A First Attempt at an LP Algorithm

CS 149 Staff

September 27, 2006

Before we start developing an algorithm to solve Linear Optimization Problems, we must first define what it is we are trying to solve. We therefore define the Linear Optimization Problem to be a 2-tuple, (P, z), where $P \subset \mathbb{R}^n$ is the space of feasible solutions, and $z : P \to \mathbb{R}$ is the cost function.

The slides define two different kinds of instances of the Linear Optimization Problem, **Standard Form**, and **Canonical Form**. If $A \in \mathbb{R}^{m*n}, b \in R^m, c \in \mathbb{R}^n$, an instance of the linear optimization problem in standard form has:
$P := \{x \in \mathbb{R}^n | Ax = b \text{ and } x \geq 0\}$ and $z(x) := c^T x$.
An instance of the linear optimization problem in canonical form has:
$P := \{x \in \mathbb{R}^n | Ax \geq b \text{ and } x \geq 0\}$ and $z(x) := c^T x$.

We would like to be able to solve both of these kinds of problems, without having to write 2 different solvers. To do this, we must first define the relation between the 2 forms.

Definition We call 2 optimization instances (P_1, z_1), (P_2, z_2) **equivalent** if and only if there exist polynomial-time computable functions $f : P_1 \to P_2$ and $g : P_2 \to P_1$ such that:

$$\forall x_1 \in P_1, z_2(f(x_1)) \leq z_1(x_1) \forall x_2 \in P_2, z_1(g(x_2)) \leq z_2(x_2)$$

We will need a lemma to make our definition useful:
Lemma If (P_1, z_1) and (P_2, z_2) are equivalent and $x_1 \in P_1$ is optimal for (P_1, z_1), $f(x_1) \in P_2$ is optimal for (P_2, z_2) and $z_1(x_1) = z_2(f(x_2))$.
Proof: Let $x_1 \in P_1$ be optimal for (P_1, z_1), and let $y_2 \in P_2$.

$$\begin{array}{ccccccccl}
 & & z_1(x_1) & \leq & z_1(g(y_2)) & & & & \text{Because } x_1 \text{ is optimal} \\
z_2(f(x_1)) & \leq & z_1(x_1) & \leq & z_1(g(y_2)) & & & & \text{Equivalence} \\
z_2(f(x_1)) & \leq & z_1(x_1) & \leq & z_1(g(y_2)) & \leq & z_2(y_2) & & \text{Equivalence} \\
z_2(f(x_1)) & \leq & z_2(y_2) & & & & & &
\end{array}$$

So we have that $f(x_1)$ has lower cost than any $y_2 \in P_2$, so it is optimal. Further, since the inequality is true for any y_2, we can set $y_2 = f(x_1)$, and get:

$z_2(f(x_1)) \leq z_1(x_1) \leq z_2(f(x_1))$, so $z_1(x_1) = z_2(f(x_2))$.
Now that we have a definition of what eqivalence is, and a useful result, we can set about showing that our two forms are equivalent.

Lemma: The standard and canonical forms of the Linear Optimization Problem are equivalent.

Proof:
Let $(LP)_c$be a LO problem in canonical form:
$min\ c'x$
$Ax \geq b$
$x \geq 0$

Let $\tilde{A} = (A, -I), \tilde{b} = b, \tilde{c} = \binom{c}{0}, \tilde{x} = \binom{x}{x_s}$. The new problem:
$min\ \tilde{c}'\tilde{x}$
$\tilde{A}\tilde{x} \geq \tilde{b}$
$\tilde{x} \geq 0$

This problem is in standard form. Let us now show that the 2 forms are equivalent:
$\tilde{c}'\tilde{x} = c'x + 0'x_s = c'x$, so the objective function is the same.
$\tilde{A}\tilde{x} = \tilde{b} \Rightarrow Ax - x_s = b$. Since $x_s \geq 0$, Ax can take on any value greater than or equal to b.
Let $\tilde{x}$ be a solution of $(LP)_s$. Then x is a solution of $(LP)_c$.
First, we prove that x is feasible in $(LP)_c$.
$\tilde{x} \geq 0 \Rightarrow x \geq 0$.
$b = \tilde{A}\tilde{x} = Ax - x_s \Rightarrow Ax = b + x_s \Rightarrow Ax \geq b$.
Intuitively, we can see that the objective function is the same, and that the feasible region is the same as well. And since we have proved that the objective function is the same, we have our functions f and g that takes us from problems in $(LP)_c$to problems in $(LP)_s$.

Now we will construct the reverse mapping.
We have a problem in $(LP)_s$:
$min\ c'x$
$Ax = b$
$x \geq 0$
.
Our corresponding problem in $(LP)_c$is:
$min\ \tilde{c}'\tilde{x}$
$\tilde{A}\tilde{x} \geq \tilde{b}$
$\tilde{x} \geq 0$

$\tilde{A} = \binom{A}{-A}$

$\tilde{b} = \binom{b}{-b}$
$\tilde{c} = c.$
We are formulating $Ax = b$ as $Ax \geq b$ and $-Ax \geq -b$. It is pretty trivial to see that this works. Since x and z are the same, $f(x) = x$ and $g(x) = x$, and we can see that the 2 forms are equivalent.
This confirms our intuition, that the 2 forms are equivalent, and makes our lives significantly easier. Since it is more natural to express a problem in canonical form (think of a situation where we have limited quantites of raw materials that we could theoretically waste), we will find ourselves converting from canonical form to standard form a lot (since solvers and the Simplex algorithm take problems in standard form, hence the name), so x_s has a name. The memebers of x_s are called **slack variables**, and they represent the distance that we could travel towards a constraint before it becomes "tight" (ie we are on the hyperplane).
Linear Algebra Review
As a review, we will look at how to perform Gaussian Elimination and read out both a basic solution and the nullspace vectors. Consider the system: $Ax = b$, where:

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & -3 & 1 & 2 & 2 & 11 \\ 0 & -2 & -8 & 0 & -4 & 2 & 4 \\ -2 & 0 & -6 & 1 & 1 & 2 & 2 \\ 0 & -1 & -4 & 0 & -2 & 2 & 9 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 9 \\ 12 \\ 2 \\ 13 \end{bmatrix}.$$

We first convert this into an augmented matrix:

$$\begin{array}{ccccccc|c} \hline -1 & 0 & -3 & 1 & 2 & 2 & 11 & 9 \\ 0 & -2 & -8 & 0 & -4 & 2 & 4 & 12 \\ -2 & 0 & -6 & 1 & 1 & 2 & 2 & 2 \\ 0 & -1 & -4 & 0 & -2 & 2 & 9 & 13 \end{array}$$

We start Gaussian Elmination by putting a 1 in the first element (first row, first column), and putting zeros in all other elements in the column. This is called *pivoting* around the first element. To do this, we multiply the first row by -1 and subtract 2 times the first row from the third row.

$$\begin{array}{ccccccc|c} \hline 1 & 0 & 3 & -1 & -2 & -2 & -11 & -9 \\ 0 & -2 & -8 & 0 & -4 & 2 & 4 & 12 \\ 0 & 0 & 0 & -1 & -3 & -2 & -20 & -16 \\ 0 & -1 & -4 & 0 & -2 & 2 & 9 & 13 \end{array}$$

We now try to put a 1 in the second element of the second row (along a diagonal), with zeros everywhere else in the second column. We do this by multiplying the second row by $-\frac{1}{2}$, and subtracting $\frac{1}{2}$ times the second row from the first.

$$\begin{array}{ccccccc|c} \hline 1 & 0 & 3 & -1 & -2 & -2 & -11 & -9 \\ 0 & 1 & 4 & 0 & 2 & -1 & -2 & -6 \\ 0 & 0 & 0 & -1 & -3 & -2 & -20 & -16 \\ 0 & 0 & 0 & 0 & 0 & 1 & 7 & 7 \end{array}$$

The next element along the diagonal is zero, so we cannot pivot around this element. Instead, we can pivot around the third element in the 4th column. In your linear algebra class, you may have given up at this point, since we now know that the matrix is not invertible (though since it is not square we knew that to begin with), but here, we know that A is not invertible, but we still need to find a basis within it anyway. We pivot by multiplying row 3 by -1, and subtracting row 3 from row 1.

$$\begin{array}{ccccccc|c}
\hline
1 & 0 & 3 & 0 & 1 & 0 & 9 & 7 \\
0 & 1 & 4 & 0 & 2 & -1 & -2 & -6 \\
0 & 0 & 0 & 1 & 3 & 2 & 20 & 16 \\
0 & 0 & 0 & 0 & 0 & 1 & 7 & 7
\end{array}$$

Almost there. We have another zero in the last element of column 5, so we choose to pivot around the last element of column 6, which means we need to add row 4 to row 2, and subtract 2 times row 4 from row 3.

$$\begin{array}{ccccccc|c}
\hline
1 & 0 & 3 & 0 & 1 & 0 & 9 & 7 \\
0 & 1 & 4 & 0 & 2 & 0 & 5 & 1 \\
0 & 0 & 0 & 1 & 3 & 0 & 6 & 2 \\
0 & 0 & 0 & 0 & 0 & 1 & 7 & 7
\end{array}$$

We finally have an identity in our matrix! This is the end of our Gaussian Elimination! The columns that form the identity matrix are called *basic* columns, and *form a basis*. Since each variable corresponds to a column, the variables corresponding to the basic columns are the *basic variables*. We can read out a *basic solution* by setting all variables (remember that each column corresponds to a variable) not in the basis to zero, and putting the corresponding element of $\mathbf{b}$ as the value of each basic variable. Here, we have:

$$\mathbf{x_0} = \begin{bmatrix} 7 \\ 1 \\ 0 \\ 2 \\ 0 \\ 7 \\ 0 \end{bmatrix}$$

We now want to find the nullspace of the matrix to find the other solutions to the system (we have $\mathbf{Ax_0} = \mathbf{b}$, but $\mathbf{A}(\mathbf{x_0} + \mathbf{x_n}) = \mathbf{b}$ if $\mathbf{Ax_n} = \mathbf{b}$). We can find the vectores of the *nullspace* by setting one of the non-basic variables to 1, and all of the basic variables to the negative of the corresponding entry in the non-basic column. This will be clearer when we list the nullspace vectors:

$$\mathbf{y_0} = \begin{bmatrix} -3 \\ -4 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \mathbf{y_1} = \begin{bmatrix} -1 \\ -2 \\ 0 \\ -3 \\ 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{y_2} = \begin{bmatrix} -9 \\ -5 \\ 0 \\ -6 \\ 0 \\ -7 \\ 1 \end{bmatrix} \tag{1}$$

Intuitively, if we were to increase a non-basic variable, all of the basic variables would need to change by exactly the negative of the basic variable's change on their equation to maintain equality. So, all solutions to $\mathbf{A}\mathbf{x} = \mathbf{b}$ are: $x_0 + \lambda_0 y_0 + \lambda_1 y_1 + \lambda_2 y_2$.