OPTIMIZATION
METHODS

Course for postgraduates of
the Department of Electrical and Computer Engineering

1996

Minyue Fu & Gjerrit Meinsma

Contents

List of Figures

List of Tables

1

Introduction

1.1 Examples of Optimization Problems

The thoery of optimization finds wide applications in many scientific and engineering disciplines. Our daily activities are confronted with all sorts of optimization problems. Although not every optimization problem can be formulated mathematically (questions like "what is the most efficient way to study the optimization theory?"), many do. It is interesting that many seemingly non-optimization problems can be formulated as optimization problems and sometimes solved effectively using optimization techniques. In the following, we state a few simple examples of such kind.

Example 1.1.1. Finding a root. Given a function $f : [0,\ 1] \to \mathbb{R}$. The problem is to find a root x of $f(\cdot)$, i.e. $f(x) = 0$.

This problem is equivalent to the following optimization problem:

$$\begin{array}{ll} \text{minimize} & f^2(x) \\ \text{subject to} & x \in [0,\ 1] \end{array} \tag{1.1}$$

Obviously, a root is found if and only if the minimum is equal to zero.

Example 1.1.2. Shape of a String. Consider a string with a uniform mass density as illustrated in Figure 1.1, the heights of the two ends are a and b, respectively. Assume that the length L of the string exceeds $\sqrt{(a-b)^2+1}$. The problem is to find the shape of the string when it is at rest.

Using the minimal potential energy principle in Newtonian Physics, we know that the shape of the string must be smooth and is such that the potential energy of the string is minimum. Denote the shape of the string by $f(x) : x \in [0,\ 1]$. Given an infinitesimal segment dx at x, the length of the corresponding string segment is $\sqrt{1+\dot{f}^2(x)}dx$. The potential energy of this segment is proportional to $f(x)\sqrt{1+\dot{f}^2(x)}dx$. Hence, the potential energy of the whole string is given by

$$\int_0^1 f(x)\sqrt{1+\dot{f}^2(x)}dx$$

Similarly, the length of the string is calculated to be

$$L = \int_0^1 \sqrt{1+\dot{f}^2(x)}dx$$

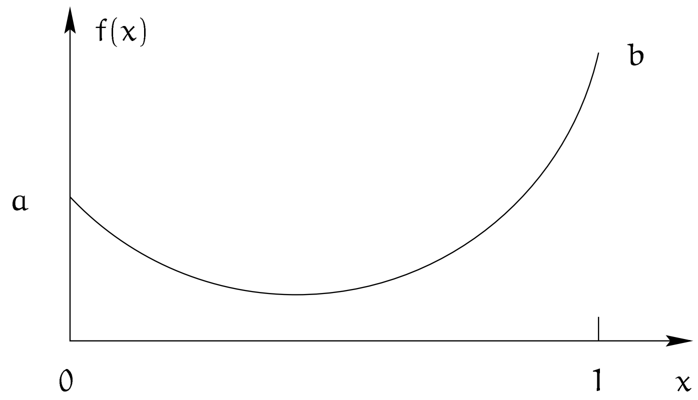

Figure 1.1: The String Problem

Hence, the shape of the string is the solution to the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \int_0^1 f(x)\sqrt{1+\dot{f}^2(x)}dx \\ \text{subject to} & L = \int_0^1 \sqrt{1+\dot{f}^2(x)}dx \\ & f(0)=a, \quad f(1)=b \end{array} \tag{1.2}$$

Example 1.1.3. Fermat's Last Theorem. Fermat's last theorem states that no integers $x, y, z \geq 1$ and $n \geq 3$ exist such that

$$x^n + y^n = z^n$$

This statement can be formulated in terms of the following optimization problem:

$$\begin{array}{ll} \text{minimize} & (x^n + y^n - z^n)^2 + \sin^2 \pi n + \sin^2 \pi x + \sin^2 \pi z \\ \text{subject to} & x \geq 1, \; y \geq 1, \; z \geq 1, \; n \geq 3 \end{array} \tag{1.3}$$

The minimum equals to 0 if and only if Fermat's last theorem is false. This is easy to see because when the minimum is 0, x, y, z and n are forced to be integers due to the sin terms, and $x^n + y^n = z^n$.

1.2 Classification of Optimization Problems

The general optimization problem can be stated as follows: Given a *feasible set* D and an *objective function* $f : D \to \mathbb{R}$, find

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in D \end{array} \tag{1.4}$$

If $D \subset \mathbb{R}^n$, the optimization problem is *finite dimensional.* If D is a subset in an infinite dimensional space (such as Banach space), the problem is infinite dimensional. Examples 1.1.1 and 1.1.3 are finite dimensional, whereas Example 1.1.2 is infinite-dimensional. In this course, we will consider finite dimensional optimization problems only.

We classify the (finite dimensional) optimization problems as follows:

Unconstrained Optimization Problems In this case, $D = \mathbb{R}^n$.

Constrained Optimization Problems In this case, D is a subset in $\mathbb{R}^n$ which is often described by a number of inequalities. A general description of the feasible set is given by

$$g_i(x) \leq 0, \quad i = 1, 2, \cdots, m \tag{1.5}$$

where $g_i : \mathbb{R}^n \to \mathbb{R}$ are given functions. Sometimes, matrix inequalities are used instead of scalar inequlaities. For example, D may be a set in which a symmetric matrix function $G(x)$ is positive definite, i.e.,

$$D = \{x : G(x) > 0, x \in \mathbb{R}^n\}$$

However, we note that every matrix inequality can be expressed as a number of scalar inequalities.

Decision Problems Associated with every optimization problem in (1.4), there is a decision problem which is stated as follows: Given a feasible set $D \subset \mathbb{R}^n$, $f : D \to \mathbb{R}$ and any scalar α, determine if

$$f(x) < \alpha, \quad \forall\, x \in D \tag{1.6}$$

The answer to the decision problem is binary: *yes* or *no*. The decision problem appears to be simpler than the optimization problem. However, as we will show in Chapter 4 that an efficient algorithm for the decision problem will automatically lead to an efficient algorithm for solving the optimization problem. Further, decision problems play an important role in computational complexity analysis (see Chapter 4).

Feasibility Problems In the above, we have implicitly assumed that the feasible set is nonempty. Although this assumption is often valid (see Examples 1.1.1-1.1.3), determining if the feasible set is empty or not is often nontrivial when the set is described by inequalities (1.5). The task of the feasibility problem is to answer this question, and in the affirmative case, to find a *feasible member* $x \in D$. In fact, the decision problem mentioned above can be reformulated as a feasibility problem: determine if the following augmented feasible set is empty:

$$\hat{D} = \{x : f(x) \geq \alpha, x \in D\} \tag{1.7}$$

Obviously, $\hat{D}$ is empty if and only if the answer to the decision problem is *yes*. So in this sense, feasibility problems are very general.

The constrained optimization problems can be further categorized as follows:

Linear Programs (LP) In this case, $f(x)$ is linear and D is described by a set of linear inequalities, i.e, given $c \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$, solve

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \geq b \end{array} \tag{1.8}$$

The inequality above is taken to be elementwise.

LP problems have been intensively studied for decades. Efficient algorithms exist for solving them.

Quadratic Programs (QP) This case is the same as LP except that $f(x)$ is a quadratic function, i.e.,

$$f(x) = x^T Q x + 2c^T x \tag{1.9}$$

for some vector $c \in \mathbb{R}^n$ and symmetric matrix $Q \in \mathbb{R}^{n\times n}$. It turns out that the QP problem is much harder than the LP problem. In fact, the QP problem is one of the most challenging problems facing the researchers in the optimization theory.

Semidefinite Programs (SDP) In this case, $f(x)$ is linear and D is described by a *linear matrix inequality*. That is, given a vector $c \in \mathbb{R}^n$ and symmetric matrices $F_i \in \mathbb{R}^{m\times n}, i = 0, 1, \cdots, n$, solve

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F_0 + \sum_{i=1}^n x_i F_i \geq 0 \end{array} \tag{1.10}$$

It turns out that the SDP probelm include the LP problem as a special case (see Chapter 6). The research in the last decade has offered efficient algorithms for solving the SDP problem.

Convex Programs In this case, $f(x)$ is a convex function and D is a convex set (see Section 2.4 for definitions of convexity). Every SDP is a convex program. The QP problem is convex when Q is positive definite (see Section 2.4).

Nonlinear Programs Either $f(x)$ is nonlinear or D is described by nonlinear inequlaities. Note that SDP probems are in general nonlinear programming problems. The QP problem is also nonlinear.

Nonconvex Programs Either $f(x)$ or D is nonconvex.

1.3 The Scope of the Course

The rest of the course include 7 chapters. The details are as follows:

Optimality Conditions This chapter introduces simple first order and second order conditions for local minima. In particular, the so-called KKT conditions, which are first order necessary conditions for local minima under some mild conditions, will be discussed in details. Also introduced in this chapter are convex programs and their important optimality features.

Basic Optimization Techniques Three classical optimization techniques will be introduced in this chapter, namely, Newton's method, quasi-Newton methods and conjugate gradient methods. These methods are popularly used for general optimization problems. The material in this chapter will serve as an prerequisite to interior point algorithms for linear programming and semidefinite programming.

Computational Complexity This chapter studies the computational complexity of optimization problems. The concept of NP-completeness will be discussed which plays an vital role in determining whether an optimization problem is inherently intractable or not, i.e., whether or not the problem is expected to be unsolvable by an algorithm with polynomial complexity. Polynomial complexity is an important feature an efficient algorithm must have when the problem size becomes large. A list of classical NP-complete problems will be given, along with many examples in Systems and Control.

Interior Point Methods for Convex Programs In this chapter, we cover two general methods for solving convex programs. These are the logarithmic barrier method and the method of centers. These methods are conceptually easy and guaranteed to converge. For special convex programs such as linear programs and semidefinite programs, we will show in the next two chapters that with minor alternations, the convergence of these algorithms takes place in polynomial time.

Linear Programming This chapter introduces two algorithms for solving the well-known and perhaps the most important optimization problem: linear programs. The first one is the classical Simplex algorithm. Detailed analysis will be given. The second one is the newly developed interior point algorithm which has polynomial complexity. By combining the two algorithms, very efficient algorithms can be developed to solve large-size linear programs.

Semidefinite Programming A much more convex optimization problem, the semidefinite programming (SDP) problem, will be studied in this chapter. SDP problems arise very often in many engineering discplines including statistics, systems and control, signal processing, etc. Techniques useful in generating SDP problems will be discussed. Many SDP examples will be given. The celebrated interior point method will be introduced.

Nonconvex Optimization Techniques The main purpose of this chapter is to present a list of commonly used ad-hoc optimization techniques which are often effective in solving difficult optimization problems. These techniques include gradient descent method, multiplier penalty function method, simulated annealing method, homotopy method, alternative projection method, Monte-Carlo test and other randomized algorithms, etc.

1.4 Notes and References

Example 1.1.3 comes from (Murty 1988), also see (Vavasis 1991).