Comp 163: Computational Geometry
Tufts University, Spring 2004
Professor Diane Souvaine
Original Scribe: Chung Peng
Current Scribe: Douglas Stetson

Linear Programming in 2D

1 Linear Programming in Multiple Dimensions

Linear programming is an algorithm for maximizing the value of some equation subject to a set of constraints. The constraints are assumed to be a set of linear equations in d dimensions. Linear programming in multiple dimensions is solved through a recursive process of solving the sub-problem in the lower next dimension. Problems in all dimensions decompose in this manner until the base case is reached - the problem broken down into 2D. So these notes will focus on the 2D case.

For example, to solve a linear programming problem in 4D, one would solve the sub-problem of it in 3D, and to solve it in 3D, one needs to first solve it in 2D.

2 Linear Programming in 2D

A problem in 2D has the following general format:

$$min_{x_1,x_2} : c_1x_1 + c_2x_2$$

$$\ni \alpha_{i,1}x_1 + \alpha_{i,2}x_2 \geq B_i$$

where i = 1,..., n constraints, and $\ni$ denotes the set of constraints. This is converted by a transformation of the axis into the following more easily manipulated form. We transfer from the x_1, x_2 axis to the x,y axis, where the x axis is parallel to $c_1x_1 + c_2x_2 = 0$. This changes the constraints to:

$$\ni y \geq \alpha_i x + b_i, where\ i \in I_1$$

$$y \leq \alpha_i x + b_i, where\ i \in I_2$$

$$a \le x \le b$$

where $y = \alpha_{i,1}x_1 + \alpha_{i,2}x_2$. I_1 is the set of lines from which the lower hull of the feasible region is formed, I_2 is the set for the upper hull. In Figure 1, I_1 would be the set of lines which are at least partially dotted, I_2 would be the set of lines which are at least partially dashed, and the feasible region is would be the central polygon. The feasible region is the set of points that satisfy the constrains, and could be the minimum solution. We will also talk about feasible value of x, which is simply a value of x for which there is some value of y such that the point (x,y) is in the feasible region.

Sample problem: find $min_{x_1,x_2} : 5x_1 - 2x_2$

$$\ni 3x_1 + 5x_2 \ge 1$$
$$-4x_1 + 5x_2 \ge 5$$
$$-2x_1 - 5x_2 \ge 2$$

The number of constraints in the sets I_1 and I_2 are as follows:

$$|I_1| + |I_2| \le n$$
$$|I_1| + |I_2| + |number\ of\ vertical\ lines| = n$$

2.1 g(x) and h(x)

g(x) is defined as the maximum of the constraints in set I_1. Likewise, h(x) is defined as the minimum of the constraints in set I_2.

$$g(x) = max\{\alpha_i x_i + b_i | i \in I_1\}$$
$$h(x) = min\{\alpha_i x_i + b_i | i \in I_2\}$$

Both of the above functions are piece-wise linear. The constraints can now be simplified to as follows:

$$\ni g(x) \le h(x)$$
$$a \le x \le b$$

These functions are shown graphically in figure 1. The dashed line is the function g(x), the dotted line is the function h(x). On the left side, they meet where x=a, on the right where x=b. This picture is of course only one possible image, there is no guarantee the lines ever meet, or the a and b are finite.

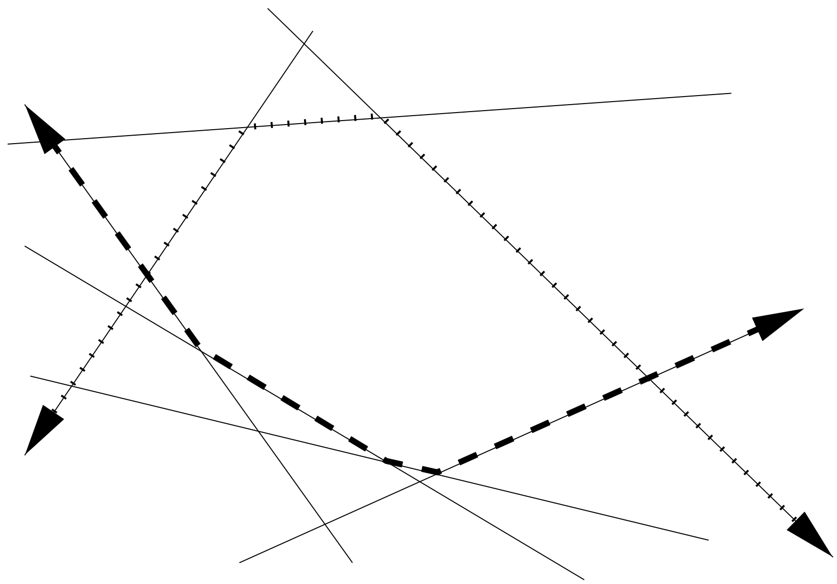

Figure 1: Sample constraints

2.2 Feasible and infeasible

As explained above, the feasible region is the set of all points that meet the constraints. If we can find the feasible region, then finding the point that minimizes (or maximizes) the solution is relatively easy. The following are some facts about feasibility which we will rely on in the following sections. We seek to find x*, the optimal (we'll assume minimal) solution. We denote x' as the value of x that is under consideration:

1. A given value x' of x, $a \leq x' \leq b$ is feasible iff $g(x') \leq h(x')$.

2. If x' is NOT feasible, any feasible values must lie to one side of x'. Since the feasible region is convex, and feasible values of x are bounded by a and b (where either or both can be infinite), a point x' outside the region must be either to the right or left of the region.

3. If x' is feasible, we will test whether x' is optimal, or if not, determine on which side of x' the minimum lies.

4. Define $f(x) = g(x) = h(x)$. For feasible x, $f(x) \leq 0$

5. For an infeasible x', $f(x') > 0$ and $g(x') > h(x')$

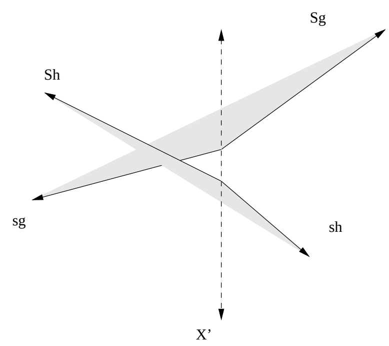

Figure 2: $s_g(x') > S_h(x')$

2.3 $s_g(x')$, $S_g(x')$, $s_h(x')$ and $S_h(x')$

Let $s_g(x')$ and $S_g(x')$ denote the min and max edge from $g(x')$

$$s_g(x') = min\{a_i : i \in I_1, a_i x' + b_i = g(x')\}$$

$$S_g(x') = max\{a_i : i \in I_1, a_i x' + b_i = g(x')\}$$

Also let $s_h(x')$ and $S_h(x')$ denote the min and max edge from $h(x')$

$$s_h(x') = min\{a_i : i \in I_2, a_i x' + b_i = h(x')\}$$

$$S_h(x') = max\{a_i : i \in I_2, a_i x' + b_i = h(x')\}$$

2.4 Finding the optimal solution x*

By comparing the values of $s_g(x')$, $S_g(x')$, $s_h(x')$ and $S_h(x')$, one can figure out where $x*$ could possibly exist in relation to x'.

1. If $s_g(x') > S_h(x')$, then x* lies to the left of x' (Figure 2)
2. If $S_g(x') > s_h(x')$, then x* lies to the right of x' (Figure 3)
3. If $s_g(x') - S_h(x') \leq 0 \leq S_g(x') - s_h(x')$, then x* is infeasible (Figure 4)

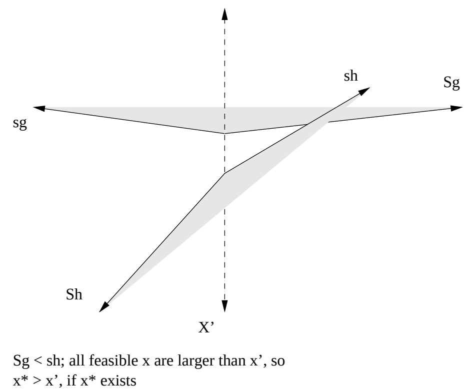

Figure 3: $S_g(x') > s_h(x')$

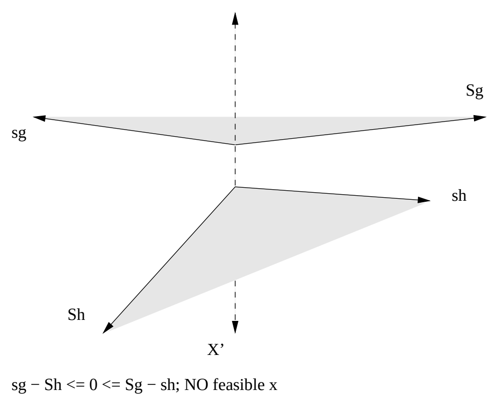

Figure 4: $s_g(x') - S_h(x') \leq 0 \leq S_g(x') - s_h(x')$

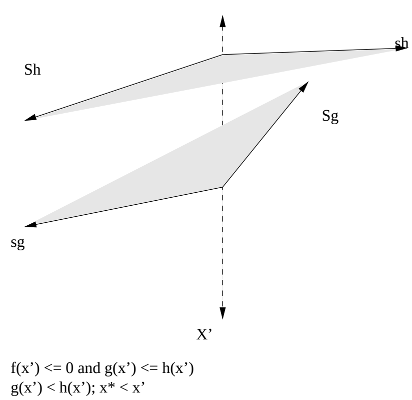

Figure 5: $x* < x'$

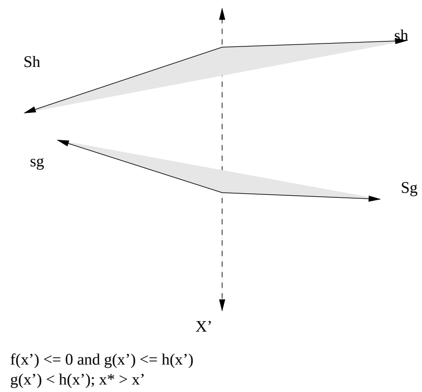

Figure 6: $x* > x'$

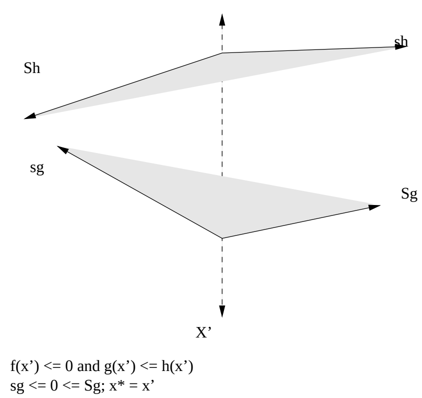

Figure 7: $x* = x'$

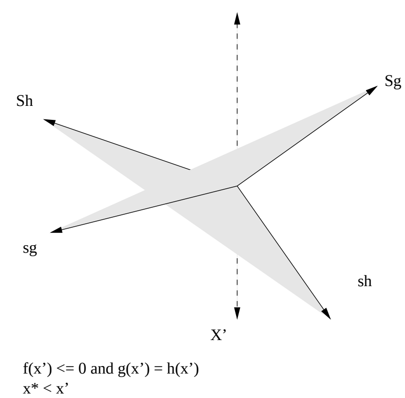

Figure 8: $x* < x'$

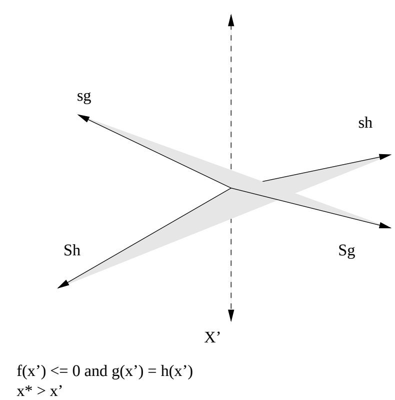

Figure 9: $x* > x'$

For x' to be in a feasible region, the following has to be true:

$$f(x') \leq 0 \ and \ g(x') \leq h(x')$$

Once a feasible region has been found, then you can know that x* exists. If $g(x') < h(x')$, then there are 3 possible cases for where x* is:

1. $x* < x'$ (Figure 5)
2. $x* > x'$ (Figure 6)
3. $x* = x'$ (Figure 7)

And if g(x') = h(x') then

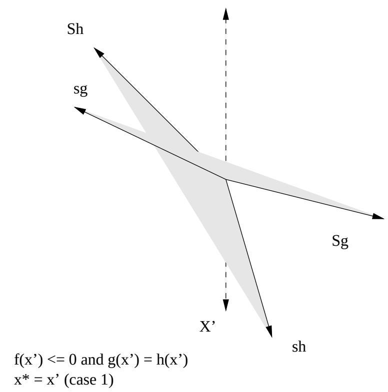

Figure 10: $x* = x'$

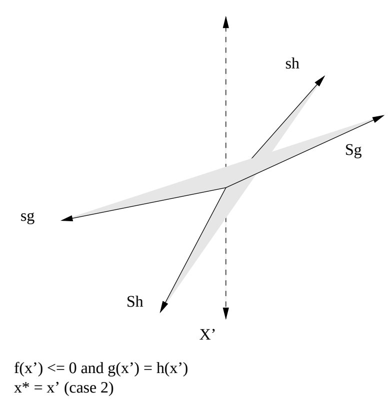

Figure 11: $x* = x'$

1. $x* < x'$ (Figure 8)

2. $x* > x'$ (Figure 9)

3. $x* = x'$ (Figure 10) or (Figure 11)

We can conclude the following theorem from the above cases:

Theorem 2.1 *If x' in [a, b] then $O(n)$ suffices to decide all of the following*

1. *if the problem is infeasible*

2. *if $x* = x'$*

3. *if $x*$ in $[a, x']$ if there exists $x*$*

4. *if $x*$ in $[x', b]$ if there exists $x*$*

2.5 Algorithm

Arrange the elements of I_1 (I_2 is handled likewise) in disjoint pairs. One of the following cases will hold:

1. If $a_i = a_j$ in pair (i, j), drop the redundant line. In other works, if the slopes of the two lines are the same, drop the one with the smaller y intercept from I_1 (or the larger from I_2)

2. Compute $x_{ij} = b_i - b_j/a_j - a_i$, the intersecting point of the 2 lines where $a_j > a_i$

   (a) If $x* \in [a, x_{ij}]$ and i, j in I1, line j is redundant

   (b) If $x* \in [x_{ij}, b]$ and i, j in I1, line i is redundant

   (c) If $x* \in [a, x_{ij}]$ and i, j in I2, line i is redundant

   (d) If $x* \in [x_{ij}, b]$ and i, j in I2, line j is redundant

The notation may be a little confusing here, remember that a and b are the minimum and maximum feasible values of x', whereas a_i is the slope of the *ith* line and b_i is the y-intercept of the *ith* line.

Next find the mediam value x_m of x_{ij} and choose the appropriate interval $[a, x_m]$ or $[x_m, b]$ containing $x*$. The correct interval contains at most half of

x_{ij}. For each of the remaining x_{ij}'s not thrown out, check the contraints as discussed above.

Since the algorithm removes 1/4 of the constraints each time, it runs in $T(n) = cn + T(3n/4) = O(n)$