3

Linear Programming

A (very) special case of convex optimization problem arises when all constraints and the objective function are linear functions. In other words, the general linear problem can be written as

$$
\begin{aligned}
&\min c^\top x \\
&\quad a_i^\top x \sim b_i \quad i = 1, \ldots, m \\
&\quad l_j \leq x_j \leq u_j \quad j = 1, \ldots, n
\end{aligned}
$$

where $\sim \in \{\leq, \geq, =\}$, $l_j \in \mathbb{R} \cup \{-\infty\}$ and $u_j \in \mathbb{R} \cup \{+\infty\}$. Notice that the domain of each variable is thus an interval of $\mathbb{R}$.

Note that strict inequalities are not *allowed.*

By definition, the feasible set of a linear program is a polyhedron, i.e., the intersection of a finite number of hyperplanes and halfspaces in $\mathbb{R}^n$. Thus, a linear program consists in optimizing a linear function over a polyhedron. Polyhedra are well studied as mathematical and geometrical objects, and have many properties that proved instrumental in the design and evolution of linear optimization theory and algorithms. For this reason, we now take a well deserved detour in LP geometry.

If the polyhedron is bounded, it is often called a polytope.

3.1 Geometry

In the following, we will assume that polyhedra are bounded (and thus polytopes).

Definition 3.1. *A point x in a polyhedron P is called an* extreme point *(or* vertex*) of P if it cannot be expressed as a strict convex combination of points of P.*

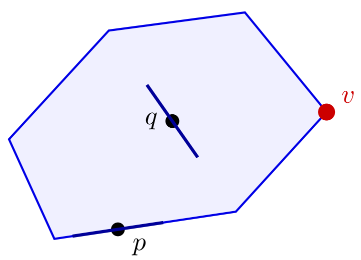

Figure 3.1: Vertex v and non-vertices p and q.

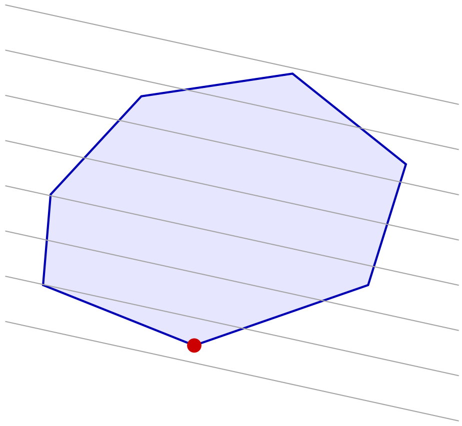

Figure 3.2: Geometrical interpretation of Corollary 3.1.

A graphical representation is depicted in Figure 3.1. Basic intuition tells us that the vertices of a polytope, which are always finite in number, are enough to describe the polytope itself: once you know the vertices, the geometrical object is uniquely determined. This (correct) intuition is formalized by the following theorem, due to Minkowski and Weyl:

Theorem 3.1. *Every point of a polytope can be expressed as a convex combination of its vertices.*

In other words, polytopes can be described in two completely different, yet equivalent, ways: either as a finite intersection of linear constraints (called H, or *external*, description) or with a finite number of vertices (called V, or *internal*, description). This has a fundamental implication for linear optimization:

Corollary 3.1. *If the feasible set P of a linear program is bounded and non empty, then there exists at least one optimal vertex.*

This result is often incorrectly quoted as "the optimal solution of a LP is always a vertex", which is wrong. Appreciate the difference w.r.t. the actual statement.

Proof. Let $x^1, \dots, x^k$ be the vertices of P and $z^* = \min\{c^\top x^i : i = 1, \dots, k\}$. Let $y \in P$ be an arbitrary feasible solution. By the Minkowski-Weyl theorem we have:

$$\begin{aligned} c^\top y &= c^\top \left(\sum_{i=1}^{k} \lambda_i x^i \right) \\ &= \sum_{i=1}^{k} \lambda_i \left(c^\top x^i \right) \\ &\geq \sum_{i=1}^{k} \lambda_i z^* \\ &= z^* \end{aligned}$$

□

Because of the above, while formally being a continuous optimization problem, LP can be reduced to a *combinatorial* problem, as we "just" need to search the optimal solution among the finitely many vertices of the feasible set. In order to turn this strategy into a proper algorithm, we need however to provide an *algebraic* characterization of vertices, i.e., a *practical* way to enumerate the vertices of a polyhedron starting from its H description (the one that we naturally start with in optimization).

In order to simplify the exposition, we will assume that the LP is described in *standard form*, i.e., all linear constraints are equations and all variables are non-negative.

$$\begin{aligned} \min\ & c^\top x \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

Note that we can assume the standard form without loss of generality, as we can bring an arbitrary linear program into this form with simple transformations that do not blow up the size of the model. Just to give a few examples:

- an inequality $a^\top x \leq b$ can be rewritten as $a^\top x + s = b$, at the cost of introducing an artificial variable $s \geq 0$;
- a finite lower bound l_j different from zero can be dealt with by just translating the corresponding variable;
- a free variable x_j can be expressed as a difference of two non-negative variables x_j^+ and x_j^-, as $x_j = x_j^+ - x_j^-$.

In addition, since we have only linear equations (except for the non-negativity constraints), we can assume that $m \leq n$ (i.e., we have at least as many columns as rows in the matrix) and the matrix A has full row rank: $\text{rank}(A) = m$.

If the matrix does not have a full row rank, the model is either trivially infeasible or it has at least one redundant constraint.

3.1.1 *Bases*

A *basis* is defined as a set of m linearly independent columns of A. We will denote with $\mathcal{B} = \{k_1, \ldots, k_m\}$ the *ordered* set of indices corresponding to the basic columns, and with $\mathcal{R}$ the set of indices associated with the non-basic columns. The partition of the columns (and thus variables) into basic and non-basic allows us to rewrite the linear system as

$$Bx_B + Rx_R = b \tag{1}$$

where B is the set of basic columns and R the set of non-basic columns. Given that B is by definition invertible, we can finally rewrite (1) as

Permuting the order of the columns, and thus variables, in a linear program is clearly immaterial.

$$x_B = B^{-1}b - B^{-1}Rx_R \tag{2}$$

from which it is clear that the value of the basic variables x_B is uniquely determined by the values assigned to the non-basic ones x_R. In particular, if all non-basic variables are assigned the value zero we obtain the solution:

Intuitively, we are using the linear equations in the system to show that we only have $n - m$ degrees of freedom.

$$x = (x_B, x_N) = (B^{-1}b, 0)$$

which is called a *basic solution*. If such solution also satisfies the non negativity constraints, then it is a *basic feasible solution*.

Clearly, the number of bases in a given linear system is a finite number, and so it is, at least in principle, possible to enumerate all possible bases, construct the corresponding basic solution, and check whether it is feasible, thus enumerating all bfs. The key fact is that there is a correspondence between the algebraic notion of bfs and the geometrical notion of vertex, as shown in the following theorem:

At most $\binom{n}{m}$.

Theorem 3.2. *A point $x \in \mathbb{R}^n$ is a vertex of the non-empty polyhedron $P = \{x \mid Ax = b, x \geq 0\}$* iff *$x$ is a bfs of the system $Ax = b$.*

Proof. $\Leftarrow$) Let x be the basic feasible solution associated to a basis B of the system. Without loss of generality we can permute the columns of B such that the positive values are in the first k positions:

$$x = [\underbrace{x_1, \ldots, x_k}_{k \geq 0}, 0, \ldots, 0]^\top$$

Note that in general $k < m$, as a basic variable could also get the value 0 (we say that the basis is *degenerate* when this happens). Of course, we cannot have $k > m$, as the non-basic variables are all set to zero by construction. Let $A_1, \ldots, A_k$ be the columns associated to $x_1, \ldots, x_k$: being part of the basis, they are linearly independent. Now, let's assume by contradiction that x is *not* a vertex, i.e., there exist $y, z \in P$ and $\lambda \in (0, 1)$ such that $x = \lambda y + (1 - \lambda)z$. Since all the entries in y and z must be non-negative, and their convex combination must match x, we clearly have that the last $n - k$ components of y and z are also zero, i.e., $y = [y_1, \ldots, y_k, 0, \ldots, 0]^\top$ and $z = [z_1, \ldots, z_k, 0, \ldots, 0]^\top$. Since they are also solutions of the linear system, this implies:

$$\begin{aligned} A_1 y_1 + A_2 y_2 + \cdots + A_k y_k &= b \\ A_1 z_1 + A_2 z_2 + \cdots + A_k z_k &= b \end{aligned}$$

Subtracting the two equations we finally obtain:

$$\underbrace{(y_1 - z_1)}_{\alpha_1} A_1 + \underbrace{(y_2 - z_2)}_{\alpha_2} A_2 + \cdots + \underbrace{(y_k - z_k)}_{\alpha_k} A_k = 0$$

So we have found a vector of multipliers $\alpha \neq 0$ proving that the vectors $A_1, \ldots, A_k$ are linearly dependent, which is the contradiction we were looking for.

$\Rightarrow$) Suppose x is vertex, Again, possibly after some permutation, we can write it as:

$$x = [\underbrace{x_1, \ldots, x_k}_{k \geq 0}, 0, \ldots, 0]^\top$$

The difference w.r.t. the previous case is that, at the moment, we cannot assume $k \leq m$. Since a vertex is a feasible solution to the system, we still have:

$$A_1 x_1 + A_2 x_2 + \cdots + A_k x_k = b \tag{$\star$}$$

Now, there are two cases:

1. The columns $A_1, \ldots, A_k$ are linearly independent (or $k = 0$). In this case we have $k \leq m$ and by arbitrarily selecting other $m - k$ linearly independent columns (always possible, as A has rank m) we can complete it to a basis

$$B = [A_1, \ldots, A_k, A_{k+1}, \ldots, A_m]$$

whose basic feasible solution is exactly x.

2. The columns $A_1, \dots, A_k$ are linearly dependent. Then there exists $\alpha \neq 0$ such that:

$$\alpha_1 A_1 + \alpha_2 A_2 + \cdots + \alpha_k A_k = 0$$

Now, we add and subtract this equation, multiplied by ε, from ($\star$), and obtain:

$$\begin{aligned} A_1(x_1 + \varepsilon\alpha_1) + A_2(x_2 + \varepsilon\alpha_2) + \cdots + A_k(x_k + \varepsilon\alpha_k) = b \\ A_1(x_1 - \varepsilon\alpha_1) + A_2(x_2 - \varepsilon\alpha_2) + \cdots + A_k(x_k - \varepsilon\alpha_k) = b \end{aligned}$$

By defining $y = [x_1 + \varepsilon\alpha_1, \dots, x_k + \varepsilon\alpha_k, 0, \dots, 0]^\top$ and $z = [x_1 - \varepsilon\alpha_1, \dots, x_k - \varepsilon\alpha_k, 0, \dots, 0]^\top$, and picking a sufficiently small ε such that $y, z \geq 0$ (always possible), we have found two points in P such that $x = \frac{1}{2}y + \frac{1}{2}z$, which is a contradiction as x is a vertex. So this latter case cannot happen, and this concludes the proof.

□

As an immediate corollary, we have that:

Corollary 3.2. *If the feasible set P of a linear program is bounded and non empty, then there exists at least one optimal basic feasible solution.*

Note that the correspondence between bases and vertices is not $1:1$: a basis uniquely defines a vertex, but a vertex might be associated with different bases, in case of degeneracy. Finally, the considerations above do not (yet) yield a practical algorithm, as the number of basic feasible solutions (as the number of vertices) grows exponentially with n and m, and thus is it impossible to enumerate them all to solve the problem.

3.2 The Primal Simplex Algorithm

The main idea behind the primal simplex is as follows: we start from a feasible basis B of the linear problem, and iteratively move to an *adjacent* basis, i.e., a basis that can be obtained from the current one by removing one column and adding another one whose cost is no worse, until we reach an optimal basis. To formalize the algorithm we need to specify:

- how to recognize an optimal basis (optimality conditions);
- how to move to a *better* adjacent basis.

3.2.1 *Optimality conditions*

Starting from (2), we can partition the objective function as:

$$\begin{aligned} z &= c_B^\top x_B + c_R^\top x_R \\ &= c_B^\top B^{-1} b + (c_R^\top - c_B^\top B^{-1} R) x_R \end{aligned} \tag{3}$$

Thus, the objective function is now expressed as a constant term plus a linear function of the non-basic variables only. The vector

$$\pi^\top = c_B^\top B^{-1} \tag{4}$$

is called *vector of multipliers* and it conveniently allows the definition of the coefficients d_j in the objective function (3). Such coefficients are called *reduced costs* and are defined as:

$$d_j = c_j - \pi^\top A_j \tag{5}$$

It is easy to show that the reduced costs of basic variables are always zero.

The reduced costs of the non-basic variables give an optimality condition for the current basis. Let us assume to set one of the non-basic variables $x_j, j \in \mathcal{R}$ to a strictly positive value. From (3), we can deduce that the objective function z will increase or decrease depending on whether $d_j > 0$ or $d_j < 0$. We can thus conclude that if

$$d_j \geq 0 \quad \forall j \in \mathcal{R} \tag{6}$$

then no non-basic variable can move away from zero and improve the objective value, and thus the current basis is optimal. Note that the condition above is only a sufficient condition for optimality, but it is possible to show that there always exists an optimal basis that satisfies it.

3.2.2 *Finding a better basis*

If the current basis B does not satisfy the optimality condition (6), then there exist a non-basic variable x_q with negative reduced cost $d_q < 0$. We can thus try to increase x_q while still satisfying equations (1) and the non negativity of basic variables, and improve the objective value as much as possible.

Let $t \geq 0$ be the (*displacement*) of variable x_q. Given that all other non-basic variables stay at zero, we can rewrite equations (1) as:

$$Bx_B(t) + tA_q = b \tag{7}$$

from which

$$\begin{aligned} x_B(t) &= B^{-1}b - tB^{-1}A_q \\ &= \beta - t\alpha_q \end{aligned} \tag{8}$$

introducing the notation $\beta = B^{-1}b$ and $\alpha_q = B^{-1}A_q$. Since we want to maintain the feasibility of the basic variables, it must hold that $x_B^i(t) = \beta_i - t\alpha_q^i \geq 0$ for all $i = 1, \ldots, m$, and this set of m inequalities defines the maximum value for t. Given

$$\mathcal{I} = \{i : \alpha_q^i > 0\} \tag{9}$$

we can compute the maximum value θ for t as

$$\theta = \frac{\beta_p}{\alpha_q^p} = \min_{i \in \mathcal{I}} \left\{ \frac{\beta_i}{\alpha_q^i} \right\} \tag{10}$$

Note that the pivot element is not *necessarily unique.*

This operation is the so-called *ratio test*. The p-th basic variable is the *blocking variable* and the element α_q^p is called *pivot element*. For $t = \theta$ the blocking variable becomes zero (like a non-basic variable): we can thus say that variable x_{k_p} exits the basis and is replaced by variable x_q. The new objective value is $\overline{z} = z + \theta d_q$. It is easy to show that the new set of columns obtained by replacing A_{k_p} with A_q is still a basis: indeed, since $B\alpha_q = A_q$, we have:

$$\sum_{i \neq p} B_i \alpha_q^i + A_{k_p} \alpha_q^p = A_q$$

Given that $\alpha_q^p \neq 0$, we thus have:

$$A_{k_p} = \frac{1}{\alpha_q^p} A_q - \sum_{i \neq p} B_i \frac{\alpha_q^i}{\alpha_q^p}$$

so any vector generated by B can also be generated by the set of vectors.

If $\mathcal{I} = \emptyset$, then there is no restriction on t and thus $t = +\infty$. In this case the objective function value z can decrease indefinitely and the problem is *unbounded*. Note also that in case of degeneracy, it may happen that $\beta_p = 0$, and thus we have a zero step length θ: in this case we change the basis but the basic feasible solution stays the same. Those are called *degenerate* simplex iterations.

3.2.3 *Algorithmic Description*

We can thus define the *revised primal simplex* algorithmically as follows:

STEP 0 (INIT) Given a feasible basis B, compute B^{-1}, $\beta = B^{-1}b$ and $z = c_B^\top x_B$.

STEP 1 Compute the vector of multipliers $\pi^\top = c_B^\top B^{-1}$.

STEP 2 (PRICING) Compute the reduced costs $d_j = c_j - \pi^\top A_j \ \forall j \in \mathcal{R}$. If $d_j \geq 0 \ \forall j$ then the current basis is optimal and we can stop. Otherwise choose an entering variable x_q with $d_q < 0$.

STEP 3 Compute $\alpha_q = B^{-1} A_q$.

STEP 4 (PIVOT STEP) Define $\mathcal{I} = \{i : \alpha_q^i > 0\}$. If $\mathcal{I} = \emptyset$ the problem is unbounded. Otherwise apply the ratio test (10), computing θ and the leaving variable x_{k_p}.

STEP 5 (UPDATE) Update the basis B and its inverse. Then update the quantities β and z

$$\beta_i = \beta_i - \theta \alpha_q^i \quad i \neq p \tag{11}$$

$$\beta_p = \theta \tag{12}$$

$$z = z + \theta d_q \tag{13}$$

Go to Step 1.

3.2.4 *Initialization*

The algorithm described so far assumes the availability of a starting basis B. What if one such basis is not readily available? In this case, we can resort to the so-called *two-phase* simplex method, where in *phase I* we solve an auxiliary problem whose purpose is to find a feasible basis, which will be the starting basis for *phase II*. Let us start with a linear program in standard form:

$$\begin{aligned} \min\ & c^\top x \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

where we further assume, w.l.o.g, that $b \geq 0$. Now, we can drop the original objective function, and add one artificial variable y_i for every row in the matrix, and minimize the sum of those variables:

$$\min e^\top y$$
$$Ax + Iy = b$$
$$x, y \geq 0$$

Note that this phase I problem has the same number of rows of the original problem, it is itself in standard form, and a primal feasible basis is readily available in the identity matrix associated with the artificial variables y. The primal simplex method can thus be directly applied. Let us assume now that this preliminary problem has been solved to optimality, with optimal value w^* and optimal solution (x^*, y^*). There are two possibilities:

1. $w^* > 0$. This implies that there is no feasible solution to the system with $y = 0$, and hence the original problem is infeasible.

And thus x^* is a feasible solution of the original problem.

2. $w^* = 0$. In this case we have $y^* = 0$. If all artificial variables y are non-basic, we have a feasible basis of the original system readily available, from which we can start phase II. If some variable y_i is basic (indicating a degenerate basis), then it can be shown (we do not give the details) that if the matrix has full row rank those artificial variables can be *pivoted out* of the basis, bringing in some x variables at zero value, with some degenerate simplex iterations, until we are back to the case in which all artificial variables are non-basic.

3.2.5 *Convergence*

In the absence of degeneracy, the primal simplex method makes a positive progress toward the objective at every iteration: thus it cannot pass through the same basis more than once, and since the number of bases is finite, this implies convergence in a finite number of steps, albeit an exponentially large one in the worst case.

Unfortunately, degeneracy cannot be ruled out as a pathological case: it does happen in practice, and it is even common in certain applications of LP. In case of degeneracy, there is thus the concrete risk that the simplex method would pass through the same basis twice via a sequence of degenerate pivots: being the simplex method a deterministic algorithm, this means that it would cycle forever, without making any progress.

There are two (completely different) ways to deal with degeneracy and avoid cycling:

1. *Anti-cycling rules.* The idea is to adopt some strategy, when choosing which variable is leaving and which is entering the basis, that guarantees the impossibility of cycling. We note that in the simplex methods there are two degrees of freedom: when choosing the entering variable, there is usually more than one variable with negative reduced cost $d_q < 0$, while when doing the ratio test there is often more than one variable that achieves the minimum ratio, and is thus candidate for leaving. A very elegant pivoting rule that avoids cycling, and thus gives a proven convergent method, is the so-called *Bland*'s rule: whenever given the choice, always pick the variable with minimum index. Unfortunately, the practical performance of this method is terrible: the two degrees of freedom are heavily exploited by state-of-the-art implementations to achieve faster (practical) convergence and numerical stability, and Bland's rule would force them to give up on all of those completely.

2. *Perturbations.* This is the method implemented in most LP solvers: when cycling is detected, the problem is *perturbed* so has to break the cycle. Of course, after some time, the perturbation needs to be removed.

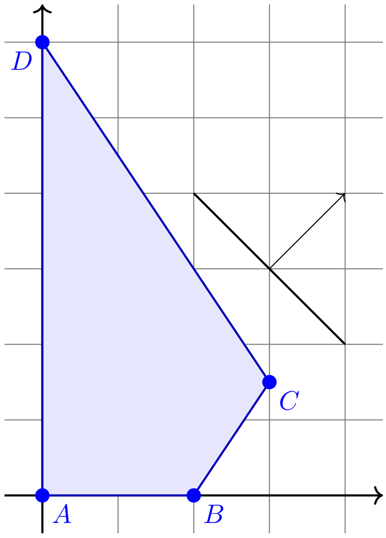

Figure 3.3: Geometrical interpretation of the numerical example.

The primal simplex method is one of the most extreme cases of difference between theoretical worst-case behaviour of an algorithm and practical performance. From the theoretical point of view, examples can be constructed, like the famous Klee and Minty perturbed hypercube, where the simplex method is forced to go through exponentially many bases before reaching the optimal one. In practice, however, its behaviour is completely different: computational experiments have shown time and again that modern implementations of the method achieve convergence in a number of iterations that is almost linear in the size of the problem. For this reason, the simplex method is still one of the most used methods for solving linear programs.

Independent of degeneracy!

And others that will become clear in the next chapters.

3.2.6 *Numerical Example*

Let us consider the following LP:

$$\begin{aligned} \max\ & x_1 + x_2 \\ & 6x_1 + 4x_2 \leq 24 \\ & 3x_1 - 2x_2 \leq 6 \\ & x_1, x_2 \geq 0 \end{aligned}$$

The geometrical interpretation is depicted in Figure 3.3. We start by bringing the problem in standard form:

$$\begin{aligned} \min\ & -x_1 - x_2 \\ & 6x_1 + 4x_2 + x_3 = 24 \\ & 3x_1 - 2x_2 + x_4 = 6 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

and noticing the newly introduced slack variables make a valid primal feasible starting basis $\mathcal{B} = \{x_3, x_4\}$. The corresponding bfs is $x = (0, 0, 24, 6)$, of objective value $z = 0$. This bfs corresponds to vertex A in the figure.

Iteration 1: The first step consists in computing the vector of multipliers $\pi^\top = c_B^\top B^{-1} = [0, 0]$. The corresponding reduced costs for the non-basic variables x_1 and x_2, obtained as $d_j = c_j - \pi^\top A_j$, have value $d_1 = -1$ and $d_2 = -1$. The optimality test $d \geq 0$ fails, so we can pick one of the two as *entering* variable: let us pick x_1. We need to compute the quantities needed for the ratio test, namely β and α_1:

$$\beta = B^{-1}b = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 24 \\ 6 \end{bmatrix} = \begin{bmatrix} 24 \\ 6 \end{bmatrix} \qquad \alpha_1 = B^{-1}A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

The ratio test thus reads:

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 24 \\ 6 \end{bmatrix} - \begin{bmatrix} 6 \\ 3 \end{bmatrix} t \geq 0$$

All coefficients in α_1 are non-negative, so $\mathcal{I} = \{3, 4\}$ and we compute the pivot step as:

$$\theta = \min \left\{ \frac{24}{6}, \frac{6}{3} \right\} = 2$$

and identify the *leaving* variable x_4. The new basis is thus $\mathcal{B} = \{x_3, x_1\}$. The corresponding bfs is $x = (2, 0, 12, 0)$, of objective value $z = -2$. This bfs corresponds to vertex B in the figure.

Iteration 2: Again, let us compute the vector of multipliers

$$\pi^\top = c_B^\top B^{-1} = \begin{bmatrix} 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & -2 \\ 0 & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{3} \end{bmatrix}$$

and the corresponding reduced costs:

$$d_2 = c_2 - \pi^\top A_2 = -\frac{5}{3}$$
$$d_4 = c_4 - \pi^\top A_4 = \frac{1}{3}$$

Only one reduced cost is negative, so the only candidate for entering the basis is x_2. We compute again the quantities for the ratio test:

$$\beta = B^{-1}b = \begin{bmatrix} 1 & -2 \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 24 \\ 6 \end{bmatrix} = \begin{bmatrix} 12 \\ 2 \end{bmatrix} \qquad \alpha_2 = B^{-1}A_2 = \begin{bmatrix} 1 & -2 \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 4 \\ -2 \end{bmatrix} = \begin{bmatrix} 8 \\ -\frac{2}{3} \end{bmatrix}$$

and thus the set of inequalities:

$$\begin{bmatrix} x_3 \\ x_1 \end{bmatrix} = \begin{bmatrix} 12 \\ 2 \end{bmatrix} - \begin{bmatrix} 8 \\ -\frac{2}{3} \end{bmatrix} t \geq 0$$

Here only one coefficient in α_2 is positive, and so we have $\mathcal{I} = \{3\}$ and $\theta = \frac{12}{8} = \frac{3}{2}$: x_3 leaves the basis. The new basis is thus $\mathcal{B} = \{x_2, x_1\}$. The corresponding bfs is $x = (3, \frac{3}{2}, 0, 0)$, of objective value $z = -\frac{9}{2}$. This bfs corresponds to vertex C in the figure.

Iteration 3: Let us compute the vector of multipliers

$$\pi^\top = c_B^\top B^{-1} = \begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{8} & -\frac{1}{4} \\ \frac{1}{12} & \frac{1}{6} \end{bmatrix} = \begin{bmatrix} -\frac{5}{24} & \frac{1}{12} \end{bmatrix}$$

and the corresponding reduced costs:

$$d_3 = c_1 - \pi^\top A_1 = \frac{5}{24}$$
$$d_4 = c_4 - \pi^\top A_4 = -\frac{1}{12}$$

Only one reduced cost is negative, so the only candidate for entering the basis is x_4. We compute the quantities for the ratio test:

$$\beta = B^{-1}b = \begin{bmatrix} \frac{1}{8} & -\frac{1}{4} \\ \frac{1}{12} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 24 \\ 6 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} \\ 3 \end{bmatrix} \qquad \alpha_4 = B^{-1}A_4 = \begin{bmatrix} \frac{1}{8} & -\frac{1}{4} \\ \frac{1}{12} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{4} \\ \frac{1}{6} \end{bmatrix}$$

and thus the set of inequalities:

$$\begin{bmatrix} x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} \\ 3 \end{bmatrix} - \begin{bmatrix} -\frac{1}{4} \\ \frac{1}{6} \end{bmatrix} t \geq 0$$

Again, only one coefficient in α_4 is positive, and so we have $\mathcal{I} = \{1\}$ and $\theta = \frac{3}{\frac{1}{6}} = 18$: x_1 leaves the basis. The new basis is thus $\mathcal{B} = \{x_2, x_4\}$. The corresponding bfs is $x = (0, 6, 0, 18)$, of objective value $z = -6$. This bfs corresponds to vertex D in the figure.

Iteration 4: Let us compute the new vector of multipliers

$$\pi^\top = c_B^\top B^{-1} = \begin{bmatrix} -1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{4} & 0 \\ \frac{1}{2} & 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{4} & 0 \end{bmatrix}$$

and the corresponding reduced costs:

$$\begin{aligned} d_1 &= c_1 - \pi^\top A_1 = \frac{1}{2} \\ d_3 &= c_3 - \pi^\top A_3 = \frac{1}{4} \end{aligned}$$

All the reduced costs are finally non-negative, so this is the optimal basis and the algorithm stops. The optimal solution, in the original space, is $x^* = (0, 6)$, of value $z^* = 6$.

Note that in this (toy) example we never encountered a degenerate basis, so we made positive progress at every iteration, and indeed moved from vertex to vertex. We remind that in general this is *not* the case: we move from basis to basis, without guarantees of positive progress at each step (if $\theta = 0$).

3.3 Duality

An optimization problem P is commonly interpreted as a *search* problem, which consists of "finding" the optimal solution x^*. Alternatively, the same problem can be viewed as an *inference* problem, which consists of deriving from the set of constraints of P (and the domain of the variables) the tightest possible lower bound on the objective function $f(x)$. The problem of finding the *proof* that yields the tightest lower bound is called the *dual* problem. This point of view is a cornerstone of mathematical programming, and optimization in general.

In this section, we will apply this dual point of view to linear programming: we will see that in this case the theory allow us to derive some strong results, in a relative simple setting, that have however a great theoretical and practical value.

Let us consider again an LP in standard form:

$$\begin{aligned} \min\ & c^\top x \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

Denote with P the feasible region defined by the constraints $Ax = b, x \geq 0$.

Definition 3.2. *Given a set $C \subseteq \mathbb{R}^n$, we say that a linear inequality $\alpha^\top x \geq \beta$ is* valid *for C, if it is satisfied by every point $x \in C$. Equivalently:*

$$\alpha^\top x \geq \beta \quad \forall x \in C$$

The dual point of view consists in trying to derive, from the constraints $Ax = b, x \geq 0$ defining the feasible region P of the problem, the tightest possible lower bound on the objective function, i.e., an inequality $c^\top x \geq c_0$ valid for P and with the highest possible value for c_0. We can thus naturally define a maximization problem of the form:

$$\begin{aligned} &\max c_0 \\ &\quad c^\top x \geq c_0 \ \forall x \in P \end{aligned}$$

Note that this a linear program with a single variable, c_0, and infinitely many constraints, one for every point in P. Using the Minkowski-Weyl theorem, we can already simplify it considerably, and reduce the number of inequalities to a finite number: if an inequality is valid for the set of vertices of P, and it will be valid for all the points in P, and thus we have the (proper) LP:

Prove it!

$$\begin{aligned} &\max c_0 \\ &\quad c^\top v^j \geq c_0 \ \forall v_j \text{ vertex of } P \end{aligned}$$

Unfortunately, as we know well, this is still impractical, as the number of vertices of a polyhedron is usually exponential in the H description of P. So we need to find a different (algebraic) characterization of the valid inequalities of P.

Let us start with a (very simple) argument that gives a sufficient condition for validity. Take the linear system $Ax = b$ and choose any vector $u \in \mathbb{R}^m$ of multipliers: clearly the equality

$$u^\top Ax = u^\top b$$

is a valid for any point pf P by construction. Starting from this equation, we can obtain valid inequalities for P by:

1. relaxing the equation into a $\geq$ inequality;
2. reducing the right-hand side;
3. since $x \geq 0$, increasing the coefficients of the x variables can only increase the left-hand side, maintaining the validity of the inequality.

In other words, one way to show that a given inequality $\alpha^\top x \geq \beta$ is valid for P is to find a vector of multipliers $u \in \mathbb{R}^m$ such that:

$$\alpha^\top \geq u^\top A \quad u^\top b \geq \beta$$

Note that, at the moment, this is only a *sufficient* condition: if we can find a vector of multipliers satisfying the condition above, then we have a proof that our inequality is valid, but what if we cannot find such a vector? Can the inequality be shown to be valid via some other argument? Perhaps surprisingly, it turns out that this cannot happen, and that the condition above is also *necessary*, a result known in the literature as the *Farkas' lemma*.

Theorem 3.3 (Farkas' lemma)**.** *An inequality $\alpha^\top x \geq \beta$ is valid for the polyhedron in standard form $P = \{Ax = b, x \geq 0\}$ if and only if there exist a vector $u \in \mathbb{R}^m$ such that:*

$$\alpha^\top \geq u^\top A \tag{14}$$

$$\beta \leq u^\top b \tag{15}$$

Proof. The condition is clearly sufficient:

$$\alpha^\top x \geq u^\top Ax = u^\top b \geq \beta$$

As for necessity, since the inequality is valid, we clearly have

This excludes $z^* = -\infty$.

$$\beta \leq z^* \min\{\alpha^\top x : x \in P\}$$

Let x^* be the optimal solution as computed by the primal simplex algorithm, with associated basis B: note that this is always well defined for the convergence properties of the simplex method. After partitioning the variables into basic and non-basic, as usual, we can define the vector of multipliers $u = \alpha_B^\top B^{-1}$, which is easily shown to verify the conditions (14) and (15) above. Indeed, condition (14) is just stating that all reduced costs are non-negative, which is the optimality criterion of the simplex method, while (15) is obtained as:

$$\beta \leq z^* = \alpha^\top x = \alpha_B^\top B^{-1} b = u^\top b$$

□

As a corollary of the Farkas' lemma, we have that the dual problem can be rewritten into:

$$\begin{aligned} &\max u^\top b \\ &\quad c^\top \geq u^\top A \end{aligned}$$

Not only this new formulation has polynomial size, but it actually has the same size of the original LP, with the constraint matrix A that gets transposed, and the objective and right-hand side vectors that swap their roles.

3.3.1 *Properties*

We have seen above that to each *primal* LP in standard form it is possible to associate another *dual* LP. Of course, the whole idea does not just apply to problems in standard form, but it can be applied to any LP, following the very same argument with the due adjustments. For example, if the primal problem has $\geq$ inequalities (resp. $\leq$), the corresponding multipliers have to be ≥ 0 (resp. ≤ 0). Along the same lines, if the variables x are free (instead of non-negative), then dual constraints turn into equations, as increasing the coefficients on the left-hand side is no longer guaranteed to preserve the validity of the inequality.

With the rules above, it is easy to show that we can the dual of the dual coincides with the primal, and hence linear programs come in so-called *primal-dual* pairs. Again, note that those problems are defined by the same data (A, b, c), where however the two vectors swap their roles and the matrix is transposed. The fundamental relation between the primal and dual LPs is given by the following two results:

Theorem 3.4 (Weak Duality)**.** *Let us denote with $P = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ and $D = \{u \in \mathbb{R}^m \mid c^\top \geq u^\top A\}$ the feasible regions of the primal and dual problem, respectively. Then, for any $\overline{x} \in P$ and for any $\overline{u} \in D$ we have $c^\top \overline{x} \geq \overline{u}^\top b$.*

Theorem 3.5 (Strong Duality)**.** *The primal problem admits a finite optimal solution x^* if and only the dual problem admits a finite optimal solution u^*, and when both have a finite optimum, the optimal values coincide, i.e., $c^\top x^* = u^{*\top} b$.*

Weak duality follows directly from the linear constraints defining the two problems, and it tells us that any feasible solution for one problem can be turned into a bound on the optimal value for the other. As a special case, it also shows that if one problem is unbounded, the other must necessarily be infeasible (by definition, no bound can exist). Note that it also applies to the optimal solutions of the two problems, where however we have the stronger result of strong duality, which follows directly from the constructive proof of the Farkas' lemma.

3.3.2 *Interpretation*

Note that, for an optimization problem, it is very easy to check that a given solution x^* is feasible: we just need to verify that all the constraints are satisfied. It is in general much more difficult to verify optimality, as optimality proofs might have exponential size (or not exist at all). Because of strong duality, this cannot happen for LP: given an optimal solution x^*, we can always construct an optimal dual solution u^*, of polynomial size, from which anyone can easily verify the optimality of x^*. Note also that the size of this proof is independent of how much work we had to spend to find it. Interestingly, we don't even have to invest additional effort to construct the dual proof, as the simplex method itself computes the optimal dual proof as a byproduct of its stopping criterion, so in some sense we get this information for free. This also shows how important duality is, as we can now realize that the construction of a dual feasible proof is at the heart of the primal simplex method. This will be a recurring pattern, as most algorithms for (convex) optimization are designed around the construction of this *primal-dual* optimal pair.