Lecture 1: Introduction to mathematical optimization

Rajat Mittal
IIT Kanpur

1 Mathematical optimization

An optimization is a process of maximizing or minimizing a quantity under given constraints. Most of the problems in this world are optimization. You have to maximize (happiness/peace/money) or minimize (poverty, grief, wars etc.). Unfortunately we are not solving any of those problems. On a smaller scale, there are many real world problems where we need to optimize mathematical quantities and the constraints can also be represented as mathematical functions. For example, optimizing time in the production cycle of an industry, optimizing tax in a tax-return, optimizing length in a tour are mathematical optimization problems we encounter in our daily life.

Formally, any problem of the form:

$$\min f_0(x)$$
$$\text{s.t. } f_i(x) \leq b_i \text{ for } i = 1, 2, \ldots, m$$

is called a mathematical optimization problem. Here $f_0$ is the objective/optimization function and $f_i \leq b_i$ are called constraints. The task here is to find the max/min value of $f_0(x)$, s.t., $x$ satisfies all the constraints. An $x$ satisfying all the constraints is called a feasible solution. The set of all $x$'s satisfying all the constraints is called the feasible region.

$$S = \{x : f_i(x) \leq b_i \text{ for } i \in [m]\}$$

A feasible solution $x^*$ is called an optimal solution if it has the smallest objective value among all the feasible solutions. So for any feasible $z$, i.e., $f_i(z) \leq b_i$ for all $i \in [m]$, we know, $f_0(z) \geq f_0(x^*)$. Let’s consider some examples,

- Take our/mine favorite example of world peace, $x$-actions, $f_0$- peace function, $f_i$- many, dont kill everyone/anyone
- Other example: Satisfiability, given a boolean satisfiability formula, $(x_1 \lor x_2 \lor x_4), (\bar{x}_2 \lor x_4 \lor x_1), \ldots$

$$\max \# \text{ of clauses satisfied}$$
$$\text{s.t. } x \in \{0, 1\}^n$$

- Portfolio optimization – Every variable represents amount spend in each asset. Constraints might be on budget/availability/expected return. Objective is to minimize risk.
- Data fitting – The task is to find some model from some class of models which fit the data. What are the constraints, objective function, variables.

NOTE: Remember that optimal solution need not be unique. One of the special case is when variables have symmetry, in this case some kind of permutation can be applied to get multiple optimal solutions.

2 Classes of optimization problems

It is quite clear from the previous discussion that general optimization problems seem to be really hard. Hence, we are interested in classes of optimization problems which can be solved easily or have specific properties. These different classes differ in the kind of constraints and objective functions that are allowed
to be included in these problems. One of the example is Linear programming, the main focus for this course (constraints and objective function have to be linear).

A natural question might be, What kind of classes should be studied? A class of problems is interesting if:

- Many real world problems which can be modeled as a problem in that class.
- Problems in the class are easily/efficiently solved.
- Problems in the class have nice properties (e.g., Duality), which can give us more information about the structure of the problem (this will become clear later).

Linear programming satisfies all the above properties and hence a natural candidate to be studied. Our emphasis will be to understand why Linear programming can be solved efficiently, how to solve them and see some applications of them in the field of theoretical computer science.

3 Linear Programming

3.1 Definition

Linear programming is one of the well studied classes of optimization problem. We already discussed that a linear program is one which has linear objective and constraint functions. This implies that a standard linear program looks like

\[
\begin{align*}
\min & \quad \sum c_i x_i = c^T x \\
\text{subject to} & \quad a_i^T x_i \leq b_i \quad \forall i \in \{1, \cdots, m\}
\end{align*}
\]

Here the vectors \(c, a_1, \cdots, a_m \in \mathbb{R}^n\) and scalars \(b_i \in \mathbb{R}\) are the problem parameters.

3.2 Examples

- Max flow: Given a graph, start(s) and end node (t), capacities on every edge; find out the maximum flow possible through edges.

\[\text{Fig. 1. Max flow problem: there will be capacities for every edge in the problem statement}\]
The linear program looks like:

\[
\begin{align*}
\text{max} & \quad \sum_{s,u} f(s,u) \\
\text{s.t.} & \quad \sum_{u,v} f(u,v) = \sum_{v,u} f(v,u) \forall v \neq s, t \\
& \quad 0 \leq f(u,v) \leq c(u,v)
\end{align*}
\]

**NOTE:** There exist another linear program for the same problem, which can be made using the flow through paths.

- Another example: This time, I will give the linear program and you will tell me what real world situation can be modelled using this :).

\[
\begin{align*}
\text{max} & \quad 2x_1 + 4x_2 \\
\text{s.t.} & \quad x_1 + x_2 \leq 10, \, x_1 \leq 4
\end{align*}
\]

### 3.3 Solving linear programs

You might have already had a course on linear optimization. So you might know that there are many known algorithms for solving linear programs; like simplex, ellipsoid and interior point method. Simplex method was one of the first methods to solve these programs. But almost all initial versions have examples which will take too long (exponential time) to solve. It is an open question if some version of simplex can run in polynomial time for all the instances. Since it is efficient in practice, it is used in many places.

The first polynomial time algorithm was Ellipsoid algorithm. It is not found to be very efficient in practice. Few years later, interior point method was developed and shown to be in polynomial time. Since it is efficient in practice and is provably fast, it is implemented in a lot of places.

Because of the abundance of algorithms to solve linear programs, researchers were really excited about this paradigm. There were many attempts to solve even NP hard problems (like traveling salesman problem) using linear programming. Notice that this will prove one of the most fundamental questions of complexity theory, P=NP. This is because we know that linear programs can be solved in polynomial time.

Recently there was a big result by Wolf et. al. where they showed that most of these techniques are bound to fail. They showed that the traveling salesman polytope or its extension will require exponential number of constraints. This result will be covered later in the course.

### 4 Convex optimization

Convex optimization is a generalization of linear programming where the constraints and objective function are convex. It is interesting because most of the algorithms for linear programming can be generalized to convex optimization too. More importantly, many more problems can be expressed in this framework than linear programming. Many subclasses of convex optimization like semidefinite programming and least square problem are also widely used and have important applications in various fields.