1 Linear Programming and Polyhedra

1.1 Introduction to linear programming

Linear programming captures one of the most canonical and influential constrained optimization problems. More precisely, it asks to maximize or minimize a linear objective under linear inequality and equality constraints. Below is a concrete example of a linear programming problem:

$$\begin{array}{rrcrcrcr}
\max & 6x_1 & + & 5x_2 & + & 5.5x_3 & & \\
\text{subject to} & 10x_1 & - & x_2 & - & 2.5x_3 & \geq & 11.5 \\
& -21x_1 & + & x_2 & - & 6x_3 & \geq & -104 \\
& 4.25x_1 & + & 2.75x_2 & - & x_3 & \leq & 24 \\
& & & x_2 & & & \geq & 0 \\
& 10x_1 & - & x_2 & + & 35x_3 & \geq & 49 \\
& x_1 & & & + & 2x_3 & = & 12\ ,
\end{array}$$

where the variables x_1, x_2, x_3 all take real values. Such a problem is often called a *linear program* or simply *LP*. As highlighted in the above example, the objective depends linearly on the variables, and each constraints imposes either a lower bound, an upper bound, or an equality condition on a linear form of the variables. Finally, in a linear program one can either ask to maximize or minimize the objective. Hence, formally, a general linear program is of the following form:

$$\begin{array}{rrcl}
\max / \min & c^\top x & & \\
& Ax & \leq & e \\
& Bx & \geq & f \\
& Cx & = & g\ ,
\end{array} \qquad \text{(general LP)}$$

where A, B, and C are real matrices, and c, e, f, and g are real column vectors of the appropriate dimensions.

Linear programs can be reformulated in various equivalent forms. When talking about structural results or algorithms, it is often convenient to fix one particular form of writing a linear program, like the *canonical form*, which looks as follows:

$$\begin{array}{rrcl}
\max & c^\top x & & \\
& Ax & \leq & b \\
& x & \geq & 0\ ,
\end{array} \qquad \text{(LP in canonical form)}$$

where, if $n \in \mathbb{Z}_{\geq 0}$ is the number of variables and $m \in \mathbb{Z}_{\geq 0}$ the number of constraints, we have $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and the variables x take values in $\mathbb{R}^n$. Indeed, any linear program can be transformed into canonical form as follows:

(i) Any constraint of type $a^\top x \geq \beta$ can be rewritten as $(-a)^\top x \leq -\beta$.

(ii) Any constraint of type $a^\top x = \beta$ can be replaced by a pair of equivalent constraints $a^\top x \leq \beta$ and $(-a)^\top x \leq -\beta$.
(iii) Any variable x_i that is not bound by a non-negativity constraint $x_i \geq 0$ can be replaced as follows. Introduce two new variables x_i^+ and x_i^- with non-negativity constraints $x_i^+ \geq 0$ and $x_i^- \geq 0$, and replace all occurrences of x_i by $x_i^+ - x_i^-$.
(iv) If the problem is a minimization problem, then replace the objective $\min c^\top x$ by $\max -c^\top x$. This will flip the sign of all solution values. However, maximizers of the new problem are minimizers of the original problem and vice versa.

Notice that linear programming can also be equivalently described as the task to maximize a linear function over a polyhedron, which is the intersection of finitely many half-spaces in finite-dimensional Euclidean space.

1.1.1 Different types of LPs and goal of LP algorithms

Linear programs are often divided into the following three types, depending on their optimal value:

LP with finite optimum An LP with a finite optimal value. The optimum is either attained at a unique point or the LP may have multiple optimal solutions.

Unbounded LP An LP with feasible solutions of arbitrarily large value in case of a maximization problem, or of arbitrarily small value in case of a minimization problem.

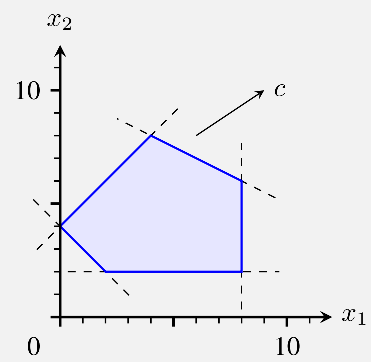
Infeasible LP An LP without any feasible solutions. If it is a maximization problem, its optimal value is often set to $-\infty$ by convention, and to ∞ for minimization problems.

In the three examples below, we exemplify the above notions on LPs in 2 dimensions.

Example 1.1: LP with finite optimum

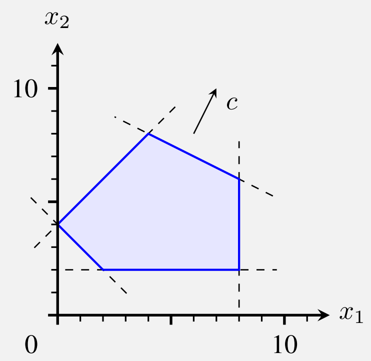
Below is an LP with a unique optimum, attained at the point $(x_1, x_2) = (8, 6)$.

$$
\begin{array}{lrcrcr}
\max & 3x_1 & + & 2x_2 & & \\
 & -x_1 & + & x_2 & \leq & 4 \\
 & x_1 & + & 2x_2 & \leq & 20 \\
 & x_1 & & & \leq & 8 \\
 & & & x_2 & \geq & 2 \\
 & x_1 & + & x_2 & \geq & 4
\end{array}
$$



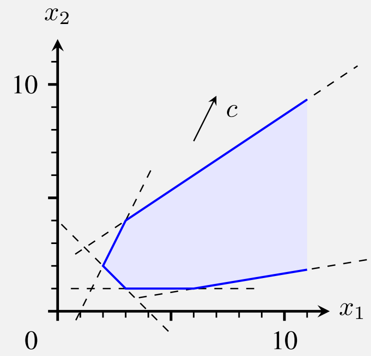
Taking the same example as above with a different objective function that is perpendicular to one of the sides of the polygon describing the feasible region, an LP with multiple optimal solutions is obtained.

$$\begin{array}{rrrrl} \max & x_1 & + & 2x_2 & \\ & -x_1 & + & x_2 & \leq 4 \\ & x_1 & + & 2x_2 & \leq 20 \\ & x_1 & & & \leq 8 \\ & & & x_2 & \geq 2 \\ & x_1 & + & x_2 & \geq 4 \end{array}$$



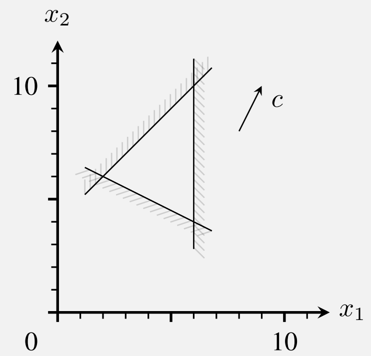
Example 1.2: Unbounded LP

$$\begin{array}{rrrrl} \max & x_1 & + & 2x_2 & \\ & x_1 & - & 6x_2 & \leq 0 \\ & & & x_2 & \geq 1 \\ & x_1 & + & x_2 & \geq 4 \\ & 2x_1 & - & x_2 & \geq 2 \\ & -2x_1 & + & 3x_2 & \leq 6 \end{array}$$



Example 1.3: Infeasible LP

$$\begin{array}{rrrrl} \max & x_1 & + & 2x_2 & \\ & -x_1 & + & x_2 & \geq 4 \\ & x_1 & + & 2x_2 & \leq 14 \\ & x_1 & & & \geq 6 \end{array}$$



When talking about an algorithm that solves linear programs, we expect the algorithm to provide answers to the following questions. First, the algorithm should detect the type of LP we are dealing with, i.e., whether it is an LP with finite optimum, an unbounded LP, or an infeasible one. Normally, in case of an LP with finite optimum, one does not require an LP algorithm to distinguish between whether there is a unique optimal solution or multiple ones. Second, the algorithm should return the following.

- If the LP has a finite optimum, the algorithm should return an optimal solution.
- If the LP is unbounded, the algorithm should return a half-line pointing in an improving unbounded direction. More precisely, assuming that we want to maximize the objective c, this consists of a feasible point $y \in \mathbb{R}^n$ and a non-zero vector $v \in \mathbb{R}^n$ such that
  (i) $y + \lambda v$ is feasible for any $\lambda \in \mathbb{R}_{\geq 0}$, and
  (ii) $c^\top v > 0$.

Hence, in the first problem of Example 1.1, an LP algorithm must return the unique optimal solution $\binom{8}{6}$, whereas in the second LP of the same example, any point on the segment connecting $\binom{8}{6}$ and $\binom{4}{8}$ can be returned. For the unbounded LP shown in Example 1.2, an LP algorithm needs to return a strictly improving half-line, for example $\binom{5}{2} + \lambda \cdot \binom{3}{1}$ for $\lambda \in \mathbb{R}_{\geq 0}$.

Whereas the above highlights minimal requirements that we expect LP algorithms to fulfill, sometimes, additional information is desired, as for example:

- In case of an LP with finite optimum such that there is a corner of the feasible region at which the optimum is attained, we may want that the LP algorithm returns an optimal solution that is a corner of the feasible region.
- In case of an LP with finite optimum, we may want a certificate of optimality.
- In case of an infeasible LP, we may want to obtain a certificate of infeasibiliy.

Informally speaking, a certificate of optimality or infeasibility is a piece of information with which one can check quickly that a certain solution is optimal or a certain LP is infeasible, respectively. We formally provide such certificates later on. The Simplex Method, which is an algorithm to solve linear programs, together with the concept of linear duality that we introduce later, allows for obtaining the above-mentioned additional information.

1.1.2 Example applications

Before we talk about structural and algorithmic results related to linear programming, we present some of its numerous, and sometimes surprising, applications.

Production planning

Linear programs, and variations and generalizations thereof, are often used in Operations Research problems coming from industrial applications. In this example, we consider an extreme simplification of such a real-world problem, to provide a glimpse of why linear programs can be relevant in such contexts, and to introduce some terminology that is prevalent in Operations Research when dealing with linear programs.

To this end, consider the following heavily simplified decision problem on how to run a coal power station. The maximum emissions allowed by law are given in Table 1.1.

Maximum amount of sulfur dioxide (SO_2)	30 mg/m^3 air
Maximum amount of smoke	12 kg/h

Table 1.1: Maximum emissions allowed by law.

The coal is being transported to the station with train wagons and stocked there. A transport system brings the coal to the mill, where it is ground to powder. Then it is pumped into the furnace. The furnace heat drives steam-powered turbines that generate electrical power.

Two different coal qualities A and B are available with different emissions, shown in Table 1.2.

coal	sulfur dioxide in exhaust gas [when using only indicated coal type in furnace]	smoke per ton [emissions per ton of burnt coal]
A	18 mg/m^3	0.5 kg
B	38 mg/m^3	1.0 kg

Table 1.2: Emissions of the two types of coals.

The heating value of the coal, measured in kilograms of steam per ton of coal, is 24,000 kg/t for coal A and 20,000 kg/t for coal B. Moreover, the mill can grind 16 t/h of coal A and 24 t/h of coal B. The transport system can transport 20 t/h, independently of the type of coal. The task is to determine the maximum performance of the power station while satisfying the maximum emissions allowances. To answer this question, we set up a linear program.

Decision variables In a steady state mode, the power plant will burn a certain amount of coal of each type per hour. Because these are the parameters that can be controlled in this decision problem, the following two variables, which correspond to these "free" parameters, are called *decision variables*:

- $x_1 \in \mathbb{R}_{\geq 0}$: amount of coal A being burnt per hour (in t/h), and
- $x_2 \in \mathbb{R}_{\geq 0}$: amount of coal B being burnt per hour (in t/h).

Objective function We first observe that to maximize the energy output, one has to maximize the steam output per time unit. Therefore we can reformulate the objective function to "find the composition of coal that maximizes the steam output per time unit". Because each ton of A and B burnt in the furnace produces 24 and 20 tons of steam per hour, respectively, the objective is to maximize the following linear form:

$$\underbrace{24\,x_1 + 20\,x_2}_{\text{objective function}} = \underbrace{z}_{\substack{\text{value of the}\\ \text{objective function}}} .$$

Hence, in the two-dimensional (x_1, x_2)-space, the level curve for every fixed value of z is a line.

Constraints In addition to the non-negativity requirements, the variables are also constrained by further physical, economic, or legal conditions.

Smoke emission constraints. The maximum smoke emission allowed is 12 kg/h. We know that coal A produces 0.5 kg smoke per ton of coal and coal B produces 1 kg of smoke per ton of coal.

When x_1 tons of coal A and x_2 tons of coal B are being burnt, we have

$$0.5\,x_1 \;+\; 1\,x_2$$

kilograms of smoke emitted per hour. This value must not exceed 12 kg/h, i.e., the following inequality guarantees the compliance with the smoke emission limit:

$$\begin{array}{lcl}\underbrace{0.5x_1 \quad + \quad 1x_2}_{\text{coefficients of the ``left-hand side (lhs)''}} & \le & \underbrace{12}_{\text{``right-hand side (rhs)''}}\ .\end{array}$$

Transport capacities. The transport system that brings the coal to the mill has a capacity of 20 t/h. The total amount of coal being transported per hour is the sum of the decision variables, i.e.,

$$x_1 + x_2 \le 20$$

describes the transport capacity constraint.

Mill performance. The mill can grind 16 t/h of coal A or 24 t/h of coal B. In other words, $1/16$ h is needed to grind one ton of coal A and $1/24$ h to grind one ton of coal B. Therefore, if x_1 tons of coal A and x_2 tons of coal B are processed, it will take the mill

$$\frac{1}{16}x_1 + \frac{1}{24}x_2 \quad \text{hours.}$$

Thus we get the following inequality for the mill performance constraint:

$$\frac{1}{16}x_1 + \frac{1}{24}x_2 \le 1\ .$$

Expressing the maximum mill performance in time units per ton, we were able to link them with the time constraints.

Sulfur emission. The maximum sulfur dioxide (SO_2) emission must not exceed 30 mg/m^3. When burning both types of coal mixed, we may assume that the mix is homogeneous, i.e., the fraction of coal A is $\frac{x_1}{x_1+x_2}$ and, accordingly, the fraction of coal B is $\frac{x_2}{x_1+x_2}$. The SO_2 emission can be computed using the weighted average of both SO_2 emissions, leading to the following emission constraint:

$$18 \cdot \frac{x_1}{x_1+x_2} + 38 \cdot \frac{x_2}{x_1+x_2} \le 30\ .$$

Even though this constraint is not linear, it can be linearized by multiplying both sides by x_1+x_2. This leads to the linear constraint

$$12x_1 - 8x_2 \ge 0\ ,$$

which ensures that the SO_2 emissions respect the given limitation.

Mathematical formulation Summarizing the above discussion, we obtain the following linear program, which is a mathematical model of the described problem.

$$
\begin{array}{lrcrcrl}
\max & 24x_1 & + & 20x_2 & & & \\
 & \frac{1}{2}x_1 & + & x_2 & \leq & 12 & \text{(smoke emission)} \\
 & x_1 & + & x_2 & \leq & 20 & \text{(transport capacity)} \\
 & \frac{1}{16}x_1 & + & \frac{1}{24}x_2 & \leq & 1 & \text{(mill performance)} \\
 & 12x_1 & - & 8x_2 & \geq & 0 & \text{(sulfur dioxide emission)} \\
 & x_1 & & & \geq & 0 & \text{(non-negativity)} \\
 & & & x_2 & \geq & 0 & \text{(non-negativity),}
\end{array}
$$

or, in matrix notation in canonical form, $\max\ c^\top x$ subject to $Ax \leq b$ and $x \geq 0$, where

$$
c = \begin{pmatrix} 24 \\ 20 \end{pmatrix}, \quad A = \begin{pmatrix} \frac{1}{2} & 1 \\ 1 & 1 \\ \frac{1}{16} & \frac{1}{24} \\ -12 & 8 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \text{and} \quad b = \begin{pmatrix} 12 \\ 20 \\ 1 \\ 0 \end{pmatrix}.
$$

Note that, to obtain canonical form, we multiplied the sulfur dioxide emission constraint by -1 to convert the inequality type from "$\geq$" into "$\leq$".

Graphical solution method As our example is defined in two variables, every constraint can be graphically represented in the (x_1, x_2)-space (see Figure 1.1). In order for a combination of activities to be *feasible*, it has to satisfy all constraints *simultaneously*, including the non-negativity constraints. We will call such points *feasible solutions* and the set of all feasible solutions is called *feasible region*. In our example, the feasible region is the shaded area in Figure 1.1.

If removing a constraint does not change the feasible region, then this constraint is called *redundant*. Thus, removing redundant constraints does not change the problem. In our example, the transport capacity constraint is redundant (see Figure 1.1). Whereas removing redundant constraints may sometimes lead to a desirable simplification of the problem, finding and eliminating them in large LPs may take considerable time, and is therefore impractical. Moreover, when considering slight modifications of the entries of an LP—which is for example done in sensitivity analysis—it may be that a redundant constraint becomes non-redundant. In any case, the approaches we will study to solve LPs work irrespectively of the presence of redundant constraints.

When superposing the level curves of the objective function with the feasible region, we immediately obtain a graphical solution method. Moving the level curves in parallel with increasing value as far as possible while still intersecting the feasible region, we can read off the optimal value and solution from the graphical illustration (see Figure 1.2): $z = 408$ is the maximum value of the objective function, which is achieved at the corner A of the feasible region, determined by $x_1 = 12$ and $x_2 = 6$. A combination of 12 tons of coal A and 6 tons of coal B per hour maximizes the energy output under the constraints discussed above.

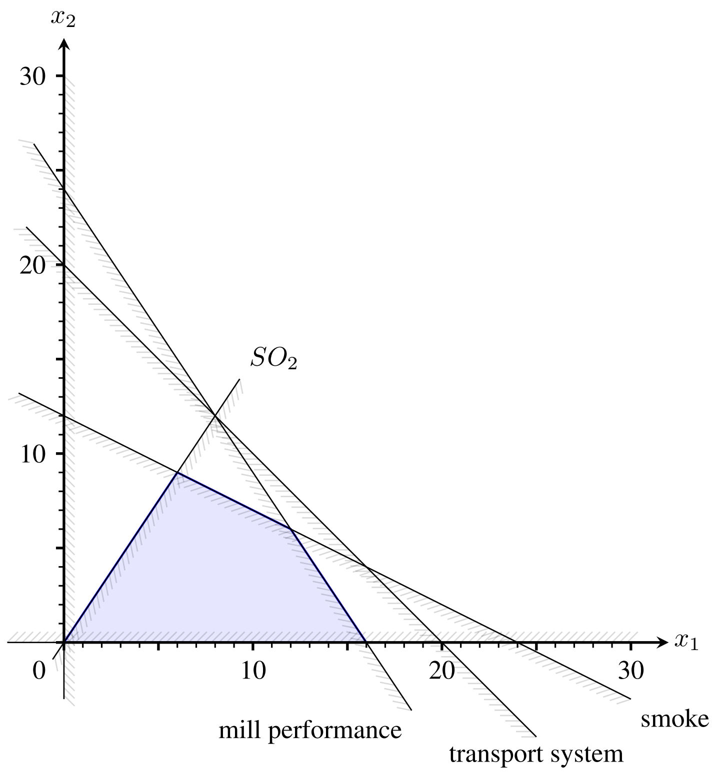


Figure 1.1: Constraints of the coal example.

ℓ_1 regression

In statistics, regression analysis is used to estimate the relationships between different variables. In this section, we consider linear ℓ_1 regression. Suppose we have a set of input points $x_1, \dots, x_k \in \mathbb{R}^n$, and for each input point x_i we have a response variable $y_i \in \mathbb{R}$ that we observed. In linear regression, we consider a model where we assume that the response variables are an affine transformation of the input points plus some random error. This leads to the question of finding an affine function $f(x) := a^\top x + \beta$, where $a \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$ such that $f(x_1), \dots, f(x_k)$ is a good approximation of $y_1, \dots y_k$. To quantify how good the approximation is, ℓ_1 regression uses the ℓ_1 norm (contrary to the more common square error, which uses the ℓ_2 norm), leading to the following optimization problem

$$\min \left\{ \sum_{i=1}^{k} |y_i - a^\top x_i - \beta| \colon a \in \mathbb{R}^n, \beta \in \mathbb{R} \right\} . \tag{1.1}$$

Notice that because the response variables are scalars, the ℓ_1 norm of the error $y_i - a^\top x_i - \beta$ is simply its absolute value. More generally, we could have response variables in a higher-dimensional space $\mathbb{R}^m$; the discussion here extends to this generalization in a straightforward way.

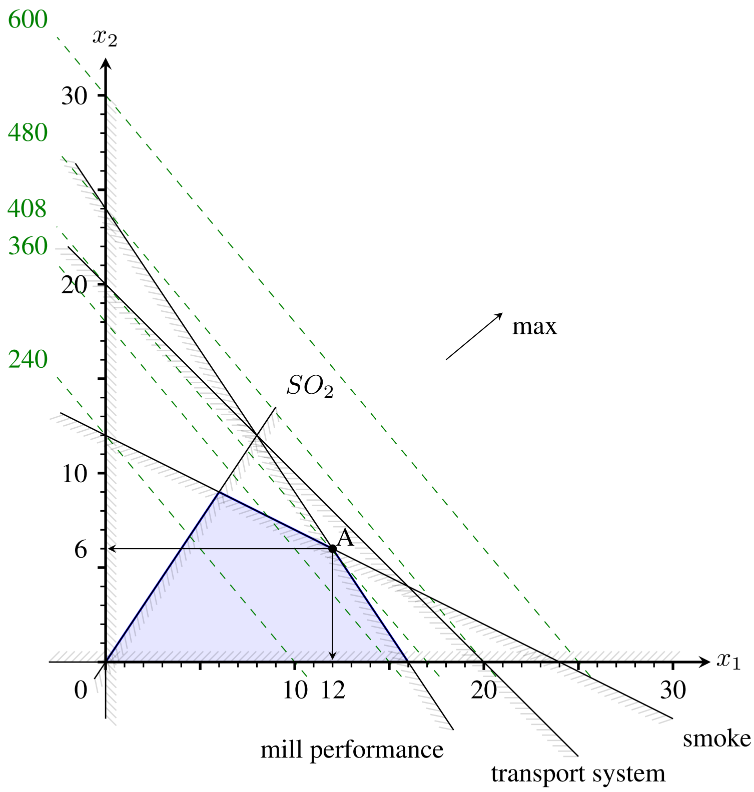


Figure 1.2: Graphical solution method. The blue dashed lines show level curves, which are perpendicular to the direction in which we maximize. Among all level curves touching the feasible region, the one corresponding to the largest objective value reveals the optimal value of the LP.

Problem (1.1) is not a linear program in its current form because the absolute value is not an affine function. However, it can be converted into one by introducing, for $i \in [k]$, a variable z_i that captures the ℓ_1 error $|y_i - a^\top x_i - \beta|$:

$$
\begin{array}{lrcrcrcll}
\min & \sum_{i=1}^{k} z_i \\
& y_i & - & a^\top x_i & - & \beta & \leq & z_i & \forall i \in [k] \\
& -y_i & + & a^\top x_i & + & \beta & \leq & z_i & \forall i \in [k] \\
& & & & & a & \in & \mathbb{R}^n \\
& & & & & \beta & \in & \mathbb{R} \\
& & & & & z & \in & \mathbb{R}^n \ .
\end{array}
\tag{1.2}
$$

Notice that the constraints imposed on the variables z_i only require $z_i \geq |y_i - a^\top x_i - \beta|$ and not the ostensibly more natural $z_i = |y_i - a^\top x_i - \beta|$. The reason is that requiring $z_i = |y_i - a^\top x_i - \beta|$

would lead to a non-convex set of feasible solutions; however, linear programs always optimize over a polyhedron, which is convex. Nevertheless, because we minimize the sum of the z_i, the fact that the z_i will be at least as large as the absolute value is sufficient. They will automatically satisfy $z_i = |y_i - a^\top x_i - \beta|$ in an optimal solution to the LP, because setting them to a larger value would just lead to a strictly worse objective value. Hence, finding the optimal coefficients in linear ℓ_1 regression can be reduced to solving a linear program.

Figure 1.3 shows an example of linear ℓ_1 regression with unidimensional input points, i.e., $n = 1$, applied to a small random data set.

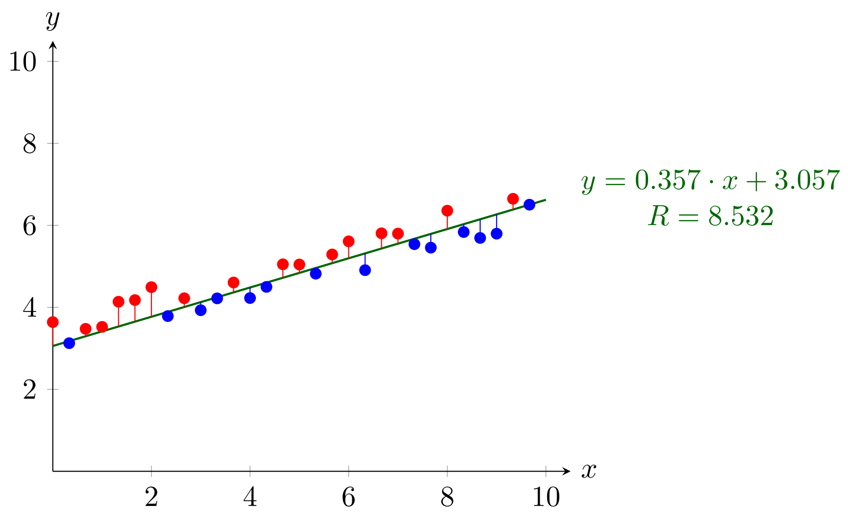


Figure 1.3: Example with 30 pairs $(x_i, y_i) \in \mathbb{R}^2$. The green line is the optimal linear ℓ_1 regression computed by solving the corresponding linear program (1.2). The sum of the vertical distances between the points and the line is the total error R that got minimized.

Resource allocation

Resource allocation is one of the classical problem settings in Operations Research. The goal is to distribute scarce resources among different activities, commonly with the objective of maximizing profit.

To exemplify this problem class, consider a fictitious Italian coffee-shop that wants to edge the competition by brewing its own special coffee types. They are doing this by combining different sorts of coffee beans into a single type. Moreover, they know how many beans of which sort they need to produce one liter of a certain coffee type. Now they need to figure out how much to produce of each coffee type (resource allocation) to maximize profit. Assume that they have fixed amounts of coffee beans of each sort (scarce resource) to be used for the next brewing cycle. Moreover, the coffee types need very different compositions of beans and each type gives a different profit per liter. This is shown in Table 1.3, where we assume that there are four bean sorts (Liberica, Excelsa, Arabica, and Robusta), and three coffee types (Prego, Barzo, Pronto).

Bean sort	Available amount	Prego	Barzo	Pronto
Liberica	300 000	100	100	300
Excelsa	350 000	200	300	100
Arabica	250 000	100	100	250
Robusta	500 000	300	200	200
profit/liter	-	10	10	20

Table 1.3: Beans needed per liter of each coffee type.

To approach this problem with linear programming, we first introduce a variable for each coffee type, capturing the amount of coffee to be brewed of the corresponding type:

$$c_j := \text{production quantity of coffee type } j \text{ (in liters)} \qquad j \in \{1, 2, 3\} \ ,$$

where $j = 1$ corresponds to coffee type Prego, $j = 2$ to Barzo, and $j = 3$ to Pronto.

Now we introduce one constraint per bean type to ensure that the total number of beans needed of each type is not exceeded by the production plan.

$$\begin{array}{lrcrcrcl} \text{Liberica:} & 100c_1 & + & 100c_2 & + & 300c_3 & \leq & 300\,000 \\ \text{Excelsa:} & 200c_1 & + & 300c_2 & + & 100c_3 & \leq & 350\,000 \\ \text{Arabica:} & 100c_1 & + & 100c_2 & + & 250c_3 & \leq & 250\,000 \\ \text{Robusta:} & 300c_1 & + & 200c_2 & + & 200c_3 & \leq & 500\,000 \end{array}$$

Finally, we have non-negativity constraints:

$$c_j \geq 0 \quad \forall j \in \{1, 2, 3\} \ .$$

The objective is to maximize the profit:

$$\max \ 10c_1 + 10c_2 + 20c_3 \ .$$

Hence, this completes the description of the problem in form of a linear program. Notice that, to keep things as simple as possible, we assumed that unused beans do not have any value. However, we could also assume that there is a salvage value per bean type for unused beans, and this could still be modeled as a linear program.

Project scheduling

Consider a project comprising of a finite set of n tasks, where task i needs t_i time for completion. Additionally, some tasks depend on others and cannot be started before all the prerequisite tasks are finished. Our goal is to minimize the makespan, i.e., the total time required to finish all tasks and, thus, the whole project. Here we assume that any number of tasks can be processed simultaneously without compromising performance or quality.

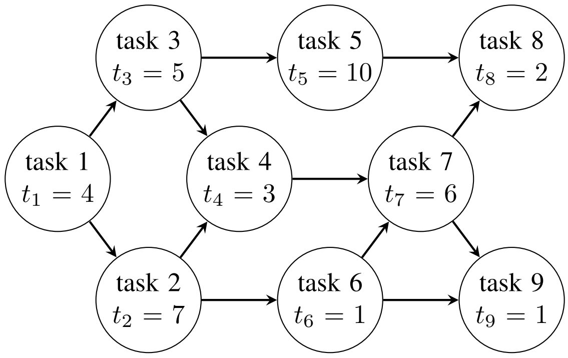


Figure 1.4: Example project scheduling problem.

Figure 1.4 shows an example project with 9 tasks and the dependencies between the tasks. For example, the arrow from task 3 to task 4 indicates that task 4 can only be started once task 3 has been completed.

We now show how this project scheduling question can be modeled by a linear program. Assuming that work on the projects starts at time 0, denote by s_v the starting time of task v. Moreover, let z be a variable we introduce to denote the completion time of the whole project, which is the completion time of the last task. We set up the following linear program.

$$\begin{array}{rrcrcll}
\min & z & & & & & \\
 & z & \geq & s_v & + & t_v & \text{for every task } v \\
 & s_v & \geq & s_u & + & t_u & \text{for every dependency } u \text{ of } v \\
 & s_v & \geq & 0 & & & \text{for every task } v \\
 & z & \geq & 0 & & &
\end{array}$$

Let us check that solving (1.1.2) indeed yields an optimal task schedule (minimizing the makespan). On the one hand, the optimal solution of the LP is a scheduling of the tasks satisfying the precedence constraints, so the optimal value is greater or equal to the minimal makespan. On the other hand, every optimal schedule satisfies the constraints of the LP, hence it is a feasible solution, and thus the optimal objective value is less or equal to the minimal makespan. Therefore, the optimal solution of the LP is precisely the minimal makespan of the whole project. Note that by backtracking from the end of the project and following tight constraints, one can construct the “bottleneck” chain of tasks that determine the project completion time (see Figure 1.5). This reasoning leads to the following interesting property: The minimum makespan is the same as the length of a longest path in the task graph, as shown in Figure 1.4, where the length of a path is the sum of the processing times of the tasks on the path.

Shortest s-t path

The problem of finding a shortest path from one site to another frequently occurs in various contexts, such as planning the quickest route from one physical location to another (from point A to point B), finding the shortest chain of social connections between two people to test the six

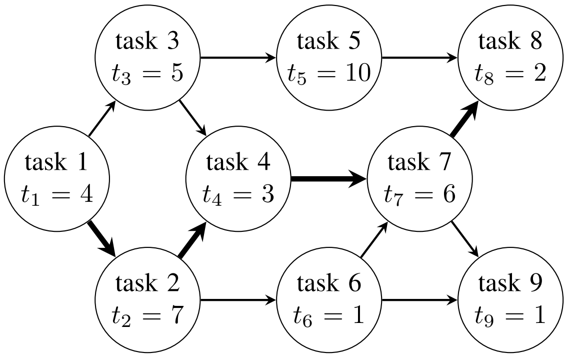


Figure 1.5: The highlighted path shows a bottleneck chain of tasks. This is a longest path in the above graph, where the length of the path is measured by summing up the processing times of the tasks on the path. Hence, in the shown example, the highlighted path has total length 22, which is the optimal makespan.

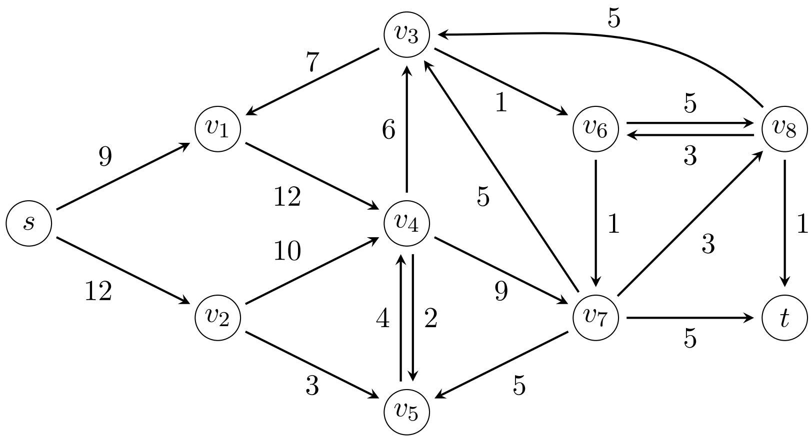


Figure 1.6: Example of finding a shortest s-t path.

degrees of separation hypothesis, or solving the Rubik's Cube in the least number of moves. In this section, we show that the shortest path problem can be reformulated in terms of an LP.

To formalize the setting of the shortest path problem, we are given a graph as shown in Figure 1.6, where a finite number of nodes is connected by arcs. Each arc a has a non-negative length $\ell(a) \in \mathbb{R}_{\geq 0}$ assigned to it; these are the numbers written next to the arcs in Figure 1.6. Moreover, we have a designated start node s and end node t. The goal is to find a shortest way to go from s to t, where we can only follow arcs from tail to head. (We formally introduce graph theoretic notions and terminology later in this course.)

To reformulate the shortest s-t path problem as an LP, we introduce a non-negative variable d_v for each node v with which we want to capture the distance from s to v. In particular, the distance from s to itself is zero, i.e., we can set $d_s = 0$. Another example distance, for the shown problem, is the distance from s to v_5, which is 15, because the shortest path from s to v_5 is $s \to v_2 \to v_5$ of length 15. Notice that the following is a valid constraint for the distances: for any arc a from u to v, the distance from s to v is no greater than the distance from s to u plus the length of the arc a. Indeed, to reach u from s, one option is to first visit v and then go from v to u over the arc a. Due to this, we will impose the constraints

$$d_v \leq d_u + \ell(a) \quad \text{for every arc } a \text{ from } u \text{ to } v,$$

on the variables d_v. In turns out that these constraints ensure that any d_v is never strictly larger then the distance from s to v. However, the discussed constraints allow for choosing distances that are significantly shorter. For example, $d_v = 0$ for every node v is a feasible solution. Hence, our linear program will actually not minimize d_t, but maximize it, which might sound counter-intuitive at first sight. In summary, below is the LP formulation we are using.

$$\begin{array}{llll} \max & d_t & & \\ & d_s = & 0 & \\ & d_v \leq & d_u + \ell(a) & \text{for every arc } a \text{ from } u \text{ to } v \\ & d_v \geq & 0 & \text{for every node } v \end{array}$$

Let us justify why the optimal value of the above LP is indeed the shortest path distance from s to t. First, for any path P, we can sum up all the constraints corresponding to arcs on the path to obtain the inequality $d_t \leq d_s + \ell(P) = \ell(P)$. Thus the maximum value of d_t we obtain through the LP is not greater than the shortest path length. Moreover, by setting d_v, for each node v, to the actual distances from s to v, all of our constraints are fulfilled. Hence, d_t is not smaller than the shortest path length from s to t, implying that the optimal LP value is indeed the distance from s to t.

Additionally, a shortest s-t path can be inferred from the optimal solution of the LP: starting from t, on each step choose an arc to the current node for which the respective constraint is tight and transition to the beginning of that arc. Figure 1.7 indicates in green next to each node v the value d_v. This is an optimal solution to the above LP. (Notice there are further optimal solutions; for example one can exchange $d_{v_1} = 9$ by $d_{v_1} = 8$. In particular, this shows that there may be vertices $v \in V \setminus \{s,t\}$ for which, even in an optimal LP solution, d_v is strictly smaller than the s-v distance.)

1.2 Polyhedra and basic convex geometry

As discussed, linear programming is about maximizing or minimizing a linear (or affine) function over a polyhedron. Not surprisingly, the study of polyhedra is key in understanding and solving linear programs. Moreover, as we will see later, polyhedra also play a central role in Combinatorial Optimization.

1.2.1 Basic notions

We start with some basic notions and terminology.

Definition 1.4: Half-space & hyperplane

A *half-space* in $\mathbb{R}^n$ is a set of the form $\{x \in \mathbb{R}^n : a^\top x \leq \beta\}$ for $a \in \mathbb{R}^n \setminus \{0\}$ and $\beta \in \mathbb{R}$. Moreover, $\{x \in \mathbb{R}^n : a^\top x = \beta\}$ is called a *hyperplane*.

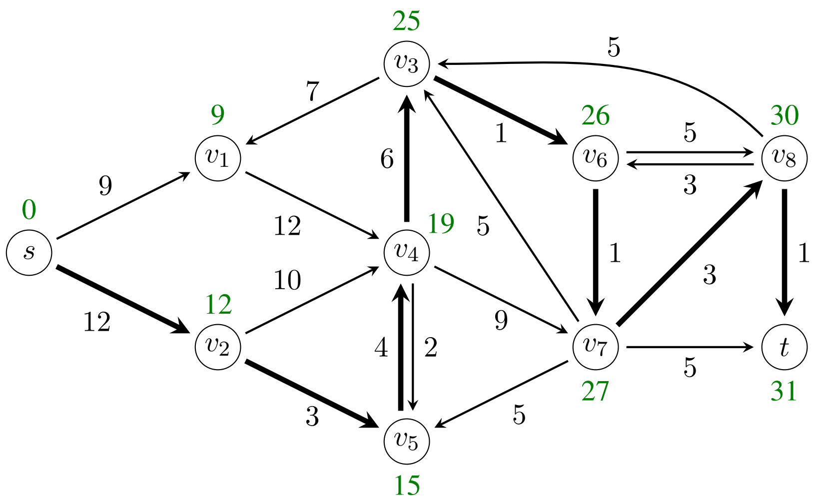


Figure 1.7: The green numbers show the d_v-values of one optimal LP solution. A shortest s-t path can be found by starting at t and backtracking along arcs that correspond to tight LP constraints.

Definition 1.5: Polyhedron & polytope

A *polyhedron* $P \subseteq \mathbb{R}^n$ is a finite intersection of half-spaces. Moreover, a bounded polyhedron is called a *polytope*.

An immediate consequence is that a polyhedron can be described by a finite set of linear inequalities, which is generally known as an *inequality description*. In other words, for a polyhedron $P \subseteq \mathbb{R}^n$, we can always write $P = \{x \in \mathbb{R}^n \colon Ax \leq b\}$ for some $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ with $m \in \mathbb{Z}_{\geq 0}$. Even though, as we already discussed in the context of linear programs, we can always describe P in terms of less-than-or-equal inequalities, we allow an inequality description of a polyhedron to have inequalities of both types and also equality constraints. Clearly, there are many equivalent inequality descriptions for the same polyhedron.

Notice that polyhedra are convex sets.[1] This follows immediately from the fact that half-spaces are convex and the intersection of any family of convex sets is convex.

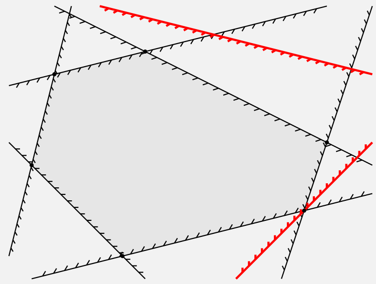
Definition 1.6: Redundancy

A linear inequality or equality of an inequality description of a polyhedron is called *redundant* if removing it from the description does not change the polyhedron.

[1] We recall that a set $X \subseteq \mathbb{R}^n$ is convex if, for any $x_1, x_2 \in X$ and $\lambda \in [0,1]$, we have $\lambda x_1 + (1-\lambda)x_2 \in X$.

Example 1.7: Redundant constraints

The picture below shows a polytope with two redundant constraints, highlighted in red.



Definition 1.8: Dimension of a polyhedron

The dimension $\dim(P)$ of a polyhedron $P \subseteq \mathbb{R}^n$ is the dimension of a smallest-dimensional affine subspace containing P, i.e.,

$$\dim(P) \coloneqq \min\{k \in \mathbb{Z}_{\geq 0} \colon \exists A \in \mathbb{R}^{n\times n} \text{ with } \operatorname{rank}(A) = n - k \;\&\; Ax = Ay \;\forall x, y \in P\} \;.$$

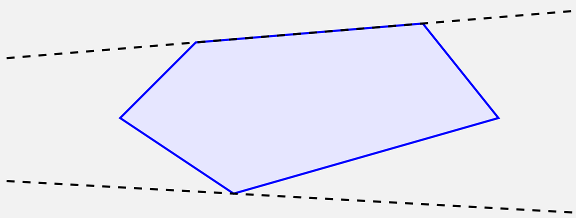
In particular, P is called *full-dimensional* if $\dim(P) = n$.

Definition 1.9: Supporting hyperplane

Let $P \subseteq \mathbb{R}^n$ be a polyhedron. A hyperplane $H = \{x \in \mathbb{R}^n \colon a^\top x = \beta\}$ is called P-*supporting*—or simply *supporting*, if P is clear from context—if $P \cap H \neq \emptyset$ and P is contained in one of the two half-spaces defined by H, i.e., either $P \subseteq \{x \in \mathbb{R}^n \colon a^\top x \leq \beta\}$ or $P \subseteq \{x \in \mathbb{R}^n \colon a^\top x \geq \beta\}$.

Example 1.10

The figure below shows a 2-dimensional polytope with two supporting hyperplanes.



Definition 1.11: Face, vertex, edge, and facet

Let $P \subseteq \mathbb{R}^n$ be a non-empty polyhedron.

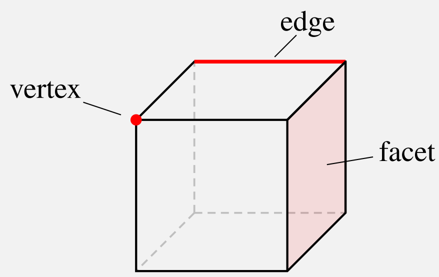
(i) A *face* of P is either P itself or the intersection of P with a supporting hyperplane.
(ii) A *vertex* of P is a 0-dimensional face of P.
(iii) An *edge* of P is a 1-dimensional face of P.
(iv) A *facet* of P is a $(\dim(P) - 1)$-dimensional face of P.

The empty polyhedron has only one face, which is the empty set. We denote by $\mathrm{vertices}(P)$ the set of all vertices of P.

Notice that a face of a polyhedron P is also a polyhedron, because it is the intersection of P with a hyperplane; indeed, this follows because a hyperplane is the intersection of two half-spaces and a polyhedron is by definition a finite intersection of half-spaces. Thus, when talking about the dimension of a face, we refer to the notion of dimension that we introduced for polyhedra.

Example 1.12

Below is a cube with three of its faces highlighted in red: a vertex, an edge, and a facet.



Note that vertices are the smallest (inclusion-wise and in terms of dimension) possible faces of a non-empty polyhedron P, and edges are the next-smallest faces P can have. On the other end, P is its largest face, and facets are the next-largest faces of P. A non-empty polyhedron does not need to have any vertices or edges; for example, a hyperplane in $\mathbb{R}^n$ for $n \geq 3$ is a polyhedron without vertices or edges. Moreover, the empty polyhedron, the whole space $\mathbb{R}^n$, as well as polytopes consisting of a single point, do not have any facets. However, all other polyhedra do have facets.

Proposition 1.13

Let $P = \{x \in \mathbb{R}^n \colon Ax \leq b\} \subseteq \mathbb{R}^n$ be a non-empty polyhedron with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, and let $F \subseteq P$. Then the following statements are equivalent.

(i) F is a face of P.
(ii) $\exists c \in \mathbb{R}^n$ such that $\delta := \max\{c^\top x \colon x \in P\}$ is finite and $F = \{x \in P \colon c^\top x = \delta\}$.
(iii) $F = \{x \in P \colon \overline{A}x = \overline{b}\} \neq \emptyset$ for a subsystem $\overline{A}x \leq \overline{b}$ of $Ax \leq b$.

Proof. Points (i) and (ii) are clearly equivalent by the definition of a face. We complete the proof by showing the equivalence between (iii) and (ii).

(iii) ⇒ (ii): Let $c = \overline{A}^\top \mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^m$ denotes the all-ones vector. We show that this vector c fulfills the condition stated in (ii). Indeed, for any $x \in P$ we have

$$c^\top x = \mathbf{1}^\top \overline{A} x \leq \mathbf{1}^\top \overline{b} \ ,$$

with equality if and only if $\overline{A}x = \overline{b}$. Hence, for $\delta := \mathbf{1}^\top b = \max\{c^\top x \colon x \in P\}$, we have

$$\{x \in P \colon c^\top x = \delta\} = \{x \in P \colon \overline{A}x = \overline{b}\} = F \ ,$$

as desired.

(ii) ⇒ (iii): Let $\overline{A}x \leq \overline{b}$ be the subsystem of all constraints of $Ax \leq b$ that are tight for all points in F. We will show that these constraints satisfy $F = \{x \in P \colon \overline{A}x = \overline{b}\}$, thus implying (iii).

Let $\widetilde{A}x \leq \widetilde{b}$ be all other constraints of $Ax \leq b$. Hence, for each constraint $\widetilde{a}^\top x \leq \widetilde{\beta}$ of $\widetilde{A}x \leq \widetilde{b}$, there is a point $x_{\widetilde{a}} \in F$ with $\widetilde{a}^\top x_{\widetilde{a}} < \beta$. By summing up all of these points $x_{\widetilde{a}}$ for all rows $\widetilde{a}$ of $\widetilde{A}$ and dividing by the number of rows of $\widetilde{A}$, a point $z \in \mathbb{R}^n$ is obtained such that

(i) $z \in F$ and hence $c^\top z = \delta$, because z is a convex combination of the $x_{\widetilde{a}}$, and
(ii) $\widetilde{A}z < \widetilde{b}$ component-wise.

We complete the proof by showing that any point $y \in P \setminus F$ fulfills $\overline{A}y \neq \overline{b}$. Because $y \in P \setminus F$, we have

$$c^\top y < \delta = c^\top z \ . \tag{1.3}$$

We now define for any $\epsilon > 0$ the point

$$x_\epsilon := z + \epsilon \cdot (z - y) \ .$$

Notice that for any $\epsilon > 0$ we have

$$c^\top x_\epsilon > \delta$$

because of (1.3). Hence, $x_\epsilon \notin P$ as $\delta = \max\{c^\top x \colon x \in P\}$. Thus, x_ϵ violates one of the constraints of $Ax \leq b$, no matter how small $\epsilon > 0$ is. For small enough $\epsilon > 0$, only constraints of $\overline{A}x \leq \overline{b}$ can be violated, because no other constraint of $Ax \leq b$ is tight at z. Hence, there is some constraint $\overline{a}^\top x \leq \overline{\beta}$ of $\overline{A}x \leq \overline{b}$ such that $\overline{a}^\top x_\epsilon > \overline{\beta}$ for any $\epsilon > 0$, which, by the definition of x_ϵ and the fact that $\overline{a}^\top z = \overline{\beta}$, implies $\overline{a}^\top y < \overline{\beta}$. Thus, $\overline{A}y \neq \overline{b}$, as desired. □

Proposition 1.13 reveals a basic property of faces, namely that the face relationship is transitive.

Corollary 1.14

Let P be a polyhedron. Then a face of a face of P is itself a face of P.

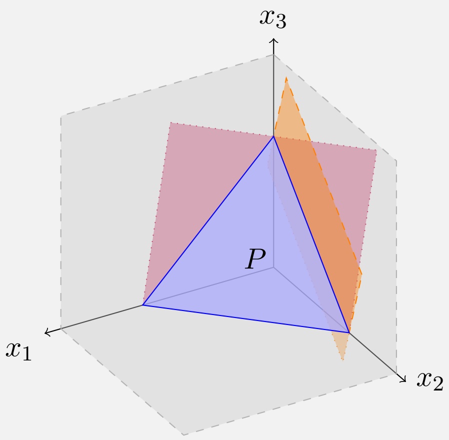
Proof. This follows immediately from point (iii) of Proposition 1.13. □

Definition 1.15: Facet-defining inequality

Let $P = \{x \in \mathbb{R}^n \colon Ax \leq b\}$ be a polyhedron. An inequality $a^\top x \leq \beta$ of the system $Ax \leq b$ is *facet-defining* if $F := P \cap \{x \in \mathbb{R}^n \colon a^\top x = \beta\}$ is a facet of P. We also say that $a^\top x \leq \beta$ is a constraint *defining* the facet F.

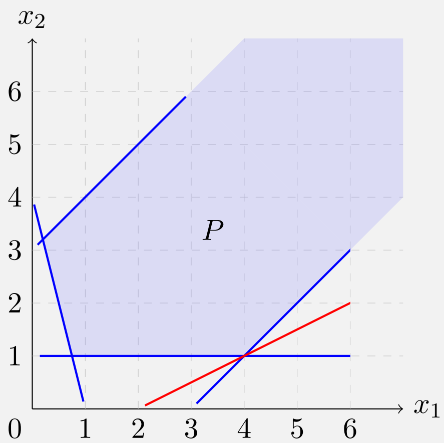
Example 1.16: Facet-defining inequalities

The example below shows a polytope $P \subseteq \mathbb{R}^3$ with $\dim(P) = 2$. The two red constraints imply an equality constraint and are responsible for P not being full-dimensional. Notice that neither of these two constraints is facet-defining even though they are both not redundant. Moreover, the orange constraint and the non-negativity constraint $x_1 \geq 0$ (highlighted in gray) both define the same facet of P.



$$P = \left\{ \begin{array}{rcrcrcr} x_1 & + & x_2 & + & x_3 & \leq & 2 \\ x_1 & + & x_2 & + & x_3 & \geq & 2 \\ x_1 & & & & & \geq & 0 \\ & & x_2 & & & \geq & 0 \\ & & & & x_3 & \geq & 0 \\ -4x_1 & + & x_2 & + & x_3 & \leq & 2 \end{array} \right\}$$

In the 2-dimensional example shown below, all constraints except for the red one are facet-defining. Whenever P is full-dimensional, every non-facet-defining constraint is redundant. However, as highlighted by the example above, this does not hold for lower-dimensional polyhedra.



$$P = \left\{ \begin{array}{rcrcr} x_1 & - & x_2 & \geq & -3 \\ 4x_1 & + & x_2 & \geq & 4 \\ & & x_2 & \geq & 1 \\ x_1 & - & 2x_2 & \leq & 2 \\ x_1 & - & x_2 & \leq & 3 \end{array} \right\}$$

Proposition 1.17

Let $P = \{x \in \mathbb{R}^n \colon Ax \leq b\}$ be a polyhedron. For any facet $F \subseteq P$ of P, there is at least one inequality $a^\top x \leq \beta$ of $Ax \leq b$ that defines F.

Proof. Let $F \subseteq P$ be a facet of P. By Proposition 1.13, F can be written as

$$F = \left\{x \in P \colon \overline{A}x = \overline{b}\right\} \quad , \tag{1.4}$$

where $\overline{A}x = \overline{b}$ is a subsystem of $Ax \leq b$. Because F is a facet of P we have $\dim(F) = \dim(P) - 1$. Hence, in particular $F \neq P$, which implies that there is an equality $\overline{a}^\top x = \overline{\beta}$ among $\overline{A}x = \overline{b}$ such that $P \not\subseteq \{x \in \mathbb{R}^n \colon \overline{a}^\top x = \overline{\beta}\}$. Hence

$$\dim(\{x \in P \colon \overline{a}^\top x = \overline{\beta}\}) < \dim(P) \quad . \tag{1.5}$$

We finish the proof by observing that

$$F = \{x \in P \colon \overline{a}^\top x = \overline{\beta}\} \quad .$$

Notice that (1.4) implies

$$F \subseteq \{x \in P : \overline{a}^\top x = \overline{\beta}\} \quad . \tag{1.6}$$

Furthermore, it is impossible that there is a point $y \in \{x \in P \colon \overline{a}^\top x = \overline{\beta}\} \setminus F$, as this would imply $\dim(\{x \in P \colon \overline{a}^\top x = \overline{\beta}\}) > \dim(F)$, which, together with (1.5) implies $\dim(F) \leq \dim(P) - 2$, thus contradicting $\dim(F) = \dim(P) - 1$. □

Definition 1.18: Extreme point

Let P be a polyhedron. A point $y \in P$ is an *extreme point* of P if it is not the midpoint of two distinct points of P, i.e., there are no $p_1, p_2 \in P$ with $p_1 \neq p_2$ and $y = \frac{1}{2}(p_1 + p_2)$.

Note that one can equivalently define an extreme point $y \in P$ of a polyhedron P by requiring that y is not a *non-trivial convex combination* of two distinct points $p_1, p_2 \in P$, i.e., we cannot write $y = \lambda p_1 + (1 - \lambda)p_2$ with $\lambda \in (0, 1)$. This follows by observing that if a point in a convex set C is a non-trivial convex combination of two distinct points in C, then it is also the midpoint of two distinct points in C. In short, this modified definition as well as the one stated in Definition 1.18 state that for y to be an extreme point in P, it should not be in the interior of a segment that lies in P.

Proposition 1.19

Let $P = \{x \in \mathbb{R}^n \colon Ax \leq b\}$ be a polyhedron and $y \in P$. Then the following statements are equivalent.

(i) y is a vertex of P.
(ii) y is the unique solution to $\overline{A}x = \overline{b}$, where $\overline{A}x \leq \overline{b}$ is the subsystem of $Ax \leq b$ containing all y-tight constraints.
(iii) y is an extreme point of P.

Proof. We show the statement by showing (i) ⇒ (iii) ⇒ (ii) ⇒ (i).

(i) ⇒ (iii): We show this statement by showing that if y is not an extreme point of P, then it is also not a vertex of P. The point y not being an extreme point implies that there are two distinct points $p_1, p_2 \in P$ with $y = \frac{1}{2}p_1 + \frac{1}{2}p_2$. For y to be a vertex, there needs to be a vector $c \in \mathbb{R}^n$ such that y is the unique maximizer of $\max\{c^\top x \colon x \in P\}$. However, this is impossible no matter how c is chosen, because $c^\top y = \frac{1}{2}c^\top p_1 + \frac{1}{2}c^\top p_2$. Hence, at least one of $c^\top p_1$ and $c^\top p_2$ is no smaller than y. Thus, y is not the unique maximizer of $\max\{c^\top x \colon x \in P\}$.

(iii) ⇒ (ii): We show this statement by showing the contraposition, i.e., we assume that y is not the unique solution to $\overline{A}x = \overline{b}$, where $\overline{A}x \leq \overline{b}$ is the subsystem of $Ax \leq b$ containing all y-tight constraints, and show that y is not an extreme point. Indeed, if y is not the unique solution of $\overline{A}x = \overline{b}$, then the dimension of $\{x \in \mathbb{R}^n \colon \overline{A}x = \overline{b}\}$ is at least one; hence, there is a non-zero vector $v \in \mathbb{R}^n$ such that $\overline{A}v = 0$. Moreover, by choosing $\epsilon > 0$ small enough we have $y + \epsilon v \in P$ and $y - \epsilon v \in P$, because all constraints of $Ax \leq b$ except for those corresponding to $\overline{A}x \leq \overline{b}$ are not y-tight, and $y \pm \epsilon v$ does not violate any y-tight constraints because $\overline{A}(y \pm \epsilon v) = \overline{A}y = \overline{b}$. However, this implies that y is the midpoint of $y + \epsilon v$ and $y - \epsilon v$, and thus, y is not an extreme point of P.

(ii) ⇒ (i): If y is the unique solution to $\overline{A}x = \overline{b}$, where $\overline{A}x \leq \overline{b}$ is the subsystem of $Ax \leq b$ containing all constraints that are y-tight, then

$$\{y\} = \{x \in P \colon \overline{A}x = \overline{b}\} \ .$$

By Proposition 1.13, $\{y\}$ is therefore a face of P and, because it is 0-dimensional, the point y is a vertex of P. □

Even though we focus here on polyhedra, we want to highlight that some natural results like Proposition 1.19 may not hold anymore when extending the involved notions to closed convex sets. For example, when defining extreme points and vertices for closed convex sets analogously to the polyhedral case, then an extreme point of a closed convex set does not need to be a vertex of it. See Figure 1.8 for an illustration.

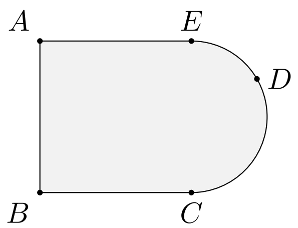


Figure 1.8: A closed and bounded convex set with an infinite number of extreme points and vertices, due to the curved boundary on the right of the object, which forms a half-circle. All of the five named points are extreme points. However, among these points, only A, B, and D are vertices. The points C and E are extreme points but not vertices because there is no supporting hyperplane that touches the set only at C or E, respectively.

Lemma 1.20

Let $P = \{x \in \mathbb{R}^n \colon Ax \leq b\}$ be a full-dimensional polyhedron, then each inequality $a^\top x \leq \beta$ of $Ax \leq b$ that is not facet-defining for P is redundant.

Proof. We prove the contraposition. Hence, assume that $a^\top x \leq \beta$ is not redundant, and we will show that it is facet-defining for P. Let $\overline{A}x \leq \overline{b}$ be all constraints of $Ax \leq b$ except for $a^\top x \leq \beta$. We define

$$Q \coloneqq \left\{x \in \mathbb{R}^n \colon \overline{A}x \leq \overline{b}\right\} \quad .$$

Because $a^\top x \leq \beta$ is not redundant, there is a point $y \in Q \setminus P$, and hence $a^\top y > \beta$. Moreover, let $z \in P$ be a point in the interior of P, which exists because P is full-dimensional. Consequently, there is some $\epsilon > 0$ such that the ball $B(z, \epsilon)$ around z of radius ϵ, i.e.,

$$B(z, \epsilon) = \{x \in \mathbb{R}^n \colon \|x - z\|_2 \leq \epsilon\} \quad ,$$

is contained in P, i.e., $B(z, \epsilon) \subseteq P$. Notice that $a^\top z < \beta$, because z is in the interior of P. Furthermore, as $B(z, \epsilon) \subseteq P \subseteq Q$ and $y \in Q$, we have that the set

$$C \coloneqq \mathrm{conv}(B(z, \epsilon) \cup \{y\})$$

satisfies

$$C \subseteq Q \quad .$$

Because C is full-dimensional and contains points in the interior of each of the sides of the hyperplane defined by $a^\top x = \beta$, namely the points y and z, we have that

$$C \cap \{x \in \mathbb{R}^n \colon a^\top x = \beta\}$$

is $(n-1)$-dimensional. Finally, $C \cap \{x \in \mathbb{R}^n \colon a^\top x = \beta\} \subseteq P$ because $C \subseteq Q$. Hence, $\{x \in P \colon a^\top x = \beta\}$ is $(n-1)$-dimensional, which implies that $a^\top x \leq \beta$ is facet-defining for P. □

Corollary 1.21

Let $P \subseteq \mathbb{R}^n$ be a polyhedron, and let f be the number of facets of P. Then, every inequality description of P requires at least f inequalities. Moreover, if P is full-dimensional, then an inequality description of P with f inequalities exists.

Proof. By Proposition 1.17, any inequality description of P needs at least one inequality per facet of P. It remains to show that there is an inequality description of P with facet-many inequalities if P is full-dimensional.

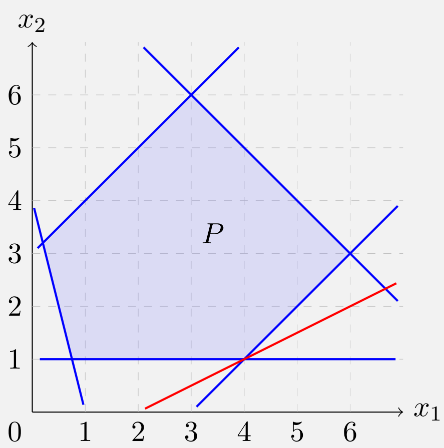
To this end, we start with an arbitrary inequality description $P = \{x \in \mathbb{R}^n \colon \overline{A}x \leq \overline{b}\}$ of P, and successively remove redundant constraints. We will be left with only facet-defining constraints due to Lemma 1.20. Let $Ax \leq b$ be the subsystem of $\overline{A}x \leq \overline{b}$ of remaining constraints after all redundant constraints have been successively removed from $\overline{A}x \leq \overline{b}$. Because only redundant constraints have been successively removed, we have $P = \{x \in \mathbb{R} \colon Ax \leq b\}$. We claim that this description has facet-many inequalities. Because each inequality of $Ax \leq b$ is facet-defining, it remains to show that no two different inequalities $a_1^\top x \leq \beta_1$ and $a_2^\top x \leq \beta_2$ of $Ax \leq b$ define the same facet $F \subseteq P$ of P. Assume by sake of contradiction that this is not the case, i.e., $F = \{x \in P \colon a_1^\top x = \beta_1\} = \{x \in P \colon a_2^\top x = \beta_2\}$. Now, because $P \subseteq \mathbb{R}^n$ is full-dimensional, there are n affinely independent points in F. As the hyperplanes $\{x \in \mathbb{R}^n \colon a_i^\top x = \beta_i\}$ for $i \in [2] := \{1, 2\}$ both contain these n affinely independent points, and must be such that $P \subseteq \{x \in \mathbb{R}^n \colon a_i^\top x \leq \beta_i\}$, the half-spaces $\{x \in \mathbb{R}^n \colon a_i^\top x \leq b_i\}$ are the same for $i \in [2]$. This contradicts the fact that we removed successively all redundant constraints, because the two constraints $a_i^\top x \leq \beta_i$ for $i \in [2]$ are redundant. (Even though this is not relevant for this proof, notice that after removing one of them, the other one may not be redundant anymore.) $\square$

Definition 1.22: Degeneracy with respect to linear inequality descriptions

Consider a linear inequality description of a polyhedron $P \subseteq \mathbb{R}^n$. A vertex $y \in \mathrm{vertices}(P)$ is called *degenerate* (w.r.t. the linear inequality description) if the number of y-tight constraints in the linear inequality description is strictly larger than n. The inequality description is called *degenerate* if there is a degenerate vertex with respect to it.

Example 1.23: Degenerate vertex & degenerate inequality description

Below is a 2-dimensional polytope with a corresponding inequality description that is degenerate, because the vertex $\binom{4}{1}$ is degenerate with respect to the given inequality description.



$$P = \left\{ \begin{array}{rcrcr} x_1 & - & x_2 & \geq & -3 \\ 4x_1 & + & x_2 & \geq & 4 \\ & & x_2 & \geq & 1 \\ x_1 & - & 2x_2 & \leq & 2 \\ x_1 & - & x_2 & \leq & 3 \\ x_1 & + & x_2 & \leq & 8 \end{array} \right\}$$

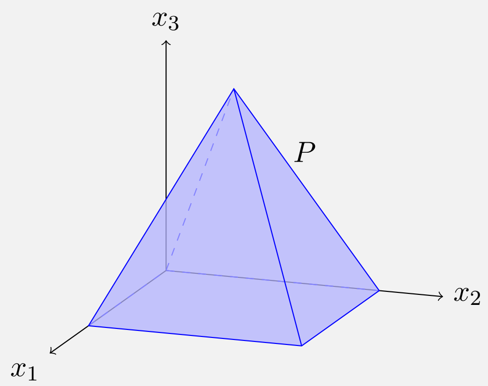
Definition 1.24: Degeneracy of a polyhedron

A polyhedron $P \subseteq \mathbb{R}^n$ is called *degenerate* if it has a vertex $y \in P$ contained on strictly more than n many facets.

Notice that even though the inequality description of the polytope in Example 1.23 is degenerate, the polytope itself is not degenerate.

Example 1.25: Degenerate polyhedron

Below is an example of a degenerate 3-dimensional polytope, namely a pyramid with a square base. It is degenerate because the apex of the pyramid is a vertex contained in 4 facets.



$$P = \left\{ \begin{array}{rrrrrr} & & & x_3 & \geq & 0 \\ 2x_1 & & - & x_3 & \geq & 0 \\ 2x_1 & & + & x_3 & \leq & 2 \\ & 2x_2 & - & x_3 & \geq & 0 \\ & 2x_2 & + & x_3 & \leq & 2 \end{array} \right\}$$

In particular, if a polyhedron $P \subseteq \mathbb{R}^n$ is degenerate, then any inequality description for it is degenerate too. This is a consequence of Proposition 1.17, which implies that for each facet there must be a facet-defining inequality.

Definition 1.26: Dominant

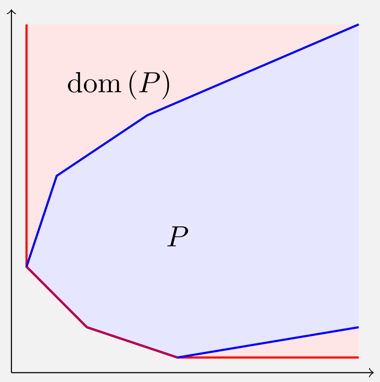
For a set $X \subseteq \mathbb{R}^n$, its dominant $\mathrm{dom}(X) \subseteq \mathbb{R}^n$ is defined by

$$\mathrm{dom}(X) := X + \mathbb{R}^n_{\geq 0} = \{x + y \colon x \in X, y \in \mathbb{R}^n_{\geq 0}\} \ .$$

For $X, Y \subseteq \mathbb{R}^n$, the set addition $X + Y := \{x + y \colon x \in X, y \in Y\}$, which we used above in the definition of the dominant, is called the *Minkowski sum*.

Example 1.27: Dominant

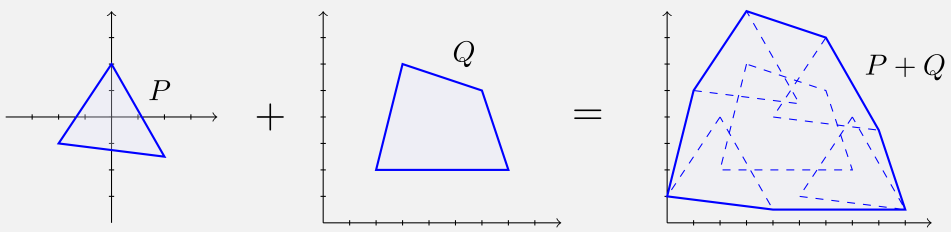
The figure below shows an unbounded polyhedron $P \subseteq \mathbb{R}^2$ together with its corresponding dominant $\mathrm{dom}(P)$.



Before moving over to the representation of polyhedra, we provide an example of a Minkowski sum, because this simple yet useful operation will be relevant also later one. Hence, it is helpful to develop some intuition for it.

Example 1.28: Minowski sum

The graphic below shows the Minkowski sum $P + Q$ of two 2-dimensional polytopes P and Q. The third pictures illustrates that $P + Q$ can be obtained by "shifting" one of the polytopes, say P, to each vertex of the other one—such that the origin in the figure showing P lies at the vertices of Q—and then taking the convex hull of the shifted versions of P. In particular, each vertex of $P + Q$ is the sum of a vertex of P and one of Q. However, not all sums of vertices in P and Q lead to a vertex of $P + Q$.



1.2.2 Representation of polyhedra

As mentioned, polyhedra are a basic family of convex sets. Not surprisingly, convexity is exploited in a multitude of results and algorithms linked to polyhedra. At the heart of convexity is the notion of a convex combination.

Definition 1.29: Convex combination

A *convex combination* of $x_1, \dots, x_k \in \mathbb{R}^n$ is a point described by $\sum_{i=1}^k \lambda_i x_i$, where $\lambda \in \mathbb{R}^k_{\geq 0}$ and $\sum_{i=1}^k \lambda_i = 1$.

In this course, convex combinations are always with respect to a finite number of elements. The fact that linear combinations with the above properties are called convex combinations is motivated by the observation that a set $X \subseteq \mathbb{R}^n$ is convex if and only if any convex combination of finitely many points in X lies in X. The convex combinations of an arbitrary point set are known as its convex hull.

Definition 1.30: Convex hull

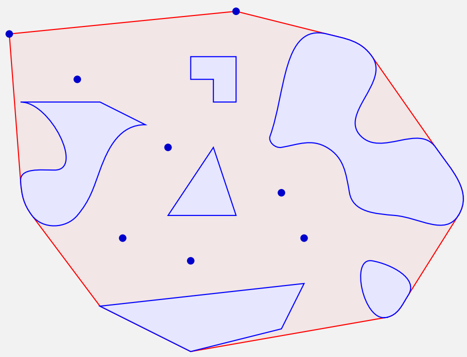
Let $X \subseteq \mathbb{R}^n$. The *convex hull* $\operatorname{conv}(X) \subseteq \mathbb{R}^n$ of X are all convex combinations of finitely many points in X, i.e.,

$$\operatorname{conv}(X) := \left\{ \sum_{i=1}^{k} \lambda_i x_i \;\middle|\; \begin{array}{l} k \in \mathbb{Z}_{\geq 1},\ x_1, \dots, x_k \in X, \text{ and} \\ \lambda_1, \dots, \lambda_k \in \mathbb{R}_{\geq 0} \text{ with } \sum_{i=1}^{k} \lambda_i = 1 \end{array} \right\} .$$

Hence, a set $X \subseteq \mathbb{R}^n$ is convex if and only if $\operatorname{conv}(X) = X$. This, together with the fact that the convex hull operator is monotone, i.e., $\operatorname{conv}(X) \subseteq \operatorname{conv}(Y)$ for any $X \subseteq Y \subseteq \mathbb{R}^n$, implies that $\operatorname{conv}(X)$ is the smallest convex set containing X.

Example 1.31: Convex hull

Example of the convex hull in 2 dimensions. The red area is the convex hull of the blue objects.



The following proposition highlights how polytopes are described by their vertices. The statement will be proven in the problem sets.

Proposition 1.32

A polytope is the convex hull of its vertices.

Moreover, as stated below, the convex hull of any finite point set is a polytope. Again, the statement will be shown in one of the problem sets.

Proposition 1.33

Let $X \subseteq \mathbb{R}^n$ be a finite set. Then $\operatorname{conv}(X)$ is a polytope.

To move from polytopes to polyhedra, we have to consider unbounded sets. One of the most basic building blocks when dealing with unbounded sets are cones, which we define below. They allow for obtaining a nice characterization of polyhedra that we mention later.

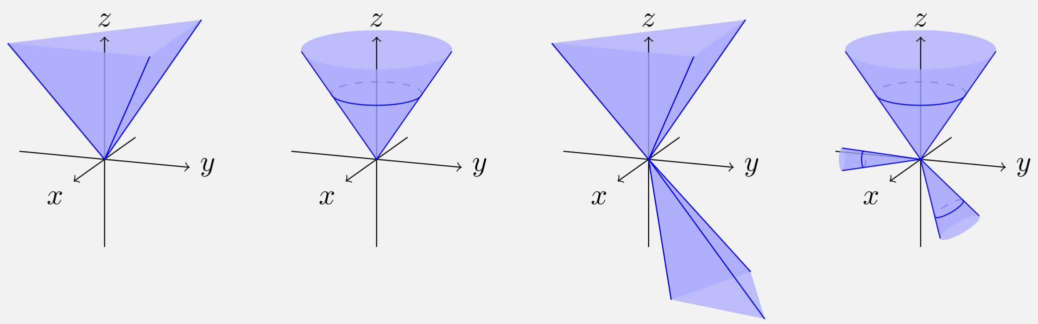
Definition 1.34: (Polyhedral) cone

A *cone* is a set $C \subseteq \mathbb{R}^n$ such that for any $x \in C$ and $\lambda \in \mathbb{R}_{\geq 0}$ we have $\lambda \cdot x \in C$. A cone that is a polyhedron is called a *polyhedral cone*.

We highlight that a cone does not need to be convex. Nevertheless, we will mostly be interested in convex cones. These can be characterized as the sets that are invariant under *conic combinations*, which are linear combinations with exclusively non-negative coefficients.

Example 1.35

Below are four examples of cones in $\mathbb{R}^3$. The first two of them are convex cones whereas the last two are not. Moreover, only the first one is a polyhedral cone. Notice that the third is not a polyhedral cone because it is not a polyhedron as it is not even convex.



Proposition 1.36

If $C \subseteq \mathbb{R}^n$ is a non-empty polyhedral cone, then

$$C = \{x \in \mathbb{R}^n \colon Ax \leq 0\} \ , \tag{1.7}$$

for some matrix $A \in \mathbb{R}^{m \times n}$, where $m \in \mathbb{Z}_{\geq 0}$. Vice-versa, any set C with a description as in (1.7) is a polyhedral cone.

Proof. Let $C \subseteq \mathbb{R}^n$ be a polyhedral cone. Because C is a polyhedron, we have

$$C = \{x \in \mathbb{R}^n \colon Ax \leq b\}$$

for some $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, and we assume without loss of generality that none of the constraints $Ax \leq b$ are redundant; for otherwise, we could first successively remove redundant constraints. We claim that $b = 0$. For otherwise, there is some constraint $a^\top x \leq \beta$ with $\beta \neq 0$ in the system $Ax \leq b$. Because $a^\top x \leq \beta$ is not redundant, there is a point $y \in C$ with $a^\top y = \beta$.

However, because C is a cone, we must have $\frac{1}{2}y, 2y \in C$; but one of these points violates the constraint $a^\top x \leq \beta$. (If $\beta > 0$, then $2y$ will violate it; otherwise $\frac{1}{2}y$ will violate the constraint.) Hence, $b = 0$, and we have $C = \{x \in \mathbb{R}^n : Ax \leq 0\}$ as desired.

Conversely, any set $C \subseteq \mathbb{R}^n$ with $C = \{x \in \mathbb{R}^n : Ax \leq 0\}$ for $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ is clearly a polyhedron and a cone, and thus a polyhedral cone. □

Another way to characterize polyhedral cones is by the fact that they have a finite set of generators, as formalized in the statement below. This will be shown in the problem sets.

Proposition 1.37

If $C \subseteq \mathbb{R}^n$ is a polyhedral cone, then

$$C = \left\{ \sum_{i=1}^{k} \lambda_i x_i : \lambda_i \geq 0 \; \forall i \in [k] \right\} \quad , \tag{1.8}$$

for some finite set of points $x_1, \dots, x_k \in \mathbb{R}^n$. The points $x_1, \dots, x_k$ are called a *set of generators* of C. Vice-versa, any set C as described in (1.8) is a polyhedral cone.

Polyhedral cones can be thought of as the most basic unbounded polyhedra. The statement below underlines this intuition by showing that any polyhedron can be written as a Minkowski sum of a polytope and a polyhedral cone. The proof of it will be covered in the problem sets.

Proposition 1.38

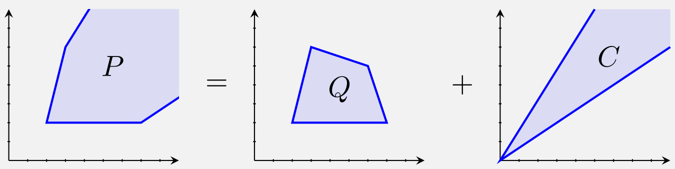
Let $P \subseteq \mathbb{R}^n$ be a polyhedron. Then

$$P = Q + C \quad ,$$

where $Q \subseteq \mathbb{R}^n$ is a polytope and $C \subseteq \mathbb{R}^n$ is a polyhedral cone. Vice-versa, the Minkowski sum of a polytope and a polyhedral cone is always a polyhedron.

Example 1.39

The graphic below shows an unbounded 2-dimensional polyhedron P and how it can be written as the Minkowski sum of a polytope Q and a polyhedral cone C.



The above characterizations of polytopes and polyhedra often allow for deriving short and elegant proofs of further important statements. One such statement, shown below, states that the family of polyhedra is invariant under affine transformations. This property is at the heart of various techniques in Combinatorial Optimization, in particular when a complex polytope, say one with a very large number of facets, is described as the affine projection of a much simpler one.

Proposition 1.40

An affine image of a polyhedron is a polyhedron, i.e., for any polyhedron $P \subseteq \mathbb{R}^n$ and any affine function $\varphi\colon \mathbb{R}^n \to \mathbb{R}^m$, the set $\varphi(P) := \{\varphi(x)\colon x \in P\}$ is a polyhedron.

Proof. We first observe that it suffices to prove the statement only for linear functions $\varphi\colon \mathbb{R}^n \to \mathbb{R}^m$ instead of affine ones. This follows from the fact that a translation of a polyhedron is again a polyhedron. Indeed, a general affine function $\varphi\colon \mathbb{R}^n \to \mathbb{R}^m$ can always be written as $\varphi(x) = \phi(x) + t$, where $\phi\colon \mathbb{R}^n \to \mathbb{R}^m$ is a linear function and $t \in \mathbb{R}^m$. Now assume that for any polyhedron $P \subseteq \mathbb{R}^n$, the image $\phi(P) \subseteq \mathbb{R}^m$ is again a polyhedron. We then have

$$\begin{aligned}\varphi(P) &= \phi(P) + t\\ &= \{x \in \mathbb{R}^m\colon Ax \leq b\} + t\\ &= \{x + t\colon x \in \mathbb{R}^m, Ax \leq b\}\\ &= \{y \in \mathbb{R}^m\colon A(y-t) \leq b\}\\ &= \{y \in \mathbb{R}^m\colon Ay \leq b + At\}\ ,\end{aligned} \tag{1.9}$$

where the second equality follows from the assumption that $\phi(P)$ is a polyhedron, and can thus be described by a linear inequality system $\phi(P) = \{x \in \mathbb{R}^m\colon Ax \leq b\}$, and the penultimate equality substitutes $x + t$ by y. Indeed, (1.9) shows that $\varphi(P)$ is a finite intersection of half-spaces and therefore a polyhedron, as desired.

Hence, let $P \subseteq \mathbb{R}^n$ and it remains to show the statement for linear functions $\varphi\colon \mathbb{R}^n \to \mathbb{R}^m$. To highlight that the function is linear, we denote it again by $\phi\colon \mathbb{R}^n \to \mathbb{R}^m$. By Proposition 1.38, any polyhedron P can be written as a Minkowski sum

$$P = Q + C\ ,$$

where $Q \subseteq \mathbb{R}^n$ is a polytope and $C \subseteq \mathbb{R}^n$ is a polyhedral cone. Observe that

$$\begin{aligned}\phi(P) &= \phi(Q + C)\\ &= \{\phi(q + c)\colon q \in Q, c \in C\}\\ &= \{\phi(q) + \phi(c)\colon q \in Q, c \in C\}\\ &= \phi(Q) + \phi(C)\ .\end{aligned} \tag{1.10}$$

We finish the proof by showing that $\phi(Q) \subseteq \mathbb{R}^m$ is a polytope and $\phi(C) \subseteq \mathbb{R}^m$ is a polyhedral cone, which, by Proposition 1.38 then implies that $\phi(Q) + \phi(C)$ is a polyedron, thus finishing the proof because $\phi(P) = \phi(Q) + \phi(C)$ by (1.10).

Because Q is a polytope, we have

$$Q = \operatorname{conv}(\operatorname{vertices}(Q))$$

by Proposition 1.32. Let $\text{vertices}(Q) = \{q_1, \dots, q_k\}$. Hence,

$$
\begin{aligned}
\phi(Q) &= \{\phi(x) \colon x \in \text{conv}(\text{vertices}(Q))\} \\
&= \left\{ \phi\left(\sum_{i=1}^{k} \lambda_i q_i\right) : \lambda \in \mathbb{R}^k_{\geq 0}, \sum_{i=1}^{k} \lambda_i = 1 \right\} \\
&= \left\{ \sum_{i=1}^{k} \lambda_i \phi(q_i) \colon \lambda \in \mathbb{R}^k_{\geq 0}, \sum_{i=1}^{k} \lambda_i = 1 \right\} \\
&= \text{conv}(\{\phi(q) \colon q \in \text{vertices}(Q)\}) \enspace ,
\end{aligned}
$$

implying that $\phi(Q)$ is the convex hull of the finitely many points $\phi(q_1), \dots, \phi(q_k)$, and thus is a polytope due to Proposition 1.32.

It remains to show that $\phi(C) \subseteq \mathbb{R}^m$ is a cone. By Proposition 1.37, there are some points $x_1, \dots, x_\ell \subseteq \mathbb{R}^n$ such that

$$
C = \left\{ \sum_{i=1}^{\ell} \lambda_i x_i \colon \lambda \in \mathbb{R}^\ell_{\geq 0} \right\} \enspace .
$$

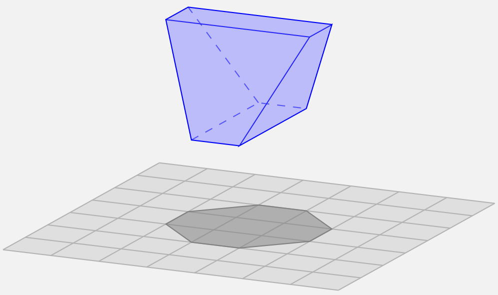
Hence,

$$
\begin{aligned}
\phi(C) &= \left\{ \phi\left(\sum_{i=1}^{\ell} \lambda_i x_i\right) : \lambda \in \mathbb{R}^\ell_{\geq 0} \right\} \\
&= \left\{ \sum_{i=1}^{\ell} \lambda_i \phi(x_i) \colon \lambda \in \mathbb{R}^\ell_{\geq 0} \right\} \enspace ,
\end{aligned}
$$

which implies by Proposition 1.37 that $\phi(C) \subseteq \mathbb{R}^m$ is a cone, as desired. □

Example 1.41: Projection of polytope

The illustration below shows an axis-parallel projection of a 3-dimensional polytope leading to a 2-dimensional octagon. First, this exemplifies that the projection of a polytope is a polytope.



Moreover, we highlight that the number of facets of the projection, which is 8, is strictly larger than the number of facets of the original polytope, which is only 6. The fact that polyhedra with many facets can sometimes be represented as projections of polyhedra with much fewer

facets is a crucial property that can sometimes be exploited to obtain compact representations of polyhedra with exponentially many facets in terms of the dimension. More precisely, there are polytopes $P \subseteq \mathbb{R}^n$ whose number of facets is $2^{\Omega(n)}$, which can be obtained as a projection of a polytope $Q \subseteq \mathbb{R}^m$ for some $m \geq n$, where the number of facets of Q is bounded by a polynomial in n.
Moving to a higher-dimensional space and then projecting is crucial to obtain a compact representation. Indeed, due to Corollary 1.21, any inequality description of a polyhedron with exponentially many facets needs exponentially many constraints; actually, at least one per facet.

Moreover, Proposition 1.40 can be leveraged to show that certain sets are polyhedra by showing that they are affine projections of polyhedra. This allows us to provide a simple proof of the following basic fact on dominants.

Proposition 1.42

The dominant of a polyhedron is a polyhedron.

Proof. Let $P \subseteq \mathbb{R}^n$ be a polyhedron. We recall the definition of the dominant of P, i.e.,

$$\operatorname{dom}(P) = P + \mathbb{R}^n_{\geq 0} \ .$$

By Proposition 1.38, the polyhedron P can be written as

$$P = Q + C \ ,$$

where $Q \subseteq \mathbb{R}^n$ is a polytope and $C \subseteq \mathbb{R}^n$ is a polyhedral cone. Hence,

$$\operatorname{dom}(P) = Q + C + \mathbb{R}^n_{\geq 0} \ .$$

We prove the result by showing that $C + \mathbb{R}^n_{\geq 0}$ is a polyhedral cone. This indeed finishes the proof because $\operatorname{dom}(P) = Q + (C + \mathbb{R}^n_{\geq 0})$ is the Minkowski sum of a polyotpe Q and a polyhedral cone $C + \mathbb{R}^n_{\geq 0}$, and thus a polyhedron due to Proposition 1.38.

By Proposition 1.37, there are points $x_1, \dots, x_k \in \mathbb{R}^n$ such that

$$C = \left\{ \sum_{j=1}^{k} \mu_j x_j \colon \mu \in \mathbb{R}^k_{\geq 0} \right\} \ .$$

Hence, by denoting by $e_i \in \mathbb{R}^n$ the i-th unit vector in $\mathbb{R}^n$, we have

$$\begin{aligned} C + \mathbb{R}^n_{\geq 0} &= \left\{ \sum_{j=1}^{k} \mu_j x_j \colon \mu \in \mathbb{R}^k_{\geq 0} \right\} + \left\{ \sum_{i=1}^{n} \lambda_i e_i \colon \lambda \in \mathbb{R}^n_{\geq 0} \right\} \\ &= \left\{ \sum_{j=1}^{k} \mu_j x_j + \sum_{i=1}^{n} \lambda_i e_i \colon \mu \in \mathbb{R}^k_{\geq 0}, \lambda \in \mathbb{R}^n_{\geq 0} \right\} \ , \end{aligned}$$

which implies by Proposition 1.37 that $C + \mathbb{R}^n_{\geq 0}$ is a cone, as desired. □

1.2.3 Convex separation theorems

Definition 1.43: (Strictly) separating hyperplanes

Let $Y, Z \subseteq \mathbb{R}^n$ be two sets. A hyperplane $H = \{x \in \mathbb{R}^n \colon a^\top x \leq \beta\}$ is called a (Y, Z)-*separating hyperplane*, or simply *separating hyperplane*, if Y is contained in one of the half-spaces defined by H and Z in the other one, i.e., either

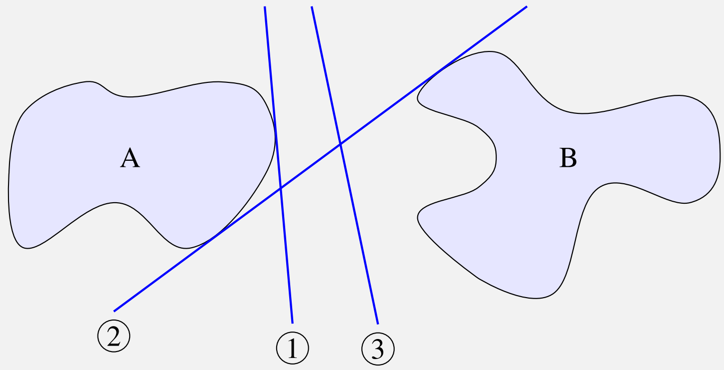
$$\begin{aligned} a^\top y \leq \beta \leq a^\top z \qquad & \forall y \in Y, z \in Z \;\; \text{, or} \\ a^\top y \geq \beta \geq a^\top z \qquad & \forall y \in Y, z \in Z \;\; . \end{aligned}$$

The hyperplane is called *strictly* (Y, Z)*-separating*, or simply *strictly separating*, if the above inequalities are strict.

If $Y = \{y\}$ is a single point, we also write (y, Z)-separating hyperplane instead of $(\{y\}, Z)$-separating hyperplane.

Example 1.44: Separating two sets

The illustration below shows two sets $A, B \subseteq \mathbb{R}^2$ together with three separating hyperplanes. Hyperplane 3 is strictly separating the sets whereas hyperplanes 1 and 2 do not separate A and B in a strict sense.



Theorem 1.45: Separating a point from a polyhedron

Let $P \subseteq \mathbb{R}^n$ be a polyhedron and $y \in \mathbb{R}^n \setminus P$. Then there is a strictly (y, P)-separating hyperplane.

Proof. Consider an inequality description of P, i.e., $P = \{x \in \mathbb{R}^n \colon Ax \leq b\}$. Because $y \in \mathbb{R}^n \setminus P$, there is an inequality $a^\top x \leq \beta$ among the system $Ax \leq b$ such that $a^\top y > \beta$. Then, the hyperplane

$$\left\{x \in \mathbb{R}^n \colon a^\top x = \frac{1}{2}\left(\beta + a^\top y\right)\right\}$$

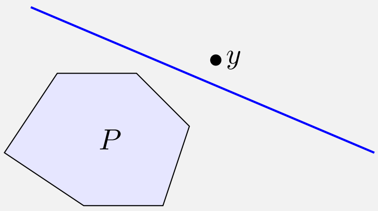
is strictly (y, P)-separating because

$$\begin{aligned} a^\top x \leq \beta &< \frac{1}{2}\left(\beta + a^\top y\right) \qquad \forall x \in P, \text{ and} \\ a^\top y &> \frac{1}{2}\left(\beta + a^\top y\right) \quad , \end{aligned}$$

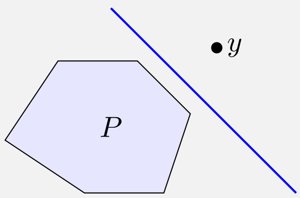
where the last inequality follows from $a^\top y > \beta$. □

Example 1.46: Separating a point from a polyhedron

The illustration below shows a hyperplane strictly separating a point from a polytope. Whenever a point is not contained in a polytope, it can always be strictly separated.



Moreover, as the proof of Theorem 1.45 highlights, one can always choose a separating hyperplane with the same normal vector as one of the constraints in any inequality description of P. In particular, if P is full-dimensional, like in the example above, we can use an inequality description that has one inequality per facet (see Corollary 1.21), and therefore separate y with a hyperplane that is parallel to a facet of P, as illustrated below.



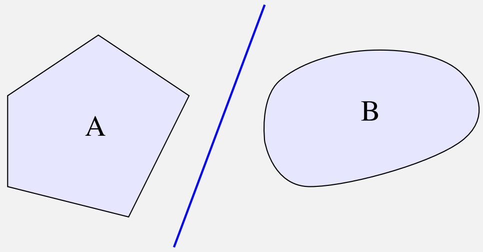
The above separation theorem is a very special case of the following much more general convex separation theorem, which we state below without proof.

Theorem 1.47

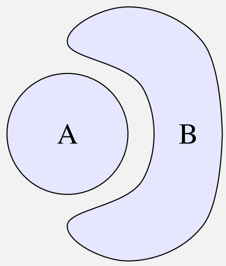
Let $Y, Z \subseteq \mathbb{R}^n$ be two disjoint closed convex sets with at least one of them being compact, then there exists a strictly (Y, Z)-separating hyperplane.

Example 1.48: Separating two sets

The illustration below shows two convex sets strictly separated by a hyperplane, which is always possible.



However, when at least one of the sets is not convex, separation may not be possible anymore as illustrated in the example below.



1.3 Simplex Method

What is known as Simplex Methods, or simply the Simplex Method, is a class of approaches to solve linear programs. Simplex Methods maintain a vertex solution that they iteratively improve by moving along the edges of the feasible region to strictly better vertex solutions. The Simplex Method is widely used in practice, despite the fact that there is no known realization of it that is guaranteed to run efficiently, i.e., in polynomial time (see Chapter 2 for more details on Computational Complexity). Moreover, most variants of the Simplex Method are even known to run in exponential time on some instances. Nevertheless, the Simplex Method shows excellent performance when applied to practical problems. Moreover, its study allows us to obtain a deeper understanding of linear programming in general and comes in handy to discuss further concepts like linear duality.

1.3.1 Geometric idea

We start with a high-level description of the main geometric idea behind the Simplex Method. Consider a linear program $\max\{c^\top x \colon x \in P\}$, where $P \subseteq \mathbb{R}^n$ is a polyhedron. Assume that we know a vertex $y \in \operatorname{vertices}(P)$. The Simplex Method seeks to improve the solution y through a local-search approach, that goes along one of the edges incident with y. In case of P being a polytope, each of these edges will lead to another vertex of P; such vertices that can be reached from y by following an edge are called *polyhedral neighbors*, or simply *neighbors*, of y (with respect to P). A key observation behind the Simplex Method is the following:

(i) Either y is an optimal solution to $\max\{c^\top x \colon x \in P\}$, or

(ii) there is an edge incident with y such that by going along the edge, the objective value strictly increases. In particular, if another vertex z is encountered along this edge, then $c^\top z > c^\top y$; otherwise, if one can continue indefinitely along the edge without leaving P, then the linear program is unbounded because we can get solutions with arbitrarily high objective value.

To illustrate the above, consider a linear program on a polytope P. Here, the Simplex Method will do a walk along the edges of P, going from one vertex to a strictly better one until an optimal vertex is reached. See Figure 1.9 for an illustration.

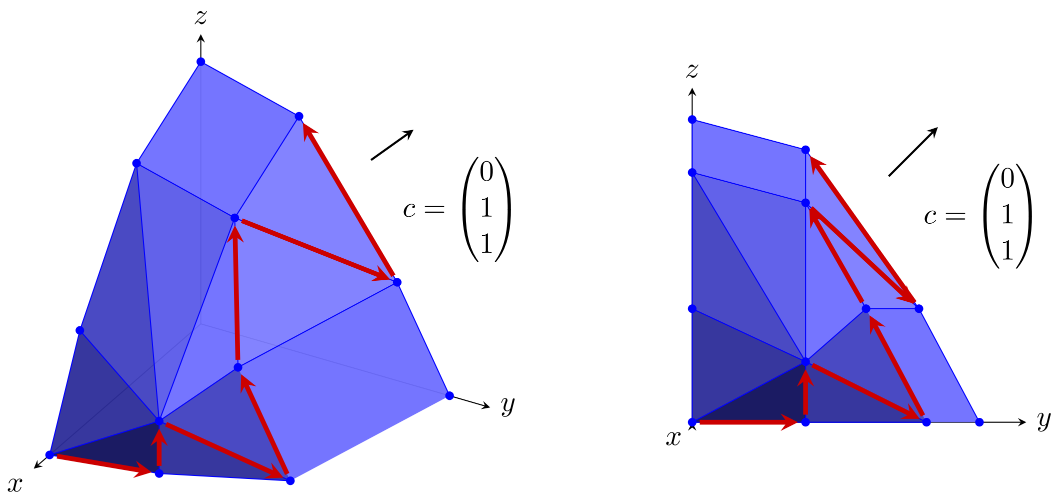


Figure 1.9: Illustration of a possible walk performed by the Simplex Method in a 3-dimensional linear program. The feasible region is the polytope depicted in blue and the task is to maximize the objective $y + z$. The right-hand side shows the projection of the feasible region onto the (y, z)-space. In this projection, one can nicely see that whenever we move from a vertex to a neighboring one, the objective function value strictly increases.

In case of an unbounded LP, which implies in particular that P is an unbounded polyhedron, the walk along the edges of P continues until $y \in \mathrm{vertices}(P)$ is found together with an improving edge direction $v \in \mathbb{R}^n$, i.e., $c^\top v > 0$, such that $y + \lambda v \in P$ for $\lambda \in \mathbb{R}_{\geq 0}$. See Figure 1.10 for an illustration.

This simple geometric intuition of the Simplex Method leaves several technical questions unanswered, including:

- Given $y \in \mathrm{vertices}(P)$, how to find an improving edge direction?
- How to find a starting vertex $y \in \mathrm{vertices}(P)$?
- How to use such a method to show that a linear program is infeasible?

We start by addressing the first question, for which the so-called *phase II* of the Simplex Method provides an answer. Once this is understood, a simple yet elegant auxiliary construction allows for answering the other two questions; this is known as *phase I* of the Simplex Method.

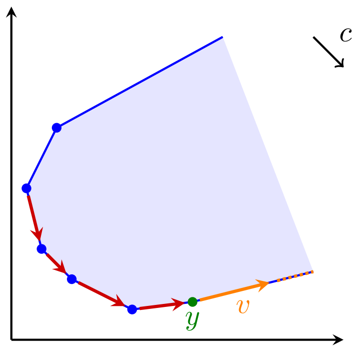


Figure 1.10: In the above unbounded example, the red arrows highlight the vertex-to-vertex walk performed by the Simplex Method. At the last vertex y there is an edge-direction v such that $y + \lambda v$ for $\lambda \geq 0$ is a feasible half-line with points of arbitrarily large objective value.

To algebraically realize the geometric idea of moving from vertex to vertex along edges, we will start with an LP in canonical form and then transform this description of the LP into the so-called *standard form*. The standard form only has equality constraints on top of the usual non-negativity requirements on the variables. This form allows us to highlight a particular vertex, and can be brought into equivalent standard forms highlighting a neighboring vertex. This way, we can capture the walk from vertex to vertex that is performed by the Simplex Method through changing the standard form. We therefore start by introducing the standard form and showing how to move between equivalent standard forms.

1.3.2 From canonical to standard form

A linear program is described in *standard form* if it only has equality constraints and its variables are required to be non-negative. Hence, it is of the following form:

$$\begin{array}{rrcl} \max & c^\top x & & \\ & Ax & = & b \\ & x & \geq & 0 \,. \end{array} \qquad \text{(LP in standard form)}$$

In the exposition that follows to introduce the Simplex Method, we will actually start with a general LP in canonical form:

$$\begin{array}{rrcl} \max & c^\top x & & \\ & Ax & \leq & b \\ & x & \geq & 0 \,, \end{array}$$

where $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, and $b \in \mathbb{R}^m$. From this canonical form, we build the corresponding standard form by introducing a new set of m *slack variables* $y \in \mathbb{R}^m$ in the following way:

$$\begin{array}{rrcccl} \max & c^\top x & & & & \\ & Ax & + & y & = & b \\ & x & & & \geq & 0 \\ & & & y & \geq & 0 \,. \end{array} \tag{1.11}$$

Indeed, the feasible values that $x \in \mathbb{R}^n$ can take in the above LP in standard form are the same as in the original LP in canonical form. Hence, the two linear programs are equivalent. Going from canonical to standard form is a very common transformation. The variables y are called *slack variables* because they measure the slack of x with respect to the originally given inequalities in canonical form.

One may wonder why we started with an LP in canonical form, and only then moved to standard form, instead of right-away starting with an LP in standard form. The main reason is that even though we will do operations in standard form from now on, we interpret these operations as walking from vertex to vertex on the polyhedron $P := \{x \in \mathbb{R}^n_{\geq 0} \colon Ax \leq b\}$, i.e., the feasible region of the original LP given in canonical form. Hence, we will use the standard form of a canonical LP description as a convenient algebraic way to represent a walk along vertices of P. Thus, the geometric interpretation is with respect to the original LP in canonical form.

Some terminology

A pair $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$ satisfying the equation system $Ax + y = b$ defined in (1.11) is called a *solution* of the LP. If a solution also satisfies $x \geq 0$ and $y \geq 0$ it is called *feasible*.

We will first discuss solutions to the system of equations $Ax + y = b$, with m equations and $n + m$ unknowns. Notice that this system has full row-rank due to the variables y. Hence, it admits solutions for any right-hand side $b \in \mathbb{R}^m$.

In what follows, we use the simple LP below, in canonical form, to exemplify some notation and terminology.

$$
\begin{array}{rrcrcrl}
\max z = & 400x_1 & + & 900x_2 & & & \\
 & x_1 & + & 4x_2 & \leq & 40 & \text{(constraint 1)} \\
 & 2x_1 & + & x_2 & \leq & 42 & \text{(constraint 2)} \\
 & 1.5x_1 & + & 3x_2 & \leq & 36 & \text{(constraint 3)} \\
 & x_1 & & & \geq & 0 & \text{(non-negativity for } x_1) \\
 & & & x_2 & \geq & 0 & \text{(non-negativity for } x_2)
\end{array}
\tag{1.12}
$$

Figure 1.11 highlights the feasible region of LP (1.12).

To transform the above LP to the corresponding standard form, we introduce three slack variables y_1, y_2, and y_3, leading to the following system, represented in tabular form below.

$$
\boxed{\mathrm{I}} \quad
\begin{array}{c|ccccc|c}
 & y_1 & y_2 & y_3 & x_1 & x_2 & 1 \\
\hline
 & 1 & 0 & 0 & 1 & 4 & 40 \\
 & 0 & 1 & 0 & 2 & 1 & 42 \\
 & 0 & 0 & 1 & 1.5 & 3 & 36
\end{array}
$$

$\underbrace{\qquad\qquad\qquad}_{\text{coefficient matrix}}$ $\underbrace{\quad}_{\text{right-hand side}}$

An important property of the standard LP form built from a canonical LP form is that there is a one-to-one correspondence between variables in the standard form and constraints in the original

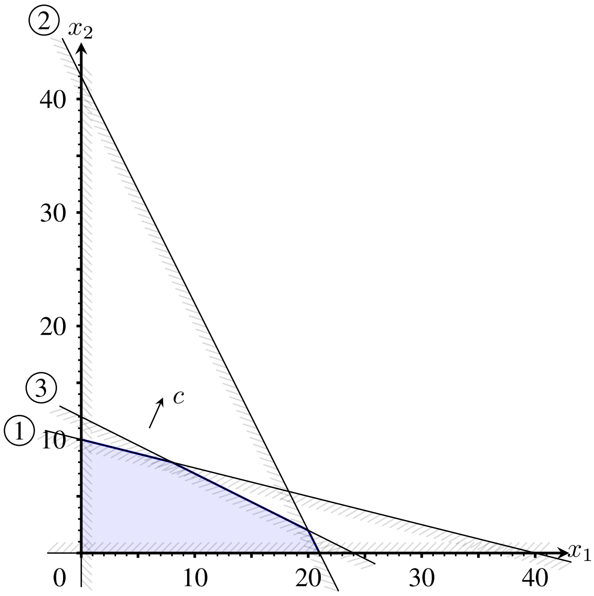


Figure 1.11: Graphical representation of the linear program (1.12).

canonical form (including also the non-negativity constraints). More precisely, even though only the variables y_1, y_2, and y_3 are called *slack variables* in the above example, also the variables x_1 and x_2 measure slack, namely the slack with respect to the non-negativity constraints in (1.12).

Due to the particular form of the linear equation system in the standard LP above, we can easily obtain a parametrized form of the set of all solutions, namely:

$$\begin{array}{rcrcrcr} y_1 & = & 40 & - & x_1 & - & 4x_2 \\ y_2 & = & 42 & - & 2x_1 & - & x_2 \\ y_3 & = & 36 & - & 1.5x_1 & - & 3x_2\ . \end{array} \tag{1.13}$$

Indeed, every choice of x_1 and x_2 determines a unique solution to the system. Correspondingly,

$$\begin{array}{rl} (x_1, x_2) & \text{are called } \textit{free variables}\text{, and} \\ (y_1, y_2, y_3) & \text{are called } \textit{dependent variables}. \end{array}$$

Clearly, the dimension of the solution space is determined by the number of free variables; hence, 2 for our example. Moreover, to each parameterized equation system, like (1.13), we associate a *basic solution*, namely the one obtained by setting all free variables to zero. Thus, the basic solution for (1.13) is $(x_1, x_2, y_1, y_2, y_3) = (0, 0, 40, 42, 36)$. By construction, such a basic solution lies on the intersection of m linearly independent constraints, i.e., the normal vectors of the constraints are linearly independent. Hence, if the basic solution is feasible, it corresponds to a vertex of the feasibility region $P = \{x \in \mathbb{R}^n_{\geq 0} \colon Ax \leq b\}$ of the original problem in canonical form. Thus, parameterized forms are a way to represent the equation system in a form highlighting a particular vertex solution. By moving from one parameterized form to another equivalent one, with different free and dependent variables, we can highlight different basic solutions. This is the way how the geometric idea of the Simplex Method of moving from vertex to vertex along edges will be realized algebraically.

Example 1.49

The tableau below corresponds to an equation system that is equivalent to system $\boxed{\text{I}}$, but with different dependent and free variables.

$$\boxed{\text{II}} \qquad \begin{array}{|ccccc|c} y_1 & y_2 & y_3 & x_1 & x_2 & 1 \\ \hline \frac{1}{4} & 0 & 0 & \frac{1}{4} & 1 & 10 \\ -\frac{1}{4} & 1 & 0 & \frac{7}{4} & 0 & 32 \\ -\frac{3}{4} & 0 & 1 & \frac{3}{4} & 0 & 6 \end{array}$$

The columns corresponding to the variables $B = (x_2, y_2, y_3)$ form an identity matrix. The solution space in its parametrized form with free variables (y_1, x_1) can directly be read from the tableau:

$$\begin{array}{rcrcrcr} x_2 & = & 10 & - & \frac{1}{4}y_1 & - & \frac{1}{4}x_1 \\ y_2 & = & 32 & + & \frac{1}{4}y_1 & - & \frac{7}{4}x_1 \\ y_3 & = & 6 & + & \frac{3}{4}y_1 & - & \frac{3}{4}x_1 \end{array}.$$

1.3.3 Equivalent linear systems

To move from one equation system to another equivalent one, *elementary row operations* can be used.

Definition 1.50: Elementary row operations

(i) Change the order of the equations.
(ii) Multiply an equation with a non-zero real number.
(iii) Add a multiple of one equation to another one.

One can easily observe that these operations do not change the set of solutions to a linear system.

Example 1.51

We illustrate these operations using our example, to show that the linear equation systems I and II indeed have the same set of solutions.

$$\boxed{\mathrm{I}} \qquad \begin{array}{|ccccc|c}
y_1 & y_2 & y_3 & x_1 & x_2 & 1 \\ \hline
1 & 0 & 0 & 1 & 4 & 40 \\
0 & 1 & 0 & 2 & 1 & 42 \\
0 & 0 & 1 & 1.5 & 3 & 36
\end{array}$$

Multiply the first equation by $\frac{1}{4}$:

$$\rightarrow \boxed{\mathrm{Ia}} \qquad \begin{array}{|ccccc|c}
y_1 & y_2 & y_3 & x_1 & x_2 & 1 \\ \hline
\frac{1}{4} & 0 & 0 & \frac{1}{4} & 1 & 10 \\
0 & 1 & 0 & 2 & 1 & 42 \\
0 & 0 & 1 & 1.5 & 3 & 36
\end{array}$$

Add $(-1)\times$ first row to the second row, and add $(-3)\times$ first row to the third row:

$$\boxed{\mathrm{II}} \qquad \begin{array}{|ccccc|c}
y_1 & y_2 & y_3 & x_1 & x_2 & 1 \\ \hline
\frac{1}{4} & 0 & 0 & \frac{1}{4} & 1 & 10 \\
-\frac{1}{4} & 1 & 0 & \frac{7}{4} & 0 & 32 \\
-\frac{3}{4} & 0 & 1 & \frac{3}{4} & 0 & 6
\end{array}$$

Definition 1.52: Equivalent equation systems

Systems of linear equations, which can be transformed to each other using elementary row operations, are called *equivalent*.

$\boxed{\mathrm{I}}$, $\boxed{\mathrm{Ia}}$, and $\boxed{\mathrm{II}}$ are equivalent systems and have therefore the same set of solutions.

Conversely, it is not hard to see that two linear equation systems with full row rank and identical solutions can be obtained from one another through elementary row operations.

As discussed, the equation system we obtained by transforming an LP given in canonical form into one in standard form has the special property that the coefficients of the slack variables in the system form an identity matrix. This form, which leads to a natural parameterization of the solution space as highlighted previously and allows us to talk about basic solutions, is typically referred to as the *tableau form* in the context of linear programming.

Definition 1.53: Tableau form

An equation system $Ax = b$ with m equations in $n+m$ variables is in *tableau form* if its coefficient matrix contains an identity matrix. In this case:

- A tuple of variables whose corresponding columns—listed in the same order in which they appear in the tuple—form an identity matrix, are called a *basis*.
- The variables used in the basis are called *basic variables*.
- All other variables are called *non-basic variables*.
- The columns corresponding to basic variables are called *basic columns*.
- The columns corresponding to non-basic variables are called *non-basic columns*.
- The *basic solution* corresponding to a basis is the unique solution obtained by setting all non-basic variables to zero; hence, the basic variables will be set to b.

Notice that, given an equation system in tableau form, it may have more than one basis. This is the reason why a basic solution is defined with respect to a basis and not with respect to an equation system in tableau form. Nevertheless, if the basis of an equation system in tableau form is unique, we also talk about *the basic solution* of the system.

As mentioned, for any values of the non-basic variables, there is a unique set of values for the basic ones that leads to a solution to the equation system. Hence, the non-basic variables correspond to what we called *free* variables and the basic ones to *non-free* variables.

Example 1.54

The following system of linear equations in tableau form has a unique basis, namely (y_1, y_2, y_3).

y_1	y_2	y_3	x_1	x_2	1
1	0	0	1	4	40
0	1	0	2	1	42
0	0	1	1.5	3	36

The system of linear equations shown below is not in tableau form because it does not contain an identity matrix.

y_1	y_2	y_3	x_1	x_2	1
0	1	0	2	1	42
$-\frac{1}{4}$	1	0	$\frac{7}{4}$	0	32
$-\frac{3}{4}$	0	1	$\frac{3}{4}$	0	6

However, it turns out to be equivalent to the first system. To check this, one can apply row operations to the second system to obtain the first one or vice versa. Applying row operations to a matrix is the same as multiplying the matrix from the left with a full-rank matrix. Clearly, up to column permutations, the only matrix $S \in \mathbb{R}^{3\times 3}$ that, when being multiplied with the above matrix, can lead to the first system, is the inverse of

$$\begin{pmatrix} 0 & 1 & 0 \\ -\frac{1}{4} & 1 & 0 \\ -\frac{3}{4} & 0 & 1 \end{pmatrix} ,$$

i.e., the matrix formed by the columns corresponding to y_1, y_2, and y_3. Hence, we need to choose

$$S = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{1}{4} & 1 & 0 \\ -\frac{3}{4} & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 4 & -4 & 0 \\ 1 & 0 & 0 \\ 3 & -3 & 1 \end{pmatrix} .$$

Finally, by multiplying S with the 3×6 matrix consisting of the entries (including its right-hand side) of the second tableau, which is not in standard form, we obtain the matrix corresponding to the first tableau. Hence, the two systems are indeed equivalent.
Moreover, the following system of equations in tableau form does not have a unique basis.

y_1	y_2	y_3	x_1	x_2	1
1	0	0	0	1	40
0	1	0	1	2	50
0	0	1	0	3	60

Indeed, there are two possible bases, namely (y_1, y_2, y_3) and (y_1, x_1, y_3).

Example 1.55

Consider the three equation systems I, Ia, and II in Example 1.51. System I is in tableau form with basis $B = (y_1, y_2, y_3)$. Hence, y_1, y_2, and y_3 are the basic variables. System II is in tableau form with basis $B = (x_2, y_2, y_3)$, whereas system Ia is *not* in tableau form.

Example 1.56

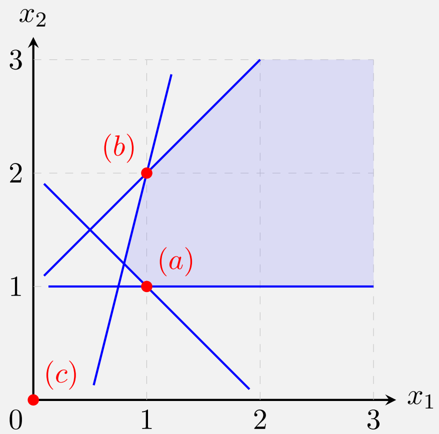
Consider the following LP, given in canonical form.

$$\begin{array}{lrcrcr} \max & -x_1 & - & x_2 & & \\ & & - & x_2 & \leq & -1 \\ & -x_1 & - & x_2 & \leq & -2 \\ & -4x_1 & + & x_2 & \leq & -2 \\ & -x_1 & + & x_2 & \leq & 1 \\ & x_1 & & & \geq & 0 \\ & & & x_2 & \geq & 0 \end{array}$$

By adding slack variables we obtain the equivalent standard form shown below.

$$
\begin{array}{lccccrcrcr}
\max & & & & & -x_1 & - & x_2 & & \\
& y_1 & & & & & - & x_2 & = & -1 \\
& & y_2 & & & -x_1 & - & x_2 & = & -2 \\
& & & y_3 & & -4x_1 & + & x_2 & = & -2 \\
& & & & y_4 & -x_1 & + & x_2 & = & 1 \\
& & & & & & & x & \in & \mathbb{R}^2_{\geq 0} \\
& & & & & & & y & \in & \mathbb{R}^4_{\geq 0}
\end{array}
$$

The graphic below shows the feasible region of the LP in the (x_1, x_2)-space.



Below, we present three equivalent equation systems, (a), (b), and (c) in tableau form that all correspond to the above-written LP in standard form. Each of these systems has a unique basis, and hence, each system has a well-defined basic solution, which is highlighted in the graphic above.
System (a) in tableau form with basic solution $(x_1, x_2, y_1, y_2, y_3, y_4) = (1, 1, 0, 0, 1, 1)$:

$$
(a) \qquad
\begin{array}{c|cccccc|c}
 & y_1 & y_2 & y_3 & y_4 & x_1 & x_2 & 1 \\
\hline
 & -1 & 0 & 0 & 0 & 0 & 1 & 1 \\
 & 1 & -1 & 0 & 0 & 1 & 0 & 1 \\
 & 5 & -4 & 1 & 0 & 0 & 0 & 1 \\
 & 2 & -1 & 0 & 1 & 0 & 0 & 1
\end{array}
$$

System (b) in tableau form with basic solution $(x_1, x_2, y_1, y_2, y_3, y_4) = (1, 2, 1, 1, 0, 0)$:

$$(b) \quad \begin{array}{|cccccc|c} y_1 & y_2 & y_3 & y_4 & x_1 & x_2 & 1 \\ \hline 0 & 0 & -\frac{1}{3} & \frac{4}{3} & 0 & 1 & 2 \\ 0 & 0 & -\frac{1}{3} & \frac{1}{3} & 1 & 0 & 1 \\ 0 & 1 & -\frac{2}{3} & \frac{5}{3} & 0 & 0 & 1 \\ 1 & 0 & -\frac{1}{3} & \frac{4}{3} & 0 & 0 & 1 \end{array}$$

System (c) in tableau form with basic solution $(x_1, x_2, y_1, y_2, y_3, y_4) = (0, 0, -1, -2, -2, 1)$:

$$(c) \quad \begin{array}{|cccccc|c} y_1 & y_2 & y_3 & y_4 & x_1 & x_2 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & -1 & -1 & -2 \\ 0 & 0 & 1 & 0 & -4 & 1 & -2 \\ 0 & 0 & 0 & 1 & -1 & 1 & 1 \end{array}$$

1.3.4 Exchange step/pivoting

Transforming one tableau into another can be done using *exchange steps*, which are also called *pivoting steps*. An exchange step is a sequence of elementary row operations that are uniquely determined by the variables "entering" and "leaving" the basis. Consider a tableau with basis B. Let $k \in [m+n]$ be the index of a non-basic column, and let $i \in [m]$ be a row such that $A_{ik} \neq 0$. The variable corresponding to column k will enter the basis and its column will be transformed to the i-th canonical vector $e_i \in \mathbb{R}^m$, i.e., for $j \in [m]$ we have $e_i(j) = 1$ if $i = j$ and $e_i(j) = 0$ otherwise. Therefore, the basic variable whose column is e_i will leave the basis, leading to a new basis B'.

The coefficient matrix A' and right-hand side b' of an equivalent tableau with basis B' can be obtained through the following rules, which simply correspond to a sequence of elementary row operations achieving the above exchange:

(i) $A'_{i\ell} = \dfrac{A_{i\ell}}{A_{ik}}$ $\qquad \ell \in [m+n]$

(ii) $A'_{j\ell} = A_{j\ell} - \dfrac{A_{jk} \cdot A_{i\ell}}{A_{ik}}$ $\qquad \ell \in [m+n], \qquad j \neq i, j \in [m]$

(iii) $b'_i = \dfrac{b_i}{A_{ik}}$

(iv) $b'_j = b_j - \dfrac{A_{jk} \cdot b_i}{A_{ik}}$ $\qquad j \in [m]$ with $j \neq i$,

where

- A_{ik} is called *pivot element*;
- the row with index i is called *pivot row*;
- the column with index k is called *pivot column*.

Typically, when talking about a pivot element A_{ik}, we do not just refer to the actual scalar A_{ik}, but also to the corresponding row index i and column index k. Due to this, it makes sense to talk about "performing an exchange step on the pivot A_{ik}", even if the matrix A contains multiple entries with the same value as A_{ik}.

Example 1.57

The unique basis of the tableau below is $B = (x_2, y_2, y_3)$.

$$\boxed{\text{II}} \qquad \begin{array}{|ccccc|c} y_1 & y_2 & y_3 & x_1 & x_2 & 1 \\ \hline \frac{1}{4} & 0 & 0 & \frac{1}{4} & 1 & 10 \\ -\frac{1}{4} & 1 & 0 & \frac{7}{4} & 0 & 32 \\ -\frac{3}{4} & 0 & 1 & \boxed{\frac{3}{4}} & 0 & 6 \end{array}$$

After an exchange step with $k = 4, i = 3$ one obtains the following equivalent system:

$$\boxed{\text{III}} \qquad \begin{array}{|ccccc|c} y_1 & y_2 & y_3 & x_1 & x_2 & 1 \\ \hline \frac{1}{2} & 0 & -\frac{1}{3} & 0 & 1 & 8 \\ \frac{6}{4} & 1 & -\frac{7}{3} & 0 & 0 & 18 \\ -1 & 0 & \frac{4}{3} & 1 & 0 & 8 \end{array}$$

The basis of this system is $B' = (x_2, y_2, x_1)$, and its parameterized solution set is the following:

$$\begin{array}{rcrcrcr} x_2 & = & 8 & - & \frac{1}{2}y_1 & + & \frac{1}{3}y_3 \\ y_2 & = & 18 & - & \frac{6}{4}y_1 & + & \frac{7}{3}y_3 \\ x_1 & = & 8 & + & y_1 & - & \frac{4}{3}y_3 \end{array}$$

Moreover, the basic solution of $\boxed{\text{III}}$ is $(x_1, x_2, y_1, y_2, y_3) = (8, 8, 0, 18, 0)$.

1.3.5 Short tableau

The short tableau is a more compact way of representing the tableau forms we have seen so far and has further important advantages. First, as already mentioned, it reduces the amount of data that has to be carried along when doing operations on the tableau. Second, it will prove very useful when we talk about the dual of a linear program, many of whose properties can be read more naturally from the short tableau. Last but not least, the short tableau avoids ambiguities in terms of what variables are used in the basis. To clearly distinguish the tableau form we already introduced form the short tableau, we will also sometimes talk about the long tableau to refer to the former.

For simplicity, we introduce the structure of a short tableau through an example. Consider the following long tableau with unique basis $B = (x_2, y_2, y_3)$.

$\boxed{\text{II}}$

$$\begin{array}{|ccccc|c}
y_1 & y_2 & y_3 & x_1 & x_2 & 1 \\
\hline
\frac{1}{4} & 0 & 0 & \frac{1}{4} & 1 & 10 \\
-\frac{1}{4} & 1 & 0 & \frac{7}{4} & 0 & 32 \\
-\frac{3}{4} & 0 & 1 & \frac{3}{4} & 0 & 6
\end{array}$$

By marking the i-th *row* of the tableau with the (basic) variable whose column is e_i and by deleting the corresponding unit vectors, we obtain a *short tableau with basis* B:

$\boxed{\text{IIb}}$

$$\begin{array}{c|cc|c}
 & y_1 & x_1 & 1 \\
\hline
x_2 & \frac{1}{4} & \frac{1}{4} & 10 \\
y_2 & -\frac{1}{4} & \frac{7}{4} & 32 \\
y_3 & -\frac{3}{4} & \frac{3}{4} & 6
\end{array}$$

$$B = (x_2, y_2, y_3)$$

1.3.6 Exchange step in short tableau

The exchange step we introduced for the long tableau can now be translated to the short one. For this consider a short tableau and let $A \in \mathbb{R}^{m\times n}$ be the matrix defining its entries, which all correspond to non-basic variables, and let b be the right-hand side of the short tableau. Hence, for the example $\boxed{\text{IIb}}$ above we have

$$A = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} \\ -\frac{1}{4} & \frac{7}{4} \\ -\frac{3}{4} & \frac{3}{4} \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 10 \\ 32 \\ 6 \end{pmatrix} .$$

An exchange step on the pivot $A_{ik} \neq 0$ in the short tableau leads to a new and equivalent short tableau with constraint matrix A' and right-hand side b' given by

$$
\begin{aligned}
A'_{ik} &= \frac{1}{A_{ik}} && (pivot\ element) && \\
A'_{jk} &= -\frac{A_{jk}}{A_{ik}} && (pivot\ column) && j \in [m], \quad j \neq i \\
A'_{i\ell} &= \frac{A_{i\ell}}{A_{ik}} && (pivot\ row) && \ell \in [n], \quad \ell \neq k \\
A'_{j\ell} &= A_{j\ell} - \frac{A_{jk} \cdot A_{i\ell}}{A_{ik}} && && j \in [m], \quad j \neq i \\
& && && \ell \in [n], \quad \ell \neq k \\
b'_i &= \frac{b_i}{A_{ik}} && && \\
b'_j &= b_j - \frac{A_{jk} \cdot b_i}{A_{ik}} && && j \in [m], \quad j \neq i \ .
\end{aligned}
\tag{1.14}
$$

Moreover, this pivoting step does not just change the entries of the short tableau, but also leads to a modification of the boundary, i.e., the variables assigned with the rows and columns. More precisely, the basic variable at row i swaps its role with the non-basic variable at column k.

Example 1.58

Consider performing an exchange step in system $\boxed{\text{IIb}}$ at the pivot element A_{32} as highlighted below.

$\boxed{\text{IIb}}$

	y_1	x_1	1
x_2	$\frac{1}{4}$	$\frac{1}{4}$	10
y_2	$-\frac{1}{4}$	$\frac{7}{4}$	32
y_3	$-\frac{3}{4}$	$\boxed{\frac{3}{4}}$	6

$B = (x_2, y_2, y_3)$

This leads to the following equivalent system.

	y_1	y_3	1
x_2	$\frac{1}{2}$	$-\frac{1}{3}$	8
y_2	$\frac{6}{4}$	$-\frac{7}{3}$	18
x_1	-1	$\frac{4}{3}$	8

with basis $B' = (x_2, y_2, x_1)$

Exercise 1.59

Verify the validity of the rules for performing an exchange step in a short tableau, i.e., show that they indeed give an equivalent linear system in tableau form with the claimed basis, by using the long tableau.

Example 1.60

For the following system of equations we can read off a tableau for the basis $B = (x_1, x_2, x_3)$:

$$\begin{array}{ccccccccc}
x_1 & & & + & 2x_4 & + & x_5 & = & 3\\
 & x_2 & & + & x_4 & - & x_5 & = & 0\\
 & & x_3 & + & x_4 & & & = & 2
\end{array}$$

$$B = (x_1, x_2, x_3) \qquad \begin{array}{c|cc|c}
 & x_4 & x_5 & 1\\ \hline
x_1 & 2 & \boxed{1} & 3\\
x_2 & 1 & -1 & 0\\
x_3 & 1 & 0 & 2
\end{array} \tag{a}$$

The corresponding basic solution for this basis is:

$$(x_1, x_2, x_3, x_4, x_5) = (3, 0, 2, 0, 0) \ .$$

We perform exchange steps to compute further basic solutions (the pivot element is marked).

(b):

$$\begin{array}{c|cc|c}
 & x_4 & x_1 & 1\\ \hline
x_5 & 2 & 1 & 3\\
x_2 & \boxed{3} & 1 & 3\\
x_3 & 1 & 0 & 2
\end{array}$$

$(0, 3, 2, 0, 3)$

(c):

$$\begin{array}{c|cc|c}
 & x_2 & x_1 & 1\\ \hline
x_5 & -\frac{2}{3} & \frac{1}{3} & 1\\
x_4 & \frac{1}{3} & \boxed{\frac{1}{3}} & 1\\
x_3 & -\frac{1}{3} & -\frac{1}{3} & 1
\end{array}$$

$(0, 0, 1, 1, 1)$

(d):

$$\begin{array}{c|cc|c}
 & x_2 & x_4 & \\ \hline
x_5 & \text{-1} & \text{-1} & 0\\
x_1 & 1 & 3 & 3\\
x_3 & 0 & 1 & 2
\end{array}$$

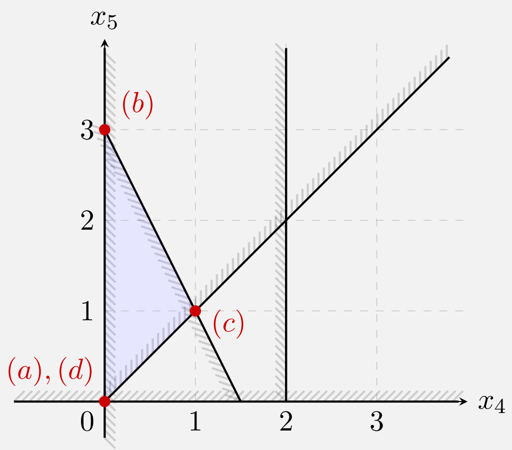
$(3, 0, 2, 0, 0)$

basic solutions as tuples $(x_1, x_2, x_3, x_4, x_5)$

Notice that the initial equation system corresponds to the standard form of an LP in canonical form with constraints given by

$$\begin{array}{ccccc}
2x_4 & + & x_5 & \leq & 3\\
x_4 & - & x_5 & \leq & 0\\
x_4 & & & \leq & 2\\
x_4 & & & \geq & 0\\
 & & x_5 & \geq & 0 \ .
\end{array}$$

Hence, we can visualize the feasible region of the equation system we started with, including non-negativity constraints, in the (x_4, x_5)-space:



The basic solutions corresponding to the tableaus (a), (b), (c), and (d) are highlighted in red in the above picture.

Example 1.60 also nicely exemplifies equivalent ways to look at an LP in terms of a different set of basic variables. More precisely, even though the original equation system corresponds to a 2-dimensional canonical LP in the variables x_4 and x_5, the other tableaus we obtained correspond to equivalent 2-dimensional LPs in a different set of variables. For example, tableau (b) corresponds to a 2-dimensional LP in the variables x_4 and x_1.

Exercise 1.61

Draw the 2-dimensional region of tuples $(x_2, x_3) \in \mathbb{R}^2$ that correspond to a feasible solution to the equation system in Example 1.60. Moreover, highlight the four basic solutions obtained in tableaus (a)–(d) in Example 1.60 in the (x_2, x_3)-space.

1.3.7 Simplex Method: phase II

The second phase of the Simplex Method requires a feasible basic solution to start with. It then performs exchange steps in a systematic way to either reach an optimal solution or to show that the given LP is unbounded.

To introduce the method, consider a general linear program in standard form obtained from one in canonical form, where we allow the objective function to contain additionally a constant term $q \in \mathbb{R}$, i.e.,

$$\begin{array}{llcllcl} \max & c^\top x & & & + \; q & & \\ & Ax & + & y & & = & b \\ & x & & & & \in & \mathbb{R}^n_{\geq 0} \\ & & & y & & \in & \mathbb{R}^m_{\geq 0} \;, \end{array}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. By introducing a variable $z \in \mathbb{R}$ for the objective, i.e., $z = c^\top x + q$, we can rewrite the problem as:

$$\begin{array}{lcccccl} \max\ z & & & & & & \\ & z & - & c^\top x & & & = & q \\ & & & Ax & + & y & = & b \\ & & & x & & & \in & \mathbb{R}^n_{\geq 0} \\ & & & & & y & \in & \mathbb{R}^m_{\geq 0}, \end{array} \tag{1.15}$$

which leads to a system of $m+1$ equations in the variables z, x, and y. Consider the tableau corresponding to (1.15) in matrix form:

$$\begin{array}{c|ccc|ccc|c} z & y_1 & \dots & y_m & x_1 & \dots & x_n & 1 \\ \hline 1 & 0 & \dots & 0 & -c_1 & \dots & -c_n & q \\ \hline 0 & & I & & & A & & b \end{array} \tag{1.16}$$

We will work with the short form of this tableau, which is given below.

$$\begin{array}{c|ccc|c} & x_1 & \dots & x_n & 1 \\ \hline z & -c_1 & \dots & -c_n & q \\ \hline y_1 & A_{11} & \dots & A_{1n} & b_1 \\ \vdots & \vdots & & \vdots & \vdots \\ y_m & A_{m1} & \dots & A_{mn} & b_m \end{array} \tag{1.17}$$

In the following, the auxiliary variable z will always stay in the basis. Therefore, when we refer to a *basic solution* or *basic variables*, this will always be with respect to the remaining variables, i.e., x and y. We highlight the special role of the variable z in the tableau by separating the equation for the objective function with a horizontal line. Still, when we perform an exchange step on a tableau as shown in (1.17), the exchange step is performed as we previously introduced it, i.e., over all rows of the tableau including the first row $-c^\top$, which corresponds to the objective.

Definition 1.62: Feasible tableau

The tableau (1.17) is called *feasible* if $b_1, \dots, b_m \geq 0$, i.e., if the corresponding basic solution is feasible.

Definition 1.63: Value of tableau

The entry q in a tableau (see 1.17) is called the *value of the tableau*. This is the objective value of the basic solution that corresponds to the tableau. We use this terminology both for feasible and infeasible tableaus.

Example 1.64: Feasible tableau

The tableau below corresponds to the linear program shown in (1.12). It is a feasible tableau.

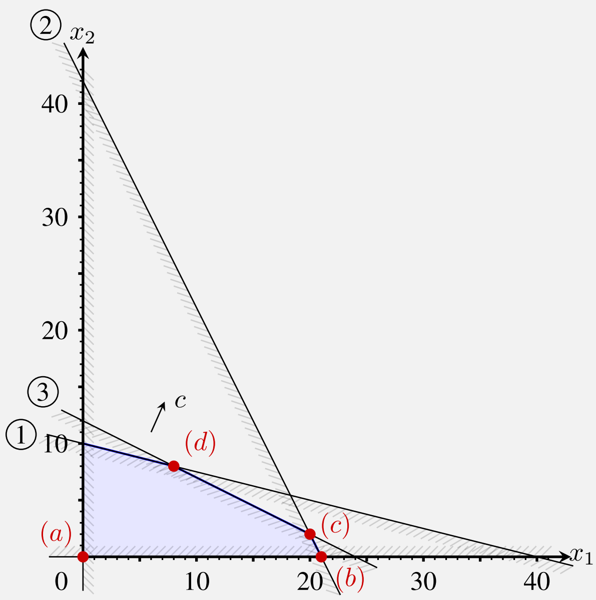
	x_1	x_2	1
z	-400	-900	0
y_1	1	4	40
y_2	2	1	42
y_3	1.5	3	36

The second phase of the Simplex Method starts with a feasible tableau and iteratively computes new feasible tableaus. Hence, the method never encounters an infeasible tableau. Given a feasible tableau, the goal is to move to another one with strictly higher objective value through a single exchange step. This will not always be possible. Hence, the second phase of the Simplex Method may sometimes pivot from one feasible tableau to another one with identical objective value. However, it will never pivot to a tableau with strictly worse objective value.

Before introducing a formal rule of how to perform exchange steps, we show in the example below how a well-chosen sequence of exchange steps solves the example linear program highlighted in (1.12).

Example 1.65

We recall the linear program shown in (1.12) in canonical form together with a graphical representation of its solution set.



The following sequence of tableaus, obtained through successive exchange steps, leads to the unique optimal solution of this LP. The corresponding basic feasible solutions are highlighted in the graphic above.

	x_1	x_2	1
z	-400	-900	0
y_1	1	4	40
y_2	$\boxed{2}$	1	42
y_3	1.5	3	36

(a)

$\longrightarrow$

	y_2	x_2	1
z	200	-700	8400
y_1	$-\frac{1}{2}$	$\frac{7}{2}$	19
x_1	$\frac{1}{2}$	$\frac{1}{2}$	21
y_3	$-\frac{3}{4}$	$\boxed{\frac{9}{4}}$	$\frac{9}{2}$

(b)

$\downarrow$

	y_2	y_3	1
z	$-\frac{100}{3}$	$\frac{2800}{9}$	9800
y_1	$\boxed{\frac{2}{3}}$	$-\frac{14}{9}$	12
x_1	$\frac{2}{3}$	$-\frac{2}{9}$	20
x_2	$-\frac{1}{3}$	$\frac{4}{9}$	2

(c)

$\longleftarrow$

	y_1	y_3	1
z	50	$\frac{700}{3}$	10400
y_2	$\frac{3}{2}$	$-\frac{7}{3}$	18
x_1	-1	$\frac{4}{3}$	8
x_2	$\frac{1}{2}$	$-\frac{1}{3}$	8

(d)

Notice that in this example, each of the 3 exchange steps strictly improved the objective value of the tableau, with the last tableau revealing the optimal objective value of 10400. Moreover, it is not hard to see that the last tableau corresponds to an optimal solution. Indeed, the objective row of the last tableau reveals that the objective value z satisfies

$$z = 10400 - 50y_1 - \frac{700}{3}y_2 \ .$$

Because all variables, including y_1 and y_2, are non-negative, this implies that no feasible solution of value higher than 10400 can be obtained. Finally, 10400 is the objective value attained by the basic solution that corresponds to the last tableau; hence, this solution is optimal.

To formally introduce a rule for selecting the pivot, consider a general feasible tableau as shown below, where we denote by x_B and x_N the basic and non-basic variables, respectively. Notice that there are always m many basic variables and n many non-basic variables. The location of the variables x_B and x_N is called the *boundary* of the tableau.

$$\begin{array}{c|ccc|c} & & x_N^\top & & 1 \\ \hline z & c_1 & \dots & c_n & q \\ \hline & A_{11} & \dots & A_{1n} & b_1 \\ x_B & \vdots & & \vdots & \vdots \\ & A_{m1} & \dots & A_{mn} & b_m \end{array} \tag{1.18}$$

When we obtain the tableau from the canonical form of an LP by transforming it first into standard form, then the variables x_B correspond to the slack variables, which we often denote by y. Moreover, because we consider a feasible tableau, we have that the right-hand side vector $b \in \mathbb{R}^m$ is non-negative.

Definition 1.66: Legal pivot element for phase II of Simplex Method

Given a feasible tableau, as shown in (1.18), a tuple $(i,k) \in [m] \times [n]$ corresponds to a legal pivot element A_{ik} for phase II of the Simplex Method if

(i) $c_k < 0$, and
(ii) $i \in \operatorname{argmin}\left\{\frac{b_j}{A_{jk}} : j \in [m] \text{ with } A_{jk} > 0\right\}$.

Pivot elements in phase II of the Simplex Method are typically chosen by first selecting the pivot column k, satisfying condition (i), and then the pivot row i to fulfill (ii). Condition (ii) is often called the *quotient rule*. Notice that by first choosing the pivot column k before choosing the row i, it may be that there is no row i that fulfills the quotient rule for column k, even though there may be a legal pivot element in another column. As we will see later, if this happens, then we do not have to look for another pivot column k, but can stop right-away because such a column turns out to prove unboundedness of the LP. Before expanding on this and further aspects, the following result motivates the definition of a legal pivot element.

Theorem 1.67

Consider a feasible tableau with a legal pivot element A_{ik} for phase II of the Simplex Method. Then the new tableau obtained after pivoting on A_{ik} satisfies the following:

(i) The new tableau is feasible.
(ii) The value of the new tableau is no less than that of the original one.
(iii) If $b_i > 0$, the value of the new tableau is strictly larger than that of the original one.

Proof.

(i) To check feasibility of the new tableau, we have to show that its right-hand side vector $b' \in \mathbb{R}^m$ is non-negative. We recall that b' is given by the pivot rules (1.14). Hence, for the pivot row we have

$$b'_i = \frac{b_i}{A_{ik}} \geq 0 \ ,$$

as $A_{ik} > 0$ because it is a legal pivot element and $b_i \geq 0$ because the original tableau is feasible. Moreover, for any other row $j \in [m] \setminus \{i\}$, the new right-hand side b'_j is given by

$$b'_j = b_j - \frac{A_{jk} \cdot b_i}{A_{ik}} \ . \tag{1.19}$$

If $A_{jk} \leq 0$, then we have $b'_j \geq b_j \geq 0$ as desired. Otherwise, if $A_{jk} > 0$, then, by the fact that A_{ik} is a legal pivot, we have

$$\frac{b_i}{A_{ik}} \leq \frac{b_j}{A_{jk}} \enspace ,$$

which, together with (1.19), again implies $b'_i \geq 0$, as desired.

(ii) When computing an exchange step, the row corresponding to the objective is treated in the same way as the other non-pivot rows. Hence, to compute the new objective value q', the same pivot rule applies as the one used to compute the new right-hand side of a non-pivot row:

$$q' = q - \frac{c_k \cdot b_i}{A_{ik}} \geq q \enspace , \tag{1.20}$$

where the inequality follows from $c_k < 0$ (legal pivot), $b_i \geq 0$ (feasibility of original tableau), and $A_{ik} > 0$ (legal pivot).

(iii) The above reasoning also immediately implies that $q' > q$ if $b_i > 0$.

□

Notice that the conditions for a pivot element to be legal, i.e., Definition 1.66, leave significant freedom in terms of which pivot to choose. Indeed, different rules of which legal pivot to choose lead to different variations of the Simplex Method. One very common rule is to choose the pivot column k such that c_k is the most negative value among all columns. This choice guarantees the largest *marginal increase* in the objective in the sense of how the objective changes per unit of increase of the non-basic variable corresponding to the pivot column.

Moreover, it may be that no legal pivot element exists. This can happen either because $c_k \geq 0$ for $k \in [n]$, or because for every column $k \in [n]$ with $c_k < 0$, there is no row $i \in [m]$ with $A_{ik} > 0$. In this case, the statement below implies that either the current basic solution is optimal or the problem is unbounded.

Theorem 1.68

Consider a feasible tableau (given as in (1.18)).

(i) If $c_k \geq 0 \ \forall k \in [n]$, then the basic solution corresponding to (1.18) is an *optimal solution* to the corresponding LP.

(ii) If $\exists k \in [n]$ with $c_k < 0$ and $A_{jk} \leq 0 \ \forall j \in [m]$, then the underlying LP is unbounded.

Moreover, a certificate of unboundedness can be obtained as follows. For $\lambda \in \mathbb{R}_{\geq 0}$, let x_B^λ, x_N^λ be the following assignments to the basic and non-basic variables, respectively:

$$\begin{aligned} x_B^\lambda &= b - \lambda A_{\cdot k} \\ (x_N^\lambda)_k &= \lambda \\ (x_N^\lambda)_\ell &= 0 \quad \forall \ell \in [n] \setminus \{k\} \ , \end{aligned}$$

where $A_{\cdot k}$ is the k-th column of A and $(x_N^\lambda)_k$ is the value of the non-basic variable corresponding to that column. Then, for any $\lambda \in \mathbb{R}_{\geq 0}$, $x^\lambda := \binom{x_B^\lambda}{x_N^\lambda}$ is feasible with objective value going to ∞ as $\lambda \to \infty$.

Proof.

(i) We can read off from the tableau that the objective value z satisfies

$$z = q - c^\top x_N \ .$$

Notice that the above expression is always at most q because $c \geq 0$ and $x_N \geq 0$. Hence, no feasible solution can have objective value strictly larger than q. Because the basic solution for the current feasible tableau has objective value equal to q, it is optimal.

(ii) To show this point, we only have to show that the claimed certificate of unboundedness is correct. To this end, we first observe that for any $\lambda \in \mathbb{R}_{\geq 0}$, the point x^λ is a solution by construction, i.e., it fulfills all equality constraints. Moreover, x^λ is non-negative. This is obvious for the non-basic variables, and follows for the basic variables by observing that $b \geq 0$, due to feasibility of the tableau, and $A_{\cdot k} \leq 0$ by assumption. Finally, the objective value achieved by x^λ is

$$z = q - \lambda c_k \ ,$$

which indeed goes to ∞ for $\lambda \to \infty$.

□

Theorem 1.68 covers the case where no legal pivot element exists anymore. Nevertheless, Theorem 1.68 may apply even though there still is a legal pivot element in the tableau. More precisely, there may be a column $k \in [n]$ satisfying point (ii) of Theorem 1.68, even though a legal pivot element exists in a different column. The fact that Theorem 1.68 also applies in this case is convenient, because it allows us to first fix a column k with $c_k < 0$ and then only focus on that column for legal pivots. As we will see, this is precisely what the Simplex Method does.

Theorem 1.68 naturally leads to the following terminology.

Definition 1.69: Optimal and unbounded tableaus

A feasible tableau (as in 1.18) is called *optimal* if $c_k \geq 0\ \forall k \in [n]$. It is called *unbounded* if $\exists k \in [n]$ with $c_k < 0$ and $A_{jk} \leq 0\ \forall j \in [m]$.

Example 1.70

To exemplify the above statements, we start with the tableau below and perform legal pivots as long as possible.

The first tableau below is a feasible starting tableau, and we choose x_2 as the variable entering the basis, i.e., we want to pivot on the column corresponding to x_2. The corresponding quotients for this column are indicated to the right of the tableau.

$$
\begin{array}{c|rr|l}
 & x_1 & x_2 & 1 \\
\hline
z & -400 & -900 & 0 \\
\hline
y_1 & 1 & \boxed{4} & 40 \\
y_2 & 2 & 1 & 42 \\
y_3 & 1.5 & 3 & 36
\end{array}
\qquad
\begin{array}{lll}
\text{quotient} & \frac{b_j}{A_{jk}} & \\
 & & \\
10 & \leftarrow & \text{minimum} \\
42 & & \\
12 & &
\end{array}
$$

After performing an exchange step with respect to the indicated pivot, we obtain the tableau below, which, due to Theorem 1.67 (i), is feasible because the pivot was legal.

$$
\begin{array}{c|rr|r}
 & x_1 & y_1 & 1 \\
\hline
z & -\frac{700}{4} & \frac{900}{4} & 9000 \\
\hline
x_2 & \frac{1}{4} & \frac{1}{4} & 10 \\
y_2 & \frac{7}{4} & -\frac{1}{4} & 32 \\
y_3 & \boxed{\frac{3}{4}} & -\frac{3}{4} & 6
\end{array}
\qquad
\begin{array}{lll}
\text{quotient} & & \\
 & & \\
40 & & \\
128/7 & & \\
8 & \leftarrow & \text{minimum}
\end{array}
$$

The current feasible basic solution is

$$(x_1, x_2, y_1, y_2, y_3) = (0, 10, 0, 32, 6)$$

with objective value $z = 9000$. Notice that the objective value strictly increased from the value of the original tableau, which was 0. The strict increase in objective value is guaranteed by Theorem 1.67 (iii) in this case.

We continue with a pivot on the circled element to obtain the optimal tableau below.

	y_3	y_1	1
z	$\frac{700}{3}$	50	10400
x_2	$-\frac{1}{3}$	$\frac{1}{2}$	8
y_2	$-\frac{7}{3}$	$\frac{6}{4}$	18
x_1	$\frac{4}{3}$	-1	8

The corresponding basic solution of this tableau, which is optimal due to Theorem 1.68 (i), is

$$(x_1, x_2, y_1, y_2, y_3) = (8, 8, 0, 18, 0)$$

with optimal objective value $z = 10400$. Again, the strict increase in objective value compared to the previous tableau was guaranteed by Theorem 1.67 (iii).

We are now ready to formally state phase II of the Simplex Method.

Algorithm 1: Phase II of Simplex Method

Input: Feasible tableau as shown in (1.18).

1. **Choice of pivot:**
   (a) *Choice of pivot column* (variable entering basis):
   If $c_k \geq 0\ \forall k \in [n]$, then **stop**. (The current basic solution is optimal due to Theorem 1.68 (i).) Otherwise, choose $k \in [n]$ with $c_k < 0$.
   (b) *Choice of pivot row* (variable leaving basis):
   If $A_{jk} \leq 0\ \forall j \in [m]$, then stop. (The problem is unbounded, see Theorem 1.68 (ii).) Otherwise, choose
   $$i \in \operatorname{argmin}\left\{\frac{b_j}{A_{jk}} : j \in [m] \text{ with } A_{jk} > 0\right\} .$$
2. **Exchange step:**
   Perform an exchange step on pivot element A_{ik} and **go back** to step 1.

As previously mentioned, the above description leaves open which pivot column or row to choose if there are several options fulfilling the above conditions. One of the most important questions in the field is whether there exists a good way to select the pivot column and row in the Simplex Method to obtain a polynomial-time algorithm. Interestingly, depending on how the pivot column and row are chosen in the Simplex Method, it is not even clear whether the algorithm terminates. We will get back to that point in Section 1.3.8.

Example 1.71: Phase II of Simplex Method

In this example, we apply phase II of the Simplex Method to solve the following LP given in canonical form.

$$\begin{array}{rrcrcl} \max & -3x_1 & + & 2x_2 & & \\ & x_1 & - & x_2 & \leq & 1 \\ & -2x_1 & + & x_2 & \leq & 1 \\ & -x_1 & + & x_2 & \leq & 2 \\ & x_1 & & & \geq & 0 \\ & & & x_2 & \geq & 0 \end{array}$$

We first transform it into standard form by introducing three slack variables y_1, y_2, and y_3.

$$\begin{array}{rlccccccccc} \max & z = & & & & - & 3x_1 & + & 2x_2 & & \\ & & y_1 & & & + & x_1 & - & x_2 & = & 1 \\ & & & y_2 & & - & 2x_1 & + & x_2 & = & 1 \\ & & & & y_3 & - & x_1 & + & x_2 & = & 2 \\ & & & & & & & & x & \in & \mathbb{R}^2_{\geq 0} \\ & & & & & & & & y & \in & \mathbb{R}^3_{\geq 0} \end{array}$$

We now apply phase II of the Simplex Method. For this example, there is a unique legal pivot element at each step, which is circled in the tableaus below.

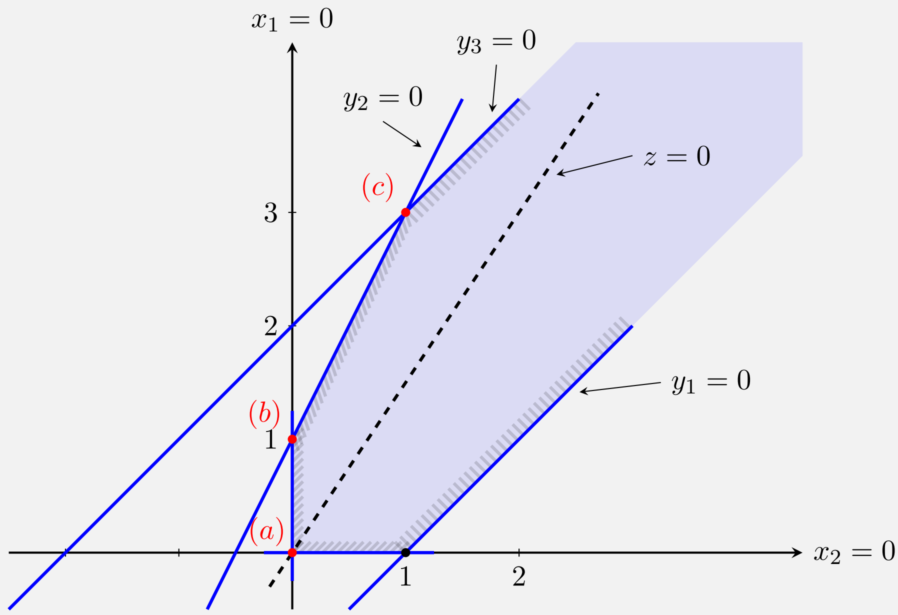
$$\begin{array}{c|cc|c} & x_1 & x_2 & 1 \\ \hline z & 3 & -2 & 0 \\ \hline y_1 & 1 & -1 & 1 \\ y_2 & -2 & \boxed{1} & 1 \\ y_3 & -1 & 1 & 2 \end{array} \quad \rightarrow \quad \begin{array}{c|cc|c} & x_1 & y_2 & 1 \\ \hline z & -1 & 2 & 2 \\ \hline y_1 & -1 & 1 & 2 \\ x_2 & -2 & 1 & 1 \\ y_3 & \boxed{1} & -1 & 1 \end{array} \quad \rightarrow \quad \begin{array}{c|cc|c} & y_3 & y_2 & 1 \\ \hline z & 1 & 1 & 3 \\ \hline y_1 & 1 & 0 & 3 \\ x_2 & 2 & -1 & 3 \\ x_1 & 1 & -1 & 1 \end{array}$$

tableau (a) tableau (b) tableau (c)

The last tableau is an optimal tableau because the coefficients of the non-basic variables in the objective row, namely two 1s, are both non-negative. Hence, the corresponding basic solution is optimal. It is

$$(x_1, x_2, y_1, y_2, y_3) = (1, 3, 3, 0, 0) \ .$$

The graphic below highlights, in the (x_1, x_2)-space, the vertices of the feasible region that got traversed by this run of the Simplex Method.



If we replace the objective function and maximize $\overline{z} = x_2$ instead of $-3x_1 + 2x_2$, then the algorithm will stop with the tableau below.

tableau (c')

	y_3	y_2	
$\overline{z}$	2	-1	3
y_1	1	0	3
x_2	2	-1	3
x_1	1	-1	1

This tableau is unbounded. As highlighted in Theorem 1.68, we can read off an unbounded half-line as follows. The half-line starts at the current basic solution, which is

$$\begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 3 \\ 0 \\ 0 \end{pmatrix} .$$

Moreover, the unbounded improving direction can be read off the column corresponding to y_2, and is

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} .$$

Hence, the following improving half-line, described in the space $(x_1, x_2, y_1, y_2, y_3)$, is an unboundedness certificate:

$$\begin{pmatrix}1\\3\\3\\0\\0\end{pmatrix} + \lambda \cdot \begin{pmatrix}1\\1\\0\\1\\0\end{pmatrix} \qquad \text{for } \lambda \in \mathbb{R}_{\geq 0} \enspace .$$

By only considering the first two coordinates, we obtain an unboundedness certificate in the original (x_1, x_2)-space:

$$\begin{pmatrix}1\\3\end{pmatrix} + \lambda \cdot \begin{pmatrix}1\\1\end{pmatrix} \qquad \text{for } \lambda \in \mathbb{R}_{\geq 0} \enspace .$$

Exercise 1.72

Verify that one can indeed obtain tableau (c') in Example 1.71 through pivoting steps when starting with the modified objective $\max \overline{z} = x_2$.

Degeneracy

One of the most central questions we left open so far, is whether the Simplex Method terminates in a finite number of steps. It turns out that this is closely related to the notion of degeneracy, which we already saw in the context of polyhedra, and can be rephrased in terms of tableaus and basic solutions as follows.

Definition 1.73

The basic solution to a tableau (given as in 1.18) is called *degenerate*, if there is an index $i \in [m]$ such that $b_i = 0$. In this case, the tableau is also called *degenerate*.

This notion of degeneracy, defined in the context of tableaus, is indeed tightly linked to the notion of degeneracy that we encountered in the context of polyhedra, as highlighted in the following statement.

Proposition 1.74

A feasible tableau is degenerate if and only if its basic solution is a degenerate vertex of the feasible region of the problem (with respect to either the canonical and standard form).

Proof. Assume that

$$\begin{array}{rrcl} \max & c^\top x & & \\ & Ax & \leq & b \\ & x & \geq & 0 \end{array} \tag{1.21}$$

is the LP we started with in canonical form, and

$$\begin{array}{lrcrcl} \max & c^\top x & & & & \\ & Ax & + & y & = & b \\ & x & & & \geq & 0 \\ & & & y & \geq & 0 \end{array} \tag{1.22}$$

is the standard form of the LP we used to build the tableau. Now let $(\overline{x}, \overline{y})$ be the basic solution to a feasible tableau, and let η be the total number of zeros in $\overline{x}$ and $\overline{y}$. Notice that the number of tight constraints among the original constraints in canonical form, i.e., $Ax \leq b$ and $x \geq 0$, is equal to η. Hence, $\overline{x}$, which is a vertex of

$$P := \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\} \enspace ,$$

is degenerate if and only if $\eta > n$. Notice that in any basic solution, all n-many non-basic variables are set to zero. Hence, we have $\eta > n$ if and only if at least one basic variable is set to zero. This is the case if and only if the tableau is degenerate.

Finally, we highlight that the same statement also holds when considering the point $\left(\begin{smallmatrix}\overline{x}\\ \overline{y}\end{smallmatrix}\right)$ as a vertex of

$$Q := \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{n+m} : Ax + y = b, x \geq 0, y \geq 0 \right\} \enspace ,$$

i.e., the feasible region for the LP in standard form. Indeed, the number of tight constraints for $\left(\begin{smallmatrix}\overline{x}\\ \overline{y}\end{smallmatrix}\right)$ in the description of Q is equal to $m + \eta$, which leads to degeneracy in the $(n+m)$-dimensional space $\mathbb{R}^{n+m}$ if and only if $\eta > n$, as in the canonical form. □

We now observe that without degeneracy, any realization of phase II of the Simplex Method, as described in Algorithm 1, will be successful, i.e., it will terminate. We recall that when talking about a "realization" of the algorithm, we mean a specific way to choose the pivot column and row in case of multiple options.

Theorem 1.75: Finiteness of Simplex Method in non-degenerate case

If phase II of the Simplex Method never encounters a degenerate tableau, then it will stop after a finite number of steps. (This holds no matter how the pivot column and row is chosen if there are several legal options.)

Proof. Whenever phase II of the Simplex Method does an exchange step, the value of the tableau strictly improves because of Theorem 1.67 (iii). (Notice that non-degeneracy of the tableau guarantees that the right-hand side b_i of the pivot row is strictly positive, which is required by Theorem 1.67 (iii).) This implies that we will never encounter the same tableau twice. However, for any given problem, there is only a finite number of possible tableaus, because there are only finitely many options to partition all variables into a basic and non-basic group. Hence, the algorithm must terminate. □

Thus, if the inequality description $Ax \leq b, x \geq 0$ of an LP in canonical form is not degenerate, then phase II of the Simplex Method will always terminate in a finite number of steps. This is a consequence of Theorem 1.75 together with Proposition 1.74.

To summarize what we discussed so far, if phase II of the Simplex Method never encounters a degenerate tableau, then it will stop, which implies that we will either have found an optimal solution or shown that the problem is unbounded. However, to obtain a working algorithm for solving general linear programs, we also need to be able to handle the case of degenerate basic solutions.

1.3.8 Cycling and Bland's rule

Even though, due to Theorem 1.75, we know that phase II of the Simplex Method will stop if there is no degeneracy, it is not yet clear whether degeneracy is actually a problem for the algorithm or whether it is merely an issue for the analysis we employed. It turns out that degeneracy can indeed lead to Simplex Methods that do not terminate. Notice that if the Simplex Method does not terminate, then it will visit some tableaus an infinite number of times. This is called *cycling*. The terminology of cycling is even more perspicuous when considering a Simplex Method with a deterministic rule for choosing the pivot row and column that does not terminate. In this case, the first time the same tableau is encountered twice, the method will keep cycling through the precise same exchange steps performed between these two encounters.

We start with an example that highlights the issue of cycling in degenerate tableaus.

Example 1.76: Cycling of Simplex Method

Consider the following linear program.

$$\begin{array}{rcccccccccccl} \max\ z = & & & & 2x_3 & + & 2x_4 & - & 8x_5 & - & 2x_6 & & \\ & & x_2 & - & 7x_3 & - & 3x_4 & - & 7x_5 & + & 2x_6 & = & 0 \\ & x_1 & & + & 2x_3 & + & x_4 & - & 3x_5 & - & x_6 & = & 0 \\ & & & & & & & & & & x & \in & \mathbb{R}^6_{\geq 0} \end{array}$$

We will apply phase II of the Simplex Method to it with the following rule for choosing a pivot element. When there are several options for choosing the pivot column or row, we always choose the leftmost or topmost one, respectively. This leads to the following tableaus.

$$(a)\quad \begin{array}{c|cccc|c} & x_3 & x_4 & x_5 & x_6 & 1 \\ \hline z & -2 & -2 & 8 & 2 & 0 \\ \hline x_2 & -7 & -3 & 7 & 2 & 0 \\ x_1 & \boxed{2} & 1 & -3 & -1 & 0 \end{array} \qquad (b)\quad \begin{array}{c|cccc|c} & x_1 & x_4 & x_5 & x_6 & 1 \\ \hline z & 1 & -1 & 5 & 1 & 0 \\ \hline x_2 & \frac{7}{2} & \boxed{\frac{1}{2}} & -\frac{7}{2} & -\frac{3}{2} & 0 \\ x_3 & \frac{1}{2} & \frac{1}{2} & -\frac{3}{2} & -\frac{1}{2} & 0 \end{array}$$

$$(c)\quad \begin{array}{c|cccc|c} & x_1 & x_2 & x_5 & x_6 & 1 \\ \hline z & 8 & 2 & -2 & -2 & 0 \\ \hline x_4 & 7 & 2 & -7 & -3 & 0 \\ x_3 & -3 & -1 & \boxed{2} & 1 & 0 \end{array} \qquad (d)\quad \begin{array}{c|cccc|c} & x_1 & x_2 & x_3 & x_6 & 1 \\ \hline z & 5 & 1 & 1 & -1 & 0 \\ \hline x_4 & -\frac{7}{2} & -\frac{3}{2} & \frac{7}{2} & \boxed{\frac{1}{2}} & 0 \\ x_5 & -\frac{3}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \end{array}$$

(e)

	x_1	x_2	x_3	x_4	1
z	-2	-2	8	2	0
x_6	-7	-3	7	2	0
x_5	$\boxed{2}$	1	-3	-1	0

(f)

	x_5	x_2	x_3	x_4	1
z	1	-1	5	1	0
x_6	$\frac{7}{2}$	$\boxed{\frac{1}{2}}$	$-\frac{7}{2}$	$-\frac{3}{2}$	0
x_1	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{3}{2}$	$-\frac{1}{2}$	0

(g)

	x_5	x_6	x_3	x_4	1
z	8	2	-2	-2	0
x_2	7	2	-7	-3	0
x_1	-3	-1	2	1	0

After six exchange steps, we again find a tableau with basis $B = (x_2, x_1)$. This is not the precise same tableau as (a) because the order of the columns is different. However, one can check that the next six exchange steps will choose the precise same entering and leaving variables as the first six and result in the original tableau (a). Hence, the tableaus will keep changing in this cyclic order with the employed pivot rule.

One of the easiest pivot selection rules to avoid cycling is *Bland's rule*, which is defined as follows.

Definition 1.77: Bland's pivot rule

All variables are first ordered in an arbitrary way (e.g., according to increasing index). Whenever a pivot column or row is to be selected and there are several options, we choose the column or row corresponding to the variable appearing first in the fixed order.

Hence, if we are given a tableau with variable $x_1, \dots, x_k$ and apply Bland's rule (using the order induced by the indices), then, among all possible pivot columns, we select the one whose corresponding variable has smallest index, and do the same in the next step for selecting the pivot row.

Theorem 1.78

When applying Bland's rule for choosing the pivot, phase II of the Simplex Method does not cycle. More precisely, with Bland's rule, the Simplex Method will never encounter two tableaus with the same set of basic (and therefore also non-basic) variables.

Example 1.79

Using Bland's rule in Example 1.76 leads to a different pivot choice in tableau (f):

$$(f)\quad \begin{array}{c|cccc|c} & x_5 & x_2 & x_3 & x_4 & 1 \\ \hline z & 1 & -1 & 5 & 1 & 0 \\ \hline x_6 & \frac{7}{2} & \frac{1}{2} & -\frac{7}{2} & -\frac{3}{2} & 0 \\ x_1 & \frac{1}{2} & \boxed{\frac{1}{2}} & -\frac{3}{2} & -\frac{1}{2} & 0 \end{array}$$

This results in the following optimal tableau:

$$\begin{array}{c|cccc|c} & x_5 & x_1 & x_3 & x_4 & 1 \\ \hline z & 2 & 2 & 2 & 0 & 0 \\ \hline x_6 & 3 & -1 & -2 & -1 & 0 \\ x_2 & 1 & 2 & -3 & -1 & 0 \end{array}$$

Notice that both in Example 1.76 and also here we always pivoted on a row whose right-hand side was zero. Consequently, all tableaus we encountered have identical basic solutions. (Though they have different bases.) Hence, in this special case, the basic solution of tableau (a) was already optimal. (Even though the tableau is not an optimal one!) However, it is not hard to modify the example such that after breaking out of the cycling with Bland's rule, one can still find tableaus with strictly higher value through further exchange steps.

Proof of Theorem 1.78. Assume for the sake of obtaining a contradiction that we cycle with Bland's rule. Recall that Bland's rule is deterministic, hence we will always cycle in the same order through the same tableaus. Some variables will sometimes be basic and sometimes non-basic, whereas others may always be either basic or non-basic. We first observe that we can assume that no variables of the latter type exist, i.e., no variable will always be basic or always be non-basic. Indeed, consider one tableau we encounter when cycling, and assume that such a variable exists, say an always basic one. Then we can simply delete the row corresponding to the variable from the tableau and obtain a smaller tableau where Bland's rule cycles. Clearly, because the pivot elements in the cycle are never chosen on the row corresponding to the deleted basic variable, the selection of the pivot elements does not change. Analogously, we can delete from a tableau any column corresponding to a variable that is always non-basic. Hence, we can assume that we have a cycling example for Bland's rule where each variable is sometimes basic and sometimes non-basic in the tableaus encountered in the cycle.

An important observation is that when cycling, the basic solution never changes. Because each variable is at least once non-basic on the cycle, and hence has value 0 in the corresponding basic solution, this implies that the basic solution to all tableaus encountered in the cycle is the all-zeros vector. In particular, the right-hand side of any tableau encountered in the cycle consists of zeros only. Moreover, the constant entry in the objective function row always remains the same; hence, we can assume without loss of generality that it is zero.

Let $x_1, \ldots, x_k$ be all variables involved in the tableau, and the numbering is chosen such that Bland's rule always chooses the variable with the smallest index if there are several options. Now consider a tableau in the cycle in which x_k is basic and is chosen as the basis-leaving variable, i.e., the pivot element is chosen in the row of x_k. We denote this tableau as tableau (a) and let x_i be the basis-entering variable. Figure 1.12 highlights the signs of the entries in the column corresponding to x_i.

$$
(a) \qquad
\begin{array}{c|c|c}
 & x_i & \\
\hline
z & - & 0 \\
\hline
 & \ominus & 0 \\
 & \vdots & \vdots \\
 & \ominus & 0 \\
x_k & + & 0 \\
 & \ominus & 0 \\
 & \vdots & \vdots \\
 & \ominus & 0
\end{array}
$$

Figure 1.12: Tableau (a) shown above is a tableau encountered during cycling where, in the next exchange step, the basis-leaving variable is x_k. We denote by x_i the basis-entering variable. As discussed, all right-hand sides are 0. Moreover, we highlight the signs of the values in the column corresponding to x_i. Here, '$-$' stands for a strictly negative number, '$+$' for a strictly positive one, and '$\ominus$' for a non-positive one. The signs follow by the fact that we choose a legal pivot element, which implies the '$+$' and '$-$' signs, and because we apply Bland's rule, which implies in this case that no other choice than x_k for basis-leaving variable was possible, because x_k is the variable with the largest index and all quotients are 0. Due to this, all the '$\ominus$' signs above follow.

Additionally, there is a tableau in the cycle where x_k is non-basic but will enter the basis during the next exchange step. We denote this tableau as tableau (b).

$$
(b) \qquad
\begin{array}{c|ccccccc|c}
 & & & & x_k & & & & \\
\hline
z & \oplus & \cdots & \oplus & - & \oplus & \cdots & \oplus & 0 \\
\hline
 & & & & & & & & 0 \\
 & & & & & & & & \vdots \\
 & & & & & & & & 0
\end{array}
$$

Figure 1.13: Tableau (b) shown above is a tableau encountered during cycling where, in the next exchange step, the basis-entering variable is x_k. Because Bland's rule chooses x_k as last option when selecting a basis-entering variable, this implies that it is the only non-basic variable with a strictly negative entry in the objective row, as highlighted above.

Consider now the following point $d \in \mathbb{R}^k$, where we think of the k coordinates of d as values for $(x_1, x_2, \dots, x_k)$. All entries of d are zero except for d_i and for the entries corresponding to variables in tableau (a) that are basic. We set $d_i = 1$ and then set the entries corresponding to the basic variables of tableau (a) to the unique values such that d is a solution. Due to the sign pattern highlighted in Figure 1.12, we have

(i) $d_k < 0$,
(ii) $d_j \geq 0 \ \forall j \in [k-1]$, and
(iii) the objective value of the solution d is strictly better than the all-zeros solution because the objective coefficient of x_i in tableau (a) is strictly negative.

Even though this is not crucial for this proof, we highlight that d is not feasible because $d_k < 0$.

Note that we can evaluate the objective value of the solution d by plugging its values into the objective row of any equivalent tableau, which must lead to the same value. However, if we evaluate the objective value of d by plugging its values into the objective row of tableau (b), we obtain—due to the sign pattern of d and of the objective row of tableau (b) as highlighted in Figure 1.13—that its objective value is strictly worse than the one of the all-zeros solution. This contradicts point (iii) above and finishes the proof. □

Practical experience shows that Bland's pivot selection rule often significantly increases the running time of the Simplex Method. On the other hand, countless practical examples also show that the Simplex Method with the simple pivot rule that chooses the basis-entering variable (pivot column) of largest marginal increase very rarely gets stuck in a degeneracy. Consequently, many common implementations of the Simplex Method mainly follow the latter rule and only deviate from it if necessary. One way to get a practical pivot rule that is guaranteed not to get stuck is to only change to a different rule that does not cycle, like Bland's rule, if another rule that is typically faster, like the one based on marginal improvements, does not strictly improve the objective value during a predefined number of steps.

It is important to know that degeneracy is not a rare phenomenon. In particular, as we will see later, most classical combinatorial optimization problems can naturally be rephrased as highly degenerate linear programs. In these problems, degeneracy appears very naturally and is not just an artefact of a badly-chosen inequality description with redundant constraints that lead to degeneracy.

1.3.9 Simplex Method: phase I

Phase II of the Simplex Method assumes that we start with a feasible tableau, i.e., we know a basic feasible solution. We now consider the problem of obtaining a basic feasible solution. This can be achieved with what is called *phase I of the Simplex Method*, which is a technique to reduce the problem back to the one solved by phase II of the Simplex Method. More precisely, in phase I, an auxiliary problem is defined for which we know a trivial basic feasible solution and, by optimizing the auxiliary problem, we either obtain a feasible basis for the original problem—and can then start phase II with that basis—or learn that the original problem is infeasible.

Consider a general LP in canonical form for which we want to find a basic feasible solution (of the corresponding problem in standard form).

$$\begin{array}{rrcl} \max & c^\top x & & \\ & Ax & \leq & b \\ & x & \in & \mathbb{R}^n_{\geq 0} \ , \end{array} \tag{1.23}$$

where, as usual, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$.

We construct the following auxiliary problem, which uses one new auxiliary variable x_0; moreover, we denote by $\mathbf{1}$ the all-ones vector in $\mathbb{R}^m$.

$$\begin{array}{rrcl} \max & -x_0 & & \\ & Ax - \mathbf{1} \cdot x_0 & \leq & b \\ & x & \in & \mathbb{R}^n_{\geq 0} \\ & x_0 & \in & \mathbb{R}_{\geq 0} \end{array} \tag{1.24}$$

Notice that problem (1.25) is always feasible. Actually, one can even fix $x \in \mathbb{R}^n$ arbitrarily and still obtain a feasible solution by choosing $x_0 \geq 0$ large enough. In particular, x_0 can be thought of as allowing a violation in the original constraints. By maximizing $-x_0$, we aim at minimizing this violation. This leads to the following relation between the two problems.

Observation 1.80

LP (1.23) is feasible $\Leftrightarrow$ LP (1.25) has optimal value 0.

We now show the following two points:

(i) One can easily determine a feasible basis for the auxiliary LP (1.25) such that phase II of the Simplex Method can be applied to it, and

(ii) if the auxiliary LP has optimal value zero, then an optimal tableau for it allows for reading off a feasible starting basis for the original LP, which can be used to solve the original problem with phase II of the Simplex Method.

To apply phase II of the Simplex Method to the auxiliary problem (1.25), we first write it in standard form by introducing the slack variables $x_s \in \mathbb{R}^m$:

$$\begin{array}{rrcl} \max & -x_0 & & \\ & x_s + Ax - \mathbf{1} \cdot x_0 & = & b \\ & x & \in & \mathbb{R}^n_{\geq 0} \\ & x_0 & \in & \mathbb{R}_{\geq 0} \\ & x_s & \in & \mathbb{R}^m_{\geq 0} \ , \end{array} \tag{1.25}$$

and get the corresponding tableau, where we use $\tilde{z} = -x_0$ as the variable for the auxiliary objective function:

$$\begin{array}{c|c|r|c} & x & x_0 & 1 \\ \hline \tilde{z} & 0 & 1 & 0 \\ \hline & & -1 & \\ x_s & A & -1 & b \\ & & -1 & \end{array} \tag{1.26}$$

We assume that b is not non-negative; for otherwise, there is no need to apply phase I of the Simplex Method because one could simply right-away apply phase II with the slack variables as basis. Hence, tableau (1.26) is not feasible. However, it is easy to transform the above tableau into a feasible one through the following well-chosen exchange step.

(i) Choose x_0 as the variable entering the basis, i.e., as the pivot column.
(ii) Choose as basis-leaving variable a row with most negative b-value, i.e., a row with index $i \in \operatorname{argmin}\{b_\ell \colon \ell \in [m]\}$.

Note that in the second step above, we can indeed choose a row with negative b-value because we assumed that (1.26) is not feasible. Hence, $b_i < 0$.

Proposition 1.81

Performing an exchange step on tableau (1.26) using a pivot element as described above leads to a feasible tableau.

Proof. To verify that the new tableau, after performing an exchange step on a pivot element as described above, is feasible, we have to check that the new right-hand side b' is non-negative. For this, let $\overline{A} = [A \ \ -\mathbf{1}]$ be the constraint matrix of the auxiliary problem (1.25), which includes the negative all-ones column corresponding to x_0. Let $\overline{A}_{ik}$ be a pivot element as described above. In particular, $k = n + 1$.

The new right-hand side b' is expressed by the pivot rules (1.14). Notice that $\overline{A}_{ik} = -1$. Hence, the new right-hand side b'_i of the pivot row is indeed non-negative:

$$b'_i = \frac{b_i}{\overline{A}_{ik}} = -b_i > 0 \ .$$

Moreover, also for any non-pivot row $j \in [m] \setminus \{i\}$, we have non-negativity of the new right-hand side:

$$b'_j = b_j - \frac{\overline{A}_{jk} \cdot b_i}{\overline{A}_{ik}} = b_j - b_i \geq 0 \ ,$$

where the second equality is due to $\overline{A}_{jk} = \overline{A}_{ik} = -1$, and the inequality follows from our choice of the pivot row. $\square$

Hence, Proposition 1.81 provides the last missing step to solve the auxiliary problem with phase II of the Simplex Method. Moreover, the optimal tableau of the auxiliary problem can be used to solve the original problem with phase II of the Simplex Method. We show how this can be done in the example below.

Example 1.82

We start with the following linear program in canonical form. Notice that there are two constraints with strictly negative right-hand sides. Hence, we cannot start phase II of the Simplex Method by transforming this problem into standard form and then using the slack variables as basis.

$$\begin{array}{rcrcrcr} \max & x_1 & - & x_2 & + & x_3 & & \\ & 2x_1 & - & x_2 & + & 2x_3 & \leq & 4 \\ & 2x_1 & - & 3x_2 & + & x_3 & \leq & -5 \\ & x_1 & + & x_2 & - & 2x_3 & \leq & -1 \\ & & & & & x_1, x_2, x_3 & \geq & 0 \end{array}$$

We now start with phase I of the Simplex Method and build the auxiliary problem.

$$\begin{array}{rcrcrcrcrcr} \max & & & & & & & - & x_0 & & \\ & 2x_1 & - & x_2 & + & 2x_3 & - & x_0 & \leq & 4 \\ & 2x_1 & - & 3x_2 & + & x_3 & - & x_0 & \leq & -5 \\ & x_1 & + & x_2 & - & 2x_3 & - & x_0 & \leq & -1 \\ & & & & & & & x_0, x_1, x_2, x_3 & \geq & 0 \end{array}$$

Again, we use $\tilde{z} = -x_0$ for the auxiliary objective function, and write the problem in tableau form:

	x_1	x_2	x_3	x_0	1
$\tilde{z}$	0	0	0	1	0
x_4	2	−1	2	−1	4
x_5	2	−3	1	$\boxed{-1}$	−5
x_6	−1	1	−2	−1	−1

This tableau is not feasible, however it will turn feasible after performing an exchange step on the above-explained pivot element. More precisely, the pivot element, which is circled in the table above, is in the column of the auxiliary variable x_0 and in the row of the most negative b_i entry. After performing the exchange step, the following tableau is obtained, which is feasible by choice of the pivot element.

	x_1	x_2	x_3	x_5	1
$\tilde{z}$	2	−3	1	1	−5
x_4	0	2	1	−1	9
x_0	−2	3	−1	−1	5
x_6	−3	$\boxed{4}$	−3	−1	4

Starting from the tableau above, we perform phase II of the Simplex Method. The chosen pivot elements are circled.

	x_1	x_6	x_3	x_5	1
$\tilde{z}$	-0.25	0.75	-1.25	0.25	-2
x_4	1.5	-0.5	2.5	-0.5	7
x_0	0.25	-0.75	$\boxed{1.25}$	-0.25	2
x_2	-0.75	0.25	-0.75	-0.25	1

	x_1	x_6	x_0	x_5	1
$\tilde{z}$	0	0	1	0	0
x_4	1	1	-2	0	3
x_3	0.2	-0.6	0.8	-0.2	1.6
x_2	-0.6	-0.2	0.6	-0.4	2.2

(1.27)

The tableau above is optimal with value 0. Hence, the original linear program is feasible. Moreover, the basic solution to the above tableau, which is

$$(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = (0, 0, 2.2, 1.6, 3, 0, 0) \ ,$$

corresponds to a feasible solution to the original problem by dropping the x_0 coordinate, i.e.,

$$(x_1, x_2, x_3, x_4, x_5, x_6) = (0, 2.2, 1.6, 3, 0, 0) \ .$$

Moreover, the basis of (1.27) is a feasible basis for the original LP. Indeed, by simply removing the x_0 column from tableau (1.27), a tableau for the original problem is obtained, modulo the fact that the objective function $\tilde{z}$ is the auxiliary one and not the original one. We can obtain the original objective, expressed in the basis (x_4, x_3, x_2), as follows. By setting $x_0 = 0$ in tableau (1.27), we obtain the relations

$$\begin{array}{rcrcrcrcl} x_3 & = & -0.2x_1 & + & 0.6x_6 & + & 0.2x_5 & + & 1.6 \ , \text{ and} \\ x_2 & = & 0.6x_1 & + & 0.2x_6 & + & 0.4x_5 & + & 2.2 \ . \end{array}$$

By plugging in the above expressions for x_2 and x_3 into the original objective function $z = x_1 - x_2 + x_3$, we get the objective in the new basis (x_4, x_3, x_2):

$$\begin{array}{rrcrcrcrcr} z = & x_1 & - & (0.6x_1 & + & 0.2x_6 & + & 0.4x_5 & + & 2.2) \\ & & + & (-0.2x_1 & + & 0.6x_6 & + & 0.2x_5 & + & 1.6) \\ = & & & 0.2x_1 & + & 0.4x_6 & - & 0.2x_5 & - & 0.6 \ . \end{array}$$

Hence, the tableau for the original problem in the basis (x_4, x_3, x_2) is the following:

	x_1	x_6	x_5	1
z	-0.2	-0.4	0.2	-0.6
x_4	1	1	0	3
x_3	0.2	-0.6	-0.2	1.6
x_2	-0.6	-0.2	-0.4	2.2

Starting from this *feasible tableau*, we can now use phase II of the Simplex Method to solve the original problem.

Remark 1.83

Note that the optimal tableau obtained after solving the auxiliary problem, i.e., tableau (1.27) in Example 1.82, may be such that x_0 is a basic variable of value 0, due to degeneracy. In this case, the basis of the auxiliary problem, which contains x_0, is clearly not a feasible basis of the original one, where no x_0 variable exists. However, whenever this happens, one can simply perform an additional exchange step on any non-zero pivot element in the row of x_0, to obtain another optimal tableau where x_0 is non-basic.
Observe that, whenever x_0 is basic, there must always be a non-zero element in the row corresponding to x_0. For otherwise, all solutions to the auxiliary problem would have the same x_0 value. However, this is not true, because there are feasible solutions for arbitrarily high values of x_0.

Finally, we highlight that one can avoid the reconstruction of the original objective function after obtaining an optimal tableau for phase I of the Simplex Method by carrying along the original objective function in the tableau. More precisely, when constructing the tableau for the auxiliary LP for phase I of the Simplex Method, one can add one additional row corresponding to the original objective z as follows:

$$
\begin{array}{lcc|c|c|c|cl}
 & & & & \text{auxiliary column} & & & \\
 & & & & \downarrow & & & \\
 & & & x & x_0 & 1 & & \\ \hline
\text{auxiliary row} & \rightarrow & \tilde{z} & 0 & 1 & 0 & \leftarrow & \text{objective function for phase I} \\ \hline
\text{objective function} & \rightarrow & z & -c & 0 & 0 & \leftarrow & \text{objective function for phase II} \\ \hline
 & & & & -1 & & & \\
 & & x_s & A & \vdots & b & & \\
 & & & & -1 & & & \\ \hline
\end{array}
$$

Starting from this tableau, we continue as before. More precisely, we first perform an exchange step to obtain a feasible tableau where x_0 is a basic variable. We then perform phase II of the Simplex Method with the special rule that we never choose a pivot on the row corresponding to z. More precisely, the row corresponding to z is ignored for choosing pivot elements or for determining whether the tableau is feasible. Nevertheless, whenever the pivot element is fixed and an exchange step is performed, the row corresponding to z is treated like any other row.

This way, the original objective remains expressed in the current non-basic variables throughout phase I of the Simplex Method. Hence, when phase I of the Simplex Method terminates with an optimal tableau of value 0, and with x_0 being non-basic, then it suffices to delete the column corresponding to x_0 and the row corresponding to $\widetilde{z}$ to obtain a feasible tableau for the original problem. From this tableau, we can then start phase II of the Simplex Method.

1.4 Linear duality

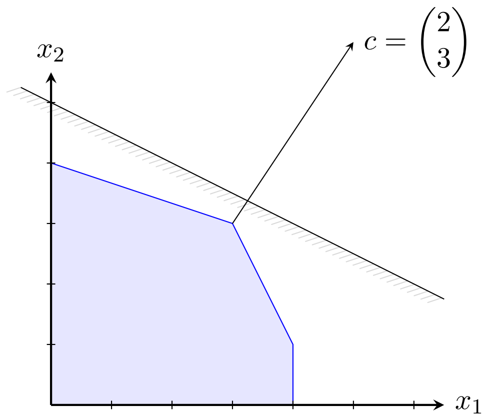
It turns out that for every linear program there exists a closely related linear program, called the *dual problem*. The dual program can be seen as a different way to look at the original problem, which is typically referred to as the *primal*. The close relation between primal and dual can be exploited in several ways. In particular, the dual provides quick ways for showing optimality of an optimal primal solution. Moreover, given an optimal solution for either the primal or the dual often allows for quickly obtaining an optimal solution to the other problem. This connection can sometimes be used to find an optimal solution to the primal more quickly. More precisely, sometimes it is faster to use for example the Simplex Method to solve the dual problem and then derive from an optimal dual solution an optimal primal one. Duality is also key in the context of sensitivity analysis, and has various algorithmic applications, for example in so-called *primal-dual* procedures, which solve a problem by simultaneously exploiting properties of the primal and dual of a problem.

In the next section, we introduce linear duality as a way to find strong upper bounds for the optimal value of a linear program.

1.4.1 Motivation: finding bounds on optimal value

Consider the following linear program in canonical form. Its feasible region is highlighted below on the right-hand side together with the objective function, which is maximized at the point $(x_1, x_2) = (3, 3)$.

$$
\begin{array}{rrcrcl}
\max & 2x_1 & + & 3x_2 & & \\
 & x_1 & + & 3x_2 & \leq & 12 \\
 & 2x_1 & + & x_2 & \leq & 9 \\
 & x_1 & & & \leq & 4 \\
 & x_1 & + & 2x_2 & \leq & 10 \\
 & x_1 & & & \geq & 0 \\
 & & & x_2 & \geq & 0
\end{array}
$$



Assume we want to find an upper bound on the optimal value of the above linear program. One way to obtain such a bound is by multiplying the constraint $x_1 + 2x_2 \leq 10$ by 2 to obtain

$$2x_1 + 4x_2 \leq 20 \ .$$

Because any solution to the LP satisfies $x_1, x_2 \geq 0$, we have

$$2x_1 + 3x_2 \leq 2x_1 + 4x_2 \leq 20 \ .$$

Hence, 20 is an upper bound on the optimal LP value. An obvious questions is whether we can obtain a better bound by a similar technique. Indeed, by adding up the constraints $x_1+3x_2 \leq 12$ and $x_1 \leq 4$ we get

$$2x_1 + 3x_2 \leq 16 \ ,$$

showing that the optimal LP value is upper bounded by 16. In general, to obtain a valid bound for the objective $2x_1 + 3x_2$, one can try to multiply each '$\leq$'-constraint by some non-negative factor such that a linear constraint $\alpha x_1 + \beta x_2 \leq \gamma$ is obtained with $\alpha \geq 2$, $\beta \geq 3$. Then γ is an upper bound to the optimal value of the LP.

Hence, finding the best such upper bound is itself an LP where we have a variable for each constraint of the original LP, except for the non-negativity constraints. Those variables describe the coefficients in the conic combination of the constraints. Furthermore, there is a constraint for each variable in the original LP. For the example above it looks as follows.

$$\begin{array}{rrcrcrcrcl}
\min & 12y_1 & + & 9y_2 & + & 4y_3 & + & 10y_4 & & \\
& y_1 & + & 2y_2 & + & y_3 & + & y_4 & \geq & 2 \\
& 3y_1 & + & y_2 & & & + & 2y_4 & \geq & 3 \\
& y_1 & & & & & & & \geq & 0 \\
& & & y_2 & & & & & \geq & 0 \\
& & & & & y_3 & & & \geq & 0 \\
& & & & & & & y_4 & \geq & 0
\end{array}$$

This problem is called the *dual problem* to the original problem. The unique optimal solution of this dual problem is $(y_1, y_2, y_3, y_4) = (\frac{4}{5}, \frac{3}{5}, 0, 0)$ and leads to an optimal value of 15. The way how we set up this dual problem implies that 15 is an upper bound to the value of the original LP. As it turns one, the optimal value of the dual LP is even equal to the optimal value of the original LP, which is achieved at the point $(3, 3)$. This is no coincidence. It is at the heart of one of the most important results in linear programming, namely strong duality.

In the following, we start by making the above discussion on constructing the dual more formal before covering some key results linked to linear duality, like strong duality.

1.4.2 Dual of a linear program

Let us revisit the above discussion in more generality. Consider a general linear program in canonical form, which we will call the *primal problem* to distinguish it later from the *dual problem*:

$$\begin{array}{rrcl}
\max & c^\top x & & \\
& Ax & \leq & b \\
& x & \geq & 0 \ .
\end{array} \tag{PLP}$$

Following the above discussion, its corresponding dual problem is:

$$\begin{array}{rrcl}
\min & b^\top y & & \\
& A^\top y & \geq & c \\
& y & \geq & 0 \ .
\end{array} \tag{DLP}$$

Notice that for each constraint in the primal there is a variable in the dual and vice versa. In particular, if the primal problem has n variables and m constraints (not counting the non-negativity constraints) then the dual problem has m variables and n constraints (not counting the non-negativity constraints).

Dualization of LPs not in canonical form

So far, we focussed on dualizing linear programs in canonical form. However, dualitzation can be applied to a linear program in an arbitrary form. Of course, one can simply bring an arbitrary linear program first into canonical form and then dualize it. However, one can also dualize any linear program directly, without first transforming it into canonical form. For example, if we have a constraint of type '$\geq$' in the primal, then we can use a dual variable for this constraint that is non-positive instead of non-negative.

Recall that if we first transform a problem into canonical form, then this can lead to extra variables and/or constraints, which leads to extra constraints and/or variables in the dual. Nevertheless, one can simplify the resulting dual to avoid the introduction of extra variables or constraints.

In the problem sets, we will expand on the above discussion about dualizing linear programs that are not in canonical form. In what follows, we focus on the case of primal problems that are in canonical form.

Dual of dual is primal

An interesting property of linear duality is that the dual of the dual is the primal problem. One way to see this is to first transform (DLP) into canonical form:

$$\begin{array}{rrcl} -\max & -b^\top y & & \\ & -A^\top y & \leq & -c \\ & y & \geq & 0 \ , \end{array} \tag{1.28}$$

and then dualizing the above problem, which leads to

$$\begin{array}{rrcl} -\min & -c^\top x & & \\ & -Ax & \geq & -b \\ & x & \geq & 0 \ , \end{array}$$

which is again the primal problem (PLP). As discussed, it is not necessary to first transform (DLP) into (1.28) to dualize it. One can immediately dualize it by again deriving how one can bound the objective value $b^\top y$ with the available constraints.

1.4.3 Weak and strong linear duality

Notice that by the way how we constructed the dual problem (DLP) from the primal problem (PLP), the value of any feasible solution to the dual upper bounds the value of any feasible solution to the primal. This result is known as *weak duality* and is easy to prove formally.

Theorem 1.84: Weak duality

Let x, y be feasible solutions to (PLP) and (DLP), respectively. Then

$$c^\top x \leq b^\top y \enspace .$$

Proof. The result follows from

$$\underbrace{c^\top}_{\leq y^\top A} x \leq y^\top \underbrace{Ax}_{\leq b} \leq y^\top b = b^\top y \enspace ,$$

where the first inequality holds due to $c \leq A^\top y$ and $x \geq 0$, and the second inequality follows from $Ax \leq b$ and $y \geq 0$. □

Weak duality comes already with some interesting implications. It implies that if the primal is unbounded, then the dual must be infeasible. Since the primal-dual relation is symmetric, it also implies that if the dual is unbounded then the primal is infeasible. Furthermore, if one can find a primal feasible solution x and a dual feasible solution y such that $c^\top x = b^\top y$, then both x and y are optimal solutions for the primal and dual, respectively. Hence, in this case, the dual solution is an optimality certificate of the primal solution and vice versa.

Notice that weak duality does not rule out that both the primal and dual are infeasible, which is indeed possible.

Example 1.85: Infeasible primal and dual LP

Below is a pair of a primal and dual linear program that are both infeasible.

$$\begin{array}{rrcrcr}
\max & 2x_1 & - & x_2 & & \\
 & x_1 & - & x_2 & \leq & 1 \\
 & -x_1 & + & x_2 & \leq & -2 \\
 & x_1 & & & \geq & 0 \\
 & & & x_2 & \geq & 0
\end{array}
\qquad\qquad
\begin{array}{rrcrcr}
\min & y_1 & - & 2y_2 & & \\
 & y_1 & - & y_2 & \geq & 2 \\
 & -y_1 & + & y_2 & \geq & -1 \\
 & y_1 & & & \geq & 0 \\
 & & & y_2 & \geq & 0
\end{array}$$

Notice that the existence of such an example is not surprising due to the following. Consider two independent linear programs in canonical form, one that is infeasible and one for which the dual is infeasible. Then we can simply combine them into a single linear program maximizing the sum of both objectives. It is not hard to check that this leads to a new linear program for which both the primal and dual are infeasible.

Interestingly, when the primal and dual are both feasible—which is often the most interesting case—then a cornerstone result in linear duality implies that their optimal values are equal. This result is known as *strong duality*.

Theorem 1.86: Strong duality

If the primal has a finite optimum, then also the corresponding dual problem has a finite optimum and their values are equal, i.e., there exists a feasible solution x for (PLP) and a feasible solution y for (DLP) such that $c^\top x = b^\top y$.

We provide a proof of Theorem 1.86 later in Section 1.4.4, after having discussed the dual interpretation of the simplex tableau, which allows us to simultaneously make statements about the primal and dual problem using the same tableau.

Strong duality implies that if an LP has a finite optimum, then the optimality of a solution to that LP can *always* be certified by exhibiting a dual solution with the same value. Moreover, strong duality has numerous implications in various combinatorial optimization problems. In particular, most max-min relations in Combinatorial Optimization, like the max-flow min-cut theorem, can be interpreted as a consequence of strong duality.

We recall that every linear program either (i) has a finite optimum, (ii) is unbounded, i.e., allows for solutions with arbitrarily strong objective values, or (iii) is infeasible. The table below shows in which of those 3 states an LP and its corresponding dual can be, where a check mark (✓) indicates that the corresponding combination is possible and a cross (✗) that it is not.

		dual		
		finite	unbounded	infeasible
primal	finite	✓	✗	✗
	unbounded	✗	✗	✓
	infeasible	✗	✓	✓

The above table is symmetric because the dual of the dual is the primal. The first row and first column follow from strong duality. The second row and column follow from weak duality. Finally, the check mark at the bottom right follows from the fact that there are pairs of primal and dual LPs that are both infeasible (see Example 1.85).

1.4.4 Dual interpretation of simplex tableau

We now discuss how the simplex tableau for the dual problem is tightly related to the one of the primal problem. More precisely, by reading the tableau differently, one can interpret the primal tableau as one for the dual problem. To highlight this connection, consider again the dual problem (DLP) in canonical form, i.e.,

$$\begin{array}{lrcl} \max & -b^\top y & & \\ & -A^\top y & \leq & -c \\ & y & \geq & 0\ , \end{array}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. Introducing the objective function variable $w = b^\top y$ for the dual and the dual slack variables $y^s \in \mathbb{R}^n$, we write the dual problem in standard form:

$$\begin{array}{lrrrcr} \max & -w & & & = & -b^\top y \\ & & y^s & -A^\top y & = & -c \\ & & & y, y^s & \geq & 0\ . \end{array}$$

Given an initial primal tableau for the primal, with objective $z = c^\top x$ and slack variables x^s, we can directly read the dual relations by reading the tableau column-wise with the equality sign next to the variables:

$$\begin{array}{c|c|c} & x\ \ (y^s) & 1\ \ (w) \\ \hline z\ \ (1) & -c^\top & 0 \\ \hline x^s\ \ (y) & A & b \end{array} \quad \begin{array}{c} (\shortparallel) \\ (+) \end{array}$$

Here, x^s and y^s denote the primal and dual slack variables, respectively. When reading the tableau column-wise, we interpret the boundary of the tableau differently, as highlighted by the parenthetical expressions in the tableau above:

(i) The role of the primal variables x is replaced by the dual slack variables y^s.
(ii) The primal slack variables x^s are replaced by the dual variables y.
(iii) The primal objective z takes the role of the dual constant 1.
(iv) The primal constant column 1 takes the role of the dual objective w.

This so-called *dual reading* or *dual interpretation* leads to:

$$\begin{array}{rcrcl} y^s & = & -c & + & A^\top y \\ w & = & 0 & + & b^\top y\ , \end{array}$$

which indeed corresponds to the dual constraints in standard form. In particular, this way of reading uses the following one-to-one correspondence between the primal slack variables and the dual variables:

Relation between primal/dual variables and slacks

x_j^s	primal slack variable	$\longleftrightarrow$	y_j	dual variable
x_i	primal variable	$\longleftrightarrow$	y_i^s	dual slack variable

A crucial fact is that the dual reading method works for any primal tableau, i.e., it always leads to an equivalent equation system for the dual. Hence, any primal tableau can be interpreted as a dual tableau when using dual reading. This key property, which readily leads to a proof of the strong duality theorem, is formally stated below.

Lemma 1.87

For any primal tableau, the dual reading of the tableau leads to an equivalent equation system for the dual (including the objective function).

Proof. To set notation, assume that the original primal problem in standard form is described as usual by:

$$\begin{array}{rrcrcl} \max & z & & & = & c^\top x \\ & x^s & + & Ax & = & b \\ & & & x, x^s & \geq & 0\ , \end{array} \tag{1.29}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. We recall that the corresponding dual problem in standard form is given by:

$$\begin{array}{rrrcl} \max & -w & & = & -b^\top y \\ & y^s & -A^\top y & = & -c \\ & & y, y^s & \geq & 0\ . \end{array} \tag{1.30}$$

Now consider a primal tableau for (1.29) (with respect to an arbitrary basis):

$$\begin{array}{c|c|c} & x_N & 1 \\ \hline z & -\bar{c}^\top & \bar{q} \\ \hline x_B & \bar{A} & \bar{b} \end{array}\ , \tag{1.31}$$

where x_B and x_N are just names for the vectors of basic and non-basic variables, respectively, in the above tableau. Our task is to show that by dual reading the tableau (1.31), one obtains an equation system for the dual problem that is equivalent to the one in (1.30). Notice that the dual equation system obtained from dual reading (1.31) clearly has full rank because it is in tableau form. Hence, one only has to show that each of the dual constraints stemming from (1.31) is implied by the equations in (1.30), i.e., it is a linear combination of the equations in (1.30).

Now consider an arbitrary column of (1.31); this can either be one of the n columns corresponding to the non-basic variables x_N or the last column, which corresponds to the constant 1. Let us call the considered column the *selected column*. Notice that any such column corresponds to a dual equation when using dual reading. To represent the entries of the selected column in a way that we can naturally interpret both in the primal and the dual, we define a vector in the space $Q := \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}$, where we interpret the $m + n + 2$ coordinates of the space Q as follows:

(i) The first entry corresponds to the primal objective z and thus to the dual constant term.
(ii) The next m entries correspond to the primal slacks x^s and thus to the dual variables y.
(iii) The next n entries correspond to the primal variables x and thus to the dual slacks y^s.
(iv) The last entry corresponds to the primal constant term and thus to the dual objective w.

Now let $\eta \in Q$ be the vector defined as follows:

(i) If the selected column corresponds to a variable of x_N, then we set the η-entry that corresponds to this variable to -1. Otherwise, if the selected column corresponds to the constant column, then we set the η-entry that corresponds to the constant term to 1.

(ii) For any row of the tableau (1.31), which either corresponds to z or a variable in x_B, we set the corresponding entry of η to the value that the selected column has in this row.

(iii) All other entries of η are set to zero.

We now observe a few properties of the vector η. First, by reading the tableau (1.31) in the primal way, we see that adding or subtracting η to any primal solution given in the space Q, another primal solution is obtained. Notice that whether z, x^s, and x is a primal solution can be rephrased as follows:

$$z \in \mathbb{R}, x^s \in \mathbb{R}^m, x \in \mathbb{R}^n \text{ is a primal solution} \quad \Leftrightarrow \quad \begin{pmatrix} 1 & 0^\top & -c^\top & 0 \\ 0 & I & A & -b \end{pmatrix} \cdot \begin{pmatrix} z \\ x^s \\ x \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \ .$$

Hence, this implies

$$\begin{pmatrix} 1 & 0^\top & -c^\top & 0 \\ 0 & I & A & -b \end{pmatrix} \cdot \eta = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \ .$$

Thus, η is in the null-space of the row space of $\left(\begin{smallmatrix} 1 & 0^\top & -c^\top & 0 \\ 0 & I & A & -b \end{smallmatrix}\right)$. This null-space is spanned by the columns of

$$\begin{pmatrix} c^\top & 0 \\ -A & b \\ I & 0 \\ 0 & 1 \end{pmatrix} ,$$

which implies that there exists some $\lambda \in \mathbb{R}^{n+1}$ such that

$$\eta = \begin{pmatrix} c^\top & 0 \\ -A & b \\ I & 0 \\ 0 & 1 \end{pmatrix} \cdot \lambda \ . \tag{1.32}$$

Moreover, observe that the dual equation stemming from dual reading the tableau (1.31) is given by

$$\eta^\top \begin{pmatrix} 1 \\ y \\ y^s \\ -w \end{pmatrix} = 0 \ . \tag{1.33}$$

To show that the above constraint is implied by the equations in (1.30), we have to show that any dual solution $y \in \mathbb{R}^m$, $y^s \in \mathbb{R}^n$, and $w \in \mathbb{R}$ fulfills (1.33). Observe that the fact of y, y^s, and w being a dual solution, i.e., fulfilling the equations of (1.30), can be written in matrix form as follows:

$$\begin{pmatrix} c & -A^\top & I & 0 \\ 0 & b^\top & 0^\top & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ y \\ y^s \\ -w \end{pmatrix} = 0 \ . \tag{1.34}$$

Finally, we conclude that (1.33) indeed holds for any dual solution y, y^s, w because

$$\eta^\top \begin{pmatrix} 1 \\ y \\ y^s \\ -w \end{pmatrix} = \lambda^\top \begin{pmatrix} c & -A^\top & I & 0 \\ 0 & b^\top & 0^\top & 1 \end{pmatrix} \begin{pmatrix} 1 \\ y \\ y^s \\ -w \end{pmatrix} = 0 \ ,$$

where the first equality follows from (1.32) and the second one from (1.34). □

Lemma 1.87 allows for providing a quick proof of the strong duality theorem of linear programming, i.e., Theorem 1.86. Indeed, the dual interpretation of the simplex tableau, together with Proposition 1.87, show that whenever we have an optimal (primal) tableau, it also reveals an optimal dual solution. This fact is at the heart of the proof below.

Proof of Theorem 1.86. Assume that we are given a linear (primal) program with a finite optimum. Now consider an optimal tableau for the problem. Notice that such a tableau exists because first there is a feasible tableau due to feasibility of the problem. (Indeed, phase I of the Simplex Method implies this.) Then there is also an optimal tableau because of phase II of the Simplex Method which, when run with Bland's rule, must either return an optimal tableau or an unbounded direction, which does not exist because the primal optimum is finite. In an optimal tableau the row corresponding to the primal objective is a non-negative vector. Hence, when reading the tableau using dual reading, we obtain a feasible basic dual solution. Moreover, because the objective value obtained through dual reading is the same as through primal reading, we obtain that the dual basic solution y for the optimal tableau satisfies $b^\top y = c^\top x$, where x is the primal basic solution of the optimal tableau. Hence, by weak duality, x and y are optimal primal and dual solutions, respectively. □

Example 1.88

Consider the following (primal) linear program in canonical form.

$$\begin{array}{rrcrcrcrcl} \max & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & \\ & x_1 & - & x_2 & - & x_3 & + & 3x_4 & \leq & 1 \\ & 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \leq & 55 \\ & -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \leq & 3 \\ & & & & & & x_1, x_2, & x_3, x_4 & \geq & 0 \end{array}$$

Below is an optimal tableau that corresponds to the above LP.

	$x_1\,(y_1^s)$	$x_3\,(y_3^s)$	$x_1^s\,(y_1)$	$x_3^s\,(y_3)$	$1\,(w)$
$(1)\,z$	1	2	11	6	29
$(y_2^s)\,x_2$	2	4	5	3	14
$(y_4^s)\,x_4$	1	1	2	1	5
$(y_2)\,x_2^s$	−5	−9	−21	−11	1

Because the tableau is optimal, the basic primal solution, i.e.,

$$(x_1, x_2, x_3, x_4) = (0, 14, 0, 5) \ ,$$

is an optimal primal solution to the LP, and the basic dual solution of the above tableau, i.e.,

$$(y_1, y_2, y_3, y_4) = (11, 0, 6) \ ,$$

which is obtained by dual reading the tableau, is an optimal dual solution. Both solutions are obtained from the primal and dual reading, respectively, when setting the corresponding non-basic variables to zero. Notice that if a slack variable x_i^s (y_i^s) is in the optimal basis, then the corresponding dual variable y_i (x_i) is non-basic and therefore equal to zero. This relationship is known as *complementary slackness* and will be discussed in more detail in Section 1.4.5.

Remark 1.89: Notation of primal & dual variables and slacks

Due to the strong correspondence between the primal (dual) slack variables and the dual (primal) variables, both variables are often denoted by the same symbol. This simplifies the dual reading, but also involves the risk of mistaking the primal slack variables for the dual variables and vice versa. For instance, let us consider the following tableau, corresponding to a primal problem in canonical form:

	x	1
z	$-c^\top$	0
y	A	b

In the primal interpretation, x denotes the variables and y the slack variables, whereas in the dual interpretation, x denotes the slack variables and y the variables. Moreover, z is the primal objective but dual constant, whereas 1 is the primal constant but dual objective.

Moreover, all terminology we introduced for (primal) tableaus, like degeneracy, optimality, feasibility, can be defined analogously with respect to the dual. More precisely, given a primal tableau, we call it *dual feasible* if its corresponding dual basic solution (obtained through dual reading) is feasible, i.e., all entries in the objective row are non-negative. It is called *dual degenerate* if at least one of the objective row entries is zero. Finally, there is no difference between the notions of a primal and dual optimal tableau.

1.4.5 Complementary slackness

The complementary slackness theorem formalizes a crucial relation between optimal primal and dual solutions, which we already observed in Example 1.88.

Theorem 1.90: Complementary slackness theorem

Consider a pair of primal and dual linear programs with finite optima:

$$\begin{array}{rcl} \max\ c^\top x & & \\ Ax & \leq & b \\ x & \geq & 0 \end{array} \qquad\qquad \begin{array}{rcl} \min\ b^\top y & & \\ A^\top y & \geq & c \\ y & \geq & 0\ . \end{array}$$

Let $\overline{x}$ be a feasible primal solution and $\overline{y}$ a feasible dual solution. Then both $\overline{x}$ and $\overline{y}$ are optimal solutions (for the primal and dual, respectively) if and only if

(i) $(b - A\overline{x})^\top \overline{y} = 0$, and
(ii) $(A^\top \overline{y} - c)^\top \overline{x} = 0$.

Proof. Consider again the chain of relations that we used to derive the weak duality theorem:

$$c^\top \overline{x} \ \leq\ \overline{y}^\top A\overline{x} \ \leq\ \overline{y}^\top b\ , \tag{1.35}$$

which is based on the relations $c \leq A^\top \overline{y}$, $A\overline{x} \leq b$, and $\overline{x}, \overline{y} \geq 0$. Now, by strong duality, $\overline{x}$ and $\overline{y}$ are optimal if and only if $c^\top \overline{x} = b^\top \overline{y}$, i.e., the left-hand side and right-hand side of (1.35) are equal. This happens if and only if both inequalities in (1.35) are equalities, which is equivalent to

(i) $(b - A\overline{x})^\top \overline{y} = 0$, and
(ii) $(A^\top \overline{y} - c)^\top \overline{x} = 0$,

as desired. □

As we will see in the problem sets, apart from providing an important insight on the relation between optimal primal and dual solutions, the complementary slackness theorem can often be used to derive an optimal dual solution from a primal one or vice versa.