NOTES ON LINEAR OPTIMIZATION
(outline of an important application of linear systems)

'Linear optimization' refers to maximizing/minimizing linear functions of many variables (say, n) subject to linear constraints (say, m). The constraints are given by linear inequalities; usually $m < n$, i.e. the number of variables is greater than the number of constraints (often much greater.)

In applications these problems arise when one needs to allocate limited resources optimally (something we all need to do!) We seek to maximize/minimize a 'utility function' F of the allocations $x_1, x_2, \dots, x_n$. (F could mean profit, loss, pleasure, etc.)

$$F = c_1x_1 + c_2x_2 + \dots c_nx_n + L$$

The constraints are of two types. First, the allocated amounts cannot be negative:

$$x_i \geq 0, \quad i = 1, \dots, n.$$

They also have to satisfy m inequalities of the type:

$$a_{i1}x_1 + a_{i2}x_2 + \dots a_{in}x_n \leq b_i, \quad i = 1, \dots m.$$

These represent limitations in cost, risk, transportation capacity, whatever.(We can deal with inequalities $\geq$ by flipping signs in the corresponding equations.)

Here is a very simplistic example: a man wants to divide his 10 days of vacation in the year between days at the beach and days in the mountains. It takes him a whole day to drive to the beach, so the cost in days of x_2 days at the beach is $2x_2$; on the other hand, a day in the mountains costs \$900 and a day at the beach only \$300 (including the cost of getting there); this unfortunate person has only \$4500 in his vacation budget, and feels a day in the mountains is worth twice as much as a day at the beach. How can he maximize his pleasure under such stark conditions? (He may not be able to use all his vacation days!)

With x_1, x_2 denoting the number of mountain (resp. beach) days, he seeks to maximize $F = 2x_1 + x_2$, subject to the constraints:

$$\begin{cases} x_1 + 2x_2 & \leq 10 \\ 3x_1 + x_2 & \leq 15 \\ x_1 \geq 0, & x_2 \geq 0 \end{cases}$$

We can always replace the inequalities by equalities by introducing 'slack variables'. In this example (where the name 'slack' is appropriate) the system above is clearly equivalent to:

$$\begin{cases} x_1 + 2x_2 + s_1 & = 10 \\ 3x_1 + x_2 + s_2 & = 15 \\ \quad x_i \geq 0, & s_i \geq 0 \end{cases}$$

(The price for the slack is having to work in $\mathbb{R}^4$, rather than $\mathbb{R}^2$.)

Solving this max/min problem has two parts: understanding the geometry of the solution set of the constraints (i.e., positive solutions of a linear system) and finding the max/min values. Once the first part has been done, the second is very easy.

You will recall that, in general, the solution set of a non-homogeneous linear system in n variables is a 'plane' (not through the origin) of dimension $n-r$, where r is the rank of the matrix defining the system. If we require all the variables to be non-negative, we get an $n-r$ dimensional *convex subset* of $\mathbb{R}^n$.

Definition. $C \subset \mathbb{R}^n$ is *convex* if given any $p, q \in C$, the line segment joining them is entirely contained in C.

Examples of convex sets were shown in class.

(We always take C to be a *closed set* i.e., it includes all its boundary points.) It is easy to see that the solution set C of a system of the form:

$$Ax \leq b, \quad x \geq 0$$

where A is an $m \times n$ matrix, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, is a convex subset of $\mathbb{R}^n$. (We define $v \leq w$ for vectors v, w to mean $v_i \leq w_i$ for all i). Indeed the line segment from $p \in \mathbb{R}^n$ to $q \in \mathbb{R}^n$ may be parametrized by:

$$p + t(q-p) = (1-t)p + tq, \qquad t \in [0,1],$$

where in the second way of writing the parametrization it is evident that the coefficients of p and q are *positive* (or non-negative, to be precise.) Thus we are allowed to multiply inequalities by $1-t$ and t, and obtain, assuming $p, q \in C$:

$$A((1-t)p + tq) = (1-t)Ap + tAq \leq (1-t)b + tb = b,$$

showing the points in the segment from p to q are also in C (it is clear that $(1-t)p_i + tq_i \geq 0$ if $p_i \geq 0$ and $q_i \geq 0$).

If you draw some convex polygons, you will notice they differ from convex regions with smooth boundary by having ‘vertices’. The formal name for vertices is ‘extreme points’.

Definition. An *extreme point* p of a convex set $C \subset \mathbb{R}^n$ is a boundary point of C with the property: if p is in any line segment σ entirely contained in C, then p is an endpoint of σ.

(Check that vertices of convex polygons have this property, and also that *every* boundary point of a convex region with smooth boundary is an extreme point.)

Extreme points are important for the following reason: every max or min of a linear function on a convex set is achieved at an extreme point. Put more formally:

Theorem. Let $C \subset \mathbb{R}^n$ be convex. If a linear function F as above achieves its max (or min) over C at a point of C, then it must achieve this max (or min) value at an extreme point of C.

Note that if C is not bounded, the max (and/or the min) may not be achieved at all; when it is, it may also be achieved at non-extreme points. This ‘theorem’ is (geometrically) almost obvious, since the level sets of F are ‘hyperplanes’ (planes of dimension $n-1$), and we can just move these hyperplanes in from far away, until they first ‘touch’ C- necessarily at an extreme point.

Fortunately there is a simple way to find all the extreme points of a convex set defined, as above, as the solution set of $m+n$ inequalities (some of which may be equalities; n of these inequalities are just the positivity requirement: $x_i \geq 0$.)

To find the extreme points: set n of the $m+n$ inequalities to equality (by turns) and solve the corresponding $n \times n$ system. Then check that the solution satisfies the remaining inequalities (and discard it if it doesn’t).

Exercise. Use this method to find the extreme points of the region defined by the system in the ‘vacation’ example above (without the ‘slack’ variables), and sketch its solution set.

Once the extreme points have been found, we can then take any linear function (like $2x_1 + x_2$ in the example), compute its value at each extreme points, and compare values to find the max/min.

That is, as long as we have 2 variables, 2 constraints and 4 extreme points! Already with 5 variables and 2 equality constraints we get a possible total of 10 extreme points, and the number grows rapidly. (With m

equality constraints and n variables, the number of possible extreme points is: $\frac{n!}{m!(n-m)!}$.) In application areas like economics, the military, transportation, etc, where the number of variables is large, this elementary method is not practical, and we need a new idea.

The 'new idea' is called *the simplex method*, invented (discovered?) by G.Dantzig in the late 1940s (under a USAF research grant) That's a long time ago- so this is 'old hats' as mathematics, but still very useful. In a nutshell, the simplex algorithm searches for extreme points and evaluates the utility function we seek to minimize simultaneously, using the following basic observation:

For a linear function in a convex set, any ***local*** *min (or max) is automatically a* ***global*** *min (resp. max).*

(You may remember from Calculus this is in general false for non-linear functions.)

The algorithm works as follows:

(1) Pick any extreme point of C (by setting $n - m$ of the variables x_i to zero) and evaluate F there;

(2) Find the adjacent extreme points: increase one of the variables originally set to zero as far as allowed by the constraints (that is, until one of the positive variables is forced to be zero). Evaluate F at these new extreme points; if no smaller values of F are found, we were lucky: the extreme point picked initially was already the global min.

(3) Otherwise, as soon as you find a smaller value of F, move to that extreme point and start over (that is, try to decrease F by searching *its* neighboring extreme points- you can skip the one that you just came from!)

This is best seen by trying a couple of examples:

Example. Minimize $F = 6x_1 + 3x_2 + x_3 + 7x_4 + 8x_5$, subject to:

$$x_1 + 2x_2 + 7x_3 = 9, \quad 6x_2 + 3x_3 + x_4 - x_5 = 9, \quad x_i \geq 0.$$

Example/exercise. Minimize $F = 11 - x_3 - x_4 - x_5$, subject to:

$$x_1 + x_3 - x_4 + 2x_5 = 2, \quad x_2 - x_3 + 2x_4 + x_5 = 1, \quad x_i \geq 0.$$

(The answer is between 1 and 8.)

Exercise. Don't forget to solve the vacation problem!

Solution of the first problem. The system defining the constraints has rank 2 in five variables, so the general solution depends on 3 parameters. The 'feasible set' C is a three-dimensional convex subset of $\mathbb{R}^5$. It is not bounded, since we can increase x_5 without violating the second constraint, simply by increasing x_4 by the same amount; so F does not have a *max* in C, but it does have a *min*.

The extreme points (=vertices) of C are obtained by setting three of the coordinates equal to zero; there are $5 \times 4/2 = 10$ possibilities. What is a good place to start? Since we are trying to minimize F, it makes sense to set to zero the variables with largest coefficient: $x_1 = x_4 = x_5 = 0$; solving for the two remaining variables we get the vertex (0,1,1,0,0), where F takes the value 4- this is the number to beat, by looking at the neighboring vertices.

Increase the value of x_1 as far as possible, keeping $x_4 = x_5 = 0$. We can increase x_1 until either $x_2 = 0$ or $x_3 = 0$. In the first case, we get $x_3 = 3$, forcing $x_1 < 0$ (not allowed); while if $x_3 = 0$ we get $x_2 = 3/2, x_1 = 6$; at this vertex clearly $F > 36$, so we drop it.

As we continue this, it is best to record the results in a table (see below). Now we try to increase x_4, keeping $x_1 = x_5 = 0$. With $x_2 = 0$, we get the vertex $(0, 0, 9/7, 36/7, 0)$, where again $F > 36$, no good. With $x_3 = 0$, we get $x_2 = 9/2$, $27 + x_4 = 9$, forcing x_4 to be negative. So nothing here, either.

Finally, trying to increase x_5 with $x_1 = x_4 = 0$, we find if $x_2 = 0$: $x_3 = 9/7, x_5 < 0$ (bad), and if $x_3 = 0$: $x_2 = 9/2$, $x_5 = 18$, which makes F too big. So the original vertex is the winner, and the minimum value of F is 4.

TABLE

action	x_1	x_2	x_3	x_4	x_5	F
start	0	1	1	0	0	4
$x_1 up$	–	0	3	0	0	?
$x_1 up$	6	3/2	0	0	0	big
$x_4 up$	0	0	9/7	36/7	0	big
$x_4 up$	0	9/2	0	–	0	?
$x_5 up$	0	0	9/7	0	–	?
$x_5 up$	0	9/2	0	0	18	big

Conclusion: there are three vertices adjacent to the starting one, and the value of F is higher in each one (we were lucky). We did have to search a total of 7 vertices (of a possible 10), so this may not look like much economy- but the search method was systematic enough to be programmable.