A Tour of Convex Optimization

Theo Diamandis

January 17, 2023

1 Convex Optimization Problem Classes

In this lecture, we will go over a number of classes of convex optimization problems, including linear programs, quadratic programs, second-order cone programs, and geometric programs. (We will see semidefinite programs later.) With software like `Convex.jl`, you neither need to know the problem class your problem belongs to nor how to reformulate it into a standard form. However, having some idea of the problem class will help you understand how hard your problem is to solve and what kind of solution you can expect. Importantly, there are specialized solvers for particular problem classes. Much of this lecture follows [BV04, Ch. 4].

Properties. Recall that a convex optimization problem is

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & Ax = b \\ & f_i(x) \leq 0, \quad i = 1, \ldots, m, \end{array} \tag{1}$$

where $x \in \mathbf{R}^n$ is the decision variable and the functions $f_0, \ldots, f_m$ are convex. These problems are particularly useful because

- All locally optimal points are globally optimal.[1]
- We can prove that a point is optimal (or provide a suboptimality gap, *i.e.,* a certificate that a point has objective value no worse than ε from the optimal value).
- They can be solved efficiently.

Let X denote the feasible set of (1) (which, importantly, is a convex set). Then $x \in X$ is optimal if and only if $x \in X$ and

$$\nabla f_0(x)^T (y - x) \geq 0 \quad \text{for all } y \in X.$$

[1] We call x locally optimal if there is an $R > 0$ such that for all feasible z with $\|z - x\|_2 < R$, $f_0(z) \geq f_0(x)$.

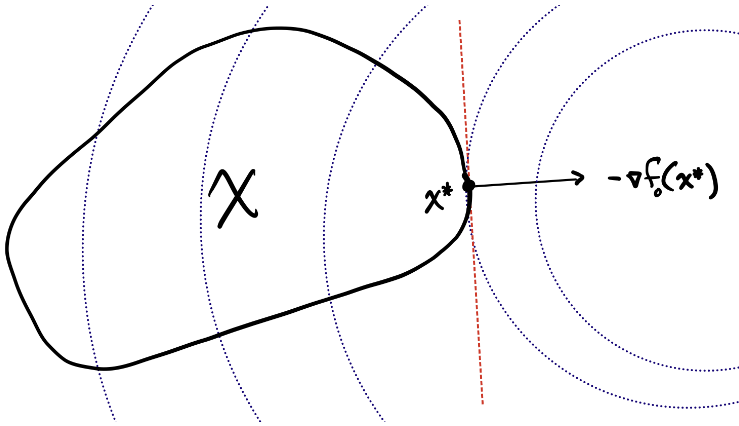


Figure 1: Illustration of optimality conditions. Blue dashed lines indicate level sets of f_0, and the red line indicates a supporting hyperplane at $x^\star$.

In general, we will call $x^\star$ a *solution* or *optimal point* if the condition above holds, and we will denote the *optimal value* $p^\star = f_0(x^\star)$. If the problem is infeasible, then $p^\star = \infty$, and if the problem is unbounded below, then $p^\star = -\infty$ (usually this means you setup your problem incorrectly). In other words, all feasible directions are aligned with the gradient, so a small step in that direction will not decrease the objective. This condition has a nice geometric interpretation. First, if $\nabla f_0(x) = 0$, then x is optimal because in any direction we move, the objective increases. (Recall that for unconstrained convex problems, the optimality condition is just that $\nabla f(x) = 0$.) Second, if $\nabla f_0(x) \neq 0$, then $-\nabla f_0(x)$ defines a suporting hyperplane to the feasible set X at the point x.

Tools of the trade A lot of the problems we'll look at in the course are not necessarily convex as written. We'll use a number of transformations to transform these problems into equivalent convex ones. Informally, we will say that two optimization problems are *equivalent* if the solution of one is readily obtained from the solution of the other, and vice-versa. For example, consider the optimization problem

$$\begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 \\ \text{subject to} & 2x_1/(1+x_2^2) + 1 \leq 0 \\ & (x_1 + x_2)^2 = 0. \end{array}$$

This problem is not convex but is equivalent to the convex problem

$$\begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 \\ \text{subject to} & 2x_1 + 1 + x_2^2 \leq 0 \\ & x_1 + x_2 = 0. \end{array}$$

Often you will have to work with problems a bit to get them into a form that is tractable to solve. Common transformations include

- Adding slack variables for linear inequalities
- Introducing equality constraints
- Using the epigraph of a function
- Minimizing over a subset of variables (partial minimization)
- Doing a change of variables (*e.g.,* log transform)
- Taking a monotone increasing function of the objective
- Dropping constants in the objective

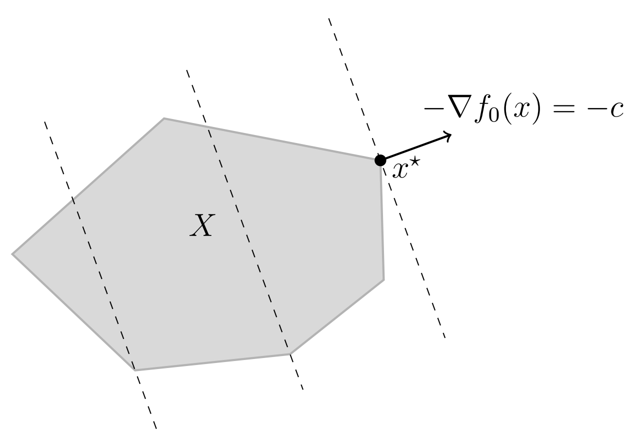
Over this lecture, we'll see these transformations come up in problem classes. `Convex.jl` automates many of these transformations for you to find an equivalent problem to your input that can be passed to a solver.

1.1 Linear program (LP)

In linear programs, the objective and constraint functions are affine. These problems commonly show up in operations research (*e.g.,* scheduling, supply chain management, transportation planning, etc.). They are of the form

$$\begin{array}{ll} \text{minimize} & c^T x + d \\ \text{subject to} & Gx \leq h \\ & Ax = b. \end{array} \tag{2}$$

The feasible set of an LP (*i.e.,* $X = \{x \mid Gx \leq h \text{ and } Ax = b\}$) is a polyhedron, and if the problem is feasible, we can always find an optimal solution at a vertex.[2] This fact is intuitive when considering the geometry of the feasible set and the gradient of the objective:



[2]This fact is important for the simplex method, which hops from vertex to vertex until it finds an optimal solution.

The dashed lines indicate level sets of $-\nabla f_0(x)$. If the negative gradient is orthogonal to a face of the polyhedron, we can find a solution at the face's vertices, although there will be many solutions (*i.e.,* points with the optimal objective value). Also note that we can always drop constants in the objective without changing the optimal solution (but the optimal value may change).

Sometimes you will see these problems written in the 'standard form'

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0. \end{array}$$

This form is, in fact, equivalent to (2). We can see this by applying two tricks. First, every number can be written as the sum of its positive and its negative part, *i.e.,*

$$x = x_+ - x_-,$$

where $x_+ = \max\{x, 0\}$ and $x_- = \max\{-x, 0\}$. Note that $x_+, x_- \geq 0$. Second, we can add *slack variables* to turn inequality constraints in (2) to equality constraints:

$$Gx \leq h \iff Gx + s = h, \quad s \geq 0.$$

These two tricks will come up often. Convince yourself that the 'standard form' of the LP is indeed equivalent to our general form (2). While some of these tricks aren't important for us in this course, they are especially important for solvers, which usually specialize on a particular form of the problem.

Example: transportation planning. First we consider a transportation planning problem where we want to ship goods from m sources to n destinations in a way that minimizes the total shipping cost. Each source has a supply s_i and each destination has a demand d_j. The shipping cost from source i to destination j is c_{ij}. This problem can be formulated as

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{ij} c_{ij} x_{ij} \\ \text{subject to} & \displaystyle\sum_j x_{ij} \leq s_i, \quad i = 1, \ldots, m \\ & \displaystyle\sum_i x_{ij} \geq d_j, \quad j = 1, \ldots, n \end{array}$$

where $s \in \mathbf{R}^m_+$ is the supply at each source, $d \in \mathbf{R}^n_+$ is the demand at each source, c_{ij} is the per unit shipping cost from i to j, and the variables x_{ij} are the amount of goods shipped from i to j. Since the objective and all constraints are affine, this is an LP.

Example: basis pursuit. Consider the problem of recovering a sparse vector x from a linear measurements $b_i = a_i^T x$, $i = 1, \dots, m$:

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = b \end{array}$$

(If you're interested in applications of this problem, Google 'compressed sensing'.) Recall from the first lecture that we can reformulate ℓ_1 objectives into LPs via the *epigraph* and the fact that $|y| \le z \iff -z \le y \le z$. The equivalent problem is

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T t \\ \text{subject to} & -t \le x \le t \\ & Ax = b. \end{array}$$

Note that the variables are $t \in \mathbf{R}^n$ and $x \in \mathbf{R}^n$, and the inequality is taken componentwise. Alternatively, we could split up x into its positive and negative parts and note that $|y| = y_+ + y_-$. This trick gives the equivalent problem

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T (x_+ + x_-) \\ \text{subject to} & Ax_+ + Ax_- = b \\ & x_+, x_- \ge 0. \end{array}$$

Clearly, this problem is also an LP.

Linear-fractional program. It turns out that we can also solve problems of the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & Gx \le h \\ & Ax = b, \end{array} \tag{3}$$

where the objective is a fraction of two linear functions:

$$f_0(x) = \frac{c^T x + d}{e^T x + f}, \qquad \mathbf{dom}\, f_0(x) = \{x \mid e^T x + f > 0\}.$$

This problem is quasiconvex (we will talk about quasiconvexity shortly), but it is also equivalent to the LP

$$\begin{array}{ll} \text{minimize} & c^T y + dz \\ \text{subject to} & Gy \le hz \\ & Ay = bz \\ & e^T y + fz = 1 \\ & z \ge 0. \end{array}$$

To show this equivalence, consider an x that is feasible for the original problem. Then $y = x/(e^T x + f)$ and $z = 1/(e^T x + f)$ are feasible for the LP and give the same objective value. Thus, the optimal value of the LP is less than or equal to that of the original problem. Reverse the argument by considering a feasible (y, z) and taking $x = y/z$ where $z \neq 0$ (a more careful argument must be made when $z = 0$). Thus the two problems have the same optimal value and are equivalent insofar as the solution to one can be readily obtained from the solution to the other.

1.2 Quadratic programs

A quadratic program replaces the linear objective with a quadratic one:

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x + r \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array}$$

where $P \in \mathbf{S}^n_+$, so the objective is convex. Again, our feasible set is a polyhedron. Clearly the set of QPs is a strict superset of the set of LPs. In lecture 1, we saw the radiation treatment planning problem, which is a bounded least squares problem and therefore a QP.

Portfolio optimization. Consider the problem of optimizing a portfolio of n assets. We want to find the portfolio that maximizes the expected return while keeping the risk (measured by the variance of the returns) in control. Thus, our objective function is

$$f_0(x) = \mathbb{E}\left[\mu^T x\right] - (\gamma/2)\mathrm{Var}\left(\mu^T x\right),$$

where $\mu \in \mathbf{R}^n$ is the return on each asset (a random variable), $x \in \mathbf{R}^n$ is our portfolio allocation, and γ is a risk aversion parameter. Additionally, we will require a long-only portfolio, *i.e.,* $x \geq 0$. Letting $\bar{\mu} = \mathbb{E}[\mu]$, we can write this problem can as a QP:

$$\begin{array}{ll} \text{maximize} & \bar{\mu}^T x - (\gamma/2)x^T \Sigma x \\ \text{subject to} & \mathbf{1}^T x = 1 \\ & x \geq 0. \end{array}$$

where $\Sigma \in \mathbf{R}^{n \times n}$ is the covariance matrix of returns. The variable x represents the fraction of the portfolio allocated to each asset, which is why we normalize it to sum to 1. Alternatively, we could have used the constraint $\mathbf{1}^T x \leq B$, where B is our total investment budget.

Pareto optimality. In the proceeding example, we have two objectives: the expected return $\bar{\mu}^T x$ and the risk $-x^T \Sigma x$. We want both of these to be small. In general, we cannot minimize both of these objectives simultaneously; there's some tradeoff. We say that x^{po} is *Pareto optimal* if there is no other feasible point that is better for at least one objective and no worse for any of the others. The set of pareto optimal points can be found by scalarization of the objective and solving the resulting problem for a range of values of the scalarization

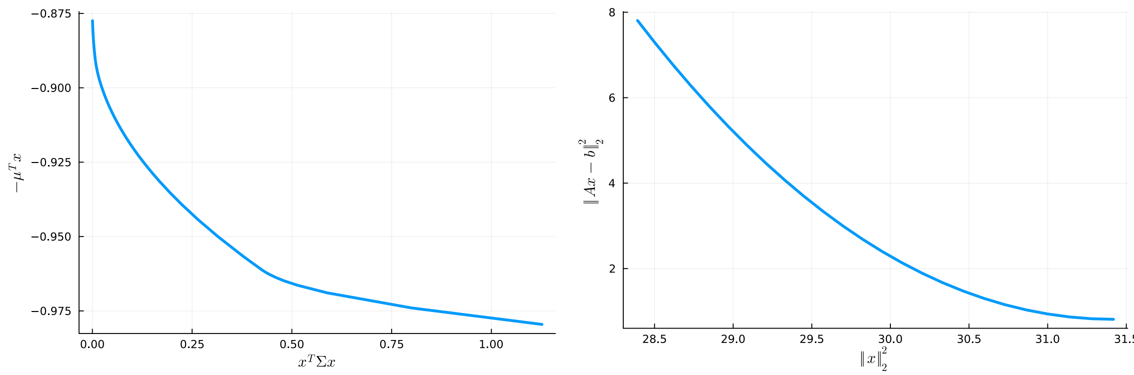


Figure 2: Pareto optimal frontiers for randomly generated instances of the portfolio optimization problem (left) and regularized least squares problem (right). Note that we plot the negative return in the portfolio optimization example.

parameters (usually taken to be on a log scale). Another example is the regularized least squares problem

$$\text{minimize} \quad (1/2)\|Ax - b\|_2^2 + \lambda\|x\|_2^2,$$

where varying λ traces out the Pareto frontier.

Quadraticly constrained quadratic program (QCQP). A close relative of the QP is the QCQP, which adds quadratic constraints:

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P_0 x + q_0^T x + r \\ \text{subject to} & (1/2)x^T P_i x + q_i^T x + r_i \leq 0, \qquad i = 1, \ldots, m \\ & Ax = b. \end{array}$$

Note that this class is a superset of the QP (let $P_i = 0$ for all i). If $P_i \in \mathbf{S}_{++}^n$, then the feasible set is the intersection of m ellipsoids and the affine set $\{x \mid Ax = b\}$.

Rocket landing. Model predictive control (MPC), sometimes called receding horizon control, chooses the control input by repeatedly solving a convex optimization problem. In the case of landing a rocket, we will discretize time from $t = 0$ to T into K steps. Thus, each step has length $h = T/K$. At each time step, we solve a convex optimization problem to determine the thrust force to apply. The solution generates a plan for the entire trajectory. In a real control system, we execute one step of this plan and then re-solve the problem with updated parameters and a smaller discretization step h. In this example, we'll setup the problem for $t = 0$.

The (discretized) rocket dynamics are given by

$$v_{k+1} = v_k + (h/m)f_k - hge_3, \qquad \text{and} \qquad p_{k+1} = p_k + (h/2)(v_k + v_{k+1}),$$

where $p_k \in \mathbf{R}^3$ denotes the position, $v_k \in \mathbf{R}^3$ denotes the velocity, $f_k \in \mathbf{R}^3$ denotes the thrust force at step k, m is the rocket mass (assumed constant for simplicity) and g is the gravitational acceleration. We are limited to thrusts f_k less than some maximum $F^{\max}$. The rocket starts at $p(0)$ with velocity $v(0)$, and we want to land the rocket at the origin at time T, *i.e.*, $p_K = 0$ and $v_K = 0$. Furthermore, the rocket must remain in the region

$$(p_k)_3 \geq \alpha \|((p_k)_1, (p_k)_2)\|_2,$$

where α indicates the minimum glide slope (what type of convex set is this?). Finally, we aim to find a trajectory that minimizes the (discretized) fuel use,

$$\sum_{k=1}^{K-1} \|f_k\|_2.$$

Finally, putting this all together, we have the optimization problem

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{k=0}^{K-1} \|f_k\|_2 & \\
\text{subject to} & v_{k+1} = v_k + (h/m)f_k - hge_3, & k = 0, \ldots, K-1 \\
& p_{k+1} = p_k + (h/2)(v_k + v_{k+1}), & k = 0, \ldots, K-1 \\
& \|f_k\| \leq F^{\max} & k = 0, \ldots, K \\
& (p_k)_3 \geq \alpha \|((p_k)_1 . (p_k)_2)\|_2, & k = 0, \ldots, K \\
& p_K = 0, \quad v_K = 0, \quad p_0 = p(0), \quad v_0 = \dot{p}(0). &
\end{array}$$

For some videos of this idea in action, check out

- http://www.youtube.com/watch?v=2t15vP1PyoA
- https://www.youtube.com/watch?v=orUjSkc2pG0
- https://www.youtube.com/watch?v=1B6oiLNyKKI
- https://www.youtube.com/watch?v=ZCBE8ocOkAQ

1.3 Second-order cone program (SOCP)

The next step our hierarchy is the second-order cone program (SOCP). These problems have the form

$$\begin{array}{lll}
\text{minimize} & f^T x & \\
\text{subject to} & \|A_i x + b_i\|_2 \leq c_i^T x + d_i, & i = 1, \ldots, m \\
& Fx = g &
\end{array}$$

This form is more general than LPs, QPs, and QCQPs. For example, we can write a QP as an SOCP by introducing a slack variable t and writing

$$\begin{array}{ll}
\text{minimize} & q^T x + (1/2)t \\
\text{subject to} & Dx \leq d \\
& x^T Q x \leq t \\
& Ax = b
\end{array}$$

Note that $x^TQx \le t \iff \|(Q^{1/2}x, (t-1)/2)\|_2 \le (t+1)/2$.

Example: facility location. An urban planner wants to choose a location $x \in \mathbf{R}^2$ for a new warehouse that minimizes the worst-case distance to n distribution centers, located at $y_1, \ldots, y_n$. This problem can be written as

$$\text{minimize} \quad \max_{k=1,\ldots,n} \|y_k - x\|_2$$

This problem is equivalent to the second order cone program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subect to} & \|y_k - x\| \le t, \quad k = 1, \ldots, n. \end{array}$$

Here, we used an epigraph transformation.

Robust linear programming. Often, the parameters in optimization problems are uncertain. In our supply example, we might not know the exact demand and instead have an estimate. Working with the expected values can produce very 'fragile' solutions (*i.e.,* a small change in the parameters can cause the computed optimal solution to become infeasible). Consider the LP

$$\begin{array}{ll} \text{minimize} & c^Tx \\ \text{subject to} & a_i^Tx \le b, \qquad i = 1, \ldots, m. \end{array}$$

There are two common approaches to handling uncertainty (we assume only the a_i's have uncertainty, without loss of generality).

First, a deterministic approach: we require the constraints to hold for all a_i in some uncertainty set $\mathcal{E}_i$. A common choice is the ellipsoidal uncertainty set.

$$\mathcal{E}_i = \{\bar{a}_i + P_iu \mid \|u\|_2 \le 1\},$$

where the expected value of a_i, denoted $\bar{a}_i$, is the center of the ellipsoid, whose shape is defined by the singular values and vectors of P_i. Then the robust LP

$$\begin{array}{ll} \text{minimize} & c^Tx \\ \text{subject to} & a_i^Tx \le b \quad \forall a_i \in \mathcal{E}_i, \qquad i = 1, \ldots, m \end{array}$$

is equivalent to the SOCP

$$\begin{array}{ll} \text{minimize} & c^Tx \\ \text{subject to} & a_i^Tx + \|P_i^Tx\|_2 \le b, \qquad i = 1, \ldots, m. \end{array}$$

This result holds because $\sup_{\|u\|_2 \le 1} (\bar{a}_i + P_iu)^Tx = \bar{a}_i^Tx + \|P_i^Tx\|_2$.

Second, a stochastic approach: we require the constraints to hold with probability η. We assume that a_i is Gaussian with mean $\bar{a}_i$ and covariance Σ_i. Then

$$\mathbb{P}\big(a_i^Tx \le b\big) = \Phi\left(\frac{b - \bar{a}_i^Tx}{\|\Sigma_i^{1/2}x\|_2}\right),$$

where Φ is the cumulative distribution function of the standard normal. This fact means that the stochastic LP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \mathbb{P}\big(a_i^T x \leq b\big) \geq \eta \qquad i = 1, \ldots, m \end{array}$$

with $\eta \geq 1/2$ is equivalent to the SOCP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \bar{a}_i^T x + \Phi^{-1}(\eta) \|\Sigma_i^{1/2} x\|_2 \leq b_i \qquad i = 1, \ldots, m. \end{array}$$

1.4 Geometric programs

Geometric programs allow us to tackle a wide class of problems with convex optimization by using a log transform of the variables (this is a very powerful trick!). A *monomial* function has the form

$$f(x) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}, \qquad \mathbf{dom}\, f = \mathbf{R}^n_{++},$$

with $c > 0$ and $a_i \in \mathbf{R}$. A *posynomial* is the sum of monomials. A geometric program has the form

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 1, & i = 1, \ldots, m \\ & h_i(x) = 1, & i = 1, \ldots, p \end{array} \tag{4}$$

with f_i posynomial and h_i monomial. Since the variables x_i are positive, we can use the log transform, $y = \log x$. For a monomial f, we have

$$f(e^{y_1}, \ldots, e^{y_n}) = \exp\left(\sum_{i=1}^n a_i y_i + b\right) = \exp(a^T y + b).$$

Taking the logarithm of the objective (why can we do this?) and constraints in (4), we get the equivalent convex problem

$$\begin{array}{ll} \text{minimize} & \log\left(\displaystyle\sum_{k=1}^K \exp\left(a_{0k}^T y + b_{0k}\right)\right) \\ \text{subject to} & \log\left(\displaystyle\sum_{k=1}^K \exp\left(a_{ik}^T y + b_{ik}\right)\right) \leq 0, \qquad i = 1, \ldots, m \\ & Gy + d = 0. \end{array}$$

Geometric programs often come up in engineering design applications.[3]

[3]For some examples, see `https://web.stanford.edu/~boyd/papers/gp_tutorial.html`. The digital circuit gate sizing example comes from this paper.

Example: pipe design. A heated fluid at temperature T (degrees above ambient temperature) flows in a pipe with fixed length and circular cross section with radius r. A layer of insulation, with thickness $w \ll r$, surrounds the pipe to reduce heat loss through the pipe walls. The design variables in this problem are T, r, and w. The heat loss is (approximately) proportional to Tr/w, so over a fixed lifetime, the energy cost due to heat loss is given by $\alpha_1 Tr/w$. The cost of the pipe, which has a fixed wall thickness, is approximately proportional to the total material, *i.e.,* it is given by $\alpha_2 r$. The cost of the insulation is also approximately proportional to the total insulation material, *i.e.,* $\alpha_3 rw$ (using $w \ll r$). The total cost is the sum of these three costs. The heat flow down the pipe is entirely due to the flow of the fluid, which has a fixed velocity, *i.e.,* it is given by $\alpha 4Tr^2$. The constants α_i are all positive, as are the variables T, r, and w. Now the problem: maximize the total heat flow down the pipe, subject to an upper limit C max on total cost, and the constraints

$$T_{\min} \leq T \leq T_{\max}, \qquad r_{\min} \leq r \leq r_{\max}, \qquad w_{\min} \leq w \leq w_{\max}, \qquad w \leq 0.1r.$$

The problem can be expressed as a geometric program as given. The problem is

$$\begin{array}{ll} \text{maximize} & \alpha_4 T r^2 \\ \text{subject to} & \alpha_1 T r w^{-1} + \alpha_2 r + \alpha_3 r w \leq C_{\max} \\ & T_{\min} \leq T \leq T_{\max} \\ & r_{\min} \leq r \leq r_{\max} \\ & w_{\min} \leq w \leq w_{\max} \\ & w \leq 0.1r. \end{array}$$

A bit of algebra yields the GP

$$\begin{array}{ll} \text{minimize} & \alpha_4^{-1} T^{-1} r^{-2} \\ \text{subject to} & C_{\max}^{-1}\alpha_1 T r w^{-1} + C_{\max}^{-1}\alpha_2 r + C_{\max}^{-1}\alpha_3 r w \leq 1 \\ & T_{\min} T^{-1} \leq 1, \qquad T_{\max}^{-1} T \leq 1 \\ & r_{\min} r^{-1} \leq 1, \qquad r_{\max}^{-1} r \leq 1 \\ & w_{\min} w^{-1} \leq 1, \qquad w_{\max}^{-1} w \leq 1 \\ & 10 w r^{-1} \leq 1. \end{array}$$

Digital circuit gate sizing. *(From [Boy+07].)* We consider a digital circuit, consisting of a number of logic gates, each with one output and one or more inputs. A path through the circuit is a sequence of gates from a circuit input to a circuit output (*e.g.,* $1 \to 3 \to 6$ in Figure 3). Consider the simple circuit in Figure 3. There are five paths through the circuit. For each gate $i = 1, \ldots, n$, we will choose a scale factor $x_i \geq 1$. The total circuit area is then

$$A = \sum_{i=1}^{n} a_i x_i = a^T x,$$

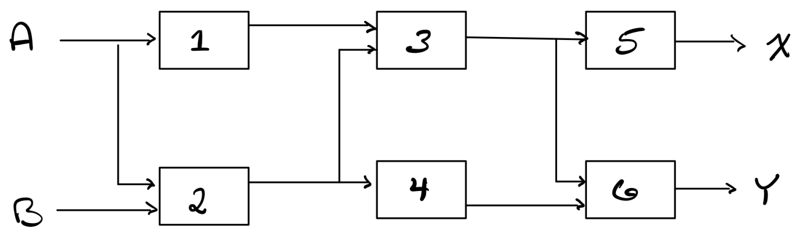


Figure 3: Digital circuit used in GP example.

where a_i is the area of gate i with unit scaling. The total power consumed is

$$P = \sum_{i=1}^{n} f_i e_i x_i = p^T x,$$

where f_i is the frequency of gate i and e_i is the energy lost when it transitions. Each gate's input capacitance C_i is an affine function of the scale factor, and its driving resistance is inversely proportional to the scale factor:

$$C_i = \alpha_i + \beta_i x_i, \qquad R_i = \gamma_i / x_i.$$

Finally, the delay D_i of a gate is the product of its driving resistance and the sum of the input capacitances of the gates its output is connected to,

$$D_i = \begin{cases} R_i \sum_{j \in F(i)} C_j & \text{if } i \text{ is not an output gate} \\ R_i C_i^{\text{out}} & \text{if } i \text{ is an output gate,} \end{cases}$$

where $F(i)$ is the set of gates whose input is connected to the output of gate i and C_i^{out} is the load capacitance of the output gate. We can see that D_i is a posynomial function of the scale factors. Our goal is to choose the scale factors that minimize the worst-case delay D for any path through the circuit. For our circuit, there are only 5 paths through the gates: $1,3,5$, $1,3,6$, $2,3,5$, $2,3,6$, and $2,4,6$. Thus, the worst-case delay is

$$D = \max\{D_1 + D_3 + D_5, D_1 + D_3 + D_6, D_2 + D_3 + D_5, D_2 + D_3 + D_6, D_2 + D_4 + D_6\}.$$

We will constrain $P \leq P^{\max}$ and $A \leq A^{\max}$. It is clear that P, A, and D_i are posynomials. We can reformulate the max function as a GP by noticing that if $f_1(x)$ and $f_2(x)$ are posynomials, then we can rewrite $\max\{f_1(x), f_2(x)\}$ using the epigraph trick. This allows us to find the minimum delay circuit by solving the geometric program

$$
\begin{array}{ll}
\text{min.} & t \\
\text{s.t.} & D_1 + D_3 + D_5 \leq t, \qquad D_1 + D_3 + D_6 \leq t \\
& D_2 + D_3 + D_5 \leq t, \qquad D_2 + D_3 + D_6 \leq t \\
& D_2 + D_4 + D_6 \leq t \\
& D_i = \begin{cases} R_i \sum_{j \in F(i)} C_j & \text{if } i \text{ is not an output gate} \\ R_i C_i^{\text{out}} & \text{if } i \text{ is an output gate} \end{cases} \\
& C_i = \alpha_i + \beta_i x_i, \qquad R_i = \gamma_i / x_i \\
& a^T x \leq A^{\max}, \qquad p^T x \leq P^{\max} \\
& 1/x_i \leq 1.
\end{array}
$$

1.5 Log-concavity

A function f is *log-concave* if $\log f$ is concave. We mostly care about log-concave (vs. log-convex) functions. Many common probability densities (*e.g.,* Gaussians) are log-concave. Even the uniform distribution is log concave (think of the extended value extension of this function). Simple transformation of the definition of concavity yields the condition

$$f(\lambda x + (1-\lambda)y) \geq f(x)^\lambda f(y)^{1-\lambda}$$

for $\lambda \in [0,1]$. This says that the function at an average is greater than the geometric mean of the endpoints.

Some properties. Many important composition properties of log-concave functions can be worked out in the same way we worked out properties of convex functions. Many have natural probability interpretations. These properties include

- Product preserves log-concavity (*e.g.,* joint distribution of independent random variables)
- The sum does **NOT** necessarily preserve log-concavity (*e.g.,* mixture distributions are not necessarily log-concave)
- Integration preserves log-concavity (*e.g.,* marginal distributions & CDFs of log-concave distributions)
  - Convolutions are log-concave, so the pdf of the sum of random variables with log-concave densities is log-concave.
  - If C is a convex set and y is a random variable with a log-concave pdf, then $f(x) = \mathbb{P}(x + y \in C)$ is log-concave (*e.g.,* yield functions)

2 Quasiconvex optimization

A function f is quasiconvex is $\mathbf{dom}\, f$ is convex and the sublevel sets

$$S_\alpha = \{x \in \mathbf{dom}\, f \mid f(x) \leq \alpha\}$$

are convex for all $\alpha \in \mathbf{R}$. Equivalently, we have that

$$f(\lambda x + (1-\lambda)y) \leq \max\{f(x), f(y)\}$$

for all x, y in $\mathbf{dom}\, f$ and $\lambda \in [0,1]$. In $\mathbf{R}$, these functions are unimodal: they are monotone decreasing to some point and then increasing afterwards. Similar to convexity, we say that f is quasiconcave if $-f$ is quasiconvex (equivalently, if every superlevel set is convex). Finally, we say that f is quasilinear if it is both quasiconvex and quasiconcave. Some examples of these functions are

- $\sqrt{|x|}$ is quasiconvex on $\mathbf{R}$
- $\lceil x \rceil = \mathbf{ceil}(x) = \inf\{z \in \mathbf{Z} \mid z \geq x\}$ is quasilinear
- $\log x$ is quasilinear on $\mathbf{R}_{++}$
- $f(x) = x_1 x_2$ is quasiconcave on $\mathbf{R}^2_{++}$
- The linear fractional function we defined previously is quasilinear where the denominator is positive
- The distance ratio $f(x) = \|x-a\|_2/\|x-b\|_2$ is quasiconcave on $\mathbf{dom}\, f = \{x \mid \|x-a\|_2 \leq \|x-b\|_2\}$

A more complicated example is the internal rate of return, defined as

$$\mathrm{IIR}(x) = \inf\left\{r \geq 0 \mid \sum_{i=0}^{n} x_i(1+r)^{-i} = 0\right\}.$$

This is the smallest interest rate r such that the present value of the cash flows x_i is zero. We assume that $x_0 < 0$ and $x_0 + \cdots + x_n > 0$. This function is quasiconcave on $\mathbf{R}^n_{++}$, since the superlevel set is the intersection of open halfspaces:

$$\{x \mid \mathrm{IIR}(x) \geq R\} = \bigcap_{0 \leq r < R} \left\{x \mid \sum_{i=0}^{n} \frac{x_i}{(1+r)^i} > 0\right\}$$

Note that the sum of quasiconvex functions is **NOT** quasiconvex. For example, try adding two unimodal functions in $\mathbf{R}$ (*e.g.*, $\sqrt{|x|}$ and $\sqrt{|x-2|}$). Like convexity, a number of operations preserve quasiconvexity. Importantly, if f_i is quasiconvex, then the nonnegative weighted maximum

$$f(x) = \max\{w_1 f_1(x), \ldots, w_m f_m(x)\}$$

with $w_i \geq 0$ is also quasiconvex.

Solving quasiconvex problems. If f_0 is quasiconvex, then there exists a family of functions ϕ_t such that ϕ_t is convex in x for a fixed t and the t-sublevel set of f_0 is a 0-sublevel set of ϕ_t:

$$f_0(x) \leq t \iff \phi_t(x) \leq 0.$$

To see that such a representation always exists, recall that the sublevel set is convex and consider

$$\phi_t(x) = \begin{cases} 0 & f(x) \leq t \\ \infty & \text{otherwise.} \end{cases}$$

For a more practical example, consider the function $f_0(x) = p(x)/q(x)$ where p is convex, q is concave, and we have that $p(x) \geq 0$ and $q(x) > 0$ on $\mathbf{dom}\, f_0$. Then we can take

$$\phi_t(x) = p(x) - tq(x),$$

which is convex for $t \geq 0$. This formulation allows us to solve quasiconvex optimization problems via a series of convex feasibility problems

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & \phi_t(x) \leq 0 \\ & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array} \tag{5}$$

If (5) is feasible, then $t \geq p^\star$. If not, $t \leq p^\star$. Given bounds l and u such that $p^\star \in [l, u]$ and some tolerance ε, the bisection method for quasiconvex optimization is

1. Let $t = (l + u)/2$
2. Solve the convex feasibility problem (5)
3. If (5) is feasible, set $u = t$, else $l = t$
4. Repeat until $u - l \leq \varepsilon$

This method requires exactly $\lceil \log_2(1/\varepsilon) \rceil$ iterations. Note that at termination, you have both a feasible point with objective value u and a certificate that the objective value is greater than l.

Example: Von Neumann growth problem. Consider an economy with n sectors, each of which has an activity level $x_i > 0$ in the current period and $x_i^+ > 0$ in the next period. The economy consists of m goods. A set of activity levels x consumes $b_i^T x$ and produces $a_i^T x$ of good i. The goods consumed in the next period cannot exceed the goods produced in the current period. In Von Neumman's growth problem, we wish to find a set of activity levels that maximizes the minimum growth rate. This problem can be phrased as the quasiconvex optimization problem

$$\begin{array}{ll} \text{max.} & \min_i x_i^+/x_i \\ \text{s.t.} & Bx^+ \leq Ax \\ & x^+ \geq 0 \end{array}$$

Since this problem is homogenous in x and x^+, we can replace the implicit constraint that $x > 0$ with the explicit constraint $x \geq 1$. To deal with the quasiconcave objective function, we will first instead deal with its negative[4]

$$f(x) = -\min_i x_i^+/x_i = \max_i -x_i^+/x_i.$$

Now, consider that for a fixed t, we have

$$\max_i -x_i^+/x_i \leq t \iff -x_i^+ - x_i t \leq 0 \quad \text{for all } i.$$

Thus, we can take $\phi_t(x^+, x) = \max_i\{-x_i^+ - x_i t\}$, which is convex, as it is the pointwise maximum of a set of affine functions.

Acknowledgements

Thank you to John Drago for helpful edits.

References

[Boy+07] Stephen Boyd et al. "A tutorial on geometric programming". In: *Optimization and engineering* 8.1 (2007), pp. 67–127.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4]In general, note that $p^\star = \text{max. } f_0(x) = -(\text{min. } (-f_0)(x))$.