ECE 490: Introduction to Optimization **Fall 2018**

Lecture 17

Overview of Constrained Optimization

Lecturer: Bin Hu, Date:10/30/2018

Now we shift our focus to constrained optimization. In this lecture, we give an overview of constrained optimization. We will clarify what type of constrained optimization problems will be studied in the rest of the semester.

17.1 General Formulation of Constrained Optimization

Recall that in general an optimization problem has the following form:

$$\min_{x \in X} f(x)$$

where x is the decision variable, f is the objective function, and X is some feasible set. In most situations, the feasible set X can be decoded by some equality and inequality constraints. This leads to the following general formulation for constrained optimization:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \ i = 1, \ldots, m \\ & h_j(x) = 0, \ j = 1, \ldots, l \end{aligned} \tag{17.1}$$

We have inequality constraints in the form of $g_i(x) \leq 0$ and equality constraints in the form of $h_j(x) = 0$. The optimal value of the above problem is defined as $f^* = \inf\{f(x) | g_i(x) \leq 0, \ i = 1, \ldots, m, \ h_j(x) = 0, \ j = 1, \ldots, l\}$. If there does not exist any x satisfying $g_i(x) \leq 0$ and $h_j(x) = 0$ for all (i, j), we have $f^* = \infty$. In this case, we say the problem (17.1) is infeasible. If the problem (17.1) is unbounded below, we have $f^* = -\infty$. For example, consider the following problem

$$\begin{aligned} \text{minimize} \quad & x + y \\ \text{subject to} \quad & x^2 + y^2 \leq 1 \\ & x + y = 100 \end{aligned}$$

We cannot find any real number pair (x, y) to satisfy $x^2 + y^2 \leq 1$ and $x + y = 100$ simultaneously. For this case we have $f^* = \infty$.

Let's look at another example. We consider the following simple problem

$$\begin{aligned} \text{minimize} \quad & x \\ \text{subject to} \quad & x \leq 1 \end{aligned}$$

In this case, the problem is unbounded below. Hence we have $f^* = -\infty$.

Clearly, the forms of g_i and h_j will affect whether we can efficiently solve (17.1) or not. Depending on the properties of g_i and h_j, the problem (17.1) can become very challenging. In this lecture, we will briefly discuss what types of g_i and h_j will be covered in this course.

17.2 Feasibility Problem

The so-called feasibility problem can be viewed as a special case of the general constrained optimization problem. Specifically, the feasibility problem has the following form:

$$\begin{aligned} \text{find} \quad & x \\ \text{subject to} \quad & g_i(x) \leq 0, \ i = 1, \ldots, m \\ & h_j(x) = 0, \ j = 1, \ldots, l \end{aligned} \tag{17.2}$$

It asks whether we can find a point x such that the constraints $g_i(x) \leq 0$ and $h_j(x) = 0$ are satisfied for all (i, j). This can be considered as a special case of (17.1) where f is set to be any constant function. For example, (17.2) can be reformulated as

$$\begin{aligned} \text{minimize} \quad & 0 \\ \text{subject to} \quad & g_i(x) \leq 0, \ i = 1, \ldots, m \\ & h_j(x) = 0, \ j = 1, \ldots, l \end{aligned} \tag{17.3}$$

By the above formulation, we have $f^* = 0$ if the original problem (17.2) is feasible, and $f^* = \infty$ if (17.2) is infeasible.

17.3 Overview of Optimization with Equality Constraints

General nonlinear equality constraints can be very difficult to handle. For general h_j, the feasibility problem can already be extreme difficult since it involves solving nonlinear equations. In this course, we mostly focus on linear equality constraints in the form of $Ax = b$ where A is some matrix and b is some vector. Notice linear constraints and affine constrains are typically exchangeable in the optimization literature. To be clear, we will talk about the following problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & Ax - b = 0 \end{aligned} \tag{17.4}$$

The above problem can be more or less viewed as an unconstrained optimization problem since we can use the linear equation $Ax = b$ to eliminate some variables in $f(x)$ and then the resultant problem becomes an unconstrained optimization. However, sometimes we still want to treat (17.4) as a constrained optimization problem. Here is one example that we prefer to reformulate unconstrained optimization problems as constrained optimization problems in the form of (17.4). Consider a composite optimization problem: $\min f(x) + g(Tx)$ where f is differentiable, g is the ℓ_1-norm, and T is some structure matrix. This problem arises from many applications, e.g. sensor/actuator allocation in control. The matrix T captures some important topology structure in such applications. If $T = I$, the problem can be handled by the proximal gradient method. For general T, one way to handle the composition structure is to set $z = Tx$ and reformulate the problem as

$$\begin{aligned} \text{minimize} \quad & f(x) + g(z) \\ \text{subject to} \quad & Tx - z = 0 \end{aligned}$$

which can be solved using primal-dual type of methods.

There are a large family of primal-dual methods that are designed to solve constrained optimization problems in the form of (17.4). We will talk about these algorithms in future lectures.

For more complicated h_j, one heuristic is to linearize h_j at each x_k for all k and iteratively solve a linear constrained optimization problem. We will briefly discuss this in the future lectures.

17.4 Overview of General Constrained Optimization

For a general problem with both equality and inequality constraints, we will first talk about the optimality conditions (the famous Karush-Kuhn-Tucker (KKT) conditions). We will also briefly talk about some general techniques (e.g. sequential quadratic programming) that one can try for general problems.

We will cover convex programming in more details. In this case, we have linear equality constraints and g_i is convex for all i. Notice any equality constraint $h_j(x) = 0$ can be equivalently rewritten as two inequality constraints $h_j(x) \leq 0$ and $h_j(x) \geq 0$. Therefore, in convex programming, we expect h_j and $-h_j$ are both convex. This naturally leads to the linear equality constraints. We will talk about penalty methods, barrier functions, and multiplier methods for such problems.

There are several types of convex programming problems that have been extensively studied in the literature.

- Linear programming (LP):

$$\begin{aligned} \text{minimize} \quad & c^\mathsf{T} x \\ \text{subject to} \quad & Gx - r \leq 0 \\ & Ax - b = 0 \end{aligned} \tag{17.5}$$

  where $Gx - r \leq 0$ means all the entries of the vector $(Gx - r)$ are non-positive. Many problems can be reformulated as LP problems. For example, the piecewiselinear minimization problem minimize $max_{i=1,\ldots,m}(a_i^\top x + b_i)$ can be reformulated as

$$\begin{aligned} &\text{minimize} \quad t \\ &\text{subject to} \quad a_i^\top x + b_i \le t, \; i = 1, \ldots, m \end{aligned} \tag{17.6}$$

which is in the form of (17.5) if we augment x and t as our new decision variable vector and choose (c, G, r) as

$$c = \begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix}, \; G = \begin{bmatrix} a_1^\top & -1 \\ a_2^\top & -1 \\ \vdots & -1 \\ a_m^\top & -1 \end{bmatrix}, \; r = \begin{bmatrix} -b_1 \\ -b_2 \\ \vdots \\ -b_m \end{bmatrix}.$$

Many solvers are available for LPs. Both the simplex method and the interior-point methods have been implemented for practical large scale problems. In Matlab, you can use the function linprog to solve LPs.

- Second-order cone programming (SOCP):

$$\begin{aligned} &\text{minimize} \quad c^\top x \\ &\text{subject to} \quad \|F_i x + d_i\| \le f_i^\top x + r_i, \; i = 1, \ldots, m \\ &\qquad\qquad\quad Ax - b = 0 \end{aligned} \tag{17.7}$$

SOCP is more general than LP but also slightly more difficult. One example for SOCP is the so-called robust LP problem. There are also many solvers for SOCP.

- Semidefinite programming (SDP):

$$\begin{aligned} &\text{minimize} \quad c^\top x \\ &\text{subject to} \quad x_1 F_1 + x_2 F_2 + \cdots + x_p F_p + F_0 \le 0 \\ &\qquad\qquad\quad Ax - b = 0 \end{aligned} \tag{17.8}$$

where $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$ is the decision variable vector, F_i are symmetric matrices for all i, and the matrix inequality holds in the semidefinite sense. Hence we want to find x such that $x_1 F_1 + x_2 F_2 + \cdots + x_p F_p + F_0$ is a negative semidefinite matrix. SDP is very powerful. Recall the following general condition we used to construct dissipation inequality:

$$\begin{bmatrix} A^\top PA - \rho^2 P & A^\top PB \\ B^\top PA & B^\top PB \end{bmatrix} - \sum_{j=1}^{j} \lambda_j X_j \le 0 \tag{17.9}$$

Given (A, B, X_j, ρ^2), we can augment P and λ_j as our decision variable vector and rewrite the above condition in the form of (17.8). So basically the dissipation inequality approach relies on solving feasibility problems of SDPs. Numerical solvers are also available for SDPs. For example, we can download the cvx package and then write Matlab codes to solve SDPs.