The Simplex Algorithm

CSCI 4113/6101
Instructor: Norbert Zeh
September 22, 2025

This topic discusses the Simplex Algorithm, a classical and fairly simple algorithm for *solving* linear programs, that is, for finding optimal solutions of linear programs. As explained when discussing the complexity of linear and integer linear programming, the Simplex Algorithm does not run in polynomial time in the worst case, but it is remarkably fast in practice.

The Simplex Algorithm expects its input in standard form. This is not a restriction, as we have shown that every LP can be transformed into an equivalent LP in standard form. Throughout this discussion of the Simplex Algorithm, all LPs will be in standard form, so we simply refer to them as *linear programs* without explicitly stating the assumption that they are in standard form.

The Simplex Algorithm can be summarized as follows: Every LP in standard form has a unique special solution called its **basic solution**. If this solution is feasible, it is called a **basic feasible solution** (BFS). The first step of the Simplex Algorithm is to decide whether the given LP P is feasible, whether it has a feasible solution at all. If it doesn't, the algorithm terminates and reports that P is infeasible. If P has a feasible solution, then the Simplex Algorithm transforms it into an equivalent LP $P^{(0)}$ whose basic solution is feasible. More precisely, the algorithm tries to construct such an LP $P^{(0)}$ without first testing whether P is feasible; if this fails, then P is infeasible, and the algorithm reports this. Now recall the concept of equivalence of two LPs: they have the same set of feasible solutions and assign the same objective function value to any such solution. Thus, the BFS of $P^{(0)}$ is also a feasible solution of P. After constructing $P^{(0)}$, the Simplex Algorithm continues to transform this LP, producing a sequence of LPs $P^{(0)}, \ldots, P^{(t)}$ that are all equivalent to P and all have basic feasible solutions. The transformation of each LP $P^{(s)}$ into the next LP $P^{(s+1)}$, called **pivoting**, also ensures that the BFS of $P^{(s+1)}$ has an objective function value that is no less, ideally greater, than the objective function value of the BFS of $P^{(s)}$. Thus, the algorithm makes steady progress towards better and better solutions. Once the Simplex Algorithm can verify that the BFS of the current LP $P^{(t)}$ is optimal, something that is particularly easy to check for a BFS, it terminates and reports this solution. Since $P^{(t)}$ is equivalent to P, this solution must also be an optimal solution of P.

Our discussion of the Simplex Algorithm is organized as follows: We start with a discussion of **tableaux** in § 1, a tabular representation of LPs in standard form that the Simplex Algorithm uses. Sec. 2 introduces what the basic solution of an LP in standard form is. This is followed, in § 3, by a discussion of how the Simplex Algorithm recognizes that the current BFS is optimal or that the LP is unbounded. If the Simplex Algorithm cannot confirm (yet) that the current BFS is optimal or that the LP is unbounded, then it tries to construct an equivalent tableau with a better BFS. This is called pivoting and is discussed in § 4. Starting with a tableaux—that is, with an LP—whose basic solution is feasible, the Simplex Algorithm repeatedly applies this pivot operation to obtain equivalent tableaux whose basic solutions get better and better until it determines either that the current BFS is an optimal solution or that the LP is unbounded. This leaves the question of how we find the initial tableau whose basic solution is feasible. As we will discuss in § 5, the Simplex Algorithm does this by applying the basic approach for solving LPs discussed so far to an auxiliary LP derived from the LP we are trying to solve—how is that for bootstrapping?! This will complete the description of the Simplex Algorithm. With a complete description of the Simplex Algorithm in hand, § 6 will walk through an entire run of the Simplex Algorithm on an example LP, which will hopefully aid your understanding of the algorithm (and will allow you to appreciate that it is a really simple algorithm after all). A discussion of the Simplex Algorithm isn't complete without addressing **cycling**. Recall that the goal of pivoting is to obtain LPs with better and better basic feasible solutions. Sometimes though, a pivoting step fails to improve the solution. In fact, the solution does not change at all, only the basis of the tableau changes. This may cause the Simplex Algorithm to get stuck at the same BFS while cycling through a given set of bases indefinitely; it never terminates. Sec. 7 discusses **Bland's Rule** as one of many rules that can be used to choose which variable leaves and which variable enters the basis in each pivoting step, and we prove that this rule ensures that the Simplex Algorithm does not cycle. This immediately gives us an upper bound on the number of iterations that the Simplex Algorithm executes before producing an answer, which we prove in § 8.

1 TABLEAUX

We will use the following LP as a running example to illustrate how the Simplex Algorithm works:

$$
\begin{aligned}
&\text{Maximize } x_1 - 2x_2 + x_3 \\
\text{s.t. } y_1 \qquad\qquad & - \ \ x_1 - 2x_2 - 2x_3 = -3 \\
y_2 \qquad & -3x_1 - 2x_2 - \ \ x_3 = -5 \\
y_3 \quad & + \ \ x_1 \qquad\quad + 3x_3 = 10 \\
y_4 & + \ \ x_1 + \ \ x_2 + \ \ x_3 = \ \ 9 \\
x_1, x_2, x_3, y_1, y_2, & \, y_3, y_4 \geq \ \ 0
\end{aligned}
\tag{1}
$$

The basic variables are y_1, y_2, y_3, y_4. The non-basic variables are x_1, x_2, x_3. Recall from our discussion of standard form that the basic variables have coefficient 1 in exactly one of the constraints, and coefficient 0 in all other constraints and in the objective function. In particular, by writing the basic variables in the correct order at the beginning of the vector of variables in the matrix/vector notation of the LP—$Az = b$—the matrix A decomposes into an identity matrix and an arbitrary matrix whose columns correspond to the non-basic variables. In this concrete example, we have

$$
z = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad b = \begin{pmatrix} -3 \\ -5 \\ 10 \\ 9 \end{pmatrix}, \quad \text{and} \quad A = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & -1 & -2 & -2 \\ 0 & 1 & 0 & 0 & -3 & -2 & -1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 & 1 & -2 & 1 \end{array}\right).
$$

The objective function of the LP can similarly be written as $cz + d$, where in this case,

$$c = (0,0,0,0,1,-2,1) \quad \text{and} \quad d = 0.$$

Given that the LP is in standard form, we have the constraint that every variable should be non-negative. We simply remember this and don't write it down explicitly. The other pieces of information can be collected in a tabular form that shows much better which coefficients in A and c are associated with which variable. We write the variables across the top, basic variables before non-basic variables. Below this, we write down the matrix A and the vector c, thereby clearly showing the association of columns in A and c with their corresponding variables. We complete the table by adding a column on the left that contains b and $-d$. (We'll discuss shortly why we write $-d$, not d, in the bottom-left corner of the table.) For our example LP, this looks as follows:

	Basic				Non-basic		
	y_1	y_2	y_3	y_4	x_1	x_2	x_3
-3	1	0	0	0	-1	-2	-2
-5	0	1	0	0	-3	-2	-1
10	0	0	1	0	1	0	3
9	0	0	0	1	1	1	1
0	0	0	0	0	1	-2	1

This is called the **tableau** representation of this LP. Note two important properties of a tableau that follow immediately from the LP being in standard form:

- The portion of A in the basic columns is an identity matrix.
- The objective function coefficients of all basic variables, written below this identity matrx, are 0.

To highlight these two properties better, we omit all 0s from the tableau from here on (except in the bottom-left corner if d happens to be 0) and, for compactness, we omit the labelling of columns as basic and non-basic; we simply remember that the left group of variables are the basic ones, and the right group of variables are the non-basic ones. This gives us the final tableau representation we will use throughout the rest of these notes:

	y_1	y_2	y_3	y_4	x_1	x_2	x_3
-3	1				-1	-2	-2
-5		1			-3	-2	-1
10			1		1		3
9				1	1	1	1
0					1	-2	1

(2)

We will use transformations of tableaux to implement transformations of the LPs they represent. We are interested in two simple transformations: column swaps and basic row operations.

1.1 Column Swaps

A column swap does just that: it simply swaps two columns of the tableau. We will use this operation to swap a basic column and a non-basic column. For example, swapping the second basic column (corresponding to y_2) with the second non-basic column (corresponding to x_2) in (2) produces the following tableau:

	y_1	x_2	y_3	y_4	x_1	y_2	x_3
−3	1	−2			−1		−2
−5		−2			−3	1	−1
10			1		1		3
9		1		1	1		1
0		−2			1		1

(3)

Note that we swap the *entire* two columns: variables, matrix entries, and objective function coefficients. Thus, as long as we swap two columns associated with variables, the associations between variables and their coefficients in constraints and objective function do not change. Since we do not change the column storing b and $-d$ either, we have the following observation:

Observation 1. *For any tableau T_2 obtained from a tableau T_1 by swapping two columns in T_1 that correspond to variables, T_1 and T_2 represent the same linear program.*

Obs. 1 implies in particular that if T_2 is obtained from T_1 by a column swap operation, then the two LPs respresented by T_1 and T_2 are equivalent; after all, they are the same LP.

1.2 Basic Row Operations

The operation that actually transforms the LP represented by the tableau is called a basic row operation. This operation takes two of the rows of the tableau, say the ith and the hth row, and adds a multiple of the ith row to the hth row. For example, adding -1 times the second row of the tableau (3) to the first row produces the tableau

	y_1	x_2	y_3	y_4	x_1	y_2	x_3
2	1				2	−1	−1
−5		−2			−3	1	−1
10			1		1		3
9		1		1	1		1
0		−2			1		1

For this to produce an equivalent tableau, we need to impose three conditions when adding a multiple of the ith row to the hth row:

- Kind of obviously, neither the ith row nor the hth row should be the top row of the tableau holding the variable names. (What would it even mean to add a number to a name?)
- The row i cannot be the last row storing the objective function (but h can). That is, we are allowed to add a multiple of a row respresenting a constraint to another row representing a constraint, possibly the same row, or to the row representing the objective function, but we are not allowed to add a multiple of the objective function row to any other row, not even to the objective function

row itself.

- If $h = i$, then the coefficient with which we multiply the ith row before adding it to the hth row cannot be -1. This is necessary because adding -1 times the ith row to the ith row would completely zero out this row, that is, it would eliminate the constraint represented by this row.

Traditionally, one distinguishes between two types of row operations: multiplying a given row by a non-zero constant, and adding a multiple of one row to a *different* row. These are exactly the two types of operations the Simplex Algorithm will use (together with column swaps). In order to avoid repetitive proofs though, we use the observation here that multiplying the ith row by a coefficient α is the same as adding $\alpha - 1$ times the ith row to the ith row, so both types of row operations can be expressed as adding rows to rows, subject to the conditions above.

Throughout the remainder of this discussion of the Simplex Algorithm, we will reserve the term **solution** for any assignment $\hat{z}$ of values to the variables in the vector z such that $A\hat{z} = b$. Such a solution is **feasible** if it additionally satisfies the condition that $\hat{z} \geq 0$. This coincides exactly with the definition of a feasible solution of the LP, but we used the term "solution" before to refer to an *arbitrary* assignment of values to the variables in the LP—it didn't even have to satisfy $A\hat{z} = b$. In this discussion of the Simplex Algorithm, we call two LPs **equivalent** if they have the same set of solutions (and, therefore, also the same set of feasible solutions) and they assign the same objective function value to any solution.

Lemma 2. *If a tableau T_2 is obtained from a tableau T_1 via a basic row operation, then the two LPs represented by these two tableaux are equivalent.*

Proof. To establish equivalence of the two LPs, we need to prove that they have the same set of feasible solutions and that any solution has the same objective function value in both LPs. Let the two LPs represented by the two tableaux be

$$\begin{aligned} &\text{Maximize } cz + d \\ &\text{s.t. } b = Az \\ &z \geq 0 \end{aligned} \tag{4}$$

and

$$\begin{aligned} &\text{Maximize } c'z + d' \\ &\text{s.t. } b' = A'z \\ &z \geq 0. \end{aligned} \tag{5}$$

We distinguish two cases:

If the row operation adds α times the ith constraint row to the objective function row, then this clearly does not change the set of solutions, as $A' = A$ and $b' = b$ in this case. Thus, we need to prove that every solution $\hat{z}$ has the same objective function value in both LPs. We have

$$\begin{aligned} c'_j &= c_j + \alpha \cdot a_{ij} \quad \forall j \in [n] \\ d' &= d - \alpha \cdot b_i. \end{aligned}$$

(This is why we write $-d$ instead of d in the bottom-left corner of the tableau, so that adding α times the ith constraint to the objective function row *subtracts* αb_i from d instead of adding it.) Since every solution $\hat{z}$ satisfies $A\hat{z} = b$, we have in particular that

$$\sum_{j=1}^{n} a_{ij}\hat{z}_j = b_i.$$

Therefore,

$$\begin{aligned}
\sum_{j=1}^{n} c'_j\hat{z}_j + d' &= \sum_{j=1}^{n}(c_j + \alpha a_{ij})\hat{z}_j + (d - \alpha b_i) \\
&= \sum_{j=1}^{n} c_j\hat{z}_j + d + \alpha\left(\sum_{i=1}^{n} a_{ij}\hat{z}_j - b_i\right) \\
&= \sum_{j=1}^{n} c_j\hat{z}_j + d,
\end{aligned}$$

that is, every solution $\hat{z}$ has the same objective function value in both LPs.

If the row operation adds α times the ith constraint to the hth constraint, then this does not change the objective function and thus does not change the objective function value of any solution $\hat{z}$. We need to prove that the two LPs have the same set of solutions, that $A'\hat{z} = b'$ if and only if $A\hat{z} = b$.

First assume that $A\hat{z} = b$. Then

$$\sum_{j=1}^{n} a_{kj}\hat{z}_j = b_k \quad \forall k \in [m]. \tag{6}$$

If $k \neq h$, this implies that

$$\sum_{j=1}^{n} a'_{kj}\hat{z}_j = b'_k,$$

because $a'_{kj} = a_{kj}$, for all $j \in [n]$, and $b'_k = b_k$ in this case.

For $k = h$, we have

$$\begin{aligned}
\sum_{j=1}^{n} a'_{kj}\hat{z}_j &= \sum_{j=1}^{n}(a_{hj} + \alpha a_{ij})\hat{z}_j \\
&= \sum_{j=1}^{n} a_{hj}\hat{z}_j + \alpha\sum_{j=1}^{n} a_{ij}\hat{z}_j \\
&= b_h + \alpha b_i \\
&= b'_h,
\end{aligned}$$

by (6).

Now assume that $A'\hat{z} = b'$. Then

$$\sum_{j=1}^{n} a'_{kj}\hat{z}_j = b'_k \quad \forall k \in [m]. \tag{7}$$

If $k \neq h$, this implies that

$$\sum_{j=1}^{n} a_{kj}\hat{z}_j = b_k, \tag{8}$$

because $a'_{kj} = a_{kj}$, for all $j \in [n]$, and $b'_k = b_k$ in this case.

For $k = h$, we distinguish whether $h = i$ or $h \neq i$. If $h = i$, then $a'_{hj} = (1+\alpha)a_{hj}$, for all $j \in [n]$, $b'_h = (1+\alpha)b_h$, and $\alpha \neq -1$. Thus, $1+\alpha \neq 0$ and

$$\begin{aligned}
\sum_{j=1}^{n} a'_{hj}\hat{z}_j = b'_h \quad &\Longrightarrow \quad (1+\alpha)\sum_{j=1}^{n} a_{hj}\hat{z}_j = (1+\alpha)b_h \\
&\Longrightarrow \qquad\qquad \sum_{j=1}^{n} a_{hj}\hat{z}_j = b_h.
\end{aligned}$$

If $h \neq i$, then

$$\begin{aligned}
\sum_{j=1}^{n} a_{hj}\hat{z}_j + \alpha \sum_{j=1} a_{ij}\hat{z}_j &= \sum_{j=1}^{n} a'_{hj}\hat{z}_j \\
&= b'_h \\
&= b_h + \alpha b_i
\end{aligned} \tag{9}$$

and, since $i \neq h$ and by (8),

$$\sum_{j=1}^{n} a_{ij}\hat{z}_j = b_i.$$

The latter implies that

$$\alpha \sum_{j=1}^{n} a_{ij}\hat{z}_j = \alpha b_i.$$

Together with (9), this shows that

$$\sum_{j=1}^{n} a_{hj}\hat{z}_j = b_h. \qquad \square$$

2 Basic Solutions and Basic Feasible Solutions

Let $z_{j_1}, \ldots, z_{j_m}$ be the basic variables of the current tableau, from left to right. Then the following is one possible solution of the LP represented by the tableau:

$$\begin{aligned}
\hat{z}_{j_i} &= b_i \quad && \forall i \in [m] \\
\hat{z}_j &= 0 \quad && \forall j \in [n] \setminus \{j_1, \ldots, j_m\}
\end{aligned}$$

In words, this solution sets all non-basic variables to 0 and assigns to each basic variable z_{j_i} the constant b_i of the ith constraint. Since z_{j_i} is the only basic variable with a non-zero coefficient in the ith constraint, and this coefficient is 1, we have

$$\sum_{j=1}^{m} a_{ij}\hat{z}_j = \hat{z}_{j_i} + \sum_{j \in [m] \setminus \{j_1,\ldots,j_m\}} a_{ij}\hat{z}_j = \hat{z}_{j_i} = b_i,$$

that is, the ith constraint is satisfied. Since this is true for all $i \in [m]$, $\hat{z}$ is indeed a solution of the LP. We call this the **basic solution** of the tableau. As an example, the basic solution of the tableau (2) is:

$$\begin{array}{llll} y_1 = -3 & y_2 = -5 & y_3 = 10 & y_4 = 9 \\ x_1 = 0 & x_2 = 0 & x_3 = 0 & \end{array}$$

This is not a feasible solution because $y_1, y_2 < 0$. If the basic solution is feasible, that is, if it assigns non-negative values to all variables. then it as called the **basic feasible solution** (BFS) of the tableau. If the basic solution is not feasible, then the tableau does not have a BFS, because the basic solution of the tableau is fully determined by the tableau. The LP may have a a feasible solution in this case, but it is not the basic solution of the tableau.

Observe that all non-basic variables have value 0 in the basic solution, and all basic variables have objective function coefficient 0. Thus, every variable contributes 0 to the objective function value of the basic solution. Therefore, if the objective function is $cz + d$, the objective function value of the basic solution is simply d, the negation of the value in the bottom-left corner of the tableau.

3 Recognizing Optimal Solutions and Unbounded LPs

The Simplex Algorithm should stop if the BFS of the current tableau is an optimal solution or if the algorithm determines that the LP is infeasible. In the former case, we have found the solution we are looking for, so the algorithm reports this solution. In the latter case, there is no point in looking for an optimal solution because for every feasible solution, there exists a better solution. Therefore, the Simplex Algorithm reports that the LP is unbounded in this case. In this section, we discuss how the Simplex Algorithm detects that the BFS of the current tableau is optimal or that the LP is unbounded.

Proposition 3. *If all non-basic variables have non-positive objective function coefficients and the basic solution of the tableau is feasible, then this solution is an optimal solution of the LP.*

Proof. We assume explicitly that the basic solution $\hat{z}$ of the tableau is a BFS, so if it is not an optimal solution, then there exists another feasible solution $\tilde{z}$ with

$$\sum_{j=1}^{n} c_j \tilde{z}_j > \sum_{j=1}^{n} c_j \hat{z}_j.$$

This implies that there must exist an index j such that $c_j \tilde{z}_j > c_j \hat{z}_j$ and, therefore, $\tilde{z}_j \neq \hat{z}_j$. Since all basic variables have objective function coefficient 0, z_j must be a non-basic variable. Since $\hat{z}$ is the BFS of the tableau, this implies that $\hat{z}_j = 0$ and, therefore, $c_j \hat{z}_j = 0$ and $c_j \tilde{z}_j > 0$. Since $\tilde{z}$ is a feasible solution, we have $\tilde{z}_j \geq 0$. Therefore, since $\tilde{z}_j \neq \hat{z}_j$, we have $\tilde{z}_j > 0$. Since we just observed that $c_j \tilde{z}_j > 0$, this implies that $c_j > 0$.

We have shown that if the BFS is not an optimal solution, then there exists an objective function coefficient $c_j > 0$. The proposition states the contrapositive. □

The next claim we made was that if there exists a non-basic variable with a positive objective function coefficient and with onle non-positive coefficients in all constraints, then the LP is unbounded. The following proposition proves this:

Proposition 4. *If the basic solution of the tableau is feasible and there exists a non-basic variable z_h with $c_h > 0$ and $a_{ih} \le 0$, for all $i \in [m]$, then the LP is unbounded.*

Proof. It suffices to prove that the following solution $\hat{z}$ is a feasible solution of the LP, for any $\Delta > 0$:

$$\begin{aligned} \hat{z}_h &= \Delta \\ \hat{z}_{j_i} &= b_i - a_{ih}\Delta & \forall i \in [m] \\ \hat{z}_j &= 0 & \forall j \in [n] \setminus \{h, j_1, \dots, j_m\}. \end{aligned}$$

This solution has objective function value

$$\sum_{j=1}^{n} c_j \hat{z}_j + d = c_h \Delta + d$$

because all basic variables have objective function coefficient 0 and every non-basic variable z_j with $j \ne h$ satisfies $\hat{z}_j = 0$. Thus, we can choose Δ arbitrarily large to achieve an arbitrarily large objective function value: the LP in unbounded.

To show that $\hat{z}$ is a feasible solution, we need to prove that it satisfies all equality constraints, and that $\hat{z}_j \ge 0$, for all $j \in [n]$.

First, consider an arbitrary equality constraint:

$$\sum_{j=1}^{n} a_{ij} z_j = b_i.$$

We have $a_{ij_i} = 1$. For all $i' \ne i$, we have $a_{ij_{i'}} = 0$. For all $j \ne h$, we have $\hat{z}_j = 0$. Thus, this constraint is satisfied if

$$b_i = \hat{z}_{j_i} + a_{ih}\hat{z}_h = \hat{z}_{j_i} + a_{ih}\Delta.$$

Since $\hat{z}_{j_i} = b_i - a_{ih}\Delta$, this is the case.

It remains to prove that all variables have non-negative values. Since $\hat{z}_h = \Delta \ge 0$ and every basic variable z_j with $j \ne h$ satisfies $\hat{z}_j = 0$, all non-basic variables have non-negative values. So let z_{j_i} be an arbitrary basic variable. Then $\hat{z}_{j_i} = b_i - a_{ih}\Delta$. Since $a_{ih} \le 0$ and $\Delta \ge 0$, we have $a_{ih}\Delta \le 0$ and, therefore, $\hat{z}_{j_i} \ge b_i$. Since the basic solution of the tableau is feasible, we have $b_i \ge 0$. Thus, $\hat{z}_{j_i} \ge 0$. □

4 Pivoting

As long as the current tableau does not satisfy the conditions of Props. 3 and 4, the Simplex Algorithm repeatedly applies **pivot operations**. Each such operation transforms the current tableau into a tableau whose BFS is no worse, ideally better, than the BFS of the current tableau. For this to work, the current tableau must have a BFS. Therefore, to illustrate pivoting, we need a tableau with a BFS. The following tableau has a BFS (because all the entries of the vector b in the leftmost column are non-negative):

$$
\begin{array}{|c|cccc|ccc|}
\hline
 & x_1 & x_2 & y_3 & y_4 & y_1 & x_3 & y_2 \\
\hline
1 & 1 & & & & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\
1 & & 1 & & & -\frac{3}{4} & \frac{5}{4} & \frac{1}{4} \\
9 & & & 1 & & -\frac{1}{2} & \frac{7}{2} & \frac{1}{2} \\
7 & & & & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\hline
1 & & & & & -2 & 4 & 1 \\
\hline
\end{array}
\tag{10}
$$

This tableau is equivalent to (2): you can verify that it can be obtained from (2) by column swaps that arrange the columns in the order shown and then performing row operations to restore the tableau to standard form. We will discuss later how we can find a tableau with a BFS for any input tableau, or to decide that no such tableau exists.

Given this tableau, our goal is to construct a new tableau whose BFS has a greater objective function value. Remember, the negation of the objective function value of the BFS is written in the bottom-left corner of the tableau. We are trying to decrease this value.

What we will actually be doing is to look for a better solution of the current tableau, and then we rearrange the tableau to ensure that the solution we have found is the BFS of the new tableau.

If we try to change the values of many variables at once in an attempt to obtain a better solution, then we may have to deal with complex interactions between these changes to variable values: increasing one variable my increase the objective function value, but then we may have to change the values of other variables to keep the solution feasible, which may decrease the objective function value. Whether this results in a net loss or gain depends on the coefficients of variables in the constraints and in the objective function. Better to focus on changing few variables at a time, or rather to change only variables with very simple interactions with each other. This is where the structure of a BFS helps tremendously. We start with a few important observations; organizing these will immediately lead us to the definition of the pivot operation:

- Basic variables have coefficient 0 in the objective function. Thus, we can increase or decrease their values without affecting the objective function value at all. In particular, if we aim to increase the objective function value, we need to change the values of non-basic variables.

- In the BFS, non-basic variables have value 0. Thus, since all variables are required to be non-negative in a feasible solution, if we change the value of a non-basic variable, we can only increase it.

- Individually, increasing the value of a non-basic variable increases the objective function value only if this non-basic variable has a positive objective function coefficient.

- If we increase the value of one non-basic variable, this generally makes the solution infeasible, so we need to adjust the values of other variables to return to satisfying all equality constraints. Adjusting the values of basic variables is by far the easiest way to do this, for two reasons: (1) As already observed, basic variables have coefficient 0 in the objective function, so changing their values does not negate the gains in the objective function value we have made by increasing the value of the non-basic variable we chose. (2) Every basic variable has a non-zero coefficient in exactly one constraint. Thus, we can adjust the basic variables one by one to restore the solution to satisfying each equality constraint in turn.

- We need to ensure though that the values of all basic variables remain non-negative. Thus, if increasing the value of the non-basic variable forces us to decrease the value of a basic variable accordingly, then we need to limit by how much we increase the non-basic variable to avoid making any of the basic variables negative.

This gives us almost the complete description of pivoting already: A pivot operation chooses a non-basic variable with a positive objective function value. Since we assume that the current tableau does not meet the conditions of Prop. 3, such a variable exists. The pivot operation has to decide by how much it can increase this variable, and adjust the values of basic variables to maintain satisfaction of the equality constraints, while keeping all variable values non-negative. We discuss this next. The final step of pivoting is to rearrange the tableau so the new solution we have constructed is the BFS of this new tableau.

Assume that we have chosen the non-basic variable z_h as the one whose value we want to increase. In (10), x_3 and y_2 have positive objective function coefficients, so we can choose either of them. Let's choose x_3.[1] Assume we increase z_h by some amount $\Delta > 0$. Since the value of z_h in the current BFS is 0, Δ *is* the new value of z_h after this update. By how much do we need to adjust the values of basic variables to maintain satisfaction of the equality constraints? Consider the ith constraint,

$$\sum_{j=1}^{i} a_{ij} z_j = b_i.$$

The value of the corresponding basic variable z_{j_i} in the current BFS is b_i. Here is the important part: if z_h is the only non-basic variable whose value we change, then only z_{j_i} and z_h make non-zero contributions to the left-hand side of this equation, because all basic variables apart from z_{j_i} have coefficient 0 in this constraint and and all non-basic variables (except z_h after increasing it) have value 0 in the BFS. This simplifies the constraint to the equality

$$z_{j_i} + a_{ih} z_h = b_i.$$

Since we assign the value Δ to z_h, the new value of z_{j_i} must satisfy

$$\begin{aligned} z_{j_i} + a_{ih}\Delta &= b_i \\ z_{j_i} &= b_i - a_{ih}\Delta. \end{aligned} \tag{11}$$

We need to choose Δ so that this new value of z_{j_i} is non-negative, that is, so that $b_i - a_{ih}\Delta$ is non-negative. Since $b_i \geq 0$ (the current basic solution is feasible) and $\Delta \geq 0$ (we do not decrease the value of z_h), this is true no matter how large we choose Δ if $a_{ih} \leq 0$. In this case, the ith constraint imposes no bound on by how much we can increase the value of z_h. If $a_{ih} > 0$, then $z_{j_i} \geq 0$ as long as

$$\Delta \leq \frac{b_i}{a_{ih}}.$$

[1] This may seem like the better choice because x_3 has the greater objective function coefficient than y_2, so an increase of x_3 leads to a greater increase of the objective function value than the same increase of y_2. However, the constraints may be such that y_2 can increase by a much greater amount than x_3 before some basic variable becomes negative. Thus, choosing the variable with the greater objective function coefficient doesn't always ensure that we make rapid progress towards an optimal solution. In general, it is very difficult to develop rules that ensure that we make progress towards the optimal solution quickly.

Since this condition must be satisfied by every constraint where z_h's coefficient is positive, we obtain that the maximum value Δ by which we can increase z_h is

$$\Delta = \min\left\{\frac{b_i}{a_{ih}} \,\middle|\, i \in [m], a_{ih} > 0\right\}. \tag{12}$$

Note that this quantity is well-defined because the current tableau does not satisfy the conditions of Prop. 4, so there exists at least one index $i \in [m]$ with $a_{ih} > 0$.

We restore satisfaction of all equality constraints by updating all basic variables according to (11). This gives the solution

$$\begin{aligned}
\hat{z}_h &= \Delta \\
\hat{z}_{j_i} &= b_i - a_{ih}\Delta \quad && \forall i \in [m] \\
\hat{z}_j &= 0 && \forall j \in [n] \setminus \{h, j_1, \dots, j_m\}.
\end{aligned} \tag{13}$$

In our example tableau (10), given that we chose $z_h = x_3$, we have

$$\Delta = \min\left\{\frac{1}{5/4}, \frac{9}{7/2}, \frac{7}{1/4}\right\} = \frac{4}{5}.$$

Accordingly, we obtain the solution:

$$\begin{aligned}
x_1 &= 1 + \frac{1}{2} \cdot \frac{4}{5} = \frac{7}{5} \\
x_2 &= 1 - \frac{5}{4} \cdot \frac{4}{5} = 0 \\
x_3 &= \frac{4}{5} \\
y_1 &= 0 \\
y_2 &= 0 \\
y_3 &= 9 - \frac{7}{2} \cdot \frac{4}{5} = \frac{31}{5} \\
y_4 &= 7 - \frac{1}{4} \cdot \frac{4}{5} = \frac{34}{5}
\end{aligned} \tag{14}$$

This is easily verified to be a feasible solution of (10). The following two lemmas prove that (13) is always a feasible solution provided the current tableau has a BFS, and its objective function value is no less than the objective function value of the BFS.

Lemma 5. *If the basic solution of the tableau is feasible, then so is the solution $\hat{z}$ defined by* (12) *and* (13)*. Moreover, there exists an index $i \in [m]$ such that $\hat{z}_{j_i} = 0$.*

Proof. Prop. 4 proves that $\hat{z}$ satisfies all equality constraints of the LP, no matter how we choose Δ, and that $\hat{z}_{j_i} \geq 0$ if $a_{ih} \leq 0$. Thus, we only need to prove that $\hat{z}_{j_i} \geq 0$ also if $a_{ih} > 0$, and that there exists an index $i \in [m]$ such that $z_{j_i} = 0$.

Consider an arbitrary index $k \in [m]$ such that $a_{kh} > 0$. Then

$$\Delta = \min\left\{\frac{b_i}{a_{ih}} \,\middle|\, i \in [m], a_{ih} > 0\right\} \leq \frac{b_k}{a_{kh}}.$$

Therefore,

$$\hat{z}_{j_k} = b_k - a_{kh}\Delta \geq b_k - a_{kh}\frac{b_k}{a_{kh}} = 0.$$

If we choose the index $k \in [m]$ such that

$$\frac{b_k}{a_{kh}} = \min\left\{ \frac{b_i}{a_{ih}} \;\middle|\; i \in [m], a_{ih} > 0 \right\}$$

(such an index clearly exists), then

$$\hat{z}_{j_k} = b_k - a_{kh}\Delta = b_k - a_{kh}\frac{b_k}{a_{kh}} = 0.$$

This proves that there exists an index k with $\hat{z}_{j_k} = 0$. $\square$

LEMMA 6. *Let $\hat{z}$ be the solution defined by* (12) *and* (13). *If $c_h > 0$, then $\sum_{j=1}^{m} c_j\hat{z}_j + d \geq d$, that is, $\hat{z}$ has an objective function value no less than that of the BFS of the tableau.*

Proof. If z_j is a non-basic variable and $j \neq h$, then $\hat{z}_j = 0$. If z_j is a basic variable, then $c_j = 0$. Therefore,

$$\sum_{j=1}^{m} c_j\hat{z}_j + d = c_h\hat{z}_h + d.$$

Since $c_h > 0$, we have $c_h\hat{z}_h + d \geq d$ if and only if $\hat{z}_h \geq 0$. This condition is satisfied because, by Lem. 5, $\hat{z}$ is a feasible solution. $\square$

Why can't we guarantee that the new solution is strictly better than the current BFS? Don't we increase z_h and, thereby, increase the objective function value because $c_h > 0$? Not necessarily. We may have $\Delta = 0$, namely if there is an index $i \in [m]$ with $a_{ih} > 0$ and $b_i = 0$. As we will discuss in § 7, this may cause the Simplex Algorithm to cycle, and we need to be careful to prevent cycling. For now, we will ignore this issue.

Eq. (13) is a feasible solution, but in general, it is not a basic solution of the current tableau because, in general, $\hat{z}_h > 0$, and z_h is currently a non-basic variable. Lem. 5 states that there exists a basic variable z_{j_i} whose value is $\hat{z}_{j_i} = 0$. If we move z_h into the basis, and z_{j_i} out of the basis, then we restore the property that all non-basic variables are 0. This is easily achieved by swapping the columns of the tableau that correspond to z_h and z_{j_i}. In (10), given our choice of $z_h = x_3$, we have $z_{j_i} = x_2$. Swapping the two corresponding columns gives us the following tableau:

	x_1	x_3	y_3	y_4	y_1	x_2	y_2
1	1	$-\frac{1}{2}$			$\frac{1}{2}$		$-\frac{1}{2}$
1		$\frac{5}{4}$			$-\frac{3}{4}$	1	$\frac{1}{4}$
9		$\frac{7}{2}$	1		$-\frac{1}{2}$		$\frac{1}{2}$
7		$\frac{1}{4}$		1	$\frac{1}{4}$		$\frac{1}{4}$
1		4			-2		1

(15)

Since this tableau is obtained from (10) via a column swap, it is equivalent to (10), but we obviously have a new problem: the tableau is no longer in standard form, because z_h, now being the basic variable corresponding to the ith constraint, should have coefficient 1 in the ith constraint, and coefficient 0 in all other constraints, and in the objective function.

We can convert (15) into an equivalent tableau in standard form by applying basic row operations: We multiply the second row by $\frac{4}{5}$ and then add the resulting row multiplied by $\frac{1}{2}$, $-\frac{7}{2}$, $-\frac{1}{4}$, and -4 to the first, third, and fourth constraint rows, and to the objective function row, respectively. This gives the following tableau in standard form:

$$\begin{array}{|r|cccc|ccc|}
\hline
 & x_1 & x_3 & y_3 & y_4 & y_1 & x_2 & y_2 \\
\hline
\frac{7}{5} & 1 & & & & \frac{1}{5} & \frac{2}{5} & -\frac{2}{5} \\
\frac{4}{5} & & 1 & & & -\frac{3}{5} & \frac{4}{5} & \frac{1}{5} \\
\frac{31}{5} & & & 1 & & \frac{8}{5} & -\frac{14}{5} & -\frac{1}{5} \\
\frac{34}{5} & & & & 1 & \frac{2}{5} & -\frac{1}{5} & \frac{1}{5} \\
\hline
-\frac{11}{5} & & & & & \frac{2}{5} & -\frac{16}{5} & \frac{1}{5} \\
\hline
\end{array} \qquad (16)$$

Since basic row operations produce tableaux equivalent to the original tableau, by Lem. 2, this tableau is once again equivalent to (15) and, therefore, to (10).

Note that the basic solution of this tableau is feasible and is exactly the solution (14) we determined earlier. As the next proposition shows, this is not a coincidence: swapping the columns corresponding to z_h and z_{j_i} always produces a tableau with a BFS, and this BFS is exactly the solution in (13). Also note that the BFS of (15) has objective function value -1 (the negation of the value in the bottom-left corner), whereas the BFS of the new tableau as objective function value $\frac{11}{5}$, an improvement. Again, this is not a coincidence because we already proved, in Lem. 6, that the solution in (13) has an objective function value no less than that of the BFS at the beginning of the pivot operation.

Proposition 7. *The tableau produced by a pivot operation has the solution defined by* (12) *and* (13) *as its BFS.*

Proof. Since z_j is a non-basic variable of the updated tableau, for all $j \in [n] \setminus \{h, j_1, \dots, j_m\}$, each such variable has value $\hat{z}_j = 0$, as required.

The variable z_h is the basic variable corresponding to the ith constraint of the updated tableau. Thus, if b'_i is the entry in the leftmost column of the ith constraint, then $\hat{z}_h = b'_i$. The coefficient of z_h in the ith constraint of the original tableau is a_{ih}. To ensure that z_h has coefficient 1 in the ith constraint of the updated tableau, we divide the ith row by a_{ih}, that is,

$$\hat{z}_h = b'_i = \frac{b_i}{a_{ih}}.$$

Since the index i was chosen so that $\frac{b_i}{a_{ih}} = \Delta$, this shows that $\hat{z}_h = \Delta$, as required.

Since z_{j_i} is a non-basic variable of the updated tableau, it has value 0. The choice of i ensures that

$$b_i - a_{ih}\Delta = b_i - a_{ih}\frac{b_i}{a_{ih}} = 0,$$

so z_{j_i} has the value $b_i - a_{ih}\Delta$, as required.

For $i' \in [m] \setminus \{i\}$, we have $\hat{z}_{i'} = b'_{i'}$. Thus, $\hat{z}_{i'} = b_{i'} - a_{i'h}\Delta$ if $b'_{i'} = b_{i'} - a_{i'h}\Delta$. The coefficient of z_h in the (i')th constraint of the original tableau is $a_{i'h}$. To ensure that z_h has coefficient 0 in the updated tableau, we subtract $a_{i'h}$ times the ith constraint of the updated tableau from the (i')th constraint of the original tableau. Thus,

$$b'_{i'} = b_{i'} - a_{i'h} b'_i.$$

Since we already argued that $b'_i = \Delta$, this shows that $b'_{i'} = b_{i'} - a_{i'h}\Delta$. □

We arrived at the pivot operation from first principles, but this left the description of this operation a bit scattered. Here is the description of the entire operation in one place. A pivot operation performs the following steps:

- Choose a non-basic variable z_h with positive objective function coefficient. Since we apply the pivot operation only if the tableau does not satisfy the conditions of Prop. 3, such a variable exists.

- Calculate

  $$\Delta = \min \left\{ \frac{b_i}{a_{ih}} \;\middle|\; i \in [m], a_{ih} > 0 \right\}$$

  and choose one of the indices i such that $a_{ih} > 0$ and $\Delta = \frac{b_i}{a_{ih}}$. Since we apply the pivot operation only if the tableau does not satisfy the conditions of Prop. 4, Δ is well-defined, and such an index i exists.

- Swap the ith basic column corresponding to variable z_{j_i} with the non-basic column corresponding to z_h, thereby moving z_h into the basis, and z_{j_i} out of the basis.

- Restore the tableau to standard form using basic row operations.

Since the tableau this produces is obtained from the original tableau via a column swap and basic row operations, Obs. 1 and Lem. 2 show that the two tableaux are equivalent. Lems. 5 and 6 and Prop. 7 show that if the basic solution of the original tableau was feasible, then so is the basic solution of the tableau obtained after pivoting, and its objective function value is no less than the objective function value of the BFS of the original tableau.

5 Finding a Tableau with a BFS

The pivot operation we have discussed allows us to transform a tableau with a BFS into a new tableau whose BFS is no worse than the BFS of the current tableau. The Simplex Algorithm applies this operation repeatedly until it arrives at a BFS that is an optimal solution. In the absence of cycling, this happens in a finite number of iterations. Therefore, all we need to figure out to obtain a complete description of the Simplex Algorithm is how to determine whether the given LP is feasible and, if so, find an equivalent tableau $P^{(0)}$ whose BFS is feasible.

If the basic solution $\hat{z}$ of the tableau P given to the Simplex Algorithm as input is feasible, that is, if the vector b in the leftmost column of the tableau satisfies $b \geq 0$, then P clearly has a feasible solution and $\hat{z}$ is in fact a BFS, so we can set $P^{(0)} = P$. Therefore, assume that P's basic solution is infeasible.

To decide whether P has *any* feasible solution, we solve the auxiliary LP Q:

$$\begin{aligned} &\text{Maximize } -s \\ &\text{s.t. } b = Az - 1s \\ &z \geq 0 \\ &s \geq 0, \end{aligned}$$

where $1 = (1, \ldots, 1)^T$ is the m-element vector with all components equal to 1 and s is a new slack variable. In other words, Q is obtained from P by subtracting s from the right-hand side of each equality constraint and changing the objective function to $-s$. Q has the same basis B as P. If $\hat{z}$ is the basic solution of P, then $(\hat{z}, \hat{s})$ with $\hat{s} = 0$ is the basic solution of Q. To illustrate this, consider the tableau (3) again, reproduced here for easier reference:

	y_1	y_2	y_3	y_4	x_1	x_2	x_3
-3	1				-1	-2	-2
-5		1			-3	-2	-1
10			1		1		3
9				1	1	1	1
0					1	-2	1

As observed before, its basic solution $\hat{z}$ is not feasible because $\hat{y}_1, \hat{y}_2 < 0$. The tableau representing the corresponding auxiliary LP Q is obtained by zeroing out the objective function row and adding a column representing s on the right, with coefficient -1 in every row.

	y_1	y_2	y_3	y_4	x_1	x_2	x_3	s
-3	1				-1	-2	-2	-1
-5		1			-3	-2	-1	-1
10			1		1		3	-1
9				1	1	1	1	-1
0								-1

Let us defer the discussion of how to solve Q. How does solving it help us to decide whether P is feasible?

LEMMA 8. *P is feasible if and only if Q's optimal solution $(\tilde{z}, \tilde{s})$ has objective function value $-\tilde{s} = 0$.*

Proof. Given the constraint $s \geq 0$, Q has no feasible solution with objective function value greater than 0, so any solution with objective function value 0 must be optimal.

If P is feasible, then there exists a solution $\tilde{z}$ of P such that $b = A\tilde{z}$ and $\tilde{z} \geq 0$. Setting $\tilde{s} = 0$ gives $b = A\tilde{z} - 1\tilde{s}$ and $\tilde{s} \geq 0$. Thus, $(\tilde{z}, \tilde{s})$ is a feasible solution of Q with objective function value $-\tilde{s} = 0$ and is thus an optimal solution of Q.

Conversely, assume that Q has an optimal solution $(\tilde{z}, \tilde{s})$ with objective function value $-\tilde{s} = 0$. Then $\tilde{s} = 0$. Thus, $b = A\tilde{z} - 1\tilde{s} = A\tilde{z}$ and $\tilde{z} \geq 0$, that is, $\tilde{z}$ is a feasible solution of P. $\square$

5.1 From an Optimal Solution of Q to a Tableau with a BFS

By Lem. 8, we can answer that P is infeasible if Q's optimal solution has objective function value less than 0. So assume that Q has an optimal solution $(\tilde{z}, \tilde{s})$ with objective function value 0. As shown in the proof of Lem. 8, $\tilde{z}$ is a feasible solution of P. Thus, it suffices to convert P into an equivalent LP $P^{(0)}$ that has $\hat{z}^{(0)} = \tilde{z}$ as its BFS.

As we discuss below, we solve Q using the Simplex Algorithm. Thus, if Q has a feasible solution $(\tilde{z}, \tilde{s})$ with $\tilde{s} = 0$, then we find a tableau that has such a solution as its BFS and represents an LP Q' equivalent to Q. This tableau is obtained from the tableau representation of Q via column swaps and basic row operations. Next, we discuss how to obtain a tableau representation of $P^{(0)}$ from it:

- First we ensure that s is not a basic variable of the tableau. If it is, assume that s is the basic variable that corresponds to the ith constraint. Then we pick a non-basic variable z_h with $a_{ih} \neq 0$, and we apply a pivot operation to move z_h into the basis, and s out of the basis. As Lem. 9 below shows, such a non-basic variable exists. Lem. 10 below shows that this only results in a change of basis but leaves the BFS unchanged.

- Now consider the tableau obtained from the tableau representation of Q' by dropping the non-basic column corresponding to s and replacing the objective function coefficient of every variable z_j with its objective function coeffcient c_j in P. This tableau has $\tilde{z}$ as a BFS. Moreover, it can be obtained from the tableau representation of P by swapping columns to bring them in the desired order and then applying the same basic row operations that were used to obtain Q' from Q, except those basic row operations that modify the objective function row. Thus, the LP P' it represents is equivalent to P.

- P' is almost in standard form (becaus Q' is), except that there may be basic variables with non-zero objective function coefficients. We can use basic row operations (subtracting c_{j_i} times the ith constraint from the objective function row, for all $i \in [m]$) to ensure that all basic variables have objective function coefficient 0. The resulting tableau is in standard form and, since we did not change the basis or the constraint rows, has $\tilde{z}$ as a BFS. Since this tableau is obtained from the tableau representation of P' via basic row operations, and P' is equivalent to P, this tableau represents an LP $P^{(0)}$ equivalent to P, and its basic solution is feasible. Thus, this is the tableau we can use as the starting point for the search for an optimal solution of P via repeated pivoting.

Lemma 9. *Consider a tableau representing an LP Q' equivalent to Q, and assume that s is the basic variable corresponding to the hth constraint of this tableau. Then there exists a non-basic variable z_j with $a_{ij} \neq 0$.*

Proof. Since s is the basic variable corresponding to the hth constraint of Q', it has coefficient 1 in the hth constraint. Since Q' is obtained from Q via basic row operations, the ith constraint in Q' is a linear combination of the constraints in Q. In particular, we have

$$1 = a'_{hs} = \sum_{i=1}^{m} \alpha_i a_{is},$$

where a'_{hs} is the coefficient of s in the hth constraint of Q', and a_{is} is the coefficient of s in the ith constraint of Q. Since $a_{is} = -1$, for all $i \in [m]$, we therefore have

$$\sum_{i=1}^{m} \alpha_i = -1.$$

This implies that there exists at least one index i such that $\alpha_i < 0$. Let z_{j_i} be the basic variable that corresponds to the ith constraint of Q. Then $a'_{hj_i} = \alpha_i < 0$ because

$$a'_{hj_i} = \sum_{i'=1}^{m} \alpha'_i a_{i'j_i}$$

but $a_{i'j_i} = 1$ for $i' = i$, and $a_{i'j_i} = 0$ for $i' \neq i$. Since z_{j_i} is basic in Q, but s is not, we have $z_{j_i} \neq s$. Since s is the basic variable corresponding to the hth constraint of Q', every other basic variable of Q' has coeffcient 0 in the hth constraint of Q'. Thus, since $a'_{hj_i} < 0$ and $z_{j_i} \neq s$, j_i is non-basic in Q'. Thus, we can choose $z_j = z_{j_i}$ as the desired non-basic variable of Q'. □

Lemma 10. *Consider a tableau representing an LP equivalent to Q, and assume that s is the basic variable corresponding to the hth constraint of this tableau, and that z_j is a non-basic variable with $a_{ij} \neq 0$. Then pivoting to move s out of the basis, and z_j into the basis produces a tableau with the same BFS.*

Proof. The BFS $(\tilde{z}, \tilde{s})$ of the current tableau satisfies

$$\begin{aligned}
\tilde{s} &= b_h = 0 \\
\tilde{z}_{j_i} &= b_i & \forall i \in [m] \setminus \{h\} \\
\tilde{z}'_j &= 0 & \forall j' \in [n] \setminus \{j_1, \dots, j_m\}.
\end{aligned}$$

By Prop. 7, the BFS $(\hat{z}, \hat{s})$ of the tableau produced by the pivot operation satisfies

$$\begin{aligned}
\hat{s} &= 0 \\
\hat{z}_{j_i} &= b_i - a_{is}\Delta & \forall i \in [m] \setminus \{h\} \\
\hat{z}_j &= \Delta \\
\hat{z}_{j'} &= 0 & \forall j' \in [n] \setminus \{j, j_1, \dots, j_m\},
\end{aligned}$$

where $\Delta = \frac{b_h}{a_{hj}} = 0$. Thus,

$$\begin{aligned}
\hat{s} &= 0 && = \tilde{s} \\
\hat{z}_{j_i} &= b_i - a_{is}\Delta = b_i && = \tilde{z}_{j_i} & \forall i \in [m] \setminus \{h\} \\
\hat{z}_j &= \Delta = 0 && = \tilde{z}_j \\
\hat{z}_{j'} &= 0 && = \tilde{z}_{j'} & \forall j' \in [n] \setminus \{j, j_1, \dots, j_m\}.
\end{aligned}$$

□

5.2 How to Solve Q

Remember that we obtained the tableau representation of Q from the tableau representation of P by adding a non-basic column corresponding to s and changing the objective function to $-s$. This implies in particular that, if $\hat{z}$ is the basic solution of the tableau representation of P, then $(\hat{z}, 0)$ is the basic solution of the tableau representation of Q. Since we use the search for a feasible solution of P via Q only if $\hat{z}$ is infeasible, $(\hat{z}, 0)$ is infeasible in this case, too. If we want to use the Simplex Algorithm to solve Q, we need to transform Q into an equivalent LP $Q^{(0)}$ whose tableau representation has a BFS. If we don't want to get stuck in an infinite recursion, then we have to do this without constructing yet another auxiliary LP. Thankfully, the structure of Q is such that a single careful pivot operation produces this LP $Q^{(0)}$.

Consider the index h such that

$$b_h = \min\{b_i \mid i \in [m]\}.$$

Then we obtain $Q^{(0)}$ by using a pivot operation to move s into the basis and z_{j_h} out of the basis. Since $Q^{(0)}$ is obtained from Q via a pivot operation, that is, via a column swap and basic row operations, Q and $Q^{(0)}$ are equivalent. By the following lemma, $Q^{(0)}$ has a BFS.

Lemma 11. *The basic solution of $Q^{(0)}$ is feasible.*

Proof. Note that s has coefficient -1 in every constraint and that $b_h = \min\{b_i \mid i \in [m]\} < 0$ because the basic solution of Q is not feasible. In $Q^{(0)}$, s has coefficient 1, that is, the hth constraint is multiplied by -1. Thus, the basic solution $(\hat{z}, \hat{s})$ of $Q^{(0)}$ satisfies

$$\hat{s} = b_h^{(0)} = -b_h > 0.$$

Since s has coefficient 0 in the ith constraint of $Q^{(0)}$ and coefficient -1 in the ith constraint of Q, for $i \neq h$, the ith constraint of $Q^{(0)}$ is obtained by adding the hth constraint of $Q^{(0)}$ to the ith constraint of Q. Thus,

$$\hat{z}_{j_i} = b_i^{(0)} = b_i + b_h^{(0)} = b_i - b_h \geq 0,$$

where the last inequality follows because $b_h = \min\{b_i \mid i \in [m]\} \leq b_i$.

This shows that all basic variables of $Q^{(0)}$ have non-negative values in $(\hat{z}, \hat{s})$. Since all non-basic variables have value 0 in this solution, this shows that $(\hat{z}, \hat{s})$ is feasible. □

6 A Complete Example

The details of the Simplex Algorithm will become much clearer if we work through it using an example. We will use the LP (1) we already used as a running example throughout the discussion. For reference, here it is again:

$$\begin{aligned}
&\text{Maximize } x_1 - 2x_2 + x_3 \\
\text{s.t. } y_1 \quad &- \ \ x_1 - 2x_2 - 2x_3 = -3 \\
y_2 \quad &- 3x_1 - 2x_2 - \ \ x_3 = -5 \\
y_3 \quad &+ \ \ x_1 \qquad\quad + 3x_3 = 10 \\
y_4 &+ \ \ x_1 + \ \ x_2 + \ \ x_3 = \ \ 9 \\
x_1, x_2, x_3, &y_1, y_2, y_3, y_4 \geq \ \ 0
\end{aligned}$$

And here is its tableau representation:

	y_1	y_2	y_3	y_4	x_1	x_2	x_3
-3	1				-1	-2	-2
-5		1			-3	-2	-1
10			1		1		3
9				1	1	1	1
0					1	-2	1

(17)

The basic solution of this tableau is not feasible, so we need to use the technique from the previous section to find an equivalent tableau whose basic solution *is* feasible. We then use repeated pivoting to find an optimal solution.

6.1 Finding a Tableau with a BFS

As discussed in § 5, we find a tableau equivalent to (17) whose basic solution solution is feasible by solving the following auxiliary LP:

	y_1	y_2	y_3	y_4	x_1	x_2	x_3	s
-3	1				-1	-2	-2	-1
-5		1			-3	-2	-1	-1
10			1		1		3	-1
9				1	1	1	1	-1
0								-1

(18)

All we did here was to add the non-basic variable s to the LP, with coefficient -1 in every constraint, and we changed the objective function to $-s$.

The basic solution of this tableau is not feasible, so we need to apply the special pivoting step discussed in § 5.2 to transform it into an equivalent tableau with a BFS. We have $\min\{b_1, b_2, b_3, b_4\} = b_2 = -5$. Thus, to obtain an equivalent tableau whose basic solution is feasible, we perform a pivot step that puts s into the basis and removes y_2 from the basis. basis. First we reorder the columns of the tableau to reflect the new basis:

	y_1	s	y_3	y_4	x_1	x_2	x_3	y_2
-3	1	-1			-1	-2	-2	
-5		-1			-3	-2	-1	1
10		-1	1		1		3	
9		-1		1	1	1	1	
0		-1						

Next we need to ensure that s has coefficient 1 in the second constraint and coefficient 0 in all other constraints and in the objective function. To this end, we multiply the second constraint by -1, and then we add the result to every other row of the tableau because s's coefficient is -1 in every row of the tableau:

	y_1	s	y_3	y_4	x_1	x_2	x_3	y_2
2	1				2		−1	−1
5		1			3	2	1	−1
15			1		4	2	4	−1
14				1	4	3	2	−1
5					3	2	1	−1

(19)

Since $b \geq 0$ in this tableau, its BFS is feasible. is objective function value is −5 (the negation of the 5 in the bottom right corner). To find an optimal solution of this tableau, we use standard pivot operations now.

Since there are non-basic variables with positive objective function coefficients, we cannot guarantee yet that the current BFS is optimal. So we pick an arbitrary non-basic variable whose objective function coefficient function is positive; x_1 is a valid choice because its coefficient is 3. This is the non-basic variable that enters the basis in the pivot operation we are about to perform. To choose the basic variable that leaves the basis, we need to find a constraint in which x_1 has a positive coefficient a_{i,x_1} and which minimizes the value $\frac{b_i}{a_{i,x_1}}$.[2] We have

$$\frac{b_1}{a_{1,x_1}} = \frac{2}{2} = 1 \qquad \frac{b_2}{a_{2,x_1}} = \frac{5}{3} \qquad \frac{b_3}{a_{3,x_1}} = \frac{15}{4} \qquad \frac{b_4}{a_{4,x_1}} = \frac{14}{4}.$$

The smallest of these values is $\frac{b_1}{a_{1,x_1}} = 1$, so the first basic variable, y_1, is the one that needs to leave the basis.

Having identified the pair of variables that need to enter or leave the basis, we now implement the pivot operation as always, by first swapping the columns corresponding to these two variables,

	x_1	s	y_3	y_4	y_1	x_2	x_3	y_2
2	2				1		−1	−1
5	3	1				2	1	−1
15	4		1			2	4	−1
14	4			1		3	2	−1
5	3					2	1	−1

and the performing basic row operations to restore the tableau to standard form:

	x_1	s	y_3	y_4	y_1	x_2	x_3	y_2
1	1				$\frac{1}{2}$		$-\frac{1}{2}$	$-\frac{1}{2}$
2		1			$-\frac{3}{2}$	2	$\frac{5}{2}$	$\frac{1}{2}$
11			1		−2	2	6	1
10				1	−2	3	4	1
2					$-\frac{3}{2}$	2	$\frac{5}{2}$	$\frac{1}{2}$

[2]Given that the columns of the tableau keep being rearranged, I refer to the columns of the tableau not by index but by the variables they represent here. So a_{i,x_1} is the coefficient of x_1 in the ith constraint.

Specfically, we divided the first row by 2 and then subtracted this row multiplied by 3, 4, 4, and 3 from the remaining constraint rows and from the objective function row, respectively.

Note that the BFS of this LP, $x_1 = 1$, $s = 2$, $y_3 = 11$, $y_4 = 10$, and $y_1 = x_2 = x_3 = y_2 = 0$ has objective function value -2, which is an improvement over the previous solution's objective function value, -5.

We still have non-basic variables with positive objective function coefficients, so we need to continue pivoting. This time, we choose x_2 as the variable that should enter the basis. The variable x_2 has coefficients 2, 2, and 3 in the second, third, and fourth constraints, respectively. The coefficient in the first constraint is 0, so only s, y_3, and y_4 are valid candidates for basic variables that can leave the basis. Of the three values $\frac{2}{2} = 1$, $\frac{11}{2}$, and $\frac{10}{3}$, 1 is the smallest. Thus, s is the basic variable that should leave the basis. Again, we start by rearranging the tableau to reflect the new basis:

	x_1	x_2	y_3	y_4	y_1	s	x_3	y_2
1	1				$\frac{1}{2}$		$-\frac{1}{2}$	$-\frac{1}{2}$
2		2			$-\frac{3}{2}$	1	$\frac{5}{2}$	$\frac{1}{2}$
11		2	1		-2		6	1
10		3		1	-2		4	1
2		2			$-\frac{3}{2}$		$\frac{5}{2}$	$\frac{1}{2}$

To give x_2 a coefficient of 1 in the second constraint and coefficient 0 everywhere else, we divide the second constraint by 2 and then subtract the result multiplied by 2, 3, and 2 from the third and fourth constraint and from the objective function, respectively:

	x_1	x_2	y_3	y_4	y_1	s	x_3	y_2
1	1				$\frac{1}{2}$		$-\frac{1}{2}$	$-\frac{1}{2}$
1		1			$-\frac{3}{4}$	$\frac{1}{2}$	$\frac{5}{4}$	$\frac{1}{4}$
9			1		$-\frac{1}{2}$	-1	$\frac{7}{2}$	$\frac{1}{2}$
7				1	$\frac{1}{4}$	$-\frac{3}{2}$	$\frac{1}{4}$	$\frac{1}{4}$
0						-1		

(20)

Now, all non-basic variables have non-positive coefficients in the objective function. Thus, the BFS of the current tableau is an optimal solution *of the auxiliary tableau* (18)*!* We are still in the initialization phase of the Simplex Algorithm.

Since this optimal solution has objective function value 0 (as stated in the bottom-left corner of the tableau), our original LP (18) is feasible. Thus, we can convert (20) into a that is equivalent to (17) and has a BFS.

In this example, we do not need to remove s from the basis of (20) because it is not in the basis of this tableau. Thus, we start by removing the non-basic column corresponding to s from the tableau and restoring the objective function coefficients that all variables have in (17):

$$
\begin{array}{|c|cccc|ccc|}
\hline
 & x_1 & x_2 & y_3 & y_4 & y_1 & x_3 & y_2 \\
\hline
1 & 1 & & & & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\
1 & & 1 & & & -\frac{3}{4} & \frac{5}{4} & \frac{1}{4} \\
9 & & & 1 & & -\frac{1}{2} & \frac{7}{2} & \frac{1}{2} \\
7 & & & & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\hline
0 & 1 & -2 & & & & 1 & \\
\hline
\end{array}
\tag{21}
$$

We restore this tableau to standard form by subtracting the first constraint from the objective function and adding twice the second constraint to the objective function. This gives the following tableau:

$$
\begin{array}{|c|cccc|ccc|}
\hline
 & x_1 & x_2 & y_3 & y_4 & y_1 & x_3 & y_2 \\
\hline
1 & 1 & & & & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\
1 & & 1 & & & -\frac{3}{4} & \frac{5}{4} & \frac{1}{4} \\
9 & & & 1 & & -\frac{1}{2} & \frac{7}{2} & \frac{1}{2} \\
7 & & & & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\hline
1 & & & & & -2 & 4 & 1 \\
\hline
\end{array}
\tag{22}
$$

Since (21) can be obtained from (17) via column swaps and the same basis row operations that produced (20) from (18), and (22) is obtained from (21) via additional basic row operations, (22) is equivalent from (17). Its basic solution,

$$x_1 = 1 \quad x_2 = 1 \quad y_3 = 9 \quad y_4 = 7 \quad y_1 = x_3 = y_2 = 0,$$

is a BFS because all variables have non-negative values. We are now ready to continue pivoting to find an optimal solution of (22), which is also an optimal solution of (17).

6.2 Finding an Optimal Solution

First let us try to move x_3 into the basis. x_3 has positive coefficients in the second, third, and fourth constraints. The corresponding upper bounds imposed on x_3 by these constraints are

$$\frac{1}{5/4} = \frac{4}{5} \qquad \frac{9}{7/2} = \frac{18}{7} \qquad \frac{7}{1/4} = 28.$$

The tightest of these upper bounds is $\frac{4}{5}$. Thus, if we want to bring x_3 into the basis, the second basic variable, x_2, needs to leave the basis. Again, we start by rearranging the tableau,

	x_1	x_3	y_3	y_4	y_1	x_2	y_2
1	1	$-\frac{1}{2}$			$\frac{1}{2}$		$-\frac{1}{2}$
1		$\frac{5}{4}$			$-\frac{3}{4}$	1	$\frac{1}{4}$
9		$\frac{7}{2}$	1		$-\frac{1}{2}$		$\frac{1}{2}$
7		$\frac{1}{4}$		1	$\frac{1}{4}$		$\frac{1}{4}$
1		4			-2		1

and basic row operations restore this tableau to standard form:

	x_1	x_3	y_3	y_4	y_1	x_2	y_2
$\frac{7}{5}$	1				$\frac{1}{5}$	$\frac{2}{5}$	$-\frac{2}{5}$
$\frac{4}{5}$		1			$-\frac{3}{5}$	$\frac{4}{5}$	$\frac{1}{5}$
$\frac{31}{5}$			1		$\frac{8}{5}$	$-\frac{14}{5}$	$-\frac{1}{5}$
$\frac{34}{5}$				1	$\frac{2}{5}$	$-\frac{1}{5}$	$\frac{1}{5}$
$-\frac{11}{5}$					$\frac{2}{5}$	$-\frac{16}{5}$	$\frac{1}{5}$

Next, let's move y_1 into the basis because its objective function coefficient is $\frac{2}{5}$. y_1 has positive coefficients in the first, third, and fourth constraints. The corresponding upper bounds on the value of y_1 imposed by these constraints are

$$\frac{7/5}{1/5} = 7 \qquad \frac{31/5}{8/5} = \frac{31}{8} \qquad \frac{34/5}{2/5} = 17,$$

of which $\frac{31}{8}$ is the tightest. Thus, y_3 needs to leave the basis. We swap the columns corresponding to these variables to reflect the change of basis,

	x_1	x_3	y_1	y_4	y_3	x_2	y_2
$\frac{7}{5}$	1		$\frac{1}{5}$			$\frac{2}{5}$	$-\frac{2}{5}$
$\frac{4}{5}$		1	$-\frac{3}{5}$			$\frac{4}{5}$	$\frac{1}{5}$
$\frac{31}{5}$			$\frac{8}{5}$		1	$-\frac{14}{5}$	$-\frac{1}{5}$
$\frac{34}{5}$			$\frac{2}{5}$	1		$-\frac{1}{5}$	$\frac{1}{5}$
$-\frac{11}{5}$			$\frac{2}{5}$			$-\frac{16}{5}$	$\frac{1}{5}$

and basic row operations restore the tableau to standard form:

	x_1	x_3	y_1	y_4	y_3	x_2	y_2
$\frac{5}{8}$	1				$-\frac{1}{8}$	$\frac{3}{4}$	$-\frac{3}{8}$
$\frac{25}{8}$		1			$\frac{3}{8}$	$-\frac{1}{4}$	$\frac{1}{8}$
$\frac{31}{8}$			1		$\frac{5}{8}$	$-\frac{7}{4}$	$-\frac{1}{8}$
$\frac{21}{4}$				1	$-\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
$-\frac{15}{4}$					$-\frac{1}{4}$	$-\frac{5}{2}$	$\frac{1}{4}$

We still have a non-basic variable with positive coefficient in the objective function left: y_2. y_2 has positive coefficients in the second and fourth constraints. The upper bounds on the value of y_2 imposed by these constraints are

$$\frac{25/8}{1/8} = 25 \qquad \frac{21/4}{1/4} = 21,$$

of which 21 is the tighter one. Thus, y_4 needs to leave the basis if y_2 enters the basis. We swap the columns corresponding to these variables to reflect the change of basis,

	x_1	x_3	y_1	y_2	y_3	x_2	y_4
$\frac{5}{8}$	1			$-\frac{3}{8}$	$-\frac{1}{8}$	$\frac{3}{4}$	
$\frac{25}{8}$		1		$\frac{1}{8}$	$\frac{3}{8}$	$-\frac{1}{4}$	
$\frac{31}{8}$			1	$-\frac{1}{8}$	$\frac{5}{8}$	$-\frac{7}{4}$	
$\frac{21}{4}$				$\frac{1}{4}$	$-\frac{1}{4}$	$\frac{1}{2}$	1
$-\frac{15}{4}$				$\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{5}{2}$	

and basic row operations restore the tableau to standard form:

	x_1	x_3	y_1	y_2	y_3	x_2	y_4
$\frac{17}{2}$	1				$-\frac{1}{2}$	$\frac{3}{2}$	$\frac{3}{2}$
$\frac{1}{2}$		1			$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$
$\frac{13}{2}$			1		$\frac{1}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$
21				1	-1	2	4
-9						-3	-1

Finally, all non-basic variables have non-positive coefficients in the objective function. Thus, the current basic solution,

$$\begin{aligned} x_1 &= \frac{17}{2} & y_1 &= \frac{13}{2} \\ x_2 &= 0 & y_2 &= 21 \\ x_3 &= \frac{1}{2} & y_3 &= 0 \\ & & y_4 &= 0, \end{aligned}$$

is an optimal solution; its objective function value 9.

7 Bland's Anti-Cycling Rule*

As already mentioned briefly, the Simplex Algorithm may get stuck at a BFS that is a BFS for multiple tableaux, simply cycling through these tableaux without making progress towards tableaux with better BFSs. Constructing simple examples where this occurs is surprisingly tricky.[3] Your are welcome to Google "When does Simplex cycle" and work through the examples that come up if you have the energy for it.

[3] I didn't find any example online or in the literature that wasn't underconstrained in the sense that it didn't have more variables than constraints. I tried to construct my own example that wasn't underconstrained based on the geometric intuition about when cycling occurs, but I failed.

Here, we will simply accept that picking the non-basic variable that should enter the basis in a pivoting step arbitrarily from among those with positive objective function coefficients, and choosing the basic variable that should leave the basis arbitrarily from among those whose corresponding constraints are tightest, may cause the Simplex Algorithm to cycle. There are several rules one can use to choose the variables leaving and entering the basis in a pivot operation carefully (i.e., *not* arbitrarily), in a manner that prevents cycling. In this section, we discuss one such rule, called Bland's rule.

Recall that the variable in the LP form a vector $z = (z_1, \ldots, z_n)^T$, that is, every variable has an index. As we solve the LP, we swap columns in the tableau, the variables my change position in the tableau. It is important to note that when we say "index", we mean the index of the variable, which may be different from the number of the column in the current tableau that corresponds to the variable.

An oversimplified but easy to remember statement of Bland's rule is, "Always pick the variable with smallest index." A more precise description of the rule considers the two key steps of pivoting: choosing the non-basic variable that should enter the basis, and then choosing the basic variable that should leave the basis. We discuss these two steps next.

In our description of pivoting, we allowed any non-basic variable with positive objective function coefficient to be chosen as the one to enter the basis. Bland's rule tells us to choose the non-basic variable with minimum index from among all those with positive objective function coefficient. Assume we chose the variable z_j to enter the basis.

If the LP isn't unbounded, then there exists at least one constraint such that $a_{ij} > 0$. Thus, the following quantity is well-defined:

$$\Delta = \min \left\{ \frac{b_i}{a_{ij}} \;\middle|\; i \in [m], a_{ij} > 0 \right\}$$

Our earlier description of pivoting stated that the basic variable that should leave the basis is any variable z_{j_h} such that $\frac{b_h}{a_{hj}} = \Delta$. Once again, Bland's rule tells us to choose the basic variable z_{j_h} with minumum index j_h from among the variables that satisfy this condition.

This is the complete description of Bland's rule. The remainder of this section proves that using this rule ensures that the Simplex Algorithm does not cycle. In particular, it proves that Bland's rule ensures that there are no two tableaux with the same basis in the sequence of tableaux constructed by the Simplex Algorithm. Before we can prove this, need two simple facts about the behaviour of the Simplex Algorithm when it *does* cycle. Let $\langle T^{(1)}, \ldots, T^{(t)} \rangle$ be the sequence of tableaux constructed by the Simplex Algorithm, and let $\langle \hat{z}^{(1)}, \ldots \hat{z}^{(t)} \rangle$ be the sequence of the BFSs of these tableaux. If the algorithm cycles, then there exist two indices a and b such that $T^{(a)}$ and $T^{(b)}$ have the same basis.

Lemma 12. *If two tableaux $T^{(a)}$ and $T^{(b)}$ have the same basis, then $\hat{z}^{(a)} = \hat{z}^{(b)}$ and both solutions have the same objective function value.*

Proof. If $\hat{z}^{(a)} = \hat{z}^{(b)}$, then both solutions do have the same objective function value because $T^{(a)}$ and $T^{(b)}$ are equivalent and thus assign the same objective function value to any solution.

To prove that $\hat{z}^{(a)} = \hat{z}^{(b)}$, let $z_{j_1}, \ldots, z_{j_m}$ be the variables in the common basis of $T_{(a)}$ and $T_{(b)}$, and let the equality constraints of the LP represented by $T_{(a)}$ be $b = Az$. Then recall recall from § 2 that setting $\hat{z}_j^{(a)} = 0$ for all $j \notin \{j_1, \ldots, j_m\}$ uniquely determines the values of all basic variables in the basic solution $\hat{z}^{(a)}$ of $T^{(a)}$: $\hat{z}_{j_i} = b_j$, for all $i \in [m]$. Is particular, $\hat{z}^{(a)}$ is the only solution of $T^{(a)}$ that satisfies $\hat{z}_j^{(a)} = 0$, for all $j \notin \{j_1, \dots, j_m\}$.

Since $\hat{z}^{(b)}$ is the BFS of $T^{(b)}$, and $\{j_1, \dots, j_m\}$ is the basis of $T_{(b)}$, we have $\hat{z}^{(b)} = 0$, for all $j \notin \{j_1, \dots, j_m\}$. Since $T^{(a)}$ and $T^{(b)}$ are equivalent, $\hat{z}^{(b)}$ being a solution of $T^{(b)}$ implies that $\hat{z}^{(b)}$ is also a solution of $T^{(a)}$. Thus, since $\hat{z}^{(a)}$ is the only solution of $T^{(a)}$ with $\hat{z}_j^{(a)} = 0$, for all $j \notin \{j_1, \dots, j_m\}$, we must have $\hat{z}^{(a)} = \hat{z}^{(b)}$. ☐

Lemma 13. *If two tableaux $T^{(a)}$ and $T^{(b)}$ with $a < b$ have the same basis, then $\hat{z}^{(a)} = \cdots = \hat{z}^{(b)}$.*

Proof. By Lem. 12, $\hat{z}^{(a)}$ and $\hat{z}^{(b)}$ have the same objective function value. By Lem. 6, there is no index r such that $\hat{z}^{(r+1)}$ has a lower objective function value than $\hat{z}^{(r)}$. Together, these two facts imply that $\hat{z}^{(a)}, \dots, \hat{z}^{(b)}$ all have the same objective function value.

Now assume for the sake of contradiction that there exists an index $a \le r \le b$ such that $\hat{z}^{(r)} \neq \hat{z}^{(a)}$. Then we can choose r to be the smallest such index, that is, so that $\hat{z}^{(a)}) = \cdots = \hat{z}^{(r-1)}$. Let the LP represented by $T^{(r-1)}$ be

$$\begin{aligned} \text{Maximize } & cz + d \\ \text{s.t. } & b = Az \\ & z \ge 0. \end{aligned}$$

Then, if $\{j_1, \dots, j_m\}$ is the basis of $T^{(r-1)}$, we have

$$\begin{aligned} \hat{z}_{j_i}^{(r-1)} &= b_i && \forall i \in [m], \\ \hat{z}_j^{(r-1)} &= 0 && \forall j \in [n] \setminus \{j_1, \dots, j_m\}. \end{aligned}$$

If z_h is the non-basic variable that enters the basis in the pivot operation that produces $T^{(r)}$ from $T^{(r-1)}$, then, by Prop. 7,

$$\begin{aligned} \hat{z}_{j_i}^{(r)} &= b_i - a_{ih}\Delta && \forall i \in [m], \\ \hat{z}_h^{(r)} &= \Delta, \\ \hat{z}_j^{(r)} &= 0 && \forall j \in [n] \setminus \{h, j_1, \dots, j_m\}, \end{aligned}$$

where Δ is defined as in (12).

The non-basic variable z_h to enter the basis is chosen such that $c_h > 0$. Thus, the objective function values of $\hat{z}^{(r-1)}$ and $\hat{z}^{(r)}$ differ by $c_h\Delta$. As alrealy observed, $\hat{z}^{(r-1)}$ and $\hat{z}^{(r)}$ have the same objective function value. Therefore, $\Delta = 0$. This implies that

$$\begin{aligned} \hat{z}_{j_i}^{(r)} &= b_i - a_{ih}\Delta = b_i = \hat{z}_{j_i}^{(r-1)} = \hat{z}_{j_i}^{(a)} && \forall i \in [m], \\ \hat{z}_h^{(r)} &= \Delta \qquad\quad\;\; = 0 \;\;= \hat{z}_h^{(r-1)} = \hat{z}_h^{(a)}, \\ \hat{z}_j^{(r)} &= 0 \qquad\qquad\qquad = \hat{z}_j^{(r-1)} = \hat{z}_j^{(a)} && \forall j \in [n] \setminus \{h, j_1, \dots, j_m\}, \end{aligned}$$

Thus, $\hat{z}^{(r)} = \hat{z}^{(a)}$, contradicting the choice of $\hat{z}^{(r)}$. This proves the lemma. ☐

We are now ready to prove that Bland's rule prevents the Simplex Algorithm from cycling:

Proposition 14. *If the Simplex Algorithm uses Bland's rule, then no two tableaux in the sequence of tableaux produced by the algorithm have the same basis.*

Proof. Assume for the sake of contradiction that $T^{(a)}$ and $T^{(b)}$ have the same basis, for two indices $a < b$.

Let us call a variable z_j **fickle** if it belongs to the basis of one but not all of the tableaux $T^{(a)}, \dots, T^{(b)}$. Since two consecutive tableaux do not have the same basis, but the two bases of these two tableaux both have size m, it follows that there exist at least two fickle variables.

Let ℓ be the maximum index of all fickle variables, and let $a \le p, q < b$ be indices such that z_ℓ is a basic variable of $T^{(p)}$ and of $T^{(q+1)}$ but not of $T^{(p+1)}$ nor of $T^{(q)}$. In other words, z_ℓ leaves the basis in the pth pivot operation and enters the basis in the qth pivot operation. Since $T^{(a)}$ and $T^{(b)}$ have the same basis and z_ℓ is fickle, these indices exist.

Let z_h be the variable that enters the basis in the pth pivot operation, that is, z_h is a non-basic variable of $T^{(p)}$ and a basic variable of $T^{(p+1)}$. Then z_h is also fickle and, thus, by the choice of z_ℓ, $h < \ell$.

Now assume that $T^{(p)}$ represents the LP

$$\begin{aligned} \text{Maximize } & cz + d \\ \text{s.t. } & b = Az \\ & z \ge 0, \end{aligned}$$

and $T^{(q)}$, the LP

$$\begin{aligned} \text{Maximize } & c'z + d' \\ \text{s.t. } & b' = A'z \\ & z \ge 0. \end{aligned}$$

Let the basis of $T^{(p)}$ be $\{j_1, \dots, j_m\}$.

Every (not necessarily feasible) solution $\tilde{z}$ of $T^{(p)}$ that satisfies $\tilde{z}_h = \Delta$, for an arbitrary $\Delta \in \mathbb{R}$, and $\tilde{z}_j = 0$, for all $j \in [n] \setminus \{h, j_1, \dots, j_m\}$, satisfies

$$\tilde{z}_{j_i} = b_i - a_{ih}\Delta \quad \forall i \in [m].$$

Since all basic variables of $T^{(p)}$ have objective function coefficient 0 in $T^{(p)}$, the objective function value of this solution is

$$c_h\Delta + d.$$

Since $\tilde{z}_j = 0$, for all $i \in [n] \setminus \{h, j_1, \dots, j_m\}$, and $T^{(p)}$ and $T^{(q)}$ are equivalent, this value is the same as

$$c'_h\Delta + \sum_{i=1}^{m} c'_{j_i}\tilde{z}_{j_i} + d' = c'_h\Delta + \sum_{i=1}^{m} c'_{j_i}(b_i - a_{ih}\Delta) + d'.$$

Since this holds for every choice of Δ, this implies that

$$c_h = c'_h - \sum_{i=1}^{m} c'_{j_i} a_{ih}.$$

Now observe that, since z_h enters the basis in the pth pivot operation, we have $c_h > 0$. Since z_ℓ enters the basis in the qth pivot operation, we have $c'_\ell > 0$. If $c'_h > 0$, then z_h is a non-basic variable of $T^{(q)}$ that can be chosen to enter the basis in the qth pivot operation. Since $h < \ell$, as already observed, this would imply that ℓ does not enter the basis in the qth pivot operation, a contradiction. Thus, we must have

$c'_h \le 0$ and, therefore,

$$\sum_{i=1}^{m} c'_{j_i} a_{ih} < 0.$$

This implies that there exists an index $i \in [m]$ such that

$$c'_{j_i} a_{ih} < 0.$$

In particular $c'_{j_i} \neq 0$. Since all basic variables of $T^{(q)}$ have objective function coefficient 0 in $T^{(q)}$, this shows that z_{j_i} is not a basic variable of $T^{(q)}$. Since it *is* a basic variable of $T^{(p)}$, we conclude that z_{j_i} is fickle. By the choice of ℓ, this implies that $j_i \le \ell$.

Since z_ℓ is a basic variable of $T^{(p)}$, we have $\ell = j_{i_\ell}$, for some index $1 \le i_\ell \le m$. This index satisfies $a_{i_\ell h} > 0$ because z_ℓ leaves the basis in the pth pivot operation and z_h enters the basis in this operation. Since ℓ enters the basis in the qth pivot operation, we have $c'_\ell > 0$. Therefore,

$$c'_\ell a_{i_\ell \ell} > 0.$$

Since $c'_{j_i} a_{ih} < 0$, this shows that $j_i \neq \ell$. Therefore, since $j_i \le \ell$, we have $j_i < \ell$.

To finish the proof, observe that $c'_{j_i} \le 0$ because $j_i < \ell$ and z_ℓ is chosen to enter the basis in the qth pivot operation. Since $c'_{j_i} a_{ih} < 0$, this implies that $a_{ih} > 0$. Since z_{j_i} is not a basic variable of $T^{(q)}$, it has value $\hat{z}_{j_i} = 0$ in the BFS $\hat{z}$ of $T^{(q)}$. Since $T^{(p)}$ and $T^{(q)}$ have the same BFS, by Lem. 13, this implies that $b_i = 0$, that is, $\frac{b_i}{a_{ih}} = 0$. Since $\frac{b_{i'}}{a_{i'h}} \ge 0$, for every index $i' \in [m]$ that satisfies $a_{i'h} > 0$, this shows that z_{j_i} can be chosen to leave the basis in the pth pivot operation. Finally, we have the desired contradiction because $j_i < \ell$, z_ℓ leaves the basis in the pth pivot operation, and Bland's rule chooses the variable with smallest index from among those that can leave the basis in each pivot operation.

Since we derived this contradiction from the assumption that there exist two tableaux $T^{(a)}$ and $T^{(b)}$ with the same basis, no two such tableaux can exist, which is what the proposition states. □

8 Running Time

This discussion of the Simplex Algorithm has been rather long, so let us end it with a really short section. We have established the correctness of the Simplex Algorithm, and we have shown that using Bland's Rule ensures that the algorithm does not cycle. To round off the discussion, we need to bound the running time of the algorithm.

A tableau with n variables and m constraints has size $N = (m+1)(n+1)$. It is fair to consider N to be the input size of the algorithm. It is easily verified that every pivoting operation takes $O(N)$ time:

- In $O(n)$ time, we can inspect all objective function coefficients of non-basic variables, find the variables with positive objective function coefficients and choose the variable z_j with mininum index j among them.

- In $O(m)$ time, we can inspect the constraints and identify those that satisfy $a_{ij} > 0$ and minimize the quantity $\frac{b_i}{a_{ij}}$ among them. We then inspect the basic variables corresponding to these constraints and choose the one with minimum index j_h among them.

- With an efficient representation of the tableau, we can swap the columns corresponding to z_{j_h} and z_j in constant time. Using a naive implementation, this column swap can certainly be performed in $O(N)$ time.

- A single basic row operation can be implemented in $O(n)$ time, as it requires adding multiples of the $n+1$ entries in one row of the tableau to the $n+1$ entries in another row of the tableau. Since we can restore the tableau to standard form after a column swap by applying $m+1$ basic row operations, restoring the tableau to standard form takes $O(mn) = O(N)$ time.

Since a single pivoting operation takes $O(N)$ time, all we have to do is to bound the number of such operations the Simplex Algorithm performs before reporting an optimal solution of the fact that the LP is infeasible or unbounded. The following proposition provides such a bound.

Proposition 15. *If the Simplex Algorithm uses Bland's anti-cycling rule, it terminates after at most $\binom{n}{m}$ pivot operations.*

Proof. The basis of a tableau consists of m basic variables, chosen from the n variables of the tableau. Thus, there are $\binom{n}{m}$ distinct bases to choose from. Prop. 14 shows that Bland's rule ensures that no two tableaux in the sequence of tableaux constructed by the algorithm have the same basis. Thus, the sequence of tableaux constructed by the algorithm contains at most $\binom{n}{m}$ tableaux, which implies that the algorithm performs at most $\binom{n}{m}$ pivot operations. □

The upper bound on the number of pivot operations performed by the Simplex Algorithm established by Prop. 15 is not polynomial, and there exist artificial inputs where the algorithm does indeed need an exponential number of iterations to terminate. In practice, however, this exponential running time rarely materializes.