Linear, Quadratic, and Mixed-Integer Linear Programming: Theory and Applications to Optimal Control

James Harrison (with some slides from Marco Pavone)

Stanford University

Optimization Approaches for Optimal Control

Today:

- Linear programming
- Quadratic programming
- Convex optimization
- Mixed integer linear programming (MILP)

Linear programming

Tools:

- Huge number of applications (e.g., production planning, pattern classification, multi-commodity flow problems, path planning)
- Can be solved very efficiently on huge problem instances (millions of variables and constraints)
- Popular solvers: **CPLEX**, MATLAB (`linprog`), GLPK

The problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \geq \mathbf{b} \end{aligned}$$

where $\mathbf{x} = (x_1, \ldots, x_n)^T$

Linear programming

Important points:

- Feasible set is a convex polytope
- The search for optimal solutions can be restricted to corner points (if they exist...)

Tools:

- `X = linprog(f,A,b,Aeq,beq,LB,UB)`:
  solves the problem `min f'x subject to:  A*x <= b`
- CPLEX

Two main solution methods:

- Simplex method: solves the problem by visiting extreme points, on the boundary of the feasible set, each time improving the cost
- Interior point methods: find an optimal solution while moving in the interior of the feasible set

Simplex method

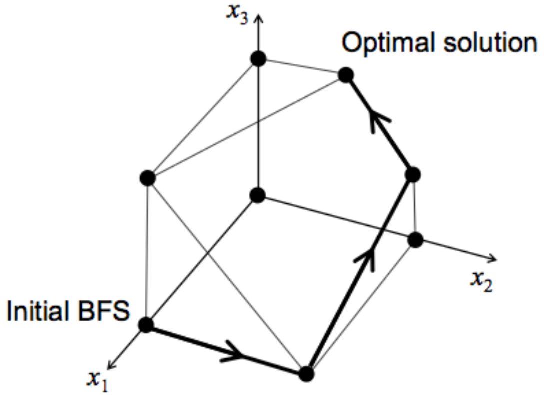

Figure: Geometric interpretation of simplex iterations. Image from MIT Open Courseware

Quadratic programming

The problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & c^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T Q\mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

where Q is an $n \times n$ positive semidefinite matrix

- Quadratic cost with linear constraints $\rightarrow$ convex optimization
- Used frequently in SCP, MPC
- Solved efficiently with interior point methods
- `X = quadprog(H,f,A,b)`
  solves the problem: `min 0.5*x'*H*x + f'*x subject to:`
  `A*x <= b`

Convex programming

- Models large class of control/mission planning problems
- Can be solved very efficiently (some of them even online!)

Function f is convex if domain is a convex set and

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$$

for all $x, y \in \mathrm{dom}(f), 0 \leq \alpha \leq 1$. If f convex, $-f$ is concave.

Convex functions: examples

On $\mathbb{R}$:

- affine: $ax + b$, any $a, b \in \mathbb{R}$ (also concave!)
- exponential: $\exp(ax)$, any $a \in \mathbb{R}$
- powers: x^a on $x > 0$, for $a \geq 1$ and $a \leq 0$ (concave for $0 \leq a \leq 1$)

On $\mathbb{R}^n$:

- affine: $\mathbf{a}^T\mathbf{x} + b$
- norms: $\|\mathbf{x}\|_p$ for $p \leq 1$

On $\mathbb{R}^n \times m$:

- affine: $\text{tr}(A^T X) + b = \sum_{i=1}^n \sum_{j=1}^m A_{ij}X_{ij} + b$
- spectral norm $\|X\|_2 = \sigma_{max}(X)$

Convex optimization problems

Optimization problem is convex if it takes the form

$$\begin{aligned} \min \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq 0, i = 1, \ldots, n \\ & h_i(\mathbf{x}) = 0, i = 1, \ldots, p \end{aligned}$$

and $f_0, \ldots, f_n$ are convex, and equality constraints are affine.

How can we check if our optimization problem is convex?

Checking convexity

How do we check the convexity of f?

- Verify definition (convex combinations are above function)
- Show $\nabla^2 f(\mathbf{x})$ is positive semi-definite everywhere
- Show that f can be written as known convex functions under convexity-preserving operations:
  - nonnegative weighted sum
  - composition with affine function
  - pointwise maximum and supremum
  - composition
  - minimization, $f(\mathbf{x}) = \inf_{\mathbf{y} \in \mathcal{C}} g(\mathbf{x}, \mathbf{y})$
  - perspective, $f(\mathbf{x}, t) = t\, f(\mathbf{x}/t)$

Convex programming tools

Tools

- Open-source software for convex optimization: `SeDuMi`: Primal-dual interior-point method
  `http://sedumi.ie.lehigh.edu/`
- `Yalmip`: User-friendly MATLAB interface
  `https://yalmip.github.io/`
- `CVX`: Matlab/Python/Julia-based modeling system
  `http://cvxr.com/cvx/`

Mixed-integer linear programming (MILP)

The problem:

$$\begin{aligned} \min \quad & c^T\mathbf{x} + d^T\mathbf{y} \\ \text{subject to} \quad & A\mathbf{x} + B\mathbf{y} \leq \mathbf{b} \\ & \mathbf{x}, \mathbf{y} \geq \mathbf{0} \\ & \mathbf{x} \text{ integer} \end{aligned}$$

- Same as the linear programming problem except that some of the variables are restricted to take integer values
- Captures **non-convexity** and **logic**
- Very powerful framework used in a variety of applications, e.g., task assignment, **mission planning**
- Much more difficult problem than linear programming

MILP modeling techniques: general

Examples:

- Binary choice
- Forcing constraints
- Relations between variables
- Disjunctive constraints
- Restricted range of values
- Arbitrary piecewise linear cost functions

MILP modeling techniques: general

Binary choice: A binary variable x can be used to encode a choice between two alternatives:

$$x_j \in \{0, 1\}$$

Example: n items each with weight w_j; total allowable weight K. Formulation:

$$\sum_{j-1}^{n} w_j x_j \leq K$$

$$x_j \in \{0, 1\} \quad j = 1, \ldots, n$$

Forcing constraints: Certain decisions are *dependent*. Example: decision A can be made only if decision B is made. Formulation: associate A (respectively B) with binary variable x_A (respectively x_B). Then pose the constraint:

$$x_A \leq x_B$$

$$x_A, x_B \in \{0, 1\}$$

MILP modeling techniques: general

Relations between variables: A constraint of the form

$$\sum_{j=1}^{n} x_j \le 1$$

$$x_j \in \{0,\ 1\} \quad j = 1, \ldots, n$$

i.e., **at most** one of the variables can be one

Disjunctive constraints: Big-M method: assume we want either $\mathbf{a}^T\mathbf{x} \le b$ or $\mathbf{c}^T\mathbf{x} \le d$. Formulation (M **big**):

$$\mathbf{a}^T\mathbf{x} \le b + M\,y_1$$

$$\mathbf{c}^T\mathbf{x} \le d + M\,y_2$$

$$y_1 + y_2 \le 1, \quad y_1, y_2 \in \{0, 1\}$$

MILP modeling techniques: general

Restricted range of values: Assume we want to restrict x to take values in $\{a_1, \ldots, a_m\}$. Formulation:

$$x = \sum_{j=1}^{m} a_j \, y_j$$

$$\sum_{j=1}^{m} y_j = 1$$

$$y_j \in \{0, 1\} \quad j = 1, \ldots, n$$

MILP modeling techniques: mission planning

Collision avoidance: we want to ensure

$$x \le x_l \quad \text{or} \quad y \le y_l \quad \text{or} \quad x \ge x_u \quad \text{or} \quad y \ge y_u$$

Formulation:

$$\begin{aligned}
x &\le x_l + My_1 \\
y &\le y_l + My_2 \\
x &\ge x_u - My_3 \\
y &\ge y_u - My_4 \\
\sum_{i=1}^{4} y_i &\le 3 \qquad y_i \in \{0, 1\}
\end{aligned}$$

MILP modeling techniques: mission planning

Minimum time of arrival: Assume we want to arrive at target $\mathbf{x}_f$ in minimum time. Formulation:

$$\mathbf{x}_f - M(1 - y_k) \leq \mathbf{x}(k) \leq \mathbf{x}_f + M(1 - y_k)$$

$$\sum_{k=1}^{N} y_k = 1$$

$$y_k \in \{0, 1\}$$

and the objective function would be

$$\min \sum_{k=1}^{N} k\, y_k$$

MILP modeling techniques: mission planning

Task assignment: Want to optimally assign n tasks to n UAVs.
Formulation:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, y_{ij} \\
\text{subject to} \quad & \sum_{i=1}^{n} y_{ij} = 1, \quad j = 1, \ldots, n \\
& \sum_{j=1}^{n} y_{ij} = 1, \quad i = 1, \ldots, n \\
& y_{ij} \in \{0, 1\}
\end{aligned}
$$

Special case: it can be solved with the simplex method!

MILP modeling: summary

- This collection of examples is not an exhaustive list
- Formulations are not unique, look for one with “few” variables and constraints
- Big-M method very handy but usually complicates the optimization process
- Modeling reference: Christodoulos A Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*, 1995.

MILP solution: Branch & Bound

- Divide and conquer approach to explore the set of feasible integer solutions
- Instead of exploring the entire feasible set, it uses bounds on the optimal cost to avoid exploring certain parts of the set

Let F be the set of feasible solutions to the problem

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in F \end{aligned}$$

Key 1: The set F is partitioned into a collection of subsets $F_1, F_2, \ldots, F_k$, and each of the following subproblems is solved separately:

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in F_i \end{aligned}$$

MILP solution: Branch & Bound

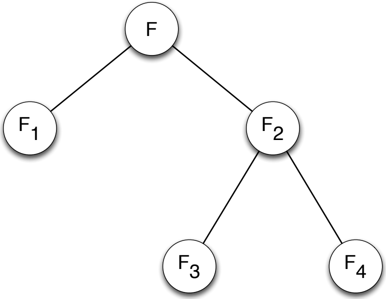

Figure: Tree of subproblems

Key 2: There is a fairly efficient algorithm to compute a lower bound $b(F_i)$ to the optimal cost, i.e.,:

$$b(F_i) \leq \min_{\mathbf{x} \in F_i} \mathbf{c}^T \mathbf{x}$$

MILP solution: Branch & Bound

Key 3: Let U be an upper bound on the optimal cost. If $b(F_i) \geq U$, then no need to consider this problem further

Algorithm:

1. Select an active subproblem F_i
2. If infeasible delete it, otherwise compute $b(F_i)$
3. If $b(F_i) \geq U$ delete the subproblem
4. If $b(F_i) < U$ either obtain an optimal solution or break the sub problem into further subproblems and add them to the list of active subproblems

Main "free parameters":

- Different ways of choosing the subproblems (e.g., breadth-first versus depth-first)
- Different ways of obtaining $b(F_i)$
- Several ways of breaking the problem

Tools to solve MILPs

Solvers:

- CPLEX[1]: `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`
- GLPK (GNU): `http://www.gnu.org/software/glpk/`

Interfaces:

- MATLAB
- AMPL[2]: `http://www.ampl.com/`

[1] Manual: `ftp://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmancplex.pdf`

[2] Manual: `http://www.ampl.com/BOOKLETS/amplcplex100userguide.pdf`

Conclusions

- MILP represent a powerful modeling framework, which together with control methods and some engineering insights can lead to sophisticated online mission planning

References for linear optimization and MILP:

- Dimitris Bertsimas, and John N. Tsitsiklis. *Introduction to linear optimization*. 1997.

References for MILP and control:

- Alberto Bemporad and Manfred Morari. *Control of systems integrating logic, dynamics, and constraints*. Automatica 35.3 (1999): 407-427.
- Arthur Richards and Jonathan P. How. *Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility*. American Control Conference, 2003.
- Tom Schouwenaars, Jonathan How, and Eric Feron. *Receding horizon path planning with implicit safety guarantees*. American Control Conference, 2004.
- Richards, Arthur, and Jonathan How. *Mixed-integer programming for control*. American Control Conference, 2005.