1 The Simplex Method

We will present an algorithm to solve linear programs of the form

$$\begin{aligned} \text{maximize} \quad & c^{\mathsf{T}} x \\ \text{subject to} \quad & Ax \preceq b \\ & x \succeq 0 \end{aligned} \tag{1}$$

assuming that $b \succeq 0$, so that $x = 0$ is guaranteed to be a feasible solution. Let n denote the number of variables and let m denote the number of constraints.

A simple transformation modifies any such linear program into a form such that each variable is constrained to be non-negative, and all other linear constraints are expressed as *equations* rather than *inequalities*. The key is to introduce additioinal variables, called *slack variables* which account for the difference between and left and right sides of each inequality in the original linear program. In other words, linear program (1) is equivalent to

$$\begin{aligned} \text{maximize} \quad & c^{\mathsf{T}} x \\ \text{subject to} \quad & Ax + y = b \\ & x, y \succeq 0 \end{aligned} \tag{2}$$

where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$.

The solution set of $\{Ax + y = b,\ x \succeq 0,\ y \succeq 0\}$ is a polytope in the $(n+m)$-dimensional vector space of ordered pairs $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$. The simplex algorithm is an iterative algorithm to solve linear programs of the form (2) by walking from vertex to vertex, along the edges of this polytope, until arriving at a vertex which maximizes the objective function $c^{\mathsf{T}} x$.

To illustrate the simplex method, for concreteness we will consider the following linear program.

$$\begin{aligned} \text{maximize} \quad & 2x_1 + 3x_2 \\ \text{subject to} \quad & x_1 + x_2 \le 8 \\ & 2x_1 + x_2 \le 12 \\ & x_1 + 2x_2 \le 14 \\ & x_1, x_2 \ge 0 \end{aligned}$$

This LP has so few variables, and so few constraints, it is easy to solve it by brute-force enumeration of the vertices of the polytope, which in this case is a 2-dimensional polygon.

The vertices of the polygon are $\left[\begin{smallmatrix}0\\7\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}2\\6\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}4\\4\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}6\\0\end{smallmatrix}\right]$, $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$. The objective function $2x_1 + 3x_2$ is maximized at the vertex $\left[\begin{smallmatrix}2\\6\end{smallmatrix}\right]$, where it attains the value 22. It is also easy to certify that this is the optimal value, given that the value is attained at $\left[\begin{smallmatrix}2\\6\end{smallmatrix}\right]$: simply add together the inequalities

$$\begin{aligned} x_1 + x_2 &\le 8 \\ x_1 + 2x^2 &\le 14 \end{aligned}$$

to obtain

$$2x_1 + 3x_2 \le 22,$$

which ensures that no point in the feasible set attains an objective value greater than 22.

To solve the linear program using the simplex method, we first apply the generic transformation described earlier, to rewrite it in *equational form* as

$$\begin{aligned} \text{maximize} \quad & 2x_1 + 3x_2 \\ \text{subject to} \quad & x_1 + x_2 + y_1 = 8 \\ & 2x_1 + x_2 + y_2 = 12 \\ & x_1 + 2x_2 + y_3 = 14 \\ & x_1, x_2, y_1, y_2, y_3 \ge 0 \end{aligned}$$

From now on, we will be partitioning the five variables into three which are allowed to take non-zero values (called the *basic variables* or *basis*) and two whose values will be set to zero (called the *non-basic variables* or *non-basis*). We will use the linear equations to express the basic variables, as well as the objective function, as affine functions of the non-basic variables. Initially the non-basis is $\{x_1, x_2\}$ and the linear program can be written in the form

$$\begin{aligned} \text{maximize} \quad & 2x_1 + 3x_2 \\ \text{subject to} \quad & y_1 = 8 - x_1 - x_2 \\ & y_2 = 12 - 2x_1 - x_2 \\ & y_3 = 14 - x_1 - 2x_2 \\ & x_1, x_2, y_1, y_2, y_3 \ge 0 \end{aligned}$$

which emphasizes that each of y_1, y_2, y_3 is determined as a function of x_1, x_2. Now, as long as the non-basis contains a variable which has a positive coefficient in the objective function, we select one such variable and greedily increase its value until one of the non-negativity constraints becomes tight. At that point, one of the other variables attains the value zero: it leaves the basis, and the variable whose value we increased enters the basis. For example, we could choose to increase x_1 from 0 to 6, at which point $y_2 = 0$. Then the new non-basis becomes $\{y_2, x_2\}$. Rewriting the equation $y_2 = 12 - 2x_1 - x_2$ as

$$x_1 = 6 - \frac{1}{2}y_2 - \frac{1}{2}x_2, \tag{3}$$

we may substitute the right side of (3) in place of x_1 everywhere in the above linear program, arriving at the equivalent form

$$\begin{aligned} \text{maximize} \quad & 12 - y_2 + 2x_2 \\ \text{subject to} \quad & y_1 = 2 + \tfrac{1}{2}y_2 - \tfrac{1}{2}x_2 \\ & x_1 = 6 - \tfrac{1}{2}y_2 - \tfrac{1}{2}x_2 \\ & y_3 = 8 + \tfrac{1}{2}y_2 - \tfrac{3}{2}x_2 \\ & x_1, x_2, y_1, y_2, y_3 \geq 0 \end{aligned}$$

At this point, x_2 still has a positive coefficient in the objective function, so we increase x_2 from 0 to 4, at which point $y_1 = 0$. Now x_2 enters the basis, and the new non-basis is $\{y_1, y_2\}$. We use the equation $x_2 = 4 + y_2 - 2y_1$ to substitute a function of the non-basic variables in place of x_2 everywhere it appears, arriving at the new linear program

$$\begin{aligned} \text{maximize} \quad & 20 - 4y_1 + y_2 \\ \text{subject to} \quad & x_2 = 4 + y_2 - 2y_1 \\ & x_1 = 4 - y_2 + y_1 \\ & y_3 = 2 - y_2 + 3y_1 \\ & x_1, x_2, y_1, y_2, y_3 \geq 0 \end{aligned}$$

Now we increase y_2 from 0 to 2, at which point $y_3 = 0$ and the new non-basis is $\{y_1, y_3\}$. Substituting $y_2 = 2 - y_3 + 3y_1$ allows us to rewrite the linear program as

$$\begin{aligned} \text{maximize} \quad & 22 - y_1 - y_3 \\ \text{subject to} \quad & x_2 = 6 + y_1 - y_3 \\ & x_1 = 2 - 2y_1 + y_3 \\ & y_2 = 2 + 3y_1 - y_3 \\ & x_1, x_2, y_1, y_2, y_3 \geq 0 \end{aligned} \tag{4}$$

At this point, there is no variable with a positive coefficient in the objective function, and we stop.

It is trivial to verify that the solution defined by the current iteration—namely, $x_1 = 2$, $x_2 = 6$, $y_1 = 0$, $y_2 = 2$, $y_3 = 0$—is optimal. The reason is that we have managed to write the objective function in the form $22 - y_1 - y_3$. Since the coefficient on each of the variables y_1, y_3 is negative, and y_1 and y_3 are constrained to take non-negative values, the largest possible value of the objective function is achieved by setting both y_1 and y_3 to zero, as our solution does.

More generally, if the simplex method terminates, it means that we have found an equivalent representation of the original linear program (2) in a form where the objective function attaches a non-positive coefficient to each of the non-basic variables. Since the non-basic variables are required to be non-negative, the objective function is maximized by setting all the non-basic variables to zero, which certifies that the solution at the end of the final iteration is optimal.

There is another way that the simplex method can terminate, which is not illustrated by the example above. It may happen, in one iteration, that we choose a variable with a positive coefficient in the objective function, and we find that we can increase this variable to an arbitrarily large value without violating any of the non-negativity constraints. (This situation happens when the increasing variable appears with a non-negative coefficient in each of the equations that defines the basic variables as an affine function of the non-basic variables.) In that case, we have verified that the optimum of the linear program is equal to $+\infty$, i.e. the objective function $c^\intercal x$ takes unboundedly large values as x ranges over the set of vectors satisfying the constraints.

Note that, in our running example, the final objective function assigned coefficient -1 to both y_1 and y_3. This is closely related to the fact that the simple "certificate of optimality" described above (before we started running the simplex algorithm) we obtained by summing the first and third inequalities of the original linear program, each with a coefficient of 1. We will see in the following section that this is not a coincidence.

Before leaving this discussion of the simplex method, we must touch upon a subtle issue regarding the question of whether the algorithm always terminates. A basis is an m-element subset of $n + m$ variables, so there are at most $\binom{n+m}{m}$ bases; if we can ensure that the algorithm never returns to the same basis as in a previous iteration, then it must terminate. Note that each basis determines a unique point $(x, y) \in \mathbb{R}^{n+m}$—defined by setting the non-basic variables to zero and assigning to the remaining variables the unique values that satisfy the equation $Ax + y = b$—and as the algorithm proceeds from basis to basis, the objective function value at the corresponding points never decreases. If the objective function strictly increases when moving from basis B to basis B', then the algorithm is guaranteed never to return to basis B, since the objective function value is now strictly greater than its value at B, and it will never decrease. On the other hand, it is possible for the simplex algorithm to shift from one basis to a different basis with the same objective function value; this is called a *degenerate pivot*, and it can only happen when there is a basic variable whose value is 0 at the current solution.

There exist *pivot rules* (i.e., rules for selecting the next basis in the simplex algorithm) that are designed to avoid infinite loops of degenerate pivots. Perhaps the simplest such rule is *Bland's rule*, which always chooses the entering variable to be the lowest-numbered non-basic variable that has a positive coefficient in the objective function. Similarly, in case the leaving variable is not uniquely determined, Bland's rule chooses the lowest-numbered possibility. Although the rule is simple to define, proving that it avoids infinite loops is not easy, and we will omit the proof from these notes. Instead, the following section is devoted to presenting a pivot rule that is much more algorithmically costly, but leads to an easier and more conceptual proof of termination.

1.1 A non-cycling pivot rule based on infinitesimals

In order for a degenerate pivot to be possible when solving a given linear program using the simplex method, the equation $Ax+y=b$ must have a solution in which $n+1$ or more of the variables take the value 0. Generically, a system of m linear equations in $m+n$ unknown does not have solutions with strictly more than n of the variables equal to 0. If we modify the linear system $Ax+y=b$ by perturbing it slightly, we should expect that such a modification will, generically, eliminate the possibility of encountering degenerate pivots when running the simplex algorithm.

These algebraic considerations can also be visualized in geometric terms. The linear system $Ax+y=b$ consists of m linearly independent equations in $m+n$ unknown, so its solution set is an n-dimensional subspace of the $m+n$-dimensional vector space of all (x,y) pairs. In this n-dimensional space, the inequalities $x \succeq 0, y \succeq 0$ define $m+n$ half-spaces, each bounded by a hyperplane of the form $\{x_j = 0\}$ or $\{y_i = 0\}$. In n-dimensional space, n hyperplanes in general position will intersect at a single point, whereas $n+1$ or more hyperplanes in general position will have an empty intersection. For example, in three dimensions, any three planes in general position have one intersection point, whereas four planes in general position have an empty intersection. However, it is perfectly possible for four planes (not in general position) to have a non-empty intersection, as when the four faces of a square pyramid meet at its apex. If one were to run the simplex algorithm to optimize a linear function on a three-dimensional polyhedron such as a square pyramid, it is possible that one or more iterations of the algorithm would start at the apex of the pyramid, with a non-basis consisting of any three of the four slack variables corresponding to the four sides of the pyramid. A degenerate pivot would substitute a new non-basis by replacing one of these three slack variables with the slack variable that was omitted from the previous non-basis. After performing this operation, and setting the variables in the new non-basis equal to zero, we remain situated at the same vertex as before, namely the apex of the pyramid.

Now consider modifying the polyhedron by perturbing the height of each face of the pyramid by a very small amount — potentially a different small quantity for each of the faces. This operation modifies the shape of the top of the polyhedron slightly; the apex point is replaced with a very short line segment joining two vertices, as illustrated in Figure 1.

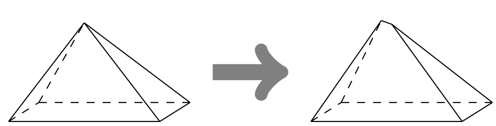

Figure 1: A pyramid and its generic perturbation

The modified polyhedron has no degenerate vertices; every vertex is situated at the intersection of three (and only three) faces. The degenerate pivot at the apex of the pyramid has been replaced by a non-degenerate pivot that makes progress by moving along the short line segment joining the two topmost vertices of the modified polyhedron.

The foregoing discussion about perturbations could potentially by implemented by adding a short random vector ϵ to the vector b occurring on the right side of the equation $Ax+y=b$, resulting in the new equation $Ax + y = b + \epsilon$. To justify that this works, one first needs to reason about the probability that the modified polyhedron has no degenerate vertices. This probability turns out to be equal to 1, assuming that the distribution of the vector ϵ is absolutely continuous with respect to Lebesgue measure, but that fact requires proof. Next one would need to reason about whether every vertex of the new polyhedron is situated near a vertex of the original polyhedron (i.e., at a distance tending to zero as the length of the vector ϵ tends to zero). This also turns out to be true, but again requires proof.

Instead of using probability, we will adopt a more algebraic formalism that gracefully sidesteps some of these issues. The idea will be to treat the coefficients of the linear program as belonging to an extension field of the real numbers that is still totally ordered, but contains infinitesimal numbers representing the perturbations.

Definition 1. An *ordered field* is a field $\mathbb{F}$ with a distinguished subset $\mathbb{F}_{>0}$, called the set of *positive elements*, that is closed under addition and multiplication, does not contain 0, and satisfies the property that for every non-zero $x \in \mathbb{F}$, exactly one of x and $-x$ is positive. One can define a total ordering on $\mathbb{F}$ by specifying that $x < x'$ if and only if $x' - x$ is positive. Inequalities defined in this way obey the usual algebraic rules for manipulating inequalities, for example the rule which asserts that the validity of an inequality is preserved under scaling both sides by the same positive scalar.

Definition 2. If $\mathbb{F}$ is an ordered field that contains $\mathbb{R}$ as a subfield, an element $x \in \mathbb{F}$ is called *finite* if there is some $r \in \mathbb{R}$ such that $-r < x < r$, and otherwise we refer to x as *infinite*. If $x \neq 0$ and x^{-1} is infinite we say that x is *infinitesimal*. The finite elements constitute a subring $\mathbb{F}_{<\infty}$ and the infinitesimals are an ideal in this subring. There is a surjective homomorphism $\mathbb{F}_{<\infty} \to \mathbb{R}$ whose kernel is the ideal of infinitesimal elements. This homomorphism is defined by mapping each finite $x \in \mathbb{F}$ to the real number

$$[x] \triangleq \inf\{r \in \mathbb{R} \mid r > x\}. \tag{5}$$

Lemma 1. *There exists an ordered field $\mathbb{F}$ that contains $\mathbb{R}$ as a subfield, and whose ideal of infinitesimal elements is an infinite-dimensional vector space over $\mathbb{R}$.*

Proof. Let ϵ be a formal variable (i.e., a symbol with no numerical meaning) and let $\mathbb{F}$ be the field of rational functions $\mathbb{R}(\epsilon)$. In other words, an element of $\mathbb{F}$ is an equivalence class of fractions P/Q where P and Q are both polynomials in the formal variable ϵ, with real coefficients, and $Q \neq 0$. The equivalence relation is defined by specifying that $P_1/Q_1 = P_2/Q_2$ if and only if $P_1Q_2 = P_2Q_1$. For example, $\frac{3-3\epsilon^2}{1-\epsilon^3}$ and $\frac{3+\epsilon}{1+\epsilon+\epsilon^2}$ are two expressions representing the same element of $\mathbb{R}(\epsilon)$.

The ordering of $\mathbb{F}$ is defined as follows. A non-zero polynomial $P = a_0 + a_1\epsilon + \cdots + a_n\epsilon^n$ is positive if and only if the first non-zero element of the coefficient sequence $a_0, a_1, \ldots, a_n$ is positive. A quotient P/Q is positive if and only if P and Q are either both positive or both negative.

Under this ordering, the monomials $\epsilon, \epsilon^2, \epsilon^3, \ldots$ constitute an infinite set of infinitesimal elements that are linearly independent over $\mathbb{R}$. □

We defined linear programs and the simplex algorithm using the field of real numbers, but the problem definition and the algorithm both make sense over any ordered field. In particular, we can let K be an ordered extension field of $\mathbb{R}$ that contains m linearly independent infinitesimal elements $\epsilon_1, \epsilon_2, \ldots, \epsilon_m$, and we can modify the linear program (2) by replacing the equation $Ax + y = b$ with $Ax + y = b + \epsilon$, where ϵ denotes the column vector $(\epsilon_1, \epsilon_2, \ldots, \epsilon_m)^\intercal$.

Claim 2. *Fix a matrix A and vector b over the real numbers, and fix an ordered extension field $\mathbb{F} \supset \mathbb{R}$ that contains m linearly independent infinitesimal elements $\epsilon_1, \epsilon_2, \ldots, \epsilon_m$. Let $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_m)^\intercal$. For any $x \in \mathbb{F}^n$ and $y \in \mathbb{F}^m$ satisfying $Ax + y = b + \epsilon$, at most n of the elements $x_1, \ldots, x_n, y_1, \ldots, y_m$ are equal to 0.*

Proof. For $i = 1, \ldots, m$ the equation

$$\epsilon_i = \sum_{j=1}^{n} a_{ij} x_j \;+\; y_i \;-\; b_i$$

shows that ϵ_i belongs to the $\mathbb{R}$-linear subspace of K generated by $\{1, x_1, \ldots, x_n, y_1, \ldots, y_m\}$. Of course, the element $1 \in \mathbb{F}$ also belongs to this subspace, which we will denote by L. Since $1, \epsilon_1, \ldots, \epsilon_m$ are linearly independent over $\mathbb{R}$, the dimension of L over $\mathbb{R}$ is at least $m + 1$. In particular, since the set $\{1, x_1, \ldots, x_n, y_1, \ldots, y_m\}$ contains a basis for L, it must contain at least $m + 1$ non-zero elements, and hence at most n elements are equal to 0. $\square$

Claim 3. *Fix $A, b, \mathbb{F}, \epsilon$ as in Claim 2. When one solves the linear program*

$$\begin{aligned} &\textit{maximize} && c^\intercal x \\ &\textit{subject to} && Ax + y = b + \epsilon \\ & && x, y \succeq 0 \end{aligned} \tag{6}$$

using the simplex method, there are no degnerate pivots and the algorithm terminates after performing at most $\binom{m+n}{n}$ pivots.

Proof. A degenerate pivot, by definition, occurs when $n + 1$ or more coordinates of the vectors x, y are non-zero. Claim 2 shows that this can never happen when (x, y) belongs to the solution set of $Ax + y = b + \epsilon$. Hence, there are no degenerate pivots. This means that the value of the objective function $c^\intercal x$ (interpreted as an element of the ordered field $\mathbb{F}$) strictly increases with each iteration of the algorithm. Consequently, no two iterations of the algorithm can use the same basis, and the number of iterations is bounded above by the number of bases, which is $\binom{m+n}{n}$. $\square$

A pivot rule that guarantees termination. To define a non-cycling pivot rule for the simplex method solving an ordinary linear program (i.e., with coefficients in $\mathbb{R}$) our strategy will be to run the simplex method in tandem on two linear programs: one defined over $\mathbb{R}$ with constraint set $\{Ax + y = b, x, y \succeq 0\}$, and another defined over the extension field $\mathbb{F}$ with constraint set $\{Ax + y = b + \epsilon, x, y \succeq 0\}$. Let us distinguish the two linear programs by calling the first one $LP_{\mathbb{R}}$ and the second one $LP_{\mathbb{F}}$. We run the simplex method to solve $LP_{\mathbb{F}}$, and we let $x^{(t)}, y^{(t)}$ denote the values of the vectors x, y at the start of iteration t, for $t = 0, 1, \ldots, T$, where T either denotes the final iteration of the algorithm, or the last iteration in which all components of the vector $x^{(T)}$ are finite.

Recall the homomorphism $x \mapsto [x]$ from $\mathbb{F}_{<\infty}$ to $\mathbb{R}$ defined in equation (5). We will abuse notation and apply this homomorphism to vector spaces as well: if x is a vector with components in $\mathbb{F}_{<\infty}$ we let $[x]$ denote the vector over $\mathbb{R}$ obtained by applying the operation $x_i \mapsto [x_i]$ to each component of x. Now take the sequence of vector pairs $\{x^{(t)}, y^{(t)}\}_{t=0}^{T}$ representing the execution of the simplex method solving $LP_{\mathbb{F}}$ and translate it to the sequence of vector pairs $\{[x^{(t)}], [y^{(t)}]\}_{t=0}^{T}$. This sequence represents an initial segment of a valid execution of the simplex method solving $LP_{\mathbb{R}}$. In every iteration, n of the variables in the set $\{[x_j^{(t)}] \mid j = 1, \ldots, n\} \cup \{[y_i^{(t)}] \mid i = 1, \ldots, m\}$ are equal to zero, because the n variables in the non-basis of the corresponding iteration of the $LP_{\mathbb{F}}$ are equal to 0, and $[0] = 0$. The remaining m variables in the set are non-negative, because their counterparts in the $LP_{\mathbb{F}}$ execution are non-negative, and the homomorphism $x \mapsto [x]$ preserves non-negativity.

To conclude our analysis we will show that $\{[x^{(t)}], [y^{(t)}]\}_{t=0}^{T}$ represents a *terminating* execution of the simplex method solving $LP_{\mathbb{R}}$. By the definition of T, we know that in iteration T the simplex method solving $LP_{\mathbb{F}}$ either terminates (because when the objective function is written in terms of the current non-basis, every coefficient is non-positive) or it chooses a variable in the non-basis and increases its value from 0 to an infinite element of $\mathbb{F}$. In the former case the simplex method solving $LP_{\mathbb{R}}$ also terminates at iteration T because its objective function coefficients are non-positive. (The homomorphism $x \mapsto [x]$ preserves non-positivity.) In the latter case, the simplex method solving $LP_{\mathbb{R}}$ terminates in iteration T because it discovers that the optimum is equal to $+\infty$.

1.2 The simplex method takes exponential time in the worst case

An example due to Klee and Minty illustrates that the simplex method can take exponential time in the worst case. Consider the linear program

$$\begin{array}{ll} \text{maximize} & x_n \\ \text{subject to} & 0 \le x_1 \le 1 \\ \delta x_i \le x_{i+1} \le 1 - \delta x_i & \text{for } 1 \le i < n \end{array} \tag{7}$$

Here, δ is a positive number less than 1/2.

The polyhedron defined by the constraints of this linear program is shaped like an n-dimensional hypercube with tilted sides, as depicted in Figure 2. It is called a *Klee-Minty cube*. There is an execution of the simplex method that visits each of the 2^n vertices of the Klee-Minty cube, starting from $(0, 0, \ldots, 0)$ and ending at $(0, 0, \ldots, 1)$. The sequence of vertices can be defined recursively as follows. The "back face" of the Klee-Minty cube, where the equation $x_n = \delta x_{n-1}$ is satisfied, is a copy of the $(n-1)$-dimensional Klee-Minty cube. We run an execution of the simplex method on this $(n-1)$-dimensional Klee-Minty cube, with the modified objective function x_{n-1}. The equation $x_n = \delta x_{n-1}$ guarantees that

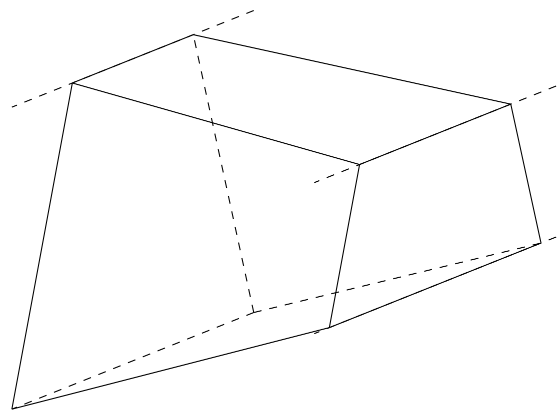

Figure 2: A Klee-Minty cube in three dimensions

the true objective function increases as the modified objective function increases, which means that this remains a valid execution of the simplex method on the back face of the n-dimensional Klee-Minty cube. After reaching the vertex $(0, 0, \dots, 1, \delta)$, we move from the back face to the front face, where the equation $x_n = 1 - \delta x_{n-1}$ is satisfied, arriving at the vertex $(0, 0, \dots, 1, 1 - \delta)$. Then we run through the vertices of the front face by reversing the order in which we visited the corresponding vertices on the back face. As we do this, x_{n-1} strictly decreases with each pivot; due to the equation $x_n = 1 - \delta x_{n-1}$ this means that x_n strictly increases with each pivot, confirming that this is a valid execution of the simplex algorithm.

2 The Simplex Method and Strong Duality

An important consequence of the correctness and termination of the simplex algorithm is *linear programming duality*, which asserts that for every linear program with a maximization objective, there is a related linear program with a minimization objective whose optimum matches the optimum of the first LP.

Theorem 4. *Consider any pair of linear programs of the form*

$$\begin{array}{llcll} \text{maximize} & c^{\intercal} x & & \text{minimize} & b^{\intercal} \eta \\ \text{subject to} & Ax \preceq b & \textit{and} & \text{subject to} & A^{\intercal} \eta \succeq c \\ & x \succeq 0 & & & \eta \succeq 0 \end{array} \tag{8}$$

If the optimum of the first linear program is finite, then both linear programs have the same optimum value.

Proof. Before delving into the formal proof, the following intuition is useful. If a_i denotes the i^{th} row of the matrix A, then the relation $Ax \preceq b$ can equivalently be expressed by stating that $a_i^{\intercal} x \leq b_i$ for $j = 1, \dots, m$. For any m-tuple of non-negative coefficients $\eta_1, \dots, \eta_m$, we can form a weighted sum of these inequalities,

$$\sum_{j=1}^{m} \eta_i a_i^{\mathsf{T}} x \leq \sum_{j=1}^{m} \eta_i b_i, \tag{9}$$

obtaining an inequality implied by $Ax \preceq b$. Depending on the choice of weights $\eta_1, \ldots, \eta_m$, the inequality (9) may or may not imply an upper bound on the quantity $c^{\mathsf{T}}x$, for all $x \succeq 0$. The case in which (9) implies an upper bound on $c^{\mathsf{T}}x$ is when, for each variable x_j ($j = 1, \ldots, n$), the coefficient of x_j on the left side of (9) is greater than or equal to the coefficient of x_j in the expression $c^{\mathsf{T}}x$. In other words, the case in which (9) implies an upper bound on $c^{\mathsf{T}}x$ for all $x \succeq 0$ is when

$$\forall j \in \{1, \ldots, n\} \quad \sum_{i=1}^{m} \eta_i a_{ij} \geq c_j. \tag{10}$$

We can express (9) and (10) more succinctly by packaging the coefficients of the weighted sum into a vector, η. Then, inequality (9) can be rewritten as

$$\eta^{\mathsf{T}} A x \leq \eta^{\mathsf{T}} b, \tag{11}$$

and the criterion expressed by (10) can be rewritten as

$$\eta^{\mathsf{T}} A \succeq c^{\mathsf{T}}. \tag{12}$$

The reasoning surrounding inequalities (9) and (10) can now be summarized by saying that for any vector $\eta \in \mathbb{R}^m$ satisfying $\eta \succeq 0$ and $\eta^{\mathsf{T}} A \succeq c^{\mathsf{T}}$, we have

$$c^{\mathsf{T}} x \leq \eta^{\mathsf{T}} A x \leq \eta^{\mathsf{T}} b \tag{13}$$

for all $x \succeq 0$ satisfying $Ax \preceq b$. (In hindsight, proving inequality (13) is trivial using the properties of the vector ordering $\preceq$ and our assumptions about x and η.)

Applying (13), we may immediately conclude that the minimum of $\eta^{\mathsf{T}} b$ over all $\eta \succeq 0$ satisfying $\eta^{\mathsf{T}} A \succeq c^{\mathsf{T}}$, is greater than or equal to the maximum of $c^{\mathsf{T}} x$ over all $x \succeq 0$ satisfying $Ax \preceq b$. That is, the optimum of the first LP in (8) is less than or equal to the optimum of the second LP in (8), a relation known as *weak duality*.

To prove that the optima of the two linear programs are equal, as asserted by the theorem, we need to furnish vectors x, η satisfying the constraints of the first and second linear programs in (8), respectively, such that $c^{\mathsf{T}} x = b^{\mathsf{T}} \eta$. To do so, we will make use of the simplex algorithm and its termination condition. At the moment of termination, the objective function has been rewritten in a form that has no positive coefficient on any variable. In other words, the objective function is written in the form $v - \xi^{\mathsf{T}} x - \eta^{\mathsf{T}} y$ for some coefficient vectors $\xi \in \mathbb{R}^n$ and $\eta \in \mathbb{R}^m$ such that $\xi, \eta \succeq 0$.

An invariant of the simplex algorithm is that whenever it rewrites the objective function, it preserves the property that the objective function value matches $c^{\mathsf{T}} x$ for all pairs $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$ such that $Ax + y = b$. In other words, we have

$$\forall x \in \mathbb{R}^n \quad v - \xi^{\mathsf{T}} x - \eta^{\mathsf{T}}(b - Ax) = c^{\mathsf{T}} x. \tag{14}$$

Equating the constant terms on the left and right sides, we find that $v = \eta^\intercal b$. Equating the cofficient of x_j on the left and right sides for all j, we find that $\eta^\intercal A = \xi^\intercal + c^\intercal \succeq c^\intercal$. Thus, the vector η satisfies the constraints of the second LP in (8).

Now consider the vector (x, y) which the simplex algorithm outputs at termination. All the variables having a non-zero coefficient in the expression $-\xi^\intercal x - \eta^\intercal y$ belong to the algorithm's non-basis, and hence are set to zero in the solution (x, y). This means that

$$v = v - \xi^\intercal x - \eta^\intercal y = c^\intercal x$$

and hence, using the relation $v = \eta^\intercal b$ derived earlier, we have $c^\intercal x = b^\intercal \eta$ as desired. □