Discrete Optimization

Guoli Ding
http://math.lsu.edu/~ding

Last Updated: November 8, 1999
Table of Contents

Lectures

An Overview ........................................................................................................... 1
Linear Programming ................................................................................................. 2
  Introduction ........................................................................................................... 2
  Examples of Linear Programming Problems ......................................................... 3
  Linear Algebra Background ................................................................................... 4
  Fourier-Motzkin Elimination ............................................................................... 6
  The Geometry of Linear Programming ................................................................ 8
  The Simplex Method ............................................................................................... 9
  Representation of Polyhedra ............................................................................... 14
  Network Flow Problems ..................................................................................... 15
  Graphs and Digraphs ........................................................................................... 15
  Network Flow Problems ..................................................................................... 17

Integer Programming
  Introduction
  Totally Unimodular Matrices

Homework

  Set 1 .................................................................................................................. 20
  Set 2 .................................................................................................................. 22
  Set 3

Index ..................................................................................................................... 24
Lecture 1. August 24

This is the first class of this course. We start with some information about the course.

<table>
<thead>
<tr>
<th>INSTRUCTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: Guoli Ding</td>
</tr>
<tr>
<td>Office: 220 Lockett Hall</td>
</tr>
<tr>
<td>Phone number: 388-1671</td>
</tr>
<tr>
<td>E-mail: <a href="mailto:ding@math.lsu.edu">ding@math.lsu.edu</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OFFICE HOURS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:30 – 11:30 TuTh and by appointment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TEXT BOOK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Linear Optimization, by D. Bertsimas and J. N. Tsitsiklis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GRADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students’ grades are determined by three sets of homework.</td>
</tr>
</tbody>
</table>

The next is an overview on the materials that will be covered this semester.

The main issue of discrete optimization is to develop efficient algorithms to solve the following optimization problem

Maximize \[ c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \]
subject to \[ a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n \leq b_1 \]
\[ a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n \leq b_2 \]
\[ \vdots \]
\[ a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n \leq b_m \]
\[ x_1, x_2, ..., x_n \text{ are integral.} \] (1)

That is, among all vectors \( x = (x_1, x_2, ..., x_n)^T \) that satisfy (1), we want to either find one for which \( c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \) is maximized, or report so if no such \( x \) exists. This problem is often referred as the integer programming problem.

There is no known efficient algorithm that solves a general integer programming problem, and most researchers believe that no such algorithm exists. What available today are algorithms that are either not so efficient or not so accurate. For all these approaches, there is one thing in common, they all base on the theory of linear programming. When the integrality constraints are dropped from (1), then we call it a linear programming problem. Unlike the integer programming problem, there are several “efficient” algorithms that solve the linear programming problems. These algorithms include the simplex method, the ellipsoid method, and the interior point method. In this course, we will concentrate on the simple method, because it is the most widely used algorithm. We will also discuss the main ideas of the other two methods because they are theoretically better than the simplex method.

Even though we do not know how to solve the general integer programming problem efficiently, we do know some special cases of it for which there are efficient algorithms available. One of the best solved cases is the network flow problem. This problem has enormous applications in many areas and it has a lot of generalizations. We will cover this problem in detail in this course.

It should be emphasized that just having an algorithm is not enough for us, what really matters is to have an “efficient” algorithm. The following example illustrates the dramatic difference of two different algorithms. By definition, the determinant of an \( n \times n \) matrix \( A = (a_{ij}) \) is

\[
\sum_{\pi} (-1)^{\pi_1}a_{1\pi(1)}a_{2\pi(2)}...a_{n\pi(n)}
\]

where the sum is taken over all permutations \( \pi \) of \( \{1, 2, ..., n\} \). If the definition is used to compute the
determinate of $A$, it takes at least $n! \cdot n$ additions and multiplications. However, if we use Gaussian elimination, it takes less than $\frac{2}{3} n^3$ additions and multiplications. To see the difference of these two algorithms, take $n = 50$. Suppose we use a computer that performs 100,000 arithmetic operations per second. Then Gaussian elimination computes $|A|$ in 0.8 seconds, while using the definition takes $4.8 \cdot 10^{53}$ years!

The next is an introduction to linear programming. We start with an example which shows how a real world problem can be formulated as a linear programming problem. Suppose there are 10 workers and 10 jobs. It is known that worker $i$ works on job $j$ will produce a profit of $p_{ij}$. The question is how to assign the 10 jobs to the workers, one job a worker, so that the total profit is maximized.

Let us introduce 100 variables $x_{ij}$ for $i, j = 1, 2, \ldots, 10$, where $x_{ij} = 1$ is interpreted as assigning job $j$ to worker $i$. Then the following optimization problem

$$
\text{Max} \sum_{i,j} p_{ij} x_{ij}
$$

s.t. \hspace{0.5cm} \sum_{i=1}^{10} x_{ij} = 1 \quad \text{for } j = 1, 2, \ldots, 10

\sum_{j=1}^{10} x_{ij} = 1 \quad \text{for } j = 1, 2, \ldots, 10

$$x_{ij} \geq 0 \quad \text{for } i, j = 1, 2, \ldots, 10$$

has an optimal solution for which each $x_{ij}$ equals either 0 or 1. We will prove the existence of this optimal solution and thus the assignment problem will be solved.

In the following, we use a small example to illustrate some ideas on how to solve linear programming problems. Consider

$$
\text{Max} \quad 3x + 2y \\
\text{s.t.} \quad 2x + 3y \leq 12, \\
\quad \quad \quad \quad \quad x - y \geq 5, \\
\quad \quad \quad \quad \quad y \leq 3.
$$

Observe that an equation $ax + by = c$ defines a line and $ax + by \leq c$ defines a half-plane (WHY?). Thus the feasible region $R$ of this problem—the set of points $(x, y)$ that satisfy all three inequalities, is the intersection of the three half-planes. For each given scalar $z$, the set of points that satisfy $3x + 2y = z$ is a line. Different values of $z$ lead to different lines, all of them parallel to each other. In particular, increasing $z$ corresponds to moving the line $3x + 2y = z$ along the direction of the vector $(3, 2)$. Since we are interested in maximizing $3x + 2y$, we would like to move the line $3x + 2y = z$ as far as possible, provided the line meets $R$. It is easy to see that the best we can do is to take $z = 17$ and $(x, y) = (5.4, 0.4)$. Notice that the optimal solution $(5.4, 0.4)$ is a “corner” point of $R$. We will show that, in general, the optimal solution can always be chosen as a corner point.

The way the simplex method works is to start with a corner point and then move to a better corner point until it reaches the optimal solution. In contrast, the ellipsoid method and the interior point method start with an interior point of the feasible region and then move to another interior point until it “reaches” the optimal solution. In the worst senior, the simplex method may have to exhaust all corner points before reaching the optimal solution and that takes a very long time. However, for “ordinary” problems, the simplex method works very fast. By comparing the worst cases, the ellipsoid method and interior point method are better then the simplex method. But when dealing with real world problems, that is, when comparing the “ordinary” cases, the simple method and the interior point method are better. The ellipsoid method is important because it is the first theoretically efficient algorithm that solves the linear programming problem and it has many theoretical implications.

As in all linear programming courses, the first thing we do is to see some examples on how to formulate
a linear programming problem.

**Product-Mix Problem.** A company can manufacture $n$ different products using $m$ different resources. The company has access to a maximum of $b_j$ units of resource $j$, where $j = 1, 2, ..., m$. One unit of product $i$ requires $a_{ij}$ units of resource $j$ and results in a revenue of $c_i$ dollars. Formulate a “product-mix” that maximizes the revenue subject to the resource constraints.

Let $x_i$ be number of units of product $i$ to be manufactured. Then the problem is to maximize $c_1x_1 + c_2x_2 + ... + c_nx_n$. There are two types of restrictions. The first is the resource limitation constraint, $a_{1j}x_1 + a_{2j}x_2 + ... + a_{nj}x_n \leq b_j$, for all $j$, and the second is the nonnegativity constraint, $x_i \geq 0$, for all $i$. So we have

$$\text{Max } c_1x_1 + c_2x_2 + ... + c_nx_n$$

subject to

$$a_{1j}x_1 + a_{2j}x_2 + ... + a_{nj}x_n \leq b_j \text{ for } j = 1, 2, ..., m$$

$$x_i \geq 0 \text{ for } i = 1, 2, ..., n$$

**Lecture 2. August 26**

**Racial-Desegregation Plan ("Busing Plan").** A school district is divided into $n$ different neighborhoods, each with its own high school. The student population of the entire district divided into the following three racial categories – whites, blacks and “others”. Let $w_i, b_i, o_i$ be the number of white, black and other students in neighborhood $i$. There is a limit of $C_i$ to the number of students that the school in neighborhood $i$ can accommodate. Further, there is a school desegregation order, which stipulates that the percentage of students from any racial category in any neighborhood school must not differ from the corresponding percentage for the entire district by more that 10%. Formulate a linear programming problem which finds a (permissible) school assignment for all the students that minimizes the sum of the distance traveled by all students, assuming that any student from neighborhood $i$ who is assigned to school $j$ has to travel a distance $d_{ij}$.

Let $x_{ij}$, $y_{ij}$, and $z_{ij}$ be white, black, and other students from neighborhood $i$ to school $j$. Then

Minimize $\sum_{i,j} d_{ij}(x_{ij} + y_{ij} + z_{ij})$

subject to

$\sum_j x_{ij} = w_i, \sum_j y_{ij} = b_i, \sum_j z_{ij} = o_i$ for all $i$

$\sum_i (x_{ij} + y_{ij} + z_{ij}) \leq C_j$ for all $j$

$p_w - 0.1 \leq \frac{\sum_i x_{ij}}{w_i} \leq p_w + 0.1$ for all $j$

$p_b - 0.1 \leq \frac{\sum_i y_{ij}}{b_i} \leq p_b + 0.1$ for all $j$

$p_o - 0.1 \leq \frac{\sum_i z_{ij}}{o_i} \leq p_o + 0.1$ for all $j$

$x_{ij}, y_{ij}, z_{ij} \geq 0$ for all $i, j$

where $p_w = T_w/T$, $p_b = T_b/T$, $p_o = T_o/T$, and $T_w = \sum_i w_i$, $T_b = \sum_i b_i$, $T_o = \sum_i o_i$, $T = T_w + T_b + T_o$. 

3
LINEAR REGRESSION. Given \( n \) data points \((a_i, b_i), i = 1, 2, \ldots, n\) in the plane, we want to find a line that “best fits” the given data. In this instance, we want to find a linear function \( f(x) \) which minimizes the maximum absolute error \( \max_i \{|f(a_i) - b_i|\} \). Formulate a linear program that would accomplish this.

Let \( f(x) = mx + c \) and let \( z \) be a new variable such that \( z \geq |ma_i + c - b_i| \). Then we have

\[
\begin{align*}
\text{Minimize} & \quad z \\
\text{subject to} & \quad -(ma_i + c - b_i) \leq z \leq ma_i + c - b_i \quad \text{for all } i
\end{align*}
\]

Notice that this is a Linear programming problem on \( z, m, \) and \( c \).

PROBLEMS INVOLVING ABSOLUTE VALUES. Consider the problem of minimizing \( c_1|x_1| + c_2|x_2| + \cdots + c_n|x_n| \), subject to \( a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b \), where we assume \( c_i \geq 0 \) for all \( i \). Formulate this as a linear programming problem.

There are several standard methods that deal with this kind of problems. The following uses the observation that \( |x| \) is the smallest number \( z \) that satisfies \( z \geq x \) and \( z \geq -x \). Let us introduce \( n \) new variables \( z_1, z_2, \ldots, z_n \) and

\[
\begin{align*}
\text{Minimize} & \quad \sum_{i=1}^{n} c_i z_i \\
\text{subject to} & \quad a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b \\
& \quad z_i \geq x_i \quad \text{and} \quad z_i \geq -x_i \quad \text{for all } i.
\end{align*}
\]

Before we continue, we provide some linear algebra background that will be used latter.

We call \( r \) an \( n \)-dimensional row vector if

\[
r = [r_1, r_2, \ldots, r_n]
\]

for some real numbers \( r_1, r_2, \ldots, r_n \), and \( c \) an \( n \)-dimensional column vector if

\[
c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}
\]

for some real numbers \( c_1, c_2, \ldots, c_n \). If \( r_i = c_i \) for all \( i \), then we call \( r \) the transpose of \( c \), writing \( r = c' \), and \( c \) the transpose of \( r \), writing \( c = r' \). We denote by \( \mathbb{R}^n \) the set of all \( n \)-dimensional column vectors.

Let \( \alpha \) and \( \beta \) be real numbers and let

\[
x = [x_1, x_2, \ldots, x_n] \quad \text{and} \quad y = [y_1, y_2, \ldots, y_n].
\]

Then we define

\[
\alpha x + \beta y = [\alpha x_1 + \beta y_1, \alpha x_2 + \beta y_2, \ldots, \alpha x_n + \beta y_n]
\]

called a linear combination of \( x \) and \( y \). If \( x \) and \( y \) are column vectors of the same dimension, then we define \( \alpha x + \beta y \) to be the transpose of \( \alpha x' + \beta y' \). Similarly, we can define the linear combination of several vectors of the same dimension.

If \( r \) and \( c \) are vectors defined above, then we define their inner product \( rc \) as:

\[
r_1c_1 + r_2c_2 + \cdots + r_nc_n.
\]

We call two vectors \( x \) and \( y \) in \( \mathbb{R}^n \) perpendicular if \( x'y = y'x = 0 \). A vector \( x \in \mathbb{R}^n \) is a unit vector if \( x'x = 1 \).
Let $A$ be an $m \times n$ matrix
\[
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{bmatrix}
\]
Then we can view $A$ as either $m$ row vectors $R_1, R_2, \ldots, R_m$ or $n$ column vectors $C_1, C_2, \ldots, C_n$.

Therefore, for each $x \in \mathbb{R}^n$, we can define $Ax$ as an $m$-dimensional column vector by either
\[
Ax = \begin{bmatrix} R_1 x \\ R_2 x \\ \vdots \\ R_m x \end{bmatrix} \quad \text{or} \quad Ax = x_1 C_1 + x_2 C_2 + \cdots + x_n C_n.
\]
Notice that these two definitions are equivalent. Similarly, for each $y \in \mathbb{R}^m$, we define $y'A$ as
\[
y'A = y_1 R_1 + y_2 R_2 + \cdots + y_m R_m \quad \text{or equivalently} \quad y'A = [y'C_1, y'C_2, \ldots, y'C_n].
\]
In general, if $B$ is an $n \times t$ matrix with columns $D_1, D_2, \ldots, D_t$, then we define $AB$ as
\[
AB = [AD_1, AD_2, \ldots, AD_t] \quad \text{or equivalently} \quad AB = \begin{bmatrix} R_1 D \\ R_2 D \\ \vdots \\ R_m D \end{bmatrix}.
\]
We also consider three elementary operations on matrices.
(i) interchanging two rows;
(ii) multiply a row by a scaler;
(iii) adding a multiple of a row to another row.
An $n \times n$ matrix $A = (a_{ij})$ is called an identity matrix and denoted by $I$ if $a_{ii} = 1$ for all $i$ and $a_{ij} = 0$ for all $i \neq j$. The results of applying one elementary operation to the identity matrix are called elementary matrices. It is not difficult to see that applying an elementary operation to a matrix $A$ is the same as multiplying the corresponding elementary matrix to the left of $A$. Therefore, Gaussian elimination can be stated as the following.

Gaussian elimination. If $b \in \mathbb{R}^n$ is nonzero, then there exists a matrix $Q$, which is a product of elementary matrices, such that $Qb$ is the transpose of $[1, 0, 0, \ldots, 0]$.

A square matrix $A$ is called invertible or nonsingular if there this a matrix $B$ such that $AB = BA = I$. Such a matrix $B$ is called the inverse of $A$ and denoted by $A^{-1}$.

Theorem. (a) All elementary matrices are invertible.
(b) If both $A$ and $B$ are invertible, then so it $AB$ and $(AB)^{-1} = B^{-1}A^{-1}$.
(c) A matrix is invertible if and only if it can be expressed as the product of elementary matrices.

The rank of a matrix is defined to be an integer $r$ such that the largest nonsingular submatrix has size $r \times r$. For a collection of vectors $x_1, x_2, \ldots, x_k$, we say that they are linearly dependent if there exist real numbers $a_1, a_2, \ldots, a_k$, not all of them zero, such that $a_1 x_1 + a_2 x_2 + \cdots + a_k x_k = 0$; otherwise, they
are called *linearly independent*. The *rank* of a set of vectors is the largest integer $r$ such that the set has $r$ linearly independent members.

**Theorem.** Let $A$ be a matrix. Then the following are equivalent:

(a) The rank of $A$ is $r$,
(b) The rank of the columns of $A$ is $r$,
(c) The rank of the rows of $A$ is $r$.

**Theorem.** Let $A$ be an $m \times n$ matrix of rank $r$. Then there are pairwise perpendicular unit vectors $x_1, x_2, ..., x_{n-r}$, such that every solution of $Ax = 0$ is a linear combination of $x_1, x_2, ..., x_{n-r}$.

**Theorem.** Let $A$ be an $n \times n$ matrix. Then the following are equivalent:

(a) $A$ is invertible,
(b) The determinant of $A$ is nonzero,
(c) The rows of $A$ are linearly independent,
(d) The columns of $A$ are linearly independent,
(e) For each $b \in \mathbb{R}^n$, the equation $Ax = b$ has a unique solution,
(f) For some $b \in \mathbb{R}^n$, the equation $Ax = b$ has a unique solution.

**Lecture 3.** August 31

Using the terminology of matrices, the linear programming problem can be stated as:

Maximize $c'x$
subject to $Ax \leq b$

An algorithm is said to be capable of solving this problem means that, for any given $A$, $b$, and $c$, the algorithm either outputs a vector such that it is an optimal solution, or returns a message indicating that no optimal solution exists. Observe that the corresponding feasibility problem, the problem of deciding if there exists $x$ with $Ax \leq b$, can be solved as an optimization problem by taking $c = 0$. It is also true that the optimization problem can be transformed into a feasibility problem (we will see this in a few weeks) and thus the optimization problem is “equivalent” to the feasibility problem.

Given an $m \times n$ matrix $A$ and a vector $b$ in $\mathbb{R}^n$. We say that $Ax \leq b$ is feasible (or consistent in some books) if there is at least one vector $x$ in $\mathbb{R}^n$ that satisfies the inequality. Next, we see how we can solve the following:

**Problem.** Determine if $Ax \leq b$ is feasible, and find a vector $x$ if it is.

We first see how we can solve the problem when $n = 1$, that is, there is only one variable, say $x$. We illustrate the idea using the following example (where $m = 4$): $3x \leq 5$, $0x \leq 1$, $-2x \leq 7$, $-x \leq -1$. Rewrite these four inequalities as $x \leq 5/3$, $0 \leq 1$, $-2/7 \leq x$, $1 \leq x$. It follows that $x$ satisfies the four given inequalities if and only if $1 \leq x \leq 5/3$. In general, all $m$ inequalities $a_i x \leq b_i$ can be partitioned into three groups: those with $a_i > 0$, those with $a_i = 0$, and those with $a_i < 0$. Just like in the example, it is not difficult to see that there is a solution if and only if $0 \leq b_i$ for all $i$ with $a_i = 0$ and $\max\{ \frac{b_i}{a_i} : a_i < 0 \} \leq \min\{ \frac{b_i}{a_i} : a_i > 0 \}$. Observe that the last inequality is equivalent to $\frac{b_i}{a_i} \leq \frac{b_j}{a_j}$ for all choices of $i$ and $j$ with $a_i < 0$ and $a_j > 0$.

Before we get into the general case, let us consider the following example, where $m = 3$, $n = 2$.

$$x - y \leq 1, \quad y \leq 3, \quad -x - y \leq -5$$ (1)
Rewritten (1) as \[ x \leq 1 + y, \quad 0x \leq 3 - y, \quad -x \leq -5 + y \] (2)

If we consider \( x \) as the only variable of (2), then we conclude that (2) is feasible (which is equivalent to (1) being feasible) if and only if \[ 0 \leq 3 - y, \quad 5 - y \leq 1 + y \] is feasible. Since (3) has only one variable, we can solve it using the idea we discussed above and get, say \( y = 2 \) (which is not unique). Then from (2) we deduce that (1) is feasible and \((x, y) = (3, 2)\) is a solution.

In general, we can use the same idea to reduce the number of variables without changing the feasibility of the inequalities. If the set of given inequalities is feasible, then we can keep reducing the number of variables until we get only one variable which we can solve. From this solution we can trace back to get a solution of the original inequalities. This method is very similar to Gaussian elimination and is known as Fourier-Motzkin elimination.

Next we will analyze Fourier-Motzkin elimination in the general case. Suppose the given inequality is \( Ax \leq b, \) where \( A = (a_{i,j}) \) is an \( m \times n \) matrix, \( x \in \mathbb{R}^n, \) and \( b \in \mathbb{R}^m. \)

Suppose we want to eliminate variable \( x_n. \) Then we can partition the inequalities into three groups: those with \( a_{i,n} > 0, \) those with \( a_{i,n} = 0, \) and those with \( a_{i,n} < 0. \) In another words, we can partition \( \{1, 2, \ldots, m\} \) into three subsets: \( I^+ = \{i : a_{i,n} > 0\}, \) \( I^0 = \{i : a_{i,n} = 0\}, \) and \( I^- = \{i : a_{i,n} < 0\}. \) If both \( I^+ \) and \( I^- \) are empty, that is, every entry of the last column of \( A \) is zero, then we do not need to do anything since \( x_n \) is already eliminated. If exactly one of \( I^+ \) and \( I^- \) is empty, then \( Ax \leq b \) is feasible if and only if \( A_0 x \leq b_0, \) which consists of inequalities corresponding to those \( i \in I_0, \) is feasible (WHY?).

If both \( I^+ \) and \( I^- \) are not empty, then it is easy to see that \( Ax \leq b \) is feasible if and only if

\[ a_{i,1} x_1 + a_{i,2} x_2 + \ldots + a_{i,n-1} x_{n-1} \leq b_i \quad \text{for all } i \in I^0; \] and

\[ (b_i - a_{i,1} x_1 - a_{i,2} x_2 - \ldots - a_{i,n-1} x_{n-1})/a_{i,n} \leq (b_h - a_{h,1} x_1 - a_{h,2} x_2 - \ldots - a_{h,n-1} x_{n-1})/a_{h,n} \quad \text{for all } i \in I^- \text{ and } h \in I^+ \] (2)

is feasible. Observe that (2) can be rewritten as

\[
\left(\frac{a_{i,1}}{a_{h,n}} + \frac{a_{i,2}}{a_{h,n}} + \ldots + \frac{a_{i,n-1}}{a_{h,n}}\right)x_1 + \left(\frac{a_{h,2}}{a_{h,n}} + \frac{a_{i,2}}{a_{h,n}} + \ldots + \frac{a_{i,n-1}}{a_{h,n}}\right)x_2 + \ldots + \left(\frac{a_{h,n-1}}{a_{h,n}} + \frac{a_{i,n-1}}{a_{h,n}}\right)x_{n-1} \leq \frac{b_i}{a_{h,n}} + \frac{b_h}{a_{h,n}}
\] (3)

which is exactly the sum of the \( h \)th and the \( i \)th inequality of \( Ax \leq b \) divided by \( a_{h,n} \) and \(-a_{i,n},\) respectively. Therefore, if we write inequalities (1) and (2) in a matrix form \(Cx \leq d,\) then the last column of \( C\) consists of 0’s, since \( x_n \) has been eliminated, and \( C \) has \(|I^0| + |I^+| \cdot |I^-|\) rows. In addition, the reformulation (3) implies that there is a matrix \( P \) with nonnegative entries such that \( C = PA \) and \( d = Pb \) (WHY?). In conclusion, we have shown that

**Proposition.** For any given \( A \) and \( b, \) there exists a nonnegative matrix \( P \) such that the last column of \( PA \) is zero and \( Ax \leq b \) is feasible if and only if \( PAx \leq Pb \) is feasible.

**Lecture 4.** September 2

If we apply this proposition \( n \) times, we have the following conclusion.

**Theorem.** For any given \( A \) and \( b, \) there exists a nonnegative matrix \( P \) with \( PA = 0 \) such that \( Ax \leq b \) is feasible if and only if \( Pb \geq 0. \)

Very often, we use the following form of the last theorem.

**Theorem.** \( Ax \leq b \) is infeasible if and only if there exists a nonnegative vector \( y \in \mathbb{R}^m \) such that \( y^T A = 0 \) and \( y^T b < 0. \)

The following looks more general but can be proved by using the last result.
Theorem. The set of inequalities
\[ Ax + By = e, \quad Cx + Dy \leq f, \quad \text{and} \quad x \geq 0 \]
is infeasible if and only if there exists row vectors \( u \) and \( v \) such that
\[ uA + vC \geq 0, \quad uB + vD = 0, \quad v \geq 0, \quad \text{and} \quad ue + vf < 0. \]

Corollary. Let \( A \) be an \( m \times n \) matrix and let \( b \in \mathbb{R}^m \). Then \( Ax = b \) has no solution if and only if there exists \( y \in \mathbb{R}^m \) such that \( y'A = 0 \) and \( y'b \neq 0 \).

The following is known as the duality theorem of linear programming.

Duality Theorem. Consider the following two optimization problems:

Minimize \( c'x \) \hspace{1cm} Maximize \( y'b \)
subject to \( Ax = b \) \hspace{1cm} subject to \( y'A \leq c' \)
\( x \geq 0 \) \hspace{1cm} \( x \geq 0 \)

1. If \( y'A \leq c' \) is feasible but \( Ax = b, x \geq 0 \) is infeasible, then the maximum of \( y'b \) is \( +\infty \).
2. If \( Ax = b, x \geq 0 \) is feasible but \( y'A \leq c' \) is infeasible, then the minimum of \( c'x \) is \( -\infty \).
3. If both \( y'A \leq c' \) and \( Ax = b, x \geq 0 \) are feasible then \( \text{Min} \ c'x = \text{Max} \ y'b \).

Proof. 1 and 2 follow from the last theorem. To prove 3, we make the following observation. For any \( x \in \mathbb{R}^n, y \in \mathbb{R}^m \) such that \( Ax = b, x \geq 0 \), and \( y'A \leq c' \), we always have \( y'b = y'(Ax) = (y'A)x \leq c'x \).

Thus, to prove 3, it is enough to prove that \( y'b \geq c'x \). \( Ax = b, x \geq 0 \), and \( y'A \leq c' \) have a solution, which can be shown using the last theorem.

Remarks. 1. It is possible that both \( \{y'A \geq c'\} \) and \( \{Ax = b, x \geq 0\} \) are infeasible. For example, when \( m = n = 1, A = 0, b = 1 \) and \( c = -1 \).

2. Each time we apply Fourier-Motzkin elimination, one variable is eliminated, but the total number of inequalities can go from \( m \) to as many as \( m^2/4 \). Thus eliminating \( k \) variables may increase the number of inequalities from \( m \) to \( 4(m/4)^2 \), which is a very high price to pay for eliminating \( k \) variables. Therefore, in most of the time, this is not the best algorithm for solving the feasibility problem. However, as we have seen, its theoretical implications are very powerful. For instance, we conclude from the above Duality theorem that the optimization problem
\[ \max \{y'b : y'A \leq c'\} \]
can be solved as a feasibility problem
\[ y'A \leq c', \quad Ax = b, \quad x \geq 0, \quad \text{and} \quad y'b = c'x. \]

Lecture 5. September 7

Today, we consider the geometry of linear programming.

Let \( A \) be an \( m \times n \) matrix and \( b \in \mathbb{R}^m \). Then the set of vectors \( x \in \mathbb{R}^n \) that satisfy \( Ax \geq b \) is called a polyhedron. Our first observation about a polyhedron is its convexity. A set \( S \) of vectors is convex if for any \( x, y \in S \), and any \( \lambda \in [0, 1] \), we have \( \lambda x + (1 - \lambda)y \in S \). Geometrically, it means that any line segment with it two ends in \( S \) must be contained in \( S \) completely.

Observation. Every polyhedron is convex.

We have noticed from some 2-dimensional examples that an optimal solution to a linear programming problem tends to occur at a “corner” point. We now introduce three ways to capture the properties of a “corner” point and then prove that all of them are equivalent.

Let \( P \) be a polyhedron and let \( x^* \in P \). We call \( x^* \) an extreme point of \( P \) if there are no \( y, z \in P \), both different from \( x^* \), such that \( x = \frac{1}{2}y + \frac{1}{2}z \). Geometrically, what it says is that \( x^* \) is not the middle point
of any nontrivial segment that is contained in $P$. We call $x^*$ a vertex of $P$ if there exists $c \in \mathbb{R}^n$ such that $c^* x < c^* x$ for all $x \in P$ with $x \neq x^*$. This definition says that $P$ is on one side of the hyperplane \{x : cx = c^* x\} which meets $P$ only at the point $x^*$.

Let $P = \{x \in \mathbb{R}^n : Ax \geq b\}$ be a polyhedron and let $x^* \in \mathbb{R}^n$. Let the rows of $A$ be $R_1, R_2, ..., R_m$ and $A_1$ be the submatrix of $A$ that consists of rows $R_i$ for which $R_i x^* = b_i$. We call $x^*$ a basic solution if the rank of $A_1$ is $n$. If a basic solution is in $P$, then it is called a basic feasible solution. In addition, every $x$ in $P$ is called a feasible solution.

**Theorem.** Let $x^*$ be a vector in a polyhedron $P$. Then the following are equivalent:

(a) $x^*$ is a vertex.
(b) $x^*$ is an extreme point.
(c) $x^*$ is a basic feasible solution.

**Corollary 2.1.** A polyhedron can have only a finite number of vertices.

We next prove that the “corner” points as described above are exactly what we are looking for. That is, every optimal value can always be obtained at a “corner” point. But first we need to realize that not every polyhedron has a vertex. To understand which polyhedron has at least one vertex, we have the following definition. A polyhedron $P$ contains a line if there exists a point $x$ in $P$ and a nonzero vector $d$ such that $x + \lambda d \in P$ for all real numbers $\lambda$.

**Lecture 6.** September 9

**Theorem.** Suppose that a polyhedron $P = \{x : Ax \geq b\}$ is not empty. Then the following are equivalent:

(a) $P$ has at least one vertex.
(b) $P$ does not contain a line.
(c) The rank of $A$ is $n$.

**Theorem.** Consider $\min \{c^* x : x \in P\}$ where $P = \{x : Ax \geq b\}$. If the minimum is not $-\infty$ and $P$ has at least one vertex, then there exists an optimal solution which is a vertex of $P$.

Now we begin to discuss simplex method. We know from Homework 1 that every linear programming problem can be formulated in the following standard form:

\[
\begin{align*}
\text{Minimize} & \quad c^* x \\
\text{subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

(1)

We will use this form throughout the discussion. As in our textbook, the columns of $A$ are denoted by $A_1, A_2, ..., A_n$, the rows of $A$ are denoted by $a_1, a_2, ..., a_m$, and $\{x : Ax = b, x \geq 0\}$, the set of feasible solutions, is denoted by $P$. First, let us observe that we can always assume that the rank of $A$ equals $m$.

**Observation.** Suppose $P \neq \emptyset$. If the rank of $A$ is $k$, $a_{i_1}, a_{i_2}, ..., a_{i_k}$ are linearly independent, and $Q = \{x : a_{i_1} x = b_{i_1}, a_{i_2} x = b_{i_2}, ..., a_{i_k} x = b_{i_k}, x \geq 0\}$, then $P = Q$.

From now on, we will assume, unless otherwise stated, that the rank of $A$ in (1) is $m$.

**Lecture 7.** September 14

Suppose the $i_1$th, $i_2$th, ..., and $i_k$th columns of $A$ form a matrix $B$. Then, for each vector $x \in \mathbb{R}^n$, we denote by $x_B$ the subvector of $x$ that consists of the $i_1$th, $i_2$th, ..., and $i_k$th entries of $x$. 
Observation. A vector $x \in P$ is a basic feasible solution if and only if there is an $m \times m$ invertible submatrix $B$ of $A$ such that $x_B = B^{-1}b \geq 0$ and $x_i = 0$ for all other entries.

A submatrix $B$ as described in this observation will be called a basis matrix, and $x$ the basic feasible solution associated with $B$. In addition, if $B$ consists of $A_{i_1}, A_{i_2}, ..., A_{i_m}$, then these columns are called basic columns, $x_{i_1}, x_{i_2}, ..., x_{i_m}$ are called basic variables and $i_1, i_2, ..., i_m$ are called basic indices. These indices are usually referred as $B(1), B(2), ..., B(m)$.

Next we examine when a basic feasible solution optimal. Let $x$ be a basic feasible solution associated with a basis matrix $B$. Then the value of the objective function is $\mathbf{c}x = c_Bx_B = c_BB^{-1}b$. If this is optimal, from the duality theorem of linear programming we know that it should be the optimal value of $\max\{yb : yA \leq c\}$, which means $y = c_BB^{-1}$ should be an optimal solution, and thus $c_BB^{-1}A \leq c$ holds. The following theorem formalizes this observation.

**Theorem.** Let $x$ be a basic feasible solution associated with a basis matrix $B$.

(a) If $c_BB^{-1}A \leq c$, then $x$ is optimal.
(b) Suppose $c_BB^{-1}A_{j} > c_j$ for some $j$. Let $d$ be the vector with $d_B = -B^{-1}A_j$, $d_j = 1$, and $d_i = 0$ for all other $i$. Then $Ad = 0$ and $c'd < 0$.
(c) If $x$ is optimal and $x_B > 0$, then $c_BB^{-1}A \leq c'$.

A basic feasible solution as described in (c) will be called nondegenerate. Equivalently, a basic feasible solution is degenerate if and only if more than $n - m$ of its entries are zero.

When we consider $x + \theta d$, where $d$ is the vector in the last theorem and $\theta > 0$, the nonbasic variable $x_j$ becomes positive and all other nonbasic variables remain at zero. Let us define $c' = c' - c_BB^{-1}A$. If $\tau_j < 0$, then $c'(x + \theta d) = c'x + \theta c_j$. It follows that we should move along the direction $d$ as far as possible, that is, we need to find $\theta^* = \max\{\theta \geq 0 : x + \theta d \in P\}$.

Consider the following problem:

**minimize**

\[ x_1 \]

**subject to**

\[ x_1 + x_2 + x_3 + x_4 = 2 \]
\[ 2x_1 + 3x_3 + 4x_4 = 2 \]
\[ x_1, x_2, x_3, x_4 \geq 0. \]

Clearly, we have

\[ A = \begin{bmatrix} 1 & 2 & 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 3 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad \text{and} \quad c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \]

Let $B = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$. Then $x = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ and $c' - c_BB^{-1}A = [0, 0, -3, -4]$. Let us take $j = 3$. Then $d = \begin{bmatrix} -3/2 \\ 1/2 \\ 1/0 \end{bmatrix}$ and thus $x + \theta d \geq 0$ if and only if $\theta \leq 2/3$. If we take $\theta = 2/3$, it is easy to verify that $y = x + \theta d$ is a basic feasible solution associated with the basis matrix $B$ which consists of $A_2$ and $A_3$. Moreover, $y$ is optimal since $c' - c_BB^{-1}A \geq 0$.

**Lecture 8. September 16**

In general, since $Ad = 0$, we have $A(x + \theta d) = Ax = b$ for all $\theta$, and thus $x + \theta d$ can become infeasible only if one of its components becomes negative. There are two possibilities:

1. $d \geq 0$, then $x + \theta d \geq 0$ for all $\theta \geq 0$, the vector $x + \theta d$ is always feasible and thus $\theta^* = \infty$. 

(2) For some $i$, we have $d_i < 0$. It is easy to verify that $\mathbf{x} + \theta \mathbf{d} \geq 0$ implies $\theta^* = \min_{d_{B(i)} < 0} \left\{ \frac{-x_{B(i)}}{d_{B(i)}} \right\}$.

Let $l$ be an index with $d_{B(l)} < 0$ and $-x_{B(l)}/d_{B(l)} = \theta^*$. Let $\bar{B}$ be the matrix obtained from $B$ by replacing $A_{B(l)}$ with $A_j$, and $y = x + \theta^* \mathbf{d}$.

**Theorem 3.2.** $\bar{B}$ is a basis matrix and $y$ is the basic feasible solution associated with $\bar{B}$.

Notice that we do not need to assume the nondegeneracy of $\mathbf{x}$ to prove this theorem.

**An iteration of the simplex method**

1. In a typical iteration, we start with a basis consisting of the basic columns $A_{B(1)}, A_{B(2)}, \ldots, A_{B(m)}$, and an associated feasible solution $\mathbf{x}$.

2. Compute $\mathbf{c}$. If $\mathbf{c} \geq 0$, stop, $\mathbf{x}$ is optimal; otherwise, choose an index $j$ with $c_j < 0$.

3. Compute $\mathbf{u} = B^{-1} A_j$. If $\mathbf{u} \leq 0$, stop, the optimal value is $-\infty$.

4. If $\mathbf{u} \not\leq 0$, let $\theta^* = \min \{ x_{B(i)}/u_i : u_i > 0 \}$.

5. Let $l$ be such that $\theta^* = x_{B(l)}/u_l$. Form a new basis by replacing $A_{B(l)}$ with $A_j$. If $\mathbf{y}$ is the new basic feasible solution, the values of the new basic variables are $y_j = \theta^*$ and $y_{B(i)} = x_{B(i)} - \theta^* u_i$, for $i \neq l$.

**Observation.** If the simplex method terminates, then either we find an optimal basis and an associated basic feasible solution which is optimal, or we find a vector $\mathbf{d}$ satisfying $A\mathbf{d} = 0$, $\mathbf{d} \geq 0$, and $\mathbf{c} \mathbf{d} < 0$, and thus, in this case, the optimal value is $-\infty$.

If $\mathbf{x}$ is nondegenerate, we must have $\theta^* > 0$. It follows that $\mathbf{c}' \mathbf{y} = \mathbf{c}'(\mathbf{x} + \theta^* \mathbf{d}) = \mathbf{c}' \mathbf{x} + \theta^* c_j < \mathbf{c}' \mathbf{x}$. Combining this observation with corollary 2.1 we deduce the following.

**Theorem.** Suppose that there is at least one basic feasible solution and all basic feasible solutions are non-degenerate. Then the simplex method terminates after a finite number of iterations.

**Lecture 9.** September 21

Now we talk about the tableau implementation of the simplex method. Let $B$ be a basis matrix. The **simplex tableau** $T(B)$ is defined to be:

$$
\begin{array}{c|c}
-c_B^0 B^{-1} b & c' - c_B^0 B^{-1} A \\
\hline
B^{-1} b & B^{-1} A
\end{array}
$$

Elements of this tableau will be denoted by $u_{i,j}$ for $i = 0, 1, \ldots, m$ and $j = 0, 1, \ldots, n$.

**An iteration of the tableau implementation**

1. In a typical iteration, we start with $T(B)$ for some basis $B$.

2. If $u_{0,j} \geq 0$ for all $j > 0$, stop, $\mathbf{x}$ is optimal; otherwise, choose an index $j > 0$ with $u_{0,j} < 0$. The $j$th column is called the **pivot column**.

3. If $u_{i,j} \leq 0$ for all $i$, stop, the optimal value is $-\infty$; otherwise, choose and index $l$ such that $x_{B(l)}/u_{l,j} = \min \{ x_{B(i)}/u_{i,j} : u_{i,j} > 0 \}$. The $l$th row is called the **pivot row**.
4. Add to each row of the tableau a constant multiple of the \( l \)th row so that \( u_{l,j} \), called the pivot element, becomes one and all other entries of the \( j \)th column become zero. This process is called a pivot.

**Example.** Consider minimize 
\[
-10x_1 - 12x_2 - 12x_3
\]
subject to 
\[
\begin{align*}
x_1 + 2x_2 + 2x_3 + x_4 &= 20 \\
2x_1 + x_2 + 2x_3 + x_5 &= 20 \\
2x_1 + 2x_2 + x_3 + x_6 &= 20
\end{align*}
\]

**Lecture 10. September 23**

**Theorem.** After each iteration, \( T(B) \) is transformed into \( T(B') \).

We have proved that if the simplex method terminate then the output is the right answer. In addition, if all basic feasible solutions are nondegenerate, then the method must terminate. Next we consider the case when not all basic feasible solutions are nondegenerate. If we apply the same algorithm, then there are two more possible outcomes:

1. If the current basic feasible solution \( x \) is degenerate, \( \theta^* \) can be zero, in which case, the new basic feasible solution \( y \) is the same as \( x \). This happens if \( x_{B(i)} = 0 \) and \( d_{B(i)} < 0 \) for some \( i \). Nevertheless, we can still define a new basis \( B' \), as described in the algorithm, and Theorem 3.2 still holds. Therefore, even though we do not have a new basic feasible solution, we do have a new basis, and thus we can still consider this as an improvement.

2. Even if \( \theta^* \) is positive, it may happen that more than one of the original basic variables becomes zero at the new point \( x + \theta^*d \). Since only one of them exits the basis, the others remain in the basis at the zero level, and the new basic feasible solution is degenerate.

Observe that a sequence of basis changes while staying at the same basic feasible solution may lead back to the initial basis, in which case the algorithm may loop indefinitely. This undesirable phenomenon is known as cycling.

In most of the cases, the pivot element is not unique. We often have several choices for the pivot row and column. Now we impose a new rule on how to choose the pivot element so that the simplex method is guaranteed to terminate. First, a definition. A vectors \( u \) is lexicographically positive if its first nonzero entry is positive. We also call a vector \( u \) lexicographically larger (or smaller) than another vector \( v \) if \( u - v \) (or \( v - u \), respectively) is lexicographically positive. Symbolically, we write \( u >_L v \) or \( u <_L v \).

**Lexicographic pivoting rule:**
1. Choose \( j \) arbitrarily, as long as \( u_{0,j} \) is negative.
2. For each \( i \) with \( u_{i,j} > 0 \), divide the \( i \)th row of \( T(B) \) (including the 0th column) by \( u_{i,j} \) and then choose \( l \) such that the \( l \)th row is the lexicographically smallest.

**Theorem.** Suppose that the simplex algorithm starts with all the rows in the simplex tableau, other than the 0th row, lexicographically positive. Suppose the lexicographic pivoting rule is followed. Then
(a) Other than the 0th row, every row of the simplex tableau is lexicographically positive.
(b) The 0th row strictly increases lexicographically at each iteration.
(c) The simplex method terminates after a finite number of iterations.

In order to apply the lexicographic pivoting rule, an initial tableau with lexicographically positive rows is required. But if an initial tableau without this property is available, we can simply rename the variables so that the basic variables are the first \( m \) ones. This is equivalent to rearranging the tableau so that the
first \( m \) columns of \( B^{-1}A \) are the \( m \) unite vectors. The resulting tableau clearly has lexicographically positive rows, as desired.

Lecture 11. September 28

Consider our standard problem \( \min \{ cx : Ax = b, x \geq 0 \} \). By possibly multiplying some of the inequalities by \(-1\), we can assume, without loss of generality, that \( b \geq 0 \). To find the initial basis, we introduce a vector \( y \) of artificial variables and construct the following auxiliary problem.

\[
\begin{align*}
\text{minimize} & \quad y_1 + y_2 + \ldots + y_m \\
\text{subject to} & \quad Ax + y = b \\
& \quad x, y \geq 0.
\end{align*}
\]

This problem clearly has a finite optimal value. To determine its minimum, the simplex method can be applied without any difficulty: by taking the initial basic feasible solution \( x = 0, y = b \). The outcome will be a basic feasible solution of our original problem, as long as it has one.

Finding an initial basic feasible solution
1. By multiplying some of the constraints by \(-1\), change the problem so that \( b \geq 0 \).
2. Introduce artificial variables and apply the simplex method to the auxiliary problem.
3. If the optimal value in the auxiliary problem is positive, then the final basis can be served as an initial basis of the original problem.
4. If the optimal value in the auxiliary problem is zero and no artificial variable is in the final basis, then the final basis can be served as an initial basis of the original problem.
5. If the optimal value in the auxiliary problem is zero but some artificial variables are in the final basis, then we need to eliminate these variables as follows. Suppose an artificial variable \( y_t \) is in the basis, and it is the \( l \)-th basic variable, that is, in the column index by \( y_t \), the \( l \)-th entry is the only nonzero one. Choose a nonzero entry from the \( l \)-th row of \( B^{-1}A \), say, from the \( j \)-th column, and then make a pivot at this element. The result is to replace \( y_t \) by \( x_j \) as a new basic variable. Repeat this operation until all the artificial variables are driven out of the basis.

We have proved the duality theorem using Fourier-Motzkin elimination. Next, we prove it again using the simplex method.

Another proof of the duality theorem. By applying the simplex method to the standard linear programming problem, we have three possible outcomes. (a) There is no feasible solution; (b) The minimum is \(-\infty\); (c) There is an optimal solution.

If (a) happens, then the simplex method provides a basis matrix \( B \) of the auxiliary problem. That is, \( p_B' B^{-1} (A, I) \leq p' \) and \( p_B' B^{-1} b > 0 \), where \( p' = (0', 1') \). Let \( q' = p_B' B^{-1} \). Then \( q' A \leq 0 \) and \( q' b > 0 \). It follows that the maximum of \( y' b \) is \( \infty \), as long as the dual problem is feasible.

If (b) happens, then the simplex method provides a vector \( d \) for which \( Ad = 0 \) and \( c' d < 0 \). Clearly, it follows that the dual problem is infeasible.

If (c) happens, then the simplex method provides a basis matrix \( B \) and a basic feasible solution \( x \) for which \( c_B' B^{-1} A \leq c' \), \( B^{-1} b \geq 0 \) and \( c' x = c_B' B^{-1} b \). Let \( y = c_B B^{-1} \). Then it is clear that \( y \) is a feasible solution of the dual problem and \( y' b = c' x \). Therefore, we have maximum=minimum.

Lecture 12. September 30

There are many theorems that are equivalent with the duality theorem. The following is an example.

Farkas’ Lemma. Given column vectors \( A_1, ..., A_n, \) and \( b \). Then either
(a) \( b \) can be expressed as a nonnegative linear combination of \( A_1, ..., A_n, \) or
(b) for some row vector \( p \) we have \( pA_1 \geq 0, ..., pA_n \geq 0, \) and \( pb < 0, \) but not both.
By considering $\max\{0'x : Ax = b, x \geq 0\}$, where $A$ is the matrix with columns $A_1, \ldots, A_n$, it is not difficult to prove Farkas Lemma using the duality theorem. The following is a proof of the duality theorem using Farkas Lemma (Thus these two theorems are equivalent).

**Corollary.** $Ax \leq b$ is infeasible if and only if there is a vector $y \geq 0$ with $y'A = 0$ and $y'b < 0$.

**Corollary.** Suppose $y'A \leq c'$ is feasible. Let $b$ be a vector and $\alpha$ be a real number. Then $y'b \leq \alpha$ hold for all $y$ with $y'A \leq c'$ if and only if there is a vector $x \geq 0$ with $Ax = b$ and $c'x \leq \alpha$.

**Another proof of the duality theorem.** We only consider the case when both the primal and the dual have feasible solutions. We have shown that $\min\{c'x : Ax = b, x \geq 0\} \geq \max\{y'b : y'A \leq c'\}$. Thus there is a number $\alpha$ with $c'x \geq \alpha \geq y'b$ for all feasible solutions $x$ and $y$. From the last corollary we deduce that there is a vector $x^* \geq 0$ with $Ax^* = b$ and $c'x^* \leq \alpha$. It follows that $c'x^* = \alpha$ and thus $x^*$ is an optimal solution of the primal problem. It remains to show that $y'A \leq c'$, $y'b \geq \alpha$ is feasible. Suppose otherwise. From the second last corollary, it follows that there is a vector $p \geq 0$ and a scaler $t \geq 0$ such that $Ap = tb$ and $c'p < ta$. If $t > 0$, then, by setting $x = p/t$, we have $Ax = b$ while $c'x < \alpha$, a contradiction. But if $t = 0$, we have $Ap = 0$ and $c'p < 0$, which is again a contradiction, since $y'A \leq c'$ is feasible.

**Lecture 13. October 5**

Next, we provided an independent proof of Farkas’ lemma. First some background material. A subset $X$ of $\mathbb{R}^n$ is closed if, whenever a sequence $x^1, x^2, \ldots$ of members of $X$ converges to some $x$, then $x$ belongs to $X$. If $x_1, x_2, \ldots, x_n$ are the entries of $x$, then $||x||$ is defined to be $(x_1^2 + x_2^2 + \ldots + x_n^2)^{1/2}$. The next is a well-known fact.

**Proposition.** If $X$ is an infinite set of vectors such that $||x|| < R$ for all $x$ in $X$, where $R$ is a real number, then there exists a convergent infinite sequence of distinct members of $X$.

**Lemma 4.10.** Let $X$ be a nonempty and closed subset of $\mathbb{R}^n$. Let $z$ be a vector not in $X$. Then there exists a vector $x^*$ in $X$ such that $||z - x^*|| \leq ||z - x||$ for all $x$ in $X$.

**Lemma 4.11.** Let $X$ be a nonempty, closed, and convex subset of $\mathbb{R}^n$. Let $z$ be a vector not in $X$. Then there exists a vector $c$ such that $c'z < c'x$ for all $x$ in $X$.

**Lecture 14. October 7**

**Lemma.** Let $A$ be a set of vectors and $b$ be a nonnegative linear combination of some vectors in $A$. Then $b$ can be expressed as a nonnegative linear combination of some linearly independent vectors in $A$.

**Another proof of the Farkas’ lemma.**

**Lecture 15. October 12**

Our next topic is the representation theorem of polyhedra. Let $A$ be an $m \times n$ matrix of rank $r$. From linear algebra we know that there are $n - r$ pairwise perpendicular unit vectors $h_1, h_2, \ldots, h_{n-r}$ such that every solution of $Ax = 0$ can be expressed as a linear combination of these $n - r$ vectors. Let $P = \{x \in \mathbb{R}^n : Ax \geq b\}$ and $Q = \{y \in \mathbb{R}^n : Ay \geq b, h_1'y = 0, h_2'y = 0, \ldots, h_{n-r}'y = 0\}$.

**Lemma.** Let $P$ and $Q$ be defined as above.

(a) If $P$ is not empty, then $Q$ has at least one vertex.

(b) $x \in P$ if and only if $x = y + \theta_1 h_1 + \theta_2h_2 + \ldots + \theta_{n-r} h_{n-r}$ for some $y \in Q$, and some real numbers $\theta_1, \theta_2, \ldots, \theta_{n-r}$.
A nonzero vector $d$ is an extreme ray of the polyhedron $P = \{ x : Ax \geq b \}$ if $Ad \geq 0$ and $A$ has $n-1$ linearly independent rows $a_1, a_2, \ldots, a_{n-1}$ such that $a_id = 0$ for all $i$. From the definition it is clear that $P$ has only finitely many extreme rays.

**Lemma.** Suppose $P = \{ x : Ax \geq b \}$ has at least one extreme point. Then $\min \{ c'x : x \in P \}$ equals $-\infty$ if and only if $c'd < 0$ for some extreme ray $d$ of $P$.

**Lemma.** Suppose $P = \{ x : Ax \geq b \}$ has at least one extreme point. Let $x_1, x_2, \ldots, x_p$ be all extreme points of $P$, and let $d_1, d_2, \ldots, d_q$ be all extreme rays of $P$. Let

$$Q = \left\{ \sum_{i=1}^{p} \lambda_i x_i + \sum_{j=1}^{q} \mu_j d_j : \lambda_i \geq 0, \mu_j \geq 0, \sum_{i=1}^{p} \lambda_i = 1 \right\}.$$  

Then, $Q = P$.

**Theorem.** Let $P$, $x_i$, $d_j$, and $h_k$ be as defined above. Let

$$Q = \left\{ \sum_{i=1}^{p} \lambda_i x_i + \sum_{j=1}^{q} \mu_j d_j + \sum_{k=1}^{r} \theta_k h_k : \lambda_i \geq 0, \mu_j \geq 0, \sum_{i=1}^{p} \lambda_i = 1 \right\}.$$  

Then $Q = P$.

Very often, the last theorem is stated as follows.

**Theorem.** For any polyhedron $P$, there exist vectors $x_1, x_2, \ldots, x_p$ and $d_1, d_2, \ldots, d_q$ such that

$$P = \left\{ \sum_{i=1}^{p} \lambda_i x_i + \sum_{j=1}^{q} \mu_j d_j : \lambda_i \geq 0, \mu_j \geq 0, \sum_{i=1}^{p} \lambda_i = 1 \right\}.$$  

Lecture 16. October 14

The next is a strengthening of the last theorem.

**Theorem.** A set $P$ of vectors is a polyhedron if and only if there exist vectors $x_1, x_2, \ldots, x_p$ and $d_1, d_2, \ldots, d_q$ such that

$$P = \left\{ \sum_{i=1}^{p} \lambda_i x_i + \sum_{j=1}^{q} \mu_j d_j : \lambda_i \geq 0, \mu_j \geq 0, \sum_{i=1}^{p} \lambda_i = 1 \right\}.$$  

**Corollary.** A set $P$ of vectors is a bounded polyhedron if and only if it is the convex hull of finitely many vectors, that is

$$P = \left\{ \sum_{i=1}^{p} \lambda_i x_i : \lambda_i \geq 0, \sum_{i=1}^{p} \lambda_i = 1 \right\},$$  

for some vectors $x_1, x_2, \ldots, x_p$. Moreover, vectors $x_i$ can be chosen as the extreme rays of $P$.

**Corollary.** A set $C$ of vectors is a polyhedral cone, that is, $C = \{ x : Ax \geq 0 \}$ for some matrix $A$, if and only if it is the nonnegative linear combinations of some vectors. Moreover, these vectors can be chosen as the extreme rays of $C$ when $0$ is the extreme point of $C$.

Our next topic is network. A graph $G$ is a ordered pair $(N,E)$, where $N$ is a finite set of nodes or vertices of $G$, and $E$ is a set of unordered pairs of distinct nodes in $N$ called it edges of $G$. If $e = xy$ is an edge, then we say that $e$ is incident with $x$ and $y$. $x$ and $y$ are called the endpoints of $e$ and they are adjacent to each other.

A walk $W$ between two nodes $x$ and $y$ in a graph is a sequence $n_1, n_2, \ldots, n_k$ such that $\{n_1, n_k\} = \{x, y\}$.
and \( n_in_{i+1} \in E \) for all \( i < k \). \( W \) is also called an \( xy \)-walk.

- \( W \) is called a closed walk if \( x = y \).
- \( W \) is called a path (or \( xy \)-path) if all its nodes are distinct.
- \( W \) is called a cycle if it is closed and all nodes \( n_1, n_2, ..., n_k \) are distinct.

A graph \( G = (N, E) \) is connected if, for any two nodes \( x \) and \( y \) in \( N \), there is an \( xy \)-walk in \( G \).

**Observation.** A graph \( G = (N, E) \) is not connected if and only if there is a partition \( (X, Y) \) of \( N \) such that \( X \) and \( Y \) are not empty and there is no edge between \( X \) and \( Y \).

**Lecture 17. October 19**

**Observation.** There is an \( xy \)-walk if and only if there is an \( xy \)-path.

**Observation.** Let \( W \) be an \( xy \)-walk, \( C \) be a cycle, and \( e \) be an edge in both \( W \) and \( C \). Then there is an \( xy \)-walk avoiding \( e \).

**Proposition.** If an edge \( e \) of a connected graph \( G = (N, E) \) is contained in a cycle, then the graph \( (N, E - \{e\}) \) is also connected.

A connected graph is called a tree if it has no cycles.

**Theorem.** \( G = (N, E) \) is a tree if and only if \( G \) is connected and the graph \( (N, D) \) is not connected for every proper subset \( D \) of \( E \).

Let \( G = (N, E) \) and \( H = (M, D) \) be graphs. Suppose \( M \subseteq N \) and \( D \subseteq E \). Then we call \( H \) a subgraph of \( G \) and \( G \) a supergraph of \( H \). If \( H \) is a tree and \( M = N \), then \( H \) is called a spanning tree of \( G \).

**Theorem.** Let \( D \) be a set of edges of a connected graph \( G = (N, E) \). If \( (N, D) \) has no cycles, then \( G \) has a spanning tree \( (N, T) \) with \( T \supseteq D \).

A vertex in a graph is a leaf if it is adjacent to only one vertex.

**Theorem.** The following are equivalent for any graph \( G = (N, E) \).

(a) \( G \) is a tree.
(b) \( G \) is connected and every connected subgraph of \( G \) with more than one node has at least one leaf.
(c) \( G \) is connected and \( |E| = |N| - 1 \).
(d) For any two nodes \( x \) and \( y \) in \( N \), \( G \) has exactly one \( xy \)-path.
(e) For any two distinct nodes \( x \) and \( y \) of \( G \), the graph \( (N, E \cup \{e\}) \) obtained from \( G \) by adding a new edge \( e = xy \) has exactly one cycle. (Here, we allow two edges between \( x \) and \( y \), and such two edges are considered as a cycle.)

**Lecture 18. October 28**

A directed graph \( G \) is an ordered pair \( (N, A) \), where \( N \) is a finite set of nodes or vertices of \( G \), and \( A \) is a set of ordered pairs of distinct nodes in \( N \) called arcs of \( G \). Notice that \( (i, j) \) and \( (j, i) \) are two different arcs. For any arc \( (i, j) \), we say that it is outgoing from \( i \) and incoming to \( j \). We define \( O(i) \) to be the set of arcs outgoing from \( i \) and \( I(i) \) to be the set of arcs incoming to \( i \).

If \( G = (N, A) \) is a directed graph, then we define the underlying graph of \( G \) to be an undirected graph \( H = (N, E) \), where \( xy \) belongs to \( E \) if and only if \( (x, y) \) or \( (y, x) \) (could be both) belongs to \( A \). We call \( G \) connected if \( H \) is connected. A walk \( W \) between two nodes \( x \) and \( y \) in a directed graph is an alternating sequence \( n_1, a_1, n_2, a_2, ..., n_k, a_k, n_{k+1} \) of nodes and arcs such that \( n_1 = x \), \( n_{k+1} = y \), and either \( a_i = (n_i, n_{i+1}) \) (in which case we say that \( a_i \) is a forward arc) or \( a_i = (n_{i+1}, n_i) \) (in which case we say
that $a_i$ is a backward arc), for all $i \leq k$. $W$ is also called an $xy$-walk.

$W$ is called a closed walk if $x = y$. $W$ is called a path (or $xy$-path) if all its nodes are distinct. $W$ is called a cycle if it is closed and all nodes $n_1, n_2, \ldots, n_{k-1}$ are distinct.

Finally, a walk, path, or cycle is directed if it contains only forward arc.

Now we formulate the general network flow problem. Let $G = (N, E)$ be a directed graph. Let $N$ be partitioned into $N^+$ and $N^-$. Let $b_j \geq 0$ be a real number for all $j$ in $N$. Finally, let $c_{ij}$ be a real number for each $(i, j)$ in $E$. The following is the network flow problem.

$$\text{minimum } \sum_{(i, j) \in E} c_{i,j} x_{i,j}$$

subject to

$$\sum_{(i, j) \in E} x_{i,j} - \sum_{(j, i) \in E} x_{j,i} = b_j, \quad j \in N^+, \ldots,$$

$$\sum_{(j, i) \in E} x_{j,i} = \sum_{(i, j) \in E} x_{i,j} = b_j, \quad j \in N^-,$$

$x_{i,j} \geq 0$, for all $i, j$.

Observe that the equalities can be unified, if we drop the nonnegativity requirement on $b_j$. Thus we have the following equivalent problem.

$$\text{minimum } \sum_{(i, j) \in E} c_{i,j} x_{i,j}$$

subject to

$$\sum_{(j, i) \in E} x_{j,i} = \sum_{(i, j) \in E} x_{i,j} = b_j, \quad j \in N,$$

$x_{i,j} \geq 0$, for all $i, j$.

Let $A$ be the incidence matrix of $G$. That is, the rows of $A$ are indexed by nodes of $G$ and the columns of $A$ are indexed by the arcs of $G$ such that each entry $a_{i,j}$ is zero if arc $j$ is not incident with node $i$, is 1 if arc $j$ is outgoing from node $i$, and is $-1$ if arc $j$ is incoming to node $i$. Then it is easy to verify that the network flow problem can be restated as the following.

$$\text{minimum } cx$$

subject to $Ax = b,$

$x \geq 0$.

Throughout the discussion, let us assume that $G$ is connected, $|N| = n$, and $b_1 + b_2 + \ldots + b_n = 0$.

Lecture 19. November 2

Observation. Let $\{A_{e_1}, A_{e_2}, \ldots, A_{e_t}\}$ be a set of columns of $A$. Then these column vectors are linearly independent if and only if $\{s_1, e_2, \ldots, e_t\}$ does not contain a cycle.

As a consequence of this observation, $r(A) = n - 1$. Let $\tilde{A}$ be obtained from $A$ by deleting a row, and let $\tilde{b}$ be obtained from $b$ by deleting the same row. Then the network-flow problem is equivalent to

$$\text{minimum } cx$$

subject to $A\tilde{x} = \tilde{b},$

$x \geq 0$.

Lemma. An $(n - 1) \times (n - 1)$ submatrix $B$ of $\tilde{A}$ is nonsingular if and only if the corresponding edges form a spanning tree of $G$.

In the rest of the discussion, we will assume that the vertices of $G$ are indexed by 1, 2, ..., $n$ and the
the row deleted from $A$ to obtain $\tilde{A}$ is indexed by $n$.

**Lemma.** Let $x > 0$ be a feasible solution of the network flow problem. That is $Ax = b$. Then $x$ is a basic feasible solution if and only if there is a spanning tree $T$ such that $x_a = 0$ for all arcs $a$ not in $T$.

**Lecture 20.** November 9

During a typical iteration in the simplex method, one basic feasible solution is transformed into another basic feasible solution. In case of the network flow problem, it means that one spanning tree is transformed into another spanning tree. Next, we will see how such a transformation works. First, some preparations. A *circulation* is a vector $x$ indexed by arcs of $G$ such that $Ax = 0$.

**Lemma.** Let $x$ be a circulation of a directed graph $G = (N, E)$ and let $(S, T)$ be a partition of $N$. Then

$$
\sum_{i \in S, j \in T} x_{i,j} = \sum_{i \in S, j \in T} x_{j,i}.
$$

An edge $e$ in a graph $G = (N, E)$ is a cut if $N$ can be partitioned into $(S, T)$ such that $e$ is the only edge of $G$ between $S$ and $T$.

**Lemma.** If a connected graph $G = (N, E)$ has exactly one cycle $C$, then every edge not in $C$ is a cut.

**Theorem.** Suppose a basic feasible solution $x$ is transformed into another $\bar{x}$. Let $T$ and $\bar{T}$ be the corresponding spanning trees.

(a) $T$ and $\bar{T}$ only differ by one edge.

In another words, $\bar{T}$ can be obtained from $T$ as follows: adding an edge $\bar{e}$ to $T$, which create exactly one cycle $C$, and then delete an edge $e$ other than $\bar{e}$ from this cycle $C$.

(b) $\bar{x}_f = x_f$ if $f$ is not in the cycle $C$.

(c) If we call an arc in $C$ forward or backward according to if its direction agree with $\bar{x}$ in $C$, then

$$
\bar{x}_f = x_f - x_e \text{ for each forward arc } f \text{ and } \bar{x}_f = x_f + x_e \text{ for each backward arc } f.
$$

Moreover, $e$ is a backward arc and $\bar{e} = x_e$.

Therefore, if we know how to determine $\bar{e}$, then the new basic feasible solution $\bar{x}$ can be determined from the current basic feasible solution $x$ as follows. Let $T$ be the spanning tree that corresponds to $x$. First, determine the unique cycle $C$ in $T + e$. Then, among all backward arcs, find $e$ such that $x_e$ is the minimum. Finally, we define

$$
\bar{x}_f = \begin{cases} 
  x_f & \text{if } f \text{ is not in } C, \\
  x_f - x_e & \text{if } f \text{ is a backward arc in } C, \\
  x_f + x_e & \text{if } f \text{ is a forward arc in } C, \text{ including } \bar{e}.
\end{cases}
$$

To determine $\bar{e}$, we need to compute $\bar{e} = c - cB^{-1}\tilde{A}$, which can be done as follows. Let $p = cB^{-1}$, which, according to the definition of $\tilde{A}$, is a function on $\{1, 2, ..., n - 1\}$. It follows that

$$
\bar{c}_{i,j} = \begin{cases} 
  c_{i,j} - (p_i - p_j) & \text{if } i \neq n \text{ and } j \neq n, \\
  c_{i,j} - p_i & \text{if } i \neq n \text{ and } j = n, \\
  c_{i,j} + p_j & \text{if } i = n \text{ and } j \neq n.
\end{cases}
$$

This equation can be tritten more concisely if we define $p_n = 0$, in which case we have

$$
\bar{c}_{i,j} = c_{i,j} - (p_i - p_j), \quad \text{for all } (i, j) \in E.
$$

Thus, to compute $\bar{e}$, we need to compute $p$. But this can be done easily since $\bar{x}_f = 0$ for all $f$ in $T$. Therefore, $p$ can be uniquely determined from
\[ p_i - p_j = c_{i,j} \text{ for all } (i, j) \in T, \text{ and} \\
p_n = 0. \quad (\ast) \]

The simplex method for the network flow problem

1. A typical iteration starts with a basic feasible solution \( x \) associated with a spanning tree \( T \).
2. Compute \( p \) by using \((\ast)\) above.
3. Compute \( c_{i,j} = c_{i,j} - (p_i - p_j) \) for all arcs \((i, j)\) not in \( T \). If they are all nonnegative, then \( x \) is optimal and the algorithm terminates; else, choose \( e = (i, j) \) with \( c_{i,j} < 0 \).
4. Let \( C \) be the unique cycle in \( T + e \). If all arcs in \( C \) are forward, then the minimum is \(-\infty\) and the algorithm terminates.
5. Determine the new basic feasible solution \( x \) as we discussed above.

There are several ways to find an initial basic feasible solution. One is a modification of the big-M method which is described as follows. Define \( G^+ \) to be the directed graph obtained from \( G \) by adding arcs from each node \( i \) with \( b_i \geq 0 \) to each node \( j \) with \( b_j < 0 \). Also let \( c_f = M \) for these new arcs, where \( M \) is a big real number. Now finding an initial basic feasible solution in \( G^+ \) is straightforward. Thus we can solve the flow problem in \( G^+ \). One can show that, just like in the case of the big-M method, the new problem is equivalent to the original problem.

Another way of finding an initial basic feasible solution is to solve a max-flow problem, which is one of the homework problems. Next, we discuss the max-flow problem in detail.
Homework

SET 1

1. A company produces and sells two different products. The demand for each product is unlimited, but the company is constrained by cash availability and machine capacity.

Each unit of the first and second product requires 3 and 4 machine hours, respectively. There are 20,000 machine hours available in the current production period. The production costs are $3 and $2 per unit of the first and second product, respectively. The selling prices of the first and second product are $6 and $5.40 per unit, respectively. The available cash is $4,000; furthermore, 45% of the sales revenues from the first product and 39% of the sales revenues from the second product will be made available to finance operations during the current period.

(a) Formulate a linear programming problem that aims at maximizing net income subject to the cash availability and machine capacity limitations.

(b) Solve the problem graphically to obtain an optimal solution.

2. Consider the following two inequalities

\[
\begin{bmatrix}
2 & 3 & 0 \\
1 & -1 & 1 \\
-1 & 2 & -1 \\
2 & -1 & 3 \\
5 & 6 & -4
\end{bmatrix} \cdot \begin{bmatrix}
x \\
y \\
z
\end{bmatrix} \leq \begin{bmatrix}
3 \\
1 \\
2 \\
-3 \\
8
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
2 & 3 & 1 \\
1 & -1 & 2 \\
-1 & 3 & 2 \\
2 & -1 & 3 \\
5 & 6 & 4
\end{bmatrix} \cdot \begin{bmatrix}
x \\
y \\
z
\end{bmatrix} \leq \begin{bmatrix}
3 \\
1 \\
2 \\
-3 \\
8
\end{bmatrix}.
\]

In each of these two cases:

(a) Use Fourier-Motzkin elimination to eliminate \( z \).

(b) Find a matrix \( P \) as described in the Proposition on page 7.

3. **Duality Theorem.** Consider the following two optimization problems:

Minimize \( cx \)

subject to \( Ax = b \)

\( x \geq 0 \)

Maximize \( yb \)

subject to \( yA \leq c \)

Prove the following.

(a) If \( y'A \leq c' \) is feasible but \( Ax = b, x \geq 0 \) is infeasible, then the maximum of \( y'b \) is \(+\infty\).

(b) If \( Ax = b, x \geq 0 \) is feasible but \( y'A \leq c' \) is infeasible, then the minimum of \( c'x \) is \(-\infty\).

(c) If both \( y'A \leq c' \) and \( Ax = b, x \geq 0 \) are feasible then \( \text{Min } c'x = \text{Max } y'b \).

4. Consider the following optimization problem:

Minimize \( gx + hy \)

subject to \( Ax + By = e, Cx + Dy \leq f, \text{ and } x \geq 0 \)

where \( A, B, C, D, e, f, g, h \) are given matrices and vectors. Find a matrix \( M \) and vectors \( \alpha, \beta \), in terms of \( A, B, C, D, e, f, g, h \), such that the above problem is equivalent to \( \text{min}\{\alpha z : Mz = \beta, z \geq 0\} \).
Solutions

3. Proof.

(a) If the system $A x = b$, $x \geq 0$, which can be rewritten as

$$
\begin{pmatrix}
A \\
-A \\
-I
\end{pmatrix}
\begin{pmatrix}
x
\end{pmatrix}
\leq
\begin{pmatrix}
b \\
-b \\
0
\end{pmatrix},
$$

is infeasible, then, by the theorem on page 8, there exists a nonnegative row vector $(u, v, w)$ for which

$$
u A - v A - w = (u, v, w) \begin{pmatrix} A \\ -A \\ -I \end{pmatrix} = 0 \quad \text{while} \quad u b - v b = (u, v, w) \begin{pmatrix} b \\ -b \\ 0 \end{pmatrix} < 0.
$$

Let $y_0$ be a feasible solution of $y' A \leq c'$. Then it is easy to verify that $y_t = y_0' + t(v - u)$ is a feasible solution of $y' A \leq c'$ for all nonnegative reals $t$. It follows that $y_t b = y_0 b + t(v - u) b \to +\infty$ as $t \to +\infty$.

(b) If $y' A \leq c'$ is infeasible, then, by the same theorem on page 8, there exists a nonnegative vector $u$ with $A u = 0$ and $c' u < 0$. Let $x_0$ be a feasible solution of the system $A x = 0$, $x \geq 0$. Then it is easy to verify that $x_t = x_0 + t u$ is a feasible solution of system $A x = 0$, $x \geq 0$, for all nonnegative reals $t$. It follows that $c' x_t = c' x_0 + t c' u \to -\infty$ as $t \to +\infty$.

(c) For any feasible solutions $x$ and $y$ of the two problems, we have $c' x \geq (y' A) x = y'(A x) = y' b$, and thus $\min \{c' x : A x \geq b, x \geq 0\} \geq \max \{y' b : y \leq c'\}$. So we only need to show that $c' x \leq y' b$ for some feasible solutions, that is, the system $c' x \leq y' b$, $y' A \leq c'$, $A x = b$, $x \geq 0$ is feasible. First, rewrite the system as $c' x - b'y \leq 0$, $A'y \leq c$, $A x \leq b$, $-A x \leq -b$, $-x \leq 0$, which is equivalent to

$$
\begin{bmatrix}
c' \\
-b'
\end{bmatrix}
\begin{bmatrix}
x
\end{bmatrix}
\leq
\begin{bmatrix}
0 \\
c
\end{bmatrix},
$$

and

$$
\begin{bmatrix}
c' \\
-b'
\end{bmatrix}
\begin{bmatrix}
x
\end{bmatrix}
\leq
\begin{bmatrix}
0 \\
c
\end{bmatrix}.
$$

If the system is infeasible, then there exist vectors $\alpha, \beta, \gamma, \delta, \epsilon \geq 0$ for which

$$
\begin{bmatrix}
c' \\
-b'
\end{bmatrix}
\begin{bmatrix}
A' \\
0
\end{bmatrix}
\begin{bmatrix}
\alpha' \\
\beta' \\
\gamma' \\
\delta' \\
\epsilon'
\end{bmatrix}
= [0, 0], \quad \text{and} \quad
\begin{bmatrix}
\alpha' \\
\beta' \\
\gamma' \\
\delta' \\
\epsilon'
\end{bmatrix}
\begin{bmatrix}
A \\
0 \\
-A \\
-I \\
0
\end{bmatrix}
\begin{bmatrix}
\alpha \\
\beta \\
\gamma \\
\delta \\
\epsilon
\end{bmatrix}
< 0
$$

That is, $\alpha c' + (\gamma' - \delta') A = \epsilon'$, $A \beta = \alpha b$, and $c' \beta + (\gamma' - \delta') b < 0$. It follows that

$$
0 \leq \epsilon' \beta = (\alpha c' + (\gamma' - \delta') A) \beta = \alpha c' \beta + (\gamma' - \delta') \alpha b = \alpha (c' \beta + (\gamma' - \delta') b) \leq 0
$$

and thus we must have $\alpha = 0$. Since $c' \beta + (\gamma' - \delta') b < 0$, we have $c' \beta < 0$ or $(\gamma' - \delta') b < 0$. In the first case, since $A \beta = 0$, it contradicts the assumption that $y' A \leq c'$ is feasible. In the second case, since $(\gamma' - \delta') A = \epsilon' \geq 0$, it contradicts the assumption that $A x = b$, $x \geq 0$ is feasible.
Homework

SET 2  

Due Date: October 28, 1999

1. Given a polyhedron \(P = \{ x : Ax = b, \ x \geq 0 \}\) and a number \(\epsilon > 0\). Prove that there is a vector \(\overline{b}\) such that

(a) The absolute value of every component of \(b - \overline{b}\) is less than \(\epsilon\).
(b) Every basic feasible solution of \(P = \{ x : Ax = \overline{b}, \ x \geq 0 \}\) is nondegenerate.

2. Consider the following pivoting rule:

(a) Choose \(j\) such that \(c_j\) is the most negative.
(b) Choose \(l\) such that it is the smallest index with \(u_{l,j}/u_{l,0} = \theta^*\).

Explain what will happen to the following initial tableau if this rule is followed.

\[
\begin{array}{cccccccc}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\
3 & -3/4 & 20 & -1/2 & 6 & 0 & 0 & 0 \\
x_5 = 0 & 1/4 & -8 & -1 & 9 & 1 & 0 & 0 \\
x_6 = 0 & 1/2 & -12 & -1/2 & 3 & 0 & 1 & 0 \\
x_7 = 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\end{array}
\]

3. The big-M method works as follows. Consider the auxiliary problem

\[
\begin{align*}
\text{minimize} & \quad c'x + M'y \\
\text{subject to} & \quad Ax + y = b \\
& \quad x, y \geq 0
\end{align*}
\]

where \(I\) is the vector with all entries one, and \(M\) is an undetermined parameter. Whenever \(M\) is compared to another number, \(M\) will be always treated as being bigger. Clearly, we can choose the initial basis to be \(I\), and the initial basic feasible solution to be \(y = b\), and \(x = 0\). Prove that

(a) If the simplex method terminates with a solution \((x_0, y_0)\) for which \(y_0 = 0\), then \(x_0\) is an optimal solution to the original problem.
(b) If the simplex method terminates with a solution \((x_0, y_0)\) for which \(y_0 \neq 0\), then the original problem is infeasible.
(c) If the simplex method terminates with an indication that the optimal value of the auxiliary problem is \(-\infty\), then the original problem is either infeasible or its optimal value is \(-\infty\).
(d) Provide examples to show that both alternatives in part (c) are possible.

4. Let \(A\) and \(C\) be \(m_1 \times n\), and \(m_2 \times n\) matrices, respectively. Let \(b \in \mathbb{R}^{m_1}\), \(d \in \mathbb{R}^{m_2}\), and \(x \in \mathbb{R}^n\). Suppose \(Ax \geq b\) is feasible. Prove that the following are equivalent.

(a) \(Ax \geq b\), \(Cx > d\) is infeasible.
(b) There exist nonnegative vectors \(u \in \mathbb{R}^{m_1}\) and \(v \in \mathbb{R}^{m_2}\) such that

\[
u'A + v'C = 0, \quad u'b + v'd \geq 0, \quad \text{and} \quad v \neq 0
\]
Homework

Set 3

Due Date: 12/2/99

1. Let $A$ be the incidence matrix of connected directed graph $G = (N, E)$ and let $\tilde{A}$ be obtained from $A$ by deleting a row. Prove that, if $T = (N, F)$ is a spanning tree of $G$, then vectors in $\{\tilde{A}_f : f \in F\}$ are linearly independent.

2. Let $G = (N, E)$ be a directed graph, $s$, $t$ be two distinct nodes of $G$, and $u$ be a nonnegative vector in $\mathbb{R}^E$. A vector $x$ in $\mathbb{R}^E$ is an st-flow if $0 \leq x \leq u$ and

$$\sum_{(i,j) \in E} x_{ij} = \sum_{(j,i) \in E} x_{ji} \quad \text{for all } j \in N - \{s, t\}$$

Prove that, if $x$ is an st-flow and $(S, T)$ is a partition of $N$ with $s \in S$ and $t \not\in T$, then the value $x(S, T)$, which is defined as

$$x(S, T) = \sum_{(i, j) \in S, j \not\in T} x_{i, j} - \sum_{(i, j) \in S, j \not\in T} x_{j, i},$$

is independent of the partition $(S, T)$.

3. Prove that an initial basic feasible solution of the following optimization problem can be found by solving a maximum flow problem. Given a directed graph $G = (N, E)$ and vectors $b \in \mathbb{R}^N$, $c \in \mathbb{R}^E$, and $u \in \mathbb{R}^E$. The problem is to find a vector $x \in \mathbb{R}^E$ with $Ax = b$ and $0 \leq x \leq u$, where $A$ is the incidence matrix of $G$, such that $cx$ is minimized.

4. Use max-flow min-cut theorem to prove the following.

(a) Let $k$ be a positive integer and $G$ be a directed graph with two special nodes, $s$ and $t$. Suppose $G$ does not have $k$ arc-disjoint directed st-paths. Then there must exist a set $S$ of nodes for which $s \in S$, $t \not\in S$, and there are less than $k$ arcs $(i, j)$ with $i \in S$ and $j \not\in S$.

(b) Let $k$ be a positive integer and $G$ be a graph with two special nodes, $s$ and $t$. Suppose $G$ does not have $k$ edge-disjoint st-paths. Then there must exist a set $S$ of nodes for which $s \in S$, $t \not\in S$, and there are less than $k$ edges $ij$ with $i \in S$ and $j \not\in S$. 

23
Index

absolute value, problems with, 4
adjacent, 15
algorithm efficiency, 1
arc, 16
artificial variable, 13
assignment problem, 2
backward arc, 17
basic column, 10
basic feasible solution, 9
basic indices, 10
basic solution, 9
basic variable, 10
basis matrix, 10
Big-M method, 19
big-M method, 22
busing problem, 3
circulation, 18
closed walk, 16, 17
column vector, 4
connected graph, 16
covex, 8
convex hull, 15
cut edge, 18
cycle, 16, 17
cycling, 12
degenerate basic feasible solution, 10
directed cycle, 17
directed graph, 16
directed path, 17
directed walk, 17
discrete optimization, 1
Duality theorem, 8, 13
edge, 15
elementary matrix, 5
elementary operation, 5
ellipsoid method, 1
extent point, 8
feasibility problem, 6
feasible linear inequalities, 6
feasible solution, 9
forward arc, 17
Fourier-Motzkin elimination, 7
Gaussian elimination, 5
graph, 15
identity matrix, 5
incidence matrix, 17
incident, 15
incoming arc, 16
inner product, 4
integer programming, 1
interior point method, 1
inverse, 5
invertible matrix, 5
lexicographic pivoting rule, 12
linear combination, 4
linear programming, 1
linear regression problem, 4
linearly dependent, 5
linearly independent, 6
network flow problem, 17
node, 15
nonsingular matrix, 5
outgoing arc, 16
path, 16, 17
perpendicular, 4
pivot, 12
pivot column, 11
pivot element, 12
pivot row, 11
polyhedral cone, 15
polyhedron, 8
product-mix problem, 3
rank, 5
row vector, 4
simplex method, 1, 9
simplex tableau, 11
spanning tree, 16
standard form, 9
subgraph, 16
supergraph, 16
transpose, 4
tree, 16
underlying graph, 16
unit vector, 4
vertex, 9
vertex of a graph, 15
walk, 15–17