ENGG 5501: Foundations of Optimization **2016–17 First Term**

Handout 1: Introduction

Instructor: Anthony Man–Cho So September 5, 2016

1 Basic Notions in Optimization

In this course we will consider a class of **mathematical programming** problems that can be expressed in the form:

$$(P) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X. \end{array}$$

Here, $f : \mathbb{R}^n \to \mathbb{R}$ is called the **objective function**, and $X \subset \mathbb{R}^n$ is called the **feasible region**. Thus, $x = (x_1, \ldots, x_n)$ is an n–dimensional vector, and we shall agree that it is represented in *column* form. The entries $x_1, \ldots, x_n$ are called the **decision variables** of (P). As the above formulation suggests, we are interested in a **global minimizer** of (P), which is defined as a point $x^* \in X$ such that $f(x^*) \leq f(x)$ for all $x \in X$. We call $f(x^*)$ the **optimum value** of (P). A related notion is that of a **local minimizer**, which is defined as a point $x' \in X$ such that for some $\epsilon > 0$, we have $f(x') \leq f(x)$ for all $x \in X \cap B(x', \epsilon)$. Here,

$$B(x', \epsilon) = \left\{ x \in \mathbb{R}^n : \|x - x'\|_2 \leq \epsilon \right\}$$

is the **Euclidean ball** of radius $\epsilon > 0$ centered at x' (recall that for $x \in \mathbb{R}^n$, the 2–**norm** of x is defined as $\|x\|_2^2 = \sum_{i=1}^n x_i^2 \equiv x^T x$). Note that a global minimizer is automatically a local minimizer, but the converse is not necessarily true. In this course we shall devote a substantial amount of time to characterize the minimizers of (P) and study how the structures of f and X affect our ability to solve (P). Before we do that, however, let us observe that problem (P) is quite general. For example, when $X = \mathbb{R}^n$, we have an **unconstrained optimization** problem; when X is *discrete* (e.g. $X = \{-1, +1\}^n \subset \mathbb{R}^n$), we have a **discrete optimization** problem. Other important classes of optimization problems include:

- **Linear Programming (LP) Problems:** Here, f is a **linear function**; i.e., a function of the form

  $$f(x) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \equiv c^T x$$

  with $c = (c_1, \ldots, c_n) \in \mathbb{R}^n$, and X is a set defined by **linear inequalities**; i.e., it takes the form

  $$X = \left\{ x \in \mathbb{R}^n : a_i^T x \leq b_i \quad \text{for } i = 1, \ldots, m \right\} \tag{1}$$

  with $a_1, \ldots, a_m \in \mathbb{R}^n$ and $b_1, \ldots, b_m \in \mathbb{R}$. In more compact notation, we may write a linear programming problem as follows:

  $$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b, \end{array}$$

  where A is an $m \times n$ matrix whose i–th row is a_i^T, and $b = (b_1, \ldots, b_m)$ is an m–dimensional column vector (for any two vectors $u, v \in \mathbb{R}^n$, the inequality $u \leq v$ means $u_i \leq v_i$ for $i = 1, \ldots, n$). As we shall see later in the course, LP problems can be solved very efficiently.

- **Quadratic Programming (QP) Problems:** Here, X is as in (1), and f is a quadratic function; i.e., a function of the form

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} x_i x_j \equiv x^T Q x,$$

where $Q = [Q_{ij}]$ is an $n \times n$ matrix. Note that we may assume without loss that Q is symmetric. This follows from the fact that

$$x^T Q x = x^T \left(\frac{Q + Q^T}{2} \right) x.$$

- **Semidefinite Programming (SDP) Problems:** Given an $n \times n$ symmetric matrix Q, we say that Q is **positive semidefinite** (denoted by $Q \succeq \mathbf{0}$ or $Q \in \mathcal{S}_+^n$ if we want to make the dimension explicit) if $x^T Q x \geq 0$ for all $x \in \mathbb{R}^n$. Let $A_1, \ldots, A_m$ and C be $n \times n$ symmetric matrices, and let $b_1, \ldots, b_m \in \mathbb{R}$. Consider the optimization problem

$$\begin{aligned} \text{minimize} \quad & b^T y \\ \text{subject to} \quad & C - \sum_{i=1}^{m} y_i A_i \succeq \mathbf{0}, \\ & y \in \mathbb{R}^m. \end{aligned} \tag{2}$$

Problem (2) is a so–called *semidefinite programming (SDP)* problem. Its feasible region is given by

$$X = \left\{ y \in \mathbb{R}^m : C - \sum_{i=1}^{m} y_i A_i \succeq \mathbf{0} \right\}.$$

It is a routine exercise to show that the optimization problem

$$\begin{aligned} \text{minimize} \quad & C \bullet Z \equiv \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} Z_{ij} \\ \text{subject to} \quad & A_i \bullet Z = b_i \qquad\qquad \text{for } i = 1, \ldots, m, \\ & Z \succeq \mathbf{0} \end{aligned} \tag{3}$$

can be cast into the form (2). Hence, Problem (3) is an instance of SDP. To determine the feasible region of (3), observe that since the matrix Z is symmetric, it is completely determined by, say, the entries on and above the diagonal. Hence, the feasible region of (3) can be expressed as

$$X = \left\{ (Z_{11}, Z_{12}, \ldots, Z_{nn}) \in \mathbb{R}^{n(n+1)/2} : A_i \bullet Z = b_i \text{ for } i = 1, \ldots, m;\ Z \succeq \mathbf{0} \right\}.$$

Similar to LP problems, SDP problems can also be solved efficiently.

- **Polynomial Optimization (PO) Problems:** Here, f is a **real–valued polynomial** of degree, say, d. In other words, it can be expressed as

$$f(x) = \sum_{|\alpha| \leq d} f_\alpha \, x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n},$$

where $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}$, $|\alpha| = \sum_{i=1}^n \alpha_i$, and f_α is the coefficient of the term $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$. The set X is defined by **polynomial inequalities**; i.e., it takes the form

$$X = \{x \in \mathbb{R}^n : g_i(x) \geq 0 \quad \text{for } i = 1, \ldots, m\},$$

where $g_1, \ldots, g_m : \mathbb{R}^n \to \mathbb{R}$ are also real–valued polynomials. In general, PO problems are very difficult to solve. However, under some mild assumptions, they can be efficiently approximated by a series of SDP problems, at least in theory.

The aforementioned classes of problems capture a wide range of applications. However, in order to convert a particular application into a problem of the form (P), we need to first identify the data and decision variables and then formulate the objective function and constraints. Let us now illustrate this process via some examples.

2 Formulating Optimization Problems

2.1 An Air Traffic Control Problem

Suppose that n airplanes are trying to land at the Hong Kong International Airport. Airplane i will arrive at the airport within the time interval $[a_i, b_i]$, where $i = 1, \ldots, n$. For simplicity, we shall assume that the airplanes arrive in the order $1, 2, \ldots, n$. Due to safety concerns, the control tower of the airport would like to maximize the so–called *shortest metering time*; i.e., the minimum over all inter–arrival times between two consecutive airplanes. How then should the airport assign the arrival time of each airplane?

Here, the decision variables are the arrival times of the airplanes, which we denote by $t_1, \ldots, t_n$. Then, we have the following optimization problem:

$$\begin{array}{lll} \text{maximize} & \min_{1 \leq j \leq n-1}(t_{j+1} - t_j) & \\ \text{subject to} & a_i \leq t_i \leq b_i & \text{for } i = 1, \ldots, n, \\ & t_i \leq t_{i+1} & \text{for } i = 1, \ldots, n-1. \end{array} \tag{4}$$

It is not immediately clear that (4) can be formulated as an LP, but it can be done as follows. Let z be a new decision variable. Then, we may rewrite (4) as

$$\begin{array}{lll} \text{maximize} & z & \\ \text{subject to} & t_{i+1} - t_i \geq z & \text{for } i = 1, \ldots, n-1, \\ & a_i \leq t_i \leq b_i & \text{for } i = 1, \ldots, n, \\ & t_i \leq t_{i+1} & \text{for } i = 1, \ldots, n-1, \end{array}$$

which is an LP. We should point out that the above reformulation works only because we are *maximizing* instead of minimizing the quantity $\min_{1 \leq j \leq n-1}(t_{j+1} - t_j)$. In particular, the following problems:

$$\begin{array}{lll} \text{minimize} & \min_{1 \leq j \leq n-1}(t_{j+1} - t_j) & \\ \text{subject to} & a_i \leq t_i \leq b_i & \text{for } i = 1, \ldots, n, \\ & t_i \leq t_{i+1} & \text{for } i = 1, \ldots, n-1 \end{array} \tag{5}$$

and

$$\begin{array}{lll} \text{minimize} & z & \\ \text{subject to} & t_{i+1} - t_i \geq z & \text{for } i = 1, \ldots, n-1, \\ & a_i \leq t_i \leq b_i & \text{for } i = 1, \ldots, n, \\ & t_i \leq t_{i+1} & \text{for } i = 1, \ldots, n-1 \end{array} \tag{6}$$

are *not* equivalent, since the optimum value of (5) is finite (in fact, it is always non–negative), while the optimum value of (6) is $-\infty$.

For another air traffic control application that utilizes optimization techniques, see [1].

2.2 A Data Fitting Problem

The previous example shows that sometimes one may be able to convert an optimization problem into an LP via some transformations. Here is another illustration of such possibility. Suppose that we are given m data pairs (a_i, b_i), where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i = 1, \ldots, m$, with $m \geq n + 1$. We suspect that these pairs are essentially generated by an **affine function**; i.e., a function $f : \mathbb{R}^n \to \mathbb{R}$ of the form $f(y) = x^T y + t$ for some $x \in \mathbb{R}^n$ and $t \in \mathbb{R}$. However, the output of the function is usually corrupted by an additive noise. Thus, the relationship between a_i and b_i is better described as $b_i = a_i^T x + t + \epsilon_i$, where $x \in \mathbb{R}^n$ and $t \in \mathbb{R}$ are the parameters of the affine function, and $\epsilon_i \in \mathbb{R}$ is the noise in the i–th measurement. Our goal then is to determine the parameters $x \in \mathbb{R}^n$ and $t \in \mathbb{R}$ of the affine function that best fits the data. To measure the goodness of fit, we can try to minimize some sort of error measure. One popular measure is the 1–*norm* of the *residual errors*, which is defined as

$$\Delta_1 = \sum_{i=1}^{m} \left| b_i - a_i^T x - t \right| = \|b - Ax - te\|_1,$$

where A is the $m \times n$ matrix whose i–th row is a_i^T, and $e \in \mathbb{R}^m$ is the vector of all ones. In other words, our optimization problem is simply

$$\min_{x \in \mathbb{R}^n,\, t \in \mathbb{R}} \quad \sum_{i=1}^{m} \left| b_i - a_i^T x - t \right|. \tag{7}$$

Here, the objective function is nonlinear. However, we can turn problem (7) into an LP as follows. We first introduce m new decision variables $z_1, \ldots, z_m \in \mathbb{R}$. Then, it is not hard to see that (7) is equivalent to the following LP:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} z_i \\ \text{subject to} \quad & b_i - a_i^T x - t \leq z_i \qquad \text{for } i = 1, \ldots, m, \\ & -b_i + a_i^T x + t \leq z_i \quad \text{for } i = 1, \ldots, m. \end{aligned}$$

Now, what if we want to minimize the 2–norm of the residual errors? In other words, we would like to solve the following problem:

$$\min_{x \in \mathbb{R}^n,\, t \in \mathbb{R}} \quad \Delta_2 = \|b - Ax - te\|_2^2 = \sum_{i=1}^{m} \left(b_i - a_i^T x - t \right)^2. \tag{8}$$

It turns out that this is a particularly simple QP. In fact, since (8) is an unconstrained optimization problem with a *differentiable* objective function, we can solve it using calculus techniques. Indeed, suppose for simplicity that $\bar{A}$ has full column rank, so that $\bar{A}^T \bar{A}$ is invertible. Then, the (unique) optimal solution $(x^*, t^*) \in \mathbb{R}^n \times \mathbb{R}$ to (8) is given by

$$\begin{bmatrix} x^* \\ t^* \end{bmatrix} = \left(\bar{A}^T \bar{A} \right)^{-1} \bar{A}^T b \qquad \text{with } \bar{A} = \begin{bmatrix} a_1^T & 1 \\ \vdots & \vdots \\ a_m^T & 1 \end{bmatrix} \in \mathbb{R}^{m \times (n+1)}.$$

On the other hand, if $\bar{A}$ does not have full column rank, then it can be shown that for any $z \in \mathbb{R}^{n+1}$, the vector

$$\begin{bmatrix} x^* \\ t^* \end{bmatrix} = \left(\bar{A}^T \bar{A}\right)^\dagger \bar{A}^T b + \left(I - (\bar{A}^T \bar{A})^\dagger \bar{A}^T \bar{A}\right) z$$

is optimal for (8). It is worth noting that the matrix $I - (\bar{A}^T \bar{A})^\dagger \bar{A}^T \bar{A}$ is simply the orthogonal projection onto the nullspace of $\bar{A}^T \bar{A}$. In particular, when $\bar{A}$ does not have full column rank, the nullspace of $\bar{A}^T \bar{A}$ is non–trivial.

In the above discussion, we assume that the number of observations m exceeds the number of parameters n; specifically, $m \geq n + 1$. However, in many modern applications (such as biomedical imaging and gene expression analyses), the number of observations is much smaller than the number of parameters. Thus, one can typically find infinitely many parameter pairs $(\bar{x}, \bar{t}) \in \mathbb{R}^n \times \mathbb{R}$ that fit the data perfectly; i.e., $b_i = a_i^T \bar{x} + \bar{t}$ for $i = 1, \ldots, m$. To make the data fitting problem meaningful, it is then necessary to impose additional assumptions. An intuitive and popular one is that the actual number of parameters responsible for the input–output relationship is small. In other words, most of the entries in the parameter vector $x \in \mathbb{R}^n$ should be zero, though we do not know a priori where those entries are. There are several ways to formulate the data fitting problem under such an assumption. For instance, one can consider the following constrained optimization approach:

$$\begin{array}{ll} \text{minimize} & \|b - Ax - te\|_2^2 \\ \text{subject to} & \|x\|_0 \leq K, \\ & x \in \mathbb{R}^n,\, t \in \mathbb{R}. \end{array} \tag{9}$$

Here, $\|x\|_0$ is the number of non–zero entries in the parameter vector $x \in \mathbb{R}^n$, and $K \geq 0$ is a user–defined threshold that controls the sparsity of x. Alternatively, one can consider the following penalty approach:

$$\min_{x \in \mathbb{R}^n,\, t \in \mathbb{R}} \left\{ \|b - Ax - te\|_2^2 + \mu \|x\|_0 \right\}, \tag{10}$$

where $\mu > 0$ is a penalty parameter. However, due to the combinatorial nature of the function $x \mapsto \|x\|_0$, both of the above formulations are computationally difficult to solve. In fact, it can be shown, in a formal sense, that an efficient algorithm for solving problems (9) and (10) is unlikely to exist. To obtain more tractable formulations, a widely used approach is to replace $\|\cdot\|_0$ by $\|\cdot\|_1$. We will see later in the course why this is a good idea from a computationally perspective. For now, we should note that such an approach changes the original problems, and a natural question is whether there is any correspondence between the solutions to the original problems and those to the modified problems. This question has been extensively studied in the fields of high–dimensional statistics and compressive sensing over the past decade or so. We refer the interested reader to the book [2] for details and further pointers to the literature.

At this point let us reflect a bit on the above examples. Intuitively, a linear problem (say, an LP) should be easier than a nonlinear problem, and a differentiable problem should be easier than a non–differentiable one. However, the above examples show that these need not be the case. Indeed, even though the 2–norm problem (8) is a QP, its optimal solution has a nice characterization, while the corresponding 1–norm problem (7) does not have such a feature. On the other hand, even though the objective function in (7) is non–differentiable, the problem can still be solved easily via LP. Also, we have seen from problems (9) and (10) that the inclusion of a seemingly simple constraint or objective may render an originally easy optimization problem (namely, Problem (8)) intractable.

From the above discussion, it is natural to ask what makes an optimization problem difficult. While it is hard to give an answer to such question without over–generalizing, let us at least identify a possible source of difficulty. What distinguishes the seemingly very similar problems (8) and (9) is that the former is a so–called **convex optimization problem**, while the latter is not. We shall define the notion of convexity and study it in more detail later.

2.3 Eigenvalue Optimization

Suppose that we are given k $n \times n$ symmetric matrices $A_1, \ldots, A_k$. Consider the function $A : \mathbb{R}^k \to \mathbb{R}^{n \times n}$ defined by

$$A(x) = \sum_{i=1}^{k} x_i A_i.$$

Note that by definition, the matrix $A(x)$ is symmetric for any $x \in \mathbb{R}^k$. Now, a problem that is frequently encountered in practice is that of choosing an $x \in \mathbb{R}^k$ so that the largest eigenvalue of $A(x)$ is minimized (see, e.g., [3] for details). It turns out that such a problem can be formulated as an SDP. To prove this, we need the following result:

Proposition 1 *Let A be an arbitrary $n \times n$ symmetric matrix, and let $\lambda_{\max}(A)$ denote the largest eigenvalue of A. Then, we have $tI \succeq A$ if and only if $t \geq \lambda_{\max}(A)$.*

Proof Suppose that $tI \succeq A$, or equivalently, $tI - A \succeq \mathbf{0}$. Then, for any $u \in \mathbb{R}^n \backslash \{\mathbf{0}\}$, we have $u^T(tI - A)u = tu^T u - u^T A u \geq 0$, or equivalently,

$$t \geq \frac{u^T A u}{u^T u}.$$

Since this holds for an arbitrary $u \in \mathbb{R}^n \backslash \{\mathbf{0}\}$, we have

$$t \geq \max_{u \in \mathbb{R}^n \backslash \{\mathbf{0}\}} \frac{u^T A u}{u^T u}. \tag{11}$$

By the Courant–Fischer theorem, the right–hand side of (11) is precisely $\lambda_{\max}(A)$.

The converse can be established by reversing the above arguments. This completes the proof. □

Proposition 1 allows us to formulate the above eigenvalue optimization problem as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & tI - A(x) \succeq \mathbf{0}, \end{array} \tag{12}$$

which is of the form (2). Hence, problem (12) is an SDP.

References

[1] D. Bertsimas, M. Frankovich, and A. Odoni. Optimal Selection of Airport Runway Configurations. *Operations Research*, 59(6):1407–1419, 2011.

[2] P. Bühlmann and S. van de Geer. *Statistics for High–Dimensional Data: Methods, Theory and Applications*. Springer Series in Statistics. Springer–Verlag, Berlin/Heidelberg, 2011.

[3] A. S. Lewis and M. L. Overton. Eigenvalue Optimization. *Acta Numerica*, 5:149–190, 1996.