ISEN 629: Engineering Optimization

Lecture 1

Sergiy Butenko

Industrial and Systems Engineering
Texas A& M University

Fall 2007

Introduction

In many applications, one deals with optimization problems in the form

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

where

- $X \subseteq \Re^n$;
- $f(x) : X \to \Re$ is a continuous function of n variables.

The major general areas of mathematical programming:

1. Modeling
2. Optimization theory
3. Numerical methods
4. Implementation and application

Introduction

- Conjecture: In general, optimization problems are **unsolvable**.
- In modeling a practical problem it is important to clearly understand the properties of the model developed and the computational consequences of selecting a particular model.
- Often we are forced to choose between a "good" (realistic) model, which we cannot solve, and a "bad" (rough) model, which can be solved efficiently.

As far as the laws of mathematics refer to reality, they are not certain, and as far as they are certain, they do not refer to reality.
-Albert Einstein

Introduction

- The most widespread optimization models are linear models.
- Practitioners prefer to deal with solvable models!
- A linear approximation may be poor, but consequences of choosing the linear model are often predictable.
- In some cases, choosing a "rough" model may be a better strategy than trying to solve a model without any guarantee for success.
- To design a right model, it is important to know the generic classes of optimization problems that are "efficiently solvable".
- Convex optimization problems represent a wide class of problems that allow efficient numerical solutions.

Nonlinear Optimization: A General Formulation

We will deal with different variants of the following problem:

$$\begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & f_j(x) \le \{\ge, =\} 0, \;\; j = 1, \ldots, m, \\ & x \in S, \end{array} \tag{1}$$

where

- $x = (x^{(1)}, x^{(2)}, \ldots, x^{(n)})^T \in \mathbb{R}^n$ is an n-dimensional real vector;
- S - a subset of $\mathbb{R}^n$;
- $f_0(x), \ldots, f_m(x) : \mathbb{R}^n \to \mathbb{R}$ - some real-valued functions of x.
- $f_0(x)$ is called the objective function of our problem;
- the vector function $f(x) = (f_1(x), \ldots, f_m(x))^T$ is called the vector of functional constraints;
- the set S is called the basic feasible set of our problem.
- the set $Q = \{x \in S : f_j(x) \le 0, j = 1, \ldots, m\}$ is called the feasible set.

Nonlinear Optimization: A General Formulation

Minimization problems are naturally classified as follows:

- Unconstrained problems: $Q \equiv \mathbb{R}^n$;
- Constrained problems: $Q \subset \mathbb{R}^n$;
- Smooth problems: all $f_j(x)$ are differentiable;
- Nonsmooth problems: there is a nondifferentiable $f_k(x)$;
- Linearly constrained problems: all functional constraints are linear and S is a polyhedron;
- Linear programming problems: linearly constrained problems with linear objective;
- Quadratic programming problems: linearly constrained problems with quadratic objective;
- Quadratically constrained quadratic problems: all f_j are quadratic.

Nonlinear Optimization: A General Formulation

Classification based on the properties of feasible set:

- Our problem is called feasible if $Q \neq \emptyset$.
- The problem is called strictly feasible if there exists $x \in Q$ such that $f_j(x) < 0$ (or > 0) for all inequality constraints and $f_j(x) = 0$ for all equality constraints [Slater condition].

We also distinguish different types of solution to our problem:

- $x^* \in Q$ is called a global optimal solution to (1) if $f_0(x^*) \le f_0(x)$ for all $x \in Q$. In this case $f_0(x^*)$ is called the (global) optimal value of the problem.
- $x^* \in Q$ is called a local optimal solution to (1) if $f_0(x^*) \le f_0(x)$ for all $x \in B_\epsilon(x^*) \cap Q$ for some $\epsilon > 0$, where $B_\epsilon(x^*) = \{x : \|x - x^*\| < \epsilon\}$.

Performance of numerical methods

- Imagine that we are given a problem $\mathcal{P}$, and we need to select a numerical method that is the best for solving $\mathcal{P}$.
- It may be incorrect to talk about the best method for a particular problem (instance). Indeed, if we consider a method that just outputs $x^* = 0$, it is unbeatable for an instance with the solution $x^* = 0$.
- Therefore, we will talk about the performance of a method with respect to a class of problems $\mathcal{F} \ni \mathcal{P}$. Indeed, usually the numerical methods are developed for solving many different problems with similar characteristics.
- Thus, a performance of method $\mathcal{M}$ on the whole class $\mathcal{F}$ is a natural characteristic of its efficiency.
- We assume that $\mathcal{M}$ does not have complete information about a particular problem $\mathcal{P}$, but only some attributes are known.

Performance of numerical methods

- A set of attributes known for a numerical scheme is called the model of the problem.
- We denote the model by Σ.
- Usually the model consists of problem formulation, description of classes of functional components, etc.
- In order to recognize the problem $\mathcal{P}$, the method should be able to collect specific information about $\mathcal{P}$.
- The process of collecting the data can be conveniently described by the notion of an oracle.
- An oracle $\mathcal{O}$ is a unit, which answers the successive questions of the method $\mathcal{M}$, which is trying to solve the problem by collecting and handling the answers.

Performance of numerical methods

- In general, each problem can be described by different models, and for each problem we can develop different types of oracles.
- But if we fix Σ and $\mathcal{O}$, it is natural to define the performance of $\mathcal{M}$ on $(\Sigma, \mathcal{O})$ as its performance on the worst $\mathcal{P}_w$ from $(\Sigma, \mathcal{O})$.
- Performance of $\mathcal{M}$ on $\mathcal{P}$ is the total amount of computational efforts required by method $\mathcal{M}$ to solve the problem $\mathcal{P}$.
- In some cases solving the problem means finding an exact solution, however it is impossible for general optimization problems. Therefore, by solving the problem $\mathcal{P}$ we will mean finding an approximate solution to $\mathcal{P}$ with an accuracy $\epsilon > 0$.
- The exact meaning of "accuracy $\epsilon > 0$" will be discussed later. For now, we will just use notation $\mathcal{T}_\epsilon$ for a stopping criterion.
- Then the problem class $\mathcal{F}$ can be formally defined as

$$\mathcal{F} \equiv (\Sigma, \mathcal{O}, \mathcal{T}_\epsilon).$$

General Iterative Scheme

Input: A starting point x_0 and an accuracy $\epsilon > 0$.
Initialization: Set $k = 0, I_{-1} = \emptyset$. [Here k is iteration counter and I_k is accumulated information set].

Main loop:

1. Call oracle $\mathcal{O}$ at x_k.
2. Update the information set: $I_k = I_{k-1} \cup (x_k, \mathcal{O}(x_k))$.
3. Apply rules of method $\mathcal{M}$ to I_k and form point x_{k+1}.
4. If stopping criterion $\mathcal{T}_\epsilon$ is satisfied then form output $\bar{x}$. Otherwise, set $k := k + 1$ and go to step 1.

Figure: General iterative scheme for solving problem $\mathcal{P}$ using method $\mathcal{M}$.

Computational Efforts

We can now formalize the definition of *computational efforts* as the description of performance. Two most expensive steps in the above scheme are

- Step 1, where we call the oracle;
- Step 3, where we form the next test point.

Thus, we can introduce two measures of complexity of method $\mathcal{M}$ for problem $\mathcal{P}$:

- *Analytical complexity:* The number of calls of oracle required to solve problem $\mathcal{P}$ up to accuracy ϵ.
- *Arithmetical complexity:* The total number of arithmetic operations (including the work of oracle and the work of method) required to solve problem $\mathcal{P}$ up to accuracy ϵ.

The local black box

We will use the following standard assumption on the oracle called the *local black box concept*:

1. The only information available for the numerical scheme is the answer of the oracle.
2. The oracle is *local*: A small variation of the problem instance far enough from the test point x does not change the answer at x.

The standard formulation (1) is called a *functional model* of optimization problems. Depending on the degree of smoothness, we can apply different types of oracle:

- Zero-order oracle: returns the value $f(x)$.
- First-order oracle: returns $f(x)$ and the gradient $f'(x)$.
- Second-order oracle: returns $f(x), f'(x)$ and the Hessian $f''(x)$.

Complexity Bounds for Global Optimization

Consider the problem

$$\min_{x \in B_n} f(x), \tag{2}$$

where

$$B_n = \{x \in \mathbb{R}^n : 0 \leq x^{(i)} \leq 1, i = 1, \ldots, n\}.$$

Let us measure distances in $\mathbb{R}^n$ using l_∞-norm:

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x^{(i)}|.$$

Assume that with respect to this norm the objective function $f(x)$ is Lipschitz continuous on B_n:

$$|f(x) - f(y)| \leq L\|x - y\|_\infty \quad \forall x, y \in B_n,$$

with some constant L (Lipschitz constant).

Complexity Bounds for Global Optimization

Consider the simple *uniform grid method* $\mathcal{G}(p)$ for solving problem (2):

1. Form $(p+1)^n$ points
$$x_{(i_1,\ldots,i_n)} = \left(\frac{i_1}{p}, \frac{i_2}{p}, \ldots, \frac{i_n}{p}\right)^T,$$
where $(i_1, \ldots, i_n) \in \{0, 1, \ldots, p\}^n$.
2. Among all points $x_{(i_1,\ldots,i_n)}$ find a point $\bar{x}$, which has the minimum value of the objective function.
3. Return the pair $(\bar{x}, f(\bar{x}))$ as the output.

This is a zero-order iterative method, without any influence of the accumulated information on the sequence of test points.

Complexity Bounds for Global Optimization

Theorem

Let f^* be the global optimal value of problem (2). Then

$$f(\bar{x}) - f^* \leq \frac{L}{2p}.$$

Proof: Exercise.

If our goal is to find $\bar{x} \in B_n : f(\bar{x}) - f^* \leq \epsilon$, then we have the following corollary:

Analytical complexity of the considered problem class for method $\mathcal{G}$ is at most

$$A(\mathcal{G}) = \left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor + 2\right)^n.$$

- Can we also derive *lower complexity bounds* for this class of problems?

Complexity Bounds for Global Optimization

Consider the class of problems $\mathcal{C}$ defined as follows:

Model:	$\min_{x \in B_n} f(x)$ $f(x)$ is l_∞-Lipschitz continuous on B_n.
Oracle:	Zero-order local black box.
Approximate solution:	Find $\bar{x} \in B_n : f(\bar{x}) - f^* \leq \epsilon$.

Theorem

For $\epsilon < L/2$, the analytical complexity of $\mathcal{C}$ for zero-order methods is at least $\left(\lfloor \frac{L}{2\epsilon} \rfloor\right)^n$.

Complexity Bounds for Global Optimization

Proof:

- Denote by $p = \lfloor \frac{L}{2\epsilon} \rfloor$. Assume that there is a method that needs $N < p^n$ calls of oracle to solve any problem from $\mathcal{C}$.
- Resisting strategy: Oracle returns $f(x) = 0$ at any test point x. So, the method can only find $\bar{x} \in B_n$ with $f(\bar{x}) = 0$.
- However, note that there exists $\hat{x} \in B_n$ such that

$$\hat{x} + \frac{1}{p} e \in B_n, \quad e = (1, \dots, 1)^T \in \mathbb{R}^n,$$

and there were no test points inside the box

$$B = \{x : \hat{x} \leq x \leq \hat{x} + \frac{1}{p} e\}.$$

Complexity Bounds for Global Optimization

Proof (continued)

- Let $x_* = \hat{x} + \frac{1}{2p} e$ and consider the function

$$\bar{f}(x) = \min\{0, L\|x - x_*\|_\infty - \epsilon\}.$$

- $\bar{f}(x)$ is l_∞-Lipschitz continuous with the constant L, its global optimal value is $-\epsilon$, and it differs from zero only inside the box $B' = \{x : \|x - x_*\|_\infty \leq \epsilon/L\} \subseteq B \equiv \{x : \|x - x_*\|_\infty \leq \frac{1}{2p}\}$ (since $2p \leq L/\epsilon$).
- Thus, $\bar{f}(x) = 0$ at all test points of our method.
- Since the accuracy of the result of our method is ϵ, we conclude that the accuracy of the result cannot be better than ϵ if the number of calls of the oracle is less than p^n. $\square$

Complexity Bounds for Global Optimization

- Recall that for the method $\mathcal{G}$ the upper bound on the analytical complexity was

$$A(\mathcal{G}) = \left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor + 2\right)^n.$$

- The derived lower bound for zero-order methods is

$$\left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor\right)^n.$$

- Thus if $\epsilon = O(L/n)$, the lower and upper bounds are asymptotically equivalent.
- This implies that $\mathcal{G}(p)$ is an optimal method for $\mathcal{C}$.

Complexity Bounds for Global Optimization

Example:

- Consider the problem class $\mathcal{F}$ defined by

$$L = 2, \; n = 10, \; \epsilon = 0.01.$$

- The lower complexity bound for this class is $\left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor\right)^n$, so to solve our example we need at least 10^{20} calls of oracle, each requiring at least $n = 10$ arithmetic operations, resulting in the total of at least 10^{21} arithmetic operations.
- If a work station carries out 10^6 arithmetic operations per second, we will need 10^{15} seconds to solve the problem.
- This is approximately 31,250,000 years!