CHAPTER 14

Linear Programming: Interior-Point Methods

In the 1980s it was discovered that many large linear programs could be solved efficiently by using formulations and algorithms from nonlinear programming and nonlinear equations. One characteristic of these methods was that they required all iterates to satisfy the inequality constraints in the problem *strictly*, so they became known as interior-point methods. By the early 1990s, a subclass of interior-point methods known as *primal-dual methods* had distinguished themselves as the most efficient practical approaches, and proved to be strong competitors to the simplex method on large problems. These methods are the focus of this chapter.

Interior-point methods arose from the search for algorithms with better theoretical properties than the simplex method. As we mentioned in Chapter 13, the simplex method can be inefficient on certain pathological problems. Roughly speaking, the time required to solve a linear program may be exponential in the size of the problem, as measured by the number of unknowns and the amount of storage needed for the problem data. For almost all practical problems, the simplex method is much more efficient than this bound would suggest, but its poor worst-case complexity motivated the development of new algorithms with better guaranteed performance. The first such method was the *ellipsoid method*, proposed by Khachiyan [180], which finds a solution in time that is at worst polynomial in the problem size. Unfortunately, this method approaches its worst-case bound on *all* problems and is not competitive with the simplex method in practice.

Karmarkar's *projective algorithm* [175], announced in 1984, also has the polynomial complexity property, but it came with the added attraction of good practical behavior. The initial claims of excellent performance on large linear programs were never fully borne out, but the announcement prompted a great deal of research activity which gave rise to many new methods. All are related to Karmarkar's original algorithm, and to the log-barrier approach described in Chapter 19, but many of the approaches can be motivated and analyzed independently of the earlier methods.

Interior-point methods share common features that distinguish them from the simplex method. Each interior-point iteration is expensive to compute and can make significant progress towards the solution, while the simplex method usually requires a larger number of inexpensive iterations. Geometrically speaking, the simplex method works its way around the boundary of the feasible polytope, testing a sequence of vertices in turn until it finds the optimal one. Interior-point methods approach the boundary of the feasible set only in the limit. They may approach the solution either from the interior or the exterior of the feasible region, but they never actually lie on the boundary of this region.

In this chapter, we outline some of the basic ideas behind primal-dual interior-point methods, including the relationship to Newton's method and homotopy methods and the concept of the central path. We sketch the important methods in this class, and give a comprehensive convergence analysis of a particular interior-point method known as a *long-step path-following* method. We describe in some detail a practical predictor-corrector algorithm proposed by Mehrotra, which is the basis of much of the current generation of software.

14.1 PRIMAL-DUAL METHODS

OUTLINE

We consider the linear programming problem in standard form; that is,

$$\min\ c^T x, \quad \text{subject to } Ax = b,\ x \geq 0, \tag{14.1}$$

where c and x are vectors in $\mathbb{R}^n$, b is a vector in $\mathbb{R}^m$, and A is an $m \times n$ matrix with full row

rank. (As in Chapter 13, we can preprocess the problem to remove dependent rows from A if necessary.) The dual problem for (14.1) is

$$\max b^T\lambda, \quad \text{subject to } A^T\lambda + s = c, s \geq 0, \tag{14.2}$$

where λ is a vector in $\mathbb{R}^m$ and s is a vector in $\mathbb{R}^n$. As shown in Chapter 13, solutions of (14.1),(14.2) are characterized by the Karush–Kuhn–Tucker conditions (13.4), which we restate here as follows:

$$A^T\lambda + s = c, \tag{14.3a}$$
$$Ax = b, \tag{14.3b}$$
$$x_i s_i = 0, \quad i = 1, 2, \ldots, n, \tag{14.3c}$$
$$(x, s) \geq 0. \tag{14.3d}$$

Primal-dual methods find solutions (x^*, λ^*, s^*) of this system by applying variants of Newton's method to the three equalities in (14.3) and modifying the search directions and step lengths so that the inequalities $(x, s) \geq 0$ are satisfied *strictly* at every iteration. The equations (14.3a), (14.3b), (14.3c) are linear or only mildly nonlinear and so are not difficult to solve by themselves. However, the problem becomes much more difficult when we add the nonnegativity requirement (14.3d), which gives rise to all the complications in the design and analysis of interior-point methods.

To derive primal-dual interior-point methods we restate the optimality conditions (14.3) in a slightly different form by means of a mapping F from $\mathbb{R}^{2n+m}$ to $\mathbb{R}^{2n+m}$:

$$F(x, \lambda, s) = \begin{bmatrix} A^T\lambda + s - c \\ Ax - b \\ XSe \end{bmatrix} = 0, \tag{14.4a}$$
$$(x, s) \geq 0, \tag{14.4b}$$

where

$$X = \text{diag}(x_1, x_2, \ldots, x_n), \qquad S = \text{diag}(s_1, s_2, \ldots, s_n), \tag{14.5}$$

and $e = (1, 1, \ldots, 1)^T$. Primal-dual methods generate iterates (x^k, λ^k, s^k) that satisfy the bounds (14.4b) strictly, that is, $x^k > 0$ and $s^k > 0$. This property is the origin of the term *interior-point*. By respecting these bounds, the methods avoid spurious solutions, that is, points that satisfy $F(x, \lambda, s) = 0$ but not $(x, s) \geq 0$. Spurious solutions abound, and do not provide useful information about solutions of (14.1) or (14.2), so it makes sense to exclude them altogether from the region of search.

Like most iterative algorithms in optimization, primal-dual interior-point methods have two basic ingredients: a procedure for determining the step and a measure of the

desirability of each point in the search space. An important component of the measure of desirability is the average value of the pairwise products $x_i s_i$, $i = 1, 2, \ldots, n$, which are all positive when $x > 0$ and $s > 0$. This quantity is known as the *duality measure* and is defined as follows:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i s_i = \frac{x^T s}{n}. \tag{14.6}$$

The procedure for determining the search direction has its origins in Newton's method for the nonlinear equations (14.4a). Newton's method forms a linear model for F around the current point and obtains the search direction $(\Delta x, \Delta\lambda, \Delta s)$ by solving the following system of linear equations:

$$J(x, \lambda, s) \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = -F(x, \lambda, s),$$

where J is the Jacobian of F. (See Chapter 11 for a detailed discussion of Newton's method for nonlinear systems.) If we use the notation r_c and r_b for the first two block rows in F, that is,

$$r_b = Ax - b, \qquad r_c = A^T\lambda + s - c, \tag{14.7}$$

we can write the Newton equations as follows:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe \end{bmatrix}. \tag{14.8}$$

Usually, a full step along this direction would violate the bound $(x, s) \geq 0$, so we perform a line search along the Newton direction and define the new iterate as

$$(x, \lambda, s) + \alpha(\Delta x, \Delta\lambda, \Delta s),$$

for some line search parameter $\alpha \in (0, 1]$. We often can take only a small step along this direction ($\alpha \ll 1$) before violating the condition $(x, s) > 0$. Hence, the pure Newton direction (14.8), sometimes known as the *affine scaling direction*, often does not allow us to make much progress toward a solution.

Most primal-dual methods use a less aggressive Newton direction, one that does not aim directly for a solution of (14.3a), (14.3b), (14.3c) but rather for a point whose pairwise products $x_i s_i$ are reduced to a lower average value—not all the way to zero. Specifically, we

take a Newton step toward the a point for which $x_i s_i = \sigma\mu$, where μ is the current duality measure and $\sigma \in [0, 1]$ is the reduction factor that we wish to achieve in the duality measure on this step. The modified step equation is then

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe + \sigma\mu e \end{bmatrix}. \tag{14.9}$$

We call σ the *centering parameter*, for reasons to be discussed below. When $\sigma > 0$, it usually is possible to take a longer step α along the direction defined by (14.16) before violating the bounds $(x, s) \geq 0$.

At this point, we have specified most of the elements of a path-following primal-dual interior-point method. The general framework for such methods is as follows.

Framework 14.1 (Primal-Dual Path-Following).
Given (x^0, λ^0, s^0) with $(x^0, s^0) > 0$;
for $k = 0, 1, 2, \ldots$
Choose $\sigma_k \in [0, 1]$ and solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta\lambda^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} -r_c^k \\ -r_b^k \\ -X^k S^k e + \sigma_k \mu_k e \end{bmatrix}, \tag{14.10}$$

where $\mu_k = (x^k)^T s^k / n$;
Set

$$(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha_k(\Delta x^k, \Delta\lambda^k, \Delta s^k), \tag{14.11}$$

choosing α_k so that $(x^{k+1}, s^{k+1}) > 0$.
end (for).

The choices of centering parameter σ_k and step length α_k are crucial to the performance of the method. Techniques for controlling these parameters, directly and indirectly, give rise to a wide variety of methods with diverse properties.

Although software for implementing interior-point methods does not usually start from a point (x^0, λ^0, s^0) that is feasible with respect to the linear equations (14.3a) and (14.3b), most of the historical development of theory and algorithms assumed that these conditions are satisfied. In the remainder of this section, we discuss this feasible case, showing that a comprehensive convergence analysis can be presented in just a few pages, using only basic mathematical tools and concepts. Analysis of the infeasible case follows the

same principles, but is considerably more complicated in the details, so we do not present it here. In Section 14.2, however, we describe a complete practical algorithm that does not require starting from a feasible initial point.

To begin our discussion and analysis of feasible interior-point methods, we introduce the concept of the *central path*, and then describe neighborhoods of this path.

THE CENTRAL PATH

The primal-dual *feasible set* $\mathcal{F}$ and *strictly feasible set* $\mathcal{F}^o$ are defined as follows:

$$\mathcal{F} = \{(x, \lambda, s) \mid Ax = b,\ A^T\lambda + s = c,\ (x, s) \geq 0\}, \tag{14.12a}$$

$$\mathcal{F}^o = \{(x, \lambda, s) \mid Ax = b,\ A^T\lambda + s = c,\ (x, s) > 0\}. \tag{14.12b}$$

The central path $\mathcal{C}$ is an arc of strictly feasible points that plays a vital role in primal-dual algorithms. It is parametrized by a scalar $\tau > 0$, and each point $(x_\tau, \lambda_\tau, s_\tau) \in \mathcal{C}$ satisfies the following equations:

$$A^T\lambda + s = c, \tag{14.13a}$$

$$Ax = b, \tag{14.13b}$$

$$x_i s_i = \tau, \qquad i = 1, 2, \ldots, n, \tag{14.13c}$$

$$(x, s) > 0. \tag{14.13d}$$

These conditions differ from the KKT conditions only in the term τ on the right-hand side of (14.13c). Instead of the complementarity condition (14.3c), we require that the pairwise products $x_i s_i$ have the same (positive) value for all indices i. From (14.13), we can define the central path as

$$\mathcal{C} = \{(x_\tau, \lambda_\tau, s_\tau) \mid \tau > 0\}.$$

It can be shown that $(x_\tau, \lambda_\tau, s_\tau)$ is defined uniquely for each $\tau > 0$ if and only if $\mathcal{F}^o$ is nonempty.

The conditions (14.13) are also the optimality conditions for a logarithmic-barrier formulation of the problem (14.1). By introducing log-barrier terms for the nonnegativity constraints, with barrier parameter $\tau > 0$, we obtain

$$\min c^T x - \tau \sum_{i=1}^{n} \ln x_i, \quad \text{subject to } Ax = b. \tag{14.14}$$

The KKT conditions (12.34) for this problem, with Lagrange multiplier λ for the equality constraint, are as follows:

$$c_i - \frac{\tau}{x_i} - A_{\cdot i}^T \lambda, \quad i = 1, 2, \ldots, n, \quad Ax = b.$$

Since the objective in (14.14) is strictly convex, these conditions are sufficient as well as necessary for optimality. We recover (14.13) by defining $s_i = \tau / x_i$, $i = 1, 2, \ldots, n$.

Another way of defining $\mathcal{C}$ is to use the mapping F defined in (14.4) and write

$$F(x_\tau, \lambda_\tau, s_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix}, \qquad (x_\tau, s_\tau) > 0. \tag{14.15}$$

The equations (14.13) approximate (14.3) more and more closely as τ goes to zero. If $\mathcal{C}$ converges to anything as $\tau \downarrow 0$, it must converge to a primal-dual solution of the linear program. The central path thus guides us to a solution along a route that maintains positivity of the x and s components and decreases the pairwise products $x_i s_i$, $i = 1, 2, \ldots, n$ to zero at the same rate.

Most primal-dual algorithms take Newton steps toward points on $\mathcal{C}$ for which $\tau > 0$, rather than pure Newton steps for F. Since these steps are biased toward the interior of the nonnegative orthant defined by $(x, s) \geq 0$, it usually is possible to take longer steps along them than along the pure Newton (affine scaling) steps, before violating the positivity condition.

In the feasible case of $(x, \lambda, s) \in \mathcal{F}$, we have $r_b = 0$ and $r_c = 0$, so the search direction satisfies a special case of (14.8), that is,

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe + \sigma \mu e \end{bmatrix}, \tag{14.16}$$

where μ is the duality measure defined by (14.6) and $\sigma \in [0, 1]$ is the centering parameter. When $\sigma = 1$, the equations (14.16) define a *centering direction*, a Newton step toward the point $(x_\mu, \lambda_\mu, s_\mu) \in \mathcal{C}$, at which all the pairwise products $x_i s_i$ are identical to the current average value of μ. Centering directions are usually biased strongly toward the interior of the nonnegative orthant and make little, if any, progress in reducing the duality measure μ. However, by moving closer to $\mathcal{C}$, they set the scene for a substantial reduction in μ on the next iteration. At the other extreme, the value $\sigma = 0$ gives the standard Newton (affine scaling) step. Many algorithms use intermediate values of σ from

the open interval $(0, 1)$ to trade off between the twin goals of reducing μ and improving centrality.

CENTRAL PATH NEIGHBORHOODS AND PATH-FOLLOWING METHODS

Path-following algorithms explicitly restrict the iterates to a neighborhood of the central path $\mathcal{C}$ and follow $\mathcal{C}$ to a solution of the linear program. By preventing the iterates from coming too close to the boundary of the nonnegative orthant, they ensure that it is possible to take a nontrivial step along each search direction. Mopreover, by forcing the duality measure μ_k to zero as $k \to \infty$, we ensure that the iterates (x^k, λ^k, s^k) come closer and closer to satisfying the KKT conditions (14.3).

The two most interesting neighborhoods of $\mathcal{C}$ are

$$\mathcal{N}_2(\theta) = \{(x, \lambda, s) \in \mathcal{F}^o \mid \|XSe - \mu e\|_2 \leq \theta\mu\}, \tag{14.17}$$

for some $\theta \in [0, 1)$, and

$$\mathcal{N}_{-\infty}(\gamma) = \{(x, \lambda, s) \in \mathcal{F}^o \mid x_i s_i \geq \gamma\mu \quad \text{all } i = 1, 2, \ldots, n\}, \tag{14.18}$$

for some $\gamma \in (0, 1]$. (Typical values of the parameters are $\theta = 0.5$ and $\gamma = 10^{-3}$.) If a point lies in $\mathcal{N}_{-\infty}(\gamma)$, each pairwise product $x_i s_i$ must be at least some small multiple γ of their average value μ. This requirement is actually quite modest, and we can make $\mathcal{N}_{-\infty}(\gamma)$ encompass most of the feasible region $\mathcal{F}$ by choosing γ close to zero. The $\mathcal{N}_2(\theta)$ neighborhood is more restrictive, since certain points in $\mathcal{F}^o$ do not belong to $\mathcal{N}_2(\theta)$ no matter how close θ is chosen to its upper bound of 1.

By keeping all iterates inside one or other of these neighborhoods, path-following methods reduce all the pairwise products $x_i s_i$ to zero at more or less the same rate. Figure 14.1 shows the projection of the central path $\mathcal{C}$ onto the primal variables for a typical problem, along with a typical neighborhood $\mathcal{N}$.

Path-following methods are akin to homotopy methods for general nonlinear equations, which also define a path to be followed to the solution. Traditional homotopy methods stay in a tight tubular neighborhood of their path, making incremental changes to the parameter and chasing the homotopy path all the way to a solution. For primal-dual methods, this neighborhood is horn-shaped rather than tubular, and it tends to be broad and loose for larger values of the duality measure μ. It narrows as $\mu \to 0$, however, because of the positivity requirement $(x, s) > 0$.

The algorithm we specify below, a special case of Framework 14.1, is known as a *long-step path-following* algorithm. This algorithm can make rapid progress because of its use of the wide neighborhood $\mathcal{N}_{-\infty}(\gamma)$, for γ close to zero. It depends on two parameters $\sigma_{\min}$ and $\sigma_{\max}$, which are lower and upper bounds on the centering parameter σ_k. The search direction is, as usual, obtained by solving (14.10), and we choose the step length α_k to be as large as possible, subject to the requirement that we stay inside $\mathcal{N}_{-\infty}(\gamma)$.

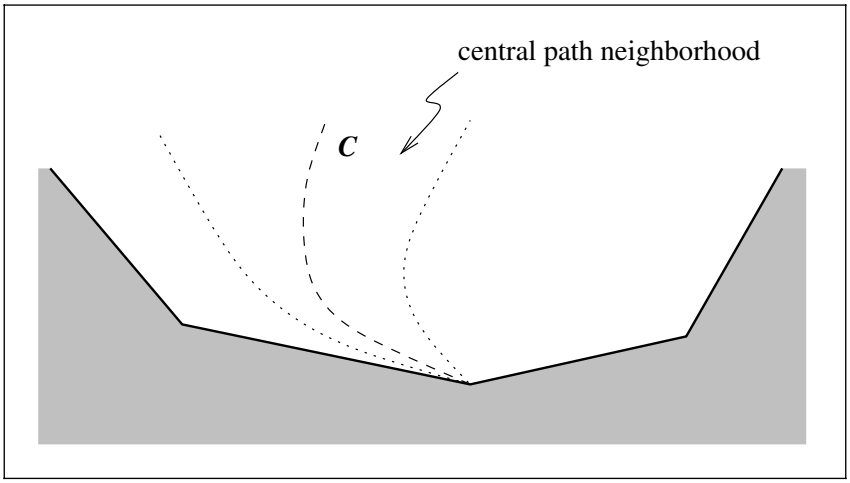

Figure 14.1 Central path, projected into space of primal variables x, showing a typical neighborhood $\mathcal{N}$.

Here and in later analysis, we use the notation

$$(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)) \stackrel{\text{def}}{=} (x^k, \lambda^k, s^k) + \alpha(\Delta x^k, \Delta\lambda^k, \Delta s^k), \tag{14.19a}$$

$$\mu_k(\alpha) \stackrel{\text{def}}{=} x^k(\alpha)^T s^k(\alpha)/n. \tag{14.19b}$$

Algorithm 14.2 (Long-Step Path-Following).

Given $\gamma, \sigma_{\min}, \sigma_{\max}$ with $\gamma \in (0, 1)$, $0 < \sigma_{\min} \leq \sigma_{\max} < 1$,
and $(x^0, \lambda^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$;
for $k = 0, 1, 2, \ldots$
Choose $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$;
Solve (14.10) to obtain $(\Delta x^k, \Delta\lambda^k, \Delta s^k)$;
Choose α_k as the largest value of α in $[0, 1]$ such that

$$(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)) \in \mathcal{N}_{-\infty}(\gamma); \tag{14.20}$$

Set $(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k(\alpha_k), \lambda^k(\alpha_k), s^k(\alpha_k))$;
end (for).

Typical behavior of the algorithm is illustrated in Figure 14.2 for the case of $n = 2$. The horizontal and vertical axes in this figure represent the pairwise products $x_1 s_1$ and $x_2 s_2$, so the central path $\mathcal{C}$ is the line emanating from the origin at an angle of 45°. (A point at the origin of this illustration is a primal-dual solution if it also satisfies the feasibility conditions

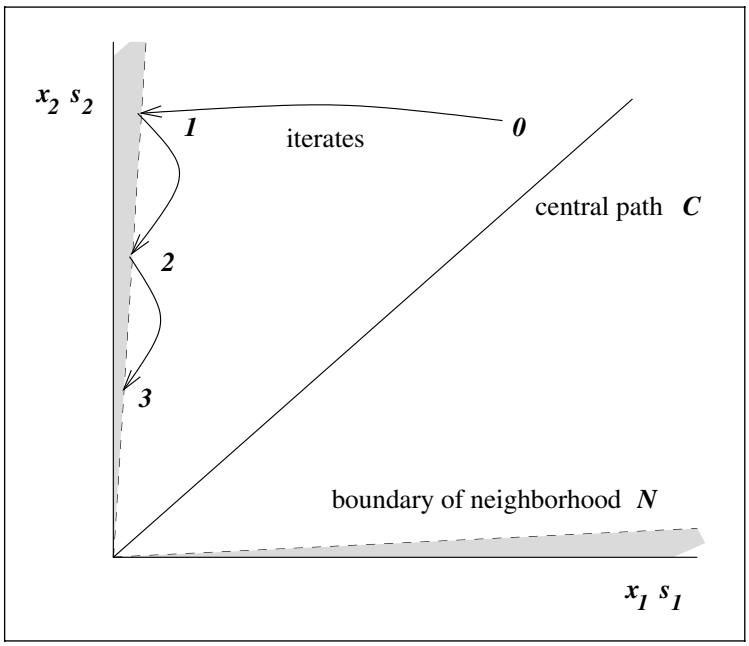

Figure 14.2 Iterates of Algorithm 14.2, plotted in (xs) space.

(14.3a), (14.3b), and (14.3d).) In the unusual geometry of Figure 14.2, the search directions $(\Delta x^k, \Delta\lambda^k, \Delta s^k)$ transform to curves rather than straight lines.

As Figure 14.2 shows (and the analysis confirms), the lower bound $\sigma_{\min}$ on the centering parameter ensures that each search direction starts out by moving away from the boundary of $\mathcal{N}_{-\infty}(\gamma)$ and into the relative interior of this neighborhood. That is, small steps along the search direction improve the centrality. Larger values of α take us outside the neighborhood again, since the error in approximating the nonlinear system (14.15) by the linear step equations (14.16) becomes more pronounced as α increases. Still, we are guaranteed that a certain minimum step can be taken before we reach the boundary of $\mathcal{N}_{-\infty}(\gamma)$, as we show in the analysis below.

The analysis of Algorithm 14.2 appears in the next few pages. With judicious choices of σ_k, this algorithm is fairly efficient in practice. With a few more modifications, it becomes the basis of a truly competitive method, as we discuss in Section 14.2.

Our aim in the analysis below is to show that given some small tolerance $\epsilon > 0$, the algorithm requires $O(n|\log\epsilon|)$ iterations to reduce the duality measure by a factor of ϵ, that is, to identify a point (x^k, λ^k, s^k) for which $\mu_k \leq \epsilon\mu_0$. For small ϵ, the point (x^k, λ^k, s^k) satisfies the primal-dual optimality conditions except for perturbations of about ϵ in the right-hand side of (14.3c), so it is usually very close to a primal-dual solution of the original linear program. The $O(n|\log\epsilon|)$ estimate is a worst-case bound on the number of iterations required; on practical problems, the number of iterations required appears

to increase only slightly (if at all) as n increases. The simplex method may require 2^n iterations to solve a problem with n variables, though in practice it usually requires a modest multiple of m iterations, where m is the row dimension of the constraint matrix A in (14.1).

As is typical for interior-point methods, the analysis builds from a purely technical lemma to a powerful theorem in just a few pages. We start with the technical result (Lemma 14.1) and use it to derive a bound on the vector of pairwise products $\Delta x_i \Delta s_i$, $i = 1, 2, \ldots, n$ (Lemma 14.2). Theorem 14.3 finds a lower bound on the step length α_k and a corresponding estimate of the reduction in μ on iteration k. Finally, Theorem 14.4 proves that $O(n|\log\epsilon|)$ iterations are required to identify a point for which $\mu_k < \epsilon$, for a given tolerance $\epsilon \in (0, 1)$.

Lemma 14.1.

Let u and v be any two vectors in $\mathbb{R}^n$ with $u^T v \geq 0$. Then

$$\|UVe\|_2 \leq 2^{-3/2}\|u + v\|_2^2,$$

where

$$U = \operatorname{diag}(u_1, u_2, \ldots, u_n), \qquad V = \operatorname{diag}(v_1, v_2, \ldots, v_n).$$

PROOF. (When the subscript is omitted from $\|\cdot\|$, we mean $\|\cdot\|_2$, as is our convention throughout the book.) First, note that for any two scalars α and β with $\alpha\beta \geq 0$, we have from the algebraic-geometric mean inequality that

$$\sqrt{|\alpha\beta|} \leq \frac{1}{2}|\alpha + \beta|. \tag{14.21}$$

Since $u^T v \geq 0$, we have

$$0 \leq u^T v = \sum_{u_i v_i \geq 0} u_i v_i + \sum_{u_i v_i < 0} u_i v_i = \sum_{i \in \mathcal{P}} |u_i v_i| - \sum_{i \in \mathcal{M}} |u_i v_i|, \tag{14.22}$$

where we partitioned the index set $\{1, 2, \ldots, n\}$ as

$$\mathcal{P} = \{i \mid u_i v_i \geq 0\}, \qquad \mathcal{M} = \{i \mid u_i v_i < 0\}.$$

Now,

$$\begin{aligned}
\|UVe\| &= \left(\|[u_i v_i]_{i\in\mathcal{P}}\|^2 + \|[u_i v_i]_{i\in\mathcal{M}}\|^2\right)^{1/2} \\
&\le \left(\|[u_i v_i]_{i\in\mathcal{P}}\|_1^2 + \|[u_i v_i]_{i\in\mathcal{M}}\|_1^2\right)^{1/2} && \text{since } \|\cdot\|_2 \le \|\cdot\|_1 \\
&\le \left(2\,\|[u_i v_i]_{i\in\mathcal{P}}\|_1^2\right)^{1/2} && \text{from (14.22)} \\
&\le \sqrt{2}\left\|\left[\frac{1}{4}(u_i+v_i)^2\right]_{i\in\mathcal{P}}\right\|_1 && \text{from (14.21)} \\
&= 2^{-3/2}\sum_{i\in\mathcal{P}}(u_i+v_i)^2 \\
&\le 2^{-3/2}\sum_{i=1}^{n}(u_i+v_i)^2 \\
&\le 2^{-3/2}\|u+v\|^2,
\end{aligned}$$

completing the proof. □

For the next result, we omit the iteration counter k from (14.10), and define the diagonal matrices ΔX and ΔS similarly to (14.5), as follows:

$$\Delta X = \operatorname{diag}(\Delta x_1, \Delta x_2, \ldots, \Delta x_n), \qquad \Delta S = \operatorname{diag}(\Delta s_1, \Delta s_2, \ldots, \Delta s_n)..$$

Lemma 14.2.
If $(x, \lambda, s) \in \mathcal{N}_{-\infty}(\gamma)$, then

$$\|\Delta X \Delta S e\| \le 2^{-3/2}(1 + 1/\gamma)n\mu.$$

PROOF. It is easy to show using (14.10) that

$$\Delta x^T \Delta s = 0. \tag{14.23}$$

By multiplying the last block row in (14.10) by $(XS)^{-1/2}$ and using the definition $D = X^{1/2}S^{-1/2}$, we obtain

$$D^{-1}\Delta x + D\Delta s = (XS)^{-1/2}(-XSe + \sigma\mu e). \tag{14.24}$$

Because $(D^{-1}\Delta x)^T(D\Delta s) = \Delta x^T \Delta s = 0$, we can apply Lemma 14.1 with $u = D^{-1}\Delta x$ and $v = D\Delta s$ to obtain

$$\begin{aligned}
\|\Delta X \Delta S e\| &= \|(D^{-1}\Delta X)(D\Delta S)e\| \\
&\le 2^{-3/2}\|D^{-1}\Delta x + D\Delta s\|^2 && \text{from Lemma 14.1} \\
&= 2^{-3/2}\|(XS)^{-1/2}(-XSe + \sigma\mu e)\|^2 && \text{from (14.24).}
\end{aligned}$$

Expanding the squared Euclidean norm and using such relationships as $x^T s = n\mu$ and

$e^T e = n$, we obtain

$$\begin{aligned}\|\Delta X \Delta S e\| &\le 2^{-3/2}\left[x^T s - 2\sigma\mu e^T e + \sigma^2\mu^2 \sum_{i=1}^{n} \frac{1}{x_i s_i}\right] \\ &\le 2^{-3/2}\left[x^T s - 2\sigma\mu e^T e + \sigma^2\mu^2 \frac{n}{\gamma\mu}\right] \qquad \text{since } x_i s_i \ge \gamma\mu \\ &\le 2^{-3/2}\left[1 - 2\sigma + \frac{\sigma^2}{\gamma}\right] n\mu \\ &\le 2^{-3/2}(1 + 1/\gamma) n\mu,\end{aligned}$$

as claimed. □

Theorem 14.3.

Given the parameters γ, $\sigma_{\min}$, and $\sigma_{\max}$ in Algorithm 14.2, there is a constant δ independent of n such that

$$\mu_{k+1} \le \left(1 - \frac{\delta}{n}\right)\mu_k, \tag{14.25}$$

for all $k \ge 0$.

PROOF. We start by proving that

$$\left(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)\right) \in \mathcal{N}_{-\infty}(\gamma) \quad \text{for all} \quad \alpha \in \left[0, 2^{3/2}\gamma\frac{1-\gamma}{1+\gamma}\frac{\sigma_k}{n}\right], \tag{14.26}$$

where $\left(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)\right)$ is defined as in (14.19). It follows that the step length α_k is at least as long as the upper bound of this interval, that is,

$$\alpha_k \ge 2^{3/2}\frac{\sigma_k}{n}\gamma\frac{1-\gamma}{1+\gamma}. \tag{14.27}$$

For any $i = 1, 2, \ldots, n$, we have from Lemma 14.2 that

$$|\Delta x_i^k \Delta s_i^k| \le \|\Delta X^k \Delta S^k e\|_2 \le 2^{-3/2}(1 + 1/\gamma) n\mu_k. \tag{14.28}$$

Using (14.10), we have from $x_i^k s_i^k \ge \gamma\mu_k$ and (14.28) that

$$\begin{aligned}x_i^k(\alpha)s_i^k(\alpha) &= \left(x_i^k + \alpha\Delta x_i^k\right)\left(s_i^k + \alpha\Delta s_i^k\right) \\ &= x_i^k s_i^k + \alpha\left(x_i^k \Delta s_i^k + s_i^k \Delta x_i^k\right) + \alpha^2 \Delta x_i^k \Delta s_i^k \\ &\ge x_i^k s_i^k (1-\alpha) + \alpha\sigma_k\mu_k - \alpha^2 |\Delta x_i^k \Delta s_i^k| \\ &\ge \gamma(1-\alpha)\mu_k + \alpha\sigma_k\mu_k - \alpha^2 2^{-3/2}(1 + 1/\gamma) n\mu_k.\end{aligned}$$

By summing the n components of the equation $S^k\Delta x^k + X^k\Delta s^k = -X^kS^ke + \sigma_k\mu_k e$ (the third block row from (14.10)), and using (14.23) and the definition of μ_k and $\mu_k(\alpha)$ (see (14.19)), we obtain

$$\mu_k(\alpha) = (1 - \alpha(1 - \sigma_k))\mu_k.$$

From these last two formulas, we can see that the proximity condition

$$x_i^k(\alpha)s_i^k(\alpha) \geq \gamma\mu_k(\alpha)$$

is satisfied, provided that

$$\gamma(1-\alpha)\mu_k + \alpha\sigma_k\mu_k - \alpha^2 2^{-3/2}(1 + 1/\gamma)n\mu_k \geq \gamma(1 - \alpha + \alpha\sigma_k)\mu_k.$$

Rearranging this expression, we obtain

$$\alpha\sigma_k\mu_k(1-\gamma) \geq \alpha^2 2^{-3/2}n\mu_k(1 + 1/\gamma),$$

which is true if

$$\alpha \leq \frac{2^{3/2}}{n}\sigma_k\gamma\frac{1-\gamma}{1+\gamma}.$$

We have proved that $\left(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)\right)$ satisfies the proximity condition for $\mathcal{N}_{-\infty}(\gamma)$ when α lies in the range stated in (14.26). It is not difficult to show that $\left(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)\right) \in \mathcal{F}^o$ for all α in the given range. Hence, we have proved (14.26) and therefore (14.27).

We complete the proof of the theorem by estimating the reduction in μ on the kth step. Because of (14.23), (14.27), and the last block row of (14.16), we have

$$\begin{aligned}
\mu_{k+1} &= x^k(\alpha_k)^T s^k(\alpha_k)/n \\
&= \left[(x^k)^T s^k + \alpha_k\left((x^k)^T\Delta s^k + (s^k)^T\Delta x^k\right) + \alpha_k^2(\Delta x^k)^T\Delta s^k\right]/n \\
&= \mu_k + \alpha_k\left(-(x^k)^T s^k/n + \sigma_k\mu_k\right) \\
&= (1 - \alpha_k(1-\sigma_k))\mu_k \\
&\leq \left(1 - \frac{2^{3/2}}{n}\gamma\frac{1-\gamma}{1+\gamma}\sigma_k(1-\sigma_k)\right)\mu_k. \qquad (14.29)
\end{aligned}$$

Now, the function $\sigma(1-\sigma)$ is a concave quadratic function of σ, so on any given interval it attains its minimum value at one of the endpoints. Hence, we have

$$\sigma_k(1-\sigma_k) \geq \min\left\{\sigma_{\min}(1-\sigma_{\min}), \sigma_{\max}(1-\sigma_{\max})\right\}, \quad \text{for all } \sigma_k \in [\sigma_{\min}, \sigma_{\max}].$$

The proof is completed by substituting this estimate into (14.29) and setting

$$\delta = 2^{3/2}\gamma \frac{1-\gamma}{1+\gamma} \min\left\{\sigma_{\min}(1-\sigma_{\min}), \sigma_{\max}(1-\sigma_{\max})\right\}. \qquad \square$$

We conclude with a result that a reduction of a factor of ϵ in the duality measure μ can be obtained in $O(n \log 1/\epsilon)$ iterations.

Theorem 14.4.

Given $\epsilon \in (0,1)$ and $\gamma \in (0,1)$, suppose the starting point in Algorithm 14.2 satisfies $(x^0, \lambda^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$. Then there is an index K with $K = O(n \log 1/\epsilon)$ such that

$$\mu_k \le \epsilon \mu_0, \qquad \text{for all } k \ge K.$$

PROOF. By taking logarithms of both sides in (14.25), we obtain

$$\log \mu_{k+1} \le \log\left(1 - \frac{\delta}{n}\right) + \log \mu_k.$$

By applying this formula repeatedly, we have

$$\log \mu_k \le k \log\left(1 - \frac{\delta}{n}\right) + \log \mu_0.$$

The following well-known estimate for the log function,

$$\log(1+\beta) \le \beta, \qquad \text{for all } \beta > -1,$$

implies that

$$\log(\mu_k/\mu_0) \le k\left(-\frac{\delta}{n}\right).$$

Therefore, the condition $\mu_k/\mu_0 \le \epsilon$ is satisfied if we have

$$k\left(-\frac{\delta}{n}\right) \le \log \epsilon.$$

This inequality holds for all k that satisfy

$$k \ge K \stackrel{\text{def}}{=} \frac{n}{\delta} \log \frac{1}{\epsilon} = \frac{n}{\delta} |\log \epsilon|,$$

so the proof is complete. $\square$

14.2 PRACTICAL PRIMAL-DUAL ALGORITHMS

Practical implementations of interior-point algorithms follow the spirit of the previous section, in that strict positivity of x^k and s^k is maintained throughout and each step is a Newton-like step involving a centering component. However, most implementations work with an infeasible starting point and infeasible iterations. Several aspects of "theoretical" algorithms are typically ignored, while several enhancements are added that have a significant effect on practical performance. In this section, we describe the algorithmic enhancements that are found in a typical implementation of an infeasible-interior-point method, and present the resulting method as Algorithm 14.3. Many of the techniques of this section are described in the paper of Mehrotra [207], which can be consulted for further details.

CORRECTOR AND CENTERING STEPS

A key feature of practical algorithms is their use of corrector steps that compensate for the linearization error made by the Newton (affine-scaling) step in modeling the equation $x_i s_i = 0, i = 1, 2, \ldots, n$ (see (14.3c)). Consider the affine-scaling direction $(\Delta x, \Delta\lambda, \Delta s)$ defined by

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^{\text{aff}} \\ \Delta\lambda^{\text{aff}} \\ \Delta s^{\text{aff}} \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe \end{bmatrix}, \tag{14.30}$$

(where r_b and r_c are defined in (14.7)). If we take a full step in this direction, we obtain

$$\begin{aligned} &(x_i + \Delta x_i^{\text{aff}})(s_i + \Delta s_i^{\text{aff}}) \\ &= x_i s_i + x_i \Delta s_i^{\text{aff}} + s_i \Delta x_i^{\text{aff}} + \Delta x_i^{\text{aff}} \Delta s_i^{\text{aff}} = \Delta x_i^{\text{aff}} \Delta s_i^{\text{aff}}. \end{aligned}$$

That is, the updated value of $x_i s_i$ is $\Delta x_i^{\text{aff}} \Delta s_i^{\text{aff}}$ rather than the ideal value 0. We can solve the following system to obtain a step $(\Delta x^{\text{cor}}, \Delta\lambda^{\text{cor}}, \Delta s^{\text{cor}})$ that attempts to correct for this deviation from the ideal:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^{\text{cor}} \\ \Delta\lambda^{\text{cor}} \\ \Delta s^{\text{cor}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\Delta X^{\text{aff}} \Delta S^{\text{aff}} e \end{bmatrix}. \tag{14.31}$$

In many cases, the combined step $(\Delta x^{\text{aff}}, \Delta\lambda^{\text{aff}}, \Delta s^{\text{aff}}) + (\Delta x^{\text{cor}}, \Delta\lambda^{\text{cor}}, \Delta s^{\text{cor}})$ does a better job of reducing the duality measure than does the affine-scaling step alone.

Like theoretical algorithms such as the one analysed in Section 14.1, practical algorithms make use of centering steps, with an adaptive choice of the centering parameter σ_k. The affine-scaling step can be used as the basis of a successful heuristic for choosing σ_k.

Roughly speaking, if the affine-scaling step (multiplied by a steplength to maintain nonnegativity of x and s) reduces the duality measure significantly, there is not much need for centering, so a smaller value of σ_k is appropriate. Conversely, if not much progress can be made along this direction before reaching the boundary of the nonnegative orthant, a larger value of σ_k will ensure that the next iterate is more centered, so a longer step will be possible from this next point. Specifically, this scheme calculates the maximum allowable steplengths along the affine-scaling direction (14.30) as follows:

$$\alpha_{\text{aff}}^{\text{pri}} \stackrel{\text{def}}{=} \min\left(1, \min_{i:\Delta x_i^{\text{aff}}<0} -\frac{x_i}{\Delta x_i^{\text{aff}}}\right), \tag{14.32a}$$

$$\alpha_{\text{aff}}^{\text{dual}} \stackrel{\text{def}}{=} \min\left(1, \min_{i:\Delta s_i^{\text{aff}}<0} -\frac{s_i}{\Delta s_i^{\text{aff}}}\right), \tag{14.32b}$$

and then defines μ_{aff} to be the value of μ that would be obtained by using these steplengths, that is,

$$\mu_{\text{aff}} = (x + \alpha_{\text{aff}}^{\text{pri}} \Delta x^{\text{aff}})^T (s + \alpha_{\text{aff}}^{\text{dual}} \Delta s^{\text{aff}})/n. \tag{14.33}$$

The centering parameter σ is chosen according to the following heuristic (which does not have a solid analytical justification, but appears to work well in practice):

$$\sigma = \left(\frac{\mu_{\text{aff}}}{\mu}\right)^3. \tag{14.34}$$

To summarize, computation of the search direction requires the solution of two linear systems. First, the system (14.30) is solved to obtain the affine-scaling direction, also known as the *predictor step*. This step is used to define the right-hand side for the corrector step (see (14.31)) and to calculate the centering parameter from (14.33), (14.34). Second, the search direction is calculated by solving

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe - \Delta X^{\text{aff}} \Delta S^{\text{aff}} e + \sigma\mu e \end{bmatrix}. \tag{14.35}$$

Note that the predictor, corrector, and centering contributions have been aggregated on the right-hand side of this system. The coefficient matrix in both linear systems (14.30) and (14.35) is the same. Thus, the factorization of the matrix needs to be computed only once, and the marginal cost of solving the second system is relatively small.

STEP LENGTHS

Practical implementations typically do not enforce membership of the central path neighborhoods $\mathcal{N}_2$ and $\mathcal{N}_{-\infty}$ defined in the previous section. Rather, they calculate the maximum steplengths that can be taken in the x and s variables (separately) without violating nonnegativity, then take a steplength of slightly less than this maximum (but no greater than 1). Given an iterate (x^k, λ^k, s^k) with $(x^k, s^k) > 0$, and a step $(\Delta x^k, \Delta\lambda^k, \Delta s^k)$, it is easy to show that the quantities $\alpha^{\text{pri}}_{k,\max}$ and $\alpha^{\text{dual}}_{k,\max}$ defined as follows:

$$\alpha^{\text{pri}}_{k,\max} \stackrel{\text{def}}{=} \min_{i:\Delta x^k_i<0} -\frac{x^k_i}{\Delta x^k_i}, \qquad \alpha^{\text{dual}}_{k,\max} \stackrel{\text{def}}{=} \min_{i:\Delta s^k_i<0} -\frac{s^k_i}{\Delta s^k_i}, \tag{14.36}$$

are the largest values of α for which $x^k + \alpha\Delta x^k \geq 0$ and $s^k + \alpha\Delta s^k \geq 0$, respectively. (Note that these formulae are similar to the ratio test used in the simplex method to determine the index that enters the basis.) Practical algorithms then choose the steplengths to lie in the *open* intervals defined by these maxima, that is,

$$\alpha^{\text{pri}}_k \in (0, \alpha^{\text{pri}}_{k,\max}), \qquad \alpha^{\text{dual}}_k \in (0, \alpha^{\text{dual}}_{k,\max}),$$

and then obtain a new iterate by setting

$$x^{k+1} = x^k + \alpha^{\text{pri}}_k \Delta x^k, \qquad (\lambda^{k+1}, s^{k+1}) = (\lambda^k, s^k) + \alpha^{\text{dual}}_k (\Delta\lambda^k, \Delta s^k).$$

If the step $(\Delta x^k, \Delta\lambda^k, \Delta s^k)$ rectifies the infeasibility in the KKT conditions (14.3a) and (14.3b), that is,

$$A\Delta x^k = -r^k_b = -(Ax^k - b), \qquad A^T\Delta\lambda^k + \Delta s^k = -r^k_c = -(A^T\lambda^k + s^k - c),$$

it is easy to show that the infeasibilities at the new iterate satisfy

$$r^{k+1}_b = \left(1 - \alpha^{\text{pri}}_k\right) r^k_b, \qquad r^{k+1}_c = \left(1 - \alpha^{\text{dual}}_k\right) r^k_c. \tag{14.37}$$

The following formula is used to calculate steplengths in many practical implementations

$$\alpha^{\text{pri}}_k = \min(1, \eta_k \alpha^{\text{pri}}_{k,\max}), \qquad \alpha^{\text{dual}}_k = \min(1, \eta_k \alpha^{\text{dual}}_{k,\max}), \tag{14.38}$$

where $\eta_k \in [0.9, 1.0)$ is chosen so that $\eta_k \to 1$ as the iterates approach the primal-dual solution, to accelerate the asymptotic convergence.

STARTING POINT

Choice of starting point is an important practical issue with a significant effect on the robustness of the algorithm. A poor choice (x^0, λ^0, s^0) satisfying only the minimal conditions $x^0 > 0$ and $s^0 > 0$ often leads to failure of convergence. We describe here a heuristic that finds a starting point that satisfies the equality constraints in the primal and dual problems reasonably well, while maintaining positivity of the x and s components and avoiding excessively large values of these components.

First, we find a vector $\tilde{x}$ of minimum norm satisfying the primal constraint $Ax = b$, and a vector $(\tilde{\lambda}, \tilde{s})$ satisfy the dual constraint $A^T\lambda + s = c$ such that $\tilde{s}$ has minimum norm. That is, we solve the problems

$$\min_x \tfrac{1}{2}x^T x \quad \text{subject to } Ax = b, \tag{14.39a}$$

$$\min_{(\lambda,s)} \tfrac{1}{2}s^T s \quad \text{subject to } A^T\lambda + s = c. \tag{14.39b}$$

It is not difficult to show that $\tilde{x}$ and $(\tilde{\lambda}, \tilde{s})$ can be written explicitly as follows:

$$\tilde{x} = A^T(AA^T)^{-1}b, \quad \tilde{\lambda} = (AA^T)^{-1}Ac, \quad \tilde{s} = c - A^T\tilde{\lambda}. \tag{14.40}$$

In general, $\tilde{x}$ and $\tilde{s}$ will have nonpositive components, so are not suitable for use as a starting point. We define

$$\delta_x = \max(-(3/2)\min_i \tilde{x}_i, 0), \quad \delta_s = \max(-(3/2)\min_i \tilde{s}_i, 0),$$

and adjust the x and s vectors as follows:

$$\hat{x} = \tilde{x} + \delta_x e, \quad \hat{s} = \tilde{s} + \delta_s e,$$

where, as usual, $e = (1, 1, \ldots, 1)^T$. Clearly, we have $\hat{x} \geq 0$ and $\hat{s} \geq 0$. To ensure that the components of x^0 and s^0 are not too close to zero and not too dissimilar, we add two more scalars defined as follows:

$$\hat{\delta}_x = \frac{1}{2}\frac{\hat{x}^T\hat{s}}{e^T\hat{s}}, \quad \hat{\delta}_s = \frac{1}{2}\frac{\hat{x}^T\hat{s}}{e^T\hat{x}}$$

(Note that $\hat{\delta}_x$ is the average size of the components of $\hat{x}$, weighted by the corresponding components of $\hat{s}$; similarly for $\hat{\delta}_s$.) Finally, we define the starting point as follows:

$$x^0 = \hat{x} + \hat{\delta}_x e, \quad \lambda^0 = \tilde{\lambda}, \quad s^0 = \hat{s} + \hat{\delta}_s e.$$

The computational cost of finding (x^0, λ^0, s^0) by this scheme is about the same as one step of the primal-dual method.

In some cases, we have prior knowledge about the solution, possibly in the form of a solution to a similar linear program. The use of such "warm-start" information in constructing a starting point is discussed in Section 14.4.

A PRACTICAL ALGORITHM

We now give a formal specification of a practical algorithm.

Algorithm 14.3 (Predictor-Corrector Algorithm (Mehrotra [207])).
Calculate (x^0, λ^0, s^0) as described above;

for $k = 0, 1, 2, \ldots$
Set $(x, \lambda, s) = (x^k, \lambda^k, s^k)$ and solve (14.30) for $(\Delta x^{\text{aff}}, \Delta\lambda^{\text{aff}}, \Delta s^{\text{aff}})$;
Calculate $\alpha_{\text{aff}}^{\text{pri}}$, $\alpha_{\text{aff}}^{\text{dual}}$, and μ_{aff} as in (14.32) and (14.33);
Set centering parameter to $\sigma = (\mu_{\text{aff}}/\mu)^3$;
Solve (14.35) for $(\Delta x, \Delta\lambda, \Delta s)$;
Calculate α_k^{pri} and α_k^{dual} from (14.38);
Set

$$x^{k+1} = x^k + \alpha_k^{\text{pri}} \Delta x,$$
$$(\lambda^{k+1}, s^{k+1}) = (\lambda^k, s^k) + \alpha_k^{\text{dual}} (\Delta\lambda, \Delta s);$$

end (for).

No convergence theory is available for Mehrotra's algorithm, at least in the form in which it is described above. In fact, there are examples for which the algorithm diverges. Simple safeguards could be incorporated into the method to force it into the convergence framework of existing methods or to improve its robustness, but many practical codes do not implement these safeguards, because failures are rare.

When presented with a linear program that is infeasible or unbounded, the algorithm above typically diverges, with the infeasibilities r_b^k and r_c^k and/or the duality measure μ_k going to ∞. Since the symptoms of infeasibility and unboundedness are fairly easy to recognize, interior-point codes contain heuristics to detect and report these conditions. More rigorous approaches for detecting infeasibility and unboundedness make use of the homogeneous self-dual formulation; see Wright [316, Chapter 9] and the references therein for a discussion. A more recent approach that applies directly to infeasible-interior-point methods is described by Todd [286].

SOLVING THE LINEAR SYSTEMS

Most of the computational effort in primal-dual methods is taken up in solving linear systems such as (14.9), (14.30), and (14.35). The coefficient matrix in these systems is usually large and sparse, since the constraint matrix A is itself large and sparse in most applications.

The special structure in the step equations allows us to reformulate them as systems with more compact symmetric coefficient matrices, which are easier and cheaper to factor than the original sparse form.

We apply the reformulation procedures to the following general form of the linear system:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -r_{xs} \end{bmatrix}. \tag{14.41}$$

Since x and s are strictly positive, the diagonal matrices X and S are nonsingular. We can eliminate Δs and add $-X^{-1}$ times the third equation in this system to the first equation to obtain

$$\begin{bmatrix} -D^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -r_c + X^{-1} r_{xs} \\ -r_b \end{bmatrix}, \tag{14.42a}$$

$$\Delta s = -X^{-1} r_{xs} - X^{-1} S \Delta x, \tag{14.42b}$$

where we have introduced the notation

$$D = S^{-1/2} X^{1/2}. \tag{14.43}$$

This form of the step equations usually is known as the *augmented system*. We can go further and eliminate Δx and add AD^2 times the first equation to the second equation in (14.42a) to obtain

$$AD^2 A^T \Delta\lambda = -r_b - AXS^{-1} r_c + AS^{-1} r_{xs} \tag{14.44a}$$

$$\Delta s = -r_c - A^T \Delta\lambda, \tag{14.44b}$$

$$\Delta x = -S^{-1} r_{xs} - XS^{-1} \Delta s, \tag{14.44c}$$

where the expressions for Δs and Δs are obtained from the original system (14.41). The form (14.44a) often is called the *normal-equations* form, because the system (14.44a) can be viewed as the normal equations (10.14) for a certain linear least-squares problem with coefficient matrix DA^T.

Most implementations of primal-dual methods are based on formulations like (14.44). They use direct sparse Cholesky algorithms to factor the matrix AD^2A^T, and then perform triangular solves with the resulting sparse factors to obtain the step $\Delta\lambda$ from (14.44a). The steps Δs and Δx are recovered from (14.44b) and (14.44c). General-purpose sparse Cholesky software can be applied to AD^2A^T, but modifications are needed because AD^2A^T may be ill-conditioned or singular. (Ill conditioning of this system is often observed during

the final stages of a primal-dual algorithm, when the elements of the diagonal weighting matrix D^2 take on both huge and tiny values.) The Cholesky technique may encounter diagonal elements that are very small, zero or (because of roundoff error) slightly negative. One approach for handling this eventuality is to skip a step of the factorization, setting the component of $\Delta\lambda$ that corresponds to the faulty diagonal element to zero. We refer to Wright [317] for details of this and other approaches.

A disadvantage of the normal-equations formulation is that if A contains any dense columns, the entire matrix AD^2A^T is also dense. Hence, practical software identifies dense and nearly-dense columns, excludes them from the matrix product AD^2A^T, and performs the Cholesky factorization of the resulting sparse matrix. Then, a device such as a Sherman-Morrison-Woodbury update is applied to account for the excluded columns. We refer the reader to Wright [316, Chapter 11] for further details.

The formulation (14.42) has received less attention than (14.44), mainly because algorithms and software for factoring sparse symmetric indefinite matrices are more complicated, slower, and less prevalent than sparse Cholesky algorithms. Nevertheless, the formulation (14.42) is cleaner and more flexible than (14.44) in a number of respects. It normally avoids the fill-in caused by dense columns in A in the matrix product AD^2A^T. Moreover, it allows free variables (components of x with no explicit lower or upper bounds) to be handled directly in the formulation. (The normal equations form must resort to various artificial devices to express such variables, otherwise it is not possible to perform the block elimination that leads to the system (14.44a).)

14.3 OTHER PRIMAL-DUAL ALGORITHMS AND EXTENSIONS

OTHER PATH-FOLLOWING METHODS

Framework 14.1 is the basis of a number of other algorithms of the path-following variety. They are less important from a practical viewpoint, but we mention them here because of their elegance and their strong theoretical properties.

Some path-following methods choose conservative values for the centering parameter σ (that is, σ only slightly less than 1) so that unit steps (that is, a steplength of $\alpha = 1$) can be taken along the resulting direction from (14.16) without leaving the chosen neighborhood. These methods, which are known as *short-step* path-following methods, make only slow progress toward the solution because they require the iterates to stay inside a restrictive $\mathcal{N}_2$ neighborhood (14.17). From a theoretical point of view, however, they have the advantage of better complexity. (A result similar to Theorem 14.4 holds with n replaced by $n^{1/2}$ in the complexity estimate.)

Better results are obtained with the *predictor-corrector* method, due to Mizuno, Todd, and Ye [208], which uses two $\mathcal{N}_2$ neighborhoods, nested one inside the other. (Despite the similar terminology, this algorithm is quite distinct from Algorithm 14.3 of Section 14.2.) Every second step of this method is a predictor step, which starts in the inner neighborhood

and moves along the affine-scaling direction (computed by setting $\sigma = 0$ in (14.16)) to the boundary of the outer neighborhood. The gap between neighborhood boundaries is wide enough to allow this step to make significant progress in reducing μ. Alternating with the predictor steps are corrector steps (computed with $\sigma = 1$ and $\alpha = 1$), which take the next iterate back inside the inner neighborhood in preparation for the next predictor step. The predictor-corrector algorithm produces a sequence of duality measures μ_k that converge superlinearly to zero, in contrast to the linear convergence that characterizes most methods.

POTENTIAL-REDUCTION METHODS

Potential-reduction methods take steps of the same form as path-following methods, but they do not explicitly follow the central path $\mathcal{C}$ and can be motivated independently of it. They use a logarithmic potential function to measure the worth of each point in $\mathcal{F}^o$ and aim to achieve a certain fixed reduction in this function at each iteration. The primal-dual potential function, which we denote generically by Φ, usually has two important properties:

$$\Phi \to \infty \quad \text{if } x_i s_i \to 0 \text{ for some } i, \text{ while } \mu = x^T s/n \not\to 0, \tag{14.45a}$$

$$\Phi \to -\infty \quad \text{if and only if } (x, \lambda, s) \to \Omega. \tag{14.45b}$$

The first property (14.45a) prevents any one of the pairwise products $x_i s_i$ from approaching zero independently of the others, and therefore keeps the iterates away from the boundary of the nonnegative orthant. The second property (14.45b) relates Φ to the solution set Ω. If our algorithm forces Φ to $-\infty$, then (14.45b) ensures that the sequence approaches the solution set.

An interesting primal-dual potential function is defined by

$$\Phi_\rho(x, s) = \rho \log x^T s - \sum_{i=1}^{n} \log x_i s_i, \tag{14.46}$$

for some parameter $\rho > n$ (see Tanabe [283] and Todd and Ye [287]). Like all algorithms based on Framework 14.1, potential-reduction algorithms obtain their search directions by solving (14.10), for some $\sigma_k \in (0, 1)$, and they take steps of length α_k along these directions. For instance, the step length α_k may be chosen to approximately minimize Φ_ρ along the computed direction. By fixing $\sigma_k = n/(n + \sqrt{n})$ for all k, one can guarantee constant reduction in Φ_ρ at every iteration. Hence, Φ_ρ will approach $-\infty$, forcing convergence. Adaptive and heuristic choices of σ_k and α_k are also covered by the theory, provided that they at least match the reduction in Φ_ρ obtained from the conservative theoretical values of these parameters.

EXTENSIONS

Primal-dual methods for linear programming can be extended to wider classes of problems. There are simple extensions of the algorithm to the monotone linear complementarity problem (LCP) and convex quadratic programming problems for which the convergence and polynomial complexity properties of the linear programming algorithms are retained. The monotone LCP is the problem of finding vectors x and s in $\mathbb{R}^n$ that satisfy the following conditions:

$$s = Mx + q, \qquad (x, s) \geq 0, \qquad x^T s = 0, \tag{14.47}$$

where M is a positive semidefinite $n \times n$ matrix and $q \in \mathbb{R}^n$. The similarity between (14.47) and the KKT conditions (14.3) is obvious: The last two conditions in (14.47) correspond to (14.3d) and (14.3c), respectively, while the condition $s = Mx + q$ is similar to the equations (14.3a) and (14.3b). For practical instances of the problem (14.47), see Cottle, Pang, and Stone [80]. Interior-point methods for monotone LCP have a close correspondence to algorithms for linear programming. The duality measure (14.6) is redefined to be the *complementarity measure* (with the same definition $\mu = x^T s/n$), and the conditions that must be satisfied by the solution can be stated similarly to (14.4) as follows:

$$\begin{bmatrix} Mx + q - s \\ XSe \end{bmatrix} = 0, \quad (x, s) \geq 0.$$

The general formula for a path-following step is defined analogously to (14.9) as follows:

$$\begin{bmatrix} M & -I \\ S & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \end{bmatrix} = \begin{bmatrix} -(Mx + q - s) \\ -XSe + \sigma \mu e \end{bmatrix},$$

where $\sigma \in [0, 1]$. Using these and similar adaptations, an extension of the practical method of Section 14.2 can also be derived.

Extensions to convex quadratic programs are discussed in Section 16.6. Their adaptation to nonlinear programming problems is the subject of Chapter 19.

Interior-point methods are highly effective in solving *semidefinite programming* problems, a class of problems involving symmetric matrix variables that are constrained to be positive semidefinite. Semidefinite programming, which has been the topic of concentrated research since the early 1990s, has applications in many areas, including control theory and combinatorial optimization. Further information on this increasingly important topic can be found in the survey papers of Todd [285] and Vandenberghe and Boyd [292] and the books of Nesterov and Nemirovskii [226], Boyd et al. [37], and Boyd and Vandenberghe [38].

14.4 PERSPECTIVES AND SOFTWARE

The appearance of interior-point methods in the 1980s presented the first serious challenge to the dominance of the simplex method as a practical means of solving linear programming problems. By about 1990, interior-point codes had emerged that incorporated the techniques described in Section 14.2 and that were superior on many large problems to the simplex codes available at that time. The years that followed saw significant improvements in simplex software, evidenced by the appearance of packages such as CPLEX and XPRESS-MP. These improvements were due to algorthmic advances such as steepest-edge pivoting (see Goldfarb and Forrest [133]) and improved pricing heuristics, and also to close attention to the nuts and bolts of efficient implementation. The efficiency of interior-point codes also continued to improve, through improvements in the linear algebra for solving the step equations and through the use of higher-order correctors in the step calculation (see Gondzio [138]). During this period, a number of good interior-point codes became freely available (such as PCx [84], HOPDM [137], BPMPD, and LIPSOL [321]) and found their way into many applications.

In general, simplex codes are faster on problems of small-medium dimensions, while interior-point codes are competitive and often faster on large problems. However, this rule is certainly not hard-and-fast; it depends strongly on the structure of the particular application. Interior-point methods are generally not able to take full advantage of prior knowledge about the solution, such as an estimate of the solution itself or an estimate of the optimal basis. Hence, interior-point methods are less useful than simplex approaches in situations in which "warm-start" information is readily available. One situation of this type involves branch-and-bound algorithms for solving integer programs, where each node in the branch-and-bound tree requires the solution of a linear program that differs only slightly from one already solved in the parent node. In other situations, we may wish to solve a sequence of linear programs in which the data is perturbed slightly to investigate sensitivity of the solutions to various perturbations, or in which we approximate a nonlinear optimization problem by a sequence of linear programs. Yıldırım and Wright [319] describe how a given point (such as an approximate solution) can be modified to obtain a starting point that is theoretically valid, in that it allows complexity results to be proved that depend on the quality of the given point. In practice, however, these techniques can be expected to provide only a modest improvement in algorithmic performance (perhaps a factor of between 2 and 5) over a "cold" starting point such as the one described in Section 14.2.

Interior-point software has the advantage that it is easy to program, relative to the simplex method. The most complex operation is the solution of the large linear systems at each iteration to compute the step; software to perform this linear algebra operation is readily available. The interior-point code LIPSOL [321] is written entirely in the Matlab language, apart from a small amount of FORTRAN code that interfaces to the linear algebra software. The code PCx [84] is written in C, but also is easy for the interested user to comprehend and modify. It is even possible for a non-expert in optimization to write an

efficient interior-point implementation from scratch that is customized to their particular application.

NOTES AND REFERENCES

For more details on the material of this chapter, see the book by Wright [316].

As noted in the text, Karmarkar's method arose from a search for linear programming algorithms with better worst-case behavior than the simplex method. The first algorithm with polynomial complexity, Khachiyan's ellipsoid algorithm [180], was a computational disappointment. In contrast, the execution times required by Karmarkar's method were not too much greater than simplex codes at the time of its introduction, particularly for large linear programs. Karmarkar's is a *primal* algorithm; that is, it is described, motivated, and implemented purely in terms of the primal problem (14.1) without reference to the dual. At each iteration, Karmarkar's algorithm performs a projective transformation on the primal feasible set that maps the current iterate x^k to the center of the set and takes a step in the feasible steepest descent direction for the transformed space. Progress toward optimality is measured by a logarithmic potential function. Descriptions of the algorithm can be found in Karmarkar's original paper [175] and in Fletcher [101, Section 8.7].

Karmarkar's method falls outside the scope of this chapter, and in any case, its practical performance does not appear to match the most efficient primal-dual methods. The algorithms we discussed in this chapter have polynomial complexity, like Karmarkar's method.

Many of the algorithmic ideas that have been examined since 1984 actually had their genesis in three works that preceded Karmarkar's paper. The first of these is the book of Fiacco and McCormick [98] on logarithmic barrier functions (originally proposed by Frisch [115]), which proves existence of the central path, among many other results. Further analysis of the central path was carried out by McLinden [205], in the context of nonlinear complementarity problems. Finally, there is Dikin's paper [94], in which an interior-point method known as primal affine-scaling was originally proposed. The outburst of research on primal-dual methods, which culminated in the efficient software packages available today, dates to the seminal paper of Megiddo [206].

Todd gives an excellent survey of potential reduction methods in [284]. He relates the primal-dual potential reduction method mentioned above to pure primal potential reduction methods, including Karmarkar's original algorithm, and discusses extensions to special classes of nonlinear problems.

For an introduction to complexity theory and its relationship to optimization, see the book by Vavasis [297].

Andersen et al. [6] cover many of the practical issues relating to implementation of interior-point methods. In particular, they describe an alternative scheme for choosing the initial point, for the case in which upper bounds are also present on the variables.

EXERCISES

14.1 This exercise illustrates the fact that the bounds $(x, s) \geq 0$ are essential in relating solutions of the system (14.4a) to solutions of the linear program (14.1) and its dual. Consider the following linear program in $\mathbb{R}^2$:

$$\min x_1, \quad \text{subject to } x_1 + x_2 = 1, \qquad (x_1, x_2) \geq 0.$$

Show that the primal-dual solution is

$$x^* = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \lambda^* = 0, \quad s^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Also verify that the system $F(x, \lambda, s) = 0$ has the spurious solution

$$x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \lambda = 1, \quad s = \begin{pmatrix} 0 \\ -1 \end{pmatrix},$$

which has no relation to the solution of the linear program.

14.2

(i) Show that $\mathcal{N}_2(\theta_1) \subset \mathcal{N}_2(\theta_2)$ when $0 \leq \theta_1 < \theta_2 < 1$ and that $\mathcal{N}_{-\infty}(\gamma_1) \subset \mathcal{N}_{-\infty}(\gamma_2)$ for $0 < \gamma_2 \leq \gamma_1 \leq 1$.

(ii) Show that $\mathcal{N}_2(\theta) \subset \mathcal{N}_{-\infty}(\gamma)$ if $\gamma \leq 1 - \theta$.

14.3 Given an arbitrary point $(x, \lambda, s) \in \mathcal{F}^o$, find the range of γ values for which $(x, \lambda, s) \in \mathcal{N}_{-\infty}(\gamma)$. (The range depends on x and s.)

14.4 For $n = 2$, find a point $(x, s) > 0$ for which the condition

$$\|XSe - \mu e\|_2 \leq \theta \mu$$

is *not* satisfied for any $\theta \in [0, 1)$.

14.5 Prove that the neighborhoods $\mathcal{N}_{-\infty}(1)$ (see (14.18)) and $\mathcal{N}_2(0)$ (see (14.17)) coincide with the central path $\mathcal{C}$.

14.6 In the long-step path-following method (Algorithm 14.2), give a procedure for calculating the maximum value of α such that (14.20) is satisfied.

14.7 Show that Φ_ρ defined by (14.46) has the property (14.45a).

14.8 Prove that the coefficient matrix in (14.16) is nonsingular if and only if A has full row rank.

✎ **14.9** Given $(\Delta x, \Delta\lambda, \Delta s)$ satisfying (14.10), prove (14.23).

✎ **14.10** Given an iterate (x^k, λ^k, s^k) with $(x^k, s^k) > 0$, show that the quantities $\alpha^{\text{pri}}_{\max}$ and $\alpha^{\text{dual}}_{\max}$ defined by (14.36) are the largest values of α such that $x^k + \alpha\Delta x^k \geq 0$ and $s^k + \alpha\Delta s^k \geq 0$, respectively.

✎ **14.11** Verify (14.37).

✎ **14.12** Given that X and S are diagonal with positive diagonal elements, show that the coefficient matrix in (14.44a) is symmetric and positive definite if and only if A has full row rank. Does this result continue to hold if we replace D by a diagonal matrix in which exactly m of the diagonal elements are positive and the remainder are zero? (Here m is the number of rows of A.)

✎ **14.13** Given a point (x, λ, s) with $(x, s) > 0$, consider the trajectory $\mathcal{H}$ defined by

$$F\left(\hat{x}(\tau), \hat{\lambda}(\tau), \hat{s}(\tau)\right) = \begin{bmatrix} (1-\tau)(A^T\lambda + s - c) \\ (1-\tau)(Ax - b) \\ (1-\tau)XSe \end{bmatrix}, \qquad (\hat{x}(\tau), \hat{s}(\tau)) \geq 0,$$

for $\tau \in [0, 1]$, and note that $\left(\hat{x}(0), \hat{\lambda}(0), \hat{s}(0)\right) = (x, \lambda, s)$, while the limit of $\left(\hat{x}(\tau), \hat{\lambda}(\tau), \hat{s}(\tau)\right)$ as $\tau \uparrow 1$ will lie in the primal-dual solution set of the linear program. Find equations for the first, second, and third derivatives of $\mathcal{H}$ with respect to τ at $\tau = 0$. Hence, write down a Taylor series approximation to $\mathcal{H}$ near the point (x, λ, s).

✎ **14.14** Consider the following linear program, which contains "free variables" denoted by y:

$$\min c^T x + d^T y, \quad \text{subject to } A_1 x + A_2 y = b, x \geq 0.$$

By introducing Lagrange multipliers λ for the equality constraints and s for the bounds $x \geq 0$, write down optimality conditions for this problem in an analogous fashion to (14.3). Following (14.4) and (14.16), use these conditions to derive the general step equations for a primal-dual interior-point method. Express these equations in augmented system form analogously to (14.42) and explain why it is not possible to reduce further to a formulation like (14.44) in which the coefficient matrix is symmetric positive definite.

✎ **14.15** Program Algorithm 14.3 in Matlab. Choose $\eta = 0.99$ uniformly in (14.38). Test your code on a linear programming problem (14.1) generated by choosing A randomly,

and then setting x, s, b, and c as follows:

$$
\begin{aligned}
x_i &= \begin{cases} \text{random positive number} & i = 1, 2, \ldots, m, \\ 0 & i = m+1, m+2, \ldots, n, \end{cases} \\
s_i &= \begin{cases} \text{random positive number} & i = m+1, m+2, \ldots, n \\ 0 & i = 1, 2, \ldots, m, \end{cases} \\
\lambda &= \text{random vector}, \\
c &= A^T\lambda + s, \\
b &= Ax.
\end{aligned}
$$

Choose the starting point (x^0, λ^0, s^0) with the components of x^0 and s^0 set to large positive values.

✎ **14.17** Show that the solutions of the problems (14.39) are given explicitly by (14.40).