

The Mathematics of Decision Making I

Joshua Maglione

September 16, 2024

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 1 |
| 1.1 | History | 2 |
| 1.2 | Five examples | 2 |
| 2 | General linear programming | 5 |
| 2.1 | Standard form | 6 |
| 2.2 | Canonical form | 7 |
| 2.3 | Matrix notation | 9 |
| 2.4 | Geometry of the feasibly set | 10 |

1 Introduction

The mathematics of decision making is very closely tied to the field of mathematical optimization. One of the primary ways mathematics is used to help guide decisions is by maximizing (or minimizing) specific outcomes subject to a list of constraints. Mathematical optimization provides the formal tools to model and solve such problems.

There are many kinds of mathematical optimization. There are two basic types depending on whether the variables to optimize or discrete or continuous. A few types of optimization are¹

- Linear Programming,
- Integer Programming,
- Stochastic programming,
- Combinatorial optimization,
- Dynamic programming.

¹“Program” is not a computer program but comes from the United States military’s use of the word for training and logistics schedules.

Unsurprisingly there are many real-world applications; to list a few we have network optimization, pricing strategy, scheduling, supervised machine learning training, supply chain optimization, and transportation problems.

In this module, we will introduce the fundamentals of **linear programming**, also called *linear optimization* and *operations research*, such as the simplex method, polyhedral geometry, and the notion of duality. Depending on the time, we may also delve into **integer programming**.

1.1 History

Mathematical optimization has quite an interesting history. In the 17th century, combinatorial optimization problems were solved using game theory, combinatorics, and ad hoc methods. In the 19th century, transportation problems involving post and rail were studied and solved. And in the 20th century with the two World Wars and rise of the assembly line, operations research took off developing the mathematics for all kinds of optimization problems.

One of the most influential figures in mathematical optimization, and linear programming in particular, is George Dantzig. He was the recipient of the President's National Medal of Science in 1975 [3] and was credited for

inventing linear programming and discovering methods that led to wide-scale scientific and technical applications to important problems in logistics, scheduling, and network optimization, and to the use of computers in making efficient use of the mathematical theory.

The proof of the simplex method, name coined by Motzkin, was developed by Dantzig in the late 1940s [1]. I find it interesting that the “inductive proof of the simplex method” was published by the Mathematics Division of the RAND Corporation in 1960 (by Dantzig) and was made classified [2]. Now, of course, it is no longer classified.

After explaining the Simplex Method to John von Neumann at the Institute of Advanced Study in Princeton during 1948, von Neumann immediately conjectured the notion of duality because of his recent foray into game theory.

1.2 Five examples

We describe five example problems that touch on the tools we will develop in this module. For now, these problems are meant to introduce basic concepts and vocabulary.

1.2.1 A diet problem

Erin is planning her breakfast and wants to make oats with milk. (These numbers of simplified and not accurate to real life.)

| | Milk (100ml) | Oats (100g) |
|---------------|--------------|-------------|
| fat | 2g | 3g |
| carbohydrates | 1g | 3g |
| protein | 4g | 3g |

Erin wants the meal to provide at least 18g of fat, at least 12g of carbohydrates, and at least 24g of protein. If milk costs 20 cents per 100ml and oats 25 cents per 100g, what mixture minimizes the cost of the desired meal?

We could express this more mathematically. For example, let x and y be variables such that $x = 1$ means 100ml of milk and $y = 1$ means 100g of oats. Calculating the grams of fat relative to x and y is

$$2x + 3y.$$

For carbohydrates it is $x + 3y$, and for protein it is $4x + 3y$. Because we want *at least* 18g of fat, we express this via

$$2x + 3y \geq 18.$$

We can set up similar inequalities for the other two:

$$2x + 3y \geq 18,$$

$$x + 3y \geq 12,$$

$$4x + 3y \geq 24.$$

Since we cannot have negative amounts of milk or oats, we have $x \geq 0$ and $y \geq 0$. Since we want to minimize costs, we want to minimize

$$C = 0.2x + 0.25y.$$

Putting all of this together, we have the following optimization problem.

Determine values for x and y that minimize

$$C = 0.2x + 0.25y$$

subject to the constraints: $x \geq 0$, $y \geq 0$, and

$$2x + 3y \geq 18,$$

$$x + 3y \geq 12,$$

$$4x + 3y \geq 24.$$

1.2.2 A transportation problem

Javier has two production sites: one in Sligo and another in Kilkenny. There are three distributing warehouses in Dublin, Galway, and Cork. The Sligo site can supply 120 products per week, whereas the site in Kilkenny can supply 140 per week. The warehouses in Dublin, Galway, and Cork need 100, 60, and 80 products per week respectively to meet demand. The shipping costs are giving in the following table.

| | Dublin | Galway | Cork |
|----------|--------|--------|------|
| Sligo | 5 | 7 | 9 |
| Kilkenny | 6 | 7 | 10 |

How many products should Javier ship from each production site to minimize total shipping costs while still meeting demand?

We need many variables, so let's define a variable for each shipment—for example, from Kilkenny to Dublin. Write them as

$$x_{kd}, x_{kg}, x_{kc}, x_{sd}, x_{sg}, x_{sc}.$$

Since Kilkenny and Sligo can only produce 140 and 120 products, respectively, we have

$$x_{kd} + x_{kg} + x_{kc} \leq 140,$$

$$x_{sd} + x_{sg} + x_{sc} \leq 120.$$

We need to meet demands, so we have

$$x_{kd} + x_{sd} \geq 100,$$

$$x_{kg} + x_{sg} \geq 60,$$

$$x_{kc} + x_{sc} \geq 80.$$

Lastly, we want to minimize cost, so we want to minimize

$$C = 6x_{kd} + 7x_{kg} + 10x_{kc} + 5x_{sd} + 7x_{sg} + 9x_{sc}.$$

Altogether we have the following linear program.

Minimize

$$C = 6x_{kd} + 7x_{kg} + 10x_{kc} + 5x_{sd} + 7x_{sg} + 9x_{sc}$$

subject to the constraints: $x_{ij} \geq 0$ for all i and j and

$$x_{kd} + x_{kg} + x_{kc} \leq 140,$$

$$x_{sd} + x_{sg} + x_{sc} \leq 120,$$

$$x_{kd} + x_{sd} \geq 100,$$

$$x_{kg} + x_{sg} \geq 60,$$

$$x_{kc} + x_{sc} \geq 80.$$

1.2.3 The travelling salesperson problem

Kofi need to deliver n products in n different cities starting in Paris. He wants to do this by visiting each city exactly one time and then returning back to Paris at the end. Which path minimizes the distance traveled?

This problem is perhaps the most famous combinatorial optimization problem and is the core problem of many other more complex problems. We will not do much more with this, but note that different “distance functions” can allow for all kinds of slow-downs and speed-ups.

1.2.4 A financial problem

Julia runs an investment and must invest exactly €100,000 in two types of securities: bond A paying a dividend of 7% and stock B paying a dividend of 9%. Due to her incredible experience, she knows that

- no more than €40,000 can be invested in stock B and
- the amount invested in bond A must be at least twice that in stock B.

How much should Julia invest in each security to maximize her return?

See if you can get the following set up.

Maximize

$$z = 0.07A + 0.09B$$

subject to the constraints: $A \geq 0$, $B \geq 0$, and

$$A + B = 100000,$$

$$B \leq 40000,$$

$$A \geq 2B.$$

2 General linear programming

Linear programs are the basis of what we consider throughout this module. In the example problems above, we sometimes wanted to maximize and sometimes we wanted to minimize. Although these are technically different, we can treat them as the same. Suppose f is some function we want to maximize. Then

$$\max(f) = -\min(-f).$$

So maximizing f is the same as minimizing $-f$. Thus, we can use the two interchangeably—as long as we correctly compensate!

General linear program

Determine values for x_1, x_2, \dots, x_n that maximize

$$z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

subject to the constraints:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2,$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \leq \quad \vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m,$$

where each of the \square can be replaced with one of $\{=, \leq, \geq\}$.

Definition 2.1. A *linear program (LP) problem* is a problem of the form above. The function z is called the *objective function*, and the m (in-)equalities are called the *constraints*.

A key feature of LPs is that the objective function as well as each of the constraint (in-)equalities are *linear* in the x_1, x_2, \dots, x_n .

2.1 Standard form

Can we play around with the constants a_{ij} and b_k to get all of the (in-)equalities into the same “shape”? For example,

$$4x_1 - 5x_2 - x_3 \geq 1$$

is equivalent to

$$-4x_1 + 5x_2 + x_3 \leq -1.$$

Thus, if we have an inequality, we can force it to use just \leq . Moreover, if we have an equality, we can use two inequalities to obtain the same solutions:

$$4x_1 - 5x_2 - x_3 = 1 \quad \text{is equivalent to} \quad \begin{cases} 4x_1 - 5x_2 - x_3 \geq 1 \text{ and} \\ 4x_1 - 5x_2 - x_3 \leq 1 \end{cases}$$

So we can transform equalities to inequalities, but what about the other way around? We will look at this soon.

In some examples, variables only took on non-negative values. This actually has an advantage of constraining the possible values of the variables, and it is something we will come back to later on. But what about situations where variables are allowed to have negative values? Suppose x_i can be negative. We can introduce two new variables, say, x_i^+ and x_i^- , and we can rewrite x_i as follows:

$$x_i = x_i^+ - x_i^-.$$

In this way, x_i can be negative while both x_i^+ and x_i^- are non-negative. Thus, we can replace all instances of x_i with $x_i^+ - x_i^-$, so that all variables take non-negative values.

Now we can define the standard form for an LP.

Linear program standard form

Determine values for x_1, x_2, \dots, x_n that maximize

$$z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

subject to the constraints: for all $i \in \{1, \dots, n\}$, $x_i \geq 0$ and

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2,$$

$$\begin{array}{ccccc} \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & \leq & b_m. \end{array}$$

Example 2.2. The following LP is not in standard form.

Determine values for x and y that minimize

$$z = 3x + 2y$$

subject to the constraints: $x \geq 0$, $y \geq 0$, and

$$2x + y \leq 4$$
$$3x - 2y \leq 6.$$

$$z = 3x + 2y$$

subject to the constraints: $x \geq 0, y \geq 0$, and

$$2x + y \leq 4$$

$$3x - 2y \leq 6.$$

We can put it into standard form as follows.

Determine values for x and y that maximize

$$z = -3x - 2y$$

subject to the constraints: $x \geq 0$, $y \geq 0$, and

$$2x + y \leq 4$$
$$3x - 2y \leq 6.$$

$$z = -3x - 2y$$

subject to the constraints: $x \geq 0, y \geq 0$, and

$$2x + y \leq 4$$

$$3x - 2y \leq 6.$$

Example 2.3. Put the following LP into standard form.

Determine values for x and y that minimize

$$z = -4x + y$$

subject to the constraints:

$$\begin{aligned}x - 3y &= 2, \\ x + y &\leq 6.\end{aligned}$$

$$z = -4x + y$$

subject to the constraints:

$$x - 3y = 2,$$

$$x + y \leq 6.$$

2.2 Canonical form

The canonical form is slightly different to that of the standard form of an LP.

Linear program canonical form

Determine values for x_1, x_2, \dots, x_s that maximize

$$z = c_1x_1 + c_2x_2 + \dots c_sx_s$$

$$z = c_1x_1 + c_2x_2 + \cdots c_sx_s$$

subject to the constraints: for all $i \in \{1, \dots, s\}$, $x_i \geq 0$ and

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1s}x_s &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2s}x_s &= b_2, \\ \vdots & \quad \quad \quad \vdots \quad \quad \quad \vdots \\ a_{r1}x_1 + a_{r2}x_2 + \dots + a_{rs}x_s &= b_r. \end{aligned}$$

Proposition 2.4. *Every LP in standard form can be brought into canonical form. In other words, every LP has an associated LP in canonical form.*

Week 1

Proof. Since we have already convinced ourselves that every LP can be brought into standard form, it suffices to show that we can convert every LP in standard form into canonical form.

The only difference between the two forms are in the constraints; namely, we need to convert an inequality of the form

$$a_1x_1 + \dots + a_nx_n \leq b \tag{2.1}$$

to an equality. To accomplish this, we introduce *slack* variables—these are just variables with a pretentious title. They simply exist to “pick up the slack”. The “slack” is just the difference of the right hand side and the left hand side of (2.1).

Let s be a (slack) variable. Then (2.1) is equivalent to

$$\begin{aligned} s &\geq 0, \\ a_1x_1 + \dots + a_nx_n + s &= b. \end{aligned}$$

Hence, we can introduce a new variable for each inequality and obtain an LP in canonical form. \square

Example 2.5. A tailor is producing jumpers and trousers. They first need to cut the fabric and then sew it together. It takes 2 hours to cut the fabric for either a pair of trousers or a jumper. It takes 5 hours to sew a pair of trousers and 3 hours for a jumper. Scissors can be used for 8 hours per day, whereas the sewing machine can be used for 15 hours per day. If a pair of trousers is sold for €120 and a jumper for €100, how many of each should be made to maximize revenue? (Let’s ignore demand.)

Write an LP in canonical form for this scenario.

Maximize

$$z = 100J + 120T,$$

subject to $J, T, s_1, s_2 \geq 0$ and

$$2J + 2T + s_1 = 8,$$

$$3J + 5T + s_2 = 15.$$

2.3 Matrix notation

Instead of writing out all of the constraints and all the terms of the objective function, we can compactly describe the same data using matrices. Define

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \quad c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}.$$

We use the relations \leq and \geq like we do with $=$ when applied to vectors, that is, they are determined coordinate wise. For example

$$\begin{bmatrix} 1 \\ 4 \end{bmatrix} \leq \begin{bmatrix} 2 \\ 5 \end{bmatrix}, \quad \begin{bmatrix} 2 \\ 3 \end{bmatrix} \not\leq \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

In symbols, $x \leq y$ if and only if $x_i \leq y_i$ for all i .

LP standard form (matrices)

For $A \in \text{Mat}_{m \times n}(\mathbb{R})$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$, maximize

$$z = c^\top x$$

subject to $x \geq 0$ and

$$Ax \leq b.$$

Notation 2.6. The letter n is the number of variables in the objective function, and m is the number of inequalities separate from $x \geq 0$.

If it is not already clear how to convert all the previous example above into the matrix form, try to work this out yourself.

Definition 2.7. A vector $x \in \mathbb{R}^n$ satisfying all the constraints of an LP (in standard form) is a *feasible solution*.

Example 2.8. Recall Example 2.5 the “sewing problem”. The following vectors are all feasible solutions:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \pi \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

The following vectors are not feasible solutions:

$$\begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 5 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}.$$

Definition 2.9. A feasible solution that maximizes the objective function of an LP is an *optimal solution*.

Now we describe the corresponding matrix form for the canonical form of an LP. It is built *from* the standard form of an LP. We write $I_n \in \text{Mat}_n(\mathbb{R})$ for the identity matrix. For matrices $A, B \in \text{Mat}_{m \times n}(\mathbb{R})$, we set

$$[A \mid B] = \begin{bmatrix} a_{11} & \cdots & a_{1n} & b_{11} & \cdots & b_{1n} \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} & b_{m1} & \cdots & b_{mn} \end{bmatrix} \in \text{Mat}_{m \times 2n}(\mathbb{R}).$$

LP canonical form (matrices)

For $A \in \text{Mat}_{m \times n}(\mathbb{R})$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^{m+n}$, maximize

$$z = c^\top x,$$

subject to $x \geq 0$ and

$$Ax = b,$$

where $c^\top = (c_1, \dots, c_n, 0, \dots, 0)$.

Exercise 1. Show that a feasible solution for an LP in standard form induces a feasible solution in canonical form. Is the converse true?

2.4 Geometry of the feasibly set

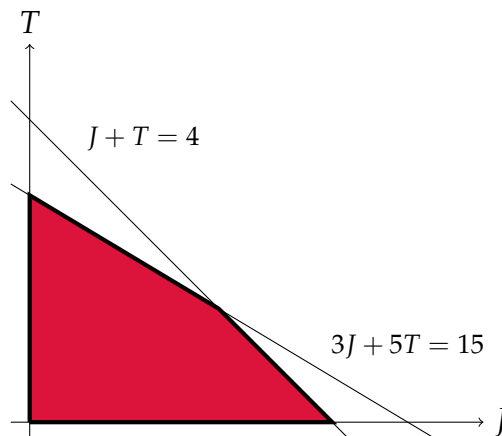
Now we begin our analysis of the set of feasible solutions to an LP. We begin by looking at the features of its geometry.

Example 2.10. Let's consider the standard form of the LP in Example 2.5. In particular, the feasible solutions are constrained by $J, T \geq 0$ and

$$2J + 2T \leq 8,$$

$$3J + 5T \leq 15.$$

We can plot the region in \mathbb{R}^2 as follows:



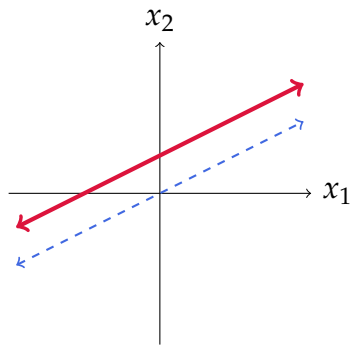
Let's consider one of our constraint inequalities:

$$a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i.$$

This can be compactly written as $a^\top x \leq b_i$ for $a \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$. The *equation*

$$a^\top x = b_i$$

defines a *hyperplane* in \mathbb{R}^n : the vector a describes the “slope” and the scalar b_i describes how far the hyperplanes shifts away from the origin.



(a) $a = (-1, 2), b = 1$.

(b) $a = (1, -1, -1), b = 1$.

References

- [1] G. B. Dantzig. Reminiscences about the origins of linear programming. *Operations Research Letters*, 1(2):43–48, 1982.
- [2] G. B. Danzig. Inductive proof of the simplex method. <https://apps.dtic.mil/sti/tr/pdf/AD0224306.pdf>, 1960.
- [3] National Science Foundation. The President’s National Medal of Science: Recipient Details. https://www.nsf.gov/od/nms/recipient_details.jsp?recipient_id=95. Accessed: 2024-09-08.