Karmarker’s Method

• Karmarker’s method is a special case of an interior point method for solving linear programming problems.

• Instead of moving from one vertex of the feasible region to another we move along a sequence of points in the interior of the feasible region.

• Karmarker’s method is an iterative technique, as opposed to the simplex method which is a finite procedure.

• The development of Karmarker’s method caused a revolution in optimization, largely because Karmarker proved that even in the worst case it requires a numerical effort that is a polynomial in \( m \) and \( n \), as opposed to an exponential for the simplex method\(^{-1}\).

• See the article by Margaret Wright for more info.

\(^{-1}\) Think about what it means for an iterative method, which in principle requires infinitely many steps, to terminate in polynomial time.

• The simplest description of Karmarker’s method, in my opinion is:


• Strang’s book used to be our textbook for Math 5710.

• Note on the term “interior”. When we have equality constraints then the feasible region forms the intersection of the first octant and an affine subspace of \( \mathbb{R}^n \). In that case, strictly speaking every feasible point is on the boundary of the feasible region. However, in this context, we say that a point \( x \) is in the “interior” of the feasible region if all of its components are positive. \( x \) still needs to satisfy the equality constraints.

• We consider again the **canonical problem**

\[
    c^T x = \min \quad \text{subject to} \quad Ax = b \quad \text{and} \quad x \geq 0.
\]

(1)

• (We don’t need the requirement that \( b \geq 0 \).)

• Following Strang, we denote iterates by super-
scripts. Thus we are looking for a sequence

\[ x^0, x^1, x^2, \ldots \]  

(2)

that converges to the solution.

**Key Ideas**

1. Move in the direction that decreases \( c^T x \) as quickly as possible.

2. Stop short of the boundary.

3. After each step, rescale your problem suitably such that the new point is again “deep in the interior” of the feasible region.

4. Stop when \( n - m \) components of the unscaled version of \( x^k \) are sufficiently small, otherwise repeat (1)-(3).

5. Plus many refinements and modifications.

   - Suppose we have a point \( x^0 \) in the “interior” of the feasible region. This means

     \[ x^0 > 0, \]  

     (3)

     i.e., all components of \( x^0 \) are positive. Recall that superscripts denote iterates.

   - The direction \( d \) of quickest change is the unit vector \( d \) minimizing \( c^T d \).

   - That direction is

     \[ d = \frac{-c}{\|c\|_2}. \]  

     (4)
• Another view: to minimize $F(x) = c^T x$ go in the direction of steepest descent, i.e., in the direction of the negative gradient:

$$-\nabla F(x) = -c.$$  \hspace{1cm} (5)

• However, moving in that direction will usually get us out of the feasible region!

• Suppose we go from one feasible vector $x^0$ to another feasible vector $x^1$.

• Then

$$Ax^0 = Ax^1 = b$$  \hspace{1cm} (6)

and hence

$$A(x^1 - x^0) = 0.$$  \hspace{1cm} (7)

• **We must move in a direction that is in the null space of $A$.**

• So we project our steepest descent direction into the null space of $A$. 
We will move in the direction \( d = -Pc \) where \( d \) is obtained by projecting the steepest descent direction \(-c\) into the null space of \( A\):

\[
d = -Pc = -(I - A^T(AA^T)^{-1}A)c \quad (8)
\]

- Of course we don’t actually compute the inverse.
- To see that this works note that

\[
-AP = A - AA^T(AA^T)^{-1}A = 0 \implies -APc = 0,
\]

i.e., \(-Pc\) is in the null space of \( A\). Moreover,

\[
P^2 = (I - A^T(AA^T)^{-1}A)^2
= I - 2A^T(AA^T)^{-1}A + A^T(AA^T)^{-1}A
= I - 2A^T(AA^T)^{-1}A + A^T(AA^T)^{-1}A
= I - A^T(AA^T)^{-1}A
= P,
\]

i.e., \( P \) is a projection.

- Of course we don’t actually compute the inverse. Instead we solve the square (positive definite) system

\[
AA^T y = Ac \quad (11)
\]

for

\[
y = (AA^T)^{-1}Ac \quad (12)
\]
and let
\[ d = -c + A^T y. \] \hfill (13)

- Then we move in the direction \( d \), i.e.,
\[ x^1 = x^0 + kd \] \hfill (14)
for some suitable \( k > 0 \).

- This will reduce the value of the objective function because
\[ c^T kd = c^T kP(-c) = -kc^T Pc < 0. \] \hfill (15)

- Exercise: How do we know that \( c^T Pc \) is positive?

- By going far enough some component of \( x^1 \) will become 0. We are then on the “boundary” of the feasible region. Let
\[ \Delta x = kd \] \hfill (16)
for the value of \( k \) for which this first happens.

- (If one or more components of \( x^1 \) are negative then \( x^1 \) is not feasible.)

- As mentioned earlier, we do not go all the way to the boundary. Instead we stop short of it! We let
\[ x^1 = x^0 + \alpha \Delta x \] \hfill (17)
where \( \alpha \) is a parameter satisfying
\[ 0 < \alpha < 1. \] \hfill (18)
• Karmarker used

\[ \alpha = 0.25 \tag{19} \]

in his proof of polynomial effort.

• Strang suggests the choice

\[ \alpha = 0.98 \tag{20} \]

in his discussion.

• To obtain the next iterate, \( x^2 \), it would be pointless to repeat this step because we would be moving in the same direction.

• Instead, we rescale the variables so that \( x^1 \) is “deep in the interior”.

• This means that all of its components are equal, and equal to 1, say.

• We use the diagonal scaling matrix:

\[
D = \begin{bmatrix}
  x_1^1 & 0 & 0 & \cdots \\
  x_2^1 & 0 & 0 & \cdots \\
  & \ddots & \ddots & \cdots \\
  0 & 0 & \cdots & x_n^1
\end{bmatrix} \tag{21}
\]

• Denote the new variable by \( X \):

\[ X = D^{-1}x \quad \text{and} \quad x = DX. \tag{22} \]
• Note that

\[ X^1 = D^{-1} x^1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}. \]  

(23)

We are again “deep in the interior”.

• The objective function, in terms of \( X \), becomes

\[ c^T x = \bar{c}^T X \]  

(24)

where

\[ \bar{c} = c^T D \]  

(25)

is the new vector of cost coefficients.

• The constraints, in terms of \( X \), become

\[ Ax = ADX = b \]  

(26)

where \( AD \) is the new constraint matrix.

• We then repeat the step!

• Stopping is complicated.

**Summary**

**The Problem.** The basic problem is

\[ c^T x = \min \quad \text{subject to} \quad Ax = b, \quad x \geq 0 \]  

(27)

where \( A \) is an \( m \times n \) (real) matrix (of full rank), \( c \in \mathbb{R}^n \) and \( b \in \mathbb{R}^m \) are given vectors, and \( m \leq n \).
Starting. We assume we are given a starting point \( x^0 \) in the interior of the feasible region. This means

\[
Ax^0 = b \quad \text{and} \quad x^0 > 0.
\]  

(28)

We set \( k = 0 \) and throughout denote the current iterate by \( x^k \).

Rescaling. We rescale the variables so that the current point is “deep in the interior” of the rescaled feasible region. The interpretation of this phrase in this context is that all components of \( x \) are the same (and equal to 1, say). Then none is closer to zero than any of the others. So we use a diagonal scaling matrix \( D \) to rephrase the problem in terms of a new variable \( X \) as follows:

\[
x = DX \quad \text{i.e.} \quad X = D^{-1}x
\]  

(29)

where

\[
D = \text{diag}(x) = \begin{bmatrix} x_1^k & & \\ & x_2^k & \\ & & \ddots \\ & & & x_n^k \end{bmatrix}
\]  

(30)

It is clear that (27) is equivalent to

\[
c^TDX = \min \quad \text{subject to} \quad ADX = b, \quad X \geq 0.
\]  

(31)

Note that since \( D \) is diagonal it is symmetric and \( D = D^T \). However, for clarity in what follows we write \( D^T \) when appropriate.
Projecting the direction of steepest descent. As described by Strang, the direction of steepest descent (i.e., $-D^Tc$) is projected into the feasible region, to give a direction

$$p = P(D^Tc) = [I - D^TA^T(ADD^TA^T)^{-1}AD] D^Tc.$$  \hspace{1cm} (32)

This equation can be obtained by replacing $A$ in (13) (in Strang’s book) with $AD$ and then multiplying the resulting $P$ with the replacement of $c$, i.e., $A^Tc$.

Computing the Projection. A simple way of computing $p$ is by following this two step procedure:

1. Solve

$$(ADD^TA^T)y = ADD^Tc.$$  \hspace{1cm} (33)

2. Set

$$p = D^Tc - D^TA^Ty.$$  \hspace{1cm} (34)

Note that (33) amounts to solving a Least Squares Problem with a design matrix $(AD)^T$, i.e., $y$ minimizes

$$\|(AD)^Ty - D^Tc\|_2.$$  \hspace{1cm} (35)

As we discussed earlier, a good way to solve (35) is via the QR factorization of $(AD)^T$. (See also Strang’s section entitled Numerical Linear Algebra.) The Choleski decomposition of $ADD^TA^T$ can also be used, but it’s not as stable since the condition number of $ADD^TA^T$ is the square of that of $(AD)^T$.
**Taking the Step.** As discussed in Strang, we now define

\[ X^{k+1} = X^k - sp = e - sp \]  \hspace{1cm} (36)

where

\[ e = [1 \: 1 \: \cdots \: 1]^T \in \mathbb{R}^n. \]  \hspace{1cm} (37)

and \( s \) is chosen such that the smallest component of \( X^{k+1} \) is just barely positive.

---

**The algorithm.** A summary of the above points is as follows:

1. Given \( x^0 > 0, \: Ax^0 = b \).

2. For \( k = 0, 1, 2, \cdots \) until satisfied do:
   
   a. Define \( D = \text{diag}(x^k) \).
   
   b. Solve \((ADD^T A^T)y = ADD^T c\).
   
   c. Let \( p = D^T c - D^T A^T y \).
   
   d. Let

   \[ s = \frac{\alpha}{\max_{i=1\ldots n} p_i} \]  \hspace{1cm} (38)

   (where Strang suggests \( \alpha = 0.98 \)).
   
   e. Let

   \[ X^{k+1} = e - sp \hspace{0.5cm} \text{and} \hspace{0.5cm} x^{k+1} = DX^{k+1}. \]  \hspace{1cm} (39)

---

**Obtaining** \( x^0 \). To obtain \( x^0 \) solve the artificial problem using the above algorithm

\[ \gamma = \min \hspace{0.5cm} \text{subject to} \hspace{0.5cm} Ax + \gamma(b - Ae) = b, \hspace{0.5cm} x \geq 0, \hspace{0.5cm} \gamma \geq 0 \]  \hspace{1cm} (40)
with the initial approximation

\[ x = e, \quad \gamma = 1. \quad \text{(41)} \]

This works because the optimal solution of the artificial problem will have \( \gamma = 0 \) which means \( Ax = b \).

**Stopping.** This is the trickiest part of the project and requires some experimentation. Usually the solution of (27) is at a vertex of the feasible region where \( n - m \) components are zero. So one possible stopping criterion is to iterate until \( n - m \) components of \( x^k \) are sufficiently small. Then those components can be set to zero, and the remaining components can be obtained by solving the linear system \( Ax = b \). However, this approach fails when there is an entire facet of global minimizers on the boundary. This situation occurs for example in the initial phase where any feasible point for the original problem gives rise to the minimum possible value \( \gamma = 0 \). Following are some other possibilities. Most likely you will need to use a combination of them:

— In the initial phase, monitor the value of \( \gamma \) and require that it be sufficiently small.

— Monitor progress in the unscaled feasible region and require that it be sufficiently small.

— Monitor the value of the objective function and require that it be sufficiently small.

— Monitor the condition number of the weighted Least Squares Problem (35) and stop when it becomes too large.
— Implement the suggestion in Strang’s first paragraph in the section entitled Dual Variables.

Projecting into the Nullspace.

• This section provides an alternative derivation of the projection matrix

\[ P = I - A^T (AA^T)^{-1} A. \quad (42) \]

• Let’s use \( b = -c \), for simplicity, and ask which matrix \( P \) will project \( b \) into the null space of \( A \).

• Suppose

\[ Pb = z \quad (43) \]

• Then \( b - z \) must be perpendicular to every vector in the null space of \( A \).

• this means that \( b - z \) is in the row space of \( A \), i.e.,

\[ b - z = A^T y \quad (44) \]

for some \( y \in \mathbb{R}^m \). Hence

\[ z = b - A^T y \quad (45) \]
• We also know that $Az = 0$:

$$Az = Ab - AA^T y = 0.$$  \hfill (46)

• This implies that

$$AA^T y = Ab \implies y = (AA^T)^{-1} Ab.$$  \hfill (47)

• This implies that

$$z = b - \tilde{A}^T (AA^T)^{-1} Ab = Pb$$  \hfill (48)

where $P$ is the required projection matrix.