

# ET.

*EFFICIENT TOOLS  
for RESEARCH*



*YALMIP: Optimization Made Easy*  
*2017-04-06*

YALMIP: Optimization Made Easy!

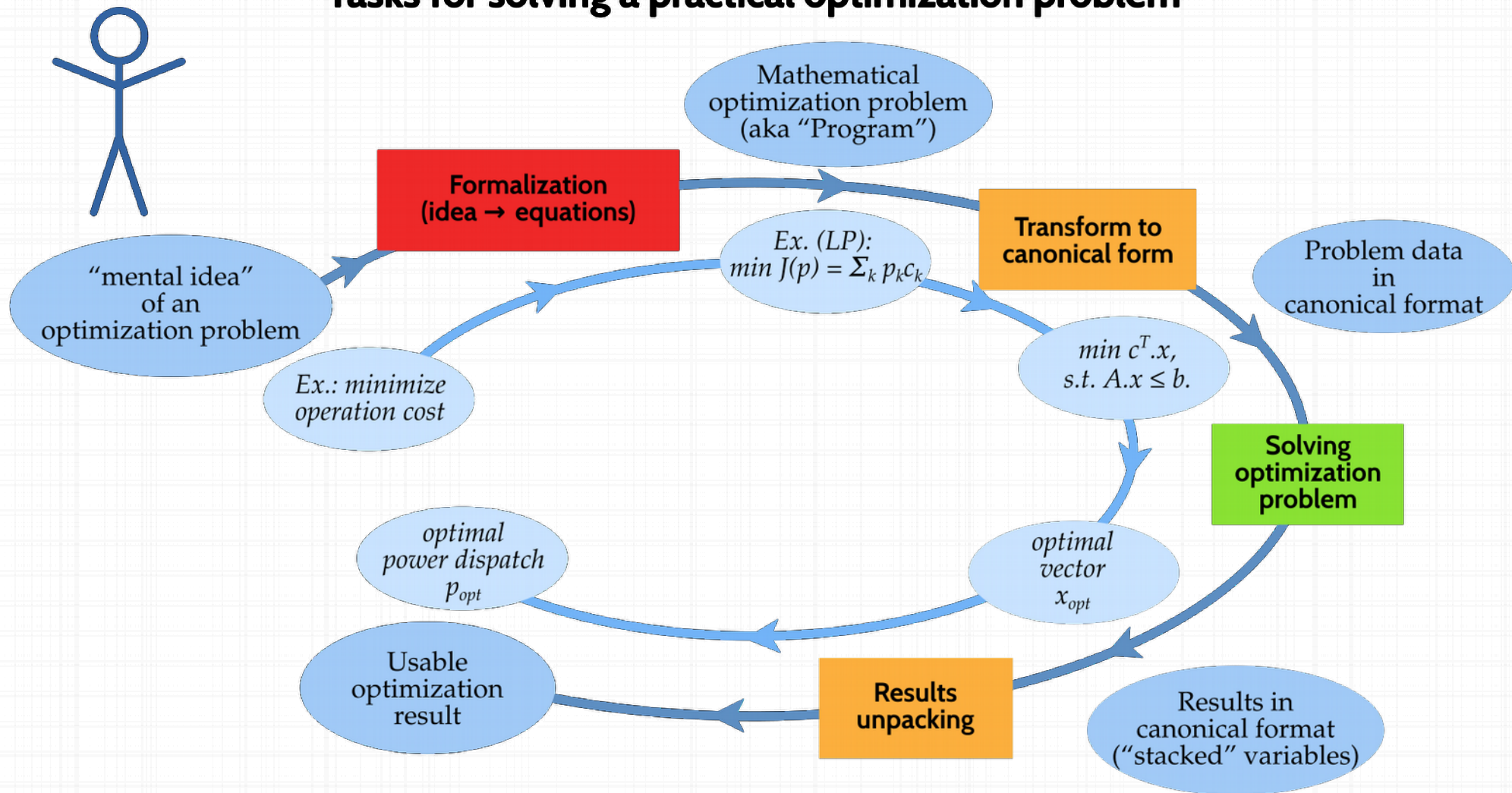
and

*Modeling Languages/Layers for Optimization*  
in general

Pierre Haessig

CentraleSupélec Rennes, April 6<sup>th</sup>, 2017

# Tasks for solving a practical optimization problem

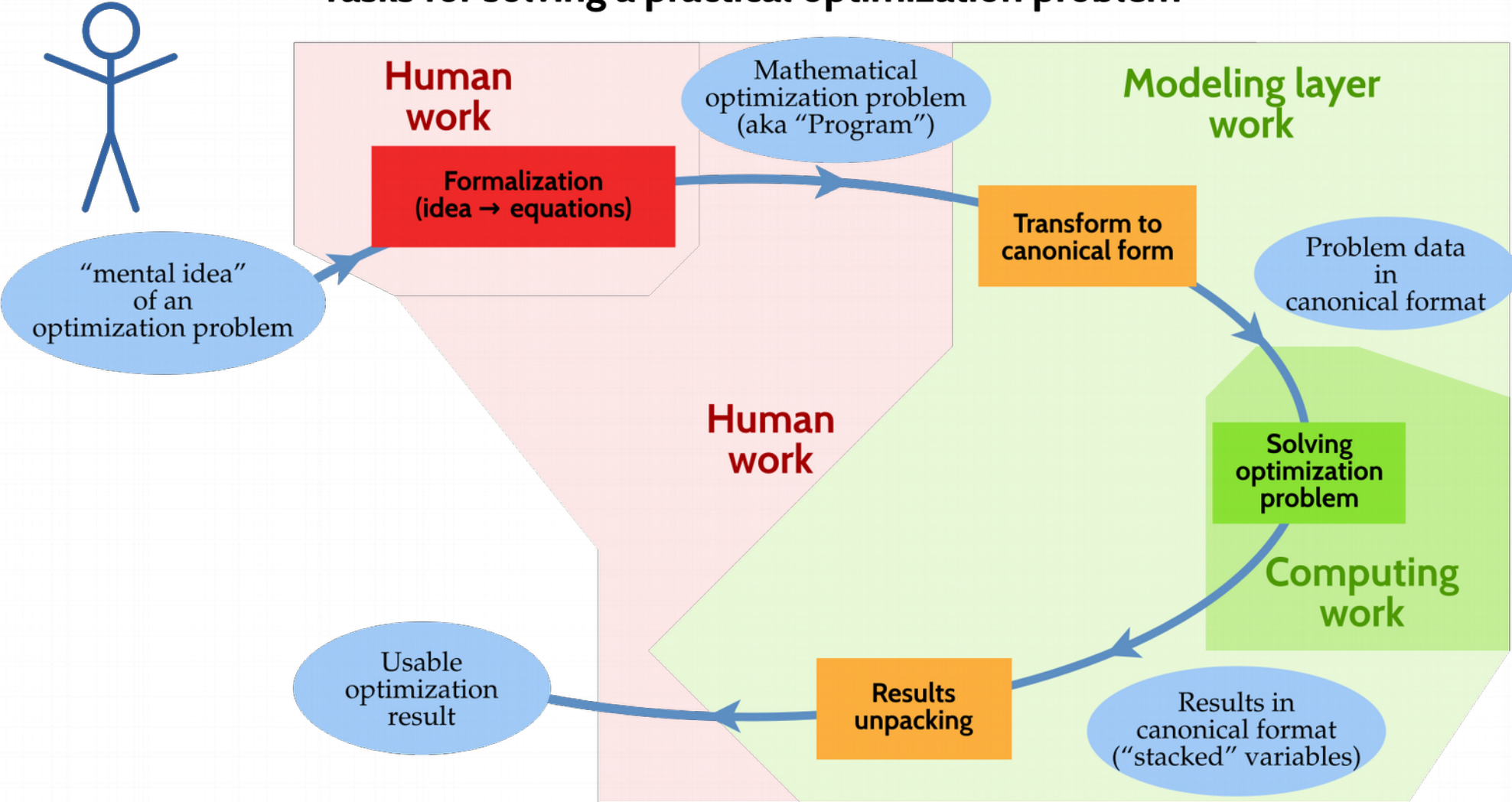


# Where is human time spent?

- 1) Formalize the problem into a Mathematical Optimization problem  
→ **core skill** of the researcher
- 2) Transform to a canonical form (**solver specific API**)  
*[now some computing time...]*
- 3) Retrieve results out of the canonical format (**again solver specific**)

YALMIP can help the researcher focus on its **core skill**

# Tasks for solving a practical optimization problem



# Modeling Languages/Layers for Optimization

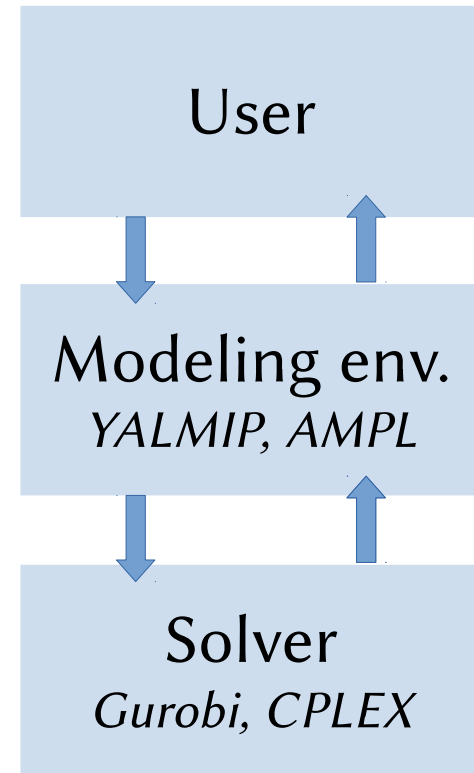
Environment	Software/Toobox/ Package
Standalone	AMPL, GAMS (~1990)
Matlab	<b>YALMIP</b> , CVX (~ 2000)
Python	PuLP, CVXPY
Julia	JuMP, Convex.jl

# Relationship to the optimization solver

- When using a modeling environment, the solver is *mostly hidden* from the user.
- The choice of the modeling layer is (mostly) independent of the choice of solver.

For ex. YALMIP provides interfaces for most common solvers:

- Gurobi, CPLEX, MOSEK (Commercial)
- GLPK, Ipopt, SEDUMI (Free)
- linprog from Matlab Optimization Toolbox



# Matlab Tooboxes

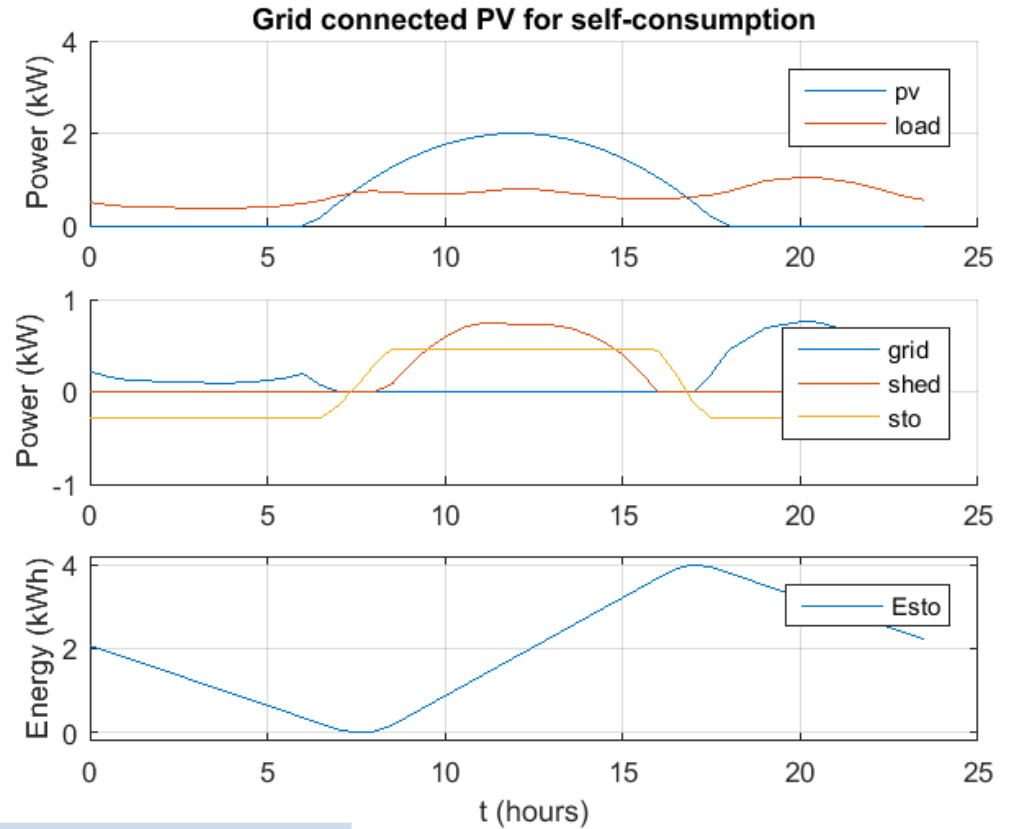
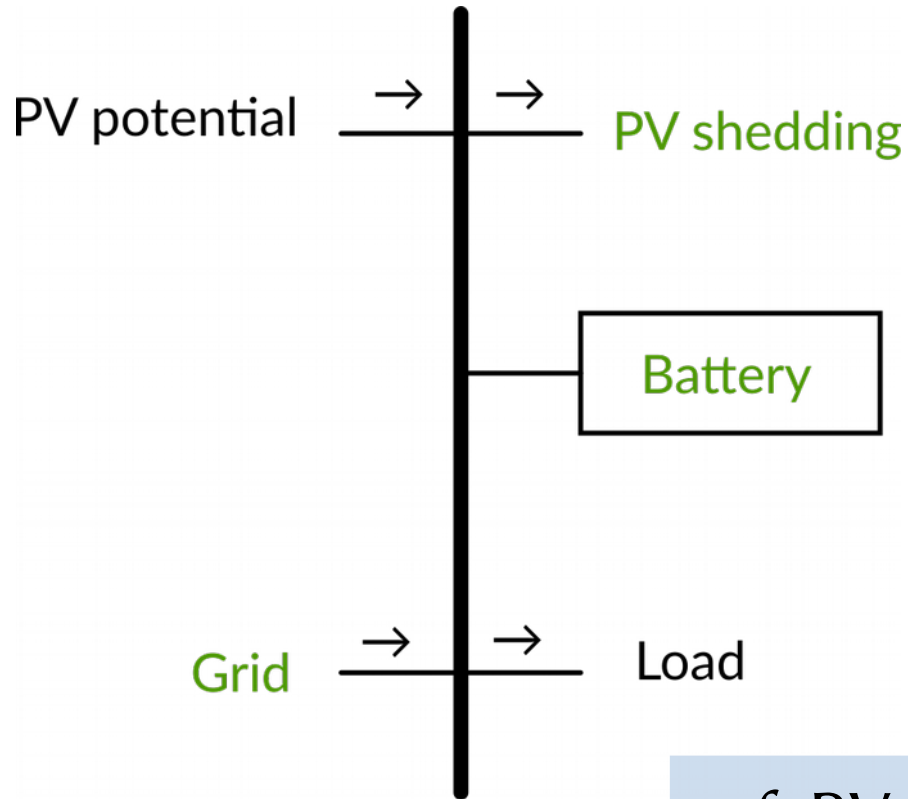
- **YALMIP**: 2001 – present, **J. Löfberg**  
from Linköping University, (post-doc at ETH Zürich)  
Project: <https://yalmip.github.io/>. Author: <http://users.isy.liu.se/en/rt/johanl/>
- **CVX** : 2005 – present, **M. Grant** and **S. Boyd**.  
from Stanford University  
*(compatibility problem with Matlab 2017)*
  - CVX Research, inc. 2012: <http://cvxr.com/>
  - M. Grant joined Continuum Analytics in 2015  
(platforms for Data Science, mostly Python).



# YALMIP quickstart

- 1) **Download** ZIP archive and unzip the archive in some folder.  
(<https://yalmip.github.io/download/>)
- 2) Add YALMIP folders (with subfolders) to the **MATLAB path**.  
(cf. <https://yalmip.github.io/tutorial/installation/> )
- 3) Start using it! You can look at the “Getting Started” tutorial.  
(<https://yalmip.github.io/tutorial/basics/>)

# Demo: Grid-connected PV-storage system



cf. PVgrid.m script

# Wrap up: advantages/drawbacks

Key advantages:

- Shorter starting time (for students), shorter development time
- **Increased agility** ( $\rightarrow$  *better research!*)
  - Quickly **compare solvers**
  - Quickly **compare different problem models** (e.g. LP vs. QP)

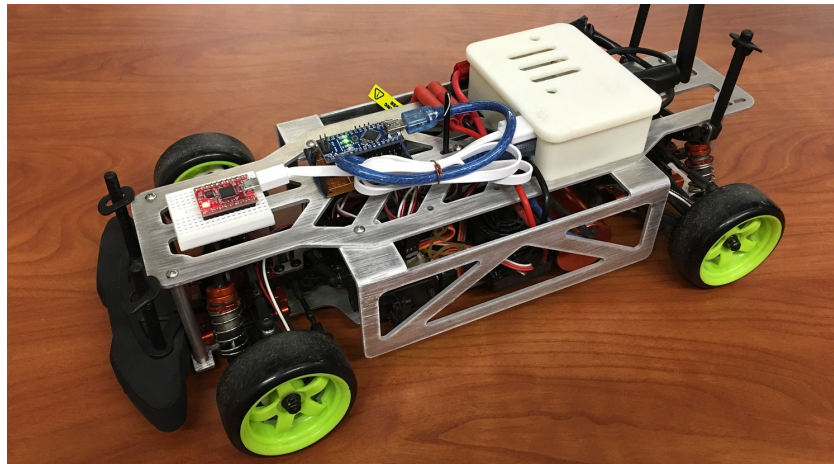
*But maybe:*

- Computational overhead?
  - e.g. less efficient when *recycling* the problem (like for MPC)?

# Application to Embedded Optimization

## Example of an **Autonomous Driving RC Car** “BARC Project”

- Implementation with Julia + **JuMP** (+ ROS + ...)
- Project pages: <http://www.barc-project.com/>, <https://github.com/MPC-Berkeley/barc>
- Presentation by Jon Gonzales (Berkeley MPC Lab) at JuliaCon 2016  
<https://www.youtube.com/watch?v=bX4TXWO7dA0>



# Standalone commercial modeling languages

- **AMPL** (A Mathematical Programming Language) <http://ampl.com/>
  - started ~1985 at Bell labs
  - “AMPL Optimization LLC” spun-off in 2002.
- **GAMS** (General Algebraic Modeling System) <https://www.gams.com>
  - started in 1970s at the World Bank (an economic modeling group)
  - commercial product by “GAMS Development Corp.” since 1987

# Some references, in chronological order

- **GAMS**: J. Bisschop and A. Meeraus, “On the development of a general algebraic modeling system in a strategic planning environment,” *Mathematical Programming Studies*, vol. 20, p. 1–29, **1982**.
- **AMPL**: R. Fourer, D. M. Gay, and B. W. Kernighan, “A Modeling Language for Mathematical Programming,” *Management Science*, vol. 36, no. 5, p. 519–554, **1990**.
- **YALMIP**: J. Löfberg, “YALMIP : a toolbox for modeling and optimization in MATLAB,” in *2004 IEEE International Conference on Robotics and Automation*, **2004**.
- **CVX**: M. Grant and S. Boyd, “Graph Implementations for Nonsmooth Convex Programs”, in *Recent Advances in Learning and Control (tribute to M. Vidyasagar)*, V. Blondel, S. Boyd, and H. Kimura, editors, Springer, **2008**, pp. 95-110.