Optimization Methods

Draft of August 26, 2005

III.
Solving Linear Programs by Interior-Point Methods

Robert Fourer

Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, Illinois 60208-3119, U.S.A.

(847) 491-3151

4er@iems.northwestern.edu
http://www.iems.northwestern.edu/~4er/

10. Essential Features

Simplex methods get to the solution of a linear program by moving from vertex to vertex along edges of the feasible region. It seems reasonable that some better method might get to an optimum faster by instead moving through the *interior* of the region, directly toward the optimal point. This is not as easy as it sounds, however.

As in other respects the low-dimensional geometry of linear programs can be misleading. It is convenient to think of two-dimensional and three-dimensional feasible regions as being polyhedrons that are fairly round in shape, but these are the cases in which a long step through the middle is easy to see and makes great progress. When there are thousands or even millions of variables, it is quite another matter to "see" a path through the feasible polyhedron, which typically is highly elongated. One possibility, building on the simplex method, would be to increase from zero not one but all variables that have negative reduced costs. No practical way has been found, however, to compute steps based only on the reduced costs that tend to move through the center of the polyhedron toward the optimum rather than across to boundary points that are far from the optimum (and are not necessarily vertex points).

The key to an effective ***interior-point method*** is to borrow a few simple ideas from nonlinear optimization. In the context of linear programming, these ideas are sufficiently elementary that we can develop them independently. Applications to general nonlinear programming will be taken up in subsequent chapters.

10.1 Preliminaries

We show in this chapter how an effective interior-point method can be derived from a simple idea for solving the optimality conditions for linear programming. We consider in particular the *complementary slackness* conditions that were derived in Part III for primal and dual linear programs in the form

$$\begin{array}{lll@{\qquad}ll} \text{Minimize} & c^T x & & \text{Maximize} & b^T \pi \\ \text{Subject to} & Ax = b & & \text{Subject to} & A^T\pi \le c \\ & x \ge 0 & & & \end{array}$$

Complementary slackness says that x^* and π^* are optimal provided that they satisfy

▷ Primal feasibility: $Ax^* = b$, $x^* \ge 0$

▷ Dual feasibility: $A^T\pi^* \le c$

▷ Complementarity:
Either $x_j^* = 0$ or $a_j^T \pi_j^* = c_j$ (or both), for each $j = 1, \ldots, n$

To make these conditions easier to work with, we begin by writing them as equations in nonnegative variables. We treat all vectors as column vectors.

We start by introducing a vector of *slack variables*, σ, so that $A^T\pi^* \le c$ may be expressed equivalently by $A^T\pi^* + \sigma^* = c$ and $\sigma^* \ge 0$. In these terms,

$A^T\pi^* = c$ if and only if $\sigma^* = 0$, so the complementary slackness conditions become $x_j^* = 0$ or $\sigma_j^* = 0$ (or both). But saying that $x_j^* = 0$ or $\sigma_j^* = 0$ is equivalent to saying that $x_j^*\sigma_j^* = 0$. Thus we have the following equivalent statement of the complementary slackness conditions: x^* and π^* are optimal provided that they satisfy

- ▷ Primal feasibility: $Ax^* = b$, $x^* \geq 0$
- ▷ Dual feasibility: $A^T\pi^* + \sigma^* = c$, $\sigma^* \geq 0$
- ▷ Complementarity: $x_j^*\sigma_j^* = 0$ for every $j = 1, \ldots, n$

These conditions comprise a "square" system of $m+2n$ equations in the $m+2n$ variables (x, π, σ), plus nonnegativity of x and σ.

It remains to collect the equations $x_j\sigma_j = 0$ into a matrix equation. For this purpose, we define *diagonal* matrices X and Σ whose only nonzero elements are $X_{jj} = x_j$ and $\Sigma_{jj} = \sigma_j$, respectively. For example, for $n = 4$,

$$X = \begin{bmatrix} x_1 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 \\ 0 & 0 & x_3 & 0 \\ 0 & 0 & 0 & x_4 \end{bmatrix} \quad \text{and} \quad \Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \end{bmatrix}.$$

The diagonal elements of $X\Sigma$ are $x_j\sigma_j$, exactly the expressions that must be zero by complementary slackness, so we could express complementary slackness as

$$X\Sigma = \begin{bmatrix} x_1\sigma_1 & 0 & 0 & 0 \\ 0 & x_2\sigma_2 & 0 & 0 \\ 0 & 0 & x_3\sigma_3 & 0 \\ 0 & 0 & 0 & x_4\sigma_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

But this is n^2 equations, of which all but n are $0 = 0$. Instead we collapse the equations to the n significant ones by writing

$$X\Sigma e = \begin{bmatrix} x_1\sigma_1 & 0 & 0 & 0 \\ 0 & x_2\sigma_2 & 0 & 0 \\ 0 & 0 & x_3\sigma_3 & 0 \\ 0 & 0 & 0 & x_4\sigma_4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1\sigma_1 \\ x_2\sigma_2 \\ x_3\sigma_3 \\ x_4\sigma_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

where e is a vector whose elements are all 1. We will write this as $X\Sigma e = 0$, with the 0 understood as in previous chapters to refer to a vector of zeroes.

We have now shown that solving the primal optimization problem for x and the dual optimization problem for π is equivalent to solving the following combined system of equations and nonnegativity restrictions:

$$\begin{aligned} &Ax = b \\ &A^T\pi + \sigma = c \\ &X\Sigma e = 0 \\ &x \geq 0,\ \sigma \geq 0 \end{aligned}$$

We can regard the *interior points* $(\bar{x}, \bar{\pi}, \bar{\sigma})$ of this system to be those that satisfy the inequalities strictly: $\bar{x} > 0$, $\bar{\sigma} > 0$. Our goal is to show how interior-point methods can generate a series of such points that tend toward a solution of the linear program.

Diagonal matrices will prove to be convenient throughout the development of interior-point methods. If F and G are matrices having the vectors $f = (f_1, \ldots, f_n)$ and $g = (g_1, \ldots, g_n)$ on their diagonals and zeroes elsewhere, then

- ▷ $F^T = F$
- ▷ $FG = GF$ is a diagonal matrix having nonzero elements $f_j g_j$.
- ▷ F^{-1} is a diagonal matrix having nonzero elements f_j^{-1}, or $1/f_j$.
- ▷ $F^{-1}G = GF^{-1}$ is a diagonal matrix having nonzero elements g_j/f_j.
- ▷ $Fe = f$, and $F^{-1}f = e$, where e is a vector of all 1's.

We write $F = \mathrm{diag}(f)$ to say that F is the diagonal matrix constructed from f. As in these examples, we will normally use lower-case letters for vectors and the corresponding upper-case letters for the corresponding diagonal matrices.

10.2 A simple interior-point method

Much as we did in the derivation of the simplex method, we'll start off by assuming that we already know primal-feasible and dual-feasible interior-point solutions $\bar{x}$ and $(\bar{\pi}, \bar{\sigma})$:

- ◇ $A\bar{x} = b$, $\bar{x} > 0$
- ◇ $A^T\bar{\pi} + \bar{\sigma} = c$, $\bar{\sigma} > 0$

Our goal is to find solutions x^* and (π^*, σ^*) such that $Ax^* = b$ and $A^T\pi^* + \sigma^* = c$, but with x^* and σ^* being not interior but instead complementary: $x_j^* \geq 0$, $\sigma_j^* \geq 0$, and at least one of the two is $= 0$, for each $j = 1, \ldots, n$. We'll later show (in Section 11.2) that the interior-point approach is easily extended to start from any point $(\bar{x}, \bar{\pi}, \bar{\sigma})$ that has $\bar{x} > 0$ and $\bar{\sigma} > 0$, regardless of whether the equations are initially satisfied.

We can now give an elementary explanation of the method. Starting from a feasible, interior-point solution $(\bar{x}, \bar{\pi}, \bar{\sigma})$, we wish to find a *step* $(\Delta x, \Delta\pi, \Delta\sigma)$ such that $(\bar{x} + \Delta x, \bar{\pi} + \Delta\pi, \bar{\sigma} + \Delta\sigma)$ is a better such solution, in the sense that it comes closer to satisfying the complementarity conditions.

To find the desired step, we substitute $(\bar{x}+\Delta x, \bar{\pi}+\Delta\pi, \bar{\sigma}+\Delta\sigma)$ into the equations for feasibility and complementarity of the solution. Writing $\bar{X} = \mathrm{diag}(x)$, $\bar{\Sigma} = \mathrm{diag}(\sigma)$ and $\Delta X = \mathrm{diag}(\Delta x)$, $\Delta\Sigma = \mathrm{diag}(\Delta\sigma)$ in line with our previous notation, we have

$$\begin{aligned}
&A(\bar{x} + \Delta x) = b \\
&A^T(\bar{\pi} + \Delta\pi) + (\bar{\sigma} + \Delta\sigma) = c \\
&(\bar{X} + \Delta X)(\bar{\Sigma} + \Delta\Sigma)e = 0
\end{aligned}$$

Since we're given $A\bar{x} = b$ and $A^T\bar{\pi} + \bar{\sigma} = c$, these equations simplify to

$$\begin{aligned}&A\Delta x = 0\\&A^T\Delta\pi + \Delta\sigma = 0\\&\bar{X}\Delta\sigma + \bar{\Sigma}\Delta x = -\bar{X}\bar{\Sigma}e - \Delta X\Delta\Sigma e\end{aligned}$$

We would like to solve these $m + 2n$ equations for the steps — the $m + 2n$ Δ-values — but although all the terms on the left are linear in the steps, the term $\Delta X\Delta\Sigma e$ on the right is nonlinear. So long as each Δx_j is small relative to $\bar{x}_j$ and each $\Delta\sigma_j$ is small relative to $\bar{\sigma}_j$, however, we can expect each $\Delta x_j\Delta\sigma_j$ to be especially small relative to $\bar{x}_j\bar{\sigma}_j$. This suggests that we can get a reasonable approximate solution for the steps by solving the linear equations that are produced by dropping the $\Delta X\Delta\Sigma e$ term from the above equations. (The same approach will return in a later chapter, in a more general setting, as *Newton's method* for solving nonlinear equations.)

With the $\Delta X\Delta\Sigma e$ term dropped, we can solve the third equation for $\Delta\sigma$,

$$\Delta\sigma = \bar{X}^{-1}(-\bar{X}\bar{\Sigma}e - \bar{\Sigma}\Delta x) = -\bar{\sigma} - \bar{X}^{-1}\bar{\Sigma}\Delta x,$$

and substitute for $\Delta\sigma$ in the second equation to get

$$\begin{aligned}&A\Delta x = 0\\&A^T\Delta\pi - \bar{X}^{-1}\bar{\Sigma}\Delta x = \bar{\sigma}\end{aligned}$$

The matrix $\bar{X}^{-1}$ requires no special work to compute; because $\bar{X}$ has nonzero entries only along its diagonal, so does $\bar{X}^{-1}$, with the entries being $1/x_j$.

Rearranging our equations in matrix terms, we have an $(m + n) \times (m + n)$ equation system in the unknowns Δx and $\Delta\pi$:

$$\begin{bmatrix} -\bar{X}^{-1}\bar{\Sigma} & A^T \\ A & 0 \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta\pi \end{bmatrix} = \begin{bmatrix} \bar{\sigma} \\ 0 \end{bmatrix}.$$

Because $\bar{X}^{-1}\bar{\Sigma}$ is another diagonal matrix — it has entries $\bar{\sigma}_j/\bar{x}_j$ along its diagonal — we can optionally solve for Δx,

$$\Delta x = -\bar{X}\bar{\Sigma}^{-1}(\bar{\sigma} - A^T\Delta\pi) = -\bar{x} + (\bar{X}\bar{\Sigma}^{-1})A^T\Delta\pi,$$

and substitute for Δx in $A\Delta x = 0$ to arrive at an $m \times m$ equation system,

$$A(\bar{X}\bar{\Sigma}^{-1})A^T\Delta\pi = A\bar{x} = b.$$

The special forms of these equation systems guarantee that they have solutions and allow them to be solved efficiently.

At this point we could hope to use $(\bar{x} + \Delta x, \bar{\pi} + \Delta\pi, \bar{\sigma} + \Delta\sigma)$ as our new, improved solution. Some of the elements of $\bar{x} + \Delta x$ and $\bar{\sigma} + \Delta\sigma$ might turn out to be negative, however, whereas they must be positive to keep our new solution within the interior of the feasible region. To prevent this, we instead take our new solution to be

$$(\bar{x} + \theta(\Delta x), \bar{\pi} + \theta(\Delta\pi), \bar{\sigma} + \theta(\Delta\sigma))$$

where θ is a positive fraction ≤ 1. Because the elements of the vectors $\bar{x}$ and $\bar{\sigma}$ are themselves > 0, we know that $\bar{x} + \theta(\Delta x)$ and $\bar{\sigma} + \theta(\Delta\sigma)$ will also be strictly positive so long as θ is chosen small enough. (The vectors Δx and $\Delta\sigma$ serve here as what are known as *step directions* rather than whole steps, and θ is the *step length*.)

The derivation of a formula for θ is much like the derivation of the "minimum ratio" criterion for the simplex method (Part II). For each $\Delta x_j \geq 0$, the value of $\bar{x} + \theta(\Delta x)$ can only increase along with θ; but for $\Delta x_j < 0$, the value decreases. Thus we require

$$\bar{x}_j + \theta(\Delta x_j) > 0 \implies \theta < \frac{\bar{x}_j}{-\Delta x_j} \quad \text{for each } j \text{ such that } \Delta x_j < 0.$$

The same reasoning applied to $\bar{\sigma} + \theta(\Delta\sigma)$ gives

$$\bar{\sigma}_j + \theta(\Delta\sigma_j) > 0 \implies \theta < \frac{\bar{\sigma}_j}{-\Delta\sigma_j} \quad \text{for each } j \text{ such that } \Delta\sigma_j < 0.$$

For θ to be less than all of these values, it must be less than their minimum, and so we require

$$\theta \leq 1, \quad \theta < \theta_x = \min_{j:\Delta x_j < 0} \frac{\bar{x}_j}{-\Delta x_j}, \quad \theta < \theta_\sigma = \min_{j:\Delta\sigma_j < 0} \frac{\bar{\sigma}_j}{-\Delta\sigma_j}.$$

To implement this criterion we pick some fraction $\alpha < 1$ and take

$$\theta = \min(1, \alpha\theta_x, \alpha\theta_\sigma).$$

It suffices to stick with one α value for all steps of the algorithm. We would like to use a value of α close to 1, but far enough from 1 to keep the solution comfortably within the interior; we'll show some examples for the choice of this value later.

We have now laid out the basic algorithm — often called the ***affine scaling*** interior-point method for linear programming — whose details are collected in Figure 10–1. Given a feasible interior solution, we compute the next solution by solving the given equations for the step direction, determining a step length, and forming a new solution accordingly.

But how do we know when to stop? Because $A\bar{x} = b$ and $A\Delta x = 0$, every new solution remains primal-feasible:

$$A(\bar{x} + \theta(\Delta x)) = A\bar{x} + \theta(A\Delta x) = b.$$

Similarly, because $A^T\bar{\pi} + \bar{\sigma} = c$ and $A^T\Delta\pi + \Delta\sigma = 0$, every new solution remains dual-feasible:

$$A^T(\bar{\pi} + \theta(\Delta\pi)) + (\bar{\sigma} + \theta(\Delta\sigma)) = A^T\bar{\pi} + \bar{\sigma} + \theta(A^T\Delta\pi + \Delta\sigma) = c.$$

All of the elements of the vectors x and σ remain strictly positive, however, whereas the complementary slackness conditions require that either $x_j = 0$ or $\sigma_j = 0$ for each $j = 1, \ldots, n$. Hence, in a sense, this algorithm can never generate an optimal solution!

```
Given $\bar{x}$ such that $A\bar{x} = b$, $\bar{x} > 0$.
Given $\bar{\pi}$, $\bar{\sigma}$ such that $A^T\bar{\pi} + \bar{\sigma} = c$, $\bar{\sigma} > 0$.

Choose a step fraction $0 < \alpha < 1$.
Choose a complementarity tolerance $\epsilon > 0$.

Repeat
    Solve $\begin{bmatrix} -\bar{X}^{-1}\bar{\Sigma} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\pi \end{bmatrix} = \begin{bmatrix} \bar{\sigma} \\ 0 \end{bmatrix}$ and set $\Delta\sigma = -\bar{\sigma} - \bar{X}^{-1}\bar{\Sigma}\Delta x$.
    Let $\theta_x = \min_{j:\Delta x_j<0} \frac{\bar{x}_j}{-\Delta x_j}$, $\theta_\sigma = \min_{j:\Delta\sigma_j<0} \frac{\bar{\sigma}_j}{-\Delta\sigma_j}$.
    Let $\theta = \min(1, \alpha\theta_x, \alpha\theta_\sigma)$.
    Update $\bar{x} = \bar{x} + \theta(\Delta x)$, $\bar{\pi} = \bar{\pi} + \theta(\Delta\pi)$, $\bar{\sigma} = \bar{\sigma} + \theta(\Delta\sigma)$.
until $\bar{x}_j\bar{\sigma}_j < \epsilon$ for all $j = 1, \ldots, n$.
```

Figure 10–1. *An "affine scaling" primal-dual interior-point algorithm.*

What we could aim to show, however, is that the complementarity terms $x_j\sigma_j$ get progressively closer to 0 as the algorithm proceeds, in such a way that they can be made as close to zero as we like provided that enough steps of the algorithm are run. In this case the values $x_j\sigma_j$ are said to *converge* to zero. We could then also hope to prove that the iterates $(\bar{x}, \bar{\pi}, \bar{\sigma})$ converge to a solution that achieves an optimal value for the objective.

For our algorithm we thus pick a small positive value ϵ, and stop iterating when all of the $\bar{\sigma}_j\bar{x}_j$ values are less than this ϵ. We do not take ϵ to be zero, because we know that the algorithm cannot reduce all $\bar{\sigma}_j\bar{x}_j$ to zero after any finite number of steps. Because computers can only work with numbers to some finite number of places, moreover, the computations are only approximate, and tend to become unreliable as the iterates approach the optimal boundary point. Tolerances of this sort are an important aspect of any convergent algorithm.

10.3 An example

We illustrate the geometry of the interior-point method on the following two-variable example, for which the feasible region is graphed in Figure 10–2:

$$
\begin{array}{ll}
\text{Minimize} & 2x_1 + 1.5x_2 \\
\text{Subject to} & 12x_1 + 24x_2 \ge 120 \\
 & 16x_1 + 16x_2 \ge 120 \\
 & 30x_1 + 12x_2 \ge 120 \\
 & x_1 \le 15 \\
 & x_2 \le 15 \\
 & x_1 \ge 0,\ \ x_2 \ge 0
\end{array}
$$

Our starting point $\bar{x}$ must lie in the interior of the feasible region shown in the figure. The iterates will progress through the interior toward the optimal vertex, $(1\,{}^{2}\!/_{3}, 5\,{}^{5}\!/_{6})$.

To illustrate the computations, we add appropriate slack variables to transform this problem to one of equalities in nonnegative variables:

$$
\begin{array}{lrrrrrrrl}
\text{Minimize} & 2x_1 & + 1.5x_2 & & & & & & \\
\text{Subject to} & 12x_1 & + \ 24x_2 & - x_3 & & & & & = 120 \\
 & 16x_1 & + \ 16x_2 & & - x_4 & & & & = 120 \\
 & 30x_1 & + \ 12x_2 & & & - x_5 & & & = 120 \\
 & x_1 & & & & & + x_6 & & = 15 \\
 & & x_2 & & & & & + x_7 & = 15 \\
 & & & & & & & x_1, \ldots, x_7 & \geq 0
\end{array}
$$

From the figure it is clear that $(\bar{x}_1, \bar{x}_2) = (10, 10)$ is a point near the middle of the feasible set. Substituting into the equations, we can easily solve for the values of the slack variables $(\bar{x}_3, \ldots, \bar{x}_7)$, which are all positive at any interior point. Then we have as an initial primal iterate,

$$
\bar{x} = \begin{bmatrix} 10 \\ 10 \\ 240 \\ 200 \\ 300 \\ 5 \\ 5 \end{bmatrix} > 0.
$$

For an initial dual iterate, the algorithm requires a $\bar{\pi}$ such that $\bar{\sigma} = c - A^T\bar{\pi} > 0$. Writing these equations explicitly, we have

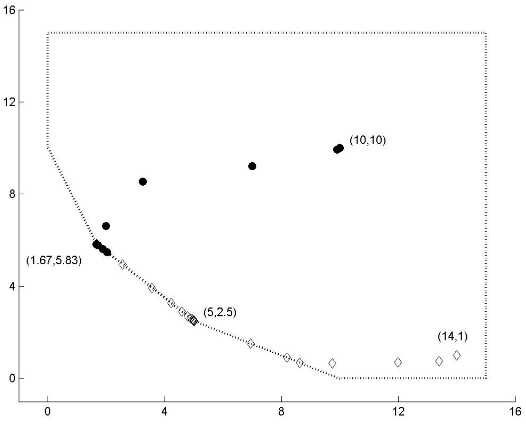

Figure 10–2. The feasible region for the example of Section 10.3, with two iteration paths of an affine-scaling interior-point method.

$$\begin{array}{rcl}
\bar\sigma_1 &=& 2 - 12\bar\pi_1 - 16\bar\pi_2 - 30\bar\pi_3 - 1\bar\pi_4 > 0\\
\bar\sigma_2 &=& 1.5 - 24\bar\pi_1 - 16\bar\pi_2 - 12\bar\pi_3 - 1\bar\pi_5 > 0\\
\bar\sigma_3 &=& 0 + 1\bar\pi_1 > 0\\
\bar\sigma_4 &=& 0 + 1\bar\pi_2 > 0\\
\bar\sigma_5 &=& 0 + 1\bar\pi_3 > 0\\
\bar\sigma_6 &=& 0 - 1\bar\pi_4 > 0\\
\bar\sigma_7 &=& 0 - 1\bar\pi_5 > 0
\end{array}$$

These can be satisfied by picking positive values for the first three π-variables, and then setting the remaining two sufficiently negative; $\bar\pi_1 = \bar\pi_2 = \bar\pi_3 = 1$ and $\bar\pi_4 = \bar\pi_5 = -60$ will do, for example. We then have

$$\bar\pi = [\ 1 \quad 1 \quad 1 \quad -60 \quad -60,\ ],$$

$$\bar\sigma = [\ 4 \quad 9.5 \quad 1 \quad 1 \quad 1 \quad 60 \quad 60\ ] > 0.$$

In our matrix terminology, we also have

$$c = [\ 2 \quad 1.5 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\ ],$$

$$A = \begin{bmatrix} 12 & 24 & -1 & 0 & 0 & 0 & 0\\ 16 & 16 & 0 & -1 & 0 & 0 & 0\\ 30 & 12 & 0 & 0 & -1 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 120\\ 120\\ 120\\ 15\\ 15 \end{bmatrix}.$$

We choose the step fraction $\alpha = 0.995$, which — for our examples — gives reliable results that cannot be improved upon by settings closer to 1. Finally, we specify a complementarity tolerance $\epsilon = 0.00001$, so that the algorithm will stop when all $\bar x_j \bar\sigma_j < .00001$.

To begin an iteration, the algorithm must form the linear equation system

$$\begin{bmatrix} -\bar X^{-1}\bar\Sigma & A^T\\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x\\ \Delta\pi \end{bmatrix} = \begin{bmatrix} \bar\sigma\\ 0 \end{bmatrix}.$$

In the matrix's upper left-hand corner is a diagonal block whose entries are $-\bar\sigma_j/\bar x_j$. Below this block is a copy of the matrix A, and to its right is a copy of A^T, with the lower right-hand block filled out by zeros. The right-hand side is a copy of $\bar\sigma$ filled out by zeros. So the entire system to be solved at the first iteration in our example is

$$\left[\begin{array}{ccccccc|ccccc}
-.4000 & 0 & 0 & 0 & 0 & 0 & 0 & 12 & 16 & 30 & 1 & 0\\
0 & -.9500 & 0 & 0 & 0 & 0 & 0 & 24 & 16 & 12 & 0 & 1\\
0 & 0 & -.0042 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -.0050 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -.0033 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & -12 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -12 & 0 & 0 & 0 & 0 & 1\\
\hline
12 & 24 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
16 & 16 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
30 & 12 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\left[\begin{array}{c}
\Delta x_1\\ \Delta x_2\\ \Delta x_3\\ \Delta x_4\\ \Delta x_5\\ \Delta x_6\\ \Delta x_7\\ \hline \Delta\pi_1\\ \Delta\pi_2\\ \Delta\pi_3\\ \Delta\pi_4\\ \Delta\pi_5
\end{array}\right]
=
\left[\begin{array}{c}
4\\ 9.5\\ 1\\ 1\\ 1\\ 60\\ 60\\ \hline 0\\ 0\\ 0\\ 0\\ 0
\end{array}\right].$$

(The vertical and horizontal lines are shown only to emphasize the matrix's block structure; they have no mathematical significance.)

The study of solving an equation system of this kind is a whole topic in itself. For now, we simply report that the solution is

$$\Delta x = \begin{bmatrix} -0.1017 \\ -0.0658 \\ -2.7997 \\ -2.6803 \\ -3.8414 \\ 0.1017 \\ 0.0658 \end{bmatrix}, \quad \Delta\pi = \begin{bmatrix} -0.9883 \\ -0.9866 \\ -0.9872 \\ 61.2208 \\ 60.7895 \end{bmatrix}.$$

We can then set

$$\begin{aligned}
\Delta\sigma &= -\bar{\sigma} - \bar{X}^{-1}\bar{\Sigma}\Delta x \\
&= -\begin{bmatrix} 4 \\ 9.5 \\ 1 \\ 1 \\ 1 \\ 60 \\ 60 \end{bmatrix} - \begin{bmatrix} .4000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .9500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .0042 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .0050 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .0033 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 \end{bmatrix} \begin{bmatrix} -0.1017 \\ -0.0658 \\ -2.7997 \\ -2.6803 \\ -3.8414 \\ 0.1017 \\ 0.0658 \end{bmatrix} \\
&= \begin{bmatrix} -3.9593 \\ -9.4375 \\ -0.9883 \\ -0.9866 \\ -0.9872 \\ -61.2208 \\ -60.7895 \end{bmatrix}.
\end{aligned}$$

The entire step vector $(\Delta x, \Delta\pi, \Delta\sigma)$ has now been computed.

The remainder of the iteration determines the length of the step. The ratio $\bar{x}_j/(-\Delta x_j)$ is computed for each of the five $\Delta x_j < 0$, and θ_x is set to the smallest:

$$\begin{aligned}
\Delta x_1 < 0:&\quad \bar{x}_1/(-\Delta x_1) = 10/0.1017 = 98.3284 \\
\Delta x_2 < 0:&\quad \bar{x}_2/(-\Delta x_2) = 10/0.0658 = 152.0034 \\
\Delta x_3 < 0:&\quad \bar{x}_3/(-\Delta x_3) = 240/2.7997 = 85.7242 \\
\Delta x_4 < 0:&\quad \bar{x}_4/(-\Delta x_4) = 200/2.6803 = 74.6187 = \theta_x \\
\Delta x_5 < 0:&\quad \bar{x}_5/(-\Delta x_5) = 300/3.8414 = 78.0972
\end{aligned}$$

The ratios $\bar{\sigma}_j/(-\Delta\sigma_j)$ are computed in the same way to determine θ_σ:

$$\begin{aligned}
\Delta\sigma_1 < 0:&\quad \bar{\sigma}_1/(-\Delta\sigma_1) = 4/3.9593 = 1.0103 \\
\Delta\sigma_2 < 0:&\quad \bar{\sigma}_2/(-\Delta\sigma_2) = 9.5/9.4375 = 1.0066 \\
\Delta\sigma_3 < 0:&\quad \bar{\sigma}_3/(-\Delta\sigma_3) = 1/0.9883 = 1.0118 \\
\Delta\sigma_4 < 0:&\quad \bar{\sigma}_4/(-\Delta\sigma_4) = 1/0.9866 = 1.0136 \\
\Delta\sigma_5 < 0:&\quad \bar{\sigma}_5/(-\Delta\sigma_5) = 1/0.9872 = 1.0130 \\
\Delta\sigma_6 < 0:&\quad \bar{\sigma}_6/(-\Delta\sigma_6) = 60/61.2208 = 0.9801 = \theta_\sigma \\
\Delta\sigma_7 < 0:&\quad \bar{\sigma}_7/(-\Delta\sigma_7) = 60/60.7895 = 0.9870
\end{aligned}$$

The step length is thus given by

$$\begin{aligned}\theta &= \min(1, \alpha\theta_x, \alpha\theta_\sigma)\\ &= \min(1, .995 \cdot 74.6187, .995 \cdot 0.9801) = 0.975159.\end{aligned}$$

Thus the iteration concludes with the computation of the next iterate as

$$\bar{x} = \bar{x} + \theta(\Delta x) = \begin{bmatrix} 10\\ 10\\ 240\\ 200\\ 300\\ 5\\ 5 \end{bmatrix} + 0.975159 \begin{bmatrix} -0.1017\\ -0.0658\\ -2.7997\\ -2.6803\\ -3.8414\\ 0.1017\\ 0.0658 \end{bmatrix} = \begin{bmatrix} 9.9008\\ 9.9358\\ 237.2699\\ 197.3863\\ 296.2541\\ 5.0992\\ 5.0642 \end{bmatrix},$$

and similarly for $\bar{\pi}$ and $\bar{\sigma}$.

Although there are 7 variables in the form of the problem that the algorithm works on, we can show the algorithm's progress in Figure 10–2 by plotting the points defined by the x_1 and x_2 components of the iterates. In the first iteration, we have moved from $(10, 10)$ to $(9.9008, 9.9358)$.

If we continue in the same way, then the algorithm carries out a total of 9 iterations before reaching a solution that satisfies the stopping conditions:

iter	x_1	x_2	θ	$\max \bar{x}_j\bar{\sigma}_j$
0	10.0000	10.0000	– – –	300.000000
1	9.9008	9.9358	0.975159	11.058316
2	6.9891	9.2249	0.423990	6.728827
3	3.2420	8.5423	0.527256	2.878729
4	1.9835	6.6197	0.697264	1.156341
5	2.0266	5.4789	0.693037	0.486016
6	1.8769	5.6231	0.581321	0.189301
7	1.7204	5.7796	0.841193	0.027134
8	1.6683	5.8317	0.979129	0.000836
9	1.6667	5.8333	0.994501	0.000004

The step length drops at iteration 2 but then climbs toward the ideal step length of 1. The $\max \bar{x}_j\bar{\sigma}_j$ term steadily falls, until at the end it has a value that is not significantly different from zero.

Consider now what happens when we try a different starting point, not so well centered:

$$\bar{x} = \begin{bmatrix} 14\\ 1\\ 72\\ 120\\ 312\\ 1\\ 14 \end{bmatrix}, \quad \bar{\pi} = \begin{bmatrix} 0.01\\ 0.01\\ 0.01\\ -1.00\\ -1.00 \end{bmatrix}.$$

The first 10 iterations proceed as follows:

iter	x_1	x_2	θ	$\max \bar{x}_j \bar{\sigma}_j$
0	14.0000	1.0000	− − −	33.880000
1	13.3920	0.7515	0.386962	21.275227
2	11.9960	0.7102	0.126892	18.625818
3	9.7350	0.6538	0.242316	14.301476
4	8.6222	0.6915	0.213267	11.413212
5	8.1786	0.9107	0.197309	9.246931
6	6.9463	1.5269	0.318442	6.536069
7	5.0097	2.4951	0.331122	4.467188
8	5.0000	2.5000	0.117760	3.942131
9	5.0000	2.5000	0.217604	3.084315
10	5.0000	2.5000	0.361839	1.968291

The iterates have become stuck near the boundary, in particular near the non-optimal vertex point (5,2.5). (The iterates indeed appear to reach the vertex point, but that is only because we have rounded the output at the 4th place.) Over the next 10 iterations the steps make little further progress, with θ falling to minuscule values:

iter	x_1	x_2	θ	$\max \bar{x}_j \bar{\sigma}_j$
11	5.0000	2.5000	0.002838	1.962706
12	4.9999	2.5001	0.000019	1.962668
13	4.9999	2.5001	0.000011	1.962647
14	4.9998	2.5002	0.000021	1.962606
15	4.9996	2.5004	0.000042	1.962525
16	4.9991	2.5009	0.000083	1.962362
17	4.9983	2.5017	0.000165	1.962037
18	4.9966	2.5034	0.000330	1.961390
19	4.9932	2.5068	0.000658	1.960100
20	4.9865	2.5135	0.001312	1.957527

Then finally the iterates start to move along (but slightly interior to) the edge defined by $16x_1 + 16x_2 \geq 120$, eventually approaching the vertex that is optimal:

iter	x_1	x_2	θ	$\max \bar{x}_j \bar{\sigma}_j$
21	4.9732	2.5268	0.002617	1.952404
22	4.9466	2.5534	0.005216	1.942219
23	4.8941	2.6059	0.010386	1.922042
24	4.7909	2.7091	0.020642	1.882350
25	4.5915	2.9085	0.040872	1.805354
26	4.2178	3.2822	0.080335	1.660168
27	3.5612	3.9388	0.155645	1.401765
28	2.5521	4.9479	0.293628	0.994245
29	1.6711	5.8289	0.418568	0.603264
30	1.6667	5.8333	0.227067	0.466640
31	1.6667	5.8333	0.398038	0.280902
32	1.6667	5.8333	0.635639	0.102350
33	1.6667	5.8333	0.864063	0.013913
34	1.6667	5.8333	0.977201	0.000317
35	1.6667	5.8333	0.994594	0.000002

Although x_1 and x_2 have reached their optimal values at the 30th iteration, the algorithm requires 5 more iterations to bring down the maximum $\bar{x}_j \bar{\sigma}_j$ and so to prove optimality. The θ step values also remain low until near the very end.

Figure 10–2 shows the paths taken in both of our examples. It's easy to see here that one starting point was much better centered that the other, but finding a well-centered starting point for a large linear program is in general a hard problem — as hard as finding an optimal point. Thus there is no reliable way to keep the affine scaling method from sometimes getting stuck near the boundary and taking a large number of iterations. This difficulty motivates a "centered" method that we consider next.

10.4 A centered interior-point method

To avoid the poor performance of the affine scaling approach, we need a way to keep the iterates away from the boundary of the feasible region, until they begin to approach the optimum. One very effective way to accomplish this is to keep the iterates near a well-centered path to the optimum.

Given the affine scaling method, the changes necessary to produce a such a centered method are easy to describe:

- ▷ Change the complementary slackness conditions $x_j\sigma_j = 0$ to $x_j\sigma_j = \mu$, where μ is a positive constant.
- ▷ Start with a large value of μ, and gradually reduce it toward 0 as the algorithm proceeds.

We explain first how these changes affect the computations — the differences are minor — and why they have the desired centering effect. We can then motivate a simple formula for choosing μ at each step.

The centering steps. The modified complementary slackness conditions represent only a minor change in the equations we seek to solve. In matrix form they are:

$$\begin{aligned}
&Ax = b\\
&A^T\pi + \sigma = c\\
&X\Sigma e = \mu e\\
&x \ge 0,\ \sigma \ge 0
\end{aligned}$$

(Since e is a vector of all ones, μe is a vector whose elements are all μ.) Because none of the terms involving variables have been changed, the equations for the step come out the same as before, except with the extra μe term on the right:

$$\begin{aligned}
&A\Delta x = 0\\
&A^T\Delta\pi + \Delta\sigma = 0\\
&\bar{X}\Delta\sigma + \bar{\Sigma}\Delta x = \mu e - \bar{X}\bar{\Sigma}e - \Delta X\Delta\Sigma e
\end{aligned}$$

Dropping the $\Delta X\Delta\Sigma$ term once more, and solving the third equation to substitute

$$\Delta\sigma = \bar{X}^{-1}(\mu e - \bar{X}\bar{\Sigma}e - \bar{\Sigma}\Delta x) = -\bar{\sigma} - \bar{X}^{-1}(\bar{\Sigma}\Delta x - \mu e),$$

into the second, we arrive at almost the same equation system, the only change being the replacement of $\bar{\sigma}$ by $\bar{\sigma} - \mu\bar{X}^{-1}e$ in the right-hand side:

$$\begin{bmatrix} -\bar{X}^{-1}\bar{\Sigma} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \pi \end{bmatrix} = \begin{bmatrix} \bar{\sigma} - \mu \bar{X}^{-1} e \\ 0 \end{bmatrix}.$$

(The elements of the vector $\mu \bar{X}^{-1} e$ are $\mu / \bar{x}_j$.) Once these equations are solved, a step is determined as in affine scaling, after which the centering parameter μ may be reduced as explained later in this section.

Intuitively, changing $x_j \sigma_j = 0$ to $x_j \sigma_j = \mu$ tends to produce a more centered solution because it encourages both x_j and σ_j to stay away from 0, and hence the boundary, as the algorithm proceeds. But there is a deeper reason. The modified complementarity conditions are the optimality conditions for a modified optimization problem. Specifically, x^* and (π^*, σ^*) satisfy the conditions:

$$\begin{aligned} &Ax^* = b,\ x^* \geq 0 \\ &A^T \pi^* + \sigma^* = c,\ \sigma^* \geq 0 \\ &x_j^* \sigma_j^* = \mu,\ \text{for each } j = 1, \ldots, n \end{aligned}$$

if and only if x^* is optimal for

$$\begin{array}{ll} \text{Minimize} & c^T x - \mu \sum_{j=1}^{n} \log x_j \\ \text{Subject to} & Ax = b \\ & x \geq 0 \end{array}$$

This is known as the *log barrier* problem for the linear program, because the log terms can be viewed as forcing the optimal values x_j^* away from zero. Indeed, as any x_j approaches zero, $-\mu \log x_j$ goes to infinity, thus ruling out any sufficiently small values of x_j from being part of the optimal solution. Reducing μ does allow the optimal values to get closer to zero, but so long as μ is positive the barrier effect remains. (The constraints $x \geq 0$ are needed only when $\mu = 0$.)

A centered interior-point algorithm is thus often called a ***barrier method*** for linear programming. Only interior points are generated, but since μ is decreased gradually toward 0, the iterates can converge to an optimal solution in

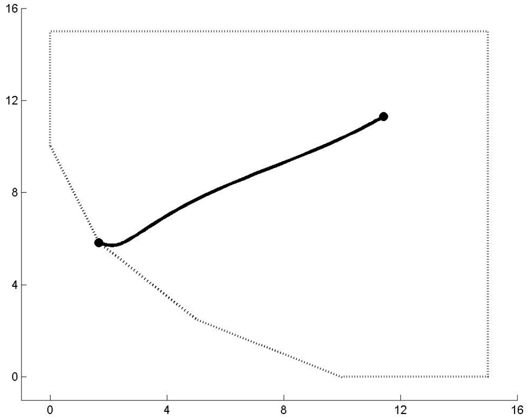

Figure 10–3. The central path for the feasible region shown in Figure 10–2.

which some of the x_j^* are zero. The proof is not as mathematically elementary as that for the simplex method, so we delay it to a future section, along with further modifications that are important to making barrier methods a practical alternative for solving linear programs.

Barrier methods also have an intuitive geometric interpretation. We have seen that as $\mu \to 0$, the optimal solution to the barrier problem approaches the optimal solution to the original linear program. On the other hand, as $\mu \to \infty$, the $c^T x$ part of the barrier problem's objective becomes insignificant and the optimal solution to the barrier problem approaches the minimizer of $\sum_{j=1}^n \log x_j$ subject to $Ax = b$. This latter point is known as the *analytic center;* in a sense it is the best-centered point of the feasible region.

If all of the solutions to the barrier problem for all values of μ are taken together, they form a curve — or *path* — from the analytic center of the feasible region to a non-interior optimal solution. Figure 10–3 depicts the central path for the feasible region plotted in Figure 10–2.

Choice of the centering parameter. If we were to run the interior-point method with a fixed value $\hat{\mu}$ of the barrier parameter, it would eventually converge to the point $(\hat{x}, \hat{\pi}, \hat{\sigma})$ on the central path that satisfies

$$\begin{aligned}&A\hat{x} = b,\ \hat{x} \ge 0\\ &A^T\hat{\pi} + \hat{\sigma} = c,\ \hat{\sigma} \ge 0\\ &\hat{x}_j\hat{\sigma}_j = \hat{\mu},\ \text{for each } j = 1, \ldots, n\end{aligned}$$

We do not want the method to find such a point for any positive $\hat{\mu}$, however, but to approximately *follow* the central path to a point that satisfies these equations for $\hat{\mu} = 0$. Thus rather than fixing μ, we would like in general to choose successively smaller μ values as the algorithm proceeds. On the other hand we do not want to choose too small a value too soon, as then the iterates may fail to become sufficiently centered and may exhibit the slow convergence typical of affine scaling.

These considerations suggest that we take a more adaptive approach. Given the current iterate $(\bar{x}, \bar{\pi}, \bar{\sigma})$, we first make a rough estimate of a value of μ corresponding to a nearby point on the central path. Then, to encourage the next iterate to lie further along the path, we set the barrier parameter at the next iteration to be some fraction of our estimate.

To motivate a formula for an estimate of μ, we observe that, if the current iterate were actually on the central path, then it would satisfy the modified complementarity conditions. For some μ, we would have

$$\bar{x}_1\bar{\sigma}_1 = \bar{x}_2\bar{\sigma}_2 = \ldots = \bar{x}_n\bar{\sigma}_n = \mu$$

The current iterate is not on the central path, so these terms are in general all different. As our estimate, however, we can reasonably take their average,

$$\mu = \frac{1}{n}\sum_{j=1}^{n} \bar{x}_j\bar{\sigma}_j = \frac{\bar{\sigma}\bar{x}}{n}.$$

Then we can take as our centering parameter at the next iteration some fixed fraction β of this estimate:

```
Given $\bar{x}$ such that $A\bar{x} = b$, $\bar{x} > 0$.
Given $\bar{\pi}$, $\bar{\sigma}$ such that $A^T\bar{\pi} + \bar{\sigma} = c$, $\bar{\sigma} > 0$.

Choose a step feasibility fraction $0 < \alpha < 1$.
Choose a step complementarity fraction $0 < \beta < 1$.
Choose a complementarity tolerance $\epsilon > 0$.

Repeat
    Let $\bar{\mu} = \beta\frac{\bar{\sigma}\bar{x}}{n}$.
    Solve $\begin{bmatrix} -\bar{X}^{-1}\bar{\Sigma} & A^T \\ A & 0 \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta\pi \end{bmatrix} = \begin{bmatrix} \bar{\sigma} - \bar{\mu}\bar{X}^{-1}e \\ 0 \end{bmatrix}$ and
        set $\Delta\sigma = -\bar{\sigma} - \bar{X}^{-1}(\bar{\Sigma}\Delta x - \bar{\mu}e)$.
    Let $\theta_x = \min_{j:\Delta x_j<0} \frac{\bar{x}_j}{-\Delta x_j}$, $\theta_\sigma = \min_{j:\Delta\sigma_j<0} \frac{\bar{\sigma}_j}{-\Delta\sigma_j}$.
    Let $\theta = \min(1, \alpha\theta_x, \alpha\theta_\sigma)$.
    Update $\bar{x} = \bar{x} + \theta(\Delta x)$, $\bar{\pi} = \bar{\pi} + \theta(\Delta\pi)$, $\bar{\sigma} = \bar{\sigma} + \theta(\Delta\sigma)$.
until $\Delta x_j\Delta\sigma_j < \epsilon$ for all $j = 1,\ldots,n$.
```

Figure 10–4. *A "barrier" primal-dual interior-point algorithm.*

$$\bar{\mu} = \beta\frac{\bar{\sigma}\bar{x}}{n}.$$

This is the last detail of the simple barrier method, which we summarize in Figure 10–4. (Among the more powerful extensions and refinements to this method is an approach that first computes an affine scaling "predictor" step, which provides estimates of $\bar{\mu}$ and of the previously dropped $\Delta X\Delta\Sigma e$ term for a subsequent barrier "corrector" step.)

10.5 Example, continued

Consider now how our first example from Section 10.3 would proceed using the barrier method. We take the step-length parameter α to be 0.99995; we can make it much closer to 1 than for the affine scaling method, because there is less danger of getting stuck near the boundary. For the parameter β that figures in the determination of the centering multiplier $\bar{\mu}$, a value of 0.1 works well. We set ϵ to 0.00001 as before.

At the start of the first iteration we have

$$\begin{aligned} \bar{\mu} &= \beta(\bar{\sigma}\bar{x})/n \\ &= 0.1(4\cdot 10 + 9.5\cdot 10 + 1\cdot 240 + 1\cdot 200 + 1\cdot 300 + 5\cdot 60 + 5\cdot 60)/7 \\ &= 21.0714 \end{aligned}$$

In the equations for the step, the term $\bar{\sigma}$ in the right-hand side is replaced by

$$\bar{\sigma} - \bar{\mu}\bar{X}^{-1}e = \begin{bmatrix} 4 \\ 9.5 \\ 1 \\ 1 \\ 1 \\ 60 \\ 60 \end{bmatrix} - 21.0714 \begin{bmatrix} 1/10 \\ 1/10 \\ 1/240 \\ 1/200 \\ 1/300 \\ 1/5 \\ 1/5 \end{bmatrix} = \begin{bmatrix} 1.8929 \\ 7.3929 \\ 0.9122 \\ 0.8946 \\ 0.9298 \\ 55.7857 \\ 55.7857 \end{bmatrix}.$$

The matrix and the rest of the right-hand side are as they were before, however, so we solve

$$\begin{bmatrix} -\bar{X}^{-1}\bar{\Sigma} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \pi \end{bmatrix} = \begin{bmatrix} \bar{\sigma} - \bar{\mu}\bar{X}^{-1}e \\ 0 \end{bmatrix} = \begin{bmatrix} 1.8929 \\ 7.3929 \\ 0.9122 \\ 0.8946 \\ 0.9298 \\ 55.7857 \\ 55.7857 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Once this system has been solved for Δx and $\Delta \pi$, the rest of the barrier iteration is the same as an affine scaling iteration. Thus we omit the details, and report only that the step length comes out to be

$$\begin{aligned} \theta &= \min(1, \alpha\theta_x, \alpha\theta_\sigma) \\ &= \min(1, .99995 \cdot 93.7075, .99995 \cdot 1.0695) = 1 \end{aligned}$$

and the next iterate is computed as

$$\bar{x} = \bar{x} + \theta(\Delta x) = \begin{bmatrix} 10 \\ 10 \\ 240 \\ 200 \\ 300 \\ 5 \\ 5 \end{bmatrix} + 1.0 \begin{bmatrix} 0.0314 \\ 0.0534 \\ 1.6576 \\ 1.3564 \\ 1.5829 \\ -0.0314 \\ -0.0534 \end{bmatrix} = \begin{bmatrix} 10.0314 \\ 10.0534 \\ 241.6576 \\ 201.3564 \\ 301.5829 \\ 4.9686 \\ 4.9466 \end{bmatrix}.$$

Whereas the first iteration of affine scaling on this example caused x_1 and x_2 to decrease, the first step of the barrier method increases them. Although the path to the solution is different, however, the number of iterations to optimality is still 9:

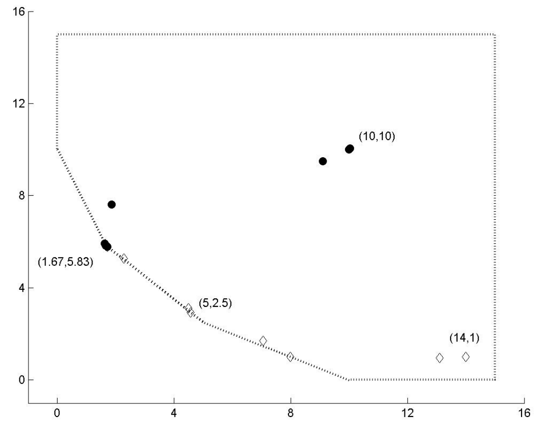

Figure 10–5. Iteration paths of the barrier interior-point method, starting from the same two points as in the Figure 10–2 plots for the affine scaling method.

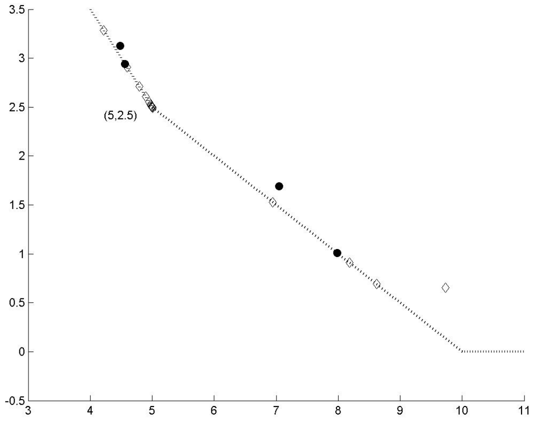

Figure 10–6. Detail from the plots for the (14,1) starting point shown in Figures 10–2 and 10–5. The affine scaling iterates (open diamonds) get stuck near the sub-optimal vertex (5,2.5), while the barrier iterates (filled circles) skip right past the vertex.

iter	x_1	x_2	θ	μ	$\max \bar{x}_j\bar{\sigma}_j$
0	10.0000	10.0000	– – –	– – –	300.000000
1	10.0314	10.0534	1.000000	21.071429	24.013853
2	9.1065	9.5002	0.896605	2.107143	5.536077
3	1.8559	7.6205	0.840474	0.406796	1.725153
4	1.6298	5.9255	0.780709	0.099085	0.654632
5	1.7156	5.7910	1.000000	0.029464	0.060567
6	1.6729	5.8294	1.000000	0.002946	0.003894
7	1.6671	5.8332	1.000000	0.000295	0.000304
8	1.6667	5.8333	1.000000	0.000029	0.000030
9	1.6667	5.8333	1.000000	0.000003	0.000003

As our discussion of the barrier method would predict, the centering parameter μ starts out large but quickly falls to near zero as the optimum is approached. Also, thanks to the centering performed at the first several steps, the step distance θ soon returns to its ideal value of 1.

When the barrier method is instead started from the poorly-centered point of Section 10.3, it requires more iterations, but only about a third as many as affine scaling:

iter	x_1	x_2	θ	μ	$\max \bar{x}_j\bar{\sigma}_j$
0	14.0000	1.0000	– – –	– – –	33.880000
1	13.0939	0.9539	0.465821	0.798571	19.325423
2	7.9843	1.0080	0.481952	0.463779	10.839497
3	7.0514	1.6930	0.443953	0.262612	6.544525
4	4.5601	2.9400	0.479414	0.157683	3.747461
5	4.4858	3.1247	0.539274	0.089647	1.806038
6	2.2883	5.2847	0.543138	0.046137	0.885009
7	1.6356	5.9110	0.495625	0.023584	0.507843
8	1.7235	5.7945	0.971205	0.013064	0.034860
9	1.6700	5.8312	1.000000	0.001645	0.002064
10	1.6669	5.8332	1.000000	0.000164	0.000167
11	1.6667	5.8333	1.000000	0.000016	0.000016
12	1.6667	5.8333	1.000000	0.000002	0.000002

Figure 10–5 plots the iteration paths taken by the barrier method from our two starting points. The path from the point that is not well-centered still tends to be near the boundary, but that is not so surprising — the first 8 iterations take a step that is .99995 times the step that would actually reach the boundary.

The important thing is that the barrier method avoids vertex points that are not optimal. This can be seen more clearly in Figure 10–6, which, for the starting point (14,1), plots the iterates of both methods in the neighborhood of the vertex (5,2.5). Affine scaling “jams” around this vertex, while the barrier iterations pass it right by.

11. Practical Refinements and Extensions

This chapter considers various improvements to interior-point methods that have proved to be very useful in practice:

- Taking separate primal and dual step lengths
- Starting from points that do not satisfy the constraint equations
- Handling simple bounds implicitly

Further improvements, not discussed here, include the computation of a predictor and a corrector step at each iteration to speed convergence, and the use of a "homogeneous" form of the linear program to provide more reliable detection of infeasible and unbounded problems.

11.1 Separate primal and dual step lengths

To keep things simple, we have defined a single step length for both the primal and the dual iterates. But to gain some flexibility, we can instead choose $\bar{x} + \theta(\Delta x)$ as the next primal iterate and $(\bar{\pi} + \phi(\Delta\pi), \bar{\sigma} + \phi(\Delta\sigma))$ as the next dual iterate. As before, since $A\bar{x} = b$ and $A\Delta x = 0$, every iterate remains primal-feasible:

$$A(\bar{x} + \theta(\Delta x)) = A\bar{x} + \theta(A\Delta x) = b.$$

Moreover, because $A^T\bar{\pi} + \bar{\sigma} = c$ and $A^T\Delta\pi + \Delta\sigma = 0$, every iterate remains dual-feasible:

$$A^T(\bar{\pi} + \phi(\Delta\pi)) + (\bar{\sigma} + \phi(\Delta\sigma)) = A^T\bar{\pi} + \bar{\sigma} + \phi(A^T\Delta\pi + \Delta\sigma) = c.$$

The different primal and dual steps retain these properties because the primal and dual variables are involved in separate constraints.

We now have for the primal variables, as before,

$$\bar{x}_j + \theta(\Delta x_j) > 0 \implies \theta < \frac{\bar{x}_j}{-\Delta x_j} \quad \text{for each } j \text{ such that } \Delta x_j < 0.$$

But now for the dual variables,

$$\bar{\sigma}_j + \phi(\Delta\sigma_j) > 0 \implies \phi < \frac{\bar{\sigma}_j}{-\Delta\sigma_j} \quad \text{for each } j \text{ such that } \Delta\sigma_j < 0.$$

It follows that

$$\theta \le 1, \quad \theta < \theta_x = \min_{j:\Delta x_j < 0} \frac{\bar{x}_j}{-\Delta x_j},$$

$$\phi \le 1, \quad \phi < \phi_\sigma = \min_{j:\Delta\sigma_j < 0} \frac{\bar{\sigma}_j}{-\Delta\sigma_j}.$$

Independent formulas for the step lengths are thus given by

$$\theta = \min(1, \alpha\theta_x), \quad \phi = \min(1, \alpha\phi_\sigma),$$

where $\alpha < 1$ is a chosen parameter as before.

11.2 Infeasible starting points

The assumption of an initial feasible solution has been convenient for our derivations, but is not essential. If $A\bar{x} = b$ does not hold, then the equations for the step are still $A(\bar{x} + \Delta x) = b$, but they only rearrange to

$$A\Delta x = b - A\bar{x}$$

rather than simplifying to $A\Delta x = 0$. Similarly if $A^T\bar{\pi} + \bar{\sigma} = c$ does not hold, then the associated equations for the step are still $A^T(\bar{\pi} + \Delta\pi) + (\bar{\sigma} + \Delta\sigma) = c$, but they only rearrange to

$$A^T\Delta\pi + \Delta\sigma = c - A^T\bar{\pi} - \bar{\sigma}$$

rather than simplifying to $A^T\Delta\pi + \Delta\sigma = 0$. As was the case in our derivation of the barrier method, however, this generalization only changes the constant terms of the step equations. Thus we can proceed as before to drop a $\Delta x\Delta\sigma$ term and eliminate $\Delta\sigma$ to give

$$\begin{bmatrix} -\bar{X}^{-1}\bar{\Sigma} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\pi \end{bmatrix} = \begin{bmatrix} (c - A^T\bar{\pi}) - \mu\bar{X}^{-1}e \\ (b - A\bar{x}) \end{bmatrix}.$$

Again the matrix is the same. The only difference is the addition of a few terms in the right-hand side. After the step equations have been solved, the computation of the step length and the determination of a new iterate can proceed as before.

Methods of this kind are called, naturally, *infeasible* interior-point methods. Their iterates may eventually achieve feasibility:

- ▷ If at any iteration the step length $\theta = 1$, then the new primal iterate $\bar{x} + \theta(\Delta x)$ is simply $\bar{x} + \Delta x$, which satisfies $A(\bar{x} + \Delta x) = A\bar{x} + A\Delta x = b + 0 = b$. Hence the next iterate is primal-feasible, after which the algorithm works like the previous, feasible one and all subsequent iterates are primal-feasible.

- ▷ If at any iteration the step length $\phi = 1$, then the new dual iterate $(\bar{\pi} + \theta(\Delta\pi), \bar{\sigma} + \theta(\Delta\sigma))$ is simply $(\bar{\pi} + \Delta\pi, \bar{\sigma} + \Delta\sigma)$, which satisfies $A^T(\bar{\pi} + \Delta\pi) + (\bar{\sigma} + \Delta\sigma) = (A^T\bar{\pi} + \bar{\sigma}) + (A^T\Delta\pi + \Delta\sigma) = c + 0 = c$. Hence the next iterate is dual-feasible, after which the algorithm works like the previous, feasible one and all subsequent iterates are dual-feasible.

After both a $\theta = 1$ and a $\phi = 1$ step have been taken, therefore, the algorithm behaves like a feasible interior-point method. It can be proved however that even if all primal step lengths $\theta < 1$ or all dual step lengths $\phi < 1$ then the iterates must still converge to a feasible point.

11.3 Bounded variables

The logic of our previous interior-point method derivations can be extended straightforwardly to the case where the variables are subject to bounds $x_j \le u_j$. We work with the following primal-dual pair, which incorporates additional dual variables corresponding to the bound constraints:

$$\begin{array}{llll}
\text{Minimize} & c^T x & \text{Maximize} & b^T \pi - \lambda u \\
\text{Subject to} & Ax = b & \text{Subject to} & \pi A - \lambda \le c \\
 & x \le u & & \lambda \ge 0 \\
 & x \ge 0 & &
\end{array}$$

There are now two sets of complementarity conditions:

- ▷ Either $x_j = 0$ or $\pi a_j - \lambda_j = c_j$ (or both), for each $j = 1, \ldots, n$
- ▷ Either $\lambda_j = 0$ or $x_j = u_j$ (or both), for each $j = 1, \ldots, n$

Writing σ_j for $c_j - (\pi a_j - \lambda_j)$ and s_j for $u_j - x_j$, these conditions are equivalent to $x_j \sigma_j = 0$ and $\lambda_j s_j = 0$.

Thus the primal feasibility, dual feasibility, complementarity, and nonnegativity conditions for the barrier problem are

$$\begin{array}{l}
Ax = b,\ x + s = u \\
A^T \pi - \lambda + \sigma = c \\
X\Sigma e = \mu e,\ \Lambda S e = \mu e \\
x \ge 0,\ \sigma \ge 0,\ s \ge 0,\ \lambda \ge 0
\end{array}$$

From this point the step equations is much as before. We substitute $\bar{x} + \Delta x$ for x, $\bar{s} + \Delta s$ for s, and so forth for the other variables; drop the terms $\Delta X \Delta \Sigma e$ and $\Delta \Lambda \Delta S e$; and use three of the five equations to eliminate the slack vectors $\Delta \lambda$, $\Delta \sigma$, and Δs. The remaining equations are

$$\begin{bmatrix} -\bar{X}^{-1}\bar{\Sigma} - \bar{S}^{-1}\bar{\Lambda} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \pi \end{bmatrix}$$

$$= \begin{bmatrix} (c + \bar{\lambda} - A^T \bar{\pi}) - \bar{S}^{-1}\bar{\Lambda}(u - \bar{x}) - \mu(\bar{X}^{-1} - \bar{S}^{-1})e \\ (b - A\bar{x}) \end{bmatrix}.$$

These equations may look a lot more complicated, but they're not much more work to assemble than the ones we previously derived. In the matrix, the only change is to replace $-\bar{X}^{-1}\bar{\Sigma}$ by $-\bar{X}^{-1}\bar{\Sigma} - \bar{S}^{-1}\bar{\Lambda}$, another diagonal matrix whose entries are $\bar{x}_j \bar{\sigma}_j + \bar{s}_j \bar{\lambda}_j$. The expression for the right-hand side vector, although it has a few more terms, still involves only vectors and diagonal matrices, and so remains inexpensive to compute.

The starting point for this extended method can be any $\bar{x} > 0$, $\bar{\sigma} > 0$, $\bar{s} > 0$, $\bar{\lambda} > 0$. As in the infeasible method described previously, the solution need not initially satisfy any of the equations. In particular, the solution may initially fail to satisfy $\bar{x} + \bar{s} = u$, and in fact $\bar{x}$ may start at a value greater then u. This is because the method handles $x \le u$ like any other constraint, adding slacks to make it just another equality in nonnegative variables.

As an alternative we can define an interior point to be one that is strictly within all its bounds, upper as well as lower. This means that the initial solution has to satisfy $0 < \bar{x} < u$ as well as $\bar{\sigma} > 0$, $\bar{\lambda} > 0$. Thus rather than treating s as an independent variable, we can define $\bar{s} = u - \bar{x} > 0$, in which case the right-hand-side term $\bar{S}^{-1}\bar{\Lambda}(u - \bar{x}) = \bar{S}^{-1}\bar{\Lambda}\bar{s} = \bar{\lambda}$ and the step equations can be simplified to look more like the ones for the non-bounded case:

$$\begin{bmatrix} -\bar{X}^{-1}\bar{\Sigma} - \bar{S}^{-1}\bar{\Lambda} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\pi \end{bmatrix} = \begin{bmatrix} (c - A^T\bar{\pi}) - \mu(\bar{X}^{-1} - \bar{S}^{-1})e \\ (b - A\bar{x}) \end{bmatrix}.$$

To proceed in this way it is necessary to pick the step length θ so that $0 < \bar{x} + \theta(\Delta x) < u$, however. Thus we require

$$\bar{x}_j + \theta(\Delta x_j) > 0 \implies \theta < \frac{\bar{x}_j}{-\Delta x_j} \quad \text{for each } j \text{ such that } \Delta x_j < 0,$$

and also

$$\bar{x}_j + \theta(\Delta x_j) < u \implies \theta < \frac{u - \bar{x}_j}{\Delta x_j} \quad \text{for each } j \text{ such that } \Delta x_j > 0.$$

To keep θ less than all these values, we must choose it less than their minimum. Adding the conditions necessary to keep $\bar{\sigma} + \theta(\Delta\sigma) > 0$ and $\bar{\lambda} + \theta(\Delta\lambda) > 0$, we arrive at

$$\theta \le 1, \quad \theta < \theta_x = \min_{j:\Delta x_j < 0} \frac{\bar{x}_j}{-\Delta x_j}, \quad \theta < \theta_s = \min_{j:\Delta x_j > 0} \frac{u - \bar{x}_j}{\Delta x_j},$$

$$\phi \le 1, \quad \phi < \phi_\sigma = \min_{j:\Delta\sigma_j < 0} \frac{\bar{\sigma}_j}{-\Delta\sigma_j}, \quad \phi < \phi_\lambda = \min_{j:\Delta\lambda_j < 0} \frac{\bar{\lambda}_j}{-\Delta\lambda_j}.$$

It then remains only to pick some fraction $\alpha < 1$ and take

$$\theta = \min(1, \alpha\theta_x, \alpha\theta_s), \quad \phi = \min(1, \alpha\phi_\sigma, \phi\theta_\lambda).$$

Despite the appeal of keeping $\bar{x}$ strictly within its bounds, however, this approach has been observed in practice to take longer to solve linear programs than the one that lets $\bar{x}$ start at any positive value. This is apparently because the latter method provides more flexibility in the choice of a well-centered starting point.