The Simplex Method

Alberto Santini

Fall 2024

Many examples in these lecture notes are adapted from popular books:

- Alexander Schrijver (1998). *Theory of linear and integer programming.* Wiley. ISBN: 0-471-98232-6.
- Vašek Chvátal (1983). *Linear Programming.* W.H. Freeman and Company. ISBN: 0-716-71195-8.
- Laurence Wolsey (2020). *Integer Programming.* 2nd Edition. Wiley. ISBN: 978-1-119-60653-6.
- Silvano Martello and Paolo Toth (1990). *Knapsack Problems: algorithms and computer implementations.* Wiley. ISBN: 978-0-471-92420-3.

In these lecture notes, we will present an algorithm to solve LPs. This algorithm, called the **simplex method**, represented an early milestone for Operational Research. It was introduced by George Dantzig in 1947; for an account of the history of this method from its author, we refer to (Dantzig 1990).

The simplex method is not the theoretically most efficient algorithm for LP. Still, it is highly effective in practice and is among the most popular ways to solve LPs on computers. In the following, we will explain how the method works using a concrete example. This example will have one striking characteristic: nothing will go *wrong*. The reader might sometimes wonder, analysing a step of the algorithm: "Well, you could do that, because so-and-so, but what if…". For clarity and ease of presentation, in fact, we will first focus on the well-behaved case. After this explanation, we will analyse the points where the algorithm could have failed and present appropriate solutions.

The general idea of this algorithm is to visit the vertices of the polytope associated with the LP. The algorithm will move from one vertex to the next, continually improving (or, at least, not worsening) the corresponding objective value. The next vertex will be chosen as a "neighbour" of the current one, i.e., one reachable by moving along an edge of the polytope. When all neighbour vertices correspond to worse objective values, the algorithm will have found the optimum. Note that limiting the search to vertices of the polytope is not a restriction (see the last exercise of the "Introduction to Linear Programming" lecture notes‘’).

1 The simplex method: an example

Let us introduce our example LP:

$$\begin{aligned} \max \quad & 5x_1 + 4x_2 + 3x_3 && (1) \\ \text{subject to} \quad & 2x_1 + 3x_2 + x_3 \leq 5 && (2) \\ & 4x_1 + x_2 + 2x_3 \leq 11 && (3) \\ & 3x_1 + 4x_2 + 2x_3 \leq 8 && (4) \end{aligned}$$

$$x_1,\ x_2,\ x_3\ \geq 0. \tag{5}$$

The first step will be to create a new **auxiliary variable** for each constraint (2)–(4):

$$\begin{aligned} x_4 &= 5 - 2x_1 - 3x_2 - x_3 && (6) \\ x_5 &= 11 - 4x_1 - x_2 - 2x_3 && (7) \\ x_6 &= 8 - 3x_1 - 4x_2 - 2x_3. && (8) \end{aligned}$$

These variables are auxiliary because their value is entirely determined by the values of the "real" variables x_1, x_2, x_3. By contrast, x_1, x_2, and x_3 are called **decision variables**. Auxiliary variables x_4, x_5, x_6 are also called **slack variables** because they hold the value of the slack of each constraint, i.e., the difference between the right-hand side and the left-hand side.

An interesting fact is that if (6)–(8) hold, then the LP can be rewritten as follows:

$$\begin{aligned} \max \quad & 5x_1 + 4x_2 + 3x_3 && (9) \\ \text{subject to} \quad & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. && (10) \end{aligned}$$

The reason is that, for example, inequality (2) is satisfied if and only if the corresponding slack variable x_4 is non-negative, i.e., $x_4 \geq 0$.

The simplex algorithm will work on the entire set of decision and slack variables. Starting from an initial feasible solution $\vec{x}$ (i.e., an assignment of values to the variables, which satisfies all constraints), the algorithm will attempt to find an improving solution $\vec{x}'$. Note that $\vec{x}'$ is improving if

$$5x_1' + 4x_2' + 3x_3' > 5x_1 + 4x_2 + 3x_3.$$

The algorithm will then replace $\vec{x}$ with the improving $\vec{x}'$, and it will try to further improve on this new solution. It will terminate when no improvement is possible.

The first step, therefore, is to find a starting feasible solution. In our example, we can simply choose $x_1 = x_2 = x_3 = 0$. These values of the decision variables, in turn, give us the values of the slack variables: $x_4 = 5$, $x_5 = 11$, $x_6 = 8$. The objective value corresponding to this solution is 0.

Next, we want to find another solution which gives a strictly better objective value. We will limit ourselves to solutions which change only one of the three decision variables while keeping the other two fixed at 0. Because x_1 has the highest coefficient in the objective function, it looks like a good candidate: an increase of 1 of x_1 increases the objective value by 5. In contrast, a corresponding increase of 1 of x_2 or x_3 would only increase the objective value by 4 or, respectively, 3. Therefore, we decide to build a new solution in which x_1 increases.

Just how much can x_1 increase, however, before one of the constraints (2)–(4) becomes violated? In other words, how much can x_1 increase before one of the slack variables x_4, x_5, x_6 becomes negative? Let us find out:

$$\begin{aligned} (6) \quad &\Rightarrow \quad & x_4 = 5 - 2x_1 \geq 0 \quad &\Rightarrow \quad & x_1 \leq \frac{5}{2} && (11) \\ (7) \quad &\Rightarrow \quad & x_5 = 11 - 4x_1 \geq 0 \quad &\Rightarrow \quad & x_1 \leq \frac{11}{4} && (12) \\ (8) \quad &\Rightarrow \quad & x_6 = 8 - 3x_1 \geq 0 \quad &\Rightarrow \quad & x_1 \leq \frac{8}{3}. && (13) \end{aligned}$$

The largest value of x_1 which satisfies all three conditions (11), (12), (13), is $x_1 = \frac{5}{2}$.

We then move to our next solution: $x_1 = \frac{5}{2}$, $x_2 = 0$, $x_3 = 0$, which corresponds to new slack variables

$$x_4 = 5 - 2\frac{5}{2} = 0 \tag{14}$$
$$x_5 = 11 - 4\frac{5}{2} = 1 \tag{15}$$
$$x_6 = 8 - 3\frac{5}{2} = \frac{1}{2}. \tag{16}$$

The objective value of this solution is $\frac{25}{2}$, which, as we wanted, is strictly better than the previous value of 0.

This is a good moment to introduce, for ease of discussion, some new terminology: we call the variables whose current value is non-zero, **basic variables**; on the other hand, the variables whose value is 0 are **non-basic variables**[1].

It is time to look back and note how easy it was to determine the largest value of x_1 that keeps the new solution feasible. This was possible because all basic variables (i.e., x_4, x_5, and x_6) were written as a function of all the non-basic variables (i.e., x_1, x_2, and x_3) through (6)–(8). In this way, as soon as we wanted to increase a variable with current value 0, we could obtain the largest feasible increase by imposing the (simultaneous) non-negativity of the basic variables. To make our life easy during the next improvement phase, then, it is convenient that we always keep a system such as (6)–(8), which expresses the current basic variables as a function of non-basic ones. At the present iteration, this means that we want to write variables x_1, x_5, and x_6 as functions of variables x_2, x_3 and x_4:

$$x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \qquad \text{by eq. (6),} \tag{17}$$
$$x_5 = 1 + 5x_2 + 0x_3 + 2x_4 \qquad \text{by eqs. (7) and (17),} \tag{18}$$
$$x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \qquad \text{by eqs. (8) and (17).} \tag{19}$$

We also rewrite the objective function in terms of the non-basic variables. This way, it will be easy to spot a variable whose value we should increase to provoke a corresponding increase in the objective function.

$$\max \quad \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \quad \text{by eqs. (1) and (17).} \tag{20}$$

The only variable with a positive coefficient in the objective function is x_3. In other words, this is the only non-basic variable whose increase will cause an improvement in the objective function.

Therefore, we set out to move to a new solution in which we increase x_3. As before, we must ask ourselves what is the largest increase of x_3 that maintains feasibility. In this case, maintaining feasibility corresponds to maintaining x_1, x_5, and x_6 non-negative. To answer the question, we use conditions (17)–(19):

$$(17) \quad \Rightarrow \quad x_1 = \frac{5}{2} - \frac{1}{2}x_3 \geq 0 \quad \Rightarrow \quad x_3 \leq 5 \tag{21}$$
$$(18) \quad \Rightarrow \quad x_5 = 1 + 0x_3 \geq 0 \quad \Rightarrow \quad \text{no restrictions} \tag{22}$$

[1] This definition will work for the moment but is slightly different from the classical definition of basic variables. We will see why in Section 2.2.

$$(19) \quad \Rightarrow \qquad x_6 = \frac{1}{2} - \frac{1}{2}x_3 \geq 0 \quad \Rightarrow \qquad x_3 \leq 1. \tag{23}$$

Note how (22) imposes no condition on x_3, while (23) imposes the most stringent of the two remaining conditions. Therefore, we move to the next solution, changing the value of x_3 from 0 to 1.

Having determined the value of three variables, $x_2 = 0$, $x_3 = 1$, $x_4 = 0$, we compute the value of the other variables, using relations (17)–(19):

$$x_1 = \frac{5}{2} - \frac{1}{2} \cdot 1 = 2 \tag{24}$$

$$x_5 = 1 + 0 \cdot 1 \ = 1 \tag{25}$$

$$x_6 = \frac{1}{2} - \frac{1}{2} \cdot 1 = 0. \tag{26}$$

The objective value of this solution is 13, which is strictly better than the previous value of $\frac{25}{2} = 12.5$. We also note that every time we make one non-basic variable basic (in this case x_3) by increasing its value, there is a basic variable which becomes non-basic (in this case x_6). This latter variable is the one which provides the tightest bound on the increase of the non-basic variable. In short, at each iteration, one variable *enters the basis* and another variable *leaves the basis*. (The basis is the set of basic variables.) We call the first the **entering variable** and the second the **leaving variable**. Swapping the leaving with the entering variable in the basis is called a **pivot**.

We are now ready for one more iteration. As before, the first thing we want to do is to write the basic variables (x_1, x_3, and x_5) as a function of non-basic ones (x_2, x_4, and x_6):

$$x_3 = 1 + x_2 + 3x_4 - 2x_6 \qquad \text{by eq. (19),} \tag{27}$$

$$x_1 = 2 - 2x_2 - 2x_4 + x_6 \qquad \text{by eqs. (17) and (27),} \tag{28}$$

$$x_5 = 1 + 5x_2 + 2x_4 + 0x_6 \qquad \text{by eq. (18).} \tag{29}$$

The objective function becomes:

$$\max \quad 13 - 3x_2 - x_4 - x_6 \quad \text{by eqs. (20) and (27).} \tag{30}$$

We now want to find a variable among non-basic x_2, x_4, and x_6, whose value we can increase to produce an increase in the objective function. But all these three variables appear with a negative coefficient in eq. (20). Therefore, increasing any of them would *decrease* the objective value. Indeed, we have reached the end of the algorithm, and our last solution was the optimal one.

To convince ourselves this is the case, note that our last solution produced an objective value of 13. Any feasible solution, moreover, must satisfy non-negativity constraints $x_2 \geq 0$, $x_4 \geq 0$, $x_6 \geq 0$. Therefore, by eq. (30), no feasible solution can have a larger objective value than 13. This means that we have found a solution whose objective value is as large as it can be: the definition of *optimal solution*.

Finally, recall that x_4, x_5, and x_6 were slack variables that we introduced, while the decision variables were x_1, x_2, and x_3. Suppose we are writing software that solves an LP using the simplex method. In that case, we want to communicate the optimal solution to our users in terms of their original decision variables. To do so, we truncate our solution and return the first three variables: $x_1 = 2$, $x_2 = 0$, $x_3 = 1$.

Exercise 1. Use the simplex method to describe *all* optimal solutions of the following LP:

$$\begin{aligned} \max \quad & 2x_1 + 3x_2 + 5x_3 + 4x_4 \\ \text{such that} \quad & x_1 + 2x_2 + 3x_3 + x_4 \leq 5 \\ & x_1 + x_2 + 2x_3 + 3x_4 \leq 3 \\ & x_1,\ x_2,\ x_3,\ x_4 \geq 0. \end{aligned}$$

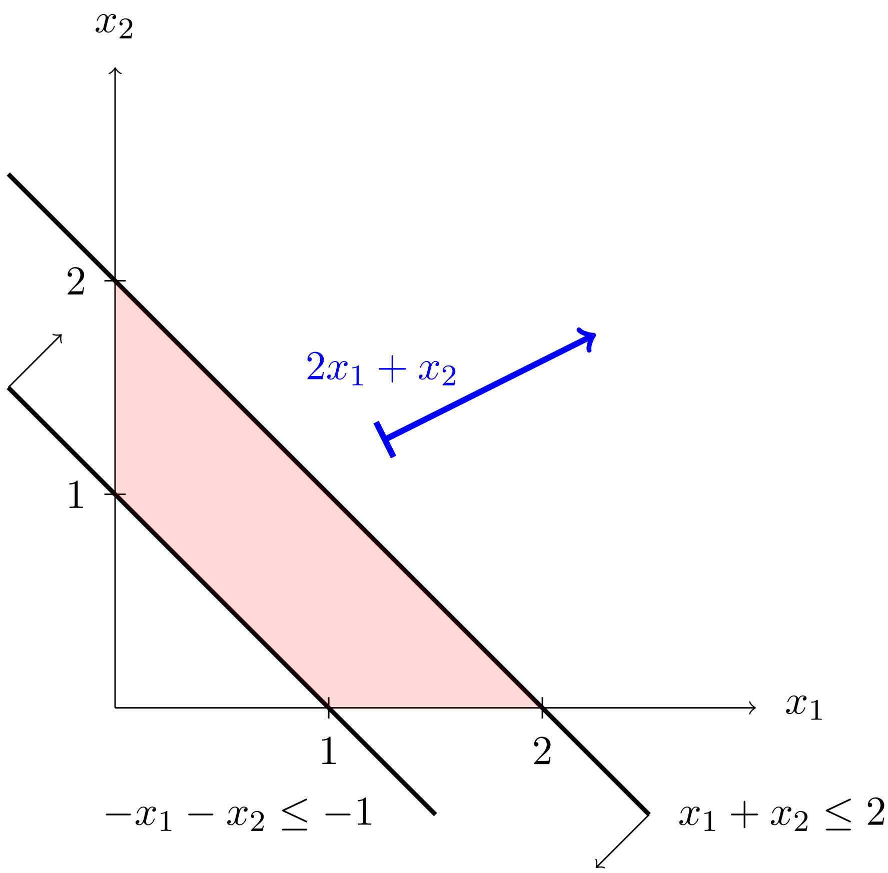
2 The simplex method: what could go wrong?

While working out the example presented in Section 1, a few things went well for us:

1. We selected the starting point of our method, the initial feasible solution, setting all decision variables to zero. We were lucky because this was indeed a feasible solution. But it must not always be the case that the point $x_j = 0$ ($j \in \{1, \ldots, n\}$) belongs to the feasible region. For example, consider the following LP:

$$\begin{aligned} \max \quad & 2x_1 - x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 2 \\ & -x_1 - x_2 \leq -1 \qquad\qquad x_1,\ x_2 \geq 0. \end{aligned}$$

   The variable assignment $x_1 = x_2 = 0$ is not feasible for this LP, as it is visible from its graphical representation.



   When we start the simplex method, we are not even sure the LP has any feasible point. How do we check that?

Question 1. Can we check if the problem is feasible, and can we always produce a feasible starting solution?

2. We could always find a non-basic variable which we could increase, the entering variable, and another basic variable which provided a bound on the increase of the entering variable, the leaving variable. For example, we increased x_1 in the first iteration. We used the rule-of-thumb (i.e., the heuristic) stating that we should choose the variable with the highest positive coefficient in the objective function.[2]

   This rule might have failed us in two ways. First, if there were no variables with a positive objective coefficient. In this case, the method would have terminated because of optimality. Second, if there is more than one non-basic variable with the same largest coefficient. In this case, how do you pick the entering variable?

   A similar situation could have happened when choosing the variable which provides the tightest bound on the increase of the entering variable. For example, when checking conditions (11)–(13), we identified x_4 as the leaving variable because it provided the tightest condition on the increase of x_1. At other times, though, we saw conditions which did not impose any bound on the increase of the entering variables, such as in (22). The rule we use could, again, fail us in two ways.

   First, if no variable provides bounds (i.e., if all conditions are analogous to (22)). In this case, we can increase the value of the entering variable as much as we want without losing feasibility. But this means we can increase the objective value as much as we want. Therefore, we can prove that the problem is unbounded and the algorithm can terminate.

   Second, if more than one variable is giving the same tightest bound on the increase of the entering variable. Again, we might wonder how to pick the leaving variable among the equally valid options.

   Question 2. How to break ties when choosing entering and leaving variables?

3. Finally, in our example, we reached a point where we could prove that no further improvement of the objective value was possible. At that moment, we could declare optimality and exit the algorithm. But is it guaranteed that the algorithm will always reach such a point (or prove that the problem is infeasible or unbounded) in a finite number of steps?

   Question 3. Can the simplex method produce an infinite sequence of solutions without ever reaching the optimal one?

In the rest of this section, we briefly go through the three above topics and provide answers to the questions raised.

2.1 Question 1: the "Phase 1 of the simplex method"

We use a neat little trick to check whether our LP is feasible and to find an initial solution. Let the LP in standard form be

$$\max \quad \sum_{j=1}^{n} c_j x_j \tag{31}$$

[2]Note: this is not necessarily the variable which guarantees the largest improvement in the objective value. Another non-basic variable could have a smaller objective coefficient, but we might be able to raise it to a larger value before violating any of the non-negativity constraints.

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij}x_j \leq b_i \qquad \forall i \in \{1,\dots,m\} \tag{32}$$
$$x_j \geq 0 \qquad \forall j \in \{1,\dots,n\}. \tag{33}$$

We transform this problem into a new LP:

$$\max \quad -x_0 \tag{34}$$
$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij}x_j - x_0 \leq b_i \qquad \forall i \in \{1,\dots,m\} \tag{35}$$
$$x_j \geq 0 \qquad \forall j \in \{0,\dots,n\}. \tag{36}$$

Note that (34)–(36) is another LP in standard form. Furthermore, this new LP is always trivially feasible. For example, one can choose $x_j = 0$ for $j \in \{1,\dots,n\}$ and then choose a sufficiently large value for x_0, in order to satisfy all constraints (35).

The "**Phase 1 of the simplex method**", then, asks to solve the new LP (34)–(36) before solving the initial LP. Note that the new variable x_0 accounts for violations of inequalities (32) in the original LP. If there is a feasible solution to the Phase-1 LP in which $x_0 = 0$, then all inequalities (32) can be satisfied in the original LP. In (34) we are maximising $-x_0$, i.e., we are minimising the value of x_0. Hence, if there is *any* solution to the Phase-1 LP in which $x_0 = 0$, this will be an optimal solution of (34)–(36). Therefore, we can solve the Phase-1 LP to establish the feasibility of the original (31)–(33).

What is more, if the original LP is feasible, then the optimal solution of the Phase-1 LP provides a feasible solution for (31)–(33): simply disregard variable x_0 and keep the value of all other variables $x_1,\dots,x_n$.

2.2 Question 2: breaking ties in the entering and leaving variables

When there is more than one possible entering or leaving variables, one might break ties arbitrarily. Doing so is a valid strategy but can lead to annoying consequences, as shown in the following example.

Consider an iteration of the simplex method with basic variables x_4, x_5, x_6 and non-basic variables x_1, x_2, x_3. The following relations, analogous to (17)–(19) and to (28)–(29), hold:

$$x_4 = 1 \qquad\qquad - 2\,x_3 \tag{37}$$
$$x_5 = 3 - 2x_1 + 4x_2 - 6\,x_3 \tag{38}$$
$$x_6 = 2 + x_1 - 3x_2 - 4x_3. \tag{39}$$

Let x_3 be the entering variable. Then all variables x_4, x_5, x_6 provide the same tight bound of $\frac{1}{2}$ on the increase of x_3. We might break the tie arbitrarily and choose x_4 as the leaving variable. The old base $\{x_1, x_2, x_3\}$ should then become, via pivoting, $\{x_1, x_2, x_4\}$. New conditions relating the basic and non-basic variables are:

$$x_3 = \frac{1}{2} \qquad\qquad - \frac{1}{2}\,x_4 \tag{40}$$
$$x_5 = \quad - 2x_1 + 4x_2 + 3\,x_4 \tag{41}$$
$$x_6 = \quad x_1 - 3x_2 + 2\,x_4. \tag{42}$$

Substituting $x_1 = x_2 = x_4 = 0$ (recall that x_1, x_2, and x_4 are non-basic), we obtain $x_3 = \frac{1}{2}$ but also $x_5 = x_6 = 0$. This violates our previous definition of basic variables as variables whose value is strictly positive: it seems like x_5 and x_6 should not be in the base. On the other hand, the pivoting operation makes exactly one variable leave the base and one enter the base. In this case, however, it looks like three columns should have left the base: x_4, x_5, and x_6.

The reason for this confusion is because we are in a special case of **degeneracy**. We call a variable **degenerate** if it is supposed to be in the base (i.e., it was in the base at the previous iteration and is *not* the leaving variable at the current iteration), but it takes value 0.

Degeneracy, by itself, is not necessarily a problem. A degenerate base implies that, at the next iteration, the tightest bound on the increase of a non-basic variable might be 0. In this case, we might execute a new pivot (one variable leaves the base, one enters it), but all variable values will stay the same and, thus, the objective value will not increase. Eventually, after a certain number of such pivots, we might find one for which the tightest bound on the increase of a non-basic variable is again strictly positive. At that iteration, the objective value will start increasing again, and it will be "business as usual": we can continue with the simplex method as we are used to. At most, we will have lost a few iterations stuck at the same objective value.

But is this "escaping" *guaranteed* to happen? If so, we can be content with our rule of arbitrarily breaking ties for the entering and leaving variables. If not, then we might get stuck in an endless loop. A topic which brings us to the next question.

2.3 Question 3: endless looping

Let us straightforwardly answer the previous question: yes, we might get stuck in an endless loop. This means that, after a certain number of iterations, we obtain a base we have already seen. For example, pivoting might produce the following bases:

$$\{x_4, x_5, x_6\} \to \{x_3, x_5, x_6\} \to \{x_1, x_3, x_6\} \to \{x_2, x_3, x_6\} \to \{x_3, x_5, x_6\} \to \cdots$$

Base $\{x_3, x_5, x_6\}$ occurs twice. After the second occurrence, pivoting following the same rule, we would again obtain base $\{x_1, x_3, x_6\}$. Continuing this way, we will obtain an endless cycle in which we keep visiting the same bases.

LPs in which this occurs for specific tie-breaking rules have been found as early as 1953. We must highlight, however, that these were examples built for the very purpose of showing that cycling is possible. In fact, there is good news for us:

1. A theorem states that the simplex method either enters an infinite loop due to ties in selecting the leaving variable or terminates in a finite number of steps. In other words, cycles induced by degeneracy are the only reason the simplex method might fail to terminate.
2. Such cycles are extremely rare in practical applications. Real-life implementations of the simplex method might simply disregard the possibility of an infinite cycle, relegating it to the realm of theoretically possible events which never happen in practice.
3. If one wants to be sure that no cycle occurs, this is also possible, at the cost of a few extra computations when breaking ties for entering and leaving variables. Two tie-breaking methods proven to prevent cycles are the *perturbation method* and the *lexicographic method*.

2.4 Summary

In short, we have seen that a practical implementation of the simplex method must take care of three fundamental aspects: initialisation, tie-breaking during pivoting, and endless cycles.

The second and third topics are tightly linked because the only way in which there can be endless cycles is if we have to break ties during pivoting. An implementer can then decide to discard the issue altogether, reassured that virtually no real-life LP will ever enter an endless cycle, or implement specific tie-breaking rules.

The issue of initialisation can be tackled by solving an auxiliary LP during the so-called "Phase 1 of the simplex method". The auxiliary LP is immune from initialisation problems because it is always feasible and admits a trivial feasible solution.