5. The Simplex Method

In this chapter we explain the simplex method for solving linear programs. We will make use of the terms *equational form* and *basic feasible solution* from the previous chapter.

Gaussian elimination in linear algebra has a fundamental theoretical and didactic significance, as a starting point for further developments. But in practice it has mostly been replaced by more complicated and more efficient algorithms. Similarly, the basic version of the simplex method that we discuss here is not commonly used for solving linear programs in practice. We do not put emphasis on the most efficient possible organization of the computations, but rather we concentrate on the main ideas.

5.1 An Introductory Example

We will first show the simplex method in action on a small concrete example, namely, on the following linear program:

$$
\begin{array}{lrcrcl}
\text{Maximize} & x_1 & + & x_2 & & \\
\text{subject to} & -x_1 & + & x_2 & \leq & 1 \\
& x_1 & & & \leq & 3 \\
& & & x_2 & \leq & 2 \\
& \multicolumn{5}{l}{x_1, x_2 \geq 0.}
\end{array}
\tag{5.1}
$$

We intentionally do not take a linear program in equational form: The variables are nonnegative, but the inequalities have to be replaced by equations, by introducing slack variables. The equational form is

$$
\begin{array}{lrcrcrcrcrcl}
\text{maximize} & x_1 & + & x_2 & & & & & & & & \\
\text{subject to} & -x_1 & + & x_2 & + & x_3 & & & & & = & 1 \\
& x_1 & & & & & + & x_4 & & & = & 3 \\
& & & x_2 & & & & & + & x_5 & = & 2 \\
& \multicolumn{11}{l}{x_1, x_2, \ldots, x_5 \geq 0,}
\end{array}
$$

with the matrix

$$
A = \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.
$$

In the simplex method we first express each linear program in the form of a *simplex tableau*. In our case we begin with the tableau

$$\begin{array}{rcrcrcr} x_3 & = & 1 & + & x_1 & - & x_2 \\ x_4 & = & 3 & - & x_1 & & \\ x_5 & = & 2 & & & - & x_2 \\ \hline z & = & & & x_1 & + & x_2 \end{array}$$

The first three rows consist of the equations of the linear program, in which the slack variables have been carried over to the left-hand side and the remaining terms are on the right-hand side. The last row, separated by a line, contains a new variable z, which expresses the objective function.

Each simplex tableau is associated with a certain basic feasible solution. In our case we substitute 0 for the variables x_1 and x_2 from the right-hand side, and without calculation we see that $x_3 = 1, x_4 = 3, x_5 = 2$. This feasible solution is indeed basic with $B = \{3, 4, 5\}$; we note that A_B is the identity matrix. The variables x_3, x_4, x_5 from the left-hand side are basic and the variables x_1, x_2 from the right-hand side are nonbasic. The value of the objective function $z = 0$ corresponding to this basic feasible solution can be read off from the last row of the tableau.

From the initial simplex tableau we will construct a sequence of tableaus of a similar form, by gradually rewriting them according to certain rules. Each tableau will contain the *same* information about the linear program, only written differently. The procedure terminates with a tableau that represents the information so that the desired optimal solution can be read off directly.

Let us go to the first step. We try to increase the value of the objective function by increasing one of the nonbasic variables x_1 or x_2. In the above tableau we observe that increasing the value of x_1 (i.e. making x_1 positive) increases the value of z. The same is true for x_2, because both variables have positive coefficients in the z-row of the tableau. We can choose either x_1 or x_2; let us decide (arbitrarily) for x_2. We will increase it, while x_1 will stay 0.

By how much can we increase x_2? If we want to maintain feasibility, we have to be careful not to let any of the basic variables x_3, x_4, x_5 go below zero. This means that the equations determining x_3, x_4, x_5 may limit the increment of x_2. Let us consider the first equation

$$x_3 = 1 + x_1 - x_2.$$

Together with the implicit constraint $x_3 \geq 0$ it lets us increase x_2 up to the value $x_2 = 1$ (while keeping $x_1 = 0$). The second equation

$$x_4 = 3 - x_1$$

does not limit the increment of x_2 at all, and the third equation

$$x_5 = 2 - x_2$$

allows for an increase of x_2 up to $x_2 = 2$ before x_5 gets negative. The most stringent restriction thus follows from the first equation.

We increase x_2 as much as we can, obtaining $x_2 = 1$ and $x_3 = 0$. From the remaining equations of the tableau we get the values of the other variables:

$$\begin{aligned} x_4 &= 3 - x_1 = 3 \\ x_5 &= 2 - x_2 = 1. \end{aligned}$$

In this new feasible solution x_3 became zero and x_2 nonzero. Quite naturally we thus transfer x_3 to the right-hand side, where the nonbasic variables live, and x_2 to the left-hand side, where the basic variables reside. We do it by means of the most stringent equation $x_3 = 1 + x_1 - x_2$, from which we express

$$x_2 = 1 + x_1 - x_3.$$

We substitute the right-hand side for x_2 into the remaining equations, and we arrive at a new tableau:

$$\begin{array}{rcrcrcr} x_2 & = & 1 & + & x_1 & - & x_3 \\ x_4 & = & 3 & - & x_1 & & \\ x_5 & = & 1 & - & x_1 & + & x_3 \\ \hline z & = & 1 & + & 2x_1 & - & x_3 \end{array}$$

Here $B = \{2,4,5\}$, which corresponds to the basic feasible solution $\mathbf{x} = (0,1,0,3,1)$ with the value of the objective function $z = 1$.

This process of rewriting one simplex tableau into another is called a **pivot step**. In each pivot step some nonbasic variable, in our case x_2, *enters* the basis, while some basic variable, in our case x_3, *leaves* the basis.

In the new tableau we can further increase the value of the objective function by increasing x_1, while increasing x_3 would lead to a smaller z-value. The first equation does not restrict the increment of x_1 in any way, from the second one we get $x_1 \le 3$, and from the third one $x_1 \le 1$, so the strictest limitation is implied by the third equation. Similarly as in the previous step, we express x_1 from it and we substitute this expression into the remaining equations. Thereby x_1 enters the basis and moves to the left-hand side, and x_5 leaves the basis and migrates to the right-hand side. The tableau we obtain is

$$\begin{array}{rcrcrcr} x_1 & = & 1 & + & x_3 & - & x_5 \\ x_2 & = & 2 & & & - & x_5 \\ x_4 & = & 2 & - & x_3 & + & x_5 \\ \hline z & = & 3 & + & x_3 & - & 2x_5 \end{array}$$

with $B = \{1,2,4\}$, basic feasible solution $\mathbf{x} = (1,2,0,2,0)$, and $z = 3$. After one more pivot step, in which x_3 enters the basis and x_4 leaves it, we arrive at the tableau

$$\begin{array}{rcrcrcr} x_1 & = & 3 & - & x_4 & & \\ x_2 & = & 2 & & & - & x_5 \\ x_3 & = & 2 & - & x_4 & + & x_5 \\ \hline z & = & 5 & - & x_4 & - & x_5 \end{array}$$

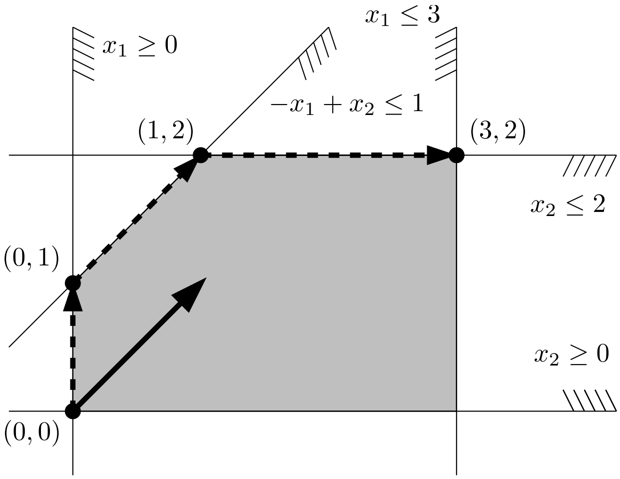
with basis $\{1,2,3\}$, basic feasible solution $\mathbf{x} = (3,2,2,0,0)$, and $z = 5$. In this tableau, no nonbasic variable can be increased without making the objective function value smaller, so we are stuck. Luckily, this also means that we have already found an optimal solution! Why?

Let us consider an *arbitrary* feasible solution $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_5)$ of our linear program, with the objective function attaining some value $\tilde{z}$. Now $\tilde{\mathbf{x}}$ and $\tilde{z}$ satisfy all equations in the final tableau, which was obtained from the original equations of the linear program by equivalent transformations. Hence we necessarily have

$$\tilde{z} \;= 5 - \tilde{x}_4 - \tilde{x}_5.$$

Together with the nonnegativity constraints $\tilde{x}_4, \tilde{x}_5 \ge 0$ this implies $\tilde{z} \le 5$. The tableau even delivers a proof that $\mathbf{x} = (3,2,2,0,0)$ is the *only* optimal solution: If $z = 5$, then $x_4 = x_5 = 0$, and this determines the values of the remaining variables uniquely.

A geometric illustration. For each feasible solution (x_1, x_2) of the original linear program (5.1) with inequalities we have exactly one corresponding feasible solution $(x_1, x_2, \ldots, x_5)$ of the modified linear program in equational form, and conversely. The sets of feasible solutions are isomorphic in a suitable sense, and we can thus follow the progress of the simplex method narrated above in a planar picture for the original linear program (5.1):



We can see the simplex method moving along the edges from one feasible solution to another, while the value of the objective function grows until it reaches the optimum. In the example we could also take a shorter route if we decided to increase x_1 instead of x_2 in the first step.

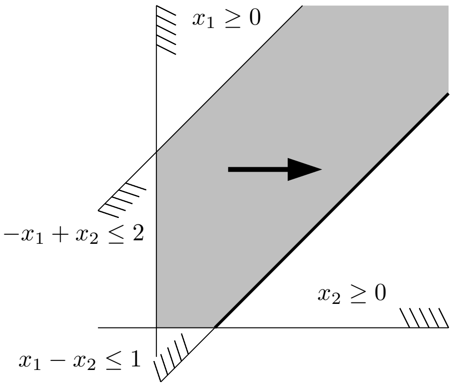
Potential troubles. In our modest example the simplex method has run smoothly without any problems. In general we must deal with several complications. We will demonstrate them on examples in the next sections.

5.2 Exception Handling: Unboundedness

What happens in the simplex method for an unbounded linear program? We will show it on the example

$$\begin{array}{ll} \text{maximize} & x_1 \\ \text{subject to} & \begin{array}{rcrcl} x_1 & - & x_2 & \le & 1 \\ -x_1 & + & x_2 & \le & 2 \end{array} \\ & x_1, x_2 \ge 0 \end{array}$$

illustrated in the picture below:



After the usual transformation to equational form by introducing slack variables x_3, x_4, we can use these variables as a feasible basis and we obtain the initial simplex tableau

$$\begin{array}{rcrcrcr} x_3 & = & 1 & - & x_1 & + & x_2 \\ x_4 & = & 2 & + & x_1 & - & x_2 \\ \hline z & = & & & x_1 & & \end{array}$$

After the first pivot step with entering variable x_1 and leaving variable x_3 the next tableau is

$$\begin{array}{rcrcrcr} x_1 & = & 1 & + & x_2 & - & x_3 \\ x_4 & = & 3 & & & - & x_3 \\ \hline z & = & 1 & + & x_2 & - & x_3 \end{array}$$

If we now try to introduce x_2 into the basis, we discover that none of the equations in the tableau restrict its increase in any way. We can thus take x_2 arbitrarily large, and we also get z arbitrarily large—the linear program is unbounded.

Let us analyze this situation in more detail. From the tableau one can see that for an arbitrarily large number $t \ge 0$ we obtain a feasible solution by setting $x_2 = t$, $x_3 = 0$, $x_1 = 1 + t$, and $x_4 = 3$, with the value of the objective function $z = 1 + t$. In other words, the semi-infinite ray

$$\{(1, 0, 0, 3) + t(1, 1, 0, 0) : t \ge 0\}$$

is contained in the set of feasible solutions. It "witnesses" the unboundedness of the linear program, since the objective function attains arbitrarily large values on it. The corresponding semi-infinite ray for the original two-dimensional linear program is drawn thick in the picture above.

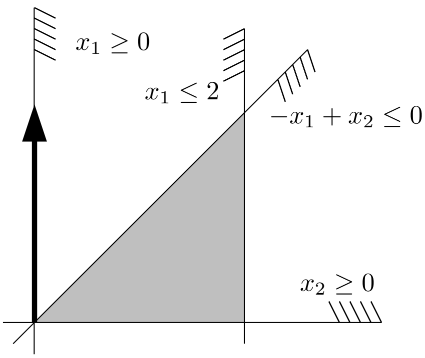
A similar ray is the output of the simplex method for all unbounded linear programs.

5.3 Exception Handling: Degeneracy

While we can make some nonbasic variable arbitrarily large in the unbounded case, the other extreme happens in a situation called a degeneracy: The equations in a tableau do not permit any increment of the selected nonbasic variable, and it may actually be impossible to increase the objective function z in a single pivot step.

Let us consider the linear program

$$\begin{array}{ll} \text{maximize} & x_2 \\ \text{subject to} & -x_1 + x_2 \leq 0 \\ & x_1 \leq 2 \\ & x_1, x_2 \geq 0. \end{array} \tag{5.2}$$



In the usual way we convert it to equational form and construct the initial tableau

$$\begin{array}{rcl} x_3 & = & x_1 - x_2 \\ x_4 & = & 2 - x_1 \\ \hline z & = & x_2 \end{array}$$

The only candidate for entering the basis is x_2, but the first row of the tableau shows that its value cannot be increased without making x_3 negative. Unfortunately, the impossibility of making progress in this case does not imply optimality, so we have to perform a degenerate pivot step, i.e., one with zero progress in the objective function. In our example, bringing x_2 into the basis (with x_3 leaving) results in another tableau with the same basic feasible solution $(0,0,0,2)$:

$$\begin{array}{rcrcrcr} x_2 & = & & & x_1 & - & x_3 \\ x_4 & = & 2 & - & x_1 & & \\ \hline z & = & & & x_1 & - & x_3 \end{array}$$

Nevertheless, the situation has improved. The nonbasic variable x_1 can now be increased, and by entering it into the basis (replacing x_4) we already obtain the final tableau

$$\begin{array}{rcrcrcr} x_1 & = & 2 & & & - & x_4 \\ x_2 & = & 2 & - & x_3 & - & x_4 \\ \hline z & = & 2 & - & x_3 & - & x_4 \end{array}$$

with an optimal solution $\mathbf{x} = (2,2,0,0)$.

A situation that forces a degenerate pivot step may occur only for a linear program in which several feasible bases correspond to a single basic feasible solution. Such linear programs are called **degenerate**.

It is easily seen that in order that a single basic feasible solution be obtained from several bases, some of the *basic* variables have to be zero.

In this example, after one degenerate pivot step we could again make progress. In general, there might be longer runs of degenerate pivot steps. It may even happen that some tableau is repeated in a sequence of degenerate pivot steps, and so the algorithm might pass through an infinite sequence of tableaus without any progress. This phenomenon is called **cycling**. An example of a linear program for which the simplex method may cycle can be found in Chvátal's textbook cited in Chapter 9 (the smallest possible example has 6 variables and 3 equations), and we will not present it here.

If the simplex method doesn't cycle, then it necessarily finishes in a finite number of steps. This is because there are only finitely many possible simplex tableaus for any given linear program, namely at most $\binom{n}{m}$, which we will prove in Section 5.5.

How can cycling be prevented? This is a nontrivial issue and it will be discussed in Section 5.8.

5.4 Exception Handling: Infeasibility

In order that the simplex method be able to start at all, we need a feasible basis. In examples discussed up until now we got a feasible basis more or less for free. It works this way for all linear programs of the form

$$\text{maximize } \mathbf{c}^T\mathbf{x} \text{ subject to } A\mathbf{x} \le \mathbf{b} \text{ and } \mathbf{x} \ge \mathbf{0}$$

with $\mathbf{b} \ge \mathbf{0}$. Indeed, the indices of the slack variables introduced in the transformation to equational form can serve as a feasible basis.

However, in general, finding any feasible solution of a linear program is equally as difficult as finding an optimal solution (see the remark in Section 1.3). But computing the initial feasible basis can be done by the simplex method itself, if we apply it to a suitable auxiliary problem.

Let us consider the linear program in equational form

$$\begin{array}{ll} \text{maximize} & x_1 + 2x_2 \\ \text{subject to} & x_1 + 3x_2 + x_3 = 4 \\ & \qquad 2x_2 + x_3 = 2 \\ & x_1, x_2, x_3 \geq 0. \end{array}$$

Let us try to produce a feasible solution starting with $(x_1, x_2, x_3) = (0, 0, 0)$. This vector is nonnegative, but of course it is not feasible, since it does not satisfy the equations of the linear program. We introduce auxiliary variables x_4 and x_5 as "corrections" of infeasibility: $x_4 = 4 - x_1 - 3x_2 - x_3$ expresses by how much the original variables x_1, x_2, x_3 fail to satisfy the first equation, and $x_5 = 2 - 2x_2 - x_3$ plays a similar role for the second equation. If we managed to find nonnegative values of x_1, x_2, x_3 for which both of these corrections come out as zeros, we would have a feasible solution of the considered linear program.

The task of finding nonnegative x_1, x_2, x_3 with zero corrections can be captured by a linear program:

$$\begin{array}{ll} \text{Maximize} & \qquad\qquad\qquad - x_4 - x_5 \\ \text{subject to} & x_1 + 3x_2 + x_3 + x_4 \qquad = 4 \\ & \qquad 2x_2 + x_3 \qquad + x_5 = 2 \\ & x_1, x_2, \ldots, x_5 \geq 0. \end{array}$$

The optimal value of the objective function $-x_4 - x_5$ is 0 exactly if there exist values of x_1, x_2, x_3 with zero corrections, i.e., a feasible solution of the original linear program.

This is the right auxiliary linear program. The variables x_4 and x_5 form a feasible basis, with the basic feasible solution $(0, 0, 0, 4, 2)$. (Here we use that the right-hand sides, 4 and 2, are nonnegative, but since we deal with *equations*, this can always be achieved by sign changes.) Once we express the objective function using the nonbasic variables, that is, in the form $z = -6 + x_1 + 5x_2 + 2x_3$, we can start the simplex method on the auxiliary linear program.

The auxiliary linear program is surely bounded, since the objective function cannot be positive. The simplex method thus computes a basic feasible solution that is optimal.

As training the reader can check that if we let x_1 enter the basis in the first pivot step and x_3 in the second, the final simplex tableau comes out as

$$\begin{array}{rcrcrcrcr} x_1 & = & 2 & - & x_2 & - & x_4 & + & x_5 \\ x_3 & = & 2 & - & 2x_2 & & & - & x_5 \\ \hline z & = & & & & - & x_4 & - & x_5. \end{array}$$

The corresponding optimal solution $(2, 0, 2, 0, 0)$ yields a basic feasible solution of the original linear program: $(x_1, x_2, x_3) = (2, 0, 2)$. The initial simplex tableau for the original linear program can even be obtained from the final tableau of the auxiliary linear program, by leaving out the columns of the auxiliary variables x_4 and x_5,[1] and by changing the objective function back to the original one, expressed in terms of the nonbasic variables:

$$\begin{array}{rcrcr} x_1 & = & 2 & - & x_2 \\ x_3 & = & 2 & - & 2x_2 \\ \hline z & = & 2 & + & x_2 \end{array}$$

Starting from this tableau, a single pivot step already reaches the optimum.

5.5 Simplex Tableaus in General

In this section and the next one we formulate in general, and mostly with proofs, what has previously been explained on examples.

Let us consider a general linear program in equational form

$$\text{maximize } \mathbf{c}^T\mathbf{x} \text{ subject to } A\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}.$$

The simplex method applied to it computes a sequence of simplex tableaus. Each of them corresponds to a feasible basis B and it determines a basic feasible solution, as we will soon verify. (Let us recall that a feasible basis is an m-element set $B \subseteq \{1, 2, \ldots, n\}$ such that the matrix A_B is nonsingular and the (unique) solution of the system $A_B\mathbf{x}_B = \mathbf{b}$ is nonnegative.)

Formally, we will define a simplex tableau as a certain system of linear equations of a special form, in which the basic variables and the variable z, representing the value of the objective function, stand on the left-hand side and they are expressed in terms of the nonbasic variables.

A **simplex tableau** $\mathcal{T}(B)$ determined by a feasible basis B is a system of $m+1$ linear equations in variables $x_1, x_2, \ldots, x_n$, and z that has the *same set of solutions* as the system $A\mathbf{x} = \mathbf{b}$, $z = \mathbf{c}^T\mathbf{x}$, and in matrix notation looks as follows:

$$\begin{array}{lcrcl} \mathbf{x}_B & = & \mathbf{p} & + & Q\,\mathbf{x}_N \\ \hline z & = & z_0 & + & \mathbf{r}^T\mathbf{x}_N \end{array}$$

where $\mathbf{x}_B$ is the vector of the basic variables, $N = \{1, 2, \ldots, n\} \setminus B$, $\mathbf{x}_N$ is the vector of nonbasic variables, $\mathbf{p} \in \mathbb{R}^m$, $\mathbf{r} \in \mathbb{R}^{n-m}$, Q is an $m \times (n-m)$ matrix, and $z_0 \in \mathbb{R}$.

[1] It may happen that some auxiliary variables are zero but still basic in the final tableau of the auxiliary program, and so they cannot simply be left out. Section 5.6 discusses this (easy) issue.

The basic feasible solution corresponding to this tableau can be read off immediately: It is obtained by substituting $\mathbf{x}_N = \mathbf{0}$; that is, we have $\mathbf{x}_B = \mathbf{p}$. From the feasibility of the basis B we see that $\mathbf{p} \geq \mathbf{0}$. The objective function for this basic feasible solution has value $z_0 + \mathbf{r}^T\mathbf{0} = z_0$.

The values of $\mathbf{p}, Q, \mathbf{r}, z_0$ can easily be expressed using B and $A, \mathbf{b}, \mathbf{c}$:

5.5.1 Lemma. *For each feasible basis B there exists exactly one simplex tableau, and it is given by*

$$Q = -A_B^{-1}A_N, \ \mathbf{p} = A_B^{-1}\mathbf{b}, \ z_0 = \mathbf{c}_B^T A_B^{-1}\mathbf{b}, \ \ \textit{and} \ \ \mathbf{r} = \mathbf{c}_N - (\mathbf{c}_B^T A_B^{-1} A_N)^T.$$

It is neither necessary nor very useful to remember these formulas; they are easily rederived if needed. The proof is not very exciting and we write it more concisely than other parts of the text and we leave some details to a diligent reader. We will proceed similarly with subsequent proofs of a similar kind.

Proof. First let us see how these formulas can be discovered: We rewrite the system $A\mathbf{x} = \mathbf{b}$ to $A_B\mathbf{x}_B = \mathbf{b} - A_N\mathbf{x}_N$, and we multiply it by the inverse matrix A_B^{-1} from the left (these transformations preserve the solution set), which leads to

$$\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N.$$

We substitute the right-hand side for $\mathbf{x}_B$ into the equation $z = \mathbf{c}^T\mathbf{x} = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N$, and we obtain

$$\begin{aligned} z &= \mathbf{c}_B^T(A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N) + \mathbf{c}_N^T\mathbf{x}_N \\ &= \mathbf{c}_B^T A_B^{-1}\mathbf{b} + (\mathbf{c}_N^T - c_B^T A_B^{-1}A_N)\mathbf{x}_N. \end{aligned}$$

Thus the formulas in the lemma do yield a simplex tableau, and it remains to verify the uniqueness.

Let $\mathbf{p}, Q, \mathbf{r}, z_0$ determine a simplex tableau for a feasible basis B, and let $\mathbf{p}', Q', \mathbf{r}', z_0'$ do as well. Since each choice of $\mathbf{x}_N$ determines $\mathbf{x}_B$ uniquely, the equality $\mathbf{p} + Q\mathbf{x}_N = \mathbf{p}' + Q'\mathbf{x}_N$ has to hold for all $\mathbf{x}_N \in \mathbb{R}^{n-m}$. The choice $\mathbf{x}_N = \mathbf{0}$ gives $\mathbf{p} = \mathbf{p}'$, and if we substitute the unit vectors $\mathbf{e}_j$ of the standard basis for $\mathbf{x}_N$ one by one, we also get $Q = Q'$. The equalities $z_0 = z_0'$ and $\mathbf{r} = \mathbf{r}'$ are proved similarly. □

5.6 The Simplex Method in General

Optimality. Exactly as in the concrete example in Section 5.1, we have the following criterion of optimality of a simplex tableau:

If $\mathcal{T}(B)$ is a simplex tableau such that the coefficients of the nonbasic variables are nonpositive in the last row, i.e., if

$$\mathbf{r} \leq \mathbf{0},$$

then the corresponding basic feasible solution is *optimal.*

Indeed, the basic feasible solution corresponding to such a tableau has the objective function equal to z_0, while for any other feasible solution $\tilde{\mathbf{x}}$ we have $\tilde{\mathbf{x}}_N \geq 0$ and $\mathbf{c}^T\tilde{\mathbf{x}} = z_0 + \mathbf{r}^T\tilde{\mathbf{x}}_N \leq z_0$.

A pivot step: who enters and who leaves. In each step of the simplex method we go from an "old" basis B and simplex tableau $\mathcal{T}(B)$ to a "new" basis B' and the corresponding simplex tableau $\mathcal{T}(B')$. A nonbasic variable x_v enters the basis and a basic variable x_u leaves the basis,[2] and hence $B' = (B \setminus \{u\}) \cup \{v\}$.

We always select the entering variable x_v first.

A nonbasic variable may enter the basis if and only if its coefficient in the last row of the simplex tableau is *positive.*

Only incrementing such nonbasic variables increases the value of the objective function.

Usually there are several positive coefficients in the last row, and hence several possible choices of the entering variable. For the time being the reader may think of this choice as arbitrary. We will discuss ways of selecting one of these possibilities in Section 5.7.

Once we decide that the entering variable is some x_v, it remains to pick the leaving variable.

The leaving variable x_u has to be such that its nonnegativity, together with the corresponding equation in the simplex tableau having x_u on the left-hand side, limits the increment of the entering variable x_v most strictly.

Expressed by a formula, this condition might look complicated because of some double indices, but the idea is simple and we have already seen it in examples. Let us write $B = \{k_1, k_2, \ldots, k_m\}$, $k_1 < k_2 < \cdots < k_m$, and $N = \{\ell_1, \ell_2, \ldots, \ell_{n-m}\}$, $\ell_1 < \ell_2 < \cdots < \ell_{n-m}$. Then the ith equation of the simplex tableau has the form

$$x_{k_i} = p_i + \sum_{j=1}^{n-m} q_{ij} x_{\ell_j}.$$

[2] The letters u and v do not denote vectors here (the alphabet is not that long, after all).

We now want to write the index v of the chosen entering variable as $v = \ell_\beta$. In more detail, we define $\beta \in \{1, 2, \ldots, n-m\}$ as the index for which $v = \ell_\beta$. Similarly, the index u of the leaving variable (which hasn't been selected yet) will be written in the form $u = k_\alpha$.

Since all nonbasic variables x_{ℓ_j}, $j \neq \beta$, should remain zero, the nonnegativity condition $x_{k_i} \geq 0$ limits the possible values of the entering variable x_{ℓ_β} by the inequality $-q_{i\beta} x_{\ell_\beta} \leq p_i$. If $q_{i\beta} \geq 0$, then this inequality doesn't restrict the increase of x_{ℓ_β} in any way, while for $q_{i\beta} < 0$ it yields the restriction $x_{\ell_\beta} \leq -p_i/q_{i\beta}$.

The leaving variable x_{k_α} is thus always such that

$$q_{\alpha\beta} < 0 \quad \text{and} \quad -\frac{p_\alpha}{q_{\alpha\beta}} = \min\left\{-\frac{p_i}{q_{i\beta}} : q_{i\beta} < 0,\ i = 1, 2, \ldots, m\right\}. \qquad (5.3)$$

That is, in the simplex tableau we consider only the rows in which the coefficient of x_v is negative. In such rows we divide by this coefficient the component of the vector $\mathbf{p}$, we change sign, and we seek a minimum among these ratios. If there is no row with a negative coefficient of x_v, i.e., the minimum of the right-hand side of equation (5.3) is over an empty set, then the linear program is unbounded and the computation finishes.

For a proof that the simplex method really goes through a sequence of feasible bases we need the following lemma.

5.6.1 Lemma. *If B is a feasible basis and $\mathcal{T}(B)$ is the corresponding simplex tableau, and if the entering variable x_v and the leaving variable x_u have been selected according to the criteria described above (and otherwise arbitrarily), then $B' = (B \setminus \{u\}) \cup \{v\}$ is again a feasible basis.*

If no x_u satisfies the criterion for a leaving variable, then the linear program is unbounded. For all $t \geq 0$ we obtain a feasible solution by substituting t for x_v and 0 for all other nonbasic variables, and the value of the objective function for these feasible solutions tends to infinity as $t \to \infty$.

The proof is one of those not essential for a basic understanding of the material.

Proof (sketch). We first need to verify that the matrix $A_{B'}$ is nonsingular. This holds exactly if $A_B^{-1} A_{B'}$ is nonsingular, since we assume nonsingularity of A_B. The matrix $A_{B'}$ agrees with A_B in $m-1$ columns corresponding to the basic variable indices $B \setminus \{u\}$. For the basic variable with index $k_i, i \neq \alpha$, we get the unit vector $\mathbf{e}_i$, in the corresponding column of $A_B^{-1} A_{B'}$.

The negative of the remaining column of the matrix $A_B^{-1} A_{B'}$ occurs in the simplex tableau $\mathcal{T}(B)$ as the column of the entering variable x_v, since $Q = -A_B^{-1} A_N$ by Lemma 5.5.1. There is a nonzero number $q_{\alpha\beta}$ in row α corresponding to the leaving variable x_u, since we have selected

x_u that way, and the other columns of $A_B^{-1}A_{B'}$ have 0 in that row. Hence the matrix is nonsingular as claimed.

Next, we need to check feasibility of the basis B'. Here we use the fact that the new basic feasible solution, that for B', can be written in terms of the old one, and the nonnegativity of its basic variables are exactly those conditions that are used for choosing the leaving variable.

In practically the same way one can show the part of the lemma dealing with unbounded linear programs. We omit further details. □

A geometric view. As we saw in Section 4.4, basic feasible solutions are vertices of the polyhedron of feasible solutions. It is not hard to verify that a pivot step of the simplex method corresponds to a move from one vertex to another along an edge of the polyhedron (where an edge is a 1-dimensional face, i.e., a segment connecting the considered vertices; see Section 4.4).

Degenerate pivot steps are an exception, where we stay at the same vertex and only the feasible basis changes. A vertex of an n-dimensional convex polyhedron is generally determined by n of the bounding hyperplanes (think of a 3-dimensional cube, say). Degeneracy can occur only if we have more than n of the bounding hyperplanes meeting at a vertex (this happens for the 3-dimensional regular octahedron, for example).

Organization of the computations. Whenever we find a new feasible basis as above, we could compute the new simplex tableau according to the formulas from Lemma 5.5.1. But this is never done since it is inefficient.

For *hand calculation* the new simplex tableau is computed from the old one. We have already illustrated one possible approach in the examples. We take the equation of the old tableau with the leaving variable x_u on the left, and in this equation we carry the entering variable x_v over to the left and x_u to the right. The modified equation becomes the equation for x_v in the new tableau. The right-hand side is then substituted for x_v into all of the other equations, including the one for z in the last row. This finishes the construction of the new tableau.

In computer implementations of the simplex method, the simplex tableau is typically not computed in full. Rather, only the basic components of the basic feasible solution, i.e., the vector $\mathbf{p} = A_B^{-1}\mathbf{b}$, and the matrix A_B^{-1} are maintained. The latter allows for a fast computation of other entries of the simplex tableau when they are needed. (Let us note that for the optimality test and for selecting the entering variable we need only the last row, and for selecting the leaving variable we need only $\mathbf{p}$ and the column of the entering variable.) With respect to efficiency and numerical accuracy, the explicit inverse A_B^{-1} is not the best choice, and in practice, it is often represented by an (approximate) LU-factorization of the matrix A_B, or by other devices that can easily be updated during a pivot step of the simplex method. Since an efficient implementation of the simplex method is not among our main concerns, we will not describe how these things are actually done.

This computational approach is called the *revised simplex method.* For m considerably smaller than n it is usually much more efficient than maintaining all of the simplex tableau. In particular, $O(m^2)$ arithmetic operations per pivot step are sufficient for maintaining an LU-factorization of A_B, as opposed to about mn operations required for maintaining the simplex tableau.

Computing an initial feasible basis. If the given linear program has no "obvious" feasible basis, we look for an initial feasible basis by the procedure indicated in Section 5.4. For a linear program in the usual equational form

$$\text{maximize } \mathbf{c}^T\mathbf{x} \text{ subject to } A\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}$$

we first arrange for $\mathbf{b} \geq \mathbf{0}$: We multiply the equations with $b_i < 0$ by -1. Then we introduce m new variables x_{n+1} through x_{n+m}, and we solve the auxiliary linear program

$$\begin{array}{ll} \text{maximize} & -(x_{n+1} + x_{n+2} + \cdots + x_{n+m}) \\ \text{subject to} & \bar{A}\,\overline{\mathbf{x}} = \mathbf{b} \\ & \overline{\mathbf{x}} \geq \mathbf{0}, \end{array}$$

where $\overline{\mathbf{x}} = (x_1, \ldots, x_{n+m})$ is the vector of all variables including the new ones, and $\bar{A} = (A \,|\, I_m)$ is obtained from A by appending the $m \times m$ identity matrix to the right. The original linear program is feasible if and only if every optimal solution of the auxiliary linear program satisfies $x_{n+1} = x_{n+2} = \cdots = x_{n+m} = 0$. Indeed, it is clear that an optimal solution of the auxiliary linear program with $x_{n+1} = x_{n+2} = \cdots = x_{n+m} = 0$ yields a feasible solution of the original linear program. Conversely, any feasible solution of the original linear program provides a feasible solution of the auxiliary linear program that has the objective function equal to 0 and is thus optimal.

The auxiliary linear program can be solved by the simplex method directly, since the new variables x_{n+1} through x_{n+m} constitute an initial feasible basis. In this way we obtain some optimal solution. If it doesn't satisfy $x_{n+1} = x_{n+2} = \cdots = x_{n+m} = 0$, we are done—the original linear program is infeasible.

Let us assume that the optimal solution of the auxiliary linear program has $x_{n+1} = x_{n+2} = \cdots = x_{n+m} = 0$. The simplex method always returns a basic feasible solution. If none of the new variables x_{n+1} through x_{n+m} are in the basis for the returned optimal solution, then such a basis is then a feasible basis for the original linear program, too, and it allows us to start the simplex method.

In some degenerate cases it may happen that the basis returned by the simplex method for the auxiliary linear program contains some of the variables $x_{n+1}, \ldots, x_{n+m}$, and such a basis cannot directly be used for the original linear program. But this is a cosmetic problem only: From the returned optimal solution one can get a feasible basis for the original linear program by simple linear algebra. Namely, the optimal solution has at most m nonzero components, and their columns in the matrix A are linearly independent. If these columns are fewer than m, we can add more linearly independent columns and thus get a basis; see the proof of Lemma 4.2.1.

5.7 Pivot Rules

A **pivot rule** is a rule for selecting the entering variable if there are several possibilities, which is usually the case. Sometimes there may also be more than one possibility for choosing the leaving variable, and some pivot rules specify this choice as well, but this part is typically not so important.

The number of pivot steps needed for solving a linear program depends substantially on the pivot rule. (See the example in Section 5.1.) The problem is, of course, that we do not know in advance which choices will be good in the long run.

Here we list some of the common pivot rules. By an "improving variable" we mean any nonbasic variable with a positive coefficient in the z-row of the simplex tableau, in other words, a candidate for the entering variable.

LARGEST COEFFICIENT. Choose an improving variable with the largest coefficient in the row of the objective function z. This is the original rule, suggested by Dantzig, that maximizes the improvement of z *per unit increase* of the entering variable.

LARGEST INCREASE. Choose an improving variable that leads to the largest *absolute* improvement in z. This rule is computationally more expensive than the LARGEST COEFFICIENT rule, but it locally maximizes the progress.

STEEPEST EDGE. Choose an improving variable whose entering into the basis moves the current basic feasible solution in a direction closest to the direction of the vector $\mathbf{c}$. Written by a formula, the ratio

$$\frac{\mathbf{c}^T(\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}})}{\|\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}}\|}$$

should be maximized, where $\mathbf{x}_{\text{old}}$ is the basic feasible solution for the current simplex tableau and $\mathbf{x}_{\text{new}}$ is the basic feasible solution for the tableau that would be obtained by entering the considered improving variable into the basis. (We recall that $\|\mathbf{v}\| = (v_1^2 + v_2^2 + \cdots + v_n^2)^{1/2} = \sqrt{\mathbf{v}^T\mathbf{v}}$ denotes the Euclidean length of the vector $\mathbf{v}$, and the expression $\mathbf{u}^T\mathbf{v}/(\|\mathbf{u}\| \cdot \|\mathbf{v}\|)$ is the cosine of the angle of the vectors $\mathbf{u}$ and $\mathbf{v}$.)

The Steepest edge rule is a champion among pivot rules in practice. According to extensive computational studies it is usually faster than all other pivot rules described here and many others. An efficient approximate implementation of this rule is discussed in the glossary under the heading "Devex."

Bland's rule. Choose the improving variable with the smallest index, and if there are several possibilities for the leaving variable, also take the one with the smallest index. Bland's rule is theoretically very significant since it prevents cycling, as we will discuss in Section 5.8.

Random edge. Select the entering variable uniformly at random among all improving variables. This is the simplest example of a *randomized pivot rule*, where the choice of the entering variable uses random numbers in some way. Randomized rules are also very important theoretically, since they lead to the current best provable bounds for the number of pivot steps of the simplex method.

5.8 The Struggle Against Cycling

As we have already mentioned, it may happen that for some linear programs the simplex method cycles (and theoretically this is the only possibility of how it may fail). Such a situation is encountered very rarely in practice, if at all, and thus many implementations simply ignore the possibility of cycling.

There are several ways that provably avoid cycling. One of them is the already mentioned Bland's rule: We prove below that the simplex method never cycles if Bland's rule is applied consistently. Unfortunately, regarding efficiency, Bland's rule is one of the slowest pivot rules and it is almost never used in practice.

Another possibility can be found in the literature under the heading *lexicographic rule*, and here we only sketch it.

Cycling can occur only for degenerate linear programs. Degeneracy may lead to ties in the choice of the leaving variable. The lexicographic method breaks these ties as follows. Suppose that we have a set S of row indices such that for all $\alpha \in S$,

$$q_{\alpha\beta} < 0 \text{ and } -\frac{p_\alpha}{q_{\alpha\beta}} = \min\left\{-\frac{p_i}{q_{i\beta}} : q_{i\beta} < 0,\, i = 1, 2, \ldots, m\right\}.$$

In other words, all indices in S are candidates for the leaving variable. We then choose the index $\alpha \in S$ for which the vector

$$\left(\frac{q_{\alpha 1}}{q_{\alpha\beta}}, \ldots, \frac{q_{\alpha(n-m)}}{q_{\alpha\beta}}\right)$$

is the smallest in the lexicographic ordering. (We recall that a vector $\mathbf{x} \in \mathbb{R}^k$ is **lexicographically smaller** than a vector $\mathbf{y} \in \mathbb{R}^k$ if $x_1 < y_1$, or if $x_1 = y_1$ and $x_2 < y_2$, etc., in general, if there is an index $j \leq k$ such that $x_1 = y_1, \ldots, x_{j-1} = y_{j-1}$ and $x_j < y_j$.) Since the matrix A has rank m, it can be checked that any two of those vectors indeed differ at some index, and so we can resolve ties between any set S of rows. The chosen row index determines the leaving variable.

It can be shown that under the lexicographic rule, cycling is impossible. In very degenerate cases the lexicographic rule can be quite costly, since it may have to compute many components of the aforementioned vectors before the ties can eventually be broken.

Geometrically, the lexicographic rule has the following interpretation. For linear programs in equational form, degeneracy means that the set F of solutions of the system $A\mathbf{x} = \mathbf{b}$ contains a point with more than $n - m$ zero components, and thus it is not in general position with respect to coordinate axes. The lexicographic rule has essentially the same effect as a well-chosen perturbation of the set F, achieved by changing the vector $\mathbf{b}$ a little. This brings F into "general position" and therefore resolves all ties, while the optimal solution changes only by very little. The lexicographic rule simulates the effects of a suitable "infinitesimal" perturbation.

Now we return to Bland's rule.

5.8.1 Theorem. *The simplex method with Bland's pivot rule (the entering variable is the one with the* smallest index *among the eligible variables, and similarly for the leaving variable) is always finite; i.e., cycling is impossible.*

This is a basic result in the theory of linear programming (the duality theorem is an easy consequence, for example). Unfortunately, the proof is somewhat demanding. Its plot is simple, though: Assuming that there is a cycle, we get a contradiction in the form of an auxiliary linear program that has an optimal solution and is unbounded at the same time.

Proof. We assume that there is a cycle, and we let the set F consist of the indices of all variables that enter (and therefore also leave) the basis at least once during the cycle. We call these the *fickle* variables. First we verify a general claim about cycling of the simplex method, valid for any pivot rule.

Claim. All bases encountered in the cycle yield the same basic feasible solution, and all the fickle variables are 0 *in it.*

Proof of the claim. Since the objective function never decreases, it has to stay constant along the cycle.

Let B be a feasible basis encountered along the cycle, let $N = \{1, 2, \ldots, n\} \setminus B$ as usual, and let $B' = (B \setminus \{u\}) \cup \{v\}$ be the next basis. The only one among the *nonbasic* variables that may possibly change value in the pivot step from B to B' is the entering variable x_v; all others remain nonbasic and thus 0. By the rule for selecting the entering variable, the coefficient of x_v in the z-row of the tableau $\mathcal{T}(B)$ (i.e., in the vector $\mathbf{r}$) is strictly positive. Since the objective function is given by $z = z_0 + \mathbf{r}^T \mathbf{x}_N$, we see that if x_v became strictly positive, the objective function would increase. Hence the basic feasible solutions corresponding to B and B', respectively, agree in all components in N. Since these components determine the remaining ones uniquely (Proposition 4.2.2), the basic feasible solution does not change at all.

Finally, since every fickle variable is nonbasic at least once during the cycle, it has to be 0 all the time. The claim is proved.

The first trick in the proof of Theorem 5.8.1 is to consider the *largest* index v in the set F. Let B be a basis in the cycle just before x_v enters, and B' another basis just before x_v leaves (and x_u enters, say). Let $\mathbf{p}, Q, \mathbf{r}, z_0$ be the parameters of the simplex tableau $\mathcal{T}(B)$, and let $\mathbf{p}', Q', \mathbf{r}', z_0'$ be the parameters of $\mathcal{T}(B')$. (We remark that neither B nor B' has to be determined uniquely.)

Next, we use Bland's rule to infer some properties of the tableaus $\mathcal{T}(B)$ and $\mathcal{T}(B')$. First we focus on the situation at B. As in Section 5.6, we write B and $N = \{1, 2, \ldots, n\} \setminus B$ as ordered sets: $B = \{k_1, k_2, \ldots, k_m\}$, $k_1 < k_2 < \cdots < k_m$, and $N = \{\ell_1, \ell_2, \ldots, \ell_{n-m}\}$, $\ell_1 < \ell_2 < \cdots < \ell_{n-m}$. Since we have chosen v as the largest index in F, and Bland's rule requires v to be the smallest index of a candidate for entering the basis, no other fickle variable is a candidate at this point. Thus all fickle variables except for x_v have nonpositive coefficients in the z-row of $\mathcal{T}(B)$. Expressed formally, if β is the index such that $v = \ell_\beta$, we have

$$r_\beta > 0 \text{ and } r_j \leq 0 \text{ for all } j \text{ such that } \ell_j \in F \setminus \{v\}. \tag{5.4}$$

Second, we consider the tableau $\mathcal{T}(B')$. We write $B' = \{k_1', k_2', \ldots, k_m'\}$, $N' = \{1, 2, \ldots, n\} \setminus B' = \{\ell_1', \ell_2', \ldots, \ell_{n-m}'\}$, we let α' be the index of the leaving variable x_v in B', i.e., the one with $k_{\alpha'}' = v$, and we let β' be the index of the entering variable x_u in N', i.e., the one with $\ell_{\beta'}' = u$. By the same logic as above, x_v is the *only* candidate for leaving the basis among all the basic fickle variables in $\mathcal{T}(B')$. Recalling the criterion (5.3) for leaving the basis, we get that $i = \alpha'$ is the only i with $k_i' \in F$ and $q_{i\beta'}' < 0$ that minimizes the ratio $-p_i'/q_{i\beta'}'$. Since $\mathbf{p}'$ specifies the values of the basic variables and all fickle variables remain 0 during the cycle, we have $p_i' = 0$ for all i with $k_i' \in F$. Consequently,

$$q_{\alpha'\beta'}' < 0 \text{ and } q_{i\beta'}' \geq 0 \text{ for all } i \text{ such that } k_i' \in F \setminus \{v\}. \tag{5.5}$$

The idea is now to construct an auxiliary linear program for which (5.4) proves that it has an optimal solution, while (5.5) shows that it is unbounded. This is a clear contradiction, which rules out the initial assumption, the existence of a cycle.

The auxiliary linear program is the following:

$$\begin{array}{ll} \text{Maximize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x}_{F\setminus\{v\}} \geq \mathbf{0} \\ & x_v \leq 0 \\ & \mathbf{x}_{N\setminus F} = \mathbf{0}. \end{array}$$

We want stress that here the variables $\mathbf{x}_{B\setminus F}$ may assume *any* signs.

Optimality of the auxiliary linear program. We let $\tilde{\mathbf{x}}$ be the basic feasible solution of our original linear program associated with the basis B. Since $\tilde{\mathbf{x}}_N = \mathbf{0}$ and $\tilde{\mathbf{x}}_F = \mathbf{0}$ (by the claim), $\tilde{\mathbf{x}}$ is feasible for the auxiliary program. Moreover, for every $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{b}$ the value of the objective function can be expressed as

$$\mathbf{c}^T\mathbf{x} = z = z_0 + \mathbf{r}^T\mathbf{x}_N.$$

For all feasible solutions $\mathbf{x}$ of the auxiliary linear program, we have

$$x_{\ell_j} \begin{cases} \geq 0 & \text{if } \ell_j \in F \setminus \{v\} \\ \leq 0 & \text{if } \ell_j = \ell_\beta = v, \end{cases}$$

and so (5.4) implies

$$r_j x_{\ell_j} \leq 0 \text{ for all } j \text{ such that } \ell_j \in F.$$

Together with $\mathbf{x}_{N\setminus F} = \mathbf{0}$, we get $\mathbf{r}^T\mathbf{x}_N \leq \mathbf{0}$, and hence $z \leq z_0$ for all feasible solutions of the auxiliary linear program. It follows that $\tilde{\mathbf{x}}$ is an optimal solution of the auxiliary linear program.

Unboundedness of the auxiliary linear program. By the claim at the beginning of the proof, $\tilde{\mathbf{x}}$ is also the basic feasible solution of our original linear program associated with the basis B'. For all solutions $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$ we have

$$\mathbf{x}_{B'} = \mathbf{p}' + Q'\mathbf{x}_{N'}. \tag{5.6}$$

Let us now change $\tilde{\mathbf{x}}_{N'}$, by letting $\tilde{x}_u$ grow from its current value 0 to some value $t > 0$. Using (5.6), this determines a new solution $\tilde{\mathbf{x}}(t)$ of $A\mathbf{x} = \mathbf{b}$; we will show that for all $t > 0$, this solution is feasible for the auxiliary problem, but that the objective function value $\mathbf{c}^T\tilde{\mathbf{x}}(t)$ tends to infinity as $t \to \infty$. Here are the details.

We set

$$\tilde{x}_{\ell'_j}(t) := \begin{cases} 0 & \text{if } \ell'_j \in N' \setminus u \\ t & \text{if } \ell'_j = \ell'_{\beta'} = u. \end{cases}$$

With $\tilde{x}_v = 0$ and $t > 0$, (5.6) and (5.5) together show that

$$\tilde{x}_{k'_i}(t) = \tilde{x}_{k'_i} + t q'_{i\beta'} \begin{cases} \geq 0 & \text{if } k'_i \in F \setminus \{v\} \\ < 0 & \text{if } k'_i = k'_{\alpha'} = v. \end{cases}$$

In particular, $\tilde{\mathbf{x}}(t)$ is again feasible for the auxiliary linear program.

Since the variable $x_u = x_{\ell'_{\beta'}}$ was a candidate for entering the basis B', we know that $r'_{\beta'} > 0$, and hence

$$\mathbf{c}^T\tilde{\mathbf{x}}(t) = z'_0 + \mathbf{r}'^T\tilde{\mathbf{x}}_{N'}(t) = z'_0 + tr'_{\beta'} \to \infty \quad \text{for } t \to \infty.$$

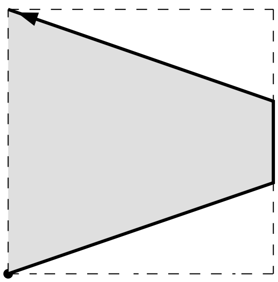
This means that the auxiliary linear program is unbounded. □

5.9 Efficiency of the Simplex Method

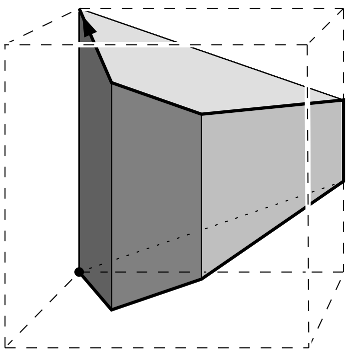
In practice, the simplex method performs very satisfactorily even for large linear programs. Computational experiments indicate that for linear programs in equational form with m equations it typically reaches an optimal solution in something between $2m$ and $3m$ pivot steps.

It was thus a great surprise when Klee and Minty constructed a linear program with n nonnegative variables and n inequalities for which the simplex method with Dantzig's original pivot rule (LARGEST COEFFICIENT) needs exponentially many pivot steps, namely $2^n - 1$!

The set of feasible solutions is an ingeniously deformed n-dimensional cube, called the *Klee–Minty cube*, constructed in such a way that the simplex method passes through all of its vertices. It is not hard to see that there is a deformed n-dimensional cube with an x_n-increasing path, say, through all vertices. Instead of a formal description we illustrate such a construction by pictures for dimensions 2 and 3:

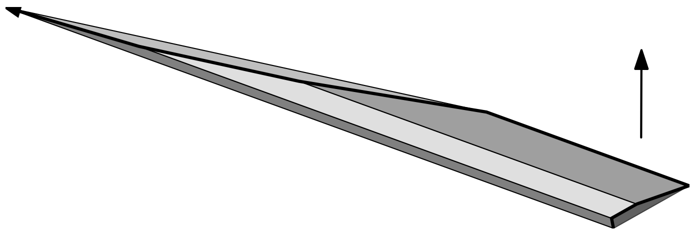


$n = 2$



$n = 3$

The deformed cube is inscribed in an ordinary cube in order to better convey the shape. With some pivot rules, the simplex method may traverse the path marked with a thick line. The particular deformed cube shown in the picture won't fool Dantzig's rule, for which the original example of Klee and Minty was constructed, though. A deformed cube that does fool Dantzig's rule looks more bizarre:



The direction of the objective function is drawn vertically. The corresponding linear program with $n = 3$ variables is simple:

$$\begin{array}{ll} \text{Maximize} & 9x_1 + 3x_2 + x_3 \\ \text{subject to} & x_1 \le 1 \\ & 6x_1 + x_2 \le 9 \\ & 18x_1 + 6x_2 + x_3 \le 81 \\ & x_1, x_2, x_3 \ge 0. \end{array}$$

It is instructive to see how, after the standard conversion to equational form, this linear program forces Dantzig's rule to go through all feasible bases.

Later on, very slow examples of a similar type were discovered for many other pivot rules, among them all the rules mentioned above. Many people have tried to design a pivot rule and prove that the number of pivot steps is always bounded by some polynomial function of m and n, but nobody has succeeded so far. The best known bound has been proved for the following simple randomized pivot rule: Choose a random ordering of the variables at the beginning of the computation (in other words, randomly permute the indices of the variables in the input linear program); then use Bland's rule for choosing the entering variable, and the lexicographic method for choosing the leaving variable. For every linear program with at most n variables and at most n constraints, the expected number of pivot steps is bounded by $e^{C\sqrt{n \ln n}}$, where C is a (not too large) constant. (Here the expectation means the arithmetic average over all possible orderings of the variables.) This bound is considerably better than 2^n, say, but much worse than a polynomial bound.

This algorithm was found independently and almost at the same time by Kalai and by Matoušek, Sharir, and Welzl. For a recent treatment in a somewhat broader context see

B. Gärtner and E. Welzl: Explicit and implicit enforcing—randomized optimization, in *Lectures of the Graduate Program*

Computational Discrete Mathematics, *Lecture Notes in Computer Science 2122* (2001), Springer, Berlin etc., pages 26–49.

A very good bound is not known even for the cleverest possible pivot rule, let us call it the clairvoyant's rule, that would always select the shortest possible sequence of pivot steps leading to an optimal solution. The *Hirsch conjecture*, one of the famous open problems in mathematics, claims that the clairvoyant's rule always reaches optimum in $O(n)$ pivot steps. But the best result proved so far gives only the bound of $n^{1+\ln n}$, due to

G. Kalai and D. Kleitman: Quasi-polynomial bounds for the diameter of graphs of polyhedra, *Bull. Amer Math. Soc.* 26(1992), 315–316.

This is better than $e^{C\sqrt{n \ln n}}$, but still worse than any polynomial function of n, and it doesn't provide a real pivot rule since nobody knows how to simulate clairvoyant's decisions by an efficient algorithm.

Here is an approach that looks promising and has been tried more recently, although without a clear success so far. One tries to modify the given linear program in such a way that polynomiality of a suitable pivot rule for the modified linear program would be easier to prove, and of course, so that an optimal solution of the original linear program could easily be derived from an optimal solution of the modified linear program.

In spite of the Klee–Minty cube and similar artificial examples, the simplex method is being used successfully. Remarkable theoretical results indicate that these willful examples are rare indeed. For instance, it is known that if a linear program in equational form is generated in a suitable (precisely defined) way at random, then the number of pivot steps is of order at most m^2 with high probability. More recent results, in the general framework of the so-called *smoothed complexity*, claim that if we take an arbitrary linear program and then we change its coefficients by small random amounts, then the simplex method with a certain pivot rule reaches the optimum of the resulting linear program by polynomially many steps with high probability (a concrete bound on the polynomial depends on a precise specification of the "small random amounts" of change). The first theorem of this kind is due to Spielman and Teng, and for recent progress see

R. Vershynin: Beyond Hirsch conjecture: Walks on random polytopes and the smoothed complexity of the simplex method, preprint, 2006.

An exact formulation of these results requires a number of rather technical notions that we do not want to introduce here, and so we omit it.

5.10 Summary

Let us review the simplex method once again.

Algorithm SIMPLEX METHOD

1. Convert the input linear program to equational form

$$\text{maximize } \mathbf{c}^T\mathbf{x} \text{ subject to } A\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}$$

   with n variables and m equations, where A has rank m (see Section 4.1).
2. If no feasible basis is available, arrange for $\mathbf{b} \geq \mathbf{0}$, and solve the following auxiliary linear program by the simplex method:

$$\begin{array}{ll} \text{Maximize} & -(x_{n+1} + x_{n+2} + \cdots + x_{n+m}) \\ \text{subject to} & \bar{A}\,\overline{\mathbf{x}} = \mathbf{b} \\ & \overline{\mathbf{x}} \geq \mathbf{0}, \end{array}$$

   where $x_{n+1}, \ldots, x_{n+m}$ are new variables, $\overline{\mathbf{x}} = (x_1, \ldots, x_{n+m})$, and $\bar{A} = (A \,|\, I_m)$. If the optimal value of the objective function comes out negative, the original linear program is infeasible; **stop**. Otherwise, the first n components of the optimal solution form a basic feasible solution of the original linear program.
3. For a feasible basis $B \subseteq \{1, 2, \ldots, n\}$ compute the simplex tableau $\mathcal{T}(B)$, of the form

$$\begin{array}{rcccc} \mathbf{x}_B & = & \mathbf{p} & + & Q\,\mathbf{x}_N \\ \hline z & = & z_0 & + & \mathbf{r}^T\mathbf{x}_N \end{array}$$

4. If $\mathbf{r} \leq \mathbf{0}$ in the current simplex tableau, return an optimal solution ($\mathbf{p}$ specifies the basic components, while the nonbasic components are 0); **stop**.
5. Otherwise, select an *entering variable* x_v whose coefficient in the vector $\mathbf{r}$ is positive. If there are several possibilities, use some pivot rule.
6. If the column of the entering variable x_v in the simplex tableau is non-negative, the linear program is unbounded; **stop**.
7. Otherwise, select a *leaving variable* x_u. Consider all rows of the simplex tableau where the coefficient of x_v is negative, and in each such row divide the component of the vector $\mathbf{p}$ by that coefficient and change sign. The row of the leaving variable is one in which this ratio is minimal. If there are several possibilities, decide by a pivot rule, or arbitrarily if the pivot rule doesn't specify how to break ties in this case.
8. Replace the current feasible basis B by the new feasible basis $(B \setminus \{u\}) \cup \{v\}$. Update the simplex tableau so that it corresponds to this new basis. Go to Step 4.

This is all we wanted to say about the simplex method here. May your pivot steps lead you straight to the optimum and never cycle!