solver MEX files with the commands in the Octave prompt (assuming the main CVX directory is in `/home/your_name/cvx`)

```
> cd /home/your_name/cvx/sedumi
> install_sedumi -nopath
> cd /home/your_name/cvx/sdpt3
> install_sdpt3 -nopath
> cd /home/your_name/cvx
> cvx_setup
```

Trying to install SeDuMi, you will get an error, saying that the header `f77blas.h` can not be located. In order to fix that, edit the file `/home/your_name/cvx/sedumi/blksdp.h` and substitute

```
#include "f77blas.h"
```

with

```
#include "openblas/f77blas.h"
```

and run on the Octave prompt

```
> install_sedumi -rebuild -nopath
```

and continue with the following commands above.