Analysis of Mathematical Optimization

Some Classes, Problems, and Algorithms

by Liam Wrubleski

April 2020

Contents

- The general idea of mathematical optimization is to find an item x in a set D that minimizes the value of a function $f : D \to \mathbb{R}$.

General Statement

Find

$$m \in \mathbb{R}, x_0 \in D$$

such that

$$m = f(x_0) = \min_{x \in D}\{f(x)\}$$

for some set D, and

$$f : D \to \mathbb{R}$$

- The function f is referred to as the **objective function**.
- In general, the set D may be any set, but here we only analyze sets $D \subseteq \mathbb{R}^n$, for any positive integer n (this is called **continuous optimization**).
- Often, the set D is specified using **constraint functions**:

$$D = \{\mathbf{x} \in \mathbb{R}^n | g_i(\mathbf{x}) \leq 0, h_j(\mathbf{x}) = 0\}$$
$$g_i : \mathbb{R}^n \to \mathbb{R}, i = 1, ..., p$$
$$h_j : \mathbb{R}^n \to \mathbb{R}, j = 1, ..., q$$

- The functions g_i are **inequality constraints**, and h_j are **equality constraints**.

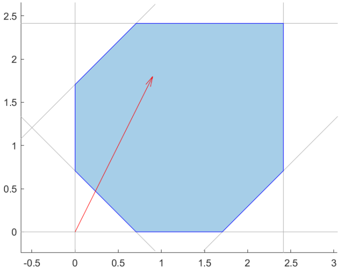
- Linear Programming (LP) is a very simple class of mathematical optimization using only inequality constraints
- When expressed in standard form, a linear programming problem has the following properties

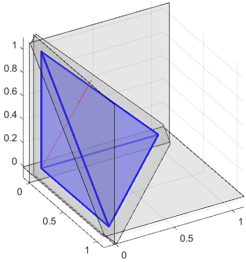
$$
\begin{aligned}
f(\mathbf{x}) &= c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
g_i(\mathbf{x}) &= a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n - b_i, i = 1, \ldots, m \\
x_j &\geq 0
\end{aligned}
$$

- This is usually expressed in standard form as

$$
\begin{array}{ll}
\text{maximize} & \mathbf{c}^T\mathbf{x} \\
\text{subject to} & A\mathbf{x} \leq \mathbf{b} \\
 & \mathbf{x} \geq \mathbf{0}
\end{array}
$$

- The space of possible solutions satisfying all of the constraints may be considered an n-polytope.
- For $n = 2, n = 3$, this gives us an intuitive geometric interpretation, shown for two contrived examples in 2d and 3d respectively on the following slides.
- The blue polytope represents the space of possible solutions, the grey lines/planes represent the constraints, and the red arrow represents the direction of optimization.





$$\begin{array}{ll} \text{maximize} & x + 2y \\ \text{subject to} & -x - y + \frac{1}{\sqrt{2}} \leq 0 \\ & -x + y - \frac{\sqrt{2}+1}{\sqrt{2}} \leq 0 \\ & -x + y + \frac{\sqrt{2}+1}{\sqrt{2}} \leq 0 \\ & x - 1 - \sqrt{2} \leq 0 \\ & y - 1 - \sqrt{2} \leq 0 \\ & x, y \geq 0 \end{array}$$

$$\begin{array}{ll} \text{maximize} & x + y + 3z \\ \text{subject to} & x + y + z - 1 \leq 0 \\ & x, y, z \geq 0 \end{array}$$

- Developed by George B. Dantzig in 1947 to solve LP problems, using the LP formulation of Leonid Kantorovich.
- Finds an optimal solution to an LP problem by moving along the edges of the polytope associated with the problem.
- Efficient on random problems, although inefficient in the worst case.

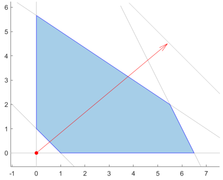
- The exact operation of the algorithm takes too long to examine here, but will be examined in my paper.
- However, as a high level examination, each pivot moves along one edge of the polytope connected to the current point.
- This requires a number of row operations equal to the number of constraints in the model.
- We will examine the intermediate results of the algorithm for a small example.

- Suppose a farmer has 10 km^2 of land on which to grow wheat and barley, and that she has 17 kg of fertilizer and 13 kg of pesticide to use to grow it.
- Every 1 km^2 of wheat requires 2 kg of fertilizer and 2 kg of pesticide, and sells for $6000 of profit.
- Every 1 km^2 of barley requires 3 kg of fertilizer and 1 kg of pesticide, and sells for $5000 of profit.
- Suppose that the farmer needs to plant at least 1 km^2 of crops, regardless of any other consideration.

How should the farmer plant her crops to maximize her profit?

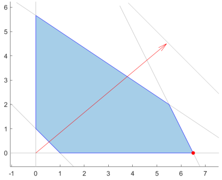
$$
\begin{aligned}
\text{maximize} \quad & 6x + 5y \\
\text{subject to} \quad & x + y - 10 \leq 0 \\
& 2x + 3y - 17 \leq 0 \\
& 2x + y - 13 \leq 0 \\
& -x - y + 1 \leq 0 \\
& x, y \geq 0
\end{aligned}
$$

Linear Programming - LP
Simplex Example

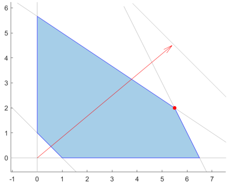


$$x = 0$$
$$y = 0$$

Note this point is not actually in the feasible space!



$$x = 6.5$$
$$y = 0$$



$$x = 5.5$$
$$y = 2$$

This is our optimal point, with a maximized profit of $43,000.

Klee-Minty Cube

Klee and Minty showed that on the following problem, the usual simplex algorithm implementation visits every vertex of the polytope, giving a worst case complexity of $O(2^n)$.

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{D} 2^{D-i} x_i \\ \text{subject to} & \sum_{i=1}^{k-1} 2^{D-i+1} x_i + x_k + k \leq 5^k, k = 1, ..., D \\ & x_i \geq 0, i = 1, ..., D \end{array}$$

The polytope for this problem looks like a squashed hypercube, which is where the name Klee-Minty Cube originates.

- A variant of linear programming, with the additional requirement that some or all of the variables must have integer values.
- This is a significantly more difficult variation of the problem.

How much harder is it?

For reference, with just 120 variables, each of which can be either 0 or 1, computing all possibilities would take 16 times longer than the age of the universe on the world's fastest supercomputer, assuming each possibility takes only one floating point operation. Allowing each variable to also be 2, it would take a billion trillion times the age of the universe.

- The way these problems are solved begins by solving the LP-relaxation of the problem.

LP Relaxation

The LP-Relaxation of the MILP problem

$$\begin{array}{ll} \text{maximize} & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) \leq 0, i = 1, \dots, m \\ & x_i \geq 0, i = 1, \dots, n \\ & x_i \in \mathbb{Z}, i = 1, \dots, k \leq n \end{array}$$

is an LP problem with the same objective and inequality constraints, but without the integrality constraints.

- Once the LP relaxation has been solved, it is used to find another solution to the constraints, but hopefully closer to a solution satisfying the integrality constraints. This is then repeated until it has been shown that the most recently found integer solution is optimal.
- The first common way of doing this is called **branch and bound**.
- The second common way of doing this is called **cutting planes**.

- This method combines branch-and-bound methods and cutting-plane methods to more quickly approach an integer value.
- Unfortunately, the best methods for balancing these techniques is an industry secret, and largely heuristic.
- In the worst case, branch-and-bound degenerates to brute force, and cutting planes generally do not help find integer solutions quickly, so even the best techniques are very difficult.

Knapsack Problem

Consider n items, each of which has a value v_i, a weight w_i, and a size s_i. You have a knapsack with a maximum weight W and a maximum volume S. The knapsack problem is to determine which items should be placed in the knapsack to maximize the carried value. This can be expressed as a MILP as follows:

$$\begin{array}{ll} \text{maximize} & \mathbf{v}^T\mathbf{x} \\ \text{subject to} & \mathbf{w}^T\mathbf{x} \le W \\ & \mathbf{s}^T\mathbf{x} \le S \\ & x_i \in \{0,1\}, i = 1, ..., n \end{array}$$

Here, x_i is 1 if you should pack item i, and 0 otherwise.

- Linear programming is one of the simplest classes of convex optimization, and while it isn't convex, mixed-integer linear programming is often studied in the same areas.
- There are a lot of different kinds of convex optimization, but they all share some characteristics
  - The objective function is convex, and
  - The constraints are convex.

Convex Sets

A set S is convex iff for any two points a, b, and any $\alpha \in [0, 1]$, $\alpha a + (1 - \alpha)b$ is also in the set S. Intuitively, if a line segment is drawn between the points a and b, the line lies entirely within the set S.

Epigraph

The epigraph of the function $f : \mathbb{R}^n \to \mathbb{R}$ is a set $S \subseteq \mathbb{R}^{n+1}$, with

$$S = \{(x_0, x_1, ...x_n) | x_0 \geq f(x_1, ...x_n)\}$$

Convex Functions

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex iff its epigraph is convex.

- Quadratic programming allows the objective function, but not the constraints functions, to take the form

$$f(\mathbf{x}) = \mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x}$$

- One special case of quadratic programming is ordinary least squares, which will be explained shortly.
- Note that Q must be positive semidefinite for f to be convex.

- A generalization of quadratic programming that allows the constraint functions to also take the form of convex quadratic functions, so the standard form looks like

$$
\begin{aligned}
\text{minimize} \quad & \mathbf{x}^T Q_0 \mathbf{x} + \mathbf{p}_0^T \mathbf{x} \\
\text{subject to} \quad & \mathbf{x}^T Q_i \mathbf{x} + \mathbf{p}_i^T \mathbf{x} + r_i \leq 0, \quad i = 1, \dots, m \\
& A\mathbf{x} = \mathbf{b}
\end{aligned}
$$

  where all $Q_i, i = 0, \dots, m$ are positive semidefinite.
- Problems that can be formatted as QCQP problems come up often in statistics (for example, constraining the variance of a random variable)

- A generalization of QCQP, this effectively allows the matrix Q to have a single negative eigenvalue (making it no longer positive semidefinite. It's generally phrased, however, as follows

$$\begin{array}{ll} \text{minimize} & \mathbf{f}^T\mathbf{x} \\ \text{subject to} & \|A_i\mathbf{x} + \mathbf{b}_i\|_2 \leq \mathbf{c}_i^T\mathbf{x} + d_i, \quad i = 1, \dots, m \\ & F\mathbf{x} = \mathbf{g} \end{array}$$

- This kind of optimization is very useful in many real-world engineering problems, as the 2-norm can be used to constrain distances and placements of items.

- This is the most general class of convex optimization discussed here, and is also the hardest to solve. However, every other convex optimization class expressed above can also be expressed as an SDP problem.
- This means a solver for SDP problems can solve all of the problems listed above, so research into SDP solvers is more advanced than research into specific solvers for most of the above problems (with the exceptions of LP and MILP).

- The vast majority of potential optimization problems are not convex. Some of these problems have an equivalent convex problem, as with the GP problems shown above.
- However, some of these problems do not have equivalent convex problems.
- Non-convex problems may have several local extrema that are not globally optimal.
- Moreover, in real-world optimization problems, we may encounter objective functions or constraints that do not have a closed form.

- In "Mode-pursuing sampling method for global optimization on expensive black-box functions", Wang et al. detail an optimization algorithm that finds the global optimum of an expensive black-box function (likely non-convex).
- This is a kind of **stochastic optimization**, which uses a stochastic process in its evaluation.
- Much of this paper details the methods used for generating the random points. This is interesting, but out of scope for this presentation, so the following slides do not detail the methods used to generate the random points. This is summarized in my paper.

- This algorithm has four primary steps:
  1. Generate a small number m of points $x^{(1)}, \dots x^{(m)}$ using the current PDF g (initialized to be uniform over S).
  2. Evaluate the black box function at these points, giving $(x^{(i)}, f(x^{(i)})$. Use these points, along with any previous points evaluated, to generate an approximation $\hat{f}$ of f, such that $\hat{f}(x^{(i)}) = f(x^{(i)}) \forall i$.
  3. Take some $c \in \mathbb{R}$ such that $c > \hat{f}(x) \forall x \in S$, and let the PDF $g(x) = k(c - \hat{f}(x))$, where k is a normalizing constant so that $\int_S g(x) dS = 1$.
  4. If some stopping criterion is met, return the generated point with the best value. Otherwise, go to step 1.

Conclusion

Today, we went over several classes of optimization problems, explored the theory behind them, where they are useful, and some methods for solving them. I hope you found this interesting, and if you have any questions please feel free to ask me, and read my paper!

The End