CS 435 : Linear Optimization Fall 2006

Lecture 11: Simplex Algorithm: Finding a neighbour of larger cost

Lecturer: *Sundar Vishwanathan* Scribe: *Sanchit Garg*
Computer Science & Engineering Indian Institute of Technology, Bombay

1 Overview

1. Basic Approach for algorithm
   (a) Start with an extreme point.
   (b) Move to a neighbour of larger cost if one exists.
2. Issues with this approach
   (a) To find starting point
   (b) How to move from one extreme point to one of its neighbour (`Peberting`)
   (c) When algorithm stops, and to show that we have the point which maximizes $c^T A$
3. We have already accomplished following results in previous lectures
   (a) A linear function on $Ax \leq b$ is maximized at an extreme point.
   (b) The intersection of n linearly independent hyperplanes for equality (and hence corresponding n linearly independent rows of A) gives an extreme point if it satisfies the other inequalities.

2 Some statements/results

1. The intersection of n linearly independent rows/hyperplanes yield an extreme point. Also the intersection of $n-1$ linearly independent hyperplanes will give a line.
2. For each extreme point, there will be n neighbouring extreme points. One corresponding to each line given by different sets of $n-1$ hyperplanes excluding one from the n hyperplanes that define the extreme point in consideration.
3. The direction of the vectors from an extreme point to its neighbours are given by the columns of $-A'^{-1}$. A' is a $n \times n$ matrix such that $A'x_0 = b$.

   Proof: A' consists of n linearly independent rows which solve to give a unique solution x_0. Consider points on the line in between x_0 and one of its neighbour y_i. Since they all lie on line segment joining x_0 and y_i, they satisfy $n-1$ rows of A' with equality and remaining ones with inequality. Lets say i^{th} row of A' will be an inequality. Therefore points in between x_0 and y_i satisfy the following equations.

$$\begin{aligned}
A_i'y_i &< b_i \\
A_1'y_i &= b_1 \\
A_2'y_i &= b_2 \\
&\vdots \\
A_{i-1}'y_i &= b_{i-1} \\
A_{i+1}'y_i &= b_{i+1} \\
&\vdots \\
A_n'y_i &= b_n
\end{aligned}$$

Clearly, direction of vectors from x_0 to its neighbours will be given by

$$y_1 - x_0, y_2 - x_0, y_3 - x_0, \ldots, y_n - x_0 \tag{1}$$

where y_i is a neighbouring extreme point of x_0.
Consider a matrix P whose columns are given by $y_i - x_0$.

$$P = (y_1 - x_0, y_2 - x_0, y_3 - x_0, \ldots y_n - x_0) \tag{2}$$

$$\begin{aligned}
&Since\ {A'}_j y_i = b_j = {A'}_j x_0, \forall j \neq i\ (i, j = 1 \ldots n) \\
&\Rightarrow {A'}_j (y_i - x_0) = 0, \forall j \neq i
\end{aligned} \tag{3}$$

$$\begin{aligned}
&Also\ {A'}_i y_i \leq b_i \\
&\&\ {A'}_i x_0 = b_i \\
&\Rightarrow {A'}_i (y_i - x_0) \leq 0
\end{aligned} \tag{4}$$

This gives $A'P$ as

$$\begin{bmatrix}
(-)ve & 0 & 0 & \ldots & 0 \\
0 & (-)ve & 0 & \ldots & 0 \\
\vdots & & & & \\
0 & 0 & 0 & \ldots & (-)ve
\end{bmatrix}$$

Now $A'(-A'^{-1})$ would be

$$\begin{bmatrix}
-1 & 0 & 0 & \ldots & 0 \\
0 & -1 & 0 & \ldots & 0 \\
\vdots & & & & \\
0 & 0 & 0 & \ldots & -1
\end{bmatrix}$$

This means that $y_i - x_0$ is a scaled version of the i^{th} column of $-A'^{-1}$. Hence the direction vectors of the neighbours of x_0 are given by the columns of $-A'^{-1}$.

3 Simplex Algorithm

Recall that we started with an extreme point and if there exists a neighbour of larger cost we moved to that neighbour. To check if x_0 has a neighbour y_i with larger cost we need to determine if the cost

increases in the direction of $y_i - x_0$ (which is given by i^{th} column of $-A'^{-1}$). If for any y_i this cost increases then y_i is our next extreme point and we repeat the procedure. The Algorithm stops when we have no such neighbour, and the output is the final extreme point.
If we have found that the cost increases in the direction of $y_i - x_0$, we need to find y_i to proceed further. Since x_0 & y_i lie on same line, they have $n - 1$ hyperplanes in common. To find y_i we need to get the n^{th} hyperplane. Consider

$$A(x_0 + \epsilon v_i) \leq b \tag{5}$$

where v_i is the direction vector $(y_i - x_0)$. With $\epsilon = 0$, we have $A'x_0 = b$. As we gradually increase ϵ, at some value one of the inequalities will become an equality. This yields the point y_i.
Note: There would be many hyperplanes that would solve with these $n - 1$ hyperplanes to give a point. But they wont be extreme point as they might not satisfy other inequalities. Only, point closest to x_0 on line will satisfy all conditions and it would be an extreme point.

What remains is to prove that if all the neighbours of x_0 have cost at most the cost of x_0, then x_0 will be a point of maximum cost. This will be taken in the next lecture.