Section Notes 1

Introduction to Optimization and Mathematical Programming

Applied Math / Engineering Sciences 121

Week of September 11, 2017

Goals for the week

- To understand the components of an optimization problem.
- To take a real world problem and systematically go about formulating an optimization problem to solve it.
- To solve linear programs via the graphical approach.
- To understand why and how we can generalize optimization problems across data sets

Contents

1 Optimization

1.1 What is optimization?

Welcome to AM/ES 121! In this class, we will frequently take real world problems, come up with models to capture the essence of these problems, and then solve the models using various techniques we study. While this course focuses on linear optimization, the concept and practice of taking a problem and turning it into a model applies to any optimization problem.

So, what is optimization? Before we get to definitions, let's think about some typical 'optimization' problems:

1. Find the shortest path from my dorm room to my first class.
2. Figure out an investment strategy to maximize profit.
3. Find a schedule that fits in all your lectures, sections, and extracurricular activities.

One characteristic that all these problems share is that there is a decision to make. We need to find a path, an investment strategy, or a schedule. There may be many different ways to make the decision, and we may value different decisions differently. In the first two examples, we see that there is a criteria by which to evaluate the decision. In the first case, we are looking for a path that is *shortest*. In the second case, we are looking for an investment strategy that *maximizes profit*. Of course, we must clearly specify the evaluation criterion. For example, when we want the shortest path, we must specify whether we want the shortest path in terms of Euclidean distance or in the amount of time to get from source to destination. Likewise, when we are maximizing profit, we should specify whether we wish to maximize expected profit or worst case profit. In the third case, it is not clear what the evaluation criterion is. Given two schedules that both fit all the events, is there a sense in which one is better than another? If so, we must clearly specify the evaluation criterion. If not, we may be satisfied with any schedule, in which case we don't need to 'optimize' across different schedules.

While the evaluation criterion tells us what we are trying to optimize, it doesn't tell us about the kind of decisions we are allowed to make. For example, for finding the shortest path, can we just draw a line between my location and the desired destination and walk along that line? Maybe if you are walking on an empty field, but otherwise, you will probably bump into a building! Similarly, for maximizing profits in making investments, there are limits to the kind of investments we can make. For example, we may only be able to make as large an investment as our budget or government regulations allows. These limits on how we can invest may impact the investment strategy we choose, just like how not being able to walk in a straight line will impact our chosen walking path. Thus, we often cannot choose our decisions entirely freely, because the set of possible decisions are often *constrained*.

Much of the work in modeling a real world problem into an optimization problem we can solve is in formally specifying what decisions we are making, how these decisions are evaluated, and how the decisions are constrained. To get into this a bit more, let's look at the Wikipedia definition of optimization:

```
Optimization (mathematics) - In mathematics, the term optimization,
or mathematical programming, refers to the study of problems in which
one seeks to minimize or maximize a real function by systematically
choosing the values of real or integer variables from within an
allowed set.
```

Okay, this is a mouthful, but let's break it down. The definition points to 3 things:

- a set of variables to choose values for
- an allowable set from which to choose the values of these variables
- a function to maximize or minimize

Every optimization problem will contain these 3 components. We call these components the *decision variables*, the *feasible region*, and the *objective function*, respectively. Let's take a closer look at each one:

- Decision variables
  These variables represent the underlying decisions that you are making. For finding the shortest path, the decision would be which path to take. In this and many other problems, you will notice that there are often many ways to represent the same underlying decision. For example, in the shortest path problem, you can model the decision with a variable representing the path you would take from start to finish, or alternatively, as a set of decision variables over possible locations such that for each location, there is a variable representing the location to visit next while in that particular location. While we won't make a big deal out of it now, the choice of decision variables is often quite important, as it can have an effect on the difficulty of representing constraints and solving the underlying optimization problem.

- Feasible region
  The feasible region defines the allowable set from which to choose the values of the decision variables. Any assignment of values to the decision variables that is in the feasible set is called a *feasible solution*. For example, any investment strategy that doesn't go over the budget and complies with regulations is a feasible investment strategy (might not be a good one, but it's feasible).

  In representing the feasible region, it is often convenient to use a set of *constraints* to impose limits on values that decision variables can take on rather than to enumerate the elements of the allowable set (which can be infinite). Note that *a constraint is only meaningful if it is a constraint on decision variables!*

- Objective function
  The objective function is how we evaluate our choice of values for the decision variables from the feasible region. Given a point in the feasible region, we can plug it into our objective function to get the objective value at that point, which gives a quantitative measure of the 'goodness' of the solution. If we are maximizing, the larger the value of the objective function, the better we like the solution. The goal then is to find the point in the feasible region that leads to maximizing (or minimizing) the objective function. Any point in the feasible region that satisfies this goal is called an *optimal solution*.

In formulating an optimization problem, we must identify and write down in precise mathematical notation each of the three components above. **Often, we will find it easiest to first identify the variables, then the objective function, followed by the constraints.** If we find it difficult to represent the constraints of the problem, it may be that our choice of variables are contributing to the difficulty and we should thus reconsider our choice.

Formulating the optimization problem is only half the work! We need to be able to solve the models we come up with and actually figure out what the optimal solution is. Note here that it is possible for an optimization problem to have no solutions (that is, the feasible set is empty), in which case we say the problem is *infeasible*. It is also possible that the solution is unbounded, that is, the feasible region is not constrained enough such that variables can take on values that cause the objective value to be arbitrarily large (for a maximization problem) or arbitrarily small (for a minimization problem).

It is worth noting that in the scheduling example, we may not have an objective function to evaluate a feasible schedule and that all feasible schedules are equally good. In this case, we call the problem a *feasibility problem* instead of an optimization problem, where the goal is to find any point in the feasible region.[1] For almost all problems we consider, there will be an objective function based on which to evaluate a solution.

2 Linear Programming

2.1 What is a linear program?

In this course we are concerned with optimization, and in particular, linear optimization. A linear program (LP) is an optimization problem that involves maximizing or minimizing of a linear objective function by choosing values for decision variables subject to linear equality and inequality constraints.

Linear program models satisfy *proportionality*, *additivity*, and *certainty*:

- Proportionality: The contribution of each variable is directly proportional to its value in the objective function and constraints. If the value of a variable is doubled, so is its contribution.
- Additivity: Contributions of variables into the objective function and constraints are via the sums (or differences) of individual contributions of each variable.
- Certainty: The objective and constraint coefficients are data, that is, they are known constants.

Exercise 1

What is an example of a problem we cannot represent with a linear program model?

End Exercise 1

[1] If the feasible set contains only one point, the problem isn't much of an optimization problem since there is only one solution anyway. We can think of such cases as feasibility problems if we like.

Exercise 2

Give an example of a linear program that does not satisfy the proportionality requirement. How about additivity? Certainty?

End Exercise 2

Exercise 3

Why would we want to limit ourselves to an LP model?

End Exercise 3

2.2 Graphical approach to solving 2 variable LPs: Cookies and Brownies

Now that we know what linear programs are, let's try to do a complete example, where we take a problem, model it, and then solve it.

Consider the following problem:

```
Chuck and Morgan are making brownies and cookies for a bake sale. Each
brownie sells for $1.00 and requires 1 ounce of dough and 3
tablespoons of sugar.  Each cookie sells for $1.50 and requires 2
ounces of dough and 4 tablespoons of sugar.  They have 4 ounces of
dough and 10 tablespoons of sugar. Chuck and Morgan wish to
maximize profit (they don't need to account for cost; the ingredients
were free from Ellie).
```

Exercise 4

What are the variables?

End Exercise 4

Exercise 5

What is the objective function?

End Exercise 5

Exercise 6

What are the constraints?

End Exercise 6

Exercise 7

Here the LP only has two variables; we can solve it via a simple graphical method. Let's graph this linear program.

- Label the axis.
- Graph the first constraint.
- Graph the second constraint.
- Locate the feasible region.
- Draw contour lines for the objective function.
- Find its furthest out point that still touches the feasible region. This is the optimal solution.

End Exercise 7

Exercise 8

Confirm algebraically that the value you found is indeed optimal.

End Exercise 8

Notice that the algebraic 'proof' of optimality required us to find a tight bound on the value of the objective function. Had we only been able to demonstrate that the objective is ≤ 4, we would not be able to conclude that our candidate solution was optimal. We will see later in the course that this concept of combining constraints to generate an algebraic 'proof' of optimality can be generalized and has important implications for how we solve and analyze linear programs.

2.3 Linear program in standard form

In standard form, a linear program can be defined as follows:

$$\max \mathbf{c}^{\mathbf{T}}\mathbf{x} \tag{1}$$
$$A\mathbf{x} \leq \mathbf{b} \tag{2}$$
$$\mathbf{x} \geq 0 \tag{3}$$

Here $\mathbf{c}^{\mathbf{T}}$ is an n-dimensional row vector of objective coefficients, $\mathbf{x}$ is an n-dimensional column vector of variables, $\mathbf{A}$ is an m by n constraint coefficient matrix, and $\mathbf{b}$ is an m-dimensional column vector of bounds on constraints. As we will see later, we can express any linear program in this form. Notice also the use of less than or equal to instead of less than comparisons in the constraints (why might this be important?).

What standard form means for our purposes is that we can develop the theory in terms of this one standard form, but write down linear programs in any form convenient for us. Furthermore, we will hardly ever write down constraints in coefficient matrix, but instead as a set of constraints.

Consider the following linear program:

$$\max 2x_1 + 3x_3 + 4x_5 \tag{4}$$
$$x_2 + 5x_3 \leq 9 \tag{5}$$
$$4x_1 + 6x_3 + 3x_5 \leq 5 \tag{6}$$
$$x_3 + x_4 \leq 7 \tag{7}$$
$$x_1, x_2, x_3, x_4, x_5 \geq 0 \tag{8}$$

Exercise 9

Put the above linear program into matrix form.

End Exercise 9

By completing the above exercises, you should now:

- know the definition of linear programming and the standard form.
- convert a set of constraints to matrix form (and vice versa).
- be able to solve 2 variable linear programs using the graphical approach.
- be able to use algebraic manipulation of the constraints to prove the optimality of graphically-found solutions.

2.4 A generalized model: using set notation and parameters

So far we have talked about linear programs in terms of variables, objective function, and constraints. While this suffices to describe a linear program, we can see that if our problem were more realistic - with more items to bake and a few more constraints - it would both be a pain to write down and impossible to illustrate with a simple graph. In an actual production problem, a manufacturer may produce thousands of products subject to tens of thousands of constraints representing various production limits. There we certainly don't want to have to write down all the constraints by hand!

While the size of the problem differs in these two cases, what we notice is that the underlying model is actually the same. For both the bake sale problem and the manufacturing problem, there will be variables representing the amount of each product to produce, and constraints on the availability of raw materials. In fact, even within the bake sale example, we see that the dough and sugar availability constraints are of the same form, but just with different numbers for amounts supplied and required for the raw materials. Furthermore, if the amount of raw material required for a product changes, only the coefficient corresponding to that data needs to change.

What we want is a general yet concise way of expressing linear programs that takes advantage of the problem's structure, so that we can define a *model* for a general problem (e.g. the production problem) that abstracts away the *data* for a particular problem we want to solve. For a production problem, a simple general model may contain an objective to maximize profit over products produced, subject to raw material requirement and availability constraints. Given that we have this model, for a particular problem instance, we need only specify the relevant data (which is known), e.g., a list of products, a list of raw materials, product prices, raw material requirements, and raw material supply. Taken together, the model with the data from a particular instance completely specifies the optimization problem for that particular problem instance.

So, how do we go about formulating generalized mathematical models? Here is where mathematical notation comes to the rescue. By using *sets* and *parameters*, we can write a compact description of the model:

- A set is a representation of a collection of related objects. For example, we can define the set of integers, the set of products, the set of raw materials, the set of animals, and so on.
- A parameter is a representation of a piece of data given to the model. For example, the supply availability is a parameter. Often we will have the same kind of parameter for all elements of a set, e.g., a supply availability parameter for each element of the raw materials set. In such cases, we can define the parameter values over a set. For example, if $R = \{\text{sugar}, \text{dough}\}$ represents the set of raw materials, we can let s_r, $r \in R$ represent the supply availability parameter corresponding to raw material r.

When formulating a model, we should begin by defining sets, followed by parameters and variables (possibly over sets). Once we have them, we can think about the objective function and constraints in terms of the parameters and variables. Remember that parameters are known constants for a particular problem instance!

Exercise 10

Let's begin generalizing the production model. What are the relevant sets? What are the relevant parameters and variables?

End Exercise 10

Exercise 11

What is the objective function?

End Exercise 11

Exercise 12

What are the constraints? Can we come up with one expression that captures the raw material usage and supply limit?

End Exercise 12

These together make up our generalized production model. Notice how, once we defined the relevant sets and parameters, we only needed one line to describe the set of constraints for this problem. For any particular instance of this problem, we now only need to specify the elements in the defined sets and the values of the corresponding parameters.

Exercise 13

Write down the sets and parameter values based on the data from the bake sale example.

End Exercise 13

The generalized model combined with the data from a particular problem instance fully specifies the optimization problem that we'd like to solve. But this is only part of the work – we need to be able to actually solve the optimization problem and arrive at a solution! For the most part, linear programs are easy to solve and most of the work is actually in coming up with a formulation that is linear. We will look into this more next time.

By completing the above exercises, you should now:

- understand the distinction and relationship between a genearlized mathematical model and the data from a particular problem instance.
- have a framework for formulating a generalized linear programming model systematically and writing it down in precise mathematical notation using sets and parameters.

2.5 Practice: expressing constraints

Let's get some practice with expressing constraints and objective functions using set and parameter notation. Consider a set of n products P, such that product $p \in P$ has a rate of production r_p denoting how many of product p can be produced in an hour. Let M be a vector of variables over how much of each product to make, such that M_p denotes the number of product p we produce. Let parameters c_p denote the cost of producing one unit of product p for all $p \in P$.

Exercise 14

Write down the following with a linear equation (if you can!):

1. There are only k hours available for production.
2. The demand for product q is d.
3. We must produce at least twice as much product $p1$ than product $p2$.
4. We must produce equal numbers of $p1$ and $p2$.
5. The average number of each product we produce must be at least l.
6. Minimize the total production cost.

End Exercise 14

2.6 A Review of AMPL syntax

As part of your current problem set, you will need to be able to create matrices of both variables and parameters in your AMPL code.

Exercise 15

1. What is the syntax for declaring a parameter?
2. What is the syntax for declaring a vector of parameters?
3. What is the syntax for declaring a matrix of parameters?
4. What is the syntax for declaring a variable?
5. What is the syntax for declaring a vector of variables?
6. What is the syntax for declaring a matrix of variables?
7. What is the syntax for defining a parameter?
8. What is the syntax for defining a vector of parameters?
9. What is the syntax for defining a matrix of parameters?

End Exercise 15

By completing the above exercises, you should now:

- be comfortable with basic AMPL syntax relevant to defining LPs.

3 Optional Exercise: Kicking a Football

Below is an optional exercise intended to help you gain a better understanding of optimization. Note that this is not a linear optimization problem.

Consider the following:

```
You are standing behind a football and want to kick it into
the air such that it lands as far as possible down the field.
```

Exercise 16

What are the decisions you have to make (*that actually influence outcome*)?

End Exercise 16

Exercise 17

What are the constraints defining the feasible region?

End Exercise 17

Exercise 18

What is the objective function? (hint: it may require several steps to express your objective function in terms of the variable you've defined)

End Exercise 18

Exercise 19

Formulate the optimization problem.

End Exercise 19

Exercise 20

Solve the problem using results from single variable calculus.

End Exercise 20

Of course, we have made many assumptions here in the problem (e.g., no air friction, no bounce n' roll). But we often have to do this, both because it can be quite difficult to capture all the factors of the problem into a mathematical model and that doing so may make the model very difficult to solve. Here we just have one variable to consider, which is the angle to kick in, and have thus just applied results from single variable calculus. Notice further that the constraint here isn't actually constraining, that is, without it, we would get the same answer.

By completing the above exercises, you should now:

- understand the elements of an optimization problem.
- have a sense for how to systematically formulate an optimization problem and write it down using precise mathematical notation.

4 Solutions

Solution 1

Generally, anything that violates linearity. Product of variables, non-linear functions of variables, etc. One physics-related example is the optional exercise at the end of this packet. In reality, there are some situations in which we can represent non-linear functions with an LP - we will see more of this later.

End Solution 1

Solution 2

Limiting the scope of our answer to just the objective function, examples could take the shape of:
Does not satisfy proportionality: max $x_1^2 + 3x_2 + 2x_3$
Does not satisfy additivity: max $2x_1x_2 + 3x_4$
Does not satisfy certainty: max $yx_1^2 + 2x_2 + 4x_5$
Note that these violations could appear in the constraints as well.

End Solution 2

Solution 3

It is expressive enough to capture a large number of problems while being very easy to solve!

End Solution 3

Solution 4

x_1: number of brownies to make.
x_2: number of cookies to make.

End Solution 4

Solution 5

$\max_{x_1,x_2} x_1 + 1.5x_2$ maximize profit

End Solution 5

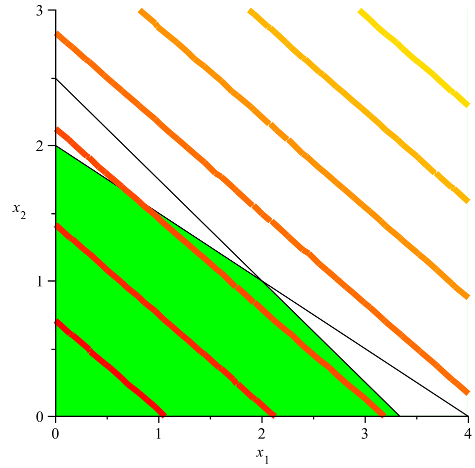
Solution 6

$x_1 + 2x_2 \leq 4$ dough availability
$3x_1 + 4x_2 \leq 10$ sugar availability

End Solution 6

Solution 7

This graph is generated using Maple. The optimal solution is $(x_1^*, x_2^*) = (2, 1)$.



End Solution 7

Solution 8

We want to show that $x^* = (2, 1)$ is optimal by showing that the objective function cannot have a value greater than the objective value at x^*, which is $2 + 1.5(1) = 3.5$. Since both constraints must be satisfied, any linear combination of the constraints must be satisfied as well. We can sum the two constraints to get:

$$4x_1 + 6x_2 \leq 14$$

Or:

$$x_1 + 1.5x_2 \leq 7/2$$

The fact that the objective function cannot be greater than 3.5 and we have a point with objective value 3.5 implies that x^* is an optimal solution to the LP.

End Solution 8

Solution 9

$$\mathbf{c^T} = \begin{pmatrix} 2 & 0 & 3 & 0 & 4 \end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix} 9 \\ 5 \\ 7 \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 5 & 0 & 0 \\ 4 & 0 & 6 & 0 & 3 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

End Solution 9

Solution 10

Sets	P	set of product types
	R	set of raw materials
Parameters	$s_r, r \in R$	amount of supply r available
	$p_i, i \in P$	per unit profit for product i
	$n_{ri}, r \in R, i \in P$	amount of raw material r required per unit production of i
Variables	$Q_i, i \in P$	quantity of product i to produce

End Solution 10

Solution 11

$$\max_Q \quad \sum_{i \in P} p_i Q_i \quad \text{maximize profit}$$

End Solution 11

Solution 12

For all raw material, the amount used must be less than the amount available. Thus we have a set of constraints, each corresponding to a particular raw material:

$$\sum_{i \in P} n_{ri} Q_i \leq s_r \quad \forall r \in R \quad \text{cannot use more raw material than is available}$$

End Solution 12

Solution 13

We have $P = \{\text{brownies, cookies}\}$, $R = \{\text{dough, sugar}\}$, $s = \{4, 10\}$, $p = \{1, 1.5\}$, $n_{\text{dough}} = \{1, 2\}$, and $n_{\text{sugar}} = \{3, 4\}$. Here we have assumed that the parameter values over sets are ordered according to the order of the elements appearing in the set (e.g., brownies before cookies).

End Solution 13

Solution 14

1. $\sum_{i \in P} (1/r_i) M_i \leq k$
2. $M_q \leq d$
3. $M_{p1} \geq 2M_{p2}$
4. $M_{p1} = M_{p2}$
5. $(\sum_{i \in P} M_i)/n \geq l$
6. $\min \sum_{j \in P} c_j M_j$

End Solution 14

Solution 15

1. `param foo >= 0;`
2. `param foo{SET} >= 0;`
3. `param foo{SET_A, SET_B} >= 0, <= 1;`
4. `var Bar >= 0;`
5. `var BarSET >= 0;`
6. `var Bar{SET_A, SET_B} >= 0;`
7. `param foo 0.376;`
8. ```
   param:  foo :=
   E1 5
   E2 6
   E3 7 ;
   ```
9. ```
   param foo:  A1 A2 :=
   B1 1 2
   B2 3 4 ;
   ```

End Solution 15

Solution 16

- How hard to kick the ball
- At what angle to send the ball flying

So we know that if you want to kick it as far as you can, you have to kick the ball as hard as you can. Given this, we can consider this decision made. Furthermore, we assume that when you kick the ball as hard as you can, it will send the ball flying at some constant speed v regardless of the angle in which you kick it. The decision then is what angle to send the ball flying, which will determine the velocity of the ball in the horizontal and vertical direction as it goes through the air. Let's use the decision variable θ to denote this choice.

End Solution 16

Solution 17

Okay, so we have got our decision variable. We need to figure out what the feasible region is. Well, given that we are kicking the ball down field, we can restrict our attention to θ in the forward and up direction, that is, non-inclusively from 0 degrees (kicking forward only) to 90 degrees (kicking it straight up into the air). We express this with a constraint:

$$0 < \theta < 90$$

Notice here that if our goal is to kick as far as we can down field, this constraint isn't actually *binding*, that is, isn't really constraining our choice of θ in our trying to maximize the objective.

To see this, notice that even without this constraint, choosing θ outside of this range could only be worse than choosing θ in this range, as we would kick backwards instead.

End Solution 17

Solution 18

We like to kick as far as possible, which is to say that we want the horizontal distance traveled when the ball lands from being kicked to be maximized. If we denote the magnitude of the velocity in the horizontal direction by v_x and denote t as the total time the ball is in the air, then the horizontal distance d_x can be expressed as:

$$d_x = v_x * t$$
$$d_x = v \cos\theta * t$$

Notice here that the horizontal velocity does not change while the ball is in the air.

Okay, so we have an expression for the distance the ball travels, but this is in terms of t. Is t a variable too? Well, the ball stays in the air until the velocity becomes 0 (pulled by gravity) and then takes the same amount of time to come down. Thus, the time of flight is:

$$t = 2 * v_y / g$$
$$t = 2 * v \sin\theta / g$$

where v_y is the magnitude of velocity in the vertical direction and g is the constant acceleration from gravity. Here we see that t is just a function of θ, so we can plug this back into the objective:

$$d_x = v \cos\theta * t$$
$$d_x = (v \cos\theta)(2 * v \sin\theta / g)$$

End Solution 18

Solution 19

$$\begin{array}{rll} \theta: & \text{angle to kick football} & \\ \max\limits_{\theta} & \frac{2v^2}{g}\cos\theta\sin\theta & \text{max horizontal distance travelled} \\ \text{subject to} & 0 < \theta < 90 & \text{only kick forward and up} \end{array}$$

End Solution 19

Solution 20

To maximize this function, we will ignore the constraint on θ as we'd reasoned and just take the derivative with respect to θ. Which gives us:

$$d'_x = (v^2/g)\cos 2\theta$$

Setting this to 0, we have:

$$\cos 2\theta = 0$$

Since $\cos 90^\circ = 0$, we have $\theta = 45^\circ$. We have solved our optimization problem!

End Solution 20