

18.433 Combinatorial Optimization

The Simplex Algorithm

October 16, 23

Lecturer: Luis Rademacher

We proved the following:

Lemma 1 (Farkas). *Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Exactly one of the following conditions is true:*

1. $\exists x \in \mathbb{R}^n : Ax = b$ and $x \geq 0$.
2. $\exists y \in \mathbb{R}^m : b^T y < 0$ and $A^T y \geq 0$.

Consider a linear problem in standard form and its dual:

$$\begin{array}{ll} (P) & \text{opt}(P) = \max c^T x \\ & \text{s.t. } Ax = b \\ & x \geq 0 \end{array} \qquad \begin{array}{ll} (D) & \text{opt}(D) = \min b^T y \\ & \text{s.t. } A^T y \geq c \end{array}$$

We say that a problem is bounded if its optimum value is finite. We proved the weak duality theorem:

Theorem 2. *If (P) and (D) are feasible, then $\text{opt}(P) \leq \text{opt}(D)$ and finite. In particular, (P) is bounded.*

The following corollary is an immediate consequence:

Corollary 3. *If (D) is feasible, then (P) is bounded or infeasible. If (P) is feasible, then (D) is bounded or infeasible.*

1 Strong duality

The previous corollary says that some combinations of being feasible (F), bounded (B), unbounded (U) and infeasible (I) for the primal and dual are not possible. For example, primal feasible, dual feasible and unbounded is impossible by the corollary. The strong duality theorem will tell us exactly which combinations are possible and, additionally, it will prove that in the case where the primal is bounded and feasible we have that $\text{opt}(P) = \text{opt}(D)$, which is the most important conclusion.

Theorem 4 (Strong duality). For a linear program (P) and its dual (D) there are only the following possibilities:

1. (P) B F and (D) B F. In this case $\text{opt}(P) = \text{opt}(D)$.
2. (P) I, (D) U F.
3. (P) U F, (D) I.
4. (P) I, (D) I.

Proof. A priori, there are 9 possible combinations. Weak duality already ruled out

- (P) B F, (D) U F,
- (P) U F, (D) B F, and
- (P) U F, (D) U F.

We will eliminate the case (P) B F, (D) I, then duality eliminates (P) I, (D) B F. Only the 4 claimed cases survive.

Assume that (P) B F. Let $z > \text{opt}(P)$. Apply Farkas' lemma to $A_0 = \begin{pmatrix} A \\ c^T \end{pmatrix}$, $b_0 = \begin{pmatrix} b \\ z \end{pmatrix}$. We know that $c^T x < z$ for all feasible x , that is, x satisfying $Ax = b$ and $x \geq 0$. In other words, for all x , $x \geq 0$ implies that $A_0 x = \begin{pmatrix} Ax \\ c^T x \end{pmatrix} \neq \begin{pmatrix} b \\ z \end{pmatrix} = b_0$, i.e. condition (1) in Farkas' lemma is not satisfied; thus, (2) in the lemma is true: there exists $y \in \mathbb{R}^m$ and there exists $\alpha \in \mathbb{R}$ such that $b^T y + z\alpha < 0$ and $A^T y + \alpha c \geq 0$.

We will now see that $\alpha < 0$. Else, let $x^* \in \mathbb{R}^n$ be primal optimal, that is a primal feasible point such that $c^T x^* = \text{opt}(P)$. Then the conditions that y and α satisfy imply:

$$x^{*T} A^T y + \alpha c^T x^* \geq 0$$

If $\alpha \geq 0$ we can get

$$b^T y + \alpha z \geq 0$$

which is a contradiction.

Thus, $\alpha < 0$ and $y_0 = -y/\alpha$ satisfies $b^T y_0 < z$ and $A^T y_0 \geq c$. That is, the dual is feasible (and bounded). Moreover, $z \geq \text{opt}(P)$ was arbitrary, i.e. for any

$$z > \max_{x \geq 0, Ax=b} c^T x$$

there exists y_0 dual feasible such that

$$\max_{x \geq 0, Ax=b} c^T x \leq \min_{y : A^T y \geq c} b^T y \leq b^T y_0 < z$$

The fact that z is arbitrary implies that

$$\max_{x \geq 0, Ax=b} c^T x = \min_{y : A^T y \geq c} b^T y.$$

□

2 Linear programs

Recall our standard form for a linear program:

$$\begin{aligned} \text{maximize } z = \mathbf{c}^t \mathbf{x}, \text{ subject to } & \mathbf{Ax} \leq \mathbf{b}, \\ & \mathbf{x} \geq 0. \end{aligned} \tag{1}$$

Let us concentrate on a concrete example, the program

$$\begin{aligned} \text{maximize } z = 5x_1 + 4x_2 + 3x_3, \text{ subject to } & 2x_1 + 3x_2 + x_3 \leq 5, \\ & 4x_1 + x_2 + 2x_3 \leq 11, \\ & 3x_1 + 4x_2 + 2x_3 \leq 8, \\ & x_1, x_2, x_3 \geq 0. \end{aligned} \tag{2}$$

3 The simplex algorithm

3.1 Insert slack variables

To solve the linear program above using the simplex algorithm, we first convert the constraints involving \leq 's to equality constraints by introducing **slack variables**. Each \leq inequality, $\sum_{j=1}^n a_{ij}x_j \leq b_i$, is replaced by the equality, $\sum_{j=1}^n a_{ij}x_j + x_{n+i} = b_i$, and an additional constraint that the slack variables are non-negative, $x_{n+i} \geq 0$.

In terms of our specific example, adding slack variables gives the linear program:

$$\begin{aligned} \text{maximize } z = 5x_1 + 4x_2 + 3x_3, \text{ subject to } & 2x_1 + 3x_2 + x_3 + x_4 = 5, \\ & 4x_1 + x_2 + 2x_3 + x_5 = 11, \\ & 3x_1 + 4x_2 + 2x_3 + x_6 = 8, \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. \end{aligned} \tag{3}$$

3.2 Increase the objective function value through a pivot

Let us focus on our specific example. By isolating the slack variables in the equalities, we see that

$$\begin{aligned}x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\x_6 &= 8 - 3x_1 - 4x_2 - 2x_3\end{aligned}\tag{4}$$

This suggests one feasible solution,

$$\begin{aligned}x_1, x_2, x_3 &= 0, \\x_4 &= 5, \\x_5 &= 11, \\x_6 &= 8.\end{aligned}\tag{5}$$

For this solution our objective function is

$$z = 5x_1 + 4x_2 + 3x_3 = 0,\tag{6}$$

which seems low. How can we do better? Perhaps we should attempt to increase x_1 .

If we increase x_1 and hold x_2 and x_3 at zero, we can calculate the required values of x_4, x_5 , and x_6 from (4). For example,

$$\begin{aligned}x_1 = 1, \quad x_2, x_3 = 0, &\Rightarrow z = 5, \quad x_4 = 3, \quad x_5 = 7, \quad x_6 = 5, \\x_1 = 2, \quad x_2, x_3 = 0, &\Rightarrow z = 10, \quad x_4 = 1, \quad x_5 = 3, \quad x_6 = 2, \\x_1 = 3, \quad x_2, x_3 = 0, &\Rightarrow z = 15, \quad x_4 = -1, \quad x_5 = -1, \quad x_6 = -1.\end{aligned}\tag{7}$$

Increasing x_1 improves the objective function value, but we cannot push x_1 too far or the slack variables become negative, as is the case when $x_1 = 3$.

We can calculate the values of x_1 that preserve nonnegativity for the slack variables from (4).

$$\begin{aligned}x_4 \geq 0 &\Rightarrow x_1 \leq \frac{5}{2}, \\x_5 \geq 0 &\Rightarrow x_1 \leq \frac{11}{4}, \\x_6 \geq 0 &\Rightarrow x_1 \leq \frac{8}{3},\end{aligned}\tag{8}$$

so the most we can increase x_1 is to $x_1 = \frac{5}{2}$.

Let's rewrite the equality constraints (4) to reflect our decision that $x_1 = \frac{5}{2}$ and $x_4 = 0$.

$$\begin{aligned}x_1 &= \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4, \\x_5 &= 11 - 4\left(\frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4\right) - x_2 - 2x_3, \\x_6 &= 8 - 3\left(\frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4\right) - 4x_2 - 2x_3.\end{aligned}\tag{9}$$

and our objective function from (3) may be written $z = \frac{25}{2} - \frac{1}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4$.

This sequence of operations is called a **pivot**, for reasons which will make more sense when we write everything in tableau form.

3.3 Repeat

We have improved our objective function value from 0 to $\frac{25}{2}$ by pivoting around x_1 . Let's try to do it again. Increasing x_2 will hurt our objective value, as will increasing x_4 . The only thing we can increase is x_3 .

$$\begin{aligned}x_1 \geq 0 &\Rightarrow x_3 \leq 5, \\x_5 \geq 0 &\text{ puts no constraint on } x_3, \\x_6 \geq 0 &\Rightarrow x_3 \leq 1,\end{aligned}\tag{10}$$

so we'll increase x_3 as far as we can, to 1.

Rewriting the equality constraints to reflect our choice of $x_3 = 1$ and $x_6 = 0$ yields

$$\begin{aligned}x_3 &= 1 + x_2 + 3x_4 - 2x_6, \\x_5 &= 1 + 5x_2 + 2x_4, \\x_1 &= 1 - 2x_2 - 2x_4 + x_6,\end{aligned}\tag{11}$$

and objective function $z = 13 - 3x_2 - x_4 - x_6$.

When the objective function is written in this form, it is clear that increasing any of the variables x_2, x_4 , or x_6 will decrease the objective function value. Therefore the value is at its maximum, 13, when $x_2, x_4, x_6 = 0$.

4 High level description of the simplex algorithm

In a linear program of the form

$$\begin{aligned}\max & c'x \\ \text{s.t.} & Ax = b \\ & x \geq 0,\end{aligned}$$

with m equality constraints and n variables, if one sets $n - m$ variables to 0 and lets the equality constraints determine the value of the rest and this values are non-negative, then

that point is feasible and is a vertex (the proof is left as an exercise); it is called a basic feasible solution. The variables that were set to 0 are called the non-basic variables, the rest are the basic variables. The set of basic variables is also called the basis. Note that if the value of a variable is non-zero then it is basic, but the converse is not true. When, in a basic feasible solution, a basic variable is zero, that solution (or basis) is said to be degenerate. The geometrical meaning of this is that each basis has an associated vertex, but a vertex can be associated to several bases (in the degenerate case).

If we denote by B the indices of the basic variables and N the indices of the non-basic variables, the program is in canonical form if $b \geq 0$ and the program is in the form:

$$\begin{aligned} \max \quad & c'_N x_N + c'_B x_B \\ \text{s.t.} \quad & A_N x_N + I x_B = b, \\ & x_N, x_B \geq 0. \end{aligned}$$

From a basic feasible solution (a vertex) and the problem in canonical form, the simplex algorithm chooses a non-basic variable that has a positive reduced cost, that is, a variable that, if increased, would increase the objective function. Then it increases the value of that variable as much as possible, without violating the non-negativity of the basic variables. That variable is made basic; (at least) one of the old basic variable becomes 0, and one becomes non-basic. The sequence of operations called a pivot (in the previous section) goes from the canonical form with respect to the old basis to the canonical form with respect to the new basis.

5 The simplex algorithm again (with better notation)

5.1 Simplex tableau notation

If we were going to program a computer to perform the manipulations we outlined in the previous sections, it would be convenient to represent the coefficients in matrix form. When expressed as a matrix, or **simplex tableau**, the linear program in (3) looks like

$$\begin{array}{cccccc|c} z & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & = \\ \hline & 2 & 3 & 1 & 1 & & & 5 \\ & 4 & 1 & 2 & & 1 & & 11 \\ & 3 & 4 & 2 & & & 1 & 8 \\ -1 & 5 & 4 & 3 & & & & 0 \end{array} \tag{12}$$

5.2 Tableau algorithm

In terms of simplex tableau, the algorithm for solving the problem above is:

1. Look at the last row of the tableau. Find a column in this row with a positive entry. This is the pivot column.

$$\begin{array}{c|c|c|c|c|c|c|c}
 z & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & = \\
 \hline
 & 2 & 3 & 1 & 1 & & & 5 \\
 & 4 & 1 & 2 & & 1 & & 11 \\
 & 3 & 4 & 2 & & & 1 & 8 \\
 -1 & 5 & 4 & 3 & & & & 0
 \end{array} \tag{13}$$

2. Among the rows whose entry r in the pivot column is positive, find the row that has the smallest ratio $\frac{s}{r}$, where s is this row's entry in the last column. This is the pivot row.

$$\begin{array}{c|c|c|c|c|c|c|c}
 z & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & = \\
 \hline
 & 2 & 3 & 1 & 1 & & & 5 \\
 \hline
 & 4 & 1 & 2 & & 1 & & 11 \\
 & 3 & 4 & 2 & & & 1 & 8 \\
 -1 & 5 & 4 & 3 & & & & 0
 \end{array} \tag{14}$$

3. Divide every entry of the pivot row by the entry in the pivot column.

$$\begin{array}{c|c|c|c|c|c|c|c}
 z & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & = \\
 \hline
 & 1 & \frac{3}{2} & \frac{1}{2} & \frac{1}{2} & & & \frac{5}{2} \\
 \hline
 & 4 & 1 & 2 & & 1 & & 11 \\
 & 3 & 4 & 2 & & & 1 & 8 \\
 -1 & 5 & 4 & 3 & & & & 0
 \end{array} \tag{15}$$

4. For every other row, subtract a multiple of the pivot row to make the entry in the pivot column zero.

$$\begin{array}{c|c|c|c|c|c|c|c}
 z & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & = \\
 \hline
 & 1 & \frac{3}{2} & \frac{1}{2} & \frac{1}{2} & & & \frac{5}{2} \\
 \hline
 & & -5 & & -2 & 1 & & 1 \\
 & & -\frac{1}{2} & \frac{1}{2} & -\frac{3}{2} & & 1 & \frac{1}{2} \\
 -1 & & -\frac{7}{2} & \frac{1}{2} & -\frac{5}{2} & & & -\frac{25}{2}
 \end{array} \tag{16}$$

5. Repeat steps one to four. If step one finds no pivot column, we've reached the optimum. If step two finds no positive ratio then z is unbounded.

5.3 Caveats

5.3.1 Cycling

If you have bad luck in step one, the algorithm could be in a cycle. This is pretty unlikely. If you use a consistent rule for deciding which of the positive entries to make the pivot column, like choose the positive entry with the smallest index (Bland, Robert G., “A combinatorial abstraction of linear programming”, *J. Combinatorial Theory*, Ser. B, 23 (1977), no. 1, 33–57.), it is impossible to cycle.

5.3.2 Initial feasible solution

According to the description of the simplex algorithm that we saw, the algorithm can be applied directly only to systems in canonical form. As in our examples, if the system starts (or can be transformed to) the form

$$\begin{aligned} \max \quad & c'x \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0, \end{aligned} \tag{17}$$

with $b \geq 0$, then the introduction of slack variables will leave the system in canonical form. The only non-trivial situation occurs when the system is in the form (17), but b has some negative components. In this case, one adds an auxiliary variable x_0 and consider the problem:

$$\begin{aligned} \min \quad & x_0 \\ \text{s.t.} \quad & Ax - x_0 \leq b \\ & x, x_0 \geq 0, \end{aligned} \tag{18}$$

After adding slack variables, one pivots at variable x_0 and the row associated to the minimum entry of b . This will leave the system in canonical form. Then one solves the problem (18) with the simplex algorithm. Consider now the situation when the algorithm finishes. If $x_0 > 0$, then the original problem (17) is infeasible. Else, $x_0 = 0$. In this case, if x_0 is basic, then we perform one more pivot operation to make it non-basic. Now that $x_0 = 0$ and non-basic, we stop considering that column in the tableau and replace the reduced costs by the cost function of (17). This is precisely a canonical form of the problem (17), so that we can apply simplex to it.

Example:

$$\begin{aligned}
 & \max -2x_1 - 3x_2 \\
 \text{s.t.} \quad & -x_1 - x_2 \leq -3 \\
 & 2x_1 - x_2 \leq -2 \\
 & x_1, x_2 \geq 0
 \end{aligned}$$

The initial tableau after adding the auxiliary variable x_0 and the slack variables x_3, x_4 is

x_0	x_1	x_2	x_3	x_4	
-1	-1	-1	1		-3
-1	2	-1		1	-2
-1					

After pivoting with respect to the first row and first column we get

x_0	x_1	x_2	x_3	x_4	
1	1	1	-1		3
	3		-1	1	1
	1	1	-1		3

This tableau is in canonical form.

6 A proof of strong duality, based on the simplex algorithm

The strong duality theorem states:

$$\begin{aligned}
 \max \sum_{j=1}^n c_j x_j, \text{ subject to} \quad & \min \sum_{i=1}^m b_i y_i, \text{ subject to} \\
 \sum_{j=1}^n a_{ij} x_j \leq b_i & = \quad \sum_{i=1}^m a_{ij} y_i \geq c_j \\
 x_j \geq 0 & \quad y_i \geq 0
 \end{aligned} \tag{19}$$

We proved it already using Farkas' Lemma. Now we will prove it again using ideas borrowed from the simplex algorithm.

Weak duality proved

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i \tag{20}$$

for any feasible values of x_j 's and y_i 's, so to prove strong duality it is sufficient to exhibit feasible x_j 's and y_i 's where the sums are equal.

Given a linear program,

$$\text{maximize } \sum_{j=1}^n c_j x_j, \text{ subject to } \begin{aligned} \sum_{j=1}^n a_{ij} x_j &\leq b_i, \\ x_j &\geq 0, \end{aligned} \quad (21)$$

let x_1^*, \dots, x_n^* be a feasible solution that maximizes the objective function, and let $z^* = \sum_{j=1}^n c_j x_j^*$ be the value of the maximum.

We introduce slack variables,

$$x_{n+i} = b_i - \sum_{j=1}^n a_{ij} x_j, \quad (22)$$

to convert the standard form linear program to the equality form we use in the simplex algorithm.

Because z^* is the maximum feasible value of the objective function we can write the value of the objective function for any feasible solution as

$$z = z^* + \sum_{k=1}^{n+m} \bar{c}_k x_k \quad (23)$$

for some $\bar{c}_k \leq 0$.

Let

$$y_i^* = -\bar{c}_{n+i}. \quad (24)$$

We will show that $z^* = \sum_{i=1}^m b_i y_i^*$ and $\sum_{i=1}^m a_{ij} y_i^* \geq c_j$. Since $y_i^* \geq 0$ by construction, this will prove that a feasible dual solution is equal to a feasible primal solution.

Begin with z :

$$z = \sum_{j=1}^n c_j x_j = z^* + \sum_{k=1}^{m+n} \bar{c}_k x_k. \quad (25)$$

We break up the sum

$$z = z^* + \sum_{j=1}^n \bar{c}_j x_j + \sum_{k=n+1}^{n+m} \bar{c}_k x_k. \quad (26)$$

Substituting (24) to remove the \bar{c} 's in the second sum gives

$$z = z^* + \sum_{j=1}^n \bar{c}_j x_j + \sum_{i=1}^m -y_i^* x_{n+i}. \quad (27)$$

Substituting (22) to remove the slack variables gives

$$z = z^* + \sum_{j=1}^n \bar{c}_j x_j + \sum_{i=1}^m -y_i^* (b_i - \sum_{j=1}^n a_{ij} x_j). \quad (28)$$

Regrouping, reversing the order of the double sum, and regrouping some more, we have

$$z = (z^* - \sum_{i=1}^m b_i y_i^*) + \sum_{j=1}^n (\bar{c}_j + \sum_{i=1}^m a_{ij} y_i^*) x_j. \quad (29)$$

Note that all the previous manipulations are just a rewriting of the objective function, and are valid for any $x \in \mathbb{R}^n$. So, it is true for $x_j = 0$. Therefore,

$$z^* = \sum_{i=1}^m b_i y_i^*, \quad (30)$$

which finishes half of the proof.

To see that the y_i 's are a feasible solution, we plug the value of z^* from (30) into (29) and set this equal to the original formula (25).

$$z = \sum_{j=1}^n (\bar{c}_j + \sum_{i=1}^m a_{ij} y_i^*) x_j = \sum_{j=1}^n c_j x_j. \quad (31)$$

These sums can be equal only if the coefficients of the x_j 's are all equal (again, because this equality is true for any $x \in \mathbb{R}^n$):

$$c_j = \bar{c}_j + \sum_{i=1}^m a_{ij} y_i^*. \quad (32)$$

But each $\bar{c}_j \leq 0$, so

$$\sum_{i=1}^m a_{ij} y_i^* \geq c_{ij}. \quad (33)$$

This completes the proof of strong duality.