Basics about Optimization

REPLAN team

March 4, 2025

Outline

1. Preliminaries
2. Basic idea
3. Typical optimization problems
4. Study cases
5. Tools

Outline

What is an optimization problem?

$$
\begin{array}{rll}
\text{minimize} & f_0(x) & \\
\text{subject to} & f_i(x) \leq 0, & i = 1, \dots, m \\
 & g_i(x) = 0, & i = 1, \dots, p
\end{array}
$$

where

- $x \in \mathbb{R}^n$ is a (scalar or vector) to be chosen
- f_0 is the objective function, to be minimized (or maximized)
- $f_1, \dots, f_m$ are the inequality constraint functions
- $g_1, \dots, g_p$ are the equality constraint functions

- x may stand for schedule or assignment values / resource allocation
- constraints limit actions or impose conditions on outcome
- the objective may be a total cost, deviation from a target, fuel use...

Info

Instead of saying how to choose x, state the problem (cost and constraints) and let the algorithm return the optimal solution!

Brief history (according to S. Boyd)

- theory (convex analysis): 1900–1970
- algorithms
  - 1947: simplex algorithm for linear programming (Dantzig)
  - 1960s: early interior-point methods (Fiacco & McCormick, Dikin, ...)
  - 1970s: ellipsoid method and other subgradient methods
  - 1980s & 90s: interior-point methods (Karmarkar, Nesterov & Nemirovski)
  - since 2000s: many methods for large-scale convex optimization
- applications
  - before 1990: mostly in operations research, a few in engineering
  - since 1990: many applications in engineering (control, signal processing, communications, circuit design, ...)
  - since 2000s: machine learning and statistics, finance (the gradient descent method)

Potential uses

- identifying the parameters in a model (the cost penalizes loss on observed data and / or model complexity)
- worst-case analysis (the cost minimizes the worst possible parameter values)
- optimization-based models (currents in a circuit to minimize total power)

Caution!

Nonlinear optimization is, in general, much more complex to solve!

Hyperplane and half-space

Let's introduce the basic notions of

- **hyperplane**: the set of form

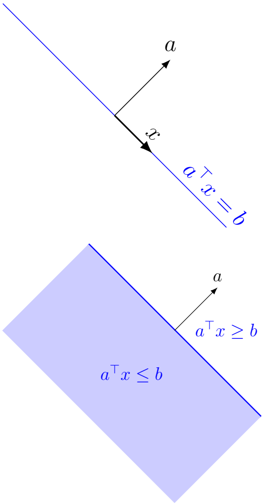
$$\left\{x \in \mathbb{R}^n : \; a^\top x = b\right\}$$

- **halfspace**: the set of form

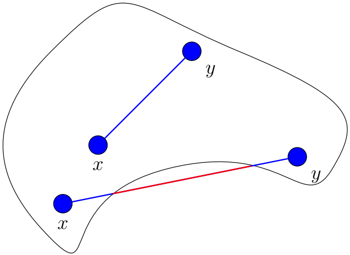
$$\left\{x \in \mathbb{R}^n : \; a^\top x \leq b\right\}$$

for $a \neq 0$ and $(a, b) \in \mathbb{R}^n \times \mathbb{R}$

A pair (a, b) will determine three sets, the halfspaces $\mathcal{H}^-, \mathcal{H}^+$ and the hyperplane $\mathcal{H}^0$ which is their separating boundary.



Convexity elements

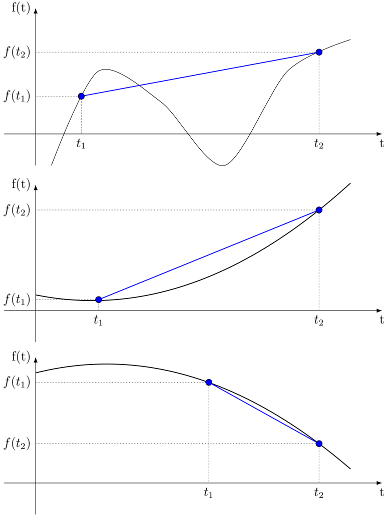


- set convexity:

$$\forall x, y \in S, \text{ we have } \lambda x + (1-\lambda) y \in S, \forall \lambda \in [0,1]$$

- function convexity:

$$f(\lambda t_1 + (1-\lambda) t_2) \leq \lambda f(t_1) + (1-\lambda) f(t_2), \forall t_1, t_2 \text{ and } \lambda \in [0,1]$$



Outline

Optimality conditions

- Assuming a convex problem, x is a solution if and only if it is feasible and

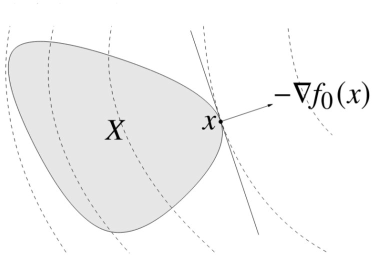
$$\nabla f_0(x)^\top (y - x) \geq 0, \quad \text{for all feasible } y$$

- for an unconstrained problem, the solution is found at

$$\min_x f_0(x) \qquad \Rightarrow \qquad \nabla f_0(x) = 0$$

- for a problem with equality constraints

$$\min_x f_0(x), \text{ s.t. } Ax = b \qquad \Rightarrow \qquad \nabla f_0(x) + A^\top \nu = 0$$



Duality

Info

Any optimization problem may be put in its dual form

Standard form

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \quad i = 1, \dots, m \\ & g_i(x) = 0, \quad i = 1, \dots, p \end{aligned}$$

Lagrangian

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

with

- λ_i is the Lagrange multiplier associated with $f_i(x) \leq 0$
- μ_i is the Lagrange multiplier associated with $g_i(x) = 0$

Computing $\max_{\lambda \geq 0} \min_x L(x, \lambda, \nu)$ gives a lower bound for $f_0(x^\star)$!

Karush-Kuhn-Tucker (KKT) conditions

If strong duality holds (strict convex cost and convex feasible domain), the optimization may be solved by ckecking the KKT conditions:

- primal constraints: $f_i(x) \leq 0,\ i = 1, \ldots, m$ and $h_i(x) = 0,\ i = 1, \ldots, p$
- dual constraints $\lambda \geq 0$
- complementarity slackness: $\lambda_i f_i(x) = 0,\ i = 1, \ldots, m$
- gradient of Lagrangian with respect to x vanishes:

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0$$

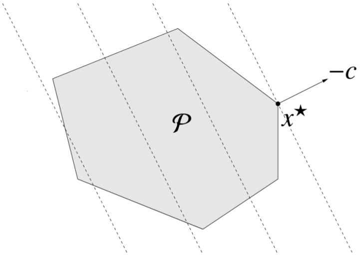
Nice tidbit, Farkas' lemma:

$$Ax \leq 0,\ c^\top x < 0 \qquad \text{OR} \qquad A^\top y + c = 0,\ y \geq 0$$

Outline

The linear program

$$\begin{aligned} \min_{x} \quad & c^\top x + d \\ \text{subject to} \quad & Gx \leq h \\ & Ax = b \end{aligned}$$

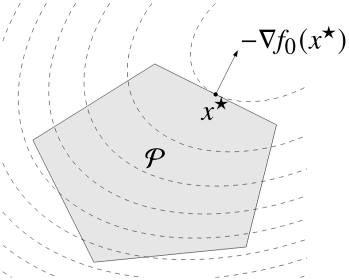


- convex problem with affine objective and constraint functions
- feasible set is a polyhedron
- if the solution is not degenerated, it is found in one of the domain's vertices

The quadratic program

$$
\begin{aligned}
\min_{x} \quad & x^\top P x/2 + q^\top x + r \\
\text{subject to} \quad & Gx \leq h \\
& Ax = b
\end{aligned}
$$

- with $P \succ 0$, the cost is convex
- feasible set is a polyhedron
- the solution is found on the boundary (not always on the domain's vertices)



Semi-definite program

$$
\begin{aligned}
\min_{x} \quad & c^\top x \\
\text{subject to } & G + F_1 x_1 + \ldots + F_n x_n \succeq 0 \\
& Ax = b
\end{aligned}
$$

with F_i, G positive definite matrices defining a Linear Matrix Inequality (LMI)

Example: matrix norm minimization

$$
\min_{x} \quad \|Ax\|_2 = \left[\lambda_{\max}\left(A(x)^\top A(x)\right)\right]^{1/2}
$$

with $A(x) = A_0 + A_1 x_1 + \ldots A_n x_n$ may be written as a SDP:

$$
\begin{aligned}
\min_{t,x} \quad & t \\
\text{subject to } & \begin{bmatrix} tI & A(x) \\ A(x)^\top & tI \end{bmatrix} \succeq 0
\end{aligned}
$$

Pareto optimality

- This time, the cost is a vector:

$$\begin{aligned} \min_x \quad & f_0(x) = (F_1(x), \ldots, F_q(x)) \\ \text{subject to} \quad & f_i(x), \quad i = 1, \ldots, m \\ & Ax = b \end{aligned}$$

- $x^\star$ should simultaneously minimize all F_i's; not an usual occurrence!
- Idea: we have a Pareto front where each solution is the best in at least a component of the cost vector
- scalarization combines multiple objectives into one, scalar, objective

$$\lambda^\top f_0(x) = \sum_{i=1}^{q} \lambda_i F_i(x)$$

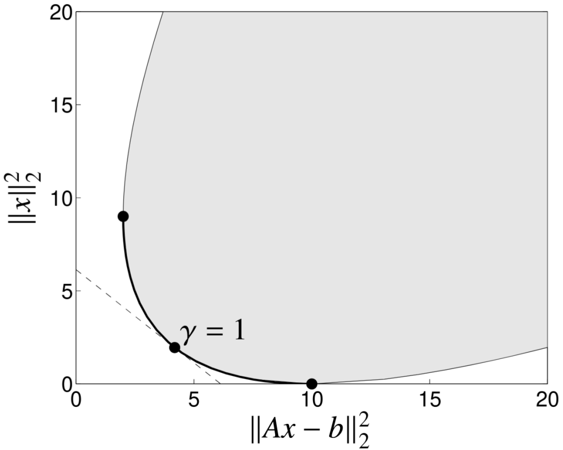
- if x is optimal for the scalar problem, then is Pareto-optimal for the original multi-criterion problem $\Rightarrow$ almost all Pareto optimal points may be found by varying $\lambda > 0$

Pareto optimality (example)

- Regularized least-squares problem:

$$\min_{x} \quad (\|Ax - b\|_2, \|x\|_2^2)$$

- scalarize the problem with, $\lambda = (1, \gamma > 0)$
- iterate for all feasible values of γ to define the Pareto front



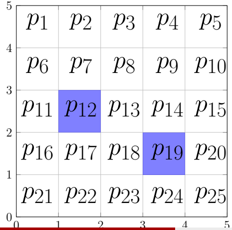
Mixed integer programs

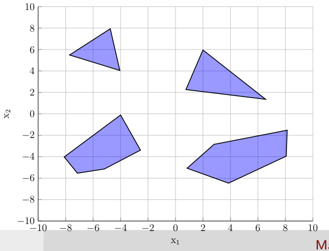
Mixed Integer Programming (MIP) is a branch of mathematical optimization where:

- (some) variables can take binary as well as integer values
- the goal is to find a solution that minimizes an objective function under a given set of constraints
- problems can easily grow to large sizes, execution time increases exponentially

MIP in motion planning:

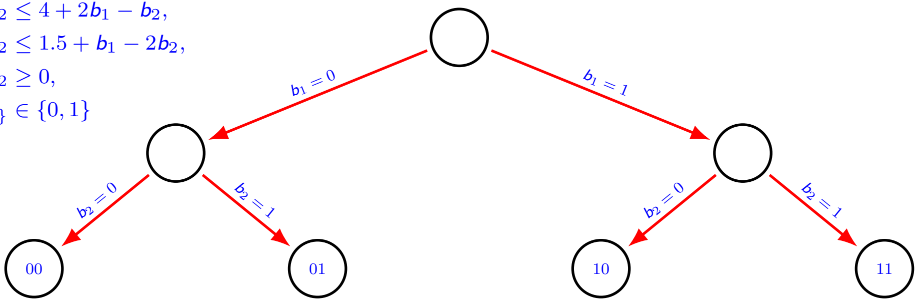
- algebraic/combinatorial: involves logical decisions and/or selection from a priori known alternatives
- geometrical: efficient mixed integer descriptions for non-convex regions





Mixed integer programs (example)

$$
\begin{aligned}
\min_{x,b} \quad & x_1 + 2x_2 - 2b_2 \\
\text{s.t.} \quad & 3x_1 + 0.5x_2 \leq 4 + 2b_1 - b_2, \\
& x_2 \leq 1.5 + b_1 - 2b_2, \\
& x_2 \geq 0, \\
& b_{\{1,2\}} \in \{0, 1\}
\end{aligned}
$$



$$
\begin{aligned}
\min_{x,b} \quad & x_1 + 2x_2 \\
\text{s.t.} \quad & 3x_1 + 0.5x_2 \leq 4, \\
& x_2 \leq 1.5, \\
& x_2 \geq 0,
\end{aligned}
$$

$$
\begin{aligned}
\min_{x,b} \quad & x_1 + 2x_2 - 2 \\
\text{s.t.} \quad & 3x_1 + 0.5x_2 \leq 3, \\
& x_2 \leq -0.5, \\
& x_2 \geq 0,
\end{aligned}
$$

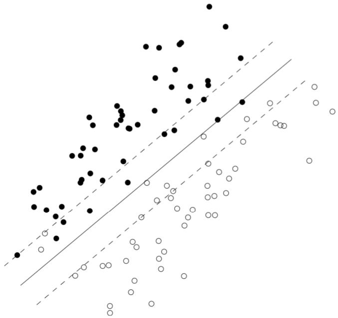
$$
\begin{aligned}
\min_{x,b} \quad & x_1 + 2x_2 \\
\text{s.t.} \quad & 3x_1 + 0.5x_2 \leq 6, \\
& x_2 \leq 2.5, \\
& x_2 \geq 0,
\end{aligned}
$$

$$
\begin{aligned}
\min_{x,b} \quad & x_1 + 2x_2 - 2 \\
\text{s.t.} \quad & 3x_1 + 0.5x_2 \leq 5, \\
& x_2 \leq 0.5, \\
& x_2 \geq 0,
\end{aligned}
$$

Outline

Approximate linear separation

$$\begin{aligned} \min_{a,b,u,v} \quad & 1^\top u + 1^\top v \\ \text{subject to} \quad & a^\top x_i \geq 1 - u_i, \quad i = 1, \dots, N, \\ & a^\top y_i \leq -1 + v_i, \quad i = 1, \dots, M, \\ & u \geq 0, \; v \geq 0 \end{aligned}$$

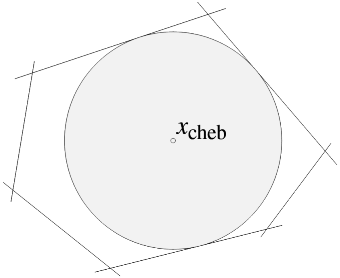


- minimize the sum of violations of the original inequalities
- ℓ_1 norm promotes sparsity so at optimum u_i, v_i are either zero or the quantity that breaks the constraints:

$$u_i = \max\{0, 1 - a^\top x_i - b_i\}, \qquad v_i = \max\{0, 1 + a^\top y_i + b_i\}$$

- change the cost to $\|a\|_2 + \gamma\left(1^\top u + 1^\top v\right)$ and we arrive at a support vector classifier

Chebisev circle



Find the largest circle inscribed inside a polytope:

$$\max_{r, x_c} \quad r$$
$$\text{subject to} \quad \{x : \; (x - x_c)^2 - r^2\} \subseteq \{x : \; Ax \leq b\}$$

Reformulation:

- consider the points x^i, $i = 1 : m$, located on the line defined by the center x_c
- and one of the directions of the half-spaces that define the polytope (a_i).
- Thus, these points can be written as:

$$x^i = x_c + \frac{1}{\|a_i\|} a_i \rho \quad \forall i \in \{1, \ldots, m\}$$

- and it is sufficient to check the validity of the half-space constraints $\left(a_i^\top x \leq b_i\right)$ only for these points x^i.

Fixed complexity idea (recap)

- Consider the discrete-time linear dynamics

$$x^+ = Ax + \delta, \quad \delta \in \Delta$$

- and polytopic sets, given in half-space formulation:

$$S(\theta) = \{x :\in \mathbb{R}^n : \ Fx \leq \theta\}, \quad \Delta = \{x :\in \mathbb{R}^m : \ F_\delta \delta \leq \theta_\delta\}$$

- The one-step forward invariant set is given by

$$x \in \Omega \xrightarrow{apply \ \ dynamics} x^+ \in \Omega^+ = A\Omega \oplus \Delta$$

The idea: keep the complexity constant!

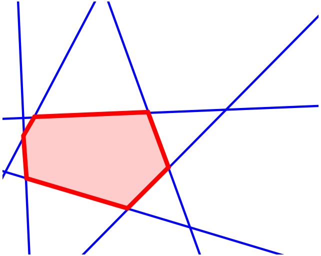
If $\Omega = S(\theta)$, find θ^+ such that $\Omega^+ \subseteq S(\theta^+)$

- Consider matrix $H \geq 0$ such that $HF = FA$ holds.
- Then, the set inclusion condition may be written as

$$AS(\theta) \oplus \Delta \subseteq S(\theta^+)$$

- which translates into the sufficient condition

$$H\theta + \max_{\delta \in \Delta} F\delta \leq \theta^+$$



Fixed complexity idea (applications)

- Smallest polyhedral over-approximation

$$\{F_1 x \leq \theta_1\} \subseteq \{F_2 x \leq \theta_2\}$$

$$\begin{aligned} \min_{H,\theta_2} \quad & 1^\top \theta_2 \\ \text{subject to} \quad & H \geq 0,\ HF_1 = F_2 \\ & H\theta_1 \leq \theta_2 \end{aligned}$$

- One-step RPI computation

$$A\{Fx \leq \theta\} \subseteq \{Fx \leq \theta\}$$

$$\begin{aligned} \min_{H,\theta} \quad & 1^\top \theta \\ \text{subject to} \quad & H \geq 0,\ HF = FA \\ & H\theta \leq \theta \end{aligned}$$

Min/max and scalar PWA modelling

Mixed-integer (MI) is useful whenever non-smooth functions have to be described:

- select the minimum from a list:

$$\underline{t} = \min_i x_i$$

MI form:

$$x_i - M(1 - z_i) \leq \underline{t} \leq x_i,$$
$$z_1 + \dots + z_n = 1,$$

- select the maximum from a list:

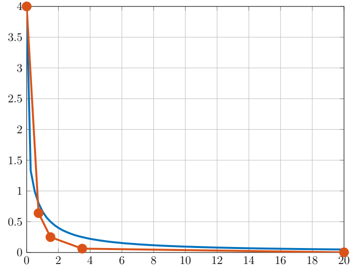
$$\overline{t} = \max_i x_i$$

MI form:

$$x_i \leq \overline{t} \leq x_i + M(1 - z_i),$$
$$z_1 + \dots + z_n = 1,$$

- epigraf of a scalar PWA function defined by $(x_i, f(x_i))$:

$$x = \sum \alpha_i x_i, \; \sum \alpha_i f(x_i) \leq t$$
$$\alpha_i \leq z_i + z_{i-1}, \; \alpha_i \geq 0, \sum \alpha_i = 1$$
$$\sum z_i = 1$$



Vertex-based modelling for an arbitrary PWA

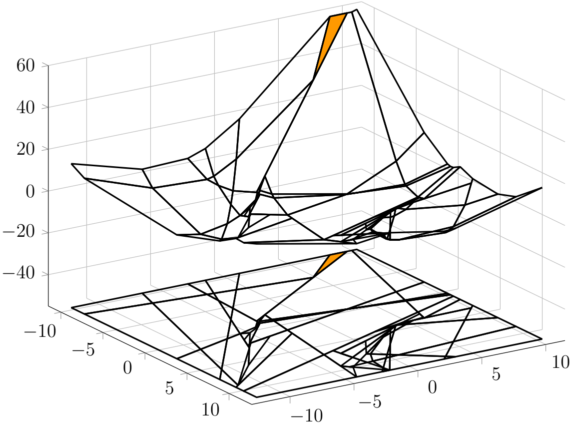
Consider the PWA function $f(x) : \mathbb{R}^n \mapsto R$ with support over the polyhedral partition $\bigcup_{i=1\ldots n} R_i$ where the vertices of all regions R_i are stored in $\mathbb{V} = \{v_j\}_{j=1\ldots m}$.

We give $f(x) \leq t$, the epigraph of $f(x)$, as:

$$x = \sum_{j=1}^{m} \alpha_j v_j, \qquad \sum_{j=1}^{m} \alpha_j f(v_j) \leq t,$$

$$\alpha_j \geq 0,\ \forall j = 1\ldots m, \qquad \sum_{j=1}^{m} \alpha_j = 1,$$

$$\alpha_j \leq \sum_{i:\ v_j \in R_i} z_i, \qquad \sum_{j=1}^{n} z_j = 1,$$



- generic formulation (holds for any PWA)
- but, the number of binary variables depends on the number of regions

Outline

Software tools

Many tools aim to reduce the effort in “preparing” the optimization problem:

- disciplined convex programming tool: CVX (Matlab, with the Python wrapper cvxpy)
- modeling language Yalmip (MATLAB), with extensive tutorials, examples and links to types of solvers
- modeling language Pyomo (Python)
- an automated differentiation tool: CasADi

Many solvers (see `https://yalmip.github.io/allsolvers/` for comprehensive details):

- IPOPT for nonlinear optimizations (with the interior point method)
- all-around (and commercial) solvers: MOSEK, GUROBI, CPLEX qhull
- OSQP: modern and easy to embedded in varied hardware

Relevant resources

- R. Fletcher (2000). *Practical methods of optimization*. John Wiley & Sons
- https://stanford.edu/~boyd/cvxbook/bv_cvxslides.pdf
- https://stellato.io/teaching/orf307/
- many others...