Chapter 5
Interior-Point Methods

Linear programs can be viewed in two somewhat complementary ways. They are, in one view, a class of continuous optimization problems each with continuous variables defined on a convex feasible region and with a continuous objective function. They are, therefore, a special case of the general form of problem considered in this text. However, linearity implies a certain degree of degeneracy, since for example the derivatives of all functions are constants and hence the differential methods of general optimization theory cannot be directly used. From an alternative view, linear programs can be considered as a class of combinatorial problems because it is known that solutions can be found by restricting attention to the vertices of the convex polyhedron defined by the constraints. Indeed, this view is natural when considering network problems such as those of early chapters. However, the number of vertices may be large, up to $n!/m!(n-m)$!, making direct search impossible for even modest size problems.

The simplex method embodies both of these viewpoints, for it restricts attention to vertices, but exploits the continuous nature of the variables to govern the progress from one vertex to another, defining a sequence of adjacent vertices with improving values of the objective as the process reaches an optimal point. The simplex method, with ever-evolving improvements, has for five decades provided an efficient general method for solving linear programs.

Although it performs well in practice, visiting only a small fraction of the total number of vertices, a definitive theory of the simplex method's performance was unavailable. However, in 1972, Klee and Minty showed by examples that for certain linear programs the simplex method will examine every vertex f. These examples proved that in the worst case, the simplex method requires a number of steps that is exponential in the size of the problem.

D.G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, International
Series in Operations Research & Management Science 228,
DOI 10.1007/978-3-319-18842-3_5

In view of this result, many researchers believed that a good algorithm, different than the simplex method, might be devised whose number of steps would be polynomial rather than exponential in the program's size—that is, the time required to compute the solution would be bounded above by a polynomial in the size of the problem.[1]

Indeed, in 1979, a new approach to linear programming, Khachiyan's ellipsoid method was announced with great acclaim. The method is quite different in structure than the simplex method, for it constructs a sequence of shrinking ellipsoids each of which contains the optimal solution set and each member of the sequence is smaller in volume than its predecessor by at least a certain fixed factor. Therefore, the solution set can be found to any desired degree of approximation by continuing the process. Khachiyan proved that the ellipsoid method, developed during the 1970s by other mathematicians, is a polynomial-time algorithm for linear programming.

Practical experience, however, was disappointing. In almost all cases, the simplex method was much faster than the ellipsoid method. However, Khachiyan's ellipsoid method showed that polynomial time algorithms for linear programming do exist. It left open the question of whether one could be found that, in practice, was faster than the simplex method.

It is then perhaps not surprising that the announcement by Karmarkar in 1984 of a new polynomial time algorithm, an interior-point method, with the potential to improve the practical effectiveness of the simplex method made front-page news in major newspapers and magazines throughout the world. It is this interior-point approach that is the subject of this chapter and the next.

This chapter begins with a brief introduction to complexity theory, which is the basis for a way to quantify the performance of iterative algorithms, distinguishing polynomial-time algorithms from others.

Next the example of Klee and Minty showing that the simplex method is not a polynomial-time algorithm in the worst case is presented. Following that the ellipsoid algorithm is defined and shown to be a polynomial-time algorithm. These two sections provide a deeper understanding of how the modern theory of linear programming evolved, and help make clear how complexity theory impacts linear programming. However, the reader may wish to consider them optional and omit them at first reading.

The development of the basics of interior-point theory begins with Sect. 5.4 which introduces the concept of barrier functions and the analytic center. Section 5.5 introduces the central path which underlies interior-point algorithms. The relations between primal and dual in this context are examined. An overview of the details of specific interior-point algorithms based on the theory are presented in Sects. 5.6 and 5.7

[1] We will be more precise about complexity notions such as "polynomial algorithm" in Sect. 5.1 below.

5.1 Elements of Complexity Theory

Complexity theory is arguably the foundation for analysis of computer algorithms. The goal of the theory is twofold: to develop criteria for measuring the effectiveness of various algorithms (and thus, be able to compare algorithms using these criteria), and to assess the inherent difficulty of various problems.

The term *complexity* refers to the amount of resources required by a computation. In this chapter we focus on a particular resource, namely, computing time. In complexity theory, however, one is not interested in the execution time of a program implemented in a particular programming language, running on a particular computer over a particular input. This involves too many contingent factors. Instead, one wishes to associate to an algorithm more intrinsic measures of its time requirements.

Roughly speaking, to do so one needs to define:

- a notion of *input size*,
- a set of *basic operations*, and
- a *cost* for each basic operation.

The last two allow one to associate a *cost of a computation*. If x is any input, the cost $C(x)$ of the computation with input x is the sum of the costs of all the basic operations performed during this computation.

Let $\mathcal{A}$ be an algorithm and $\mathcal{J}_n$ be the set of all its inputs having size n. The *worst-case cost function* of $\mathcal{A}$ is the function $T^w_{\mathcal{A}}$ defined by

$$T^w_{\mathcal{A}}(n) = \sup_{x \in \mathcal{J}_n} C(x).$$

If there is a probability structure on $\mathcal{J}_n$ it is possible to define the *average-case cost function* $T^a_{\mathcal{A}}$ given by

$$T^a_{\mathcal{A}}(n) = \mathrm{E}_n(C(x)).$$

where E_n is the expectation over $\mathcal{J}_n$. However, the average is usually more difficult to find, and there is of course the issue of what probabilities to assign.

We now discuss how the objects in the three items above are selected. The selection of a set of basic operations is generally easy. For the algorithms we consider in this chapter, the obvious choice is the set $\{+, -, \times, /, \leq\}$ of the four arithmetic operations and the comparison. Selecting a notion of input size and a cost for the basic operations depends on the kind of data dealt with by the algorithm. Some kinds can be represented within a fixed amount of computer memory; others require a variable amount.

Examples of the first are fixed-precision floating-point numbers, stored in a fixed amount of memory (usually 32 or 64 bits). For this kind of data the size of an element is usually taken to be 1 and consequently to have *unit size* per number.

Examples of the second are integer numbers which require a number of bits approximately equal to the logarithm of their absolute value. This (base 2) logarithm is usually referred to as the *bit size* of the integer. Similar ideas apply for rational numbers.

Let A be some kind of data and $\mathbf{x} = (x_1, \ldots, x_n) \in A^n$. If A is of the first kind above then we define size($\mathbf{x}$) = n. Otherwise, we define size($\mathbf{x}$) = $\sum_{i=1}^n$ bit-size (x_i).

The cost of operating on two unit-size numbers is taken to be 1 and is called the *unit cost*. In the bit-size case, the cost of operating on two numbers is the product of their bit-sizes (for multiplications and divisions) or their maximum (for additions, subtractions, and comparisons).

The consideration of integer or rational data with their associated bit size and bit cost for the arithmetic operations is usually referred to as the *Turing model of computation*. The consideration of idealized reals with unit size and unit cost is today referred as the *real number arithmetic model*. When comparing algorithms, one should make clear which model of computation is used to derive complexity bounds.

A basic concept related to both models of computation is that of *polynomial time*. An algorithm $\mathcal{A}$ is said to be a polynomial time algorithm if $T^w_{\mathcal{A}}(n)$ is bounded above by a polynomial. A problem can be solved in polynomial time if there is a polynomial time algorithm solving the problem. The notion of *average polynomial time* is defined similarly, replacing $T^w_{\mathcal{A}}$ by $T^a_{\mathcal{A}}$.

The notion of polynomial time is usually taken as the formalization of efficiency in complexity theory.

*5.2 *The Simplex Method Is Not Polynomial-Time

When the simplex method is used to solve a linear program in standard form with coefficient matrix $\mathbf{A} \in E^{m \times n}$, $\mathbf{b} \in E^m$ and $\mathbf{c} \in E^n$, the number of pivot steps to solve the problem starting from a basic feasible solution is typically a small multiple of m: usually between $2m$ and $3m$. In fact, Dantzig observed that for problems with $m \leq 50$ and $n \leq 200$ the number of iterations is ordinarily less than $1.5m$.

At one time researchers believed—and attempted to prove—that the simplex algorithm (or some variant thereof) always requires a number of iterations that is bounded by a polynomial expression in the problem size. That was until Victor Klee and George Minty exhibited a class of linear programs each of which requires an exponential number of iterations when solved by the conventional simplex method.

One form of the Klee–Minty example is

$$\begin{aligned} &\text{maximize} \quad \sum_{j=1}^{n} 10^{n-j} x_j \\ &\text{subject to} \quad 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i \leq 100^{i-1} i = 1, \ldots, n \\ &\qquad\qquad\qquad\qquad x_j \geq 0 \quad j = 1, \ldots, n \end{aligned} \tag{5.1}$$

The problem above is easily cast as a linear program in standard form.

A specific case is that for $n = 3$, giving

$$\begin{array}{ll} \text{maximize} & 100x_1 + 10x_2 + x_3 \\ \text{subject to} & x_1 \leq 1 \\ & 20x_1 + x_2 \leq 100 \\ & 200x_1 + 20x_2 + x_3 \leq 10,000 \\ & x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0. \end{array}$$

In this case, we have three constraints and three variables (along with their nonnegativity constraints). After adding slack variables, the problem is in standard form. The system has $m = 3$ equations and $n = 6$ nonnegative variables. It can be verified that it takes $2^3 - 1 = 7$ pivot steps to solve the problem with the simplex method when at each step the pivot column is chosen to be the one with the largest (because this a maximization problem) reduced cost. (See Exercise 1.)

The general problem of the class (1) takes $2^n - 1$ pivot steps and this is in fact the number of vertices minus one (which is the starting vertex). To get an idea of how bad this can be, consider the case where $n = 50$. We have $2^{50} - 1 \approx 10^{15}$. In a year with 365 days, there are approximately 3×10^7 s. If a computer ran continuously, performing a million pivots of the simplex algorithm per second, it would take approximately

$$\frac{10^{15}}{3 \times 10^7 \times 10^6} \approx 33 \text{ years}$$

to solve a problem of this class using the greedy pivot selection rule.

Although it is not polynomial in the worst case, the simplex method remains one of major solvers for linear programming. In fact, the method has been recently proved to be (strongly) polynomial for solving the Markov Decision Process with any fixed discount rate.

*5.3 *The Ellipsoid Method

The basic ideas of the ellipsoid method stem from research done in the 1960s and 1970s mainly in the Soviet Union (as it was then called) by others who preceded Khachiyan. In essence, the idea is to enclose the region of interest in ever smaller ellipsoids.

The significant contribution of Khachiyan was to demonstrate in that under certain assumptions, the ellipsoid method constitutes a polynomially bounded algorithm for linear programming.

The version of the method discussed here is really aimed at finding a point of a polyhedral set Ω given by a system of linear inequalities.

$$\Omega = \{\mathbf{y} \in E^m : \mathbf{y}^T \mathbf{a}_j \leq c_j,\ j = 1, \ldots n\}$$

Finding a point of Ω can be thought of as equivalent to solving a linear programming problem.

Two important assumptions are made regarding this problem:

(A1) There is a vector $\mathbf{y}_0 \in E^m$ and a scalar $R > 0$ such that the closed ball $S(\mathbf{y}_0, R)$ with center $\mathbf{y}_0$ and radius R, that is

$$\{\mathbf{y} \in E^m : |\mathbf{y} - \mathbf{y}_0| \leq R\},$$

contains Ω.

(A2) If Ω is nonempty, there is a known scalar $r > 0$ such that Ω contains a ball of the form $S(\mathbf{y}^*, r)$ with center at $\mathbf{y}^*$ and radius r. (This assumption implies that if Ω is nonempty, then it has a nonempty interior and its volume is at least $\text{vol}(S(\mathbf{0}, r))$.)[2]

Definition. An *ellipsoid* in E^m is a set of the form

$$E = \{\mathbf{y} \in E^m : (\mathbf{y} - \mathbf{z})^T \mathbf{Q}(\mathbf{y} - \mathbf{z}) \leq 1\}$$

where $\mathbf{z} \in E^m$ is a given point (called the *center*) and $\mathbf{Q}$ is a positive definite matrix (see Sect. A.4 of Appendix A) of dimension $m \times m$. This ellipsoid is denoted $E(\mathbf{z}, \mathbf{Q})$.

The unit sphere $S(\mathbf{0}, 1)$ centered at the origin $\mathbf{0}$ is a special ellipsoid with $\mathbf{Q} = \mathbf{I}$, the identity matrix.

The axes of a general ellipsoid are the eigenvectors of $\mathbf{Q}$ and the lengths of the axes are $\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots, \lambda_m^{-1/2}$, where the λ_i's are the corresponding eigenvalues. It can be shown that the volume of an ellipsoid is

$$\text{vol}(E) = \text{vol}(S(\mathbf{0}, 1))\Pi_{i=1}^m \lambda_i^{-1/2} = \text{vol}(S(\mathbf{0}, 1)) \det(\mathbf{Q}^{-1/2}).$$

Cutting Plane and New Containing Ellipsoid

In the ellipsoid method, a series of ellipsoids E_k is defined, with centers $\mathbf{y}_k$ and with the defining $\mathbf{Q} = \mathbf{B}_k^{-1}$, where $\mathbf{B}_k$ is symmetric and positive definite.

At each iteration of the algorithm, we have $\Omega \subset E_k$. It is then possible to check whether $\mathbf{y}_k \in \Omega$. If so, we have found an element of Ω as required. If not, there is at least one constraint that is violated. Suppose $\mathbf{a}_j^T \mathbf{y}_k > c_j$. Then

$$\Omega \subset \frac{1}{2} E_k = \{\mathbf{y} \in E_k : \mathbf{a}_j^T \mathbf{y} \leq \mathbf{a}_j^T \mathbf{y}_k\}$$

This set is half of the ellipsoid, obtained by cutting the ellipsoid in half through its center (Fig. 5.1).

[2] The (topological) interior of any set Ω is the set of points in Ω which are the centers of some balls contained in Ω.

The successor ellipsoid E_{k+1} is defined to be the minimal-volume ellipsoid containing $(1/2)E_k$. It is constructed as follows. Define

$$\tau = \frac{1}{m+1},\ \delta = \frac{m^2}{m^2-1},\ \sigma = 2\tau.$$

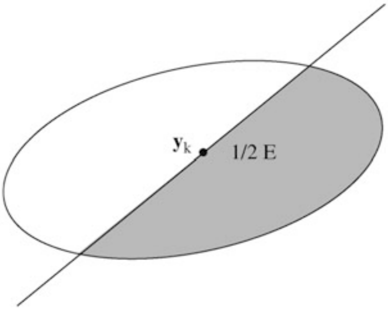

Fig. 5.1 A half-ellipsoid

Then put

$$\begin{aligned}\mathbf{y}_{k+1} &= \mathbf{y}_k - \frac{\tau}{(\mathbf{a}_j^T\mathbf{B}_k\mathbf{a}_j)^{1/2}}\mathbf{B}_k\mathbf{a}_j\\ \mathbf{B}_{k+1} &= \delta\left(\mathbf{B}_k - \sigma\frac{\mathbf{B}_k\mathbf{a}_j\mathbf{a}_j^T\mathbf{B}_k}{\mathbf{a}_j^T\mathbf{B}_k\mathbf{a}_j}\right)\end{aligned} \tag{5.2}$$

Theorem 1. *The ellipsoid* $E_{k+1} = E(\mathbf{y}_{k+1},\ \mathbf{B}_{k+1}^{-1})$ *defined as above is the ellipsoid of least volume containing* $(1/2)E_k$. *Moreover,*

$$\frac{\text{vol}(E_{k+1})}{\text{vol}(E_k)} = \left(\frac{m^2}{m^2-1}\right)^{(m-1)/2}\frac{m}{m+1} < \exp\left(-\frac{1}{2(m+1)}\right) < 1.$$

Proof. We shall not prove the statement about the new ellipsoid being of least volume, since that is not necessary for the results that follow. To prove the remainder of the statement, we have

$$\frac{\text{vol}(E_{k+1})}{\text{vol}(E_k)} = \frac{\det(\mathbf{B}_{k+1}^{1/2})}{\det(\mathbf{B}_k^{1/2})}$$

For simplicity, by a change of coordinates, we may take $\mathbf{B}_k = \mathbf{I}$. Then $\mathbf{B}_{k+1}$ has $m-1$ eigenvalues equal to $\delta = \frac{m^2}{m^2-1}$ and one eigenvalue equal to $\delta - 2\delta\tau = \frac{m^2}{m^2-1}(1-\frac{2}{m+1}) = (\frac{m}{m+1})^2$. The reduction in volume is the product of the square roots of these, giving the equality in the theorem.

Then using $(1+x)^p \leqslant e^{xp}$, we have

$$\left(\frac{m^2}{m^2-1}\right)^{(m-1)/2}\frac{m}{m+1} = \left(1+\frac{1}{m^2-1}\right)^{(m-1)/2}\left(1-\frac{1}{m+1}\right)$$
$$< \exp\left(\frac{1}{2(m+1)} - \frac{1}{(m+1)}\right) = \exp\left(-\frac{1}{2(m+1)}\right). \blacksquare$$

Convergence

The ellipsoid method is initiated by selecting $\mathbf{y}_0$ and R such that condition (A1) is satisfied. Then $\mathbf{B}_0 = R^2\mathbf{I}$, and the corresponding E_0 contains Ω. The updating of the E_k's is continued until a solution is found.

Under the assumptions stated above, a single repetition of the ellipsoid method reduces the volume of an ellipsoid to one-half of its initial value in $O(m)$ iterations. (See Appendix A for O notation.) Hence it can reduce the volume to less than that of a sphere of radius r in $O(m^2\log(R/r))$ iterations, since its volume is bounded from below by $\text{vol}(S(\mathbf{0},\ 1))r^m$ and the initial volume is $\text{vol}(S(\mathbf{0},\ 1))R^m$. Generally a single iteration requires $O(m^2)$ arithmetic operations. Hence the entire process requires $O(m^4\log(R/r))$ arithmetic operations.[3]

Ellipsoid Method for Usual Form of LP

Now consider the linear program (where $\mathbf{A}$ is $m \times n$)

$$\text{(P)} \qquad \begin{array}{ll} \text{maximize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \le \mathbf{b},\ \mathbf{x} \ge \mathbf{0} \end{array}$$

and its dual

$$\text{(D)} \qquad \begin{array}{ll} \text{minimize} & \mathbf{y}^T\mathbf{b} \\ \text{subject to} & \mathbf{y}^T\mathbf{A} \ge \mathbf{c}^T,\ \mathbf{y} \ge \mathbf{0}. \end{array}$$

Note that both problems can be solved by finding a feasible point to inequalities

$$\begin{aligned} -\mathbf{c}^T\mathbf{x} + \mathbf{b}^T\mathbf{y} &\le 0 \\ \mathbf{A}\mathbf{x} &\le \mathbf{b} \\ -\mathbf{A}^T\mathbf{y} &\le -\mathbf{c} \\ \mathbf{x},\ \mathbf{y} &\ge \mathbf{0}, \end{aligned} \tag{5.3}$$

where both $\mathbf{x}$ and $\mathbf{y}$ are variables. Thus, the total number of arithmetic operations for solving a linear program is bounded by $O((m+n)^4\log(R/r))$.

[3] Assumption (A2) is sometimes too strong. It has been shown, however, that when the data consists of integers, it is possible to perturb the problem so that (A2) is satisfied and if the perturbed problem has a feasible solution, so does the original Ω.

5.4 The Analytic Center

The new interior-point algorithms introduced by Karmarkar move by successive steps inside the feasible region. It is the interior of the feasible set rather than the vertices and edges that plays a dominant role in this type of algorithm. In fact, these algorithms purposely avoid the edges of the set, only eventually converging to one as a solution.

Our study of these algorithms begins in the next section, but it is useful at this point to introduce a concept that definitely focuses on the interior of a set, termed the set's analytic center. As the name implies, the center is away from the edge.

In addition, the study of the analytic center introduces a special structure, termed a *barrier* or *potential* that is fundamental to interior-point methods.

Consider a set $\mathcal{S}$ in a subset of $\mathcal{X}$ of E^n defined by a group of inequalities as

$$\mathcal{S} = \{\mathbf{x} \in \mathcal{X} \;:\; g_j(\mathbf{x}) \geqslant 0,\; j = 1, 2, \ldots, m\},$$

and assume that the functions g_j are continuous. $\mathcal{S}$ has a nonempty interior $\overset{\circ}{\mathcal{S}} = \{\mathbf{x} \in \mathcal{X} : g_j(\mathbf{x}) > 0, \text{ all } j\}$. Associated with this definition of the set is the *potential function*

$$\psi(\mathbf{x}) = -\sum_{j=1}^{m} \log g_j(\mathbf{x})$$

defined on $\overset{\circ}{\mathcal{S}}$.

The *analytic center* of $\mathcal{S}$ is the vector (or set of vectors) that minimizes the potential; that is, the vector (or vectors) that solve

$$\min \psi(\mathbf{x}) = \min \left\{ -\sum_{j=1}^{m} \log g_j(\mathbf{x}) : \mathbf{x} \in \mathcal{X},\; g_j(\mathbf{x}) > 0 \text{ for each } j \right\}.$$

Example 1 (A Cube). Consider the set $\mathcal{S}$ defined by $x_i \geqslant 0$, $(1 - x_i) \geqslant 0$, for $i = 1, 2, \ldots, n$. This is $\mathcal{S} = [0, 1]^n$, the unit cube in E^n. The analytic center can be found by differentiation to be $x_i = 1/2$, for all i. Hence, the analytic center is identical to what one would normally call the center of the unit cube.

In general, the analytic center depends on how the set is defined—on the particular inequalities used in the definition. For instance, the unit cube is also defined by the inequalities $x_i \geqslant 0$, $(1 - x_i)^d \geqslant 0$ with odd $d > 1$. In this case the solution is $x_i = 1/(d + 1)$ for all i. For large d this point is near the inner corner of the unit cube.

Also, the addition of redundant inequalities can change the location of the analytic center. For example, repeating a given inequality will change the center's location.

There are several sets associated with linear programs for which the analytic center is of particular interest. One such set is the feasible region itself. Another is the set of optimal solutions. There are also sets associated with dual and primal-dual formulations. All of these are related in important ways.

Let us illustrate by considering the analytic center associated with a bounded polytope Ω in E^m represented by $n(> m)$ linear inequalities; that is,

$$\Omega = \{\mathbf{y} \in E^m : \mathbf{c}^T - \mathbf{y}^T\mathbf{A} \geqslant \mathbf{0}\},$$

where $\mathbf{A} \in E^{m\times n}$ and $\mathbf{c} \in E^n$ are given and $\mathbf{A}$ has rank m. Denote the interior of Ω by

$$\overset{\circ}{\Omega} = \{\mathbf{y} \in E^m : \mathbf{c}^T - \mathbf{y}^T\mathbf{A} > \mathbf{0}\}.$$

The potential function for this set is

$$\psi_\Omega(\mathbf{y}) \equiv -\sum_{j=1}^{n} \log(c_j - \mathbf{y}^T\mathbf{a}_j) = -\sum_{j=1}^{n} \log s_j, \tag{5.4}$$

where $\mathbf{s} \equiv \mathbf{c} - \mathbf{A}^T\mathbf{y}$ is a *slack vector*. Hence the potential function is the negative sum of the logarithms of the slack variables.

The analytic center of Ω is the interior point of Ω that minimizes the potential function. This point is denoted by $\mathbf{y}^a$ and has the associated $\mathbf{s}^a = \mathbf{c} - \mathbf{A}^T\mathbf{y}^a$. The pair $(\mathbf{y}^a, \mathbf{s}^a)$ is uniquely defined, since the potential function is strictly convex (see Sect. 7.4) in the bounded convex set Ω.

Setting to zero the derivatives of $\psi(\mathbf{y})$ with respect to each y_i gives

$$\sum_{j=1}^{n} \frac{a_{ij}}{c_j - \mathbf{y}^T\mathbf{a}_j} = 0, \text{for all } i.$$

which can be written

$$\sum_{j=1}^{n} \frac{a_{ij}}{s_j} = 0, \text{ for all } i.$$

Now define $x_j = 1/s_j$ for each j. We introduce the notation

$$\mathbf{x} \circ \mathbf{s} \equiv (x_1s_1,\ x_2s_2,\ \ldots,\ x_ns_n)^T,$$

which is *component multiplication*. Then the analytic center is defined by the conditions

$$\begin{aligned} \mathbf{x} \circ \mathbf{s} &= \mathbf{1} \\ \mathbf{A}\mathbf{x} &= \mathbf{0} \\ \mathbf{A}^T\mathbf{y} + \mathbf{s} &= \mathbf{c}. \end{aligned}$$

The analytic center can be defined when the interior is empty or equalities are present, such as

$$\Omega = \{\mathbf{y} \in E^m : \mathbf{c}^T - \mathbf{y}^T\mathbf{A} \geqslant \mathbf{0},\ \mathbf{B}\mathbf{y} = \mathbf{b}\}.$$

In this case the analytic center is chosen on the linear surface $\{\mathbf{y} : \mathbf{B}\mathbf{y} = \mathbf{b}\}$ to maximize the product of the slack variables $\mathbf{s} = \mathbf{c} - \mathbf{A}^T\mathbf{y}$. Thus, in this context the interior of Ω refers to the interior of the positive orthant of slack variables: $R^n_+ \equiv \{\mathbf{s} : \mathbf{s} \geqslant \mathbf{0}\}$. This definition of interior depends only on the region of the slack variables. Even if there is only a single point in Ω with $\mathbf{s} = \mathbf{c} - \mathbf{A}^T\mathbf{y}$ for some $\mathbf{y}$ where $\mathbf{B}\mathbf{y} = \mathbf{b}$ with $\mathbf{s} > \mathbf{0}$, we still say that $\overset{\circ}{\Omega}$ is not empty.

5.5 The Central Path

The concept underlying interior-point methods for linear programming is to use nonlinear programming techniques of analysis and methodology. The analysis is often based on differentiation of the functions defining the problem. Traditional linear programming does not require these techniques since the defining functions are linear. Duality in general nonlinear programs is typically manifested through Lagrange multipliers (which are called dual variables in linear programming). The analysis and algorithms of the remaining sections of the chapter use these nonlinear techniques. These techniques are discussed systematically in later chapters, so rather than treat them in detail at this point, these current sections provide only minimal detail in their application to linear programming. It is expected that most readers are already familiar with the basic method for minimizing a function by setting its derivative to zero, and for incorporating constraints by introducing Lagrange multipliers. These methods are discussed in detail in Chaps. 11–15.

The computational algorithms of nonlinear programming are typically iterative in nature, often characterized as search algorithms. At any step with a given point, a direction for search is established and then a move in that direction is made to define the next point. There are many varieties of such search algorithms and they are systematically presented throughout the text. In this chapter, we use versions of Newton's method as the search algorithm, but we postpone a detailed study of the method until later chapters.

Not only have nonlinear methods improved linear programming, but interior-point methods for linear programming have been extended to provide new approaches to nonlinear programming. This chapter is intended to show how this merger of linear and nonlinear programming produces elegant and effective methods. These ideas take an especially pleasing form when applied to linear programming. Study of them here, even without all the detailed analysis, should provide good intuitive background for the more general manifestations.

Consider a primal linear program in standard form

$$\begin{aligned} \text{(LP) minimize} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geqslant \mathbf{0}. \end{aligned} \tag{5.5}$$

We denote the feasible region of this program by $\mathcal{F}_p$. We assume that $\overset{\circ}{\mathcal{F}}_p = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} > \mathbf{0}\}$ is nonempty and the optimal solution set of the problem is bounded.

Associated with this problem, we define for $\mu \geqslant 0$ the *barrier problem*

$$\begin{aligned}\text{(BP) minimize } & \mathbf{c}^T\mathbf{x} - \mu \sum_{j=1}^{n} \log x_j \\ \text{subject to } & \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} > \mathbf{0}.\end{aligned} \tag{5.6}$$

It is clear that $\mu = 0$ corresponds to the original problem (5.5). As $\mu \to \infty$, the solution approaches the analytic center of the feasible region (when it is bounded), since the barrier term swamps out $\mathbf{c}^T\mathbf{x}$ in the objective. As μ is varied continuously toward 0, there is a path $\mathbf{x}(\mu)$ defined by the solution to (BP). This path $\mathbf{x}(\mu)$ is termed the *primal central path*. As $\mu \to 0$ this path converges to the analytic center of the optimal face $\{\mathbf{x} : \mathbf{c}^T\mathbf{x} = z^*,\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geqslant \mathbf{0}\}$, where z^* is the optimal value of (LP).

A strategy for solving (LP) is to solve (BP) for smaller and smaller values of μ and thereby approach a solution to (LP). This is indeed the basic idea of interior-point methods.

At any $\mu > 0$, under the assumptions that we have made for problem (5.5), the necessary and sufficient conditions for a unique and bounded solution are obtained by introducing a *Lagrange multiplier* vector $\mathbf{y}$ for the linear equality constraints to form the *Lagrangian* (see Chap. 11)

$$\mathbf{c}^T\mathbf{x} - \mu \sum_{j=1}^{n} \log x_j - \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b}).$$

The derivatives with respect to the x_j's are set to zero, leading to the conditions

$$c_j - \mu/x_j - \mathbf{y}^T\mathbf{a}_j = 0, \text{ for each } j$$

or equivalently

$$\mu\mathbf{X}^{-1}\mathbf{1} + \mathbf{A}^T\mathbf{y} = \mathbf{c} \tag{5.7}$$

where as before $\mathbf{a}_j$ is the jth column of $\mathbf{A}$, $\mathbf{1}$ is the vector of 1's, and $\mathbf{X}$ is the diagonal matrix whose diagonal entries are the components of $\mathbf{x} > \mathbf{0}$. Setting $s_j = \mu/x_j$ the complete set of conditions can be rewritten

$$\begin{aligned}\mathbf{x} \circ \mathbf{s} &= \mu\mathbf{1} \\ \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{A}^T\mathbf{y} + \mathbf{s} &= \mathbf{c}.\end{aligned} \tag{5.8}$$

Note that $\mathbf{y}$ is a dual feasible solution and $\mathbf{c} - \mathbf{A}^T\mathbf{y} > \mathbf{0}$ (see Exercise 4).

Example 2 (A Square Primal). Consider the problem of maximizing x_1 within the unit square $\mathcal{S} = [0,\ 1]^2$. The problem is formulated as

$$\begin{aligned} \text{min} \quad & -x_1 \\ \text{s.t.} \quad & x_1 + x_3 = 1 \\ & \qquad x_2 + x_4 = 1 \\ & x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0,\ x_4 \geqslant 0. \end{aligned}$$

Here x_3 and x_4 are slack variables for the original problem to put it in standard form. The optimality conditions for $\mathbf{x}(\mu)$ consist of the original two linear constraint equations and the four equations

$$y_1 + s_1 = -1,\ y_2 + s_2 = 0,\ y_1 + s_3 = 0,\ y_2 + s_4 = 0$$

together with the relations $s_i = \mu/x_i$ for $i = 1, 2 \ldots, 4$. These equations are readily solved with a series of elementary variable eliminations to find

$$\begin{aligned} x_1(\mu) &= \frac{1 - 2\mu \pm \sqrt{1 + 4\mu^2}}{2} \\ x_2(\mu) &= 1/2. \end{aligned}$$

Using the "+" solution, it is seen that as $\mu \to 0$ the solution goes to $\mathbf{x} \to (1, 1/2)$. Note that this solution is not a corner of the cube. Instead it is at the analytic center of the optimal face $\{\mathbf{x} : x_1 = 1,\ 0 \leqslant x_2 \leqslant 1\}$. See Fig. 5.2. The limit of $\mathbf{x}(\mu)$ as $\mu \to \infty$ can be seen to be the point (1/2, 1/2). Hence, the central path in this case is a straight line progressing from the analytic center of the square (at $\mu \to \infty$) to the analytic center of the optimal face (at $\mu \to 0$).

Dual Central Path

Now consider the dual problem

$$\begin{aligned} \text{(LD) maximize} \quad & \mathbf{y}^T\mathbf{b} \\ \text{subject to} \quad & \mathbf{y}^T\mathbf{A} + \mathbf{s}^T = \mathbf{c}^T,\ \mathbf{s} \geqslant \mathbf{0}. \end{aligned}$$

We may apply the barrier approach to this problem by formulating the problem

$$\begin{aligned} \text{(BD) maximize} \quad & \mathbf{y}^T\mathbf{b} + \mu \sum_{j=1}^{n} \log s_j \\ \text{subject to} \quad & \mathbf{y}^T\mathbf{A} + \mathbf{s}^T = \mathbf{c}^T, \qquad \mathbf{s} > \mathbf{0}. \end{aligned}$$

We assume that the dual feasible set $\mathcal{F}_d$ has an interior $\overset{\circ}{\mathcal{F}}_d = \{(\mathbf{y}, \mathbf{s}) : \mathbf{y}^T\mathbf{A} + \mathbf{s}^T = \mathbf{c}^T,\ \mathbf{s} > \mathbf{0}\}$ is nonempty and the optimal solution set of (LD) is bounded. Then, as μ is varied continuously toward 0, there is a path $(\mathbf{y}(\mu),\ \mathbf{s}(\mu))$ defined by the solution to (BD). This path is termed the *dual central path*.

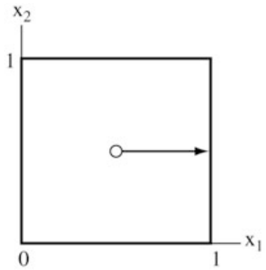

Fig. 5.2 The analytic path for the square

To work out the necessary and sufficient conditions we introduce $\mathbf{x}$ as a Lagrange multiplier and form the Lagrangian

$$\mathbf{y}^T\mathbf{b} + \mu \sum_{j=1}^{n} \log s_j - (\mathbf{y}^T\mathbf{A} + \mathbf{s}^T - \mathbf{c}^T)\mathbf{x}.$$

Setting to zero the derivative with respect to y_i leads to

$$b_i - \mathbf{a}^i\mathbf{x} = 0, \text{ for all } i$$

where $\mathbf{a}^i$ is the ith row of A. Setting to zero the derivative with respect to s_j leads to

$$\mu/s_j - x_j = 0, \text{ or } 1 - x_j s_j = 0, \text{ for all } j.$$

Combining these equations and including the original constraint yields the complete set of conditions which are identical to the optimality conditions for the primal central path (5.8). Note that $\mathbf{x}$ is indeed a primal feasible solution and $\mathbf{x} > \mathbf{0}$.

To see the geometric representation of the dual central path, consider the dual level set

$$\Omega(z) = \{\mathbf{y} : \mathbf{c}^T - \mathbf{y}^T\mathbf{A} \geqslant \mathbf{0},\ \mathbf{y}^T\mathbf{b} \geqslant z\}$$

for any $z < z^*$ where z^* is the optimal value of (LD). Then, the analytic center $(\mathbf{y}(z),\ \mathbf{s}(z))$ of $\Omega(z)$ coincides with the dual central path as z tends to the optimal value z^* from below. This is illustrated in Fig. 5.3, where the feasible region of the dual set (not the primal) is shown. The level sets $\Omega(z)$ are shown for various values of z. The analytic centers of these level sets correspond to the dual central path.

Example 3 (The Square Dual). Consider the dual of Example 2. This is

$$\begin{array}{ll} \text{max} & y_1 + y_2 \\ \text{subject to} & y_1 \leqslant -1 \\ & y_2 \leqslant 0. \end{array}$$

(The values of s_1 and s_2 are the slack variables of the inequalities.) The solution to the dual barrier problem is easily found from the solution of the primal barrier problem to be

$$y_1(\mu) = -1 - \mu/x_1(\mu),\ y_2 = -2\mu.$$

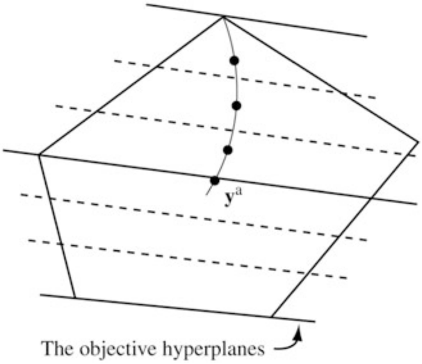

Fig. 5.3 The central path as analytic centers in the dual feasible region

As $\mu \to 0$, we have $y_1 \to -1,\ y_2 \to 0$, which is the unique solution to the dual LP. However, as $\mu \to \infty$, the vector y is unbounded, for in this case the dual feasible set is itself unbounded.

Primal–Dual Central Path

Suppose the feasible region of the primal (LP) has interior points and its optimal solution set is bounded. Then, the dual also has interior points (see Exercise 4). The primal–dual path is defined to be the set of vectors $(\mathbf{x}(\mu) > \mathbf{0},\ \mathbf{y}(\mu),\ \mathbf{s}(\mu) > \mathbf{0})$ that satisfy the conditions

$$\begin{aligned} \mathbf{x} \circ \mathbf{s} &= \mu\mathbf{1} \\ \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{A}^T\mathbf{y} + \mathbf{s} &= \mathbf{c} \end{aligned}$$

for $0 \leqslant \mu \leqslant \infty$. Hence the central path is defined without explicit reference to an optimization problem. It is simply defined in terms of the set of equality and inequality conditions.

Since conditions (5.8) and (5.9) are identical, the primal–dual central path can be split into two components by projecting onto the relevant space, as described in the following proposition.

Proposition 1. *Suppose the feasible sets of the primal and dual programs contain interior points. Then the primal–dual central path* $(\mathbf{x}(\mu),\ \mathbf{y}(\mu),\ \mathbf{s}(\mu))$ *exists for all* $\mu,\ 0 \leqslant \mu < \infty$. *Furthermore,* $\mathbf{x}(\mu)$ *is the primal central path, and* $(\mathbf{y}(\mu),\ \mathbf{s}(\mu))$ *is the dual central path. More-*

over, $\mathbf{x}(\mu)$ *and* $(\mathbf{y}(\mu),\ \mathbf{s}(\mu))$ *converge to the analytic centers of the optimal primal solution and dual solution faces, respectively, as* $\mu \to 0$*.*

Duality Gap

Let $(\mathbf{x}(\mu),\ \mathbf{y}(\mu),\ \mathbf{s}(\mu))$ be on the primal-dual central path. Then from (5.9) it follows that

$$\mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{b} = \mathbf{y}^T\mathbf{A}\mathbf{x} + \mathbf{s}^T\mathbf{x} - \mathbf{y}^T\mathbf{b} = \mathbf{s}^T\mathbf{x} = n\mu.$$

The value $\mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{b} = \mathbf{s}^T\mathbf{x}$ is the difference between the primal objective value and the dual objective value. This value is always nonnegative (see the weak duality lemma in Sect. 4.2) and is termed the *duality gap*. At any point on the primal–dual central path, the duality gap is equal to $n\mu$. It is clear that as $\mu \to 0$ the duality gap goes to zero, and hence both $\mathbf{x}(\mu)$ and $(\mathbf{y}(\mu),\ \mathbf{s}(\mu))$ approach optimality for the primal and dual, respectively.

The duality gap provides a measure of closeness to optimality. For any primal feasible $\mathbf{x}$, the value $\mathbf{c}^T\mathbf{x}$ gives an upper bound as $\mathbf{c}^T\mathbf{x} \geqslant z^*$ where z^* is the optimal value of the primal. Likewise, for any dual feasible pair $(\mathbf{y},\ \mathbf{s})$, the value $\mathbf{y}^T\mathbf{b}$ gives a lower bound as $\mathbf{y}^T\mathbf{b} \leqslant z^*$. The difference, the duality gap $g = \mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{b}$, provides a bound on z^* as $z^* \geqslant \mathbf{c}^T\mathbf{x} - g$. Hence if at a feasible point $\mathbf{x}$, a dual feasible $(\mathbf{y},\ \mathbf{s})$ is available, the quality of $\mathbf{x}$ can be measured as $\mathbf{c}^T\mathbf{x} - z^* \leqslant g$.

5.6 Solution Strategies

The various definitions of the central path directly suggest corresponding strategies for solution of a linear program. We outline three general approaches here: the primal barrier or path-following method, the primal-dual path-following method and the primal-dual potential-reduction method, although the details of their implementation and analysis must be deferred to later chapters after study of general nonlinear methods. Table 5.1 depicts these solution strategies and the simplex methods described in Chaps. 3 and 4 with respect to how they meet the three optimality conditions: Primal Feasibility, Dual Feasibility, and Zero-Duality during the iterative process.

Table 5.1 Properties of algorithms

	P-F	D-F	0-Duality
Primal simplex	√		√
Dual simplex		√	√
Primal barrier	√		
Primal-dual path-following	√	√	
Primal-dual potential-reduction	√	√	

For example, the primal simplex method keeps improving a primal feasible solution, maintains the zero-duality gap (complementarity slackness condition) and moves toward dual feasibility; while the dual simplex method keeps improving a dual feasible solution, maintains the zero-duality gap (complementarity condition) and moves toward primal feasibility (see Sect. 4.3). The primal barrier method keeps improving a primal feasible solution and moves toward dual feasibility and complementarity; and the primal-dual interior-point methods keep improving a primal and dual feasible solution pair and move toward complementarity.

Primal Barrier Method

A direct approach is to use the barrier construction and solve the the problem

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x} - \mu \sum\nolimits_{j=1}^{n} \log x_j \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} > \mathbf{0}, \end{aligned} \tag{5.9}$$

for a very small value of μ. In fact, if we desire to reduce the duality gap to ε it is only necessary to solve the problem for $\mu = \varepsilon/n$. Unfortunately, when μ is small, the problem (5.9) could be highly ill-conditioned in the sense that the necessary conditions are nearly singular. This makes it difficult to directly solve the problem for small μ.

An overall strategy, therefore, is to start with a moderately large μ (say $\mu = 100$) and solve that problem approximately. The corresponding solution is a point approximately on the primal central path, but it is likely to be quite distant from the point corresponding to the limit of $\mu \to 0$. However this solution point at $\mu = 100$ can be used as the starting point for the problem with a slightly smaller μ, for this point is likely to be close to the solution of the new problem. The value of μ might be reduced at each stage by a specific factor, giving $\mu_{k+1} = \gamma\mu_k$, where γ is a fixed positive parameter less than one and k is the stage count.

If the strategy is begun with a value μ_0, then at the kth stage we have $\mu_k = \gamma^k\mu_0$. Hence to reduce μ_k/μ_0 to below ε, requires

$$k = \frac{\log \varepsilon}{\log \gamma}$$

stages.

Often a version of Newton's method for minimization is used to solve each of the problems. For the current strategy, Newton's method works on problem (5.9) with fixed μ by considering the central path equations (5.8)

$$\begin{aligned} \mathbf{x} \circ \mathbf{s} &= \mu\mathbf{1} \\ \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{A}^T\mathbf{y} + \mathbf{s} &= \mathbf{c}. \end{aligned} \tag{5.10}$$

From a given point $\mathbf{x} \in \mathring{\mathcal{F}}_p$, Newton's method moves to a closer point $\mathbf{x}^+ \in \mathring{\mathcal{F}}_p$ by moving in the directions $\mathbf{d_x}$, $\mathbf{d_y}$ and $\mathbf{d_s}$ determined from the linearized version of (5.10)

$$\begin{aligned} \mu\mathbf{X}^{-2}\mathbf{d_x} + \mathbf{d_s} &= \mu\mathbf{X}^{-1}1 - \mathbf{c}, \\ \mathbf{A}\mathbf{d_x} &= \mathbf{0}, \\ -\mathbf{A}^T\mathbf{d_y} - \mathbf{d_s} &= \mathbf{0}. \end{aligned} \tag{5.11}$$

(Recall that $\mathbf{X}$ is the diagonal matrix whose diagonal entries are components of $\mathbf{x} > \mathbf{0}$.) The new point is then updated by taking a step in the direction of $\mathbf{d}_x$, as $\mathbf{x}^+ = \mathbf{x} + \mathbf{d}_x$.

Notice that if $\mathbf{x} \circ \mathbf{s} = \mu\mathbf{1}$ for some $\mathbf{s} = \mathbf{c} - \mathbf{A}^T\mathbf{y}$, then $\mathbf{d} \equiv (\mathbf{d_x}, \mathbf{d_y}, \mathbf{d_s}) = \mathbf{0}$ because the current point satisfies $\mathbf{Ax} = \mathbf{b}$ and hence is already the central path solution for μ. If some component of $\mathbf{x} \circ \mathbf{s}$ is less than μ, then $\mathbf{d}$ will tend to increment the solution so as to increase that component. The converse will occur for components of $\mathbf{x} \circ \mathbf{s}$ greater than μ.

This process may be repeated several times until a point close enough to the proper solution to the barrier problem for the given value of μ is obtained. That is, until the necessary and sufficient conditions (5.7) are (approximately) satisfied.

There are several details involved in a complete implementation and analysis of Newton's method. These items are discussed in later chapters of the text. However, the method works well if either μ is moderately large, or if the algorithm is initiated at a point very close to the solution, exactly as needed for the barrier strategy discussed in this subsection.

To solve (5.11), premultiplying both sides by $\mathbf{X}^2$ we have

$$\mu\mathbf{d_x} + \mathbf{X}^2\mathbf{d_s} = \mu\mathbf{X}1 \; - \mathbf{X}^2\mathbf{c}.$$

Then, premultiplying by $\mathbf{A}$ and using $\mathbf{Ad_x} = \mathbf{0}$, we have

$$\mathbf{AX}^2\mathbf{d_s} = \mu\mathbf{AX1} - \mathbf{AX}^2\mathbf{c}.$$

Using $\mathbf{d_s} = -\mathbf{A}^T\mathbf{d_y}$ we have

$$(\mathbf{AX}^2\mathbf{A}^T)\mathbf{d_y} = -\mu\mathbf{AX1} + \mathbf{AX}^2\mathbf{c}.$$

Thus, $\mathbf{d_y}$ can be computed by solving the above linear system of equations. Then $\mathbf{d_s}$ can be found from the third equation in (5.11) and finally $\mathbf{d_x}$ can be found from the first equation in (5.11), together this amounts to $O(nm^2 + m^3)$ arithmetic operations for each Newton step.

Primal-Dual Path-Following

Another strategy for solving a linear program is to follow the central path from a given initial primal-dual solution pair. Consider a linear program in standard form

$$\begin{array}{llcll} & \text{Primal} & & & \text{Dual} \\ \text{minimize} & \mathbf{c}^T\mathbf{x} & & \text{maximize} & \mathbf{y}^T\mathbf{b} \\ \text{subject to} & \mathbf{A}\mathbf{x}=\mathbf{b},\ \mathbf{x}\geqslant\mathbf{0} & & \text{subject to} & \mathbf{y}^T\mathbf{A}\leqslant\mathbf{c}^T. \end{array}$$

Assume that the interior of both primal and dual feasible regions $\overset{\circ}{\mathcal{F}} \neq \emptyset$; that is, both[4]

$$\overset{\circ}{\mathcal{F}}_p = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} > \mathbf{0}\} \neq \emptyset \quad \text{and} \quad \overset{\circ}{\mathcal{F}}_d = \{(\mathbf{y},\ \mathbf{s}) : \mathbf{s} = \mathbf{c} - \mathbf{A}^T\mathbf{y} > \mathbf{0}\} \neq \emptyset;$$

and denote by z^* the optimal objective value.

The central path can be expressed as

$$\mathcal{C} = \left\{(\mathbf{x},\ \mathbf{y},\ \mathbf{s}) \in \overset{\circ}{\mathcal{F}} : \mathbf{x} \circ \mathbf{s} = \frac{\mathbf{x}^T\mathbf{s}}{n}\mathbf{1}\right\}$$

in the primal-dual form. On the path we have $\mathbf{x} \circ \mathbf{s} = \mu\mathbf{1}$ and hence $\mathbf{s}^T\mathbf{x} = n\mu$. A *neighborhood* of the central path $\mathcal{C}$ is of the form

$$\mathcal{N}(\eta) = \{(\mathbf{x}, \mathbf{y},\ \mathbf{s}) \in \overset{\circ}{\mathcal{F}} : |\mathbf{s} \circ \mathbf{x} - \mu\mathbf{1}| < \eta\mu,\ \text{where } \mu = \mathbf{s}^T\mathbf{x}/n\} \tag{5.12}$$

for some $\eta \in (0,\ 1)$, say $\eta = 1/4$. This can be thought of as a tube whose center is the central path.

The idea of the path-following method is to move within a tubular neighborhood of the central path toward the solution point. A suitable initial point $(\mathbf{x}^0,\ \mathbf{y}^0,\ \mathbf{s}^0) \in \mathcal{N}(\eta)$ can be found by solving the barrier problem for some fixed μ_0 or from an initialization phase proposed later. After that, step by step moves are made, alternating between a predictor step and a corrector step. After each pair of steps, the point achieved is again in the fixed given neighborhood of the central path, but closer to the linear program's solution set.

The predictor step is designed to move essentially parallel to the true central path. The step $\mathbf{d} \equiv (\mathbf{d}_\mathbf{x},\ \mathbf{d}_\mathbf{y},\ \mathbf{d}_\mathbf{s})$ is determined from the linearized version of the primal–dual central path equations of (5.9), as

$$\begin{aligned} \mathbf{s} \circ \mathbf{d}_\mathbf{x} + \mathbf{x} \circ \mathbf{d}_\mathbf{s} &= \gamma\mu\mathbf{1} - \mathbf{x} \circ \mathbf{s}, \\ \mathbf{A}\mathbf{d}_\mathbf{x} &= \mathbf{0}, \\ -\mathbf{A}^T\mathbf{d}_\mathbf{y} - \mathbf{d}_\mathbf{s} &= \mathbf{0}, \end{aligned} \tag{5.13}$$

where here one selects $\gamma = 0$. (To show the dependence of $\mathbf{d}$ on the current pair $(\mathbf{x},\ \mathbf{s})$ and the parameter γ, we write $\mathbf{d} = \mathbf{d}(\mathbf{x},\ \mathbf{s},\ \gamma)$.)

[4] The symbol $\emptyset$ denotes the empty set.

The new point is then found by taking a step in the direction of $\mathbf{d}$, as $(\mathbf{x}^+, \mathbf{y}^+, \mathbf{s}^+) = (\mathbf{x}, \mathbf{y}, \mathbf{s}) + \alpha(\mathbf{d}_\mathbf{x}, \mathbf{d}_\mathbf{y}, \mathbf{d}_\mathbf{s})$, where α is the step-size. Note that $\mathbf{d}_\mathbf{x}^T\mathbf{d}_\mathbf{s} = -\mathbf{d}_\mathbf{x}^T\mathbf{A}^T\mathbf{d}_\mathbf{y} = 0$ here. Then

$$(\mathbf{x}^+)^T\mathbf{s}^+ = (\mathbf{x} + \alpha\mathbf{d}_\mathbf{x})^T(\mathbf{s} + \alpha\mathbf{d}_\mathbf{s}) = \mathbf{x}^T\mathbf{s} + \alpha(\mathbf{d}_\mathbf{x}^T\mathbf{s} + \mathbf{x}^T\mathbf{d}_\mathbf{s}) = (1 - \alpha)\mathbf{x}^T\mathbf{s},$$

where the last step follows by multiplying the first equation in (5.13) by $\mathbf{1}^T$. Thus, the predictor step reduces the duality gap by a factor $1 - \alpha$. The maximum possible step-size α in that direction is made in that parallel direction without going outside of the neighborhood $\mathcal{N}(2\eta)$.

The corrector step essentially moves perpendicular to the central path in order to get closer to it. This step moves the solution back to within the neighborhood $\mathcal{N}(\eta)$, and the step is determined by selecting $\gamma = 1$ in (5.13) with $\mu = \mathbf{x}^T\mathbf{s}/n$. Notice that if $\mathbf{x} \circ \mathbf{s} = \mu\mathbf{1}$, then $\mathbf{d} = \mathbf{0}$ because the current point is already a central path solution.

This corrector step is identical to one step of the barrier method. Note, however, that the predictor–corrector method requires only one sequence of steps, each consisting of a single predictor and corrector. This contrasts with the barrier method which requires a complete sequence for each μ to get back to the central path, and then an outer sequence to reduce the μ's.

One can prove that for any $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{N}(\eta)$ with $\mu = \mathbf{x}^T\mathbf{s}/n$, the step-size in the predictor stop satisfies

$$\alpha \geqslant \frac{1}{2\sqrt{n}}.$$

Thus, the iteration complexity of the method is $O(\sqrt{n})\log(1/\varepsilon))$ to achieve $\mu/\mu_0 \leqslant \varepsilon$ where $n\mu_0$ is the initial duality gap. Moreover, one can prove that the step-size $\alpha \to 1$ as $\mathbf{x}^T\mathbf{s} \to 0$, that is, the duality reduction speed is accelerated as the gap becomes smaller.

Primal-Dual Potential Reduction Algorithm

In this method a *primal-dual potential function* is used to measure the solution's progress. The potential is reduced at each iteration. There is no restriction on either neighborhood or step-size during the iterative process as long as the potential is reduced. The greater the reduction of the potential function, the faster the convergence of the algorithm. Thus, from a practical point of view, potential-reduction algorithms may have an advantage over *path-following algorithms* where iterates are confined to lie in certain neighborhoods of the central path.

For $\mathbf{x} \in \mathring{\mathcal{F}}_p$ and $(\mathbf{y}, \mathbf{s}) \in \mathring{\mathcal{F}}_d$ the primal-dual potential function is defined by

$$\psi_{n+\rho}(\mathbf{x}, \mathbf{s}) \equiv (n + \rho)\log(\mathbf{x}^T\mathbf{s}) - \sum_{j=1}^{n}\log(x_js_j), \tag{5.14}$$

where $\rho \geqslant 0$.

From the arithmetic and geometric mean inequality (also see Exercise 10) we can derive that

$$n \log(\mathbf{x}^T\mathbf{s}) - \sum_{j=1}^{n} \log(x_j s_j) \geqslant n \log n.$$

Then

$$\psi_{n+\rho}(\mathbf{x},\ \mathbf{s}) = \rho \log(\mathbf{x}^T\mathbf{s}) + n \log(\mathbf{x}^T\mathbf{s}) - \sum_{j=1}^{n} \log(x_j s_j) \geqslant \rho \log(\mathbf{x}^T\mathbf{s}) + n \log n. \quad (5.15)$$

Thus, for $\rho > 0$, $\psi_{n+\rho}(\mathbf{x},\ \mathbf{s}) \to -\infty$ implies that $\mathbf{x}^T\mathbf{s} \to 0$. More precisely, we have from (5.15)

$$\mathbf{x}^T\mathbf{s} \leqslant \exp\left(\frac{\psi_{n+\rho}(\mathbf{x}, \mathbf{s}) - n \log n}{\rho}\right).$$

Hence the primal–dual potential function gives an explicit bound on the magnitude of the duality gap.

The objective of this method is to drive the potential function down toward minus infinity. The method of reduction is a version of Newton's method (5.13). In this case we select $\gamma = n/(n + \rho)$ in (5.13). Notice that is a combination of a predictor and corrector choice. The predictor uses $\gamma = 0$ and the corrector uses $\gamma = 1$. The primal–dual potential method uses something in between. This seems logical, for the predictor moves parallel to the central path toward a lower duality gap, and the corrector moves perpendicular to get close to the central path. This new method does both at once. Of course, this intuitive notion must be made precise.

For $\rho \geqslant \sqrt{n}$, there is in fact a guaranteed decrease in the potential function by a fixed amount δ (see Exercises 12 and 13). Specifically,

$$\psi_{n+\rho}(\mathbf{x}^+,\ \mathbf{s}^+) - \psi_{n+\rho}(\mathbf{x},\ \mathbf{s}) \leqslant -\delta \quad (5.16)$$

for a constant $\delta \geqslant 0.2$. This result provides a theoretical bound on the number of required iterations and the bound is competitive with other methods. However, a faster algorithm may be achieved by conducting a line search along direction **d** to achieve the greatest reduction in the primal-dual potential function at each iteration.

We outline the algorithm here:

Step 1. Start at a point $(\mathbf{x}_0,\ \mathbf{y}_0,\ \mathbf{s}_0) \in \mathring{\mathcal{F}}$ with $\psi_{n+\rho}(\mathbf{x}_0,\ \mathbf{s}_0) \le \rho \log((\mathbf{s}_0)^T\mathbf{x}_0) + n \log n + O(\sqrt{n} \log n)$ which is determined by an initiation procedure, as discussed in Sect. 5.7. Set $\rho \ge \sqrt{n}$. Set $k = 0$ and $\gamma = n/(n + \rho)$. Select an accuracy parameter $\varepsilon > 0$.

Step 2. Set $(\mathbf{x},\ \mathbf{s}) = (\mathbf{x}_k,\ \mathbf{s}_k)$ and compute $(\mathbf{d}_\mathbf{x},\ \mathbf{d}_\mathbf{y},\ \mathbf{d}_\mathbf{s})$ from (5.13).

Step 3. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \bar{\alpha}\mathbf{d}_\mathbf{x}$, $\mathbf{y}_{k+1} = \mathbf{y}_k + \bar{\alpha}\mathbf{d}_\mathbf{y}$, and $\mathbf{s}_{k+1} = \mathbf{s}_k + \bar{\alpha}\mathbf{d}_\mathbf{s}$ where

$$\bar{\alpha} = \arg\min_{\alpha \geq 0} \psi_{n+\rho}(\mathbf{x}_k + \alpha\mathbf{d}_\mathbf{x},\ \mathbf{s}_k + \alpha\mathbf{d}_\mathbf{s}).$$

Step 4. Let $k = k + 1$. If $\frac{\mathbf{s}_k^T\mathbf{x}_k}{\mathbf{s}_0^T\mathbf{x}_0} \le \varepsilon$, Stop. Otherwise return to Step 2.

Theorem 2. *The algorithm above terminates in at most $O(\rho\log(n/\varepsilon))$ iterations with*

$$\frac{(\mathbf{s}_k)^T\mathbf{x}_k}{(\mathbf{s}_0)^T\mathbf{x}_0} \le \varepsilon.$$

Proof. Note that after k iterations, we have from (5.16)

$$\psi_{n+\rho}(\mathbf{x}_k,\ \mathbf{s}_k) \le \psi_{n+\rho}(\mathbf{x}_0,\ \mathbf{s}_0) - k\cdot\delta \le \rho\log((\mathbf{s}_0)^T\mathbf{x}_0) + n\log n + O(\sqrt{n}\log n) - k\cdot\delta.$$

Thus, from the inequality (5.15),

$$\rho\log(\mathbf{s}_k^T\mathbf{x}_k) + n\log n \le \rho\log(\mathbf{s}_0^T\mathbf{x}_0) + n\log n + O(\sqrt{n}\log n) - k\cdot\delta,$$

or

$$\rho(\log(\mathbf{s}_k^T\mathbf{x}_k) - \log(\mathbf{s}_0^T\mathbf{x}_0)) \le -k\cdot\delta + O(\sqrt{n}\log n).$$

Therefore, as soon as $k \ge O(\rho\log(n/\varepsilon))$, we must have

$$\rho(\log(\mathbf{s}_k^T\mathbf{x}_k) - \log(\mathbf{s}_0^T\mathbf{x}_0)) \le -\rho\log(1/\varepsilon),$$

or

$$\frac{\mathbf{s}_k^T\mathbf{x}_k}{\mathbf{s}_0^T\mathbf{x}_0} \le \varepsilon. \blacksquare$$

Theorem 2 holds for any $\rho \ge \sqrt{n}$. Thus, by choosing $\rho = \sqrt{n}$, the iteration complexity bound becomes $O(\sqrt{n}\log(n/\varepsilon))$.

Iteration Complexity

The computation of each iteration basically requires solving (5.13) for **d**. Note that the first equation of (5.13) can be written as

$$\mathbf{S}\mathbf{d}_\mathbf{x} + \mathbf{X}\mathbf{d}_\mathbf{s} = \gamma\mu\mathbf{1} - \mathbf{X}\mathbf{S}\mathbf{1}$$

where **X** and **S** are two diagonal matrices whose diagonal entries are components of $\mathbf{x} > \mathbf{0}$ and $\mathbf{s} > \mathbf{0}$, respectively. Premultiplying both sides by $\mathbf{S}^{-1}$ we have

$$\mathbf{d}_\mathbf{x} + \mathbf{S}^{-1}\mathbf{X}\mathbf{d}_\mathbf{s} = \gamma\mu\mathbf{S}^{-1}\mathbf{1} - \mathbf{x}.$$

Then, premultiplying by **A** and using $\mathbf{A}\mathbf{d}_\mathbf{x} = \mathbf{0}$, we have

$$\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{d}_\mathbf{s} = \gamma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1} - \mathbf{A}\mathbf{x} = \gamma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1} - \mathbf{b}.$$

Using $\mathbf{d}_\mathbf{s} = -\mathbf{A}^T\mathbf{d}_\mathbf{y}$ we have

$$(\mathbf{A}\mathbf{S}^{-1}\mathbf{X}\mathbf{A}^T)\mathbf{d}_\mathbf{y} = \mathbf{b} - \gamma\mu\mathbf{A}\mathbf{S}^{-1}\mathbf{1}.$$

Thus, the primary computational cost of each iteration of the interior-point algorithm discussed in this section is to form and invert the normal matrix $\mathbf{AXS}^{-1}\mathbf{A}^T$, which typically requires $O(nm^2 + m^3)$ arithmetic operations. However, an approximation of this matrix can be updated and inverted using far fewer arithmetic operations. In fact, using a rank-one technique (see Chap. 10) to update the approximate inverse of the normal matrix during the iterative progress, one can reduce the average number of arithmetic operations per iteration to $O(\sqrt{n}m^2)$. Thus, if the relative tolerance ε is viewed as a variable, we have the following total arithmetic operation complexity bound to solve a linear program:

Corollary. *Let $\rho = \sqrt{n}$. Then, the algorithm above Theorem 2 terminates in at most $O(nm^2 \log(n/\varepsilon))$ arithmetic operations.*

5.7 Termination and Initialization

There are several remaining important issues concerning interior-point algorithms for linear programs. The first issue involves termination. Unlike the simplex method which terminates with an exact solution, interior-point algorithms are continuous optimization algorithms that generate an infinite solution sequence converging to an optimal solution. If the data of a particular problem are integral or rational, an argument is made that, after the worst-case time bound, an exact solution can be rounded from the latest approximate solution. Several questions arise. First, under the real number computation model (that is, the data consists of real numbers), how can we terminate at an exact solution? Second, regardless of the data's status, is there a practical test, which can be computed cost-effectively during the iterative process, to identify an exact solution so that the algorithm can be terminated before the worse-case time bound? Here, by exact solution we mean one that could be found using exact arithmetic, such as the solution of a system of linear equations, which can be computed in a number of arithmetic operations bounded by a polynomial in n.

The second issue involves initialization. Almost all interior-point algorithms require the regularity assumption that $\overset{\circ}{\mathcal{F}} \neq \emptyset$. What is to be done if this is not true? A related issue is that interior-point algorithms have to start at a strictly feasible point near the central path.

*Termination

Complexity bounds for interior-point algorithms generally depend on an ε which must be zero in order to obtain an exact optimal solution. Sometimes it is advantageous to employ an early termination or rounding method while ε is still moderately large. There are five basic approaches.

- A "purification" procedure finds a feasible corner whose objective value is at least as good as the current interior point. This can be accomplished in strongly polynomial time (that is, the complexity bound is a polynomial only in the dimensions m and n). One difficulty is that there may be many non-optimal vertices close to the optimal face, and the procedure might require many pivot steps for difficult problems.
- A second method seeks to identify an optimal basis. It has been shown that if the linear program is nondegenerate, the unique optimal basis may be identified early. The procedure seems to work well for some problems but it has difficulty if the problem is degenerate. Unfortunately, most real linear programs are degenerate.
- The third approach is to slightly perturb the data such that the new program is nondegenerate and its optimal basis remains one of the optimal bases of the original program. There are questions about how and when to perturb the data during the iterative process, decisions which can significantly affect the success of the effort.
- The fourth approach is to guess the optimal face and find a feasible solution on that face. It consists of two phases: the first phase uses interior point algorithms to identify the complementarity partition (P^*, Z^*) (see Exercise 6), and the second phase adapts the simplex method to find an optimal primal (or dual) basic solution and one can use (P^*, Z^*) as a starting base for the second phase. This method is often called the cross-over method. It is guaranteed to work in finite time and is implemented in several popular linear programming software packages.
- The fifth approach is to guess the optimal face and project the current interior point onto the interior of the optimal face. See Fig. 5.4. The termination criterion is guaranteed to work in finite time.

The fourth and fifth methods above are based on the fact that (as observed in practice and subsequently proved) many interior-point algorithms for linear programming generate solution sequences that converge to a strictly complementary solution or an interior solution on the optimal face; see Exercise 8.

Initialization

Most interior-point algorithms must be initiated at a strictly feasible point. The complexity of obtaining such an initial point is the same as that of solving the linear program itself. More importantly, a complete algorithm should accomplish two tasks: (1) detect the infeasibility or unboundedness status of the problem, then (2) generate an optimal solution if the problem is neither infeasible nor unbounded.

Several approaches have been proposed to accomplish these goals:

- The primal and dual can be combined into a single linear feasibility problem, and a feasible point found. Theoretically, this approach achieves the currently best iteration complexity bound, that is, $O(\sqrt{n}\log(1/\varepsilon))$. Practically, a significant disadvantage of this approach is the doubled dimension of the system of equations that must be solved at each iteration.

- The big-M method can be used by adding one or more artificial column(s) and/or row(s) and a huge penalty parameter M to force solutions to become feasible during the algorithm. A major disadvantage of this approach is the numerical problems caused by the addition of coefficients of large magnitude.
- Phase I-then-Phase II methods are effective. A major disadvantage of this approach is that the two (or three) related linear programs must be solved sequentially.

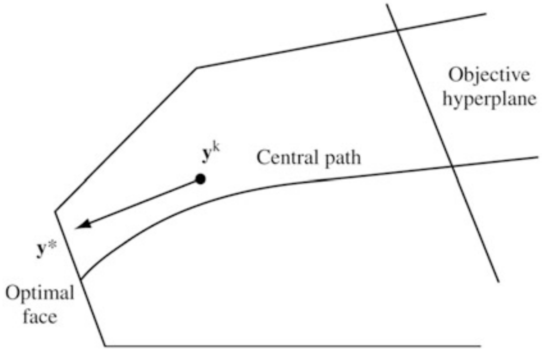

Fig. 5.4 Illustration of the projection of an interior point onto the optimal face

- A modified Phase I-Phase II method approaches feasibility and optimality simultaneously. To our knowledge, the currently best iteration complexity bound of this approach is $O(n\log(1/\varepsilon))$, as compared to $O(\sqrt{n}\log(1/\varepsilon))$ of the three above. Other disadvantages of the method include the assumption of non-empty interior and the need of an objective lower bound.

The HSD Algorithm

There is an algorithm, termed the *Homogeneous Self-Dual Algorithm* that overcomes the difficulties mentioned above. The algorithm achieves the theoretically best $O(\sqrt{n}\log(1/\varepsilon))$ complexity bound and is often used in linear programming software packages.

The algorithm is based on the construction of a homogeneous and self-dual linear program related to (LP) and (LD) (see Sect. 5.5). We now briefly explain the two major concepts, homogeneity and self-duality, used in the construction.

In general, a system of linear equations of inequalities is *homogeneous* if the right hand side components are all zero. Then if a solution is found, any positive multiple of that solution is also a solution. In the construction used below, we allow a single inhomogeneous constraint, often called a *normalizing constraint*. Karmarkar's original canonical form is a homogeneous linear program.

A linear program is termed *self-dual* if the dual of the problem is equivalent to the primal. The advantage of self-duality is that we can apply a primal-dual interior-point algorithm to solve the self-dual problem *without* doubling the dimension of the linear system solved at each iteration.

The homogeneous and self-dual linear program (HSDP) is constructed from (LP) and (LD) in such a way that the point $\mathbf{x} = \mathbf{1},\ \mathbf{y} = \mathbf{0},\ \tau = 1,\ z = 1,\ \theta = 1$ is feasible. The primal program is

$$
\begin{array}{llllll}
(HSDP) & \text{minimize} & & & & (n+1)\theta \\
& \text{subject to} & & \mathbf{A}\mathbf{x} & -\mathbf{b}\tau & +\bar{\mathbf{b}}\theta = \mathbf{0}, \\
& & -\mathbf{A}^T\mathbf{y} & & +\mathbf{c}\tau & -\bar{\mathbf{c}}\theta \geq \mathbf{0}, \\
& & \mathbf{b}^T\mathbf{y} & -\mathbf{c}^T\mathbf{x} & & +\bar{z}\theta \geq 0, \\
& & -\bar{\mathbf{b}}^T\mathbf{y} & +\bar{\mathbf{c}}^T\mathbf{x} & -\bar{z}\tau & \quad = -(n+1), \\
& & \multicolumn{4}{l}{\mathbf{y} \text{ free},\ \mathbf{x} \geq \mathbf{0},\ \tau \geq 0,\quad \theta \text{ free};}
\end{array}
$$

where

$$
\bar{\mathbf{b}} = \mathbf{b} - \mathbf{Al},\ \bar{\mathbf{c}} = \mathbf{c} - \mathbf{1},\ \bar{z} = \mathbf{c}^T\mathbf{1} + 1. \tag{5.17}
$$

Notice that $\bar{\mathbf{b}}, \bar{\mathbf{c}}$, and $\bar{\mathbf{z}}$ represent the "infeasibility" of the initial primal point, dual point, and primal-dual "gap," respectively. They are chosen so that the system is feasible. For example, for the point $\mathbf{x} = \mathbf{1},\ \mathbf{y} = \mathbf{0},\ \tau = 1,\ \theta = 1$, the last equation becomes

$$
0 + \mathbf{c}^T\mathbf{x} - \mathbf{1}^T\mathbf{x} - (\mathbf{c}^T\mathbf{x} + 1) = -n - 1.
$$

Note also that the top two constraints in (HSDP), with $\tau = 1$ and $\theta = 0$, represent primal and dual feasibility (with $\mathbf{x} \geq \mathbf{0}$). The third equation represents reversed weak duality (with $\mathbf{b}^T\mathbf{y} \geq \mathbf{c}^T\mathbf{x}$) rather than the reverse. So if these three equations are satisfied with $\tau = 1$ and $\theta = 0$ they define primal and dual optimal solutions. Then, to achieve primal and dual feasibility for $\mathbf{x} = \mathbf{1},\ (\mathbf{y},\ \mathbf{s}) = (\mathbf{0},\ \mathbf{1})$, we add the artificial variable θ. The fourth constraint is added to achieve self-duality.

The problem is self-dual because its overall coefficient matrix has the property that its transpose is equal to its negative. It is *skew-symmetric*.

Denote by $\mathbf{s}$ the slack vector for the second constraint and by κ the slack scalar for the third constraint. Denote by $\mathcal{F}_h$ the set of all points $(\mathbf{y},\ \mathbf{x},\ \tau,\ \theta,\ \mathbf{s},\ \kappa)$ that are feasible for (HSDP). Denote by $\mathcal{F}_h^0$ the set of strictly feasible points with $(\mathbf{x},\ \tau, \mathbf{s},\ \kappa) > \mathbf{0}$ in $\mathcal{F}_h$. By combining the constraints (Exercise 14) we can write the last (equality) constraint as

$$
\mathbf{1}^T\mathbf{x} + \mathbf{1}^T\mathbf{s} + \tau + \kappa - (n+1)\theta = (n+1), \tag{5.18}
$$

which serves as a normalizing constraint for (HSDP). This implies that for $0 \leq \theta \leq 1$ the variables in this equation are bounded.

We state without proof the following basic result.

Theorem 1. *Consider problems (HSDP).*

(i) (HSDP) has an optimal solution and its optimal solution set is bounded.

(ii) The optimal value of (HSDP) is zero, and

$$(\mathbf{y},\ \mathbf{x},\ \tau,\ \theta,\ \mathbf{s},\ \kappa) \in \mathcal{F}_h \textit{ implies that } (n+1)\theta = \mathbf{x}^T\mathbf{s} + \tau\kappa.$$

(iii) There is an optimal solution $(\mathbf{y}^*,\ \mathbf{x}^*,\ \tau^*,\ \theta^* = 0,\ \mathbf{s}^*,\ \kappa^*) \in \mathcal{F}_h$ *such that*

$$\begin{pmatrix} \mathbf{x}^* + \mathbf{s}^* \\ \tau^* + \kappa^* \end{pmatrix} > \mathbf{0},$$

which we call a strictly self-complementary solution.

Part (ii) of the theorem shows that as θ goes to zero, the solution tends toward satisfying complementary slackness between **x** and **s** and between τ and κ. Part (iii) shows that at a solution with $\theta = 0$, the complementary slackness is strict in the sense that at least one member of a complementary pair must be positive. For example, $x_1 s_1 = 0$ is required by complementary slackness, but in this case $x_1 = 0,\ s_1 = 0$ will not occur; exactly one of them must be positive.

We now relate optimal solutions to (HSDP) to those for (LP) and (LD).

Theorem 2. *Let* $(\mathbf{y}^*,\ \mathbf{x}^*,\ \tau^*,\ \theta^* = 0,\ \mathbf{s}^*,\ \kappa^*)$ *be a strictly-self complementary solution for (HSDP).*

(i) (LP) has a solution (feasible and bounded) if and only if $\tau^* > 0$. *In this case,* $\mathbf{x}^*/\tau^*$ *is an optimal solution for (LP) and* $\mathbf{y}^*/\tau^*,\ \mathbf{s}^*/\tau^*$ *is an optimal solution for (LD).*

(ii) (LP) has no solution if and only if $\kappa^* > 0$. *In this case,* $\mathbf{x}^*/\kappa^*$ *or* $\mathbf{y}^*/\kappa^*$*or both are certificates for proving infeasibility: if* $\mathbf{c}^T\mathbf{x}^* < 0$ *then (LD) is infeasible; if* $-\mathbf{b}^T\mathbf{y}^* < 0$ *then (LP) is infeasible; and if both* $\mathbf{c}^T\mathbf{x}^* < 0$ *and* $-\mathbf{b}^T\mathbf{y}^* < 0$ *then both (LP) and (LD) are infeasible.*

Proof. We prove the second statement. We first assume that one of (LP) and (LD) is infeasible, say (LD) is infeasible. Then there is some certificate $\bar{\mathbf{x}} \geq \mathbf{0}$ such that $\mathbf{A}\bar{\mathbf{x}} = \mathbf{0}$ and $\mathbf{c}^T\bar{\mathbf{x}} = -1$. Let $(\bar{\mathbf{y}} = \mathbf{0}, \bar{\mathbf{s}} = \mathbf{0})$ and

$$\alpha = \frac{n+1}{\mathbf{1}^T\bar{\mathrm{x}} + \mathbf{1}^T\bar{\mathbf{s}} + 1} > 0.$$

Then one can verify that

$$\tilde{\mathbf{y}}^* = \alpha\bar{\mathbf{y}},\ \tilde{\mathbf{x}}^* = \alpha\bar{\mathbf{x}},\ \tilde{\tau}^* = 0,\ \tilde{\theta}^* = 0,\ \tilde{\mathbf{s}}^* = \alpha\bar{\mathbf{s}},\ \tilde{\kappa}^* = \alpha$$

is a self-complementary solution for (HSDP). Since the supporting set (the set of positive entries) of a strictly complementary solution for (HSDP) is unique (see Exercise 6), $\kappa^* > 0$ at any strictly complementary solution for (HSDP).

Conversely, if $\tau^* = 0$, then $\kappa^* > 0$, which implies that $\mathbf{c}^T\mathbf{x}^* - \mathbf{b}^T\mathbf{y}^* < 0$, i.e., at least one of $\mathbf{c}^T\mathbf{x}^*$ and $-\mathbf{b}^T\mathbf{y}^*$ is strictly less than zero. Let us say $\mathbf{c}^T\mathbf{x}^* < 0$. In addition, we have

$$\mathbf{A}\mathbf{x}^* = \mathbf{0},\ \mathbf{A}^T\mathbf{y}^* + \mathbf{s}^* = \mathbf{0},\ (\mathbf{x}^*)^T\mathbf{s}^* = \mathbf{0} \text{ and } \mathbf{x}^* + \mathbf{s}^* > \mathbf{0}.$$

From Farkas' lemma (Exercise 5), $\mathbf{x}^*/\kappa^*$ is a certificate for proving dual infeasibility. The other cases hold similarly. ∎

To solve (HSDP), we have the following theorem that resembles the the central path analyzed for (LP) and (LD).

Theorem 3. *Consider problem (HSDP). For any* $\mu > 0$*, there is a unique* $(\mathbf{y}, \mathbf{x}, \tau, \theta, \mathbf{s}, \kappa)$ *in* $\overset{\circ}{\mathcal{F}}_h$*, such that*

$$\begin{pmatrix} \mathbf{x} \circ \mathbf{s} \\ \tau\kappa \end{pmatrix} = \mu\mathbf{1}.$$

Moreover, $(\mathbf{x}, \tau) = (\mathbf{1}, 1)$, $(\mathbf{y}, \mathbf{s}, \kappa) = (\mathbf{0}, \mathbf{0}, 1)$ *and* $\theta = 1$ *is the solution with* $\mu = 1$.

Theorem 3 defines an endogenous path associated with (HSDP):

$$C = \left\{ (\mathbf{y}, \mathbf{x}, \tau, \theta, \mathbf{s}, \kappa) \in \mathcal{F}_h^0 : \begin{pmatrix} \mathbf{x} \circ \mathbf{s} \\ \tau\kappa \end{pmatrix} = \frac{\mathbf{x}^T\mathbf{s} + \tau\kappa}{n+1}\mathbf{1} \right\}.$$

Furthermore, the potential function for (HSDP) can be defined as

$$\psi_{n+1+\rho}(\mathbf{x}, \tau, \mathbf{s}, \kappa) = (n + 1 + \rho)\log(\mathbf{x}^T\mathbf{s} + \tau\kappa) - \sum_{j=1}^{n} \log(x_j s_j) - \log(\tau\kappa), \quad (5.19)$$

where $\rho \geq 0$. One can then apply the interior-point algorithms described earlier to solve (HSDP) from the initial point $(\mathbf{x}, \tau) = (\mathbf{1}, 1)$, $(\mathbf{y}, \mathbf{s}, \kappa) = (\mathbf{0}, \mathbf{1}, 1)$ and $\theta = 1$ with $\mu = (\mathbf{x}^T\mathbf{s} + \boldsymbol{\tau\kappa})/(n + 1) = 1$.

The HSDP method outlined above enjoys the following properties:

- It does not require regularity assumptions concerning the existence of optimal, feasible, or interior feasible solutions.
- It can be initiated at $\mathbf{x} = \mathbf{1}$, $\mathbf{y} = \mathbf{0}$ and $\mathbf{s} = \mathbf{1}$, feasible or infeasible, on the central ray of the positive orthant (cone), and it does not require a big-M penalty parameter or lower bound.
- Each iteration solves a system of linear equations whose dimension is almost the same as that used in the standard (primal-dual) interior-point algorithms.
- If the linear program has a solution, the algorithm generates a sequence that approaches feasibility and optimality simultaneously; if the problem is infeasible or unbounded, the algorithm produces an infeasibility certificate for at least one of the primal and dual problems; see Exercise 5.

5.8 Summary

The simplex method has for decades been an efficient method for solving linear programs, despite the fact that there are no theoretical results to support its efficiency. Indeed, it was shown that in the worst case, the method may visit every vertex of the feasible region and this can be exponential in the number of variables and constraints. If on practical problems the simplex method behaved according to the worst case, even modest problems would require years of computer time to solve. The ellipsoid method was the first method that was proved to converge in time proportional to a polynomial in the size of the program, rather than to an exponential in the size. However, in practice, it was disappointingly less fast than the simplex method. Later, the interior-point method of Karmarkar significantly advanced the field of linear programming, for it not only was proved to be a polynomial-time method, but it was found in practice to be faster than the simplex method when applied to general linear programs.

The interior-point method is based on introducing a logarithmic barrier function with a weighting parameter μ; and now there is a general theoretical structure defining the analytic center, the central path of solutions as $\mu \to 0$, and the duals of these concepts. This structure is useful for specifying and analyzing various versions of interior point methods.

Most methods employ a step of Newton's method to find a point near the central path when moving from one value of μ to another. One approach is the predictor-corrector method, which first takes a step in the direction of decreasing μ and then a corrector step to get closer to the central path. Another method employs a potential function whose value can be decreased at each step, which guarantees convergence and assures that intermediate points simultaneously make progress toward the solution while remaining close to the central path.

Complete algorithms based on these approaches require a number of other features and details. For example, once systematic movement toward the solution is terminated, a final phase may move to a nearby vertex or to a non-vertex point on a face of the constraint set. Also, an initial phase must be employed to obtain an feasible point that is close to the central path from which the steps of the search algorithm can be started. These features are incorporated into several commercial software packages, and generally they perform well, able to solve very large linear programs in reasonable time.

5.9 Exercises

1. Using the simplex method, solve the program (5.1) and count the number of pivots required.
2. Prove the volume reduction rate in Theorem 1 for the ellipsoid method.
3. Develop a cutting plane method, based on the ellipsoid method, to find a point satisfying convex inequalities

$$f_i(\mathbf{x}) \leqslant 0,\ i = 1, \ldots, m,\ |\mathbf{x}|^2 \leqslant E^2,$$

where f_i's are convex functions of $\mathbf{x}$ in C^1.
4. Consider the linear program (5.5) and assume that $\overset{\circ}{\mathcal{F}}_p = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} > \mathbf{0}\}$ is nonempty and its optimal solution set is bounded. Show that the dual of the problem has a nonempty interior.
5. (Farkas' lemma) Prove: Exactly one of the feasible sets $\{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geqslant \mathbf{0}\}$ and $\{\mathbf{y} : \mathbf{y}^T\mathbf{A} \leqslant \mathbf{0},\ \mathbf{y}^T\mathbf{b} = 1\}$ is nonempty. A vector $\mathbf{y}$ in the latter set is called an infeasibility certificate for the former.

6. (Strict complementarity) Consider any linear program in standard form and its dual and let both of them be feasible. Then, there always exists a strictly complementary solution pair, $(\mathbf{x}^*,\ \mathbf{y}^*,\ \mathbf{s}^*)$, such that

$$x_j^* s_j^* = 0 \text{ and } x_j^* + s_j^* > 0 \text{ for all } j.$$

   Moreover, the supports of $\mathbf{x}^*$ and $\mathbf{s}^*$, $P^* = \{j : x_j^* > 0\}$ and $Z^* = \{j : x_j^* > 0\}$, are invariant among all strictly complementary solution pairs.

7. (Central path theorem) Let $(\mathbf{x}(\mu),\ \mathbf{y}(\mu),\ \mathbf{s}(\mu))$ be the central path of (5.9). Then prove

   (a) The central path point $(\mathbf{x}(\mu),\ \mathbf{y}(\mu),\ \mathbf{s}(\mu))$ is bounded for $0 < \mu \leqslant \mu^0$ and any given $0 < \mu^0 < \infty$.

   (b) For $0 < \mu' < \mu$,

$$\mathbf{c}^T\mathbf{x}(\mu') \leqslant \mathbf{c}^T\mathbf{x}(\mu) \text{ and } \mathbf{b}^T\mathbf{y}(\mu') \geqslant \mathbf{b}^T\mathbf{y}(\mu).$$

   Furthermore, if $\mathbf{x}(\mu') \neq \mathbf{x}(\mu)$ and $\mathbf{y}(\mu') \neq \mathbf{y}(\mu)$,

$$\mathbf{c}^T\mathbf{x}(\mu') < \mathbf{c}^T\mathbf{x}(\mu) \text{ and } \mathbf{b}^T\mathbf{y}(\mu') > \mathbf{b}^T\mathbf{y}(\mu).$$

   (c) $(\mathbf{x}(\mu),\ \mathbf{y}(\mu),\ \mathbf{s}(\mu))$ converges to an optimal solution pair for (LP) and (LD). Moreover, the limit point $\mathbf{x}(0)_{P^*}$ is the analytic center on the primal optimal face, and the limit point $\mathbf{s}(0)_{Z^*}$ is the analytic center on the dual optimal face, where $(P^*,\ Z^*)$ is the strict complementarity partition of the index set $\{1, 2, \ldots,\ n\}$.

8. Consider a primal-dual interior point $(\mathbf{x},\ \mathbf{y},\ \mathbf{s}) \in \mathcal{N}(\eta)$ where $\eta < 1$. Prove that there is a fixed quantity $\delta > 0$ such that

$$x_j \geqslant \delta, \text{ for all } j \in P^*$$

   and

$$s_j \geqslant \delta, \text{ for all } j \in Z^*,$$

   where $(P^*,\ Z^*)$ is defined in Exercise 6.

9. (Potential level theorem) Define the potential level set

$$\Psi(\delta) := \{(\mathbf{x},\ \mathbf{y},\ \mathbf{s}) \in \mathring{\mathcal{F}} : \psi_{n+\rho}(\mathbf{x},\ \mathbf{s}) \leqslant \delta\}.$$

   Prove

   (a)

$$\Psi(\delta^1) \subset \Psi(\delta^2) \text{ if } \delta^1 \leqslant \delta^2.$$

   (b) For every δ, $\Psi(\delta)$ is bounded and its closure $\overline{\Psi}(\delta)$ has non-empty intersection with the solution set.

10. Given $\mathbf{0} < \mathbf{x},\ \mathbf{0} < \mathbf{s} \in E^n$, show that

$$n \log(\mathbf{x}^T\mathbf{s}) - \sum_{j=1}^{n} \log(x_j s_j) \geqslant n \log n$$

and

$$\mathbf{x}^T\mathbf{s} \leqslant \exp\left[\frac{\psi_{n+p}(\mathbf{x}, \mathbf{s}) - n \log n}{p}\right].$$

11. (Logarithmic approximation) If $\mathbf{d} \in E^n$ such that $|\mathbf{d}|_\infty < 1$ then

$$\mathbf{1}^T\mathbf{d} \geqslant \sum_{i=1}^{n} \log(1 + d_i) \geqslant \mathbf{1}^T\mathbf{d} - \frac{|\mathbf{d}|^2}{2(1 - |\mathbf{d}|_\infty)}.$$

[Note:If $\mathbf{d} = (d_1,\ d_2,\ \ldots d_n)$ then $|\mathbf{d}|_\infty \equiv \max_i\{d_j\}$.]

12. Let the direction $(\mathbf{d}_\mathbf{x},\ \mathbf{d}_\mathbf{y},\ \mathbf{d}_\mathbf{s})$ be generated by system (5.13) with $\gamma = n/(n+\rho)$ and $\mu = \mathbf{x}^T\mathbf{s}/n$, and let the step size be

$$\alpha = \frac{\theta\sqrt{\min(\mathbf{Xs})}}{|(\mathbf{XS})^{-1/2}(\frac{\mathbf{x}^T\mathbf{s}}{(n+\rho)}1 - \mathbf{Xs})|}, \tag{5.20}$$

where θ is a positive constant less than 1. Let

$$\mathbf{x}^+ = \mathbf{x} + \alpha\mathbf{d}_\mathbf{x},\ \mathbf{y}^+ = \mathbf{y} + \alpha\mathbf{d}_\mathbf{y},\ \text{and } \mathbf{s}^+ = \mathbf{s} + \alpha\mathbf{d}_\mathbf{s}.$$

Then, using Exercise 11 and the concavity of the logarithmic function show $(\mathbf{x}^+,\ \mathbf{y}^+,\ \mathbf{s}^+) \in \mathring{\mathcal{F}}$ and

$$\begin{aligned}&\psi_{n+\rho}(\mathbf{x}^+,\ \mathbf{s}^+) - \psi_{n+\rho}(\mathbf{x},\ \mathbf{s})\\ &\quad\leqslant -\theta\sqrt{\min(\mathbf{Xs})}\ |(\mathbf{Xs})^{-1/2}(1 - \frac{(n+\rho)}{\mathbf{x}^{\tau}\mathrm{s}}\mathbf{Xs})| + \frac{\theta^2}{2(1-\theta)}\ .\end{aligned}$$

13. Let $\mathbf{v} = \mathbf{Xs}$ in Exercise 12. Prove

$$\sqrt{\min(\mathbf{v})}|\mathbf{V}^{-1/2}(\mathbf{1} - \frac{(n+\rho)}{\mathbf{1}^T\mathbf{V}}\mathbf{v})| \geqslant \sqrt{3/4},$$

where $\mathbf{V}$ is the diagonal matrix of $\mathbf{v}$. Thus, the two exercises imply

$$\psi_{n+\rho}(\mathbf{x}^+,\ \mathbf{s}^+) - \psi_{n+\rho}(\mathbf{x},\ \mathbf{s}) \leqslant -\theta\sqrt{3/4} + \frac{\theta^2}{2(1-\theta)} = -\delta$$

for a constant δ. One can verify that $\delta > 0.2$ when $\theta = 0.4$.

14. Prove property (5.18) for (HDSP).

15. Prove Theorem 1

References

5.1 Computation and complexity models were developed by a number of scientists; see, e.g., Cook [C5], Hartmanis and Stearns [H5] and Papadimitriou and Steiglitz [P2] for the bit complexity models and Blum et al. [B21] for the real number arithmetic model. For a general discussion of complexity see Vavasis [V4]. For a comprehensive treatment which served as the basis for much of this chapter, see Ye [Y3].

5.2 The Klee Minty example is presented in [K5]. Much of this material is based on a teaching note of Cottle on Linear Programming taught at Stanford [C6]. Practical performances of the simplex method can be seen in Bixby [B18]. The simplex method efficiency for the Markov Decision Process is due to Ye [269].

5.3 The ellipsoid method was developed by Khachiyan [K4]; more developments of the ellipsoid method can be found in Bland, Goldfarb and Todd [B20].

5.3 The analytic center for a convex polyhedron given by linear inequalities was introduced by Huard [H12], and later by Sonnevend [S8]. The barrier function was introduced by Frisch [F19]. The central path was analyzed in McLinden [M3], Megiddo [M4], and Bayer and Lagarias [B3, B4], Gill et al. [G5].

5.5 Path-following algorithms were first developed by Renegar [R1]. A primal barrier or path-following algorithm was independently analyzed by Gonzaga [G13]. Both Gonzaga [G13] and Vaidya [V1] extended the rank-one updating technique [K2] for solving the Newton equation of each iteration, and proved that each iteration uses $O(n^{2.5})$ arithmetic operations on average. Kojima, Mizuno and Yoshise [K6] and Monteiro and Adler [M7] developed a symmetric primal-dual path-following algorithm with the same iteration and arithmetic operation bounds.

5.6–5.7 Predictor-corrector algorithms were developed by Mizuno et al. [M6]. A more practical predictor-corrector algorithm was proposed by Mehrotra [M5] (also see Lustig et al. [L19] and Zhang and Zhang [Z3]). Mehrotra's technique has been used in almost all linear programming interior-point implementations. A primal potential reduction algorithm was initially proposed by Karmarkar [K2]. The primal-dual potential function was proposed by Tanabe [T2] and Todd and Ye [T5]. The primal-dual potential reduction algorithm was developed by Ye [Y1], Freund [F18], Kojima, Mizuno and Yoshise [K7], Goldfarb and Xiao [G11], Gonzaga and Todd [G14], Todd [T4], Tunçel [T10], Tutuncu [T11], and others. The homogeneous and self-dual embedding method can be found in Ye et al. [Y2], Luo et al. [L18], Andersen and Ye [A5], and many others. It is also implemented in most linear programming software packages such as SEDUMI of Sturm [S11].

5.1–5.7 There are several comprehensive text books which cover interior-point linear programming algorithms. They include Bazaraa, Jarvis and Sherali [B6], Bertsekas [B12], Bertsimas and Tsitsiklis [B13], Cottle [C6], Cottle, Pang and Stone [C7], Dantzig and Thapa [D9, D10], Fang and Puthenpura [F2], den Hertog [H6], Murty [M12], Nash and Sofer [N1], Nesterov [N2], Roos et al. [R4], Renegar [R2], Saigal [S1], Vanderebei [V3], and Wright [W8].