Section Notes 3
The Simplex Algorithm

Applied Math 121

Week of February 14, 2011

Goals for the week

- understand how to get from an LP to a simplex tableau.

- be familiar with reduced costs, optimal solutions, different types of variables and their roles.

- understand the steps of simplex phases I and II.

- be able to solve an LP problem fully using the simplex algorithm.

Contents

1 The Simplex Tableau, Reduced Costs and Optimality

The working of the simplex algorithm can best be illustrated when putting all information that is manipulated during the simplex algorithm in a special form, called the simplex tableau.

1.1 Getting from an LP to the Simplex Tableau

The simplex tableau resembles our notion of a matrix in canonical form. Thus, to put an LP into the tableau, we first need to transform it into standard equality form and we need an initial feasible basis.

Let's start with the following example:

$$
\begin{array}{llllll}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & x_1 & & & \leq & 2 \\
& x_1 & + & 2x_2 & \leq & 4 \\
& x_1, x_2 \geq 0 & & & &
\end{array}
$$

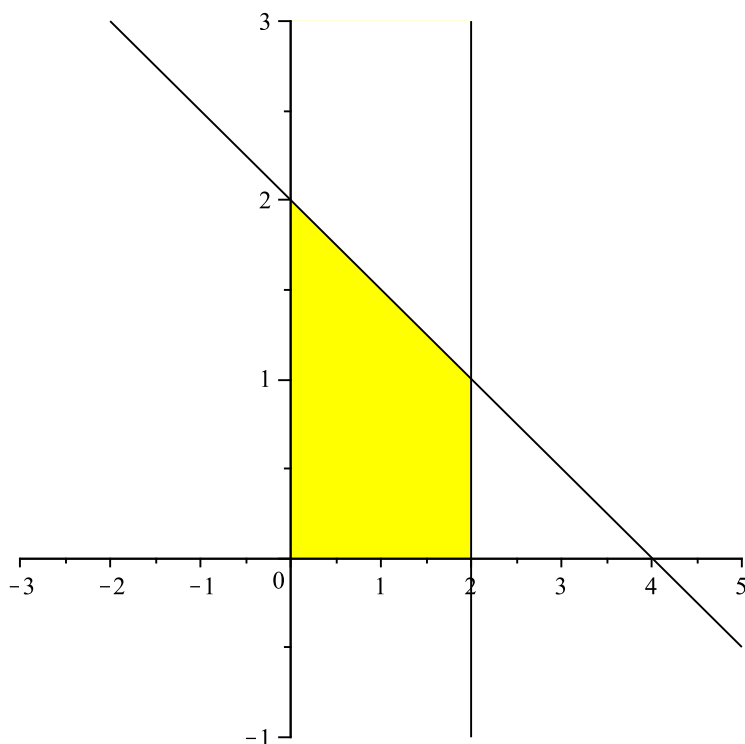Let's also draw the feasible region of the LP:

Figure 1: Feasible region of the LP

(Step 1) We will first bring the LP in standard equality form by introducing slack variables:

$$
\begin{array}{llllllllll}
\text{maximize} & x_1 & + & x_2 & & & & & \\
\text{subject to} & x_1 & & & + & x_3 & & & = & 2 \\
& x_1 & + & 2x_2 & & & + & x_4 & = & 4 \\
& x_1, x_2, x_3, x_4 \geq 0 & & & & & & & &
\end{array}
$$

2

(**Step 2**) Now we will denote the objective value of the LP with the variable z. Thus we get:

$$
\begin{array}{rrrrrrrrrrr}
\text{maximize} & x_1 & + & x_2 & & & & & = & z \\
\text{subject to} & x_1 & & & + & x_3 & & & = & 2 \\
& x_1 & + & 2x_2 & & & + & x_4 & = & 4 \\
\end{array}
$$
$$x_1, x_2, x_3, x_4 \geq 0$$

(**Step 3**) We rewrite the LP by only having z in the objective function and adding an additional constraint that represents the original objective function. Thus we get:

$$
\begin{array}{llrrrrrrrrr}
\text{maximize} & z \\
\text{subject to} & & x_1 & + & x_2 & & & & & = & z \\
& & x_1 & & & + & x_3 & & & = & 2 \\
& & x_1 & + & 2x_2 & & & + & x_4 & = & 4 \\
\end{array}
$$
$$x_1, x_2, x_3, x_4 \geq 0$$

(**Step 4**) From now on, we omit the objective function (well, we remember that we are maximizing z) and we rewrite the first constraint such that no variables are on the RHS. Thus we end up with the following system of equalities (remember the special role of the first equation though):

$$
\begin{array}{rrrrrrrrrr}
z & - & x_1 & - & x_2 & & & & = & 0 \\
& & x_1 & & & + & x_3 & & = & 2 \\
& & x_1 & + & 2x_2 & & & + & x_4 & = & 4 \\
\end{array}
$$
$$x_1, x_2, x_3, x_4 \geq 0$$

Note that in this form the decision variables have negative coefficients and the objective value shows up as z.

(**Step 5**) We need a basic feasible solution for this system of equations (ignoring the first line). Note that by choosing the slack variables to be our basic variables and setting them equal to the RHS we get the basis $[3, 4]$ and basic solution $x = [0, 0, 2, 4]^T$. Obviously this is a feasible solution because x is non-negative (this is due to the fact that the RHS of the system of equations was non-negative). Note that finding an initial feasible basis will not always be this easy. If it isn't, we need to run phase I of the simplex algorithm first (more on this later). But for now, let's assume we can find an initial feasible basis.

(Step 6) While the above system of equations might already be called a "simplex tableau", we will commonly write the tableau in the following, very special, tabular form (left: our specific example, right: with the general variables we will use):

	x_1	x_2	x_3	x_4	
z	-1	-1	0	0	0
x_3	1	0	1	0	2
x_4	1	2	0	1	4

$x = \{0, 0, 2, 4\}, z = 0$

	x_1	$\ldots$		x_n		
z	$-\bar{c}_1$	$\ldots$		$-\bar{c}_n$	z	
x_{B_1}	$\bar{a}_{11}$	$\bar{a}_{12}$	$\ldots$	$\bar{a}_{1n}$	b_1	
$x_{B_{\ldots}}$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	
x_{B_m}	$\bar{a}_{1m}$	$\bar{a}_{12}$	$\ldots$	$\bar{a}_{mn}$	$\bar{b}_m$	

x, z

This is what the simplex tableau looks like at the beginning and this tableau together with a basis matrix B and the corresponding basic feasible solution x will be the starting point for phase II of the algorithm. In every iteration of the simplex algorithm, we move from one tableau to the next one. Each element of the tableau is thereby keeping track of a particular kind of information. Let's go through the different elements of the simplex tableau.

- In the top line we simply list all **variable names**.

- In the next line we are keeping track of the **objective function**. We put z at the beginning to indicate that the value at the very end of this line corresponds to the current **objective value**. The numbers in the middle of this row correspond to the current **coefficients of the objective function**. We will sometimes refer to this row as the 0th row.

- The following m rows correspond to the m **constraints**. The numbers in the middle correspond to the current **coefficients of the variables** in the constraint. We will denote the number corresponding to constraint i and variable j with $\bar{a}_{ij}$. Note that after we performed any matrix operation, $\bar{a}_{ij}$ no longer corresponds to the a_{ij} from our initial constraint matrix A given by the LP. Furthermore, generally row i will not correspond to variable i! When we refer to the 'first row' we will always mean the the row corresponding to the first constraint. The 'second row' refers to the second constraint, etc.

- The numbers in the very last column of the constraint row correspond to the current **RHS of the equality constraints** which we will often denote with $\bar{b}$. The RHS of constraint i is referred to as $\bar{b}_i$. Note that after we performed matrix operations, this vector is no longer equal to the initial vector b from the LP.

- In the 0th column of the m rows for the m constraints we will put the variable names that correspond to the current basis. Thus, if we have variable name x_i in a particular row, the column corresponding to current basic variable x_i will only have a 1 in that row and 0's in all other rows. Note that the columns corresponding to the current basis will always form the identity matrix of dimension m-by-m.

- To keep track of the current basic solution found and the current objective value, we write down x and z behind the simplex tableau. Note that the current objective value is only given as z in the tableau.

4

1.2 Reduced Costs

As said before, the simplex algorithm moves from one tableau to the other, bringing new variables into the basis and having others leave. The question is: when have we found the optimal solution? To answer this question we look at the middle section of the 0th row of the tableau (the row starting with the z). Here we find one number for each variable. These numbers are called **reduced costs**. We will also often talk about the **reduced cost of variable** x_i which we will denote $\bar{c}_j$.

For our LP, which is a maximization problem, **the reduced cost of variable** x_i **corresponds to the decrease in the objective value when increasing that variable by 1. (Note when a variable with a negative reduced cost is increased, the objective value increases (e.g., decreases by a negative amount) because we have defined this line as** $z - x_1 - x_2... = 0$. So let's look at our example again:

	x_1	x_2	x_3	x_4	
z	-1	-1	0	0	0
x_3	1	0	1	0	2
x_4	1	2	0	1	4

$x = \{0, 0, 2, 4\}, z = 0.$

Observations for the current tableau:

- The reduced costs for our current basic variables are 0. This will always be the case!! Thus, increasing the value of the current basic variables does not help us increase the objective value.

- The reduced costs for the current non-basic variables are not equal to 0. Here, they are all negative. Thus, increasing the value of the corresponding non-basic variables (currently set to 0) will increase the objective value.

- Note: Whenever we are increasing the value of one variable, we will have to decrease the value of another variable to make sure the equality constraints are maintained. That's why the simplex algorithm iterates from one basis to the next (these are called "adjacent" bases).

1.3 Optimality

Now, we can answer the question when we have reached an optimal solution: **If we have a basic feasible solution** x **and all reduced costs are non-negative (i.e.** ≥ 0**) then** x **is an optimal feasible solution.**

> **Exercise 1**
>
> From our discussion above, why is this the case?
>
> **End Exercise 1**

1.4   Some Practice with Simplex Phase II

Now that we know when to stop, let's continue with our previous example and 'run' the simplex algorithm:

	x_1	x_2	x_3	x_4	
z	-1	-1	0	0	0
x_3	1	0	1	0	2
x_4	1	2	0	1	4

$x = \{0, 0, 2, 4\}, z = 0$.

We see that some reduced costs are negative, thus we (probably) haven't reached an optimum. Both non-basic variables x_1 and x_2 have negative reduced costs, thus they are both candidates for entering the basis. Let's choose x_1 to enter. Thus, x_1 will be our **pivot variable**, or in other words, we will **pivot on column 1**. Note that when x_1 enters the basis both equality constraints will be affected because x_1 shows up in both constraints.

> **Exercise 2**
>
> What would happen if all coefficients in the x_1 column were non-positive?
>
> **End Exercise 2**

However, in this case not all coefficients are non-positive. In fact, both coefficients are positive, thus both constraints will be affected. We now have to decide which row of the constraint matrix to pivot on. Let's choose row 1 (this is not arbitrary, we will get to this later). Thus, we subtract row 1 from row 2. This already gets us a new unit vector for the constraint matrix. To set the reduced cost of the new basic variable x_1 equal to 0, we must also add the first row to the 0th row. Thus, we end up having all 0's and one 1 in the first column. This is the new tableau:

	x_1	x_2	x_3	x_4	
z	0	-1	1	0	2
x_1	1	0	1	0	2
x_4	0	2	-1	1	2

$x = \{2, 0, 0, 2\}, z = 2$.

For the next step, there is really only one choice if we want to improve the objective value: variable x_2 has to enter the basis. Thus, let's pivot on x_2 (look at column 2). In this column, only x_4 has a positive coefficient. Thus, x_4 will have to leave the basis. To do the pivot operation, we divide row 2 by 2. Then, to set the reduced cost of x_2 equal to 0 we add the resulting row 2 to row 0 (note that we increased x_2 by the maximum amount possible before x_4 became negative). We get the following tableau:

	x_1	x_2	x_3	x_4	
z	0	0	$\frac{1}{2}$	$\frac{1}{2}$	3
x_1	1	0	1	0	2
x_2	0	1	$-\frac{1}{2}$	$\frac{1}{2}$	1

$x = \{2, 1, 0, 0\}, z = 3$.

6

Now we observe that all reduced costs are non-negative. Thus, we have found an optimal solution and the solution vector can be read directly from the tableau. All non-basic variables are equal to 0 and all basic variables are equal to the RHS of the corresponding row. Thus, $x = \{2, 1, 0, 0\}$. The objective value is the value of the entry in the upper right corner, i.e. $z = 3$. We have solved our first LP using the simplex algorithm!!

Let's verify graphically what we have done (moving from one basic feasible solution, i.e. extreme point, to the next in each iteration) and that we have indeed found the optimal solution.

2 A full iteration of the Simplex Algorithm phase II

We can now write down the mechanics of one iteration of the simplex algorithm using the tableau phase II:

1. A typical iteration starts with the tableau associated with a basis matrix B and the corresponding basic feasible solution x.

2. Examine the reduced costs in the zeroth row of the tableau. If they are all non-negative, the current basic feasible solution is optimal, and the algorithm terminates; else, choose some k for which the reduced cost $\bar{c}_k < 0$.

3. Consider the kth column, i.e. the pivot column, of the tableau, which we will denote by u. If no component of u is positive (i.e. > 0), the optimal objective value is ∞, and the algorithm terminates.

4. For each i for which u_i is positive, compute the ratio $\bar{b}_i/u_i$. Note that $\bar{b}_i$ refers to the current value of the basic variable that corresponds to row i (which we find by looking at the RHS of row i). Let l be the index of a row that corresponds to the smallest ratio, i.e. $l = \operatorname{argmin}_i\{\bar{b}_i/u_i\}$. The previously basic variable x_r that corresponds to row l exits the basis, and the variable x_k enters the basis (x_k *kicked out* variable x_r which must thus be *removed*). Note that the column corresponding to x_r had previously been equal to 1 in row l.

5. Add to each row of the tableau a constant multiple of the lth row (the pivot row) so that u_l (the pivot element) becomes one and all other entries of the pivot column become zero.

Exercise 3

Why do we use the smallest ratio test for choosing leaving variables? (e.g., what if we didn't?)

End Exercise 3

Exercise 4

This procedure iterates from one set of values to the next. What can we say about these values?

End Exercise 4

3 Degeneracy, Cycling, Anti-Cycling Rules

We have kept one little secret so far that complicates the simplex algorithm a little more: degeneracy.

Definition: Consider the standard form convex polyhedron $P = \{x \in R^n | Ax = b, x \geq 0\}$ and let x be a basic solution. Let m be the number of rows of A. The vector x is a **degenerate** basic solution if more than $n - m$ of the components of x are zero.

Note than so far we have assumed that exactly $n - m$ of the components of x are zero, namely all non-basic variables. Thus, for a basic solution to be degenerate, at least one basic variable must be zero!

3.1 Degeneracy and its Implications for the Simplex Algorithm

With degeneracy two new possibilities might be encountered by the simplex algorithm:

- If the current basic feasible solution x is degenerate, then let x_r denote the basic variable that is zero. Let l denote the corresponding row for x_r. If u_r is positive then the minimum ratio test will surely select row r as the pivot row and x_r will be removed from the basis. However, by letting another variable enter the basis, we are not able to improve the current solution because $\bar{b}_r$ which will be subtracted from $-z$ is 0. Note that the value of the entering basic variable will also be 0. Thus, we are staying at the same basic feasible solution.

- If the current basic feasible solution is not degenerate, it is still possible that in the next step of the simplex algorithm, more than one of the original basic variables becomes zero. Since only one of them exits the basis, the others remain in the basis at zero level, and the new basic feasible solution is degenerate.

When dealing with degeneracy it is possible that we have already reached an optimal solution but the reduced cost of a non-basic variable is still negative. In that case, it must be that when the new basic variable enters the basic variable that leaves was set to 0 before. Thus, we know that all reduced costs being non-negative is a sufficient condition for optimality, but if we allow for degeneracy it is not necessary. However, even with degeneracy we can always run the simplex algorithm until we find a solution where all reduced costs are non-negative. Sometimes this might include changing the basis while staying at the same basic feasible solution.

Note that it is not particularly desirable to stay at the same basic feasible solution when changing the basis. It is sometimes the case that multiple changes of the basis at a degenerate basic solution

may lead to the eventual discovery of a non-degenerate basic solution. But in general, it is also possible that upon multiple basis changes we are back where we began in which case the algorithm may cycle indefinitely.

3.2   Degeneracy and Cycling: Anti-Cycling Rules

Note that cycling cannot occur when there are no degenerate basic solutions. But when there are, we must take special care to avoid cycling. We do this via intelligent book keeping. We must avoid "doing the same bad thing twice". Note that the only freedom we have is: 1) when we choose a pivot column among all columns with positive reduced cost and 2) when we choose a pivoting row among all rows that achieve the smallest ratio in the minimum ratio test. To do so, there are two commonly used anti-cycling rules:

- **Lexicographic pivoting rule:** (see lecture notes for definitions)

- **Smallest subscript pivoting rule** (see lecture notes for definitions)

It has been proved that using either one of the two anti-cycling rules, the simplex algorithm does not cycle and thus terminates after a finite number of iterations.

4   Simplex Algorithm: Phase I

Up until this point, we have assumed that we have an initial feasible basis.

> **Exercise 5**
>
> When the problem is given to us in standard inequality form, what needs to be done to get an initial feasible tableau?
>
> **End Exercise 5**

However, in general this conversion may not be possible and thus in this section we consider how to establish an initial feasible tableau for an arbitrary problem. We'll introduce this method via the following example:

$$
\begin{array}{rrrrrrl}
\text{maximize} & 2x_1 & + & x_2 & & & \\
\text{subject to} & x_1 & + & x_2 & & & = \quad 3 \\
& -x_1 & + & x_2 & - & x_3 & = \quad 1 \\
\multicolumn{7}{c}{x_1, x_2, x_3 \geq 0}
\end{array}
$$

4.1   Introduction of artificial variables

To begin, we'll ignore the objective and concentrate on the constraints. Specifically, we'll introduce two new *artificial* variables that capture the error in the constraints:

$$
x_4 = 3 - (x_1 + x_2)
$$
$$
x_5 = 1 - (-x_1 + x_2 - x_3)
$$

We can now attempt to solve a new *auxiliary* LP that tries to force these errors to zero:

$$
\begin{array}{rrrrrrrrrr}
\text{minimize} & x_4 & + & x_5 & & & & & & \\
\text{subject to} & x_4 & = & 3 & - & (x_1 & + & x_2) & & \\
& x_5 & = & 1 & - & (-x_1 & + & x_2 & - & x_3) \\
& x_1, & x_2, & x_3, & x_4, & x_5 & \geq & 0 & &
\end{array}
$$

This LP is important, because of the following theorem:

The original LP is feasible $\Leftrightarrow$ the auxiliary LP has an optimal solution with value zero.

The forward direction is quite straightforward: given a feasible solution for the original LP, setting the artifical variables to zero yields a solution to the auxiliary problem with value zero. And, given that a solution to the auxiliary problem has value 0, then the artificial variables must likewise be zero. And thus, the non-auxiliary variables must constitute a feasible solution to the original problem.

4.2 Solve auxiliary problem

We have thus transformed the problem of determining feasibility of the original linear program, into one of solving an *auxiliary* LP. Putting this auxiliary problem in standard form we get (notice we have multiplied the objective by -1 to bring it into a maximization problem):

$$
\begin{array}{lrrrrrrrl}
\text{maximize} & w & & & & + & x_4 & + & x_5 & = & 0 \\
\text{subject to} & x_1 & + & x_2 & & & + & x_4 & & = & 3 \\
& -x_1 & + & x_2 & - & x_3 & & & + & x_5 & = & 1 \\
& x_1, & x_2, & x_3, & x_4, & x_5 & \geq & 0
\end{array}
$$

Now we use Simplex to solve this problem, noting that unlike the original problem, the auxiliary problem has an obvious initial basis – composed of the artificial variables. All we need to do is subtract the 2nd and 3rd lines from the first in order to establish a basis:

$$
\begin{array}{lrrrrrrrl}
\text{maximize} & w & & - & 2x_2 & + & x_3 & & & = & -4 \\
\text{subject to} & & x_1 & + & x_2 & & & + & x_4 & = & 3 \\
& - & x_1 & + & x_2 & - & x_3 & & + & x_5 & = & 1 \\
& & x_1, & x_2, & x_3, & x_4, & x_5 & \geq & 0
\end{array}
$$

Note: if any of the RHS values are negative, we would have to multiply by -1 first, to ensure we are in canonical form.

Now we proceed to solve via simplex. We choose index $k = 2$ as our entering index, as it is the only one with a negative reduced cost. Then we compute the ratios b_i/a_{ik}. In this case, using $i = 4$ gives us the ratio $3/1$ and using $i = 5$ gives us $1/1$. We want the smallest ratio, so we choose $i = 5$. We therefore pivot on the entry for $k = 2$ in the row corresponding to $i = 5$:

$$
\begin{array}{lrrrrrrrl}
\text{maximize} & w & & - & 2x_2 & + & x_3 & & & = & -4 \\
\text{subject to} & & x_1 & + & x_2 & & & + & x_4 & = & 3 \\
& - & x_1 & + & \boxed{x_2} & - & x_3 & & + & x_5 & = & 1 \\
& & x_1, & x_2, & x_3, & x_4, & x_5 & \geq & 0
\end{array}
$$

After the pivot we have:

$$
\begin{array}{lrrrrrrrl}
\text{maximize} & w & - & 2x_1 & & & - & x_3 & & & + & 2x_5 & = & -2 \\
\text{subject to} & & & 2x_1 & & & + & x_3 & + & x_4 & - & x_5 & = & 2 \\
& & - & x_1 & + & x_2 & - & x_3 & & & + & x_5 & = & 1 \\
& & & x_1, & x_2, & x_3, & x_4, & x_5 & \geq & 0
\end{array}
$$

This $(2, 4)$ basis gives us a feasible solution of $[0, 1, 0, 2, 0]$ with an objective value of 2. Both x_1 and x_3 have negative reduced costs. Suppose we choose x_3. Looking at the x_3 column, only index $i = 4$ has a positive coefficient (with a ratio of $2/1$). So we choose x_4 for our leaving variable, leading us to pivot around the indicated entry:

$$
\begin{array}{lrrrrrrrl}
\text{maximize} & w & - & 2x_1 & & & - & x_3 & & & + & 2x_5 & = & -2 \\
\text{subject to} & & & 2x_1 & & & + & \boxed{x_3} & + & x_4 & - & x_5 & = & 2 \\
& & - & x_1 & + & x_2 & - & x_3 & & & + & x_5 & = & 1 \\
& & & x_1, & x_2, & x_3, & x_4, & x_5 & \geq & 0
\end{array}
$$

Which gives us:

$$
\begin{array}{rlrrrrrrrrr}
\text{maximize} & w & & & & & + & x_4 & + & x_5 & = & 0 \\
\text{subject to} & & 2x_1 & & & + & x_3 & + & x_4 & - & x_5 & = & 2 \\
& & x_1 & + & x_2 & & & + & x_4 & & & = & 3 \\
& & x_1, & x_2, & x_3, & x_4, & x_5 & \geq & 0 & & &
\end{array}
$$

which gives us a basis of (2,3) for the basic feasible solution $[0, 3, 2, 0, 0]$ and an objective value of 0. Because all of our reduced costs are positive, we have an optimal solution with objective value 0. Note that we have also found a feasible solution for the original problem: $[0, 3, 2]$.

4.3 Constructing an initial tableau

We now know our original problem was feasible. However, we still need to find an initial tableau for our original problem. We begin by taking the final tableau from the auxiliary problem, removing the artificial variables, and replacing the objective with our original objective:

$$
\begin{array}{rlrrrrrr}
\text{maximize} & z & - & 2x_1 & - & x_2 & & & = & 0 \\
\text{subject to} & & & 2x_1 & & & + & x_3 & = & 2 \\
& & & x_1 & + & x_2 & & & = & 3 \\
& & x_1, & x_2, & x_3 & \geq & 0 & & &
\end{array}
$$

We still have basic variables in the objective however. Targeting (2,3) as the basis, we can remove these variables from the objective:

$$
\begin{array}{rlrrrrrr}
\text{maximize} & z & - & x_1 & & & & & = & 3 \\
\text{subject to} & & & 2x_1 & & & + & x_3 & = & 2 \\
& & & x_1 & + & x_2 & & & = & 3 \\
& & x_1, & x_2, & x_3 & \geq & 0 & & &
\end{array}
$$

From here we are ready to continue to phase II of simplex, as described in the previous section.

4.4 Phase I more formally

1. We are given a constrained system:

$$Ax = b$$
$$x \geq 0$$

where A is an $m \times n$ matrix, x is a vector of variables and the elements of the constant vector b are all non-negative (and otherwise we multiply that row by -1).

2. We introduce a set of m *artificial* variables u:

$$Ax + Iu = b$$
$$x \geq 0$$

3. We then attempt to use simplex to solve the system:

$$\max \quad \sum_{i=1}^{m} u_i$$
$$\text{s.t.} \quad Ax + Iu = b$$
$$x, u \geq 0$$

- An initial feasible tableau is easily formed for the u variables by eliminating them from the objective.

- If, after some iteration, a given artificial variable u_r has become nonbasic we delete all appearances of the variable in the tableau. We can do this because, this deletion is identical to setting the variable to zero and the resulting tableau will still attempt to minimize the remaining error in the original problem's constraints.

4. After completing simplex, there are three possibilities:

   (a) The optimal value is strictly positive, in which case the original problem is infeasible.

   (b) The optimal value is zero, and all artificial variables have been deleted. In this case we have thus created an initial feasible tableau for our original linear program.

   (c) The optimal value is zero, and one or more artificial variables is still basic. Since the objective is zero, these variables must be zero, making our tableau degenerate. Lets assume the remaining variable is u_i and remains in a line of our tableau with the form:

   $$\sum_{j=1}^{n} a_{ij} x_j + u_i = b_i$$

   Now we have two possibilities:

   i. Every coefficient a_{ij} for $j \in 1...n$ is zero. In this case the corresponding entry in the original system is a linear combination of the other equations. It will thus have no bearing on the outcome of the system and we might as well delete it and u_i as well.

   ii. Some coefficient a_{ij} is nonzero, in which case we can pivot on this entry, causing u_i to become nonbasic at which point we can delete it.

It is important to recognize the difference between the various types of variables:

Exercise 6

What are the differences among decision variables, slack variables, and artificial variables?

End Exercise 6

In general you will need to introduce artificial variables to solve Phase I.

Note: For inequality constraints with a positive RHS, you can use the slack variable for your artificial variable directly.

5   Understanding the Simplex Tableau

To make sure we understand the true meaning of all elements of the simplex tableau, let's look at the following. Assume that while solving a standard form problem, we have arrived at the following tableau, with x_3, x_4 and x_5 being the current basic variables:

	x_1	x_2	x_3	x_4	x_5	
z	δ	-2	0	0	0	10
x_3	-1	η	1	0	0	3
x_4	α	4	0	1	0	2
x_5	γ	-3	0	0	1	β

Exercise 7

The entries $\alpha, \beta, \gamma, \delta, \eta$ in the tableau are unknown parameters. For each one of the following statements, find the conditions on the parameter values that will make the statement true.

1. The current solution is feasible.

2. The optimal objective value is ∞.

3. The current solution is feasible but not optimal.

End Exercise 7

14

6 An Example of Simplex

Let us work through a complete example (easy one) illustrating the full simplex algorithm (phase I + II). We will then plot the feasible region in 3D and trace the extreme points that the simplex algorithm goes through. Here is the linear program:

$$
\begin{array}{rrrrrcr}
\text{maximize} & 2x_1 & + & 3x_2 & - & x_3 & \\
\text{subject to} & x_1 & & & & & \leq & 4 \\
& & & x_2 & & & \leq & 6 \\
& x_1 & & & - & x_3 & \leq & \text{-2} \\
& x_1, & x_2, & x_3 & \geq & 0 &
\end{array}
$$

Exercise 8

Put the LP into table form and introduce slack variables.

End Exercise 8

Exercise 9

Make the RHS positive.

End Exercise 9

Exercise 10

Now we are ready to begin Phase I: introduce artificial variables where necessary.

End Exercise 10

Exercise 11

Obtain a basic feasible solution.

End Exercise 11

Exercise 12

Now we enter phase II: construct a tableau generated from Phase I and the original objective. Iterate through until the optimal solution is reached.

End Exercise 12

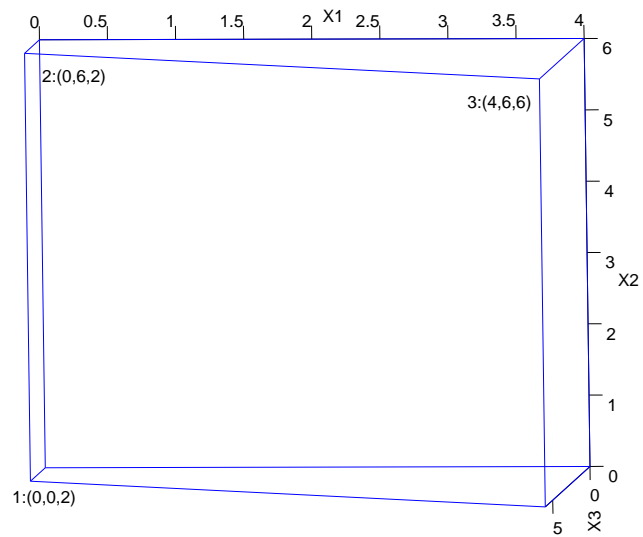A plot of the trajectory the simplex algorithm took is given in Figure 2.

Figure 2: Graph of Simplex progress

7 Pivoting in Maple/Matlab

7.1 Maple

Maple is great for doing simple matrix operations. It even has a built in pivot command. This is how it works.

Consider the following matrix, from section notes 2:

$$\begin{vmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{vmatrix}$$

To declare the matrix in Maple, we first use the linalg package:

```
with(linalg);
```

and then declare a matrix A as follows:

```
A := <<1,2>|<2,3>|<3,4>>;
```

Notice that in the above notation Maple is taking in columns, such that `<1,2>` is the column vector in the first column of A, and so on.

Okay. Let's say we want to isolate the 1 in row 1, column 1. We can use the pivot command, which takes a matrix, a row and a column:

```
pivot(A, 1, 1);
```

results in the following output from Maple:

$$\begin{vmatrix} 1 & 2 & 3 \\ 0 & -1 & -2 \end{vmatrix}$$

Notice that the element in row 1, column 1 is now "isolated".

Here we have just called the pivot command, but did not save the output of the command into a variable. If I check the value of the matrix A (by typing matrix(A) and pressing enter), I will see that its unchanged. So let's just recall the command again, this time storing the resulting matrix in a matrix B:

```
B := pivot(A,1,1);
```

In general, Maple does not normalize the isolated value to 1. To fix this, you can normalize using the mulrow command, which takes a matrix, a row, and a normalizing constant to multiply by. For example:

```
B := mulrow(B, 1, 1/B[1,1]);
```

would multiply the first row of B by 1/B[1,1], which has the effect of normalizing the first row such that A[1,1] is going to be 1. The resulting matrix is stored in B (now B is the normalized matrix) in this example, but you could store it in another matrix (say C). From here, you can call pivot again on any rows or columns you desire to isolate.

7.2 Matlab

Unlike Maple, Matlab does not have a built in pivot command, but it is easy to write a function that will perform the operation:[1]

```
function R = pivot(M, r, c)
  [d, w] = size(M);          % Get matrix dimensions
  R = zeros(d, w);           % Initialize to appropriate size
```

---

[1]Adapted from http://classes.apl.washington.edu/Math407Summer2005/pivot.xml

```
  R(r,:) = M(r, :) / M(r,c);      % Copy row r, normalizing M(r,c) to 1
  for k = 1:d                     % For all matrix rows
    if (k ~= r)                   % Other then r
      R(k,:) = M(k,:) ...         % Set them equal to the original matrix
            - M(k,c) * R(r,:);    % Minus a multiple of normalized row r, making R(k,c)=0
    end
  end
end
```

With this function defined, we can perform the identical operation as above:

```
  A = [1 2 3; 2 3 4]; B = pivot(A, 1, 1);
```

which we set B equal to:

$$\begin{vmatrix} 1 & 2 & 3 \\ 0 & -1 & -2 \end{vmatrix}$$

8 Solutions

Solution 1

The reduced costs refer to the decrease in the objective value when increasing the corresponding variable. If all reduced costs are non-negative, the value of the objective function cannot be increased further by another basis change. Thus, when iterating from one simplex tableau to the next, we can stop the algorithm when all reduced costs (all numbers in the middle section of the z row) are non-negative.

End Solution 1

Solution 2

In that case, we could set x_1 to an arbitrarily high value because doing so would only decrease the RHS of each constraint and we could increase our previous basic variables x_3 and x_4 to compensate for this while this maintaining all equality constraints. Note that because the reduced costs of the basic variables are zero, increasing them does not change the objective value. Thus, by increasing the value of x_1 we could increase the objective value arbitrarily high. Thus, the problem would be unbounded.

End Solution 2

Solution 3

We use the smallest ratio test to figure out which constraint binds first as we increase the value of the corresponding variable. If we were to increase the value of a variable beyond the smallest ratio, then the constraint with the smallest ratio will have a larger left-hand side that can only be balanced by choosing negative values for the variable leaving the basis, which we can't do due to non-negativity constraints. Unlike choosing the entering variable, we don't have a choice in choosing the leaving variable (e.g., unless there is a tie in the minimum ratios).

End Solution 3

Solution 4

The values represent basic feasible solutions. We always iterate from one basic feasible solution to the next.

End Solution 4

Solution 5

We introduce slack variables to put the program in equality form. These
slack variables will constitute a feasible basis.

End Solution 5

Solution 6

- Decision Variables: Are the variables defined in the original LP
  problem.

- Slack Variables: Are introduced to transform inequalities into
  equalities. They can take any non-negative value at feasible
  solutions to the original problem.

- Artificial Variables: Are introduced in the context of Phase I to
  attempt to find an initial feasible tableau. They must be zero at
  such a feasible solution.

End Solution 6

Solution 7

1. We need $\beta \geq 0$ for feasibility.

2. The optimal objective value is ∞ when we have a feasible solution
   in the current tableau, a nonbasic variable x_i with reduced cost
   < 0 and for the pivot column u, $u_i \leq 0$ for all i. We need $\beta \geq 0$
   for feasibility. The variable x_2 cannot satisfy all the conditions
   for the ∞ optimal objective value (because 4 is positive). For
   the variable x_1, the conditions then can be expressed as follows:
   $\alpha \leq 0, \gamma \leq 0$, and $\delta < 0$.

3. The current solution is feasible if $\beta \geq 0$. If the solution is not
   degenerate then the current solution is definitely not optimal. Thus
   a condition could be simply $\beta > 0$.

End Solution 7

Solution 8

Introduce slack variables and write the LP as a tableau:

	x_1	x_2	x_3	x_4	x_5	x_6	
z	-2	-3	1	0	0	0	0
	1	0	0	1	0	0	4
	0	1	0	0	1	0	6
	1	0	-1	0	0	1	-2

End Solution 8

Solution 9

Multiply the last line by -1 to get the RHS to be positive:

	x_1	x_2	x_3	x_4	x_5	x_6	
z	-2	-3	1	0	0	0	0
	1	0	0	1	0	0	4
	0	1	0	0	1	0	6
	-1	0	1	0	0	-1	2

End Solution 9

Solution 10

Now we are ready to begin Phase I, which proceeds by introducing an artificial variable for the constraint that can't use a slack variable:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
z	-2	-3	1	0	0	0	0	0
	1	0	0	1	0	0	0	4
	0	1	0	0	1	0	0	6
	-1	0	1	0	0	-1	1	2

End Solution 10

Solution 11

Now we replace the objective with the appropriate one for Phase I:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
w	0	0	0	0	0	0	1	0
	1	0	0	1	0	0	0	4
	0	1	0	0	1	0	0	6
	-1	0	1	0	0	-1	1	2

Straight away, we can see that $x = (0, 0, 2, 4, 6, 0, 0)$, $w = 0$ is a solution to this problem, giving us a feasible solution and basis $\{3,4,5\}$ to the original problem.[2]

End Solution 11

---

[2]Note that this is also not surprising, since we could have used x_3 as a basic variable initially without introducing the artificial variable x_7.

Solution 12

Next we construct a tableau generated from Phase I and the original objective:

	x_1	x_2	x_3	x_4	x_5	x_6	
z	-2	-3	1	0	0	0	0
x_4	1	0	0	1	0	0	4
x_5	0	1	0	0	1	0	6
x_3	-1	0	1	0	0	-1	2

We need to subtract row 2 from row 0, in order to truly establish x_3 in the basis:

	x_1	x_2	x_3	x_4	x_5	x_6	
z	-1	-3	0	0	0	1	-2
x_4	1	0	0	1	0	0	4
x_5	0	1	0	0	1	0	6
x_3	-1	0	1	0	0	-1	2

$x = \{0, 0, 2, 4, 6, 0\}, z = -2.$

Now we choose x_2 to enter. The only choice for a leaving index is x_5 with a ratio of 6/1. To accomplish this, we add 3 times row 2 to row 0:

	x_1	x_2	x_3	x_4	x_5	x_6	
z	-1	0	0	0	3	1	16
x_4	1	0	0	1	0	0	4
x_2	0	1	0	0	1	0	6
x_3	-1	0	1	0	0	-1	2

$x = \{0, 6, 2, 4, 0, 0\}, z = 16.$

Now we pick x_1 to enter. Again we have only one choice for a leaving index, x_4 with a ratio of 4/1. To accomplish this we first add row 1 to row 0 and then add row 1 to row 3:

	x_1	x_2	x_3	x_4	x_5	x_6	
z	0	0	0	1	3	1	20
x_1	1	0	0	1	0	0	4
x_2	0	1	0	0	1	0	6
x_3	0	0	1	1	0	-1	6

$x = \{4, 6, 6, 0, 0, 0\}, z = 20.$

Since all of our reduced costs are negative, we know we are done. The objective of 20 must be the maximum attainable. And our final solution is $x = (4, 6, 6)$.

End Solution 12