6

Karmarkar's Projective Scaling Algorithm

In the fall of 1984, N. K. Karmarkar of AT&T Bell Laboratories proposed a new polynomial-time algorithm for linear programming. Unlike the ellipsoid method, the new algorithm not only possesses better complexity than the simplex method in the worse-case analysis, but also shows the potential to rival the simplex approach for large-scale real-world applications. This development quickly captured the attention of everyone in the field.

Radically different from the simplex method, Karmarkar's original algorithm considers a linear programming problem over a *simplex* structure and moves through the interior of the polytope of feasible domain by transforming the space at each step to place the current solution at the center of the polytope. The concept of reaching the optimum through the interior has stimulated many new researches in developing so-called *interior-point methods.* Numerous extensions and variants have been reported.

In this chapter, we first introduce the basic idea of Karmarkar's algorithm, then describe the algorithm in detail with a proof of polynomial-time complexity. Some extensions and a computer implementation procedure will also be discussed. The so-called *affine scaling* algorithms will be left for discussion in the next chapter.

6.1 BASIC IDEAS OF KARMARKAR'S ALGORITHM

As discussed in Chapter 5, the philosophy of solving an optimization problem via an iterative scheme is to start with a "rough" solution and successively improve the current solution until a desired goal is met. The performance of an iterative algorithm depends

upon two key factors: (1) How many steps (iterations) does it take? (2) How much computation does it involve in each iteration?

The simplex method starts with an extreme point and keeps moving to a better neighboring extreme point at each iteration until an optimal solution or infeasibility is reached. In this scheme, the computational work at each iteration is minimized by limiting the searches to only those edge directions which lead to adjacent extreme points. But, as the Klee-Minty example showed, the simplex method may have to travel a long path on the boundary of the feasible domain and visit almost every extreme point before it stops. This *boundary* approach suffers from heavy computation in large-scale applications, since the feasible domain may contain a huge number of extreme points. Therefore one alternative idea is to travel across the interior of the feasible domain along a "shorter path" in order to reduce the total number of iterations. However, this *interior-point* approach usually requires the consideration of all feasible directions for a better movement at each iteration. In other words, the new philosophy is to reduce the number of iterations at the expense of heavier computation at each iteration.

In general, it is not an easy task to identify the "best direction of movement" among all feasible directions at a particular interior point of the feasible domain. However, Karmarkar noticed two fundamental insights, assuming the feasible domain is a polytope.

1. If the current interior solution is near the center of the polytope, then it makes sense to move in the direction of steepest descent of the objective function to achieve a minimum value.
2. Without changing the problem in any essential way, an appropriate transformation can be applied to the solution space such that the current interior solution is placed near the center in the transformed solution space.

The first insight can be observed in Figure 6.1. Since $\mathbf{x}^1$ is near the center of the polytope, we can improve the solution substantially by moving it in a direction of steepest descent. But if an off-center point $\mathbf{x}^2$ is so moved, it will soon be out of the feasible domain before much improvement is made.

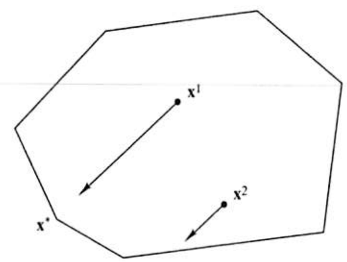

Figure 6.1

Karmarkar observed the second insight via the so-called *projective transformation,* whereby straight lines remain straight lines while angles and distances distort such that we can view any interior point as the center of the polytope in a distorted picture. One can use imagination to verify this with Figure 6.1 by viewing it at an angle and distance that makes $\mathbf{x}^2$ appear to be near the center of the polytope. Such a distortion scarcely alters anything essential to the problem but merely looks at it from a different viewpoint.

With these two fundamental insights, the basic strategy of Karmarkar's projective scaling algorithm is straightforward. We take an interior solution, transform the solution space so as to place the current solution near the center of the polytope in the transformed space, and then move it in the direction of steepest descent, but not all the way to the boundary of the feasible domain in order to have it remain as an interior solution. Then take the inverse transformation to map the improved solution back to the original solution space as a new interior solution. We repeat the process until an optimum is obtained with the desired accuracy.

6.2 KARMARKAR'S STANDARD FORM

Following the basic strategy of the projective scaling, Karmarkar's algorithm has a preferred standard form for linear programming:

$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} \tag{6.1a}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{0} \tag{6.1b}$$

$$\mathbf{e}^T\mathbf{x} = 1, \qquad \mathbf{x} \geq \mathbf{0} \tag{6.1c}$$

where $\mathbf{A}$ is an $m \times n$ dimensional matrix of full row rank, $\mathbf{e}^T = (1, 1, \ldots, 1)$ is an n-vector of all ones, and $\mathbf{c}, \mathbf{x} \in R^n$.

A feasible solution vector $\mathbf{x}$ of problem (6.1) is defined to be an *interior solution* if every variable x_i is strictly positive. Note from (6.1c) that the feasible domain is a bounded set, hence it becomes a polytope. A consistent problem in Karmarkar's standard form certainly has a finite infimum. Karmarkar made two assumptions for his algorithm.

(A1) $\mathbf{A}\mathbf{e} = \mathbf{0}$, so that $\mathbf{x}^0 = \frac{\mathbf{e}}{n} = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)^T$ is an initial interior solution.

(A2) The optimal objective value of problem (6.1) is zero.

We shall see later how a linear programming problem can be cast into Karmarkar's standard form satisfying the two assumptions. Here are a couple of examples that fit our description.

Example 6.1

$$\begin{aligned} \text{Minimize} \quad & -x_1 + 1 \\ \text{subject to} \quad & x_2 - x_3 = 0 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Example 6.2

$$\begin{aligned} \text{Minimize} \quad & -x_1 - 2x_2 + 4x_5 \\ \text{subject to} \quad & x_2 - x_3 = 0 \\ & 2x_1 - 2x_2 + 4x_3 - 4x_5 = 0 \\ & x_1 + 2x_2 + x_4 - 4x_5 = 0 \\ & x_1 + x_2 + x_3 + x_4 + x_5 = 1 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

6.2.1 The Simplex Structure

Expression (6.1c) defines a regular polygon in the n-dimensional Euclidean space, namely

$$\Delta = \{\mathbf{x} \in R^n \mid \sum_{i=1}^{n} x_i = 1, x_i \geq 0\} \tag{6.2}$$

It is clearly seen that in R^1, $\Delta = \{1\}$ which is a singleton; in R^2, it is the line segment between the points $(0, 1)$ and $(1, 0)$; in R^3, it is the triangular area formed by $(0, 0, 1)$, $(0, 1, 0)$ and $(1, 0, 0)$; and in R^4, it becomes the pyramid with vertices at $(0, 0, 0, 1)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$, and $(0, 0, 0, 1)$. It is also easy to see that, in R^n, Δ has exactly n vertices, $C(n, 2)$ edges, $C(n, n-1)$ facets, and its center at $\mathbf{e}/n$. Just noting the coordinates of the center and each vertex of Δ (see Figure 6.2), we can show that the radius of the smallest circumscribing spheroid of Δ is given by

$$R = \frac{\sqrt{n-1}}{\sqrt{n}} \tag{6.3}$$

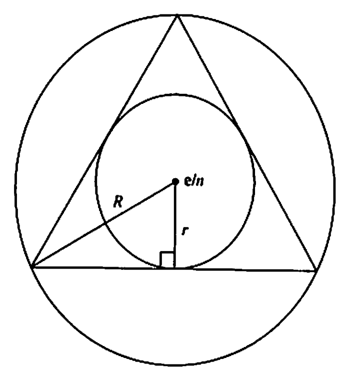

Figure 6.2

Similarly, the radius of the largest inscribing spheroid in Δ is given by

$$r = \frac{1}{\sqrt{n(n-1)}} \tag{6.4}$$

6.2.2 Projective Transformation on the Simplex

Let $\bar{\mathbf{x}}$ be an interior point of Δ, i.e., $\bar{x}_i > 0$ for $i = 1, \ldots, n$ and $\sum_{i=1}^{n} \bar{x}_i = 1$. We can define an $n \times n$ diagonal matrix

$$\bar{\mathbf{X}} = \operatorname{diag}(\bar{\mathbf{x}}) = \begin{bmatrix} \bar{x}_1 & 0 & \ldots & 0 \\ 0 & \bar{x}_2 & \ldots & 0 \\ \vdots & \vdots & & \\ 0 & 0 & \ldots & \bar{x}_n \end{bmatrix} \tag{6.5}$$

It is obvious that matrix $\bar{\mathbf{X}}$ is nonsingular and its inverse matrix $\bar{\mathbf{X}}^{-1}$ is also a diagonal matrix but with $1/\bar{x}_i$ as its ith diagonal elements for $i = 1, \ldots, n$. Moreover, we can define a *projective transformation* $T_{\bar{\mathbf{x}}}$ from Δ to Δ such that

$$T_{\bar{\mathbf{x}}}(\mathbf{x}) = \frac{\bar{\mathbf{X}}^{-1}\mathbf{x}}{\mathbf{e}^T\bar{\mathbf{X}}^{-1}\mathbf{x}} \quad \text{for each } \mathbf{x} \in \Delta \tag{6.6}$$

Notice that $\bar{\mathbf{X}}^{-1}\mathbf{x}$ is an n-dimensional column vector and $\mathbf{e}^T\bar{\mathbf{X}}^{-1}\mathbf{x}$ is a scalar which equals the sum of all elements in the vector $\bar{\mathbf{X}}^{-1}\mathbf{x}$. Therefore, the elements in $T_{\bar{\mathbf{x}}}(\mathbf{x})$ are *normalized* with sum equal to 1. In other words, $T_{\bar{\mathbf{x}}}(\mathbf{x}) \in \Delta$, and $T_{\bar{\mathbf{x}}}$ is indeed a well-defined mapping from Δ to itself.

Example 6.3

Consider the simplex Δ in R^3 as shown in Figure 6.3. Let $\mathbf{x} = (1, 0, 0)^T$, $\mathbf{y} = (0, 1, 0)^T$, $\mathbf{z} = (0, 0, 1)^T$, $\mathbf{a} = (3/10, 1/10, 3/5)^T$, $\mathbf{b} = (1/3, 0, 2/3)^T$, $\mathbf{c} = (0, 1/7, 6/7)^T$, $\mathbf{d} = (3/4, 1/4, 0)^T$.

Since point A is an interior point, we can define

$$\mathbf{X}_a = \begin{bmatrix} 3/10 & 0 & 0 \\ 0 & 1/10 & 0 \\ 0 & 0 & 3/5 \end{bmatrix}$$

Then we have

$$\mathbf{X}_a^{-1} = \begin{bmatrix} 10/3 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 5/3 \end{bmatrix}$$

Moreover, we see that $T_a(\mathbf{x}) = (1, 0, 0)^T$, $T_a(\mathbf{y}) = (0, 1, 0)^T$, $T_a(\mathbf{z}) = (0, 0, 1)^T$, $T_a(\mathbf{a}) = (1/3, 1/3, 1/3)^T$, $T_a(\mathbf{b}) = (1/2, 0, 1/2)^T$, $T_a(\mathbf{c}) = (0, 1/2, 1/2)^T$, $T_a(\mathbf{d}) = (1/2, 1/2, 0)^T$.

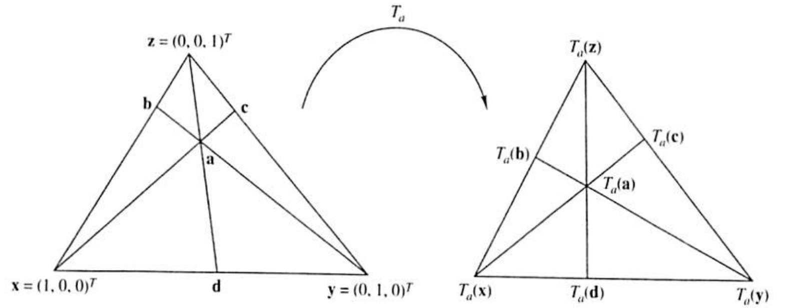

Figure 6.3

Example 6.3 showed that the scale and the angle in the transformed space are distorted such that a current interior point, in this case point a, becomes the center of Δ. In general, we can show the following results:

(T1) $T_{\bar{\mathbf{x}}}$ is a well-defined mapping from Δ to Δ, if $\bar{\mathbf{x}}$ is an interior point of Δ.
(T2) $T_{\bar{\mathbf{x}}}(\bar{\mathbf{x}}) = \mathbf{e}/n$ becomes the center of Δ.
(T3) $T_{\bar{\mathbf{x}}}(\mathbf{x})$ is a vertex of Δ if $\mathbf{x}$ is a vertex.
(T4) $T_{\bar{\mathbf{x}}}(\mathbf{x})$ is on the boundary of Δ if $\mathbf{x}$ is on the boundary.
(T5) $T_{\bar{\mathbf{x}}}(\mathbf{x})$ is an interior point of Δ if $\mathbf{x}$ is in the interior.
(T6) $T_{\bar{\mathbf{x}}}$ is a one-to-one and onto mapping with an inverse transformation $T_{\bar{\mathbf{x}}}^{-1}$ such that

$$T_{\bar{\mathbf{x}}}^{-1}(\mathbf{y}) = \frac{\bar{\mathbf{X}}\mathbf{y}}{\mathbf{e}^T\bar{\mathbf{X}}\mathbf{y}} \quad \text{for each } \mathbf{y} \in \Delta \tag{6.7}$$

6.3 KARMARKAR'S PROJECTIVE SCALING ALGORITHM

Consider a linear programming problem in Karmarkar's standard form (6.1). Its feasible domain is a polytope formed by the intersection of the *null space* of the constraint matrix $\mathbf{A}$, i.e., $\{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{0}\}$ and the simplex Δ in R^n. Let $\bar{\mathbf{x}} > \mathbf{0}$ be an interior feasible solution, then the projective transformation $T_{\bar{\mathbf{x}}}$ maps $\mathbf{x} \in \Delta$ to

$$\mathbf{y} = T_{\bar{\mathbf{x}}}(\mathbf{x}) = \frac{\bar{\mathbf{X}}^{-1}\mathbf{x}}{\mathbf{e}^T\bar{\mathbf{X}}^{-1}\mathbf{x}}$$

and we can denote $\mathbf{x}$ in terms of its image $\mathbf{y}$ by the formula

$$\mathbf{x} = \frac{\bar{\mathbf{X}}\mathbf{y}}{\mathbf{e}^T\bar{\mathbf{X}}\mathbf{y}} \tag{6.8}$$

Plugging the value of $\mathbf{x}$ into problem (6.1) according to Equation (6.8), and remembering that $T_{\bar{\mathbf{x}}}$ maps Δ onto Δ, we have a corresponding problem in the *transformed space*, namely,

$$\text{minimize} \quad \frac{\mathbf{c}^T\bar{\mathbf{X}}\mathbf{y}}{\mathbf{e}^T\bar{\mathbf{X}}\mathbf{y}} \tag{6.1'a}$$

$$\text{subject to} \quad \mathbf{A}\bar{\mathbf{X}}\mathbf{y} = \mathbf{0} \tag{6.1'b}$$

$$\mathbf{e}^T\mathbf{y} = 1, \qquad \mathbf{y} \geq \mathbf{0} \tag{6.1'c}$$

Note that in problem (6.1') the image of $\bar{\mathbf{x}}$, i.e., $\bar{\mathbf{y}} = T_{\bar{\mathbf{x}}}(\bar{\mathbf{x}}) = \mathbf{e}/n$, becomes a feasible solution that sits at the center of the simplex Δ. If we denote the constraint matrix by

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}\bar{\mathbf{X}} \\ \mathbf{e}^T \end{bmatrix} \tag{6.9}$$

then any direction $\mathbf{d} \in R^n$ in the null space of matrix $\mathbf{B}$, i.e., $\mathbf{Bd} = \mathbf{0}$, is a feasible direction of movement for $\bar{\mathbf{y}}$. But remember that the distance from the center of Δ to its closest boundary is given by the radius r in Equation (6.4). Therefore, if we denote the norm of $\mathbf{d}$ by $||\mathbf{d}||$, then

$$\mathbf{y}(\alpha) = \bar{\mathbf{y}} + \alpha r\left(\frac{\mathbf{d}}{||\mathbf{d}||}\right) \tag{6.10}$$

remains feasible to problem (6.1') as long as $\mathbf{d}$ lies in the null space of matrix $\mathbf{B}$ and $0 \leq \alpha < 1$. In particular, if $0 \leq \alpha < 1$, then $\mathbf{y}(\alpha)$ remains an interior solution, and its inverse image

$$\mathbf{x}(\alpha) = T_{\bar{\mathbf{x}}}^{-1}(\mathbf{y}(\alpha)) = \frac{\bar{\mathbf{X}}\mathbf{y}(\alpha)}{\mathbf{e}^T\bar{\mathbf{X}}\mathbf{y}(\alpha)} \tag{6.11}$$

becomes a new interior solution to the original problem (6.1). Also note that since

$$r = \frac{1}{\sqrt{n(n-1)}} > \frac{1}{n}$$

we may replace Equation (6.10) by

$$\mathbf{y}(\alpha) = \bar{\mathbf{y}} + \left(\frac{\alpha}{n}\right)\left(\frac{\mathbf{d}}{||\mathbf{d}||}\right) \tag{6.10'}$$

for $0 \leq \alpha \leq 1$, to obtain a new interior feasible solution.

After determining the structure of the feasible directions in the transformed space, we focus on finding a good feasible direction that eventually leads to an optimal solution. Since $\bar{\mathbf{y}}$ is at the center of Δ, from the first insight mentioned in Section 6.1, it makes sense to move along the steepest descent of the objective function. Although the objective function (6.1'a) is no longer a linear function—actually it is a fractional linear function—Karmarkar pointed out that the linear numerator function $\mathbf{c}^T\bar{\mathbf{X}}\mathbf{y}$ could be a good indication of the reduction of the objective function. Therefore, we take its negative gradient,

which is $-\mathbf{c}^T\bar{\mathbf{X}}$, or equivalently $-\bar{\mathbf{X}}\mathbf{c}$, as a good candidate. In order to keep feasibility, we further project the negative gradient into the null space of the constraint matrix $\mathbf{B}$. From basic knowledge of linear algebra, we have the following formula for the projected negative gradient:

$$\bar{\mathbf{d}} = -[\mathbf{I} - \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{B}]\bar{\mathbf{X}}\mathbf{c} \tag{6.12}$$

Now it is easy to describe the basic steps of Karmarkar's algorithm. The algorithm starts with an interior solution in the original space, maps the solution to the center of Δ by a projective transformation, applies Equation (6.12) to find a good moving direction, chooses an appropriate step-length and uses Equation (6.10') to move to a new interior feasible solution in the transformed space, and then maps the new solution back to the original space according to Equation (6.11) to gain a fair amount of reduction in the objective function. By repeating this iterative process, Karmarkar showed his algorithm could terminate in $O(nL)$ iterations to reach an optimal solution. We shall study his proof in the next section. Here we provide an iterative procedure for the implementation of Karmarkar's algorithm.

Step 1 (initialization): Set $k = 0$, $\mathbf{x}^0 = \mathbf{e}/n$, and L to be a large positive integer.

Step 2 (optimality check): If

$$\mathbf{c}^T\mathbf{x}^k \leq 2^{-L}\left(\mathbf{c}^T\frac{\mathbf{e}}{n}\right)$$

then stop with an optimal solution $\mathbf{x}^* = \mathbf{x}^k$. Otherwise, go to Step 3.

Step 3 (find a better solution): Let

$$\mathbf{X}_k = \operatorname{diag}(\mathbf{x}^k)$$

$$\mathbf{B}_k = \begin{bmatrix} \mathbf{A}\mathbf{X}_k \\ \mathbf{e}^T \end{bmatrix}$$

$$\mathbf{d}^k = -[\mathbf{I} - \mathbf{B}_k^T(\mathbf{B}_k\mathbf{B}_k^T)^{-1}\mathbf{B}_k]\mathbf{X}_k\mathbf{c}$$

$$\mathbf{y}^{k+1} = \frac{\mathbf{e}}{n} + \frac{\alpha}{n}\left(\frac{\mathbf{d}^k}{||\mathbf{d}^k||}\right) \qquad \text{for some } 0 < \alpha \leq 1$$

$$\mathbf{x}^{k+1} = \frac{\mathbf{X}_k\mathbf{y}^{k+1}}{\mathbf{e}^T\mathbf{X}_k\mathbf{y}^{k+1}}$$

Set $k = k + 1$; go to Step 2.

Note that in this computational procedure $\mathbf{x}^k$ is always an interior feasible solution; $\mathbf{X}_k$ is an n-dimensional diagonal matrix with the ith element of vector $\mathbf{x}^k$ as its ith diagonal element; $\mathbf{B}_k$ is the constraint matrix of a linear programming problem in Karmarkar's standard form as defined in Equation (6.9); $\mathbf{d}^k$ is a feasible direction of the projected negative gradient as defined in Equation (6.12); $\mathbf{y}^{k+1}$ is a new interior feasible solution in the transformed space as defined in Equation (6.10'); and $\mathbf{x}^{k+1}$ is a new interior feasible solution as defined by Equation (6.11). Also note that the constant L in Step 2 is usually chosen to be the problem size as defined in Chapter 5 or a multiple of the problem size such that $2^{-L} < \varepsilon$ for a given tolerance $\varepsilon > 0$. We shall prove in the next section that, if the step size α is chosen to be $1/3$, then the algorithm terminates in $O(nL)$ iterations. But for real applications, a larger value of α tends to speed up the convergence.

The following example illustrates one iteration of Karmarkar's algorithm.

Example 6.4

Solve Example 6.1 by Karmarkar's algorithm.

First we see that the linear programming problem is in Karmarkar's standard form, which satisfies both assumptions (A1) and (A2). Hence we start with

$$\mathbf{x}^0 = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$$

and note that $\mathbf{A} = [0, 1, -1]$ and $\mathbf{c}^T = (-1, 0, 0)$.

Now check Step 2. From Equation (5.4), we can choose $L = 20$ and easily see that the objective value at $\mathbf{x}^0$ is too high. Therefore we have to find a better solution.

For Step 3, we define

$$\mathbf{X}_0 = \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/3 \end{bmatrix}$$

then $\mathbf{AX}_0 = [0, 1/3, -1/3]$ and

$$\mathbf{B}_0 = \begin{bmatrix} 0 & 1/3 & -1/3 \\ 1 & 1 & 1 \end{bmatrix}$$

Moreover, the moving direction is given by

$$\mathbf{d}^0 = -[\mathbf{I} - \mathbf{B}_0^T(\mathbf{B}_0\mathbf{B}_0^T)^{-1}\mathbf{B}_0]\mathbf{X}_0\mathbf{c} = \left(\frac{2}{9}, \frac{-1}{9}, \frac{-1}{9}\right)^T$$

with norm $\|\mathbf{d}\| = \sqrt{6}/9$. For purposes of illustration, we choose $\alpha = 1/\sqrt{6}$ to obtain a new solution in the transformed space

$$\mathbf{y}^0 = (1/3, 1/3, 1/3)^T + (1/3)(1/\sqrt{6})(9/\sqrt{6})(2/9, -1/9, -1/9)^T = (4/9, 5/18, 5/18)^T$$

Hence the new interior feasible solution is given by

$$\mathbf{x}^1 = \frac{\mathbf{X}_0\mathbf{y}^0}{\mathbf{e}^T\mathbf{X}_0\mathbf{y}^0} = (4/9, 5/18, 5/18)^T$$

Continuing this iterative process, Karmarkar's algorithm will stop at the optimal solution $\mathbf{x}^* = (1, 0, 0)^T$. It is worth mentioning that if we take $\alpha = 6/\sqrt{6} > 1$, then $\mathbf{y}^1 = (1, 0, 0)^T$ and $\mathbf{x}^1 = \mathbf{x}^*$. Hence direction $\mathbf{d}^0$ really points to the optimal solution.

6.4 POLYNOMIAL-TIME SOLVABILITY

In this section we show that Karmarkar's algorithm terminates in $O(nL)$ iterations under assumptions (A1) and (A2). The key to proving this polynomial-time solvability is to find

an appropriate step-length α such that the objective value after each iteration decreases at a *geometric* rate. In particular, Karmarkar showed that, for $\alpha = 1/3$,

$$\mathbf{c}^T\mathbf{x}^k \le e^{-k/5n}(\mathbf{c}^T\mathbf{x}^0) \qquad \text{for } k = 1, 2, \ldots \tag{6.13}$$

In this way, for L (or a multiple of it) large enough such that $2^{-L}(\mathbf{c}^T\mathbf{x}^0) \approx 0$, we need only choose k satisfying

$$e^{-k/5n} \le 2^{-L} < \frac{\varepsilon}{\mathbf{c}^T\mathbf{x}^0} \tag{6.14}$$

Then we can terminate the algorithm to the precision level we want. Taking the natural logarithm of (6.14), we see the requirement becomes

$$\frac{-k}{5n} \le \log_e 2^{-L} < \log_2 2^{-L} = -L \tag{6.15}$$

In other words, if $k > 5nL$, the algorithm could be terminated with $\mathbf{c}^T\mathbf{x}^k < \varepsilon$. Hence Karmarkar's algorithm requires only a polynomial number $O(nL)$ of iterations.

Notice that (6.13) is equivalent to

$$\log_e(\mathbf{c}^T\mathbf{x}^k) \le \frac{-k}{5n} + \log_e(\mathbf{c}^T\mathbf{x}^0) \tag{6.16}$$

or

$$n\log_e(\mathbf{c}^T\mathbf{x}^k) \le n\log_e(\mathbf{c}^T\mathbf{x}^0) - \frac{k}{5} \tag{6.17}$$

This shows that the requirement (6.13) will be met if at each iteration we can reduce the function value of $n\log_e(\mathbf{c}^T\mathbf{x})$ by at least a constant of $1/5$. Remember that the direction of movement in Karmarkar's algorithm was chosen to be the projected negative gradient in order to reduce the function value of $\mathbf{c}^T\mathbf{X}_k\mathbf{y}$, which is clearly different from the desired function $n\log_e(\mathbf{c}^T\mathbf{x})$. To link these two different objectives together, Karmarkar defined a *potential function* for each interior point $\mathbf{x}$ of Δ and cost vector $\mathbf{c}$ as follows:

$$f(\mathbf{x}) \equiv f(\mathbf{x}; \mathbf{c}) = n\log_e(\mathbf{c}^T\mathbf{x}) - \sum_{j=1}^{n}\log_e x_j = \sum_{j=1}^{n}\log_e\left(\frac{\mathbf{c}^T\mathbf{x}}{x_j}\right) \tag{6.18}$$

Two simple properties can be derived from this definition. First, in the transformed solution space, we have a corresponding potential function

$$f'(\mathbf{y}) = f(\mathbf{y}; \mathbf{X}_k\mathbf{c}) = \sum_{j=1}^{n}\log_e\left(\frac{\mathbf{c}^T\mathbf{X}_k\mathbf{y}}{y_j}\right) \tag{6.19a}$$

Remember that

$$\mathbf{y} = T_{\mathbf{x}^k}(\mathbf{x}) = \frac{\mathbf{X}_k^{-1}\mathbf{x}}{\mathbf{e}^T\mathbf{X}_k^{-1}\mathbf{x}}$$

hence we have

$$f(\mathbf{y}; \mathbf{X}_k\mathbf{c}) = \sum_{j=1}^{n} \log_e \left(\frac{\mathbf{c}^T\mathbf{x}}{x_j/x_j^k} \right)$$
$$= \sum_{j=1}^{n} \log_e \left(\frac{\mathbf{c}^T\mathbf{x}}{x_j} \right) + \sum_{j=1}^{n} \log_e x_j^k$$
$$= f(\mathbf{x}; \mathbf{c}) + \log_e(\det \mathbf{X}_k) \tag{6.19b}$$

where $\det \mathbf{X}_k$ is the determinant of the diagonal matrix $\mathbf{X}_k$.

The previous equation shows that the potential function is an invariant under the projective transformation $T_{\mathbf{x}^k}$ which satisfies the relation

$$f'(\mathbf{y}) = f(\mathbf{x}) + \log_e(\det \mathbf{X}_k) \tag{6.20}$$

The second property is based on the observation that

$$f(\mathbf{x}^k) = f'\left(\frac{\mathbf{e}}{n}\right) - \log_e(\det \mathbf{X}_k)$$

and

$$f(\mathbf{x}^{k+1}) = f'(\mathbf{y}^{k+1}) - \log_e(\det \mathbf{X}_k)$$

Therefore if we can reduce the potential function $f'(\mathbf{e}/n)$ by a constant in the transformed solution space at each iteration, then $f(\mathbf{x}^k)$ is reduced by the same amount after each iteration taken in the original space. In particular, if we can show that

$$f'(\mathbf{y}^{k+1}) \le f'\left(\frac{\mathbf{e}}{n}\right) - \frac{1}{5} \qquad \text{for } k = 0, 1, 2, \ldots \tag{6.21}$$

then

$$f(\mathbf{x}^{k+1}) \le f(\mathbf{x}^k) - \frac{1}{5} \qquad \text{for } k = 0, 1, 2, \ldots$$

Consequently, we have

$$f(\mathbf{x}^k) \le f(\mathbf{x}^0) - \frac{k}{5} \qquad \text{for } k = 1, 2, \ldots$$

or

$$n \log_e \mathbf{c}^T\mathbf{x}^k - \sum_{j=1}^{n} \log_e x_j^k \le n \log_e \mathbf{c}^T\mathbf{x}^0 - \sum_{j=1}^{n} \log_e x_j^0 - \frac{k}{5}$$

Note that $\mathbf{x}^0$ is at the center $\mathbf{e}/n$ of Δ and the function value of $\sum_{j=1}^{n} \log_e x_j$ over Δ is maximized at the center of Δ, hence condition (6.17)

$$n \log_e (\mathbf{c}^T\mathbf{x}^k) \le n \log_e (\mathbf{c}^T\mathbf{x}^0) - \frac{k}{5}$$

is immediately achieved to guarantee the polynomial-time termination of Karmarkar's algorithm.

The remaining work is to show that condition (6.21) holds for an appropriately chosen step-length α in Karmarkar's algorithm. Recall from (6.19a) that

$$f'(\mathbf{y}) = n \log_e (\mathbf{c}^T\mathbf{X}_k\mathbf{y}) - \sum_{j=1}^{n} \log_e y_j$$

We examine its two terms separately. First we show a lemma as follows.

Lemma 6.1. In Karmarkar's algorithm, let

$$\mathbf{y} = \frac{\mathbf{e}}{n} + \frac{\alpha}{n}\left(\frac{\mathbf{d}}{||\mathbf{d}||}\right) \qquad \text{for some } 0 \le \alpha \le 1$$

where

$$\mathbf{d} = -[\mathbf{I} - \mathbf{B}_k^T(\mathbf{B}_k\mathbf{B}_k^T)^{-1}\mathbf{B}_k]\mathbf{X}_k\mathbf{c}$$

then

$$n \log_e (\mathbf{c}^T\mathbf{X}_k\mathbf{y}) \le n \log_e \left(\frac{\mathbf{c}^T\mathbf{X}_k\mathbf{e}}{n}\right) - \alpha \tag{6.22}$$

Proof. Note that the direction vector $\mathbf{d}$ is obtained as the projection of the negative cost vector $-\mathbf{c}^T\mathbf{X}_k$, hence $\mathbf{c}^T\mathbf{X}_k\mathbf{d} = -||\mathbf{d}||^2$. Then we have

$$\mathbf{c}^T\mathbf{X}_k\mathbf{y} = \frac{\mathbf{c}^T\mathbf{X}_k\mathbf{e}}{n} - \frac{\alpha}{n}||\mathbf{d}||$$

Moreover, we define

$$\mathbf{y}(\beta) = \frac{\mathbf{e}}{n} + \beta\frac{\mathbf{d}}{||\mathbf{d}||} \qquad \text{and} \qquad \mathbf{S}'\left(\frac{\mathbf{e}}{n}, \beta\right)$$

to be the spheroid in the transformed space which has a center at $\mathbf{e}/n$ with a radius $\beta \ge 0$. In this way, if we take

$$\beta = R = \sqrt{\frac{n-1}{n}}$$

then $\mathbf{y}(R)$ is the minimizer of the following problem:

$$\begin{aligned} \text{Minimize} \quad & \mathbf{c}^T\mathbf{X}_k\mathbf{y} \\ \text{subject to} \quad & \mathbf{A}\mathbf{X}_k\mathbf{y} = \mathbf{0} \\ & \mathbf{y} \in \mathbf{S}'\left(\frac{\mathbf{e}}{n}, R\right) \end{aligned}$$

which is a relaxation of the problem

$$\begin{aligned} \text{Minimize} \quad & \mathbf{c}^T\mathbf{X}_k\mathbf{y} \\ \text{subject to} \quad & \mathbf{A}\mathbf{X}_k\mathbf{y} = \mathbf{0} \\ & \mathbf{e}^T\mathbf{y} = 1, \quad \mathbf{y} \ge \mathbf{0} \end{aligned}$$

Notice that the latter problem is closely related to problem (6.1). By Karmarkar's second assumption (A2), we know its optimum value is zero. Hence we know the optimal objective value of the relaxed problem is nonpositive and

$$\mathbf{c}^T\mathbf{X}_k\mathbf{y}(R) = \frac{\mathbf{c}^T\mathbf{X}_k\mathbf{e}}{n} - R\|\mathbf{d}\| \le 0$$

This implies that

$$-\|\mathbf{d}\| \le -\frac{1}{R}\frac{\mathbf{c}^T\mathbf{X}_k\mathbf{e}}{n}$$

Since $R = \sqrt{(n-1)/n} < 1$, we further have

$$\mathbf{c}^T\mathbf{X}_k\mathbf{y} = \frac{\mathbf{c}^T\mathbf{X}_k\mathbf{e}}{n} - \frac{\alpha}{n}\|\mathbf{d}\| \le \left(1 - \frac{\alpha}{nR}\right)\frac{\mathbf{c}^T\mathbf{X}_k\mathbf{e}}{n} < \left(1 - \frac{\alpha}{n}\right)\frac{\mathbf{c}^T\mathbf{X}_k\mathbf{e}}{n}$$

Taking logarithms on both sides and using the fact that $\log_e(1-\alpha/n) \le -\alpha/n$, we have the desired result (6.22).

To take care of the other term, $-\sum_{j=1}^{n}\log_e y_j$, in the potential function, we have the following lemma.

Lemma 6.2. If $\mathbf{y} \in S'\left(\frac{\mathbf{e}}{n}, \frac{\alpha}{n}\right)$ then

$$-\sum_{j=1}^{n}\log_e y_j \le -\sum_{j=1}^{n}\log_e\left(\frac{1}{n}\right) + \frac{\alpha^2}{2(1-\alpha)^2} \tag{6.23}$$

Proof. Since

$$\mathbf{y} \in S'\left(\frac{\mathbf{e}}{n}, \frac{\alpha}{n}\right)$$

we know

$$y_j \ge \frac{1}{n} - \frac{\alpha}{n}$$

and hence $ny_j \ge 1 - \alpha$, for $j = 1, 2, \ldots, n$. Taking the Taylor series expansion of $\log_e(1 + (ny_j - 1))$, for each j, there is a μ_j between 1 and ny_j such that

$$\begin{aligned}\log_e(ny_j) &= \log_e(1 + (ny_j - 1)) \\ &= \log_e 1 + (ny_j - 1) - \frac{1}{2\mu_j^2}(ny_j - 1)^2\end{aligned}$$

In other words, we have $\mu_j \ge 1 - \alpha$ such that

$$\log_e(ny_j) \ge (ny_j - 1) - \frac{1}{2(1-\alpha)^2}(ny_j - 1)^2$$

Notice that

$$\sum_{j=1}^{n} ny_j = n\left(\sum_{j=1}^{n} y_j\right) = n$$

and

$$\sum_{j=1}^{n}(ny_j - 1)^2 = \|n\mathbf{y} - \mathbf{e}\|^2 = n^2\|\mathbf{y} - \frac{\mathbf{e}}{n}\|^2 \le \alpha^2$$

therefore

$$\sum_{j=1}^{n}\log_e(ny_j) \ge \frac{-\alpha^2}{2(1-\alpha)^2}$$

and (6.23) follows directly.

Combining (6.22) and (6.23), we see the potential function

$$f'(\mathbf{y}) \le f'\left(\frac{\mathbf{e}}{n}\right) - \alpha + \frac{\alpha^2}{2(1-\alpha)^2} \qquad \text{for appropriate } \alpha$$

In particular, if we choose $\alpha = 1/3$, then

$$f'(\mathbf{y}) \le f'\left(\frac{\mathbf{e}}{n}\right) - 5/24$$

Therefore condition (6.21) is satisfied, and we have the following result as a major theorem for polynomial-time solvability.

Theorem 6.1. Under the assumptions (A1) and (A2), if a step-length is chosen to be $\alpha = 1/3$, then Karmarkar's algorithm stops in $O(nL)$ iterations.

The computational work at each iteration of Karmarkar's algorithm is dominated by inverting the matrix $\mathbf{B}_k\mathbf{B}_k^T$. A simpleminded direct implementation with exact arithmetic requires $O(n^3)$ elementary operations to find the inverse matrix. Hence the total complexity of Karmarkar's algorithm becomes $O(n^4L)$. On the other hand, for finite-precision mathematics, to carry out all computations to the $O(L)$ precision level it requires $O(n^3L)$ bit operations in inverting a matrix, hence Karmarkar's algorithm requires a total of $O(n^4L^2)$ bit operations. However, as shown by N. Karmarkar, using the *rank-one* updating method, the average computation per iteration can be reduced to $O(n^{2.5}L)$ bit operations with $O(L)$ precision. This reduction results in a total of $O(n^{3.5}L^2)$ bit operations. Also note that, although when the step-length is set to be $1/3$ we can achieve the theoretic polynomial-time solvability, in real applications we may use much larger step-length to speed up the convergence. It has been confirmed the new method typically requires only 20 to 50 iterations to provide highly accurate solutions even for very large problems. We shall discuss further implementation issues in Chapter 10.

Note that at each iteration of Karmarkar's algorithm, the current solution always stays in the interior of the feasible domain, even when the algorithm terminates with a solution $\mathbf{x}^k$ such that $\mathbf{c}^T\mathbf{x}^k < 2^{-L}(\mathbf{c}^T\mathbf{e}/n)$. In order to obtain an exact extreme-point optimal solution, we have to further verify the basic and nonbasic solution variables. This can be done by a polynomial-time procedure called the *purification scheme*. The basic idea is quite simple. Looking at problem (6.1), there are $n + m + 1$ constraints (including both explicit and nonnegativity constraints) in total. If n linearly independent constraints are binding at $\mathbf{x}^k$, then it is already a basic feasible solution. Otherwise, we can find a nonzero direction $\mathbf{d}$ in the null space of the binding constraints. If $\mathbf{c}^T\mathbf{d} < 0$, then we move along direction $\mathbf{d}$, otherwise along $-\mathbf{d}$, until some additional constraints become binding for feasibility considerations. Since the feasible domain is bounded, we can always find a new solution with at least one more binding constraint. The objective value of this new solution is obviously at least as good as $\mathbf{c}^T\mathbf{x}^k$. Repeating this process, a basic feasible solution $\mathbf{x}^*$ can eventually be identified such that $\mathbf{c}^T\mathbf{x}^* < 2^{-L}(\mathbf{c}^T\mathbf{e}/n)$.

Since we can begin with the $m+1$ linearly independent explicit constraints, the purification scheme takes at most $n-(m+1)$ steps. Also note that in each step the computational complexity is polynomial, hence the purification scheme is a polynomial-time procedure. An efficient implementation requires a complexity bound of $O(m^2n)$.

It is also worth mentioning that the diagonal elements of the matrix $\mathbf{XA}^T(\mathbf{AX}^2\mathbf{A}^T)^{-1}\mathbf{AX}$ could serve as indicators of optimal basis information. To illustrate this idea, we further define $(\mathbf{M})^+$ to be the generalized inverse of matrix $\mathbf{M}$, DIAG $(\mathbf{M})$ to be a column vector formed by the diagonal elements of matrix $\mathbf{M}$, and $\mathbf{X}$ to be a diagonal matrix with x_i as its ith diagonal element. Also for an $n-$*dimensional* column vector $\mathbf{p}$, we define a new column vector

$$u(\mathbf{p}) = \text{DIAG}\,(\mathbf{X}_p^2\mathbf{A}^T(\mathbf{AX}_p^4\mathbf{A}^T)^+\mathbf{AX}_p^2)$$

In this way, we can consider the following method for locating an optimal extreme-point solution $\mathbf{x}^*$ from an approximated primal solution $\mathbf{x}^k$:

Step 1: Given a small number $\epsilon > 0$, set $j = 0$ and $\mathbf{p}^0 = \mathbf{x}^k$.

Step 2: Increase j by 1, compute $\mathbf{p}^j = u(\mathbf{p}^{j-1})$. Find

$$\mathbf{I}_1 = \{i \mid p_i^j \geq 1-\epsilon, 1 \leq i \leq n\} \quad \text{and} \quad \mathbf{I}_2 = \{i \mid p_i^j \leq \epsilon, 1 \leq i \leq n\}$$

Step 3: If $\mathbf{I}_1 \cup \mathbf{I}_2 = \{1, 2, \ldots, n\}$, then stop. Otherwise, go to Step 2.

It can be shown that, as ϵ goes to zero, if $\mathbf{x}^k$ is sufficiently close to a nondegenerate optimal vertex $\mathbf{x}^*$ of the linear programming problem, then $\{\mathbf{p}^j\}$ converges to a vector $\mathbf{p}^*$ with m ones and $n-m$ zeros with a cubic rate of convergence. In practice, when the above algorithm terminates, we set $x_i^* = 0$ for those $i \in \mathbf{I}_2$, and solve the remaining system of linear equations $\mathbf{Ax}^* = \mathbf{b}$. Further information can be found in the original work of R. Tapia and Y. Zhang.

6.5 CONVERTING TO KARMARKAR'S STANDARD FORM

Consider a standard-form general linear programming problem

$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} \tag{6.24a}$$

$$\text{subject to} \quad \mathbf{Ax} = \mathbf{b} \tag{6.24b}$$

$$\mathbf{x} \geq \mathbf{0} \tag{6.24c}$$

Our objective is to convert this problem into the standard form (6.1) required by Karmarkar, while satisfying the assumptions (A1) and (A2). We shall first see how to convert problem (6.24) into Karmarkar's form and then discuss the two assumptions.

The key feature of Karmarkar's standard form is the simplex structure, which of course results in a bounded feasible domain. Thus we want to *regularize* problem (6.24) by adding a *bounding constraint*

$$\sum_{j=1}^{n} x_j \leq Q$$

for some positive integer Q derived from the feasibility and optimality considerations. In the worst case, we can choose $Q = 2^L$, where L is the problem size. If this constraint is binding at optimality with the objective value of magnitude $-2^{O(L)}$, then we can show that the given problem (6.24) is unbounded.

By introducing a slack variable x_{n+1}, we have a new linear program:

$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} \tag{6.25a}$$

$$\text{subject to} \quad \mathbf{Ax} = \mathbf{b} \tag{6.25b}$$

$$\mathbf{e}^T\mathbf{x} + x_{n+1} = Q \tag{6.25c}$$

$$\mathbf{x} \geq \mathbf{0},\ x_{n+1} \geq 0 \tag{6.25d}$$

In order to keep the matrix structure of $\mathbf{A}$ undisturbed for sparsity manipulation, we introduce a new variable $x_{n+2} = 1$ and rewrite the constraints of problem (6.25) as

$$\mathbf{Ax} - \mathbf{b}x_{n+2} = \mathbf{0} \tag{6.26b}$$

$$\mathbf{e}^T\mathbf{x} + x_{n+1} - Qx_{n+2} = 0 \tag{6.26c}$$

$$\mathbf{e}^T\mathbf{x} + x_{n+1} + x_{n+2} = Q + 1 \tag{6.26d}$$

$$\mathbf{x} \geq \mathbf{0},\ x_{n+1} \geq 0,\ x_{n+2} \geq 0 \tag{6.26e}$$

Note that the constraint $x_{n+2} = 1$ is a direct consequence of (6.26c) and (6.26d). To normalize (6.26d) for the required simplex structure, we apply the transformation $x_j = (Q+1)y_j$, for $j = 1, \ldots, n+2$, to (6.26). In this way, we have an equivalent linear programming problem

$$\text{Minimize} \quad (Q+1)(\mathbf{c}^T\mathbf{y}) \tag{6.27a}$$

$$\text{subject to} \quad \mathbf{Ay} - \mathbf{b}y_{n+2} = \mathbf{0} \tag{6.27b}$$

$$\mathbf{e}^T\mathbf{y} + y_{n+1} - Qy_{n+2} = 0 \tag{6.27c}$$

$$\mathbf{e}^T\mathbf{y} + y_{n+1} + y_{n+2} = 1 \tag{6.27d}$$

$$\mathbf{y} \geq \mathbf{0},\ y_{n+1} \geq 0,\ y_{n+2} \geq 0 \tag{6.27e}$$

Problem (6.27) is now in the standard form required by Karmarkar. In order to satisfy assumption (A1), we may introduce an artificial variable y_{n+3} with a large cost coefficient

M as designed in the big-M method and consider the following problem:

$$\text{Minimize} \quad (Q+1)(\mathbf{c}^T\mathbf{y}) + My_{n+3} \tag{6.28a}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{y} - \mathbf{b}y_{n+2} - [\mathbf{A}\mathbf{e} - \mathbf{b}]y_{n+3} = \mathbf{0}; \tag{6.28b}$$

$$\mathbf{e}^T\mathbf{y} + y_{n+1} - Qy_{n+2} - (n+1-Q)y_{n+3} = 0; \tag{6.28c}$$

$$\mathbf{e}^T\mathbf{y} + y_{n+1} + y_{n+2} + y_{n+3} = 1; \tag{6.28d}$$

$$y_j \geq 0, \quad j = 1, \ldots, n+3 \tag{6.28e}$$

Notice that $\mathbf{y} = \mathbf{e}/(n+3)$ is clearly an initial interior feasible solution to problem (6.28). Moreover, a value M of magnitude $2^{O(L)}$ exists which does not increase the problem size and ensures a zero value of the artificial variable y_{n+3} at optimality, provided that problem (6.27) has a feasible domain.

Taking care of assumption (A2) proposes a more difficult problem for us. It is obvious that not every linear programming problem has a zero optimal objective value. However, if somehow the optimal objective value z^* of a given linear program is known, we can simply subtract z^* from the objective function (6.28a) to get a zero optimal objective value. The real challenge comes from those linear programming problems with unknown optimal objective values. We shall discuss this subject in next section.

6.6 HANDLING PROBLEMS WITH UNKNOWN OPTIMAL OBJECTIVE VALUES

Assumption (A2) requires the optimal objective value of a given linear program to be zero. For those linear programming problems with a known optimal objective value, this assumption can be easily taken care of. But for those with unknown optimal objective values, we have to figure out a process to obtain that piece of information.

Originally, Karmarkar used the so-called *sliding objective function method* to handle the problem. We let z^* be the unknown optimum value of the objective function and pick an arbitrary value $\bar{z}$. Suppose we run Karmarkar's algorithm pretending that $\bar{z}$ is the minimum value of the objective function, i.e., we try to minimize $\mathbf{c}^T\mathbf{x} - \bar{z}$ for the given linear program. We also modify Step 3 of Karmarkar's algorithm as follows:

"After finding $\mathbf{y}^{k+1}$ we check if

$$\frac{\mathbf{c}^T\mathbf{X}_k\mathbf{y}^{k+1}}{\mathbf{e}^T\mathbf{X}_k\mathbf{y}^{k+1}} < \bar{z}$$

If so, we choose a point $\bar{\mathbf{y}}^{k+1}$ on the line segment between $\frac{\mathbf{e}}{n}$ and $\mathbf{y}^{k+1}$ such that

$$\frac{\mathbf{c}^T\mathbf{X}_k\bar{\mathbf{y}}^{k+1}}{\mathbf{e}^T\mathbf{X}_k\bar{\mathbf{y}}^{k+1}} = \bar{z}$$

and assign $\mathbf{x}^{k+1} = \mathbf{T}^{-1}(\bar{\mathbf{y}}^{k+1})$ instead of $\mathbf{T}^{-1}(\mathbf{y}^{k+1})$."

In this way, if $z^* \leq \bar{z}$, then at each iteration of Karmarkar's algorithm, either we obtain a constant reduction (say 1/5 in our case) in the potential function or find a point

that achieves the assumed minimum $\bar{z}$. On the other hand, for $\bar{z} < z^*$, eventually we get a proof that the assumed minimum is lower than the actual minimum by noticing that Karmarkar's iteration is no longer able to produce a constant reduction in the potential function.

With this modification, we can describe the sliding objective function method as follows. Given that a lower bound l and an upper bound u on the objective function are known (otherwise, we can take $l = -2^{O(L)}$ and $u = 2^{O(L)}$ to start with), we further define a tentative lower bound l' and upper bound u' by

$$l' = l + (1/3)(u - l) \tag{6.29}$$

and

$$u' = l + (2/3)(u - l) \tag{6.30}$$

We pretend that l' is the minimum value of the objective function and run the modified algorithm. Karmarkar showed that in a polynomial number of iterations, the algorithm either identifies that l' is lower than the actual minimum or finds a feasible solution with an objective value lower than u'. For suppose l' is not lower than the actual minimum; then the constant reduction in the potential function in each iteration will force $\mathbf{c}^T\mathbf{x}$ to be lower than u'. When l' is found to be too low or u' is too high, we replace l by l' or u by u' correspondingly and rerun the algorithm. Since the range $u - l \geq 0$ shrinks geometrically after each run, we know that in $O(nL)$ runs the range is reduced from $2^{O(L)}$ to $2^{-O(L)}$ and an optimal solution will be identified.

Another way to handle the unknown optimal objective values is to use the information of dual variables. Consider the dual of the linear programming problem (6.1). We have

$$\text{Maximize} \quad z \tag{6.31a}$$

$$\text{subject to} \quad \sum_{i=1}^{m} a_{ij}w_i + z \leq c_j, \; j = 1, 2, \ldots, n \tag{6.31b}$$

$$\mathbf{w} \in R^m, \quad z \in R \tag{6.31c}$$

Notice that the dual problem (6.31) is always feasible, since we can choose any value of $w_1, w_2, \ldots, w_m$ and let

$$z = \min_{j=1,\ldots,n}\left(c_j - \sum_{i=1}^{m} a_{ij}w_i\right) \tag{6.32}$$

such that $(\mathbf{w}, z)$ becomes a feasible solution to problem (6.31). For simplicity, we can write (6.31b) as

$$\mathbf{A}^T\mathbf{w} + z\mathbf{e} \leq \mathbf{c} \tag{6.31b$'$}$$

and write (6.32) as

$$z = \min_j(\mathbf{c} - \mathbf{A}^T\mathbf{w})_j \tag{6.32$'$}$$

If a given linear program (6.1) satisfies assumption (A2), then we know $z \leq 0$ in the dual problem (6.31). Moreover, any dual feasible solution $(\mathbf{w}, z)$ provides a lower bound for the optimal objective value z^* of problem (6.1). One immediate question is, how do we define dual variables associated with each iteration of Karmarkar's algorithm? With this information, then we discuss how to use this dual information to handle problems with unknown optimal objective values.

To get a hint on the definition of dual variables at each iteration, we first consider the form of the dual variables $(\mathbf{w}^*, z^*)$ at optimum. Assume that $\mathbf{x}^*$ is the optimal solution to problem (6.1) and denote matrix $\mathbf{X}^* = \operatorname{diag}(x_1^*, \ldots, x_n^*)$. At optimum, we know $\mathbf{A}^T\mathbf{w}^* \leq \mathbf{c}$. By complementary slackness, we further have $\mathbf{X}^*\mathbf{A}^T\mathbf{w}^* = \mathbf{X}^*\mathbf{c}$. In order to represent $\mathbf{w}^*$ in terms of $\mathbf{x}^*$, we multiply $\mathbf{AX}^*$ on both sides. Hence we have

$$\mathbf{A}(\mathbf{X}^*)^2\mathbf{A}^T\mathbf{w}^* = \mathbf{A}(\mathbf{X}^*)^2\mathbf{c} \tag{6.33}$$

This suggests that we might obtain good dual solutions by defining

$$\mathbf{w}^k = (\mathbf{AX}_k^2\mathbf{A}^T)^{-1}\mathbf{AX}_k^2\mathbf{c} \tag{6.34}$$

and

$$z^k = \min_j(\mathbf{c} - \mathbf{A}^T\mathbf{w}^k)_j \tag{6.35}$$

at each iteration of Karmarkar's algorithm. This is indeed true under the nondegeneracy assumption, owing to the following theorem:

Theorem 6.2. Under the assumptions (A1) and (A2), if the iterates $\{\mathbf{x}^k\}$ defined in Karmarkar's algorithm converge to a nondegenerate basic feasible solution $\mathbf{x}^*$ of problem (6.1), then $\{(\mathbf{w}^k, z^k)\}$ defined by (6.34) and (6.35) converges to an optimal solution of its dual problem (6.31).

Proof. Let $\bar{\mathbf{X}}^*$ be the principal submatrix of $\mathbf{X}^*$ corresponding to the basic variables in $\mathbf{x}^*$ and

$$\begin{bmatrix} \bar{\mathbf{A}} \\ \mathbf{e}^T \end{bmatrix}$$

be the basis matrix of the given linear program corresponding to $\mathbf{x}^*$. Then $\bar{\mathbf{A}}$ has rank m and so does $\bar{\mathbf{A}}\bar{\mathbf{X}}^*$. Hence we know $\bar{\mathbf{A}}(\bar{\mathbf{X}}^*)^2\bar{\mathbf{A}}^T$ is nonsingular. Consequently, $\mathbf{A}(\mathbf{X}^*)^2\mathbf{A}^T = \bar{\mathbf{A}}(\bar{\mathbf{X}}^*)^2\bar{\mathbf{A}}^T$ is nonsingular.

By definition (6.34), we know $(\mathbf{AX}_k^2\mathbf{A}^T)\mathbf{w}^k = \mathbf{AX}_k^2\mathbf{c}$ for $k = 1, 2, \ldots$. Noticing that matrix $(\mathbf{AX}_k^2\mathbf{A}^T)$ converges to the nonsingular matrix $\mathbf{A}(\mathbf{X}^*)^2\mathbf{A}^T$ and vector $\mathbf{AX}_k^2\mathbf{c}$ converges to $\mathbf{A}(\mathbf{X}^*)^2\mathbf{c}$, it follows that $\mathbf{w}^k$ converges to the unique solution $\mathbf{w}^*$ of Equation (6.33). But we already know that the optimal solution to problem (6.31) also satisfies Equation (6.33), hence $\{(\mathbf{w}^k, z^k)\}$ must converge to the optimal dual solution.

The nondegeneracy assumption in Theorem 6.2 is essential to its validity. In order to deal with the general case as well as to handle problems with unknown optimal objective values, Todd and Burrell proposed a new way to define dual variables at each

iteration. Their basic idea is to incorporate dual information $\{(\mathbf{w}^k, z^k)\}$ into Karmarkar's algorithm, with $\{z^k\}$ being monotonically nondecreasing such that z^k can be used as an estimate of the unknown optimum value of the objective function.

Notice that for a primal feasible solution $\mathbf{x}$, $\mathbf{c}^T\mathbf{x} - z^k = \mathbf{c}^T\mathbf{x} - z^k\mathbf{e}^T\mathbf{x} = (\mathbf{c} - z^k\mathbf{e})^T\mathbf{x}$, therefore we define

$$\mathbf{c}(z^k) = \mathbf{c} - z^k\mathbf{e} \tag{6.36}$$

In this way, when z^* is unknown, we can consider replacing $\mathbf{c}$ by $\mathbf{c}(z^k)$ in the objective function at the kth iteration as an estimate. Now, assume that we can modify Karmarkar's algorithm by finding a sequence of feasible solutions $\mathbf{x}^k$, $\mathbf{w}^k$, and z^k such that

$$\mathbf{x}^k \in \mathbf{F} = \{\mathbf{x} \in R^n \mid \mathbf{Ax} = \mathbf{0}, \mathbf{e}^T\mathbf{x} = 1, \mathbf{x} > \mathbf{0}\} \tag{6.37}$$

$$\mathbf{w}^k \in R^m \tag{6.38}$$

$$z^k = \min_j(\mathbf{c} - \mathbf{A}^T\mathbf{w}^k)_j \tag{6.39}$$

$$f(\mathbf{x}^k; \mathbf{c}(z^k)) \leq f(\mathbf{x}^0; \mathbf{c}(z^k)) - \frac{k}{5} \tag{6.40}$$

at each iteration, for $k = 0, 1, \ldots$. Then, before the optimum is reached, we know $z^k \leq z^* < \mathbf{c}^T\mathbf{x}^k$. Moreover, (6.37) and (6.40) directly imply that $\mathbf{c}^T\mathbf{x}^k \leq \mathbf{c}^T\mathbf{x}^0$ and hence

$$\frac{\mathbf{c}^T\mathbf{x}^k - z^*}{\mathbf{c}^T\mathbf{x}^0 - z^*} \leq \frac{\mathbf{c}^T\mathbf{x}^k - z^k}{\mathbf{c}^T\mathbf{x}^0 - z^k}$$

Together with the definition of potential function (6.18) and inequality (6.40), we know that

$$f(\mathbf{x}^k; \mathbf{c}(z^*)) \leq f(\mathbf{x}^0; \mathbf{c}(z^*)) - \frac{k}{5} \tag{6.41}$$

Therefore, the modified algorithm will converge in the same way as Karmarkar's algorithm. The remaining question is how to construct such a sequence of improved solutions.

For $k = 0$, since we know how to take care of assumption (A1), we can choose

$$\mathbf{x}^0 = \frac{\mathbf{e}}{n}, \qquad \mathbf{w}^0 = (\mathbf{AA}^T)^{-1}\mathbf{Ac}$$

and corresponding z^0. Then (6.37)–(6.40) are clearly satisfied. We now are interested in knowing how to find $\mathbf{x}^{k+1}$, $\mathbf{w}^{k+1}$, and z^{k+1} satisfying (6.37)–(6.40), given that we proceed through the kth iteration. Before doing so, we need some notations and a key lemma. First for a $p \times n$ matrix $\mathbf{M}$ with rank p, we denote by $\mathbf{P_M} = \mathbf{I} - \mathbf{M}^T(\mathbf{MM}^T)^{-1}\mathbf{M}$ the projection mapping onto the null space of $\mathbf{M}$, i.e., $\{\mathbf{d} \in R^n \mid \mathbf{Md} = \mathbf{0}\}$. Also denote by

$$\mathbf{P_e} = \mathbf{I} - \frac{\mathbf{ee}^T}{n}$$

the projection mapping onto $\{\mathbf{d} \in R^n \mid \mathbf{e}^T\mathbf{d} = 0\}$. Furthermore, we denote

$$\hat{\mathbf{B}} = \begin{bmatrix} \hat{\mathbf{A}} \\ \mathbf{e}^T \end{bmatrix} \tag{6.42}$$

Suppose that $\hat{\mathbf{A}}$ has full row rank and $\hat{\mathbf{A}}\mathbf{e} = \mathbf{0}$, then $\hat{\mathbf{B}}$ has full row rank and

$$P_{\hat{\mathbf{B}}} = P_{\hat{\mathbf{A}}} P_{\mathbf{e}} = P_{\mathbf{e}} P_{\hat{\mathbf{A}}} \tag{6.43}$$

The key lemma is stated as follows.

Lemma 6.3. In applying the modified Karmarkar's algorithm with a given cost vector $\hat{\mathbf{c}} \in R^n$ and explicit constraint matrix $\hat{\mathbf{A}}$ such that $\hat{\mathbf{A}}\mathbf{e} = \mathbf{0}$, let $\mathbf{d}^k = -P_{\hat{\mathbf{B}}}\hat{\mathbf{c}}$, $\hat{\mathbf{w}} = (\hat{\mathbf{A}}\hat{\mathbf{A}}^T)^{-1}\hat{\mathbf{A}}\hat{\mathbf{c}}$, and $\hat{z} = \min_j(\hat{\mathbf{c}} - \hat{\mathbf{A}}^T\hat{\mathbf{w}})_j$. Then we have

$$\hat{\mathbf{c}}^T\left(\frac{\mathbf{e}}{n} + \left(\frac{\alpha}{n}\right)\frac{\mathbf{d}^k}{\|\mathbf{d}^k\|}\right) \le \left(1 - \frac{\alpha}{n}\right)\left(\frac{\hat{\mathbf{c}}^T\mathbf{e}}{n}\right) + \frac{\alpha}{n}(\hat{z})$$

Proof. Since $\mathbf{d}^k$ is the projection of $-\hat{\mathbf{c}}$, we have $\|\mathbf{d}^k\|^2 = \hat{\mathbf{c}}^T P_{\hat{\mathbf{B}}}\hat{\mathbf{c}} = -\hat{\mathbf{c}}^T\mathbf{d}^k$, and

$$\hat{\mathbf{c}}^T\left(\frac{\mathbf{e}}{n} + \left(\frac{\alpha}{n}\right)\frac{\mathbf{d}^k}{\|\mathbf{d}^k\|}\right) = \frac{\hat{\mathbf{c}}^T\mathbf{e}}{n} - \frac{\alpha}{n}\|\mathbf{d}^k\|$$

Thus it suffices to show that

$$\|\mathbf{d}^k\| \ge \frac{\hat{\mathbf{c}}^T\mathbf{e}}{n} - \hat{z}$$

Notice that

$$\mathbf{d}^k = -P_{\hat{\mathbf{B}}}\hat{\mathbf{c}} = -P_{\mathbf{e}}P_{\hat{\mathbf{A}}}\hat{\mathbf{c}} = -P_{\mathbf{e}}(\hat{\mathbf{c}} - \hat{\mathbf{A}}^T\hat{\mathbf{w}}) = -(\hat{\mathbf{c}} - \hat{\mathbf{A}}^T\hat{\mathbf{w}} - \mathbf{e}\mathbf{e}^T(\hat{\mathbf{c}} - \hat{\mathbf{A}}^T\hat{\mathbf{w}})/n)$$

Since $\hat{\mathbf{A}}\mathbf{e}/n = \mathbf{0}$, we get

$$\mathbf{d}^k = -(\hat{\mathbf{c}} - \hat{\mathbf{A}}^T\hat{\mathbf{w}}) + \left(\frac{\hat{\mathbf{c}}^T\mathbf{e}}{n}\right)\mathbf{e}$$

Also before the optimum is reached,

$$\frac{\hat{\mathbf{c}}^T\mathbf{e}}{n} > \hat{z}$$

For some i, we have

$$\hat{z} = (\hat{\mathbf{c}} - \hat{\mathbf{A}}^T\hat{\mathbf{w}})_i$$

hence

$$d_i^k = \frac{\hat{\mathbf{c}}^T\mathbf{e}}{n} - \hat{z} \ge 0 \quad \text{and} \quad \|\mathbf{d}^k\| \ge |d_i^k| = \frac{\hat{\mathbf{c}}^T\mathbf{e}}{n} - \hat{z}$$

With the help of Lemma 6.3, we show how to find $\mathbf{x}^{k+1}$, $\mathbf{w}^{k+1}$ and z^{k+1} after the kth iteration. Let $\mathbf{w} = (\mathbf{A}\mathbf{X}_k^2\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{X}_k^2\mathbf{c}(z^k)$ and $z = \min_j(\mathbf{c} - \mathbf{A}^T\mathbf{w})_j$. There are two cases, depending upon whether $z \le z^k$.

Case 1. If $z \le z^k$, then z will not be a better estimate than z^k. We shall focus on satisfying (6.37) and (6.40). In this case, since

$$\min_j(\mathbf{c}(z^k) - \mathbf{A}^T\mathbf{w})_j \le 0$$

and $\mathbf{x}^k \in \mathbf{F}$, we have

$$\min_j(\mathbf{X}_k\mathbf{c}(z^k) - \mathbf{X}_k\mathbf{A}^T\mathbf{w})_j \le 0$$

We now apply Lemma 6.3 with $\hat{\mathbf{c}} = \mathbf{X}_k\mathbf{c}(z^k)$, $\hat{\mathbf{A}} = \mathbf{A}\mathbf{X}_k$, and $\hat{\mathbf{B}} = \mathbf{B}_k$. Since the corresponding $\hat{z}$ is nonpositive, this tells that $\hat{\mathbf{c}}^T\mathbf{x}$ can be reduced by a factor of $(1-\alpha/n)$ by taking a step length of α. Thus the potential function $f(\cdot\,; \mathbf{c}(z^k))$ can be reduced by at least 1/5 as before, if we move in the original space along the direction

$$\mathbf{d}^k = -\mathbf{X}_k P_{\mathbf{B}_k}\mathbf{X}_k\left(\mathbf{c} - z^k\mathbf{e}\right)$$

This suggests that we set $\mathbf{w}^{k+1} = \mathbf{w}^k$, $z^{k+1} = z^k$ and move along $\mathbf{d}^k$ for new $\mathbf{x}^{k+1}$, then (6.37)–(6.40) holds for the $(k+1)$th iteration.

Case 2. For $z > z^k$, then $\min_j(\mathbf{c}(z^k) - \mathbf{A}^T\mathbf{w})_j > 0$ and

$$\min_j\left(\mathbf{X}_k\mathbf{c}\left(z^k\right) - \mathbf{X}_k\mathbf{A}^T\mathbf{w}\right)_j > 0. \tag{6.44}$$

Note that

$$\mathbf{X}_k\mathbf{c}\left(z^k\right) - \mathbf{X}_k\mathbf{A}^T\mathbf{w} = P_{\mathbf{A}\mathbf{X}_k}\mathbf{X}_k\mathbf{c}\left(z^k\right) = P_{\mathbf{A}\mathbf{X}_k}\left(\mathbf{X}_k\mathbf{c} - z^k\mathbf{x}^k\right)$$

If we denote $\mathbf{u} = P_{\mathbf{A}\mathbf{X}_k}\mathbf{X}_k\mathbf{c}$ and $\mathbf{v} = P_{\mathbf{A}\mathbf{X}_k}\mathbf{x}^k$, then

$$\mathbf{X}_k\mathbf{c}\left(z^k\right) - \mathbf{X}_k\mathbf{A}^T\mathbf{w} = \mathbf{u} - z^k\mathbf{v}$$

and (6.44) becomes

$$\min_j\left(\mathbf{u} - z^k\mathbf{v}\right)_j > 0.$$

Now let $\bar{z} = \mathbf{c}^T\mathbf{x}^k > z^k$. We see that

$$\left(\mathbf{e}^T(\mathbf{u} - \bar{z}\mathbf{v})P_{\mathbf{A}\mathbf{X}_k}\mathbf{e}\right)^T\left(\mathbf{X}_k\mathbf{c} - \bar{z}\mathbf{x}^k\right) = \mathbf{e}^T\left(\mathbf{X}_k\mathbf{c} - \bar{z}\mathbf{x}^k\right) = \mathbf{c}^T\mathbf{x}^k - \bar{z} = 0$$

Therefore, $\min_j(\mathbf{u} - \bar{z}\mathbf{v})_j \le 0$, since the sum of its components is zero. Consequently, there exists z^{k+1} with $z^k < z^{k+1} \le \bar{z}$ such that

$$\min_j\left(\mathbf{u} - z^{k+1}\mathbf{v}\right)_j = 0$$

In this case, z^{k+1} becomes a better estimate and we can define

$$\mathbf{w}^{k+1} = \left(\mathbf{A}\mathbf{X}_k^2\mathbf{A}^T\right)^{-1}\mathbf{A}\mathbf{X}_k^2\mathbf{c}\left(z^{k+1}\right) \tag{6.45}$$

Note that

$$\min_j\left(\mathbf{X}_k\mathbf{c}\left(z^{k+1}\right) - \mathbf{X}_k\mathbf{A}^T\mathbf{w}^{k+1}\right)_j = \min_j\left(\mathbf{u} - z^{k+1}\mathbf{v}\right)_j = 0 \tag{6.46}$$

Since $\mathbf{x}^k > \mathbf{0}$, we know $\min_j(\mathbf{c}(z^{k+1}) - \mathbf{A}^T\mathbf{w}^{k+1})_j = 0$, and hence

$$\min_j \left(\mathbf{c} - \mathbf{A}^T\mathbf{w}^{k+1}\right)_j = z^{k+1} \tag{6.47}$$

Thus $z^k < z^{k+1} \leq z^*$. Combining (6.40) with the definition of the potential function, we can show that

$$f\left(\mathbf{x}^0; \mathbf{c}\left(z^{k+1}\right)\right) - f\left(\mathbf{x}^k; \mathbf{c}\left(z^{k+1}\right)\right) \geq \frac{k}{5} \tag{6.48}$$

Moreover, from (6.46), we know $\min_j(\mathbf{X}_k\mathbf{c}(z^{k+1}) - \mathbf{X}_k\mathbf{A}^T\mathbf{w}^{k+1})_j = 0$, hence Lemma 6.3 can be applied with $\hat{c} = \mathbf{X}_k\mathbf{c}(z^{k+1})$, $\hat{\mathbf{A}} = \mathbf{A}\mathbf{X}_k$, and $\hat{\mathbf{B}} = \mathbf{B}_k$. Since the corresponding $\hat{z} = 0$, the potential function $f(\cdot\,; \mathbf{c}(z^{k+1}))$ can be reduced by at least $1/5$ as before by moving in the original space along the direction

$$\mathbf{d}^k = -\mathbf{X}_k P_{\mathbf{B}_k}\mathbf{X}_k\left(\mathbf{c} - z^{k+1}\mathbf{e}\right)$$

Combining the analysis of both cases, we state the modified step in Karmarkar's algorithm as follows:

At iteration k with $\mathbf{x}^k$, $\mathbf{w}^k$, and z^k, set $\mathbf{X}_k = \operatorname{diag}(\mathbf{x}^k)$, compute

$$\mathbf{u} = P_{\mathbf{A}\mathbf{X}_k}\mathbf{X}_k\mathbf{c}, \quad \mathbf{v} = P_{\mathbf{A}\mathbf{X}_k}\mathbf{x}^k$$

If $\min_j(\mathbf{u} - z^k\mathbf{v})_j \leq 0$, then set

$$\mathbf{w}^{k+1} = \mathbf{w}^k, \qquad z^{k+1} = z^k$$

Otherwise, find

$$z^{k+1} > z^k \quad \text{with} \quad \min_j(\mathbf{u} - z^{k+1}\mathbf{v})_j = 0$$

and set

$$\mathbf{w}^{k+1} = (\mathbf{A}\mathbf{X}_k^2\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{X}_k^2\mathbf{c}(z^{k+1})$$

Compute $\mathbf{d}^k = -\mathbf{X}_k P_{\mathbf{e}}(\mathbf{u} - z^{k+1}\mathbf{v})$, and

$$\bar{\mathbf{x}}^{k+1} = \mathbf{x}^k + \frac{1}{3n}\frac{\mathbf{d}^k}{||\mathbf{d}^k||}$$

Set

$$\mathbf{x}^{k+1} = \frac{\bar{\mathbf{x}}^{k+1}}{\mathbf{e}^T\bar{\mathbf{x}}^{k+1}}$$

The modified algorithm then generates a sequence $\{\mathbf{x}^k\}$ of primal feasible solutions and a sequence $\{(\mathbf{w}^k, z^k)\}$ of dual solutions such that both $\mathbf{c}^T\mathbf{x}^k$ and z^k converge to the unknown optimal objective value z^*.

6.7 UNCONSTRAINED CONVEX DUAL APPROACH

As pointed out in the previous section, the dual problem of Karmarkar's linear program inherits some interesting properties. In this section, we show that, given an arbitrarily small number $\varepsilon > 0$, an *ε-optimal solution* to a general linear program in Karmarkar's standard form can be found by solving an unconstrained convex programming problem.

Let us focus on the linear programming problem (6.1) and its dual problem (6.31) with an additional assumption that problem (6.1) has a strictly interior feasible solution $\mathbf{x}$ such that $x_j > 0$ for $j = 1, \ldots, n$. We consider the following simple geometric inequality:

$$\sum_{j=1}^{n} e^{y_j} \geq \prod_{j=1}^{n}\left\{\frac{e^{y_j}}{x_j}\right\}^{x_j} \tag{6.49}$$

which holds for arbitrary $y_j \in R$, and $x_j > 0$, $j = 1, 2, \ldots, n$, with

$$\sum_{j=1}^{n} x_j = 1$$

The equality in (6.49) occurs if and only if

$$x_j = \lambda e^{y_j}, \qquad j = 1, 2, \ldots, n \tag{6.50}$$

for a constant $\lambda > 0$. We further expand (6.49) by substituting

$$y_j = \left(\sum_{i=1}^{m} a_{ij}w_i - c_j\right)/\mu, \qquad \text{for } j = 1, 2, \ldots, n \text{ and } \mu > 0$$

Taking logarithms on both sides and rearranging terms, we have

$$\sum_{j=1}^{n}\left(\sum_{i=1}^{m} a_{ij}w_i - c_j\right)x_j \leq \mu\sum_{j=1}^{n} x_j \log_e x_j + \mu\log_e\left\{\sum_{j=1}^{n}\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i - c_j\right)/\mu\right]\right\} \tag{6.51}$$

which holds true for arbitrary $w_i \in R$, $i = 1, 2, \ldots, m$, $x_j > 0$, $j = 1, 2, \ldots, n$ with $\sum_{j=1}^{m} x_j = 1$, and $\mu > 0$. Moreover, inequality (6.51) becomes an equality if and only if:

$$x_j = \lambda\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i - c_j\right)/\mu\right], \qquad j = 1, 2, \ldots, n \tag{6.52}$$

Now, let us assume that the n-vector x also satisfies the constraint (6.1b) of the linear programming problem. Then

$$\sum_{j=1}^{n} a_{ij}x_j = 0, \qquad i = 1, 2, \ldots, m$$

and

$$\sum_{j=1}^{n}\left(\sum_{i=1}^{m} a_{ij}w_i - c_j\right)x_j = \sum_{i=1}^{m} w_i\left(\sum_{j=1}^{n} a_{ij}x_j\right) - \sum_{j=1}^{n} c_jx_j = -\sum_{j=1}^{n} c_jx_j \tag{6.53}$$

Hence, after rearrangement, (6.51) reduces to

$$-\mu \log_e\left\{\sum_{j=1}^{n}\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i - c_j\right)/\mu\right]\right\} \le \mathbf{c}^T\mathbf{x} + \mu\sum_{j=1}^{n} x_j\log_e x_j \tag{6.54}$$

which holds true for arbitrary $w_i \in R$, $i = 1, 2, \ldots, m$, and $x_j > 0$, $j = 1, 2, \ldots, n$, satisfying constraints (6.1b), (6.1c). The equality holds if and only if (6.52) is true. Note that in (6.54), the term $\sum_{j=1}^{n} x_j \log_e x_j$ is usually named the *entropy function* associated with a probability function.

6.7.1 ϵ-Optimal Solution

In nonlinear programming literature, inequality (6.54) is usually referred to as the "weak duality theorem," where the right-hand side is minimized and the left-hand side is maximized. To derive an ε-optimal solution we simply consider the maximization of the left-hand side of (6.54), with respect to unconstrained w_i, $i = 1, 2, \ldots, m$. If we let

$$h(\mathbf{w};\mu) = -\mu\log_e\left\{\sum_{j=1}^{n}\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i - c_j\right)/\mu\right]\right\} \tag{6.55}$$

it can be shown that $h(\mathbf{w};\mu)$ is a strictly concave function of $\mathbf{w}$. Also, under the assumption that there is a feasible interior solution to the linear programming problem (6.1), inequality (6.54) implies that $h(\mathbf{w};\mu)$ is bounded from above. Hence a unique maximum solution $\mathbf{w}^*$ exists.

Taking derivatives of $h(\mathbf{w};\mu)$ at $\mathbf{w}^*$, we have

$$\left\{\sum_{j=1}^{n}\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i^* - c_j\right)/\mu\right]\right\}^{-1}\sum_{j=1}^{n}\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i^* - c_j\right)/\mu\right]a_{ij} = 0,$$

$$i = 1, 2, \ldots, m \tag{6.56}$$

Taking second-order derivatives, we can easily verify that $\mathbf{w}^*$ really achieves the maximum of $h(\mathbf{w};\mu)$ over $\mathbf{w} \in R^n$.

Let us define the n-vector $\mathbf{x}^*$ as follows:

$$x_j^* = \left\{\sum_{j=1}^{n}\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i^* - c_j\right)/\mu\right]\right\}^{-1}\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i^* - c_j\right)/\mu\right],$$

$$j = 1, 2, \ldots, n \tag{6.57}$$

Then, equation (6.56) implies that $\mathbf{x}^*$ satisfies the constraint (6.1b), and equation (6.57) implies that $\mathbf{x}^*$ satisfies the constraints (6.1c). Hence $\mathbf{x}^*$ is a feasible solution to problem (6.1). Moreover, each x_j^* satisfies the condition specified in (6.52) with

$$\lambda = \left(\sum_{j=1}^{n}\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i^* - c_j\right)/\mu\right]\right)^{-1}$$

and hence, (6.54) becomes an equality with $\mathbf{x}$ and $\mathbf{w}$ being replaced by $\mathbf{x}^*$ and $\mathbf{w}^*$, respectively. We summmarize previous results as follows:

Theorem 6.3. Let $\mathbf{w}^*$ be the unique maximum of the concave function $h(\mathbf{w};\mu)$ with $\mu > 0$. If $\mathbf{x}^*$ is defined by Equation (6.57), then

$$h(\mathbf{w}^*;\mu) = -\mu\log_e\left\{\sum_{j=1}^{n}\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i^* - c_j\right)/\mu\right]\right\} = \mathbf{c}^T\mathbf{x}^* + \mu\sum_{j=1}^{n} x_j^*\log_e x_j^* \tag{6.58}$$

Notice that, for $\mathbf{x} \ge \mathbf{0}$ and $\mathbf{e}^T\mathbf{x} = 1$,

$$-\frac{1}{e} \le x_j\log_e x_j \le 0$$

Consequently, we know $h(\mathbf{w}^*;\mu)$ approaches $\mathbf{c}^T\mathbf{x}^*$ as μ goes to 0. Hence, when μ is sufficiently small, we can find a near-optimal solution $\mathbf{x}^*$ to the linear programming problem (6.1) by solving an unconstrained maximization problem of the concave objective function $h(\mathbf{w};\mu)$, or equivalently, minimizing an unconstrained convex function $-h(\mathbf{w};\mu)$. The remaining question is, "How small should μ be such that $\mathbf{x}^*$ obtained by (6.57) is ε-optimal, i.e., $\mathbf{c}^T\mathbf{x}^* - z^* \le \varepsilon$?"

To answer this question, we define

$$z^* = \min_{j=1,\ldots,n}\left\{c_j - \sum_{i=1}^{m} a_{ij}w_i^*\right\} \tag{6.59}$$

It can be easily seen that $(w_1^*, \ldots, w_m^*, z^*)$ is a feasible solution to the dual program (6.31). Without loss of generality, we assume that the minimum of the right-hand side of Equation (6.59) occurs at $j = 1$ and

$$z^* = (c_1 - \sum_{i=1}^{m} a_{i1}w_i^*) \tag{6.60}$$

Taking the logarithm of x_1^* as defined in Equation (6.57), and multiplying the result by μ, we have

$$\mu\log_e x_1^* = \left(\sum_{i=1}^{m} a_{i1}w_i^* - c_1\right) - \mu\log_e\left\{\sum_{j=1}^{n}\exp\left[\left(\sum_{i=1}^{m} a_{ij}w_i^* - c_j\right)/\mu\right]\right\} \tag{6.61}$$

Combining (6.58) and (6.60), we see that

$$\mu \log_e x_1^* = -z^* + \mathbf{c}^T\mathbf{x}^* + \mu \sum_{j=1}^{n} x_j^* \log_e x_j^* \tag{6.62}$$

Moreover, from the theory of linear programming, we know $0 \le \mathbf{c}^T\mathbf{x}^* - z^*$. Moreover,

$$\begin{aligned} 0 \le \mathbf{c}^T\mathbf{x}^* - z^* &= \mu \log_e x_1^* - \mu \sum_{j=1}^{n} x_j^* \log_e x_j^* \\ &= \mu \sum_{j=1}^{n} x_j^* \log_e x_1^* - \mu \sum_{j=1}^{n} x_j^* \log_e x_j^* \\ &= \mu \sum_{j=1}^{n} \log_e \left(\frac{x_1^*}{x_j^*}\right)^{x_j^*} \end{aligned} \tag{6.63}$$

Since $x_j^* > 0$, for $j = 1, 2, \ldots, n$, and

$$\sum_{j=1}^{n} x_j^* = 1$$

considering the geometric inequality again, we have

$$\sum_{j=1}^{n} x_1^* \ge \prod_{j=1}^{n} \left(\frac{x_1^*}{x_j^*}\right)^{x_j^*} \tag{6.64}$$

Since $1 \ge x_1^*$, we have

$$n \ge \prod_{j=1}^{n} \left(\frac{x_1^*}{x_j^*}\right)^{x_j^*} \tag{6.65}$$

Therefore,

$$\sum_{j=1}^{n} \log_e \left(\frac{x_1^*}{x_j^*}\right)^{x_j^*} \le \log_e n \tag{6.66}$$

and Equation (6.63) reduces to

$$0 \le \mathbf{c}^T\mathbf{x}^* - z^* \le \mu \log_e n \tag{6.67}$$

Now for any given tolerance $\varepsilon > 0$, we can define $\mu = \varepsilon / \log_e n$ to guarantee that $0 \le \mathbf{c}^T\mathbf{x}^* - z^* \le \varepsilon$.

Hence we have the following result:

Theorem 6.4. For any given $\varepsilon > 0$, we choose

$$\mu = \frac{\varepsilon}{\log_e n}$$

and let $\mathbf{w}^*$ be the unique minimum of the convex function $-h(\mathbf{w}; \mu)$. If $\mathbf{x}^*$ is defined by Equation (6.57), then

$$0 \le \mathbf{c}^T\mathbf{x}^* - z^* \le \varepsilon \tag{6.68}$$

and $(\mathbf{x}^*; \mathbf{w}^*, z^*)$ becomes an ε-optimal solution pair to the primal problem (6.1) and its dual problem (6.31).

The following example illustrates the unconstrained dual approach to linear programming problems in Karmarkar's standard form.

Example 6.5

$$\begin{aligned} \text{Minimize} \quad & -x_3 \\ \text{subject to} \quad & x_1 - x_2 = 0 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

It is easy to see that $(0, 0, 1)$ is the optimal solution. In this case, we have a corresponding unconstrained convex programming problem:

$$\begin{aligned} \text{Minimize} \quad & \mu \log_e \{\exp[z/\mu] + \exp[-z/\mu] + \exp[1/\mu]\} \\ \text{subject to} \quad & z \in R \end{aligned}$$

Taking its first-order necessary condition, we see $z^* = 0$. Also by (6.57), we have

$$x_1^* = x_2^* = \frac{1}{(1 + 1 + \exp[1/\mu])}, \qquad x_3^* = \frac{\exp[1/\mu]}{(1 + 1 + \exp[1/\mu])}$$

Therefore, both x_1^* and x_2^* decrease to 0 and x_3^* increases to 1 as μ decreases to 0.

The unconstrained convex programming approach allows us to have a different view of the linear programming problems. The potential of customizing different unconstrained optimization techniques, including the descent methods, conjugate direction methods, and quasi-Newton methods, for finding an ε-optimal solution to the linear programming problem is certainly worthy of further exploration.

6.7.2 Extension

The work in the previous section actually suggests us to consider a perturbed problem (P_μ):

$$\begin{aligned} \text{Minimize} \quad & \mathbf{c}^T\mathbf{x} + \mu \sum_{j=1}^{n} x_j \log_e x_j \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{0} \\ & \mathbf{e}^T\mathbf{x} = 1 \\ & \mathbf{x} \ge \mathbf{0} \end{aligned}$$

and its unconstrained convex dual program (D_μ):

$$\text{Maximize} \quad h(\mathbf{w};\mu) = -\mu \log_e \left\{ \sum_{j=1}^{n} \exp\left[\left(\sum_{i=1}^{n} a_{ij} w_i - c_j \right) / \mu \right] \right\}$$

subject to $\mathbf{w} \in R^m$

Under the assumption that problem (6.1), and hence (P_μ), has an interior feasible solution, there is no duality gap between problems (P_μ) and (D_μ). Moreover, when $\mathbf{A}$ has full row rank, for any given tolerance $\varepsilon > 0$, by choosing

$$\mu = \frac{\varepsilon}{\log_e n}$$

the optimal solution $\mathbf{w}^*$ of problem D_μ generates a primal feasible solution $\mathbf{x}^*$ of problem (6.1), according to Equation (6.57), and a dual feasible solution ($\mathbf{w}^*, z^*$) of problem (6.31), according to Equation (6.59), such that $|\mathbf{c}^T\mathbf{x}^* - z^*| \leq \varepsilon$.

For a linear programming problem in its standard form, we consider a corresponding problem (P'_μ):

$$\text{Minimize} \quad \mathbf{c}^T\mathbf{x} + \mu \sum_{j=1}^{n} x_j \log_e x_j$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{x} > \mathbf{0}$$

Replacing the inequality (6.19) by the following one:

$$\log_e t \leq t - 1, \qquad \text{for } t > 0 \tag{6.69}$$

and following a similar derivation procedure with

$$t_j = \frac{\exp\left(\left[\left(\sum_{i=1}^{m} a_{ij} w_i - c_j\right) / \mu\right] - 1\right)}{x_j}, \qquad \text{for } j = 1, 2, \ldots, n \tag{6.70}$$

we can construct an unconstrained concave program (D'_μ):

$$\text{Maximize} \quad h'(\mathbf{w};\mu) = \sum_{i=1}^{m} b_i w_i - \mu \left\{ \sum_{j=1}^{n} \exp\left(\left[\left(\sum_{i=1}^{n} a_{ij} w_i - c_j \right) / \mu \right] - 1 \right) \right\}$$

subject to $\mathbf{w} \in R^m$

With an additional assumption that problem (P'_μ) has a bounded feasible domain, a sufficiently small μ can be determined such that the optimal solution $\mathbf{w}^*$ of problem (D'_μ) generates an ε-optimal solution $\mathbf{x}^*$ to the original linear programming problem in

standard form according to the following conversion formula:

$$x_j^* = \exp\left(\left[\left(\sum_{i=1}^{m} a_{ij} w_i^* - c_j \right) / \mu \right] - 1 \right), \qquad \text{for } j = 1, 2, \ldots, n \tag{6.71}$$

6.8 CONCLUDING REMARKS

Karmarkar's projective scaling algorithm has stimulated a great amount of research interest in linear programming. Since the work was introduced in 1984, many variants have been proposed and many more are to come. The fundamental difference between Karmarkar's algorithm and simplex methods is the philosophy of moving in the interior versus moving on the boundary of the polytope. It is not true that Karmarkar-based interior-point methods are going to replace the simplex methods, at least in the foreseeable future. Both approaches are very sensitive to the structure of problems. The performance is heavily affected by the sophistication of implementation. A hybrid method of using the interior approach at the beginning for drastic reduction and shifting to the simplex method for a final basic feasible solution seems attractive. We shall study the interior-point methods further and discuss implementation issues in coming chapters.

REFERENCES FOR FURTHER READING

6.1. Anstreicher, K. M., "A combined phase I - phase II projective algorithm for linear programming," *Mathematical Programming* 43, 209–223 (1989).

6.2. Anstreicher, K. M., "On the performance of Karmarkar's algorithm over a sequence of iterations," *SIAM Journal on Optimization* 1, 22–29 (1991).

6.3. Bayer, D., and Lagarias, J. C., "Karmarkar's linear programming algorithm and Newton's method," *Mathematical Programming* 50, 291–330 (1991).

6.4. Fang, S. C., "A new unconstrained convex programming approach to linear programming," OR Report No. 243, North Carolina State University, Raleigh, NC (1990), *Zeischrift für Operations Research* 36, 149–161 (1992).

6.5. Fang, S. C., and Tsao, J. H-S., "Solving standard form linear programs via unconstrained convex programming approach with a quadratically convergent global algorithm," OR Report No. 259, North Carolina State University, Raleigh, NC (1991).

6.6. Gay, D., "A variant of Karmarkar's linear programming algorithm for problems in standard form," *Mathematical Programming* 37, 81–90 (1987).

6.7. Hooker, J. N., "Karmarkar's linear programming algorithm," *Interfaces* 16, 75–90 (1986).

6.8. Karmarkar, N., "A new polynomial time algorithm for linear programming," *Proceedings of the 16th Annual ACM Symposium on the Theory of Computing*, 302–311 (1984).

6.9. Karmarkar, N., "A new polynomial time algorithm for linear programming," *Combinatorica* 4, 373–395 (1984).

6.10. Kojima, M., "Determining basic variables of optimal solutions in Karmarkar's new LP algorithm," *Algorithmica* 1, 499–515 (1986).

6.11. Kortanek, K. O., and Zhu, J., "New purification algorithms for linear programming," *Naval Research Logistics* 35, 571–583 (1988).

6.12. Monteiro, R. C., "Convergence and boundary behavior of the projective scaling trajectories for linear programming," *Mathematics of Operations Research* 16, No. 4 (1991).

6.13. Rajasekera, J. R., and Fang, S. C., "On the convex programming approach to linear programming," *Operations Research Letters* 10, 309–312 (1991).

6.14. Shanno, D. F., "Computing Karmarkar projection quickly," *Mathematical Programming* 41, 61–71 (1988).

6.15. Shub, M., "On the asymptotic behavior of the projective rescaling algorithm for linear programming," *Journal of Complexity* 3, 258–269 (1987).

6.16. Stone, R. E., and Tovey, C. A., "The simplex and projective scaling algorithm as iteratively reweighted least squares methods," *SIAM Review* 33, 220–237 (1991).

6.17. Tapia, R. A., and Zhang, Y., "Cubically convergent method for locating a nearby vertex in linear programming," *Journal of Optimization Theory and Applications* 67, 217–225 (1990).

6.18. Tapia, R. A., and Zhang, Y., "An optimal-basis identification technique for interior-point linear programming algorithms," *Linear Algebra and Its Applications*, 152, 343–363 (1991).

6.19. Todd, M. J., and Burrell, B. P., "An extension to Karmarkar's algorithm for linear programming using dual variables," *Algorithmica* 1, 409–424 (1986).

6.20. Todd, M. J., and Ye, Y., "A centered projective algorithm for linear programming," *Mathematics of Operations Research* 15, 508–529 (1990).

6.21. Vanderbei, R. J., "Karmarkar's algorithm and problems with free variables," *Mathematical Programming* 43, 31–44 (1989).

6.22. Ye, Y., "Karmarkar's algorithm and the ellipsoidal method," *Operations Research Letters* 4, 177–182 (1987).

6.23. Ye, Y., "Recovering optimal basic variables in Karmarkar's polynomial algorithm for linear programming," *Mathematics of Operations Research* 15, 564–572 (1990).

6.24. Ye, Y., and Kojima, M., "Recovering optimal dual solutions in Karmarkar's polynomial algorithm for linear programming," *Mathematical Programming* 39, 305–317 (1987).

EXERCISES

6.1. Focus on the n-dimensional Euclidean space.

(a) For a given point $\mathbf{x} \in \Delta$, looking at its coordinates, how can we identify it is a vertex of Δ? on an edge of Δ? in the interior of Δ? at the center of Δ?

(b) From (a) prove that Δ has n vertices and $C(n, 2)$ edges.

(c) Show that the distance between the center and any vertex of Δ is given by

$$R = \frac{\sqrt{n-1}}{\sqrt{n}}$$

and the distance between the center and any facet of Δ is given by

$$r = \frac{1}{\sqrt{n(n-1)}}$$

6.2. For a projective transformation $T_{\bar{\mathbf{x}}}$, prove results (T1) through (T6). What can one say about its inverse transformation?

6.3. Does the projective transformation $T_{\bar{\mathbf{x}}}$ map a line segment in Δ to a line segment? Why?

6.4. Why is $\mathbf{x}(\alpha)$ in Equation (6.11) an interior feasible solution to problem (6.1)? Prove it.

6.5. Show that if matrix $\mathbf{A}$ in Equation (6.9) has full rank, then the matrix $\mathbf{B}\mathbf{B}^T$ is invertible and hence the direction $\bar{\mathbf{d}}$ in Equation (6.12) is well defined.

6.6. Carry out one more iteration of Example 6.4. Is it closer to the optimal solution?

6.7. Show that the function

$$\sum_{j=1}^{n} \log_e x_j$$

achieves its maximum value at

$$\mathbf{x}^* = \frac{\mathbf{e}}{n} \quad \text{for} \quad \mathbf{x} \in \Delta$$

6.8. Apply Karmarkar's algorithm to solve Example 6.2.

6.9. Convert the linear programming problems in Exercise 3.16 into Karmarkar's standard form satisfying Assumption (A1).

6.10. In solving problem (6.25) with $Q = 2^L$, if (6.25c) becomes a binding constraint at optimality with the objective value of magnitude $-2^{O(L)}$, show that problem (6.24) is unbounded.

6.11. Show how the inequality $\mathbf{c}^T\mathbf{x}^k \le \mathbf{c}^T\mathbf{x}^0$ is implied by (6.37) and (6.40).

6.12. Show that (6.43) is true under the assumptions that $\hat{\mathbf{A}}$ has full row rank and $\hat{\mathbf{A}}\mathbf{e} = \mathbf{0}$.

6.13. Consider $h(\mathbf{w}; \mu)$ as defined by (6.55).

(a) Find its gradient vector $\nabla h(\mathbf{w}; \mu)$.

(b) Find its Hessian matrix $H(\mathbf{w}; \mu)$.

(c) Show that $H(\mathbf{w}; \mu) = \mathbf{A}\mathbf{D}\mathbf{A}^T$ for a special diagonal matrix $\mathbf{D}$ with negative diagonal elements.

(d) When $\mathbf{A}$ is assumed to be of full row rank, show that $H(\mathbf{w}; \mu)$ is symmetric, negative definite.

(e) Conclude that $h(\mathbf{w}; \mu)$ is a strictly concave function of $\mathbf{w}$.

6.14. Derive the dual objective function $h(\mathbf{w}; \mu)$ for Example 6.2.

6.15. Code Karmarkar's algorithm and test the linear programming problems of Exercise 6.9.