

Brief Intro to Opt Solvers

NUS Summer School Tutorial

Li Chen

June 17, 2022

Overview

- ▶ The goal: Solve optimization problems numerically via solvers
- ▶ The prerequisite: Basic knowledge of computer programming
 - ▶ You can use your favorite language, such as **Matlab**, **Python**, **Julia**, C, Java, ...
 - ▶ When implementing algorithms, Matlab is the most convenient one for matrix computation, although Python (with NumPy) or Julia are also OK (the cheatsheet).

Overview

- ▶ The goal: Solve optimization problems numerically via solvers
- ▶ The prerequisite: Basic knowledge of computer programming
 - ▶ You can use your favorite language, such as **Matlab**, **Python**, **Julia**, C, Java, ...
 - ▶ When implementing algorithms, Matlab is the most convenient one for matrix computation, although Python (with NumPy) or Julia are also OK (**the cheatsheet**).
- ▶ The outline
 - ▶ Brief intro of opt solvers
 - ▶ Examples of using Julia + JuMP
 - ▶ Examples of using Matlab + CVX
 - ▶ Examples of using Python + MOSEK

What are opt solvers?

A solver is a software package that incorporates algorithms for finding solutions to one or more classes of problem.

- ▶ **Examples of opt solvers**
- ▶ Commercial solvers vs open-source solvers
 - ▶ commercial solvers are often faster and more robust, and they are free for academic users (by requesting an academic license).
 - ▶ good open-source solvers are usually based on Matlab or C.

What are opt solvers?

A solver is a software package that incorporates algorithms for finding solutions to one or more classes of problem.

- ▶ **Examples of opt solvers**
- ▶ Commercial solvers vs open-source solvers
 - ▶ commercial solvers are often faster and more robust, and they are free for academic users (by requesting an academic license).
 - ▶ good open-source solvers are usually based on Matlab or C.
- ▶ Recommended solvers: **MOSEK**, **SDPT3**
 - ▶ Because this course mainly focuses on continuous optimization, and you will solve semidefinite programs (SDPs)
 - ▶ Gurobi and CPLEX do not support SDP

What are opt solvers?

A solver is a software package that incorporates algorithms for finding solutions to one or more classes of problem.

- ▶ **Examples of opt solvers**
- ▶ Commercial solvers vs open-source solvers
 - ▶ commercial solvers are often faster and more robust, and they are free for academic users (by requesting an academic license).
 - ▶ good open-source solvers are usually based on Matlab or C.
- ▶ Recommended solvers: **MOSEK**, **SDPT3**
 - ▶ Because this course mainly focuses on continuous optimization, and you will solve semidefinite programs (SDPs)
 - ▶ Gurobi and CPLEX do not support SDP
- ▶ You also need a modeling system/ interface such as CVX, YALMIP, JuMP, to make opt solvers easier to use.

What are opt solvers?

A solver is a software package that incorporates algorithms for finding solutions to one or more classes of problem.

- ▶ **Examples of opt solvers**
- ▶ Commercial solvers vs open-source solvers
 - ▶ commercial solvers are often faster and more robust, and they are free for academic users (by requesting an academic license).
 - ▶ good open-source solvers are usually based on Matlab or C.
- ▶ Recommended solvers: **MOSEK**, **SDPT3**
 - ▶ Because this course mainly focuses on continuous optimization, and you will solve semidefinite programs (SDPs)
 - ▶ Gurobi and CPLEX do not support SDP
- ▶ You also need a modeling system/ interface such as CVX, YALMIP, JuMP, to make opt solvers easier to use.
- ▶ Regardless what solvers to use, always focus on the three elements: decision variables, constraints, and objective functions

- ▶ Download and install **Julia and JuMP**
- ▶ After installing JuMP, you can install a free solver SCS to test

```
import Pkg
Pkg.add("SCS")
```
- ▶ Then you can explore JuMP through **A quick start** or **examples**
- ▶ Use other solvers
 - ▶ SDPT3: you need to install Matlab and SDPT3 first, then install the interface **SDPT3.jl**. **(not recommended)**
 - ▶ MOSEK: you need to install MOSEK first, then install the interface **MosekTools.jl**

- ▶ Install Matlab
- ▶ Download and install CVX according to the website instruction
- ▶ You can use either SDPT3 or MOSEK, or try both
 - ▶ SDPT3 is already included in CVX as a free solver, you can set it as the default by `cvx_solver sdpt3` after `cvx_setup`
 - ▶ You need to request an academic license for MOSEK, see [Using MOSEK with CVX](#)
- ▶ After you setup the solver, you can explore CVX through many examples (in the “examples” folder), e.g., [A quick start](#)

- ▶ **Install MOSEK**, e.g., `pip install Mosek`
- ▶ You can either directly use MOSEK or use other interface/ modeling language, such as Pyomo or CVXPY
 - ▶ Directly use MOSEK: Optimizer API or **Fusion API (recommended)**