Real-time convex optimization, with applications

Jacob Mattingley, Yang Wang and Argyris Zymnis

Information Systems Laboratory, Electrical Engineering, Stanford University

Convex optimization

- a convex optimization problem is of the form

  minimize $f(x)$
  subject to $x \in \mathcal{C}$

  where
  - the cost function f is convex (graph of f curves upwards)
  - the constraint set $\mathcal{C}$ is convex (closed to averaging)
- includes linear and quadratic programming as special cases
- can solve convex optimization problems **extremely well**
  - on a generic processor with a generic method for problems of up to 10^5 variables
  - with specialized iterative methods on multiple processors for larger problems
- many applications: control, combinatorial optimization, signal processing, machine learning, finance, ...
- recent advances in convex optimization include
  - robust optimization methods to handle parameter variation
  - ℓ_1-based heuristics for finding sparse solutions
  - parsers/solvers that make rapid prototyping easy
  - code generation for embedded optimization in real-time systems

Code generation

- say we have a quadratic program (QP), with variable $x \in \mathbf{R}^n$:

  minimize $x^T P x + q^T x$
  subject to $Gx \le h, \quad Ax = b$
- could *hand*-write a fast solver, but requires much time and is complicated
- instead, describe problem family in `cvxmod` (a Python package for convex optimization code generation):

```
A = matrix(...); b = matrix(...)
P = param('P', n, n, psd=True); q = param('q', n)
G = param('G', m, n); h = param('h', m)
x = optvar('x', n)
qpfam = problem(minimize(quadform(x, P) + tp(q)*x),
                [G*x <= h, A*x == b])
```

- generate a solver for the problem family `qpfam` with

```
qpfam.codegen()
```

- output includes `qpfam/solver.c`, and various ancillary files.
- can solve an instance (in C) with

```
status = solve(params, vars, work);
```

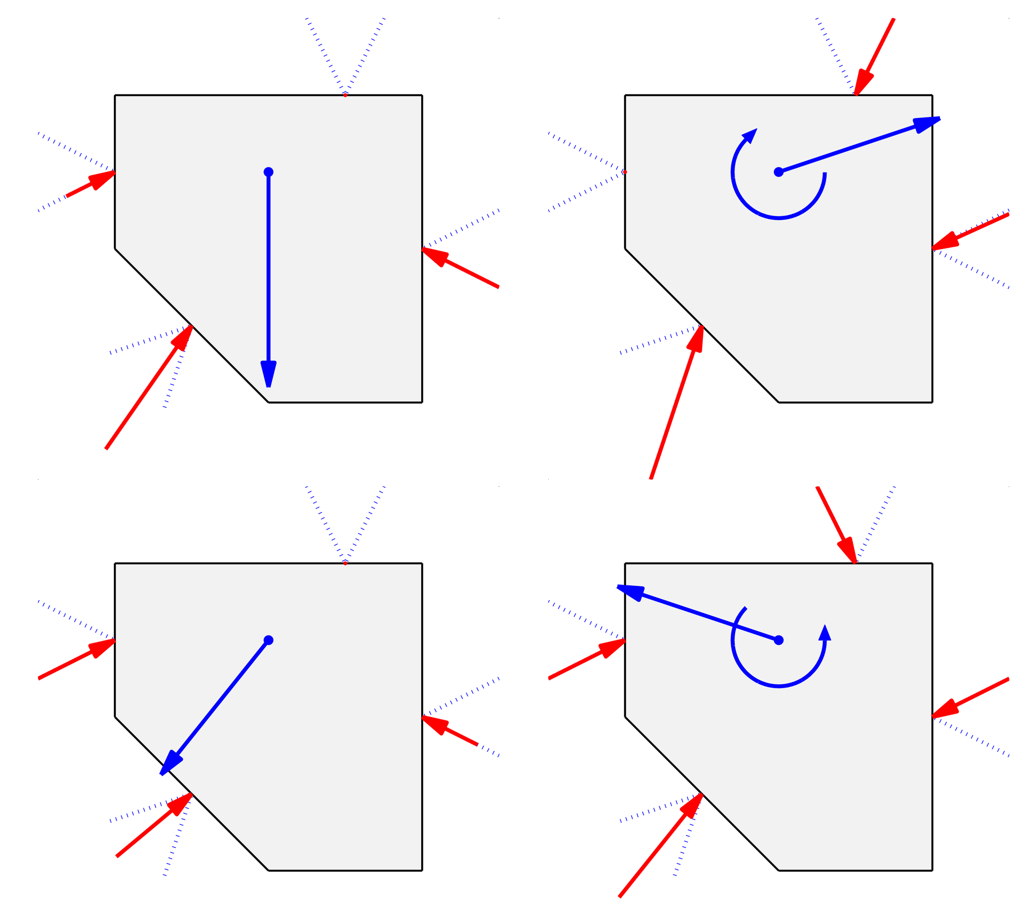
- solve times are up to 1000× faster than using off-the-shelf solver

Example 1: Grasp force optimization

- choose K grasping forces on object to
  - resist external wrench
  - respect friction cone constraints
  - minimize maximum grasp force
- convex problem (second-order cone program):

$$
\begin{array}{lll}
\text{minimize} & \max_i \|f^{(i)}\|_2 & \textit{max contact force} \\
\text{subject to} & \sum_i Q^{(i)} f^{(i)} = f^{\text{ext}} & \textit{force equillibrium} \\
& \sum_i p^{(i)} \times (Q^{(i)} f^{(i)}) = \tau^{\text{ext}} & \textit{torque equillibrium} \\
& \mu_i f_3^{(i)} \ge \left(f_1^{(i)2} + f_2^{(i)2} \right)^{1/2} & \textit{friction cone contraints}
\end{array}
$$

  variables $f^{(i)} \in \mathbf{R}^3$, $i = 1, \ldots, K$ (contact forces)
- can solve each instance in $60\mu s$



Example 2: Minimum energy processor speed scheduling

- processor adjusts its speed $s_t \in [s^{\min}, s^{\max}]$ in each of T time periods
- energy consumed in period t is $\phi(s_t)$; total energy is $E = \sum_{t=1}^{T} \phi(s_t)$
- n jobs
  - job i available at $t = A_i$; must finish by deadline $t = D_i$
  - job i requires total work $W_i \ge 0$
- $S_{ti} \ge 0$ is effective processor speed allocated to job i in period t

$$
s_t = \sum_{i=1}^{n} S_{ti}, \qquad \sum_{t=A_i}^{D_i} S_{ti} \ge W_i
$$

- choose speeds s_t and allocations S_{ti} to minimize total energy E
- when ϕ is convex, can be formulated as convex problem

$$
\begin{array}{ll}
\text{minimize} & E = \sum_{t=1}^{T} \phi(s_t) \\
\text{subject to} & s^{\min} \le s_t \le s^{\max}, \quad t = 1, \ldots, T \\
& s_t = \sum_{i=1}^{n} S_{ti}, \quad t = 1, \ldots, T \\
& \sum_{t=A_i}^{D_i} S_{ti} \ge W_i, \quad i = 1, \ldots, n
\end{array}
$$

- can solve each instance in $40\mu s$
- more sophisticated versions include
  - multiple processors
  - other constraints (thermal, speed slew rate, ...)
  - stochastic models for (future) data