In this lecture, we consider

- Convex programming hierarchy,
- Introduction to linear programming,
- Optimization problems represented as linear programs.

1 History of linear programming

Linear programming was first devised by Kantorovich in 1939 [Kan39]. Then LP was used to model problems in military operations research duing World War II, and during that time, Dantzig was part of Project SCOOP (Scientific Computation of Optimum Programs) arranged for Pentagon [CET16]. Dantzig developed the famous "Simplex method" for solving linear programs[1]. The following quote is from his note on the method written in 1985 [Dan85].

> Origins of the Simplex Method, Summer 1947
>
> The first idea that would occur to anyone as a technique for solving a linear program, aside from the obvious one of moving through the interior of the convex set, is that of moving from one vertex to the next along edges of the polyhedral set. I discarded this idea immediately as impractical in higher dimensional spaces. It seemed intuitively obvious that there would be far too many vertices and edges to wander over in the general case for such a method to be efficient.
>
> **When Hurwicz came to visit me at the Pentagon in the summer of 1947, I told him how I had discarded this vertex-edge approach as intuitively inefficient for solving LP. I suggested Instead that we study the problem in the geometry of columns rather than the usual one of the rows – column geometry incidently was the one I had used in my Ph.D. thesis on the Neyman-Pearson Lemma. We dubbed the new method "climbing the bean pole." It looked to me efficient.**
>
> I felt sufficiently confident in this special case of what later became known as the simplex method that I proceeded to modify it so that it would work for linear programs without a convexifying row. I also developed a variant for getting a starting feasible solution called Phase I. It was then that I discovered that the method was really the previously discarded vertex-edge procedure in disguise (except for an added criterion for selecting the edge on which to move). **Apparently, in one geometry the simplex method looks efficient while in another it appeared to be very inefficient! Thus the simplex method was born in August 1947.**

The simplex method works really well in practice! However, Klee and Minty found a pathological instance for the simplex method, requiring exponential time to solve if the method is used [KM72]. Later, Khachiyan announced in 1979 that he proved that the Ellipsoid method, proposed by Naum

[1] I was not able to find a specific document announcing the result in 1947.

Z. Shor [Sho77] (also Yudin and Nemirovski [YN76]), solves linear programming in (weakly) polynomial time! [Kha79], and the full proof was published in 1980 [Kha80]. Although this result is indeed a breakthrough in theory, it is a general perception that the method is not as practical. Later, Karmarkar proposed an interior-point algorithm, which is much faster than the ellipsoid method and is proved to run in polynomial time [Kar84]. There have been far greater improvements in linear programming, both in practice and theory, and it is still one of the central research topics in optimization.

The following is a list of recent progress on fast algorithms for linear programming.

- (Khachiyan, 1980 [Kha80]) $O(n^6)$ time ellipsoid method.
- (Vaidya, FOCS1989 [Vai89]) $O(n^{2.5})$ time implementation of Karmarkar's method.
- (Cohen, Lee and Song, STOC2019 [CLS19]) $O^*(n^\omega + n^{2.5-\alpha/2} + n^{2+1/6})$ time.
- (Jiang, Song, Weinstein, and Zhang, STOC2021 [JSWZ21]) $O^*(n^\omega + n^{2.5-\alpha/2} + n^{2+1/18})$ time.

Here, n is the number of variables, O^* hides $n^{o(1)}$ factors, ω is the exponent of matrix multiplication, and α is the dual exponent of matrix multiplication. Currently, there exist algorithms that achieve $\omega \sim 2.38$ and $\alpha \sim 0.31$, and it is believed that $\omega \simeq 2$ and $\alpha \simeq 1$.

2 Linear programming standard form

Remember that our linear program

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \in \mathbb{R}^d \end{aligned} \tag{4.1}$$

has the constraint system $Ax \leq b$. We say that $Ax \leq b$ is a **system of linear inequalities**. We may also have constraints of the form $Ax = b$, and we call it a **system of linear equalities**.

2.1 Inequality constraints to equality constraints

Given a system of linear inequality constraints $Ax \leq b$, we can convert it to a set of linear equality constraints.

Lemma 4.1. *Let $a_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$. Then $a_i^\top x \leq b_i$ if and only if*

$$\exists s_i \in \mathbb{R} \text{ such that } s_i \geq 0 \text{ and } a_i^\top x + s_i = b_i.$$

Proof. First, assume that $a_i^\top x \leq b_i$. Then

$$a_i^\top x + (b_i - a_i^\top x) = b_i.$$

Hence, we may set $s_i = b_i - a_i^\top x$. Then $s_i \geq 0$ because $a_i^\top x \leq b_i$, and moreover, $a_i^\top x + s_i = b_i$ by definition. Next assume that there exists some $s_i \geq 0$ such that $a_i^\top x + s_i = b_i$. Then

$$a_i^\top x = b_i - s_i \leq b_i,$$

and therefore, $a_i^\top x \leq b_i$ holds. □

By this lemma, it follows that

$$Ax \leq b \quad \leftrightarrow \quad \exists s \in \mathbb{R}^m \text{ such that } s \geq 0 \text{ and } Ax + s = b.$$

Therefore, the optimization problem (4.1) is equivalent to

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax + s = b, \\ & s \geq 0, \\ & x \in \mathbb{R}^d, \ s \in \mathbb{R}^m. \end{aligned}$$

Hence, subject to adding some **nonnegativity constraints**, we can convert inequality constraints into equality constraints. Similarly, we can argue that

$$Ax \geq b \quad \leftrightarrow \quad \exists s \in \mathbb{R}^m \text{ such that } s \geq 0 \text{ and } Ax - s = b.$$

2.2 Free variables to nonnegative variables

In the optimization problem (4.1), the variables x may have some negative components unless the system $Ax \leq b$ contains nonnegativity constraints $x \geq 0$. In that case, we say that the variables x are **free variables**. In fact, we may come up with an equivalent optimization problem where all variables are restricted to be nonnegative.

Lemma 4.2. *Note that $x_j \in \mathbb{R}$ if and only if*

$$\exists x_j^+, x_j^- \text{ such that } x_j^+, x_j^- \geq 0 \text{ and } x_j = x_j^+ - x_j^-.$$

Proof. If $x_j \geq 0$, then we set $x_j^+ = x_j$ and $x_j^- = 0$. If $x_j < 0$, then we set $x_j^+ = 0$ and $x_j^- = -x_j$. □

This lemma implies that (4.1) is equivalent to

$$\begin{aligned} \min \quad & c^\top x^+ - c^\top x^- \\ \text{s.t.} \quad & Ax^+ - Ax^- + s = b, \\ & x^+, \ x^- \geq 0, \ s \geq 0, \\ & x^+, \ x^- \in \mathbb{R}^d, \ s \in \mathbb{R}^m. \end{aligned}$$

2.3 Standard form

The following is a linear program in **standard form**

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \\ & x \in \mathbb{R}^d. \end{aligned} \tag{4.2}$$

A linear program in standard form consists of linear equality constraints and variables that are nonnegative. Although (4.1) has inequality constraints and the variables are not necessarily nonnegative, we saw that we can convert (4.1) into a linear program in standard form.

3 Simplex Method

The simplex method due to George B. Dantzig is a practical algorithm for solving linear programming. The modern optimization software tools for linear and integer programming all implement and utilize the simplex method. We briefly touch upon the algorithm in this lecture.

Let us consider the two-variable linear program as follows. For running the simplex algorithm, it is common to introduce an extra variable z to represent the objective function.

$$\begin{aligned} \max \quad & z = 5x + 4y \\ \text{s.t.} \quad & 2x + 3y \leq 150, \\ & 2x + y \leq 70, \\ & x, y \geq 0. \end{aligned}$$

Step 1: represent the linear program in standard form. The first step is to write down the linear program in standard form. To write the two-variable LP in standard form, we use slack variables s_1 and s_2 for the constraints $2x+3y \leq 150$ and $2x+y \leq 70$, respectively. Then we obtain

$$\begin{aligned} \max_{x,y} \quad & z = 5x + 4y \\ \text{s.t.} \quad & 2x + 3y + s_1 = 150, \\ & 2x + y + s_2 = 70, \\ & x, y, s_1, s_2 \geq 0. \end{aligned}$$

Step 2: construct the initial dictionary. We keep the slack variables on the left-hand side and move the others to the right-hand side as follows.

$$\begin{aligned} z &= &&+5x &&+4y, \\ s_1 &= 150 &&-2x &&-3y, \\ s_2 &= 70 &&-2x &&-y. \end{aligned}$$

Here, the initial solution is given by setting the variables on the right-hand side to 0. In that case the values of the variables on the left-hand side are automatically chosen to satisfy the constraints. Therefore, the initial solution is given by

$$(x, y) = (0, 0), \quad (s_1, s_2) = (150, 70).$$

Then the corresponding objective value is

$$z = 5x + 4y = 0.$$

For an arbitrary linear program in standard form, we choose m distinct variables where m is the number of linear equality constraints. We keep these m variables on the left-hand side and the others on the right-hand side. We call the m variables **basic variables** and the others **non-basic variables**. To separate the basic variables from the non-basic variables, we may need a proper transformation.

Step 3: choose a better dictionary. Note that we have $z = 5x + 4y$ where non-basic variables are currently set to 0. Note that the coefficients of x and y are strictly positive. This means that increasing x or y would increase the objective value z. It looks like that the rate of increase is higher with x than y, because x has a larger objective coefficient. This is the step of

- deciding the non-basic variable to become a new basic variable (choosing the entering variable).

Let us try to increase x while the other non-basic variable, y, remains 0. Consider

$$\begin{aligned} s_1 &= 150 \quad -2x, \\ s_2 &= 70 \quad -2x. \end{aligned}$$

We still want the current basic variables to be nonnegative. While keeping both s_1 and s_2 nonnegative, we may increase x up to

$$\min\left\{\frac{150}{2}, \frac{70}{2}\right\}.$$

As the value is 35, we may increase x up to 35, in which case s_1 becomes 80 while s_2 becomes 0. As the value of s_2 is now 0, we may switch the roles of x and s_2 and move s_2 to the right-hand side. This is the step of

- deciding the basic variable to become a non-basic variable (choosing the leaving variable).

Before we move x to the left-hand side, we do some matrix row operations to eliminate x from the rows not containing variable s_2. Then we obtain

$$\begin{aligned} z + 2.5s_2 &= 175 \quad +1.5y, \\ s_1 - s_2 &= 80 \quad -2y, \\ s_2 &= 70 \quad -2x \quad -y. \end{aligned}$$

Next we move s_2 to the right-hand side and x to the left-hand side.

$$\begin{aligned} z &= 175 \quad -2.5s_2 \quad +1.5y, \\ s_1 &= 80 \quad +s_2 \quad -2y, \\ x &= 35 \quad -0.5s_2 \quad -0.5y. \end{aligned}$$

Then s_1 and x are now basic variables while s_2 and y are non-basic variables. Then we obtain a new solution given by

$$(x, y) = (35, 0), \quad (s_1, s_2) = (80, 0).$$

In this case, the objective value is

$$z = 175 - 2.5s_2 + 1.5y = 175.$$

Step 4: repeat step 3. We have $z = 175 - 2.5s_2 + 1.5y$ and y is set to 0. Then we may still have some room for improvement by increasing the value of y. Then while keeping s_2 zero, we attempt to increase y. To decide how much we increase y, consider

$$\begin{aligned} s_1 &= 80 \quad -2y, \\ x &= 35 \quad -0.5y. \end{aligned}$$

While keeping s_1 and x nonnegative, we may increase y up to 40. In this case, s_1 becomes 0 and x becomes 15. Then applying the desired row operations,

$$\begin{array}{rcll} z + 0.75s_1 &=& 235 & -1.75s_2 \\ s_1 &=& 80 & +s_2 \qquad -2y, \\ x - 0.25s_1 &=& 15 & -0.75s_2 \end{array}$$

Then moving y to the left-hand side and s_1 to the right-hand side, we obtain

$$\begin{array}{rcll} z &=& 235 & -1.75s_2 \quad -0.75s_1 \\ y &=& 40 & +0.5s_2 \quad -0.5s_1, \\ x &=& 15 & -0.75s_2 \quad +0.25s_1 \end{array}$$

Then we obtain a new solution given by

$$(x, y) = (15, 40), \quad (s_1, s_2) = (0, 0).$$

In this case, the objective value is

$$z = 235 - 1.75s_2 - 0.75s_1 = 235.$$

Step 5: obtain an optimal solution. Now, we represent the objective as

$$z = 35 - 1.75s_2 - 0.75s_1$$

where the current non-basic variables are s_1 and s_2. The coefficients of s_1 and s_2 are both negative. Then increasing the value of s_1 or that of s_2 would deteriorate the objective. In fact, this means that we are currently at an optimal solution!

4 Geometry of the Simplex Algorithm

Remember that we have taken 3 solutions until we solve the two-variable linear program.

1. $(x, y) = (0, 0)$ and $(s_1, s_2) = (150, 70)$,
2. $(x, y) = (35, 0)$ and $(s_1, s_2) = (80, 0)$,
3. $(x, y) = (15, 40)$ and $(s_1, s_2) = (0, 0)$.

Let us consider the first solution given by $(x, y) = (0, 0)$ and $(s_1, s_2) = (150, 70)$. What is the point of setting $x = y = 0$ and assigning positive values fo the slack variables s_1 and s_2? Note that s_1 being positive means $2x + 3y < 150$ and the current point (x, y) is not on the line defined by $2x + 3y = 150$. Similarly, positive s_2 means that the point (x, y) is not on the line defined by $2x + y < 70$. In fact, $(x, y) = (0, 0)$ satisfies the constraints $x \geq 0$ and $y \geq 0$ at equality, and the point is at the intersection of line $x = 0$ and line $y = 0$.

The second solution has $(x, y) = (35, 0)$ and $(s_1, s_2) = (80, 0)$. Note that the point is on the line $y = 0$. Moreover, s_1 is still positive, and we have $2x+3y = 70 < 150$. Hence, the first slack variable being strictly positive means that the constraint $2x + 3y \leq 150$ is not tight. In contrast, the other slack variable is $s_2 = 0$. Note that $2x + y = 70$, so the second constraint is satisfied at equality.

The third solution has $(x, y) = (15, 40)$ and $(s_1, s_2) = (0, 0)$. Note that both x and y are strictly positive, while the slack variables are all 0. We can check that $(x, y) = (15, 40)$ satisfies the both constraints $2x + 3y \leq 150$ and $2x + y \leq 70$ at equality. In fact the point is at the intersection of the two hyperplanes $2x + 3y = 150$ and $2x + y = 70$.

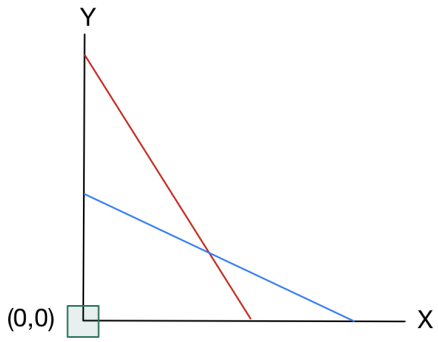

Figure 4.1: Illustrating the initial solution

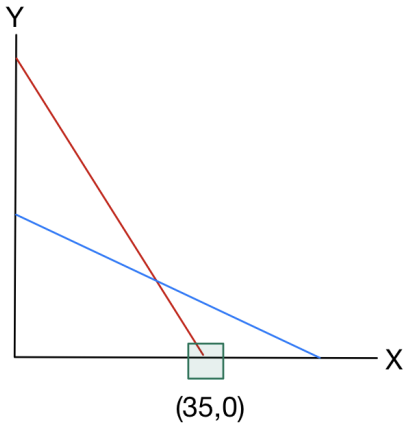

Figure 4.2: Illustrating the second solution

References

[CET16] Richard W. Cottle, B. Curtis Eaves, and Mukund N. Thapa. *Dantzig, George B. (1914–2005)*, pages 1–14. Palgrave Macmillan UK, London, 2016. 1

[CLS19] Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 938–942, New York, NY, USA, 2019. Association for Computing Machinery. 1

[Dan85] George B. Dantzig. Impact of limeiar programing on computfer dvelopment. Technical Report 85-7, Department of Operations Research, Stanford University, June 1985. 1

[JSWZ21] Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. A faster algorithm for solving general lps. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 823–832, New York, NY, USA, 2021. Association for Computing Machinery. 1

[Kan39] Leonid V. Kantorovich. The mathematical method of production planning and organization. *Management Science*, 6:363–422, 1939. 1

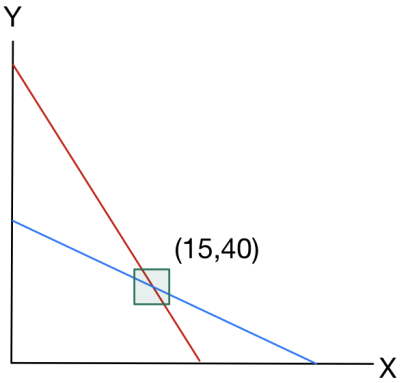

Figure 4.3: Illustrating the third solution

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, page 302–311, New York, NY, USA, 1984. Association for Computing Machinery. 1

[Kha79] Leonid.G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979. 1

[Kha80] Leonid.G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980. 1

[KM72] Victor Klee and George J. Minty. How good is the simplex algorithm? In Oved Shisha, editor, *Inequalities III (Proceedings of the Third Symposium on Inequalities held at the University of California, Los Angeles, Calif., September 1–9, 1969, dedicated to the memory of Theodore S. Motzkin)*, pages 159–175, 1972. 1

[Sho77] Naum Z. Shor. Cut-off method with space extension in convex programming problems. *Cybernetics and systems analysis*, 13(1):94–96, 1977. 1

[Vai89] P.M. Vaidya. Speeding-up linear programming using fast matrix multiplication. In *30th Annual Symposium on Foundations of Computer Science*, pages 332–337, 1989. 1

[YN76] David B. Yudin and Arkadi S. Nemirovski. Evaluation of the information com- plexity of mathematical programming problems. *Ekonomika i Matematicheskie Metody*, 12:128–142, 1976. 1