Chapter 11

NEW LINEAR PROGRAMMING ALGORITHMS, AND SOME OPEN PROBLEMS IN LINEAR COMPLEMENTARITY

Some open research problems in linear complementarity have already been posed among the exercises in previous chapters. Here we discuss some more research problems briefly.

11.1 Classification of a Given Square Matrix M

Let M be a given square matrix of order n. In Section 1.3.1 we discussed algorithms to check whether M is PD os PSD, requiring a computational effort of at most n Gaussian pivot steps, or $\mathcal{O}(n^3)$ effort in terms of multiplications and additions. Such efficient algorithms are not known to check whether M belongs to other classes of matrices discussed in Chapters 2, 3.

As an example, consider the problem of checking whether M is a non-degenerate (i. e., principally non-degenerate to be specific) matrix. The question is: given M, to find whether there exists a subset of $\{1, \ldots, n\}$ such that the principal subdeterminant of M corresponding to that subset is zero. Since this question involves the existence of a subset of $\{1, \ldots, n\}$ satisfying a specified property which is easily checked (given a subset $\mathbf{J} \subset \{1, \ldots, n\}$, we can check whether $\mathbf{J}$ satisfies this property by computing the subdeterminant of M corresponding to $\mathbf{J}$, which takes at most $\mathcal{O}(r^3)$ effort, $r = |\mathbf{J}|$), this problem is in $\mathcal{NP}$, the class of decision problems which can be solved by a polynomially bounded non-deterministic algorithm (see M. Garey and D. Johnson's book [8.12] for precise definitions of these terms). We will now show that this problem is in fact $\mathcal{NP}$-complete. Given positive integers $d_0; d_1, \ldots, d_n$, the problem of checking

whether there exists a subset of $\{d_1, \ldots, d_n\}$ whose sum is equal to d_0, known as the **subset sum problem**, is the 0-1 problem of checking whether the following system has a solution

$$\begin{aligned} &\sum_{j=1}^{n} d_j x_j = d_0 \\ &x_j = 0 \text{ or } 1 \text{ for all } j \ . \end{aligned} \tag{11.1}$$

This problem is a well-known $\mathcal{NP}$-complete problem. Define M to be the matrix

$$M = \begin{pmatrix} d_0 & d_1 & d_2 & d_3 & \ldots & d_n \\ 1 & 1 & 0 & 0 & \ldots & 0 \\ 1 & 0 & 1 & 0 & \ldots & 0 \\ 1 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \ldots & 1 \end{pmatrix} = \begin{pmatrix} d_0 & d \\ e & I_n \end{pmatrix}$$

where $d = (d_1, \ldots, d_n)$, e is the column vector of all 1's in $\mathbf{R}^n$, and I_n is the unit matrix of order n. A principal submatrix of M corresponding to a non-empty subset of $\{1, \ldots, n+1\}$ not containing 1 is a unit matrix of appropriate order, and hence has determinant 1. The principal subdeterminant of M corresponding to a subset of $\{1, \ldots, n+1\}$ of the form $\{1, i_1, \ldots, i_r\}$ can be verified to be $d_0 - (d_{i_1} + \ldots + d_{i_r})$. Thus the matrix M given above has a zero principal subdeterminant iff the system (11.1) has a solution. Since the $\mathcal{NP}$-complete problem (11.1) is a special case of the problem of checking whether a given square matrix has zero principal subdeterminant, this later problem is also an $\mathcal{NP}$-complete problem. This result is from [11.1] of R. Chandrasekaran, S. N. Kabadi and K. G. Murty.

The computational complexity of checking whether a given square matrix M is a P-matrix, P_0-matrix, Q-matrix, or Q_0-matrix is not known. For all these problems, finite algorithms are known. P- and P_0-properties can be checked by computing all the principal subdeterminants (requiring the evaluation of $2^{\mathbf{n}}$ determinantes when M is of order n). Finite algorithms for checking the Q- and Q_0-properties are provided in Exercises 3.87, 3.89 (when applied on a matrix of order n, these methods require the solution of at most $n^{\mathbf{2^n}}$ systems of linear inequalities, hence these methods though finite, are utterly impractical even for $n = 4$). No polynomially bounded algorithms for any of these problems are known so far, and it is also not known whether any of these problems is $\mathcal{NP}$-complete.

11.2 Worst Case Computational Complexity of Algorithms

In Chapter 6 we established that several of the pivotal algorithms for LCP are exponential growth algorithms in the worst case. However, the worst case computational complexity of the algorithm for solving the LCP (q, M) when M is PD symmetric matrix (or the corresponding nearest point problem) based on orthogonal projections discussed in Chapter 7 is still an open question.

11.2.1 Computational Complexity of the LCP Associated with a P-Matrix

In Chapter 8 we discussed polynomially bounded algorithms for the LCP (q, M) when M is either a Z-matrix, or a principally triangular P-matrix, or a PSD-matrix. The polynomially bounded ellipsoid methods work only when M is PSD, since they depend on the fact that the set $\{z : z^T(Mz + q) \leqq 0\}$ is convex, which may not hold when M is not PSD. None of the methods discussed in Chapter 8 are guaranteed to process the LCP (q, M) when M is a P-matrix which is not PSD. In this case the set $\{z : z^T(Mz + q) \leqq 0\}$ may not be convex. When M is a P-matrix, by the results in Chapter 3, the LCP (q, M) has the nice property of having a unique solution, but as yet no polynomially bounded algorithm is known for computing it. Establishing whether the LCP (q, M), where M is a P-matrix, can be solved by a polynomially bounded algorithm, remains an important mathematical problem in LCP theory.

11.2.2 A Principal Pivoting Descent Algorithm For the LCP Associated with a P-Matrix

In the LCP there is of course no objective function. In this algorithm from K. G. Murty [3.52] an extraneous distance function is computed and this distance decreases strictly in each step. The distance provides a measure of progress in the algorithm, it becomes zero iff we obtain a complementary feasible basis. The algorithm is a principal pivoting algorithm employing only single principal pivot steps, it can be used to solve the LCP (q, M) when M is a P-matrix. The algorithm can be initiated with any complementary basis. We now describe the algorithm.

Let A be the current complementary basis and y the corresponding complementary basic vector.

Find the nearest point in $\text{Pos}(A)$ to q in terms of the usual Euclidean distance (this can be found in polynomial time by the ellipsoid algorithm discussed in Section 8.4, or by the practically efficient algorithm discussed in Chapter 7). Let $\bar{x}$ be this nearest point and $d = ||\bar{x} - q||$, the Euclidean distance between $\bar{x}$ and q.

We will have $d = 0$ and $\bar{x} = q$ iff $q \in \text{Pos}(A)$. In this case y is a complementary feasible basic vector, and the solution of the LCP (q, M) is $(y = A^{-1}q, t = 0)$, where $t = (t_j)$ and t_j is the complement of y_j for all j.

If $d > 0$, let $\mathbf{B}(q, d) = \{x : ||x - q|| \leqq d\}$. $\mathbf{B}(q, d)$ is the closed ball with q as center and d as radius. Let $\mathbf{T}(q, \bar{x}) = \{x : (q - \bar{x})^T(x - \bar{x}) = 0\}$, it is the tangent hyperplane to $\mathbf{B}(q, d)$ at its boundary point $\bar{x}$. Since $\bar{x}$ is the nearest point in $\text{Pos}(A)$ to q, by the results in Chapter 7, $\bar{x}^T(q - \bar{x}) = 0$, $\mathbf{T}(q, \bar{x}) = \{x : x^T(q - \bar{x}) = 0\}$, it is a hyperplane containing the origin, 0. Since $\bar{x} \in \text{Pos}(A)$, we have $\bar{x} = \sum_{j=1}^{n} \alpha_j A_{.j}$ where $\alpha_j \geqq 0$ for all j. Let $\mathbf{J} = \{j : \alpha_j > 0\}$, $\bar{\mathbf{J}} = \{1, \ldots, n\} \setminus \mathbf{J}$. In this case since $q \notin \text{Pos}(A)$, by the results in Chapter 7, $\bar{x}$ must be a boundary point of $\text{Pos}(A)$, so $\bar{\mathbf{J}} \neq \emptyset$. For each

j let $D_{.j}$ be the complement of $A_{.j}$. By the results in Chapter 7, $\bar{x}$ is the orthogonal projection of q in the linear hull of $\{A_{.j} : j \in \mathbf{J}\}$, so the tangent hyperplane $\mathbf{T}(q, \bar{x})$ contains the linear hull of $\{A_{.j} : j \in \mathbf{J}\}$. By Theorem 3.20 of Section 3.3, $\mathbf{T}(q, \bar{x})$ must separate strictly, at least one of the pair of column vectors $\{A_{.j}, D_{.j}\}$ for some $j \in \bar{\mathbf{J}}$. Let $\mathbf{\Delta} = \{j : j \in \bar{\mathbf{J}}, A_{.j}$ and its complement are strictly separated by $\mathbf{T}(q, \bar{x})\}$. So $\mathbf{\Delta} \neq \emptyset$, select a $p \in \mathbf{\Delta}$ arbitrarily. Then in the notation of Chapter 7, $D_{.p}$ is on the near side of $\mathbf{T}(q, \bar{x})$, and $\text{Pos}\{\bar{x}, D_{.p}\}$ contains points which are strictly closer to q than $\bar{x}$. Thus if we make a single principal pivot step in position p in the complementary basis A, we get a new complementary basis whose pos cone contains points strictly nearer than $\bar{x}$ to q.

With $(y_1, \ldots, y_{p-1}, t_p, y_{p+1}, \ldots, y_n)$ as the new complementary basic vector, we repeat the whole process.

After each principal pivot step, the distance d strictly decreases, so a complementary basic vector can never reappear in the algorithm. Since there are only 2^n complementary basic vectors, the method must terminate after a finite number of principal pivot steps with the complementary solution for the problem.

Since the problem of finding the nearest point in a complementary cone which has a non-empty interior, to q, is equivalent to another LCP associated with a PD symmetric matrix, the method can be viewed as one for solving the LCP (q, M) associated with a P-matrix M by solving a finite number of LCP's associated with PD symmetric matrices.

The worst case computational complexity of this algorithm is still an open question.

One can get different variants of the algorithm by choosing p from $\mathbf{\Delta}$ according to different rules. One can consider the least index rule in which the p chosen from $\mathbf{\Delta}$ is always the least; or a cyclical rule like the least recently considered rule popular in implementations of the simplex algorithm. We can also consider a block principal pivoting method in which the new complementary basic vector at the end of the step is obtained by replacing each y_p in the present complementary basic vector, by its complement for each $p \in \mathbf{\Delta}$, in a block principal pivot step. The worst case computational complexity of each of these variants is currently under investigation.

Exercise

11.1 The rectilinear or L_1-distance between two points $x = (x_j)$, $y = (y_j)$ in $\mathbf{R}^n$ is defined to be $\sum_{j=1}^{n}(|x_j - y_j|)$. Consider the LCP (q, M) with M being a P-matrix. Let $y = (y_j)$ be a complementary basic vector for this problem associated with the complementary basis A. The nearest point in the complementary cone $\text{Pos}(A)$ to q in terms of the L_1-distance can be obtained by solving the LP

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{n}(u_i + v_i) \\ \text{subject to} \quad & Ay + u - v = q \\ & y, u, v \geqq 0 \ . \end{aligned}$$

If $(\bar{y}, \bar{u}, \bar{v})$ is an optimum solution to this LP, $\bar{x} = A\bar{y}$ is a nearest point in Pos(A) to q in terms of the L_1-distance.

If M is a P-matrix and $q \notin$ Pos(A), does there always exist a p such that the cone Pos $\{A_{.1}, \ldots, A_{.p-1}, D_{.p}, A_{.p+1}, \ldots, A_{.n}\}$, where $D_{.p}$ is the complement of $A_{.p}$, contains points which are strictly closer to q in terms of the L_1-distance, than $\bar{x}$? If so, discuss an efficient method for identifying such a p.

Develop a method for solving the LCP (q, M) when M is a P-matrix, that moves from one complementary cone to another, decreasing the L_1-distance to q in each step. Study the worst case computational complexity of this method.

11.3 Alternate Solutions of the LCP (q,M)

There are very nice conditions to check the uniqueness of a given solution for a linear programming problem, and to characterize and enumerate alternative optimum solutions when they exist. See [2.26].

For LCP, such characterizations or methods do not exist yet. A sufficient condition for the uniqueness of the solution for the LCP (q, M) is that M be a P-matrix. When M is not a P-matrix, alternate solutions may exist for the LCP (q, M), but in this case the algorithms discussed for the LCP find only one solution for the problem if they are able to process it, and then terminate.

Consider the LCP (q, M). Let $y = (y_j)$ be a complementary vector of variables for it, that is, for each j, $y_j \in \{w_j, z_j\}$. Let A be the complementary matrix corresponding to y. Let $t = (t_j)$ where t_j is the complement of y_j for each j. The complementary vector y leads to a solution of the LCP (q, M) iff the system

$$\begin{aligned} Ay &= q \\ y &\geqq 0 \end{aligned}$$

has a feasible solution. If $\bar{y}$ is a feasible solution of this system, $(y = \bar{y}, t = 0)$ is a solution of the LCP (q, M). If A is nonsingular, the above system has a feasible solution iff $A^{-1}q \geqq 0$, and in this case if it does have a solution, it is unique. If A is singular, the above system may have many feasible solutions. Whether it has a feasible solution or not can be determined by using Phase I of the simplex method for linear programming. If the above system is feasible, all alternate feasible solutions of it can be enumerated and the set of alternate feasible solutions compactly represented using standard results in linear programming [2.26], each such feasible solution leads to a solution of the LCP (q, M), as discussed above.

By solving the system of the type discussed above, for each of the complementary vectors of variables y, we can check whether the LCP (q, M) has a solution, and in fact

obtain all its solutions. This is a total enumeration method, requiring the solution of $2^{\mathbf{n}}$ separate systems of linear equations in non-negative variables.

Since $2^{\mathbf{n}}$ grows rapidly, the above total enumeration method for checking whether alternate solutions exist for a given LCP, or to obtain all solutions of it, is impractical unless n is very small. It would be nice if some efficient partial enumeration methods can be developed for doing the same job. These partial enumeration methods should identify subsets of complementary vectors of variables which do not lead to a solution of the LCP, and prune them, thereby saving some of the effort needed to carry out the enumeration. These methods would be similar to branch and bounds for 0-1 integer programming problems (see [1.28]) which are also partial enumeration methods.

We will now describe briefly one partial enumeration method for generating all the solutions of the LCP (q, M) discussed in K. G. Murty [11.3]. To keep the discussion simple, we make the assumption that q is nondegenerate. In this case, every complementary solution is a complementary BFS and it is adequate to enumerate among complementary basic vectors for all complementary solutions of the LCP (q, M).

The set of all variables in the LCP (q, M) is $\{w_1, \ldots, w_n; z_1, \ldots, z_n\}$. Given any subset $\mathbf{\Delta}$ of these variables, we will represent $\mathbf{\Delta}$ by a 0-1 incidence vector $a = (a_p) \in \mathbf{R}^{2n}$, a row vector, where

$$\text{for } j = 1 \text{ to } n, \quad a_j = \begin{cases} 1, & \text{if } w_j \in \mathbf{\Delta} \\ 0, & \text{if } w_j \notin \mathbf{\Delta} \end{cases}$$

$$a_{n+j} = \begin{cases} 1, & \text{if } z_j \in \mathbf{\Delta} \\ 0, & \text{if } z_j \notin \mathbf{\Delta} \end{cases}.$$

As an example, for $n = 4$, the incidence vector of the subcomplementary set $\{z_1, w_2, z_3\}$ is $(0, 1, 0, 0; 1, 0, 1, 0)$. So a complementary feasible basic vector for the LCP (q, M) corresponds to an incidence vector $x = (x_p) \in \mathbf{R}^{2n}$ satisfying $\sum_{p=1}^{2n} x_p = n$ and $x_j + x_{n+j} \geqq 1$, for each $j = 1$ to n, and the vector is a feasible basic vector, $x_p = 0$ or 1 for all $p = 1$ to $2n$. The second constraint that the vector be a feasible basic vector is not available explicitly in the form of a system of linear constraints, at the beginning; but we develop linear constraints in the x_p-variables corresponding to it during the course of the algorithm. In each step, more constraints of this type in the x_p-variables are generated and augmented to the system.

A set covering problem is a 0-1 integer programming problem of the following form.

$$\begin{aligned} \text{minimize} \quad & \sum_{p=1}^{2n} x_p \\ \text{subject to} \quad & Ex \geqq e_r \\ & x_p = 0 \text{ or } 1 \text{ for all } p \end{aligned}$$

where E is a 0-1 matrix of order $r \times 2n$ and e_r in the column vector of all 1's in $\mathbf{R}^r$. In each step, we solve a set covering problem of this form, and generate additional constraints for the set covering problem in the next step.

The set covering problem itself is an $\mathcal{NP}$-hard combinatorial optimization problem, but practically efficient branch and bound algorithms are available for it. The branch

and bound algorithm discussed in [1.28] for the set covering problem using the lower bounding strategy based on Lagrangian Relaxation may be particularly suitable, since we have to solve the problem repeatedly, with the only change between the problem in one step and the next being a few additional constraints.

A solution stack is maintained. Any solution to the LCP (q, M) found out during the algorithm is stored in the solution stack. At termination of the algorithm, this stack contains all the solutions of the LCP (q, M).

The initial set covering problem is

$$\begin{aligned} \text{minimize} \quad & \sum_{p=1}^{2n} x_p \\ \text{subject to} \quad & x_j + x_{n+j} \geqq 1, \text{ for each } j = 1 \text{ to } n \\ & x_p = 0 \text{ or } 1, \text{ for } p = 1 \text{ to } 2n \end{aligned}$$

The initial complementary basic vector is w. We will now describe a general step in the algorithm.

General Step

Let $y = (y_j)$ be the current complementary vector of variables with $y_j \in \{w_j, z_j\}$ for each $j = 1$ to n, and let A be the corresponding complementary matrix. Let $t = (t_j)$ where t_j is the complement of y_j for each $j = 1$ to n.

If A is singular, every complementary basic vector must include one of the variables from $\{t_1, \ldots, t_n\}$. Let $a \in \mathbf{R}^{2n}$ be the incidence vector of $\{t_1, \ldots, t_n\}$. Add the additional constraint "$ax \geqq 1$" to the set covering problem.

If A is nonsingular, y is a complementary basic vector, obtain the canonical tableau of the LCP (q, M) with respect to it. Suppose it is

y	t	
I	$-D$	$\bar{q}$

If $\bar{q} \geqq 0$, y is a complementary feasible basic vector, and $(y = \bar{q}, t = 0)$ is the corresponding complementary solution, include it in the stack. Every complementary basic vector different from y must include one of the variables from $\{t_1, \ldots, t_n\}$. Let $a \in \mathbf{R}^{2n}$ be the incident vector of $\{t_1, \ldots, t_n\}$. Include the additional constraint "$ax \geqq 1$" in the set covering problem.

If $\bar{q} \not\geqq 0$, y is not a feasible basic vector. For each i such that $\bar{q}_i < 0$, let $\mathbf{S}_i = \{t_j : j$ such that $-d_{ij} < 0\}$, where d_{ij} is the $(i, j)^{th}$ entry in the matrix D in the canonical tableau. Clearly, any feasible basic vector must include one of the variables from $\mathbf{S}_i$. Let a_i be the incidence vector of $\mathbf{S}_i$, include the additional constraint "$a_i x \geqq 1$" for each i satisfying $\bar{q}_i < 0$, in the set covering problem.

Solve the set covering problem together with the additional constraints added in this step.

If the optimum objective value in the set covering problem is $\geqq n+1$, terminate. The solution stack at this stage contains all the complementary solutions of the LCP (q, M).

If the optimum objective value in the set covering problem is n, let $\bar{x}$ be an optimum solution for it. Let $\bar{y}$ be the complementary vector of variables corresponding to the incidence vector $\bar{x}$. Make $\bar{y}$ the new complementary vector of variables. Go to the next step with it and the current set covering problem.

This algorithm has not been computationally tested and it is not known how it may work in practice.

Developing practically efficient partial enumeration methods for the general LCP remains a problem worth investigating.

11.4 New Approaches for Linear Programming

The well known primal simplex algorithm for linear programming starts at an extreme point of the set of feasible solutions, moves along an edge direction to an adjacent extreme point, and repeats the whole process until an optimal extreme point or an unbounded edge along which the objective value is unbounded below (for minimization problems) is reached. Thus all the direction used in the primal simplex algorithm are edge directions. Recently K. G. Murty and Y. Fathi [11.4] discussed versions of the simplex algorithm based on profitable directions of movement through the interior or relative interior of the set of feasible solutions or faces of it of dimension greater than 1. They showed that with simple modifications these methods can be proved to be finite, and can be implemented using basis inverses just as the usual versions of the simplex algorithm. Computational testes indicate that these modifications leads to improvements in the running time for solving linear programs.

N. Karmarkar [11.2] has developed an entirely new polynomially bounded algorithm for solving linear programs based on profitable search directions through the interior of the set of feasible solutions. This method closes in on an optimum by creating a sequence of spheres inside the feasible region for the LP. It is claimed that preliminary computational testing has shown this method to be much faster than the simplex algorithm for large scale linear programs. A statement of this algorithm with an intuitive justification is given in the Notation section in front of this book. Here we provide a detailed treatment of the algorithm and its polynomial boundedness.

Throughout this section the symbol e denotes the column vector of all 1s of appropriate dimension, and e^T denotes its transpose.

11.4.1 Karmarkar's Algorithm for Linear Programming

The Barrier Function Approach to Handle Inequality Constraints in Nonlinear Programming

Consider the following optimization problem (P).

$$\begin{array}{ll} \text{minimize} & \theta(x) \\ \text{subject to} & Ax = b \\ & g_i(x) \geqq 0, i = 1 \text{ to } m. \end{array} \tag{P}$$

A feasible solution x for this problem is said to be **strictly feasible** if $g_i(x) > 0$ for all $i = 1$ to m. The barrier function approach for solving this problem needs an initial strictly feasible point x^0. It generates a sequence of points $\{x^r : r = 0, 1, \ldots\}$, each x^r being a strictly feasible solution of the problem.

Barrier methods work by establishing a barrier on the boundary of the feasible region that prevents the search procedure from leaving the strictly feasible part of the feasible region. A barrier function for this problem is a continuous function $B(x)$ defined on $\mathbf{\Gamma} = \{x : g_i(x) > 0, \text{ for all } i = 1 \text{ to } m\}$ that tends to $+\infty$ as the point x approaches the boundary of $\mathbf{\Gamma}$. One commonly used barrier function is the logarithmic barrier function (suggested by K. R. Frisch in 1955)

$$B(x) = -\sum_{i=1}^{m} \log(g_i(x)).$$

Here log represents the natural logarithm. The barrier function method for (P) looks at the problem

$$\begin{array}{ll} \text{minimize} & F(x) = \theta(x) - \alpha \sum_{i=1}^{m} \log(g_i(x)) \\ \text{subject to} & Ax = b \end{array} \tag{B}$$

where α is a positive parameter known as the barrier parameter. Giving α some positive value and fixing it, the barrier method tries to solve (B), by some feasible direction descent method beginning with the initial strictly feasible point x^0. Consider the line search problem of minimizing $F(x)$ along the half-line $\{x + \lambda y : \lambda \geqq 0\}$, where x is a strictly feasible point. If $\lambda_1 > 0$ is such that $g_i(x + \lambda_1 y) = 0$ for some i between 1 to m, then the step length choosen in this line search problem will be $< \lambda_1$, since $-\log g_i(x + \lambda y) \to +\infty$ as $\lambda \to \lambda_1$ from below. Thus any line searches carried out for solving (B) beginning with a strictly feasible point will always lead to another strictly feasible point.

The barrier function method for solving (P) proceeds as follows. It selects a monotonic decreasing sequence of positive values $\{\alpha_r : r = 1, 2, \ldots\}$ converging to 0. Fixing $\alpha = \alpha_1$, it solves (B) by a feasible direction descent method, beginning with the initial strictly feasible point x^0. Suppose this terminates with the strictly feasible point x^1. Now α is changed to α_2, and the new (B) solved again beginning with the

initial strictly feasible solution x^1. The process is repeated in the same way, generating the sequence of strictly feasible points $\{x^r : r = 0, 1, \ldots\}$. Under mild conditions it can be shown that this sequence converges to a solution of (P). Karmarkar's algorithm for linear programming, closely resembles this nonlinear interior point barrier method. In his algorithm, Karmarkar uses a potential function which closely resembles the logarithmic barrier function.

We will now provide a theoretical description of Karmarkar's algorithm and proofs of its polynomial boundedness. A brief discussion on issues in implementing Karmarkar's algorithm will then follow. We divide this section into various numbered subsections, for ease of cross referencing.

1 Transforming Any LP Into Another With an Optimum Objective Value of Zero

We show that any LP can be transformed into another one with a known (minimum) objective value of zero.

Consider the LP

$$\begin{array}{lr} \text{minimize} & h\chi \\ \text{subject to} & E\chi \geqq p \\ & \chi \geqq 0. \end{array} \tag{11.2}$$

Let π denote the row vector of dual variables. It is well known (see [2.26]) that solving (11.2) is equivalent to solving the following system of linear inequalities.

$$\begin{array}{rr} h\chi - \pi p \leqq 0 \\ E\chi \geqq p \\ \pi E \leqq h \\ \chi, \quad \pi \geqq 0 \end{array} \tag{11.3}$$

There is no objective function in (11.3). If $(\bar{\chi}, \bar{\pi})$ is a feasible solution for (11.3), $\bar{\chi}$ is an optimum solution for the LP (11.2) and $\bar{\pi}$ is an optimum dual solution. If (11.3) is infeasible, either (11.2) is itself infeasible, or (11.2) may be feasible but its dual may be infeasible (in the later case, the objective value is unbounded below on the set of feasible solutions of (11.2)).

The system (11.3) can be expressed as a system of equations in nonnegative variables by introducing the appropriate slack variables. To solve the resulting system, construct the usual Phase I problem by introducing the appropriate artificial variables (see Chapter 2 in [2.26]). Let u denote the vector consisting of the variables χ_j, π_i, and the artificial variables. Let the Phase I problem corresponding to (11.3) be

$$\begin{array}{lr} \text{minimize} & gu \\ \text{subject to} & Fu = d \\ & u \geqq 0 \ . \end{array} \tag{11.4}$$

The optimum objective value in (11.4) is $\geqq 0$ (since it is a Phase I problem corresponding to (11.3)) and (11.3) is feasible iff it is zero. Let v denote the row vector of dual variables for (11.4). Consider the LP

$$\begin{array}{ll} \text{minimize} & gu - vd \\ \text{subject to} & Fu = d \\ & vF \leqq g \\ & u \geqq 0 \end{array} \tag{11.5}$$

The LP (11.5) consists of the constraints in (11.4) and its dual. From the duality theory of linear programming, the optimum objective value in (11.5) is zero (since (11.4) has a finite optimum solution). The LP (11.5) can be put in standard form for LPs by the usual transformations of introducing slack variables etc., see Chapter 2 in [2.26]. If $(\bar{u}, \bar{v})$ is optimal to (11.5), then $\bar{u}$ is optimal to (11.4). If $g\bar{u} = 0$, then the χ-portion of $\bar{u}$ is an optimum solution for (11.2). If $g\bar{u} > 0$, (11.3) is infeasible and hence (11.2) is either infeasible or has no finite optimum solution.

2 Transforming an LP Into Another With a Known Strictly Positive Feasible Solution

An LP in standard form with an optimum objective value of zero, can be transformed into another with the same property, but with a known strictly positive feasible solution. Consider the LP

$$\begin{array}{ll} \text{minimize} & gy \\ \text{subject to} & Gy = d \\ & y \geqq 0 \end{array} \tag{11.6}$$

where G is a matrix of order $m \times n$, and suppose all the data is integer and the optimum objective value in (11.6) is zero. Let $y^0 > 0$ by any integer vector in $\mathbf{R}^n$. Consider the new LP

$$\begin{array}{ll} \text{minimize} & gy + g_{n+1}y_{n+1} \\ \text{subject to} & Gy + y_{n+1}(d - Gy^0) = d \\ & y \geqq 0, \quad y_{n+1} \geqq 0 \end{array} \tag{11.7}$$

clearly $(y = y^0, y_{n+1} = 1) > 0$ is a feasible solution of (11.7). Since the optimum objective value in (11.6) is zero, the same property holds in (11.7) if g_{n+1} is sufficiently large (mathematically, it is sufficient to take $g_{n+1} > 2^s$, where s is the size of $\begin{pmatrix} G & G_{.n+1} \\ g & 0 \end{pmatrix}$, $G_{.n+1} = d - Gy^0$).

3 Transforming the Feasible Set into the Intersection of a Subspace with a Simplex

Given an LP in standard form with integer or rational data, with the optimum objective value of zero, and a strictly positive feasible solution, we can transform it into another,

for which the set of feasible solutions is $\mathbf{H} \cap \mathbf{S}$, where $\mathbf{H}$ is a subspace and $\mathbf{S}$ is the standard simplex. Consider the LP

$$\begin{array}{ll} \text{minimize} & \bar{g}y \\ \text{subject to} & \overline{G}y = \bar{d} \\ & y \geqq 0 \end{array} \tag{11.8}$$

where $\overline{G}$ is of order $m \times n$, and all the data is assumed to be integer. Let L be the size of this LP (i. e., L is the total number of digits in all the data in the LP (11.8) in binary encoding, see Sections 8.3 to 8.6 and Chapters 14, 15 in [2.26]).

Since (11.8) has an optimum solution, it has an optimum solution satisfying the additional constraint

$$\sum_{j=1}^{n} y_j \leqq M$$

where M is an upper bound depending on the size L. By the results in Chapter 8 (see also Chapter 15 in [2.26]) taking $M = 2^{\mathbf{L}}$ will do. Hence (11.8) is equivalent to

$$\begin{array}{ll} \text{minimize} & \bar{g}y \\ \text{subject to} & \overline{G}y = \bar{d} \\ & e^T y \leqq M \\ & y \geqq 0 \end{array}$$

where $e^T = (1, 1, \ldots, 1) \in \mathbf{R}^n$. By introducing the slack variable y_{n+1}, this LP is the same as

$$\begin{array}{ll} \text{minimize} & \bar{g}y \\ \text{subject to} & \overline{G}y - \frac{1}{M}\bar{d}\left(\sum\limits_{j=1}^{n+1} y_j\right) = 0 \\ & \sum\limits_{j=1}^{n+1} y_j = 1 \\ & y_j \geqq 0, \quad j = 1 \text{ to } n+1 \end{array} \tag{11.9}$$

The system $\overline{G}y - \frac{1}{M}\bar{d}\left(\sum_{j=1}^{n+1} y_j\right) = 0$ is a homogeneous system of equations, and hence its set of feasible solutions is a subspace $\mathbf{H}$ in $\mathbf{R}^{n+1}$. The system $\sum_{j=1}^{n+1} y_j = 1$, $y_j \geqq 0$ for $j = 1$ to $n+1$ defines the standard simplex $\mathbf{S}$ in $\mathbf{R}^{n+1}$. So the set of feasible solutions of (11.9) is $\mathbf{H} \cap \mathbf{S}$, as desired.

4 Minimization of a Linear Function Over a Spherical Ball or an Ellipsoid

Consider the problem

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & (x - x^0)^T(x - x^0) \leqq \rho^{\mathbf{2}} \end{array}$$

If $c = 0$, every point in the sphere is optimal to this problem. If $c \neq 0$, the optimal solution of this problem is $x^0 - c^T\rho$, it is the point obtained by taking a step of length ρ (radius of the sphere) from the center x^0 in the direction of $-c^T$.

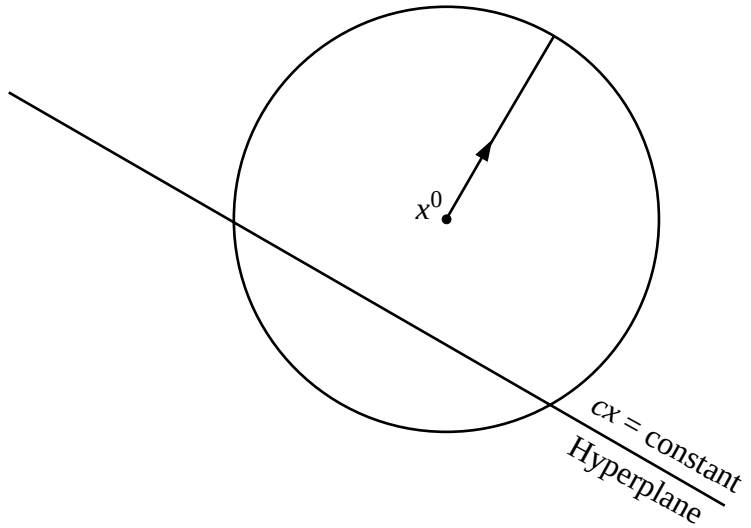

Figure 11.1 To minimize cx on the sphere, walk from the center x^0 in the direction $-c^T$, a step of length ρ =radius. The direction $-c^T$ is the steepest descent direction for the linear function cx.

Now consider the problem

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & Ax = b \\ \text{and} & x \in \mathbf{B} = \{x : ||x - x^0|| \leqq \rho\} \ . \end{array}$$

Let $\mathbf{H} = \{x : Ax = b\}$. $\mathbf{H}$ is an affine space. If $\mathbf{H} \cap \mathbf{B} \neq \emptyset$, it is a lower dimensional sphere inside the affine space $\mathbf{H}$. Again if $c = 0$, every point in $\mathbf{H} \cap \mathbf{B}$ is optimal to this problem. If $c \neq 0$, let $\bar{c}$ be the orthogonal projection of c onto $\mathbf{H}$. $\bar{c} = 0$ if c is a linear combination of the rows of A, in this case the objective function is a constant on $\mathbf{H} \cap \mathbf{B}$, and every point in it is optimal. If $\bar{c} \neq 0$, the optimal solution of this problem is the point obtained by taking a step of length equal to the radius of the lower dimensional sphere $\mathbf{H} \cap \mathbf{B}$ from its center in the direction of $-\bar{c}^T$.

Consider the following problem

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & Ax = b \\ \text{and} & x \in \mathbf{E} = \{x : (x - x^0)^T \Gamma (x - x^0) \leqq 1\} \end{array}$$

where Γ is a symmetric PD matrix of order n. So $\mathbf{E}$ is an ellipsoid. Let $\mathbf{H} = \{x : Ax = b\}$. Let F be the Cholesky factor of Γ (i. e., it is the lower triangular matrix

satisfying $FF^T = \Gamma$). To solve this problem, apply the linear transformation that transforms the ellipsoid into a sphere $\mathbf{B}$, this is

$$y = F^T(x - x^0) \quad \text{or}$$
$$x = x^0 + (F^T)^{-1}y \ .$$

This transform the affine space $\mathbf{H}$ into another affine space $\widehat{\mathbf{H}} = \{y : A(F^T)^{-1}y = (b - Ax^0)\}$, the ellipsoid $\mathbf{E}$ into the unit sphere $\mathbf{B} = \{y : ||y|| \leqq 1\}$, and the objective function cx into $c(F^T)^{-1}y + cx^0$. So the transformed problem is :

$$\begin{array}{ll} \text{minimize} & c(F^T)^{-1}y \\ \text{subject to} & y \in \widehat{\mathbf{H}} \cap \mathbf{B} \end{array}$$

which can be solved as discussed above. From the optimum solution y of this problem, we compute the optimum solution x, of the original problem using the equation $x = x^0 + (F^T)^{-1}y$.

5 Converting a Near Optimum Feasible Solution into an Optimum Feasible Solution

Consider the LP

$$\begin{array}{ll} \text{minimize} & z(x) = cx \\ \text{subject to} & Ax = b \\ & x \geqq 0 \end{array} \tag{11.10}$$

Let $\bar{x}$ be a feasible solution for it. A well known result in LP says that if $\bar{x}$ is not a BFS for this problem, then a BFS $\hat{x}$ for it satisfying $c\hat{x} \leqq c\bar{x}$ can be obtained, or it can be established that cx is unbounded below in this problem. See [2.26]. We describe the procedure for doing it here.

Let $\mathbf{J} = \{j : \bar{x}_j > 0\}$. If $\{A_{.j} : j \in \mathbf{J}\}$ is linearly independent, $\bar{x}$ is itself a BFS. If $\bar{x}$ is not BFS, $\{A_{.j} : j \in \mathbf{J}\}$ is linearly dependent. Let a linear dependence relation among these vectors be

$$\sum_{j \in \mathbf{J}} \alpha_j A_{\cdot j} = 0$$

where $(\alpha_j : j \in \mathbf{J}) \neq 0$. Such a vector $(\alpha_j : j \in \mathbf{J})$ can be computed by pivotal methods for checking linear independence of the set $\{A_{.j} : j \in \mathbf{J}\}$, see [2.26].

Since $\bar{x}$ is feasible, and from the definition of $\mathbf{J}$, we also have

$$\sum_{j \in \mathbf{J}} \bar{x}_j A_{.j} = b$$

$$\therefore \quad \sum_{j \in \mathbf{J}} (\bar{x}_j + \theta\alpha_j) A_{.j} = b$$

for all real values of θ. Define the vector $x(\theta)$ by

$$x_j(\theta) = \begin{cases} \bar{x}_j + \theta\alpha_j & \text{for } j \in \mathbf{J} \\ 0 & \text{for } j \notin \mathbf{J} \ . \end{cases}$$

Now define

$$\theta_1 = \begin{cases} -\infty, & \text{if } \alpha_j \leqq 0 \text{ for all } j \in \mathbf{J}; \\ \max\{-\frac{\bar{x}_j}{\alpha_j} : j \in \mathbf{J} \text{ and such that } \alpha_j > 0\}, & \text{otherwise} \end{cases}$$

$$\theta_2 = \begin{cases} +\infty, & \text{if } \alpha_j \geqq 0 \text{ for all } j \in \mathbf{J}; \\ \min\{-\frac{\bar{x}_j}{\alpha_j} : j \in \mathbf{J} \text{ and such that } \alpha_j < 0\}, & \text{otherwise} \ . \end{cases}$$

Clearly $\theta_1 < 0$, $\theta_2 > 0$, and $x(\theta) \geqq 0$ and hence feasible to the LP for all $\theta_1 \leqq \theta \leqq \theta_2$. Since $(\alpha_j : j \in \mathbf{J}) \neq 0$, at least one among θ_1 or θ_2 is finite. If $\sum_{j\in\mathbf{J}} c_j\alpha_j = 0$, let $\gamma = \theta_1$ or θ_2 whichever is finite, break ties arbitrarily.

If $\sum_{j\in\mathbf{J}} c_j\alpha_j > 0$, and $\theta_1 = -\infty$ then $\{x(\theta) : \theta \leqq \theta_2\}$ is a feasible half-line along which cx diverges to $-\infty$. Likewise if $\sum_{j\in\mathbf{J}} c_j\alpha_j < 0$, and $\theta_2 = +\infty$, then $\{x(\theta) : \theta \geqq \theta_1\}$ is a feasible half-line along which cx diverges to $-\infty$. If neither of these unboundedness conditions are satisfied, select $\gamma = \theta_1$ if $\sum_{j\in\mathbf{J}} c_j\alpha_j > 0$, or $\gamma = \theta_2$ if $\sum_{j\in\mathbf{J}} c_j\alpha_j < 0$.

Then $x(\gamma)$ is a feasible solution satisfying $cx(\gamma) \leqq c\bar{x}$, and the number of positive components in $x(\gamma)$ is at least one less than that in $\bar{x}$.

Repeat the same process now with the feasible solution $x(\gamma)$. After at most $|\mathbf{J}|$ of these steps, we will either obtain a BFS $\hat{x}$ satisfying $c\hat{x} \leqq c\bar{x}$, or establish that cx is unbounded below in this LP.

Example 11.1

Consider the following LP

x_1	x_2	x_3	x_4	x_5	x_6	x_7	b
1	0	0	1	0	1	−1	3
0	1	0	0	−1	2	−1	5
0	0	1	−1	1	1	−2	7
−10	4	6	2	4	8	10	$= z(x)$ minimize

$x_j \geqq 0$ for all j

Let $x^0 = (\frac{5}{2}, 6, \frac{13}{2}, \frac{1}{2}, 1, 0, 0)^T$ be the feasible solution with an objective value $z(x^0) = 43$. Denote the coeficient of x_j in $z(x)$ by c_j, and the column vector of x_j in the constraint matrix by $A_{.j}$. $\mathbf{J}$ = the set of subscripts of positive variables in x^0 is $\{1, 2, 3, 4, 5\}$. The set of columns $\{A_{.j} : j = 1 \text{ to } 5\}$ is linearly dependent, and a linear dependence relation among them is

$$-A_{.1} + A_{.3} + A_{.4} = 0$$

So the vector α^0 leading to this linear dependence relation is $(-1,0,1,1,0,0,0)^T$ and $z(\alpha^0) = 18 > 0$. The feasible solution $x^0(\theta)$ constructed in the procedure is

$$x^0(\theta) = \left(\frac{5}{2} - \theta, 6, \frac{13}{2} + \theta, \frac{1}{2} + \theta, 1, 0, 0\right)^T$$

and so $\theta_1 = -\frac{1}{2}$, $\theta_2 = \frac{5}{2}$. Since $z(\alpha^0) > 0$, we choose $\gamma = \theta_1 = -\frac{1}{2}$. The next feasible solution in $x^0(\theta_1) = x^1$ is

$$x^1 = (3,6,6,0,1,0,0)^T$$

It can be verified that $z(x^1) = 34$ and that x^1 has only 4 positive components. Continuing the procedure with x^1, the set of columns to examine is $\{A_{.1}, A_{.2}, A_{.3}, A_{.5}\}$ which again is linearly dependent, with the linear dependence relation

$$A_{.2} - A_{.3} + A_{.5} = 0 \ .$$

The vector α^1 corresponding to this linear dependence relation is $(0,1,-1,0,1,0,0)^T$ and $z(\alpha^1) = 2 > 0$. The feasible solution $x^1(\theta)$ constructed in the procedure is

$$x^1(\theta) = (3, 6+\theta, 6-\theta, 0, 1+\theta, 0, 0)^T$$

and so $\theta_1 = -1$, $\theta_2 = 6$, and since $z(\alpha^1) > 0$, we choose $\alpha = \theta_1 = -1$. The next feasible solution is $x^1(\theta_1) = x^2$, $x^2 = (3,5,6,0,0,0,0)^T$, $z(x^2) = 32$. Now x^2 is a BFS and it satisfies $z(x^2) < z(x^0)$.

Consider the LP (11.10) again. Suppose the data is integer, and L is the size of this LP. Let z^* be the unknown optimum objective value in this LP. If $\bar{x}$ is a feasible solution for this LP whose objective value is sufficiently close to the optimum objective value, e.g. if $c\bar{x}$ is within $2^{-\mathbf{L}}$ of z^*, then the BFS obtained by applying the above procedure beginning with $\bar{x}$, will be an optimum solution for the LP, by the results proved in the ellipsoid algorithm, see Chapter 8 and [2.26] and Figure 11.2. This follows because when L is the size of the LP, any BFS x satisfying, objective value at x, $z(x) \leqq z^* + 2^{-\mathbf{L}}$, has to be an optimum BFS, by the results proved under the ellipsoid algorithm.

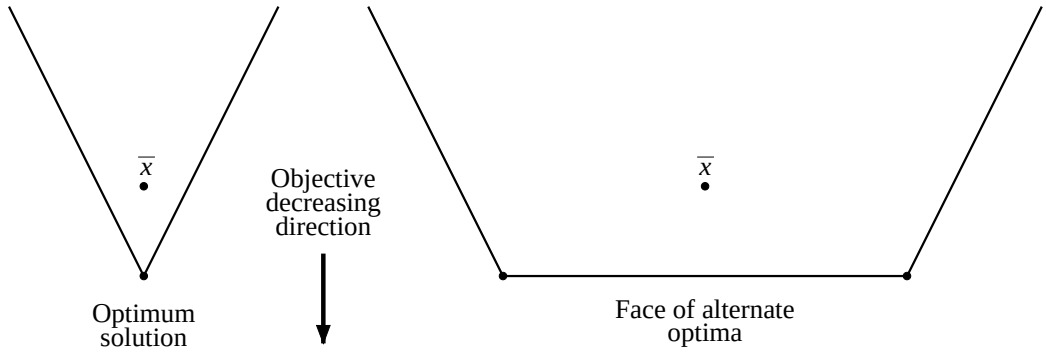

Figure 11.2 If $\bar{x}$ is near optimal, a BFS obtained by above procedure will be optimal, whether problem has unique optimum solution or has alternate optima.

Thus if a near optimal feasible solution with objective value sufficiently close to the optimum can be found, the procedure discussed in this subsection can be used to convert it into an exact optimum solution for the LP. This result is used in Karmarkar's algorithm. Karmarkar's algorithm computes a near optimal solution for an LP and then converts it into an exact optimum solution of the problem using the procedure discussed here.

6 Karmarkar's Algorithm

Consider the LP in the form

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & x \in \Omega \cap \mathbf{S} \end{array} \tag{11.11}$$

where

$$\Omega = \{x : Ax = 0\}$$

$$\mathbf{S} = \left\{x : x \geqq 0, \quad \sum_{j=1}^{n} x_j = 1\right\}$$

A is of order $m \times n$. Without any loss of generality we assume that the rank of A is m. We make the Following assumptions.

(i) $x^0 = \frac{1}{n}e$, where e is the column vector of all 1's in $\mathbf{R}^n$ is feasible to this LP.

(ii) The optimum objective value in (11.11) is zero.

Karmarkar's algorithm generates a finite sequence of feasible points $x^0, x^1, \ldots$, all of them > 0, such that cx^r is strictly decreasing. L denotes the size of (11.11).

These assumptions also imply that the rank of $\begin{pmatrix} A \\ e^T \end{pmatrix}$ is $m+1$. If $cx^0 = 0$, by the assumptions, x^0 is optimal to (11.11), we terminate. So we assume that $cx^0 > 0$. The method terminates when a feasible solution x^r satisfying $cx^r \leqq 2^{-\mathcal{O}(L)}$ is obtained, and then converts this approximate optimal solution x^r into an exact optimal solution as in Subsection 5.

If c is a linear combination of the rows of A, $cx = 0$ at all feasible solutions x, and so our assumptions imply that c is not a linear combination of the rows of A.

Now we shall describe the general step of the algorithm.

Step $r+1$: Assume we are given $x^r > 0$, $x^r \in \Omega \cap \mathbf{S}$. Let $x^r = a = (a_1, \ldots, a_n)^T$. Let $D = \text{diag}\{a_1, \ldots, a_n\} = (d_{ij})$ with $d_{ii} = a_i$, $i = 1$ to n, and $d_{ij} = 0$ for $i \neq j$. So D is a positive diagonal matrix of order $n \times n$.

We now construct a projective transformation $T : \mathbf{S} \to \mathbf{S}$, which depends on the vector a. For $x \in \mathbf{S}$,

$$T(x) = \frac{D^{-1}x}{e^T D^{-1}x} \ .$$

It can be verified that $T(x) \in \mathbf{S}$ for all $x \in \mathbf{S}$. Also, if $x \in \mathbf{S}$ satisfies $x > 0$, so is $T(x)$. So, the transformation $T(x)$ maps every point in the relative interior of $\mathbf{S}$ (i. e., a point in $\mathbf{S}$ which is > 0) into another point in the relative interior of $\mathbf{S}$. It can be verified that

$$T(a) = a^0 = \frac{1}{n}e \ .$$

If $T(x) = x'$, the inverse transformation yielding $T^{-1}(x') = x$ is

$$T^{-1}(x') = \frac{Dx'}{e^T Dx'} \ .$$

Associate the objective function cx with the potential function $f(x)$ defined over the intersection of Ω with the relative interior of $\mathbf{S}$, given by

$$f(x) = \sum_{j=1}^{n} \log\left(\frac{cx}{x_j}\right)$$

where log denotes the natural logarithm. Since all the points obtained in the algorithm will be strictly positive, they are in the relative interior of $\mathbf{S}$, and $f(x)$ is well defined at them. For x from the relative interior of $\mathbf{S}$ (i. e., $x \in \mathbf{S}$ and $x > 0$) with $T(x) = x'$, define the transformed potential function $f'(x')$ so that it satisfies $f(x) = f'(T(x)) = f'(x')$. Then it can be verified that

$$f'(y) = \sum_{j=1}^{n} \log\left(\frac{\hat{c}y}{y_j}\right) - \sum_{j=1}^{n} \log(a_j)$$

where $\hat{c} = cD$.

Let Ω' denote the tranformation of the subspace Ω under T. Thus

$$\Omega' = \{x' : ADx' = 0\} \ .$$

Now define

$$\Omega'' = \left\{ \begin{array}{r} y : \ ADy = 0 \\ e^T y = 1 \end{array} \right\}$$

$$B = \left(\begin{array}{c} AD \\ e^T \end{array} \right) \ .$$

As discussed earlier, B is of full row rank. Since $a \in \Omega$, we have $ADe = 0$, so $a^0 \in \Omega''$.

Let $\bar{\rho}$, $\underline{\rho}$ be respectively the radii of the largest sphere with center a^0 contained in the simplex $\mathbf{S}$, smallest shpere with center a^0 containing $\mathbf{S}$. See Figure 11.3.

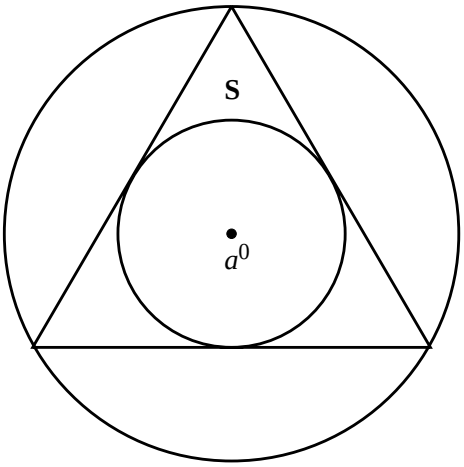

Figure 11.3 Inscribed sphere and circumscribing sphere

Then

$$\underline{\rho} = \frac{1}{\sqrt{n(n-1)}}, \qquad \bar{\rho} = \sqrt{\frac{n-1}{n}} = (n-1)\underline{\rho} \ .$$

For $0 < \alpha < 1$, define

$$\mathbf{B}(a^0, \alpha\underline{\rho}) = \{x : \ ||x - a^0|| \leqq \alpha\underline{\rho}\} \ .$$

Since $0 < \alpha < 1$, the sphere $\{x : \sum_{j=1}^{n} x_j = 1\} \cap \mathbf{B}(a^0, \alpha\underline{\rho}) \subset \mathbf{S}$. The projective transformation $T(x)$, transforms the set of feasible solutions of (11.11) into $\Omega'' \cap \mathbf{S}$. However, $T(x)$ does not transform cx into a linear function. But the potential function $f(x)$, which depends on ratios of linear functions is transformed into another function of the same form, $f'(x')$. We will show later on that a reduction in $f(x)$ leads to a reduction in cx. The problem of minimizing $f(x)$ gets transformed into that of minimizing $f'(x')$. We show later on that minimizing $f'(x')$ can be achieved approximately by optimizing a linear approximation, $\hat{c}x'$.

Instead of optimizing over $\Omega'' \cap \mathbf{S}$ in the transformed problem, we optimize over the simpler subset $\Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$. The reasons for this are explained below.

Our original problem is transformed into that of optimizing $f'(x')$ over $\Omega' \cap \mathbf{S}$. Since

$$\left\{x : \sum_{j=1}^{n} x_j = 1\right\} \cap \mathbf{B}(a^0, \underline{\rho}) \subset \mathbf{S} \subset \mathbf{B}(a^0, \bar{\rho}) \cap \left\{x : \sum_{j=1}^{n} x_j = 1\right\}$$

$$\begin{array}{ccccc} \text{min value of } f'(x') & & \text{min value of } f'(x') & & \text{min value of } f'(x') \\ \text{over } \Omega'' \cap \mathbf{B}(a^0, \underline{\rho}) & \geqq & \text{over } \Omega'' \cap \mathbf{S} & \geqq & \text{over } \Omega'' \cap \mathbf{B}(a^0, \bar{\rho}) \end{array}$$

Since $\Omega'' \cap \mathbf{B}(a^0, \rho)$ for any ρ is a sphere, optimizing over it is much easier than optimizing over $\Omega'' \cap \mathbf{S}$. To optimize $f'(x')$ over $\Omega'' \cap \mathbf{B}(a^0, \rho)$, we approximate $f'(x')$ by a linear function, $\hat{c}x'$, and the minimization of this linear function over $\Omega'' \cap \mathbf{B}(a^0, \rho)$ can be carried out very easily by the simple techniques discussed in Subsection 4.

If $\gamma_0, \gamma_1, \gamma_2$ denote the minimum value of this linear function $\hat{c}x'$ over $\Omega'' \cap \mathbf{B}(a^0, \underline{\rho})$, $\Omega' \cap \mathbf{S}$, $\Omega'' \cap \mathbf{B}(a^0, \bar{\rho})$ respectively, we have $\gamma_0 \geqq \gamma_1 \geqq \gamma_2$, and so

$$\hat{c}a^0 - \gamma_0 \leqq \hat{c}a^0 - \gamma_1 \leqq \hat{c}a^0 - \gamma_2 = \left(\frac{\bar{\rho}}{\underline{\rho}}\right)(\hat{c}a^0 - \gamma^0)$$

the last equation follows from the results in Subsection 4. So

$$\frac{\hat{c}a^0 - \gamma_0}{\hat{c}a^0 - \gamma_1} \geqq \frac{1}{n-1}$$
$$\frac{\gamma_0 - \gamma_1}{\hat{c}a^0 - \gamma_1} \leqq 1 - \frac{1}{n-1}$$

So by going from the point a^0 to the point that minimizes $\hat{c}x'$ over $x' \in \Omega'' \cap \mathbf{B}(a^0, \underline{\rho})$, we come closer to the minimum value of the objective function by a factor of $\left(1 - \frac{1}{n-1}\right)$.

In practice, we optimize over a smaller subset $\Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$ for $0 < \alpha < 1$ for the following reasons.

a) it allows for optimization of $f'(x')$ to be approximated closely by optimization of a linear function.
b) Under finite precision or other approximate arithmetic, it provides us a margin to absorb errors without going outside the simplex.

See Figure 11.4. The choice of $\alpha = \frac{1}{4}$ works (this leads to the factor δ discussed later on in Theorem 11.4 to be $> \frac{1}{32}$). In practical implementation, one may want to choose a value of α much closer to 1 for rapid convergence.

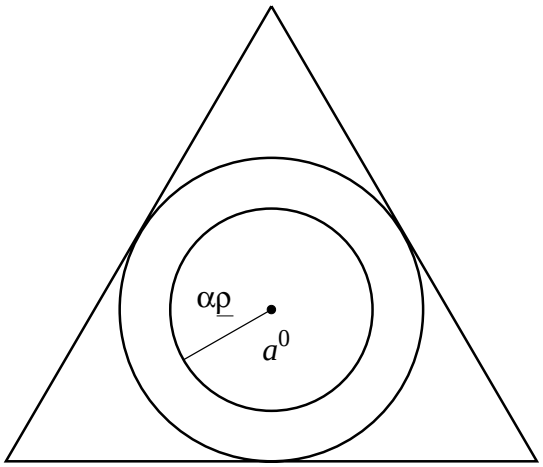

Figure 11.4 The simplex $\mathbf{S}$, and the inscribed sphere $\mathbf{B}(a^0, \alpha\underline{\rho}) \cap \{x : \sum_{j=1}^{n} x_j = 1\}$ inside it, for $0 < \alpha < 1$.

Since $\mathbf{B}(a^0, \alpha\underline{\rho})$ is a sphere with center a^0, and Ω'' is an affine space containing the point a^0, the intersection $\Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$ is a lower dimensional sphere. As discussed in Subsection 4 above, minimizing a linear function over the lower dimensional sphere $\Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$ requires taking a step from the center a^0, in the direction of the negative gradient, with step length equal to the radius of the sphere, in the affine space Ω''. We provide the details of this algorithm.

Subroutine to minimize $\hat{c}x'$ Over $x' \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$

First project $\hat{c}$ orthogonally onto the subspace $\{y : By = 0\}$. This yields

$$\hat{c}_p = \hat{c}\left[I - B^T(BB^T)^{-1}B\right]$$

If $\hat{c}_p = 0$, the objective function will have the same value at all feasible solutions, contradicting our assumptions. So $\hat{c}_p \neq 0$. Let

$$\tilde{c}_p = \frac{\hat{c}_p}{||\hat{c}_p||}$$
$$g' = a^0 - \alpha\underline{\rho}\tilde{c}_p$$

Then g' is the point which minimizes $\hat{c}x'$ over $x' \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$. We will prove this in Theorem 11.1 given below.

Now define

$$x^{r+1} = T^{-1}(g') = \frac{Dg'}{e^T Dg'} \ .$$

If $cx^{r+1} = 0$, x^{r+1} is optimal to (11.11), terminate. If $cx^{r+1} > 0$ but sufficiently small (i. e., $cx^{r+1} \leqq 2^{-\mathcal{O}(L)}$) terminate with the conclusion that x^{r+1} is near optimal to (11.11) and convert it into an exact optimal solution as in Subsection 5. If these conditions are not satisfied, go to the next step.

Proof of the Algorithm and its Polynomial Boundedness

Theorem 11.1 *The vector g' minimizes $\hat{c}x'$ over $x' \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$.*

Proof. Let $z \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$. Since Ω'' is an affine space and both g', $z \in \Omega''$, we have $B(g' - z) = 0$. So, $B^T(BB^T)^{-1}B(g' - z) = 0$. Therefore $(\hat{c} - \hat{c}_p)(g' - z) = 0$. Thus $\hat{c}(g' - z) = \hat{c}_p(g' - z) = ||\hat{c}_p||\tilde{c}_p(a^0 - \alpha\underline{\rho}\tilde{c}_p^T - z) = ||\hat{c}_p||\big(\tilde{c}_p(a^0 - z) - \alpha\underline{\rho}\big)$ (since $\tilde{c}_p\tilde{c}_p^T = ||\tilde{c}_p|| = 1$). But, $\tilde{c}_p(a^0 - z) \leqq ||\tilde{c}_p||\ ||a^0 - z||$ (by Cauchy-Schwartz inequality) $= ||a^0 - z||$ (since $||\tilde{c}_p|| = 1$) $\leqq \alpha\underline{\rho}$, since $z \in \mathbf{B}(a^0, \alpha\underline{\rho})$. Therefore $\tilde{c}_p(a^0 - z) - \alpha\underline{\rho} \leqq 0$, and therefore by the above $\hat{c}(g' - z) \leqq 0$. Hence, $\hat{c}g' \leqq \hat{c}z$ for all $z \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$, that is, g' minimizes $\hat{c}x'$ over $x' \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$.

□

Theorem 11.2 *There exists a point $\bar{x} \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$ such that*

$$\begin{array}{lll} \textit{either} & \textit{(i)} & \hat{c}\bar{x} = 0 \\ \textit{or} & \textit{(ii)} & f'(\bar{x}) \leqq f'(a_0) - \delta \end{array}$$

where δ is a positive constant depending on α.

Proof. Let x^* minimize cx over $\Omega \cap \mathbf{S}$. By hypothesis $cx^* = 0$. Define $\xi = \frac{D^{-1}x^*}{e^T D^{-1}x^*}$.

Case 1 : $\xi \in \mathbf{B}(a^0, \alpha\underline{\rho})$. In this case let $\bar{x} = \xi$. Then $\bar{x} \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$ and $\hat{c}\bar{x} = 0$, so (i) is satisfied.

Case 2 : $\xi \notin \mathbf{B}(a^0, \alpha\underline{\rho})$. In this case, let $\bar{x}$ be the point at which the line segment joining a^0 with ξ intersects the boundary of the sphere $\mathbf{B}(a^0, \alpha\underline{\rho})$. Then $\bar{x} = (1-\lambda)a^0 + \lambda\xi$ for some $0 < \lambda < 1$. Since a^0 and ξ are in Ω'', so is $\bar{x}$. So $\bar{x} \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$, and $\hat{c}\bar{x} = (1-\lambda)\hat{c}a^0 + \lambda\hat{c}\xi = (1-\lambda)\hat{c}a^0$ (since $\hat{c}\xi = cD\xi = 0$ because $cx^* = 0$). So

$$\frac{\hat{c}a^0}{\hat{c}\bar{x}} = \frac{1}{1-\lambda} \tag{11.12}$$

Now

$$\begin{aligned}
f'(a^0) - f'(\bar{x}) &= \sum_{j=1}^{n} \log\left(\frac{\hat{c}a^0}{a^0_j}\right) - \sum_{j=1}^{n} \log\left(\frac{\hat{c}\bar{x}}{\bar{x}_j}\right) \\
&= \sum_{j=1}^{n} \log\left(\left(\frac{\hat{c}a^0}{\hat{c}\bar{x}}\right)\left(\frac{\bar{x}_j}{a^0_j}\right)\right) \\
&= \sum_{j=1}^{n} \log\left(\frac{\bar{x}_j}{(1-\lambda)a^0_j}\right) \qquad \text{by (11.12)} \\
&= \sum_{j=1}^{n} \log\left(\frac{(1-\lambda)a^0_j + \lambda\xi_j}{(1-\lambda)a^0_j}\right) \\
&= \sum_{j=1}^{n} \log\left(1 + \left(\frac{\lambda}{1-\lambda}\right)\left(\frac{\xi_j}{a^0_j}\right)\right)
\end{aligned}$$

It can easily be verified that if $\gamma_i \geqq 0$ for all i, then the product $\prod_i(1+\gamma_i) \geqq 1 + \sum_i \gamma_i$. Taking logs on both sides we have $\sum_i \log(1+\gamma_i) \geqq \log(1 + \sum_i \gamma_i)$. Applying this to the above, we have

$$\begin{aligned}
f'(a^0) - f'(\bar{x}) &\geqq \log\left(1 + \left(\tfrac{\lambda}{1-\lambda}\right)\frac{\sum_{j=1}^{n}\xi_j}{(1/n)}\right), \text{ since } a^0_j = 1/n \text{ for all } j \\
&\geqq \log\left(1 + \tfrac{n\lambda}{1-\lambda}\right), \qquad \text{since } \textstyle\sum_{j=1}^{n}\xi_j = 1
\end{aligned}$$

Now, $\bar{x} = (1-\lambda)a^0 + \lambda\xi$. So $\bar{x} - a^0 = \lambda(\xi - a^0)$. Since $\bar{x}$ is on the boundary of the sphere $\mathbf{B}(a^0, \alpha\underline{\rho})$, we have $||\bar{x} - a^0|| = \alpha\underline{\rho}$, so from the above $\alpha\underline{\rho} = ||\bar{x} - a^0|| = \lambda||\xi - a^0|| \leqq \lambda\bar{\rho}$. So $\lambda \geqq \frac{(\alpha\underline{\rho})}{\bar{\rho}} = \frac{\alpha}{n-1}$. So

$$1 + \frac{n\lambda}{1-\lambda} \geqq 1 + \frac{n\left(\frac{\alpha}{n-1}\right)}{1 - \frac{\alpha}{n-1}} = 1 + \frac{n\alpha}{n-1-\alpha} \geqq 1 + \alpha$$

Therefore, from the above

$$f'(a^0) - f'(\bar{x}) \geqq \log(1+\alpha)$$

Thus taking $\delta = \log(1+\alpha)$ establishes (ii).

□

Lemma 11.1 *Let γ be a real number. If $|\gamma| \leqq \beta < 1$ then $|\log(1+\gamma) - \gamma| \leqq \frac{\gamma^2}{2(1-\beta)^2}$.*

Proof. Let $\psi(\gamma) = \log(1+\gamma)$. Then

$$\frac{d}{d\gamma}\psi(\gamma) = \frac{1}{1+\gamma}, \text{ and } \frac{d^2}{d\gamma^2}\psi(\gamma) = \frac{-1}{(1+\gamma)^2} \ .$$

By the mean value theorem of calculus applied to the function $\log(1+\gamma)$, we have

$$\log(1+\gamma) = \log(1) + \gamma\Big(\frac{d}{d\gamma}\psi(\gamma)\Big)_{\gamma=0} + \frac{\gamma^2}{2}\Big(\frac{d^2}{d\gamma^2}\psi(\bar{\gamma})\Big)$$

for some $\bar{\gamma}$ satisfying $|\bar{\gamma}| \leqq |\gamma|$. So

$$\log(1+\gamma) = \gamma - \frac{\gamma^2}{2}\Big(\frac{1}{(1+\bar{\gamma})^2}\Big)$$

$$|\log(1+\gamma) - \gamma| = \frac{\gamma^2}{2}\Big(\frac{1}{(1+\bar{\gamma})^2}\Big) \leqq \frac{\gamma^2}{2(1-\beta)^2}$$

□

Lemma 11.2 *Let $\beta = \alpha\sqrt{\frac{n}{n-1}}$. Then*

$$\left|\sum_{j=1}^{n} \log\Big(\frac{x_j}{a_j^o}\Big)\right| \leqq \frac{\beta^2}{2(1-\beta)^2} \quad \textit{for all } x \in \mathbf{B}(a^0, \alpha\underline{\rho}) \cap \mathbf{S} \ .$$

Proof. Let $x \in \mathbf{B}(a^0, \alpha\underline{\rho}) \cap \mathbf{S}$. Then $||x - a^0||^2 \leqq \alpha^2\underline{\rho}^2$. So (since $a_j^0 = \frac{1}{n}$ for all j)

$$\sum_{j=1}^{n}\Big(\frac{x_j - a_j^0}{a_j^0}\Big)^2 \leqq \frac{\alpha^2\underline{\rho}^2}{(1/n)^2} = \alpha^2\underline{\rho}^2 n^2 = \frac{\alpha^2 n^2}{n(n-1)} = \beta^2$$

So, $\left\|\frac{x_j - a_j^0}{a_j^0}\right\| \leqq \beta$ for all j. Therefore, by Lemma 11.1

$$\left|\log\Big(1 + \frac{x_j - a_j^0}{a_j^0}\Big) - \Big(\frac{x_j - a_j^0}{a_j^0}\Big)\right| \leqq \Big(\frac{x_j - a_j^0}{a_j^0}\Big)^2\Big(\frac{1}{2(1-\beta)^2}\Big)$$

$$\therefore \quad \left|\sum_{j=1}^{n}\log\Big(\frac{x_j}{a_j^0}\Big) - \sum_{j=1}^{n}\Big(\frac{x_j - a_j^0}{a_j^0}\Big)\right|$$

$$\leqq \Big(\frac{1}{2(1-\beta)^2}\Big)\left(\sum_{j=1}^{n}\Big(\frac{x_j - a_j^0}{a_j^0}\Big)^2\right)$$

$$\leqq \frac{\beta^2}{2(1-\beta)^2}$$

This implies that $\left|\sum_{j=1}^{n}\log\big(\frac{x_j}{a_j^0}\big)\right| \leqq \frac{\beta^2}{2(1-\beta)^2}$, since $\sum_{j=1}^{n}\Big(\frac{x_j - a_j^0}{a_j^0}\Big) = \frac{1}{n}\left(\sum_{j=1}^{n}(x_j - a_j^0)\right) = 0$ (as x and $a^0 \in \mathbf{S}$).

□

Theorem 11.3 *The point g' which minimimizes $\hat{c}x'$ over $x' \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$ satisfies*

$$\begin{array}{rrll} \textit{either} & \textit{(i)} & \hat{c}g' & = 0 \\ \textit{or} & \textit{(ii)} & f'(g') & \leqq f'(a^0) - \delta \end{array}$$

where δ is a constant depending on α. If $\alpha = \frac{1}{4}$, $\delta \geqq \frac{1}{32}$.

Proof. Define

$$\tilde{f}(x) = n \log\Big(\frac{\hat{c}x}{\hat{c}a^0}\Big) \ .$$

Let h be the point where $f'(x')$ achieves its minimum value over $x' \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$. Then,

$$\begin{aligned} f'(a^0) - f'(g') =& f'(a^0) - f'(h) + f'(h) - f'(g') \\ =& [f'(a^0) - f'(h)] + \big[f'(h) - \big(f'(a^0) + \tilde{f}(h)\big)\big] \\ & - \big[f'(g') - \big(f'(a^0) + \tilde{f}(g')\big)\big] + \big[\tilde{f}(h) - \tilde{f}(g')\big] \end{aligned} \tag{11.13}$$

Now if the minimum value of $\hat{c}x'$ over $x' \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$ is zero, condition (i) of the theorem holds trivially. Let us assume that this is not the case. Then by Theorem 11.2

$$f'(a^0) - f'(h) \geqq \log(1+\alpha) \ . \tag{11.14}$$

For $x' \in \mathbf{B}(a^0, \alpha\underline{\rho}) \cap \Omega''$, we have

$$\begin{aligned} f'(x') - \big(f'(a^0) + \tilde{f}(x')\big) &= \sum_{j=1}^{n} \log\Big(\frac{\hat{c}x'}{x'_j}\Big) - \sum_{j=1}^{n} \log\Big(\frac{\hat{c}a^0}{a^0_j}\Big) - n \log\Big(\frac{\hat{c}x'}{\hat{c}a^0}\Big) \\ &= -\sum_{j=1}^{n} \log\Big(\frac{x'_j}{a^0_j}\Big) \end{aligned}$$

So

$$\begin{aligned} \big|f'(x') - \big(f'(a^0) + \tilde{f}(x')\big)\big| &= \Big|\sum_{j=1}^{n} \log\Big(\frac{x'_j}{a^0_j}\Big)\Big| \\ &\leqq \frac{\beta^{\mathbf{2}}}{2(1-\beta)^{\mathbf{2}}} \quad \text{by Lemma 11.2} \end{aligned} \tag{11.15}$$

But $\tilde{f}(x')$ depends on $\hat{c}x'$ in a monotonically increasing manner. So $\tilde{f}(x')$ and $\hat{c}x'$ attain their minimum value over $x' \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$ at the same point, that is g'. So

$$\tilde{f}(h) \geqq \tilde{f}(g') \ . \tag{11.16}$$

Now from (11.15) we have, for $x' \in \Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$,

$$f'(x') - \big(f'(a^0) + \tilde{f}(x')\big) \geqq -\frac{\beta^{\mathbf{2}}}{2(1-\beta)^{\mathbf{2}}} \ . \tag{11.17}$$

Also $\Omega'' \subset \Omega'$. So both h and $g' \in \Omega'$. From (11.13), (11.14), (11.15), (11.16), (11.17), we have

$$f'(a^0) - f'(g') \geqq \log(1+\alpha) - \frac{\beta^{\mathbf{2}}}{(1-\beta)^{\mathbf{2}}} \ .$$

We know that $\log(1+\alpha) \geqq \alpha - \frac{\alpha^{\mathbf{2}}}{2}$, for $0 < \alpha < 1$. Also

$$\frac{\beta^{\mathbf{2}}}{(1-\beta)^{\mathbf{2}}} = \frac{\alpha^{\mathbf{2}} n}{(n-1)\left(1 - \alpha\left(\frac{n}{n-1}\right)^{\frac{1}{\mathbf{2}}}\right)^{\mathbf{2}}}$$

So from the above, we have

$$f'(a^0) - f'(g') \geqq \delta(n) = \alpha - \frac{\alpha^{\mathbf{2}}}{2} - \frac{\alpha^{\mathbf{2}} n}{(n-1)\left(1 - \alpha\left(\frac{n}{n-1}\right)^{\frac{1}{\mathbf{2}}}\right)^{\mathbf{2}}}$$

As $n \to \infty$, $\delta(n) \to \alpha - \frac{\alpha^{\mathbf{2}}}{2} - \frac{\alpha^{\mathbf{2}}}{(1-\alpha)^{\mathbf{2}}}$. If $n \geqq 4$, $\alpha = \frac{1}{4}$, we have $\delta(n) \geqq \frac{1}{32}$.

□

Theorem 11.4 *Either $cx^{r+1} = 0$, or $f(x^{r+1}) \leqq f(x^r) - \delta$, where δ is a constant depending only on α, as in Theorem 11.3.*

Proof. We have proved in Theorem 11.3 that either $\hat{c}g' = 0$, or $f'(g') \leqq f'(a^0) - \delta$. Now

$$\begin{aligned} x^r &= T^{-1}(a^0) \\ x^{r+1} &= T^{-1}(g') \\ f'(T(x)) &= f(x) \text{ for all } x \in \mathbf{S} \ . \end{aligned}$$

So, by applying T^{-1}, we have from the above, that either $cx^{r+1} = 0$, or $f(x^{r+1}) \leqq f(x^r) - \delta$.

□

Theorem 11.5 *In $\mathcal{O}\big(n(l + \log n)\big)$ steps, the algorithm finds a feasible point $x > 0$ such that*

$$\begin{aligned} &\textit{either} \quad cx \ = 0 \\ &\textit{or} \quad \tfrac{cx}{ca^0} \leqq 2^{-l} \end{aligned}$$

Proof. Suppose $cx^r = 0$ did not occur in the first N steps. Then, by Theorem 11.4

$$\begin{aligned} f(x^r) &\leqq f(x^{r-1}) - \delta, \ \text{ for } r = 1 \text{ to } N \\ \therefore \ f(x^r) &\leqq f(x^0) - r\delta \end{aligned}$$

$$\begin{aligned} \therefore \quad \sum_{j=1}^{n} \log\left(\tfrac{cx^r}{x_j^r}\right) &\leqq \sum_{j=1}^{n} \log\left(\tfrac{ca^0}{a_j^0}\right) - r\delta \\ \text{i. e.,} \quad n \log\left(\tfrac{cx^r}{ca^0}\right) &\leqq \sum_{j=1}^{n} \log(x_j^r) - \sum_{j=1}^{n} \log(a_j^0) - r\delta \\ &\leqq n \log(n) - r\delta, \ \text{since } x_j^r \leqq 1 \text{ and } a_j^0 = \tfrac{1}{n} \text{ for all } j \\ \therefore \quad \log\left(\tfrac{cx^r}{ca^0}\right) &\leqq \log n - \tfrac{r\delta}{n} \ . \end{aligned}$$

So if $r = \left\lceil \frac{n}{\delta}(l + \log n) \right\rceil$, we have

$$\log\left(\frac{cx^r}{ca^0}\right) \leqq -l$$
$$\text{i. e., } \left(\frac{cx^r}{ca^0}\right) \leqq 2^{-l}$$

The computation in each step involves $\mathcal{O}(n^3)$ arithmetic operations on the data in the worst case. By Theorem 11.5 and the termination conditions used in the algorithm it has to run for at most $\mathcal{O}(nL)$ steps, to come within $2^{-\mathcal{O}(L)}$ of the optimum, at which point we round the solution to get an exact optimum solution as discussed in Subsection 5. So, the algorithm needs at most $\mathcal{O}(n^4L)$ arithmetical operations on the data in the worst case, it is clearly polynomially bounded.

The final operation of converting the near optimal solution obtained at the termination of the algorithm into an exact optimal solution as discussed in Subsection 5 could be computationally expensive (it may need up to $\mathcal{O}(n)$ pivot steps). In most practical applications the data usually consists of unknown error terms and it makes sense to take the near optimal solution as it is, without the expensive final conversion. In practical LP applications, because of unknown errors in the data, a near optimal and aproximately feasible solution to the model is the usual goal, and Karmarkar's algorithm is well suited to achieve this goal.

7 Efficient Implementation of the Karmarkar Algorithm

The major piece of computation in each step of the algorithm is the computation of the projection $\hat{c}_p = \hat{c}[I - B^T(BB^T)^{-1}B]$. For this we have to find the inverse, $(BB^T)^{-1}$. Since $B = \begin{pmatrix} AD \\ e \end{pmatrix}$, we have

$$BB^T = \begin{pmatrix} AD^2A^T & ADe \\ (ADe)^T & e^TDe \end{pmatrix} = \begin{pmatrix} AD^2A^T & 0 \\ 0 & 1 \end{pmatrix}$$

since the point a used in defining the diagonal matrix D is in $\mathbf{S}$, and $a^0 = \frac{e}{n} \in \Omega''$. $(BB^T)^{-1}$ can be found efficiently if $(AD^2A^T)^{-1}$ can be. The only thing that changes in AD^2A^T from step to step is the diagonal matrix D. Let $D_r = \text{diag}(d_{11}^r, \ldots, d_{nn}^r)$ denote the diagonal matrix D in step r. We do not compute $(AD^2A^T)^{-1}$ in each step from scratch. Instead we update it to the extent necessary as we move from one step to the next.

If D_r and D_{r+1} differ in only one entry, the inverse of $AD_{r+1}^2A^T$ can be computed in $\mathcal{O}(n^2)$ arithmetic operations from $AD_r^2A^T$. For this, consider a nonsingular square matrix M of order n, $u = (u_1, \ldots, u_n)^T$, $v = (v_1, \ldots, v_n)^T$. Then the Sherman-Morrison formula states that

$$(M + uv^T)^{-1} = M^{-1} - \frac{(M^{-1}u)(M^{-1}v)^T}{1 + u^TM^{-1}v}$$

uv^T is a rank-one modification of M, and the formula shows that computation of $(M+uv^T)^{-1}$ can be done with $\mathcal{O}(n^2)$ arithmetical operations given M^{-1}. If D_r and D_{r+1} differ in only the i^{th} diagonal entry, then

$$AD_{r+1}A^T = AD_r^2A^T + \left(\left(d_{ii}^{r+1}\right)^2 - \left(d_{ii}^r\right)^2\right) A_{.i}(A_{.i})^T \ .$$

So, in this case $AD_{r+1}^2A^T$ is obtained from a rank-one modification of $AD_r^2A^T$, and the above formula can be used to get $(AD_{r+1}^2A^T)^{-1}$ from $(AD_r^2A^T)^{-1}$ with $\mathcal{O}(n^2)$ arithmetical operations. If D_r and D_{r+1} differ in t diagonal entries, we can perform t successive rank-one updates as above and obtain $(AD_{r+1}^2A^T)^{-1}$ from $(AD_r^2A^T)^{-1}$ with $\mathcal{O}(n^2t)$ arithmetical operations.

We now show that with a simple modification of the algorithm, we get a version in which $(AD_r^2A^T)^{-1}$ can be used in place of $(AD_{r+1}^2A^T)^{-1}$ as long as D_r and D_{r+1} are close in some sense.

We define the diagonal matrix $\overline{D} = \text{diag}(\overline{d}_{11}, \ldots, \overline{d}_{nn})$ as an approximation to $D_{r+1} = \text{diag}\ (d_{11}^{r+1}, \ldots, d_{nn}^{r+1})$ if

$$\frac{1}{2} \leqq \left(\frac{\bar{d}_{ii}}{d_{ii}^{r+1}}\right)^2 \leqq 2 \text{ for all } i \ .$$

We will now analyse the effect of replacing D_{r+1} by such a $\overline{D}$. Consider the following modification of the optimization problem over the inscribed sphere in the transformed space.

$$\begin{array}{lll} \text{minimize} & \hat{c}x' & \\ \text{subject to} & x' \in \Omega'' & \\ \text{and} & h(x') = (x'-a^0)^TQ(x'-a^o) \leqq \bar{\alpha}\underline{\rho} & \end{array} \tag{11.18}$$

where Q is some positive diagonal matrix. Taking $Q = I$ and $\bar{\alpha} = \alpha$ corresponds to the original problem used in Subsection 6.

Letting the row vector π, and scalar μ to be the Lagrange multipliers for (11.18), the KKT conditions for (11.18) imply

$$\hat{c} - \pi B + 2\mu(x'-a^0)^TQ = 0$$

$$\begin{aligned} \therefore \quad \hat{c}Q^{-1}B^T &= \pi BQ^{-1}B^T - 2\mu(x'-a^0)^TB^T \\ &= \pi BQ^{-1}B^T \end{aligned}$$

since both $x', a^0 \in \Omega''$ implies that $B(x'-a^0) = 0$. Using this we conclude that the optimum solution of (11.18), x', satisfies $(x'-a^0)^T = \gamma\hat{c}\big(I - Q^{-1}B^T(BQ^{-1}B^T)^{-1}B\big)Q^{-1}$ where γ is a positive scalar to be determined so that x' satisfies $(x'-a^0)^TQ(x'-a^0) = \bar{\alpha}\underline{\rho}$. Computation of this requires $(BQ^{-1}B^T)^{-1}$. Substituting $B = \begin{pmatrix} AD_{r+1} \\ e^T \end{pmatrix}$ we get

$$BQ^{-1}B^T = \begin{pmatrix} AD_{r+1}Q^{-1}D_{r+1}A^T & AD_{r+1}Q^{-1}e \\ AD_{r+1}Q^{-1}e^T & e^TQ^{-1}e \end{pmatrix} \ .$$

If the inverse of $AD_{r+1}Q^{-1}D_{r+1}A^T$ is known, $(BQ^{-1}B^T)^{-1}$ can be computed with $\mathcal{O}(n^{\mathbf{2}})$ arithmetical operations using the formula

$$\begin{pmatrix} M & p \\ p^T & q \end{pmatrix} = \frac{1}{q - p^T M^{-1} p} \begin{pmatrix} (q - p^T M^{-1} p)M^{-1} + (M^{-1}p)(M^{-1}p)^T & \vdots & -M^{-1}p \\ -(M^{-1}p)^T & \vdots & 1 \end{pmatrix}$$

Suppose $\overline{D} = D_{r+1}E$ where E is a diagonal error matrix such that $E = (e_{ij})$ with $\frac{1}{2} \leqq e_{ii}^{\mathbf{2}} \leqq 2$ for all i, and we know $(A\overline{D}A^T)^{-1}$. Then setting $Q = E^{-\mathbf{2}}$, we have $AD_{r+1}Q^{-1}D_{r+1}A^T = A\overline{D}A^T$. So using the known $(A\overline{D}A^T)^{-1}$, we can compute the optimum solution of the modified problem (11.18) using the above formulae.

Now we relate the solution of (11.18) to the main optimization problem. Since $Q_{ii} = e_{ii}^{-\mathbf{2}} \in \left[\frac{1}{2}, 2\right]$, we have

$$\frac{1}{2}(x' - a^0)^T(x' - a^0) \leqq (x' - a^0)^T Q (x' - a^0) \leqq 2(x' - a^0)^T(x' - a^0)$$

$$\begin{aligned} \mathbf{B}\Big(a^0, (\frac{\bar{\alpha}}{2})\underline{\rho}\Big) &\subseteq \{x' : (x' - a^0)^T Q (x' - a^0) \leqq \bar{\alpha}\underline{\rho}\} \\ &\subseteq \mathbf{B}(a^0, 2\bar{\alpha}\underline{\rho}) \ . \end{aligned}$$

Take $\bar{\alpha} = \frac{\alpha}{2}$ where α is the quantity used in Subsection 6 (there, we used typically $\alpha = \frac{1}{4}$). So

$$\begin{aligned} \mathbf{B}\Big(a^0, (\frac{\alpha}{4})\underline{\rho}\Big) \cap \Omega'' &\subseteq \{x' : x' \in \Omega'' \text{ and } (x' - a^0)^T Q (x' - a^0) \leqq \bar{\alpha}\underline{\rho}\} \\ &\subseteq \mathbf{B}(a^0, \alpha\underline{\rho}) \cap \Omega'' \ . \end{aligned}$$

From the first inclusion we have

$$\begin{array}{l} \text{minimum value of } f'(x') \\ \text{subject to } x' \in \Omega'' \\ \text{and } (x' - a^0)^T Q (x' - a^0) \leqq \bar{\alpha}\underline{\rho} \end{array} \qquad \leqq \qquad \begin{array}{l} \text{minimum value of } f'(x') \\ \text{subject to } x' \in \Omega'' \cap \mathbf{B}\Big(a^0, (\frac{\alpha}{4})\underline{\rho}\Big) \end{array}$$

and by Theorem 11.2 we have

$$\begin{array}{l} \text{minimum value of } f'(x') \\ \text{subject to } x' \in \Omega'' \cap \mathbf{B}\Big(a^0, (\frac{\alpha}{4})\underline{\rho}\Big) \end{array} \qquad \leqq \qquad f'(a^0) - \log\big(1 + \frac{\alpha}{4}\big) \ .$$

So, for $\bar{g}'$, the optimum solution corresponding to the modified problem (11.18), we can claim

$$f'(\bar{g}') \leqq f'(a^0) - \log\big(1 + \frac{\alpha}{4}\big)$$

and if we define $\bar{x}^{r+1} = T^{-1}(\bar{g}')$, we can as in Theorem 11.4, claim

$$f(\bar{x}^{r+1}) \leqq f(x^r) - \bar{\delta}$$

where $\bar{\delta}$ is redefined as

$$\bar{\delta} = \log\big(1 + \frac{\alpha}{4}\big) - \frac{\beta^{\mathbf{2}}}{2(1-\beta)^{\mathbf{2}}} \ .$$

This affects the number of steps by only a constant factor and the algorithm still works.

So, this is what we do, to implement the modified algorithm in an efficient manner. We maintain $(AD^{\mathbf{2}}A^T)^{-1}$. We do not change all diagonal elements of D in each step. Let $y = (y_1, \ldots, y_n)^T$ be the new solution at the end of a step. It is time to update $(AD^{\mathbf{2}}A^T)^{-1}$. Before, we defined the new D to be diag $(y_1, \ldots, y_n)$. Instead, we modify D in two stages.

Compute $\sigma = \frac{1}{n}\sum_{j=1}^{n} \frac{y_j}{d_{jj}}$ where d_{jj} are the diagonal entries in the current D. First multiply D by σ, this needs dividing $(AD^{\mathbf{2}}A^T)^{-1}$ by $\sigma^{\mathbf{2}}$ to update it accordingly. This completes stage 1.

Then, for each $j = 1$ to n, if in the matrix D at the end of stage 1, $\left(\frac{d_{jj}}{y_j}\right)^{\mathbf{2}} \notin \left[\frac{1}{2}, 2\right]$, reset $d_{jj} = y_j$ and update $(AD^{\mathbf{2}}A^T)^{-1}$ corresponding to this change by a rank-one modification as discussed above.

In essence, we carry out fewer updating operations by optimizing (after the projective transformation) over an inscribed ellipsoid (dashed in Figure 11.5) and not the inscribed sphere. (Of course we do not optimize over this sphere or ellipsoid exactly, but scale it by α or $\bar{\alpha}$ before the optimization.) We make enough updating operations to make sure that the current D matrix and current solution y always satisfy $\left(\frac{d_{jj}}{y_j}\right)^{\mathbf{2}} \in \left[\frac{1}{2}, 2\right]$, this insures that the ellipsoid is close to the inscribed sphere

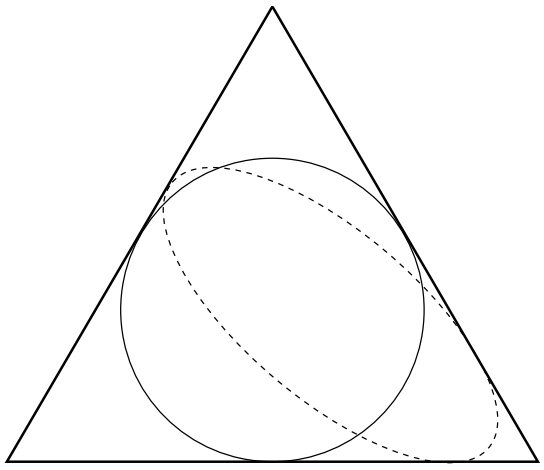

Figure 11.5

We still need only $\mathcal{O}(nL)$ steps to shrink the objective value by the required factor of $2^{-\mathcal{O}(L)}$. With this modification, N. Karmarkar has shown in [11.2] that we need to do only $\mathcal{O}(n^{\frac{\mathbf{3}}{\mathbf{2}}}L)$ updating operations. Since each updating operation requires $\mathcal{O}(n^{\mathbf{2}})$ arithmetic operations on the data, the overall algorithm needs $\mathcal{O}(n^{\mathbf{3.5}}L)$ arithmetic operations on the data in the worst case, with this modification.

8 The Sliding Objective Function Method

From Subsection 6, it is clear that Karmarkar's algorithm solves LPs for which the optimum objective value is known to be zero. As shown in Subsection 1, any LP can be transformed into one with this property, but this transformation increases the number of constraints and blows up the order of the problem, and hence may be undesirable in practical applications. In this subsection, we discuss a sliding objective value approach that can be used to solve the original problem by itself using Karmarkar's algorithm, when the optimum objective value is unknown.

For a given LP, the first problem is to determine whether it is feasible or not. Let the system of constraints be

$$\begin{aligned} Ax &= b \\ x &\geqq 0 \end{aligned}$$

where A is of order $m \times n$. As shown in Subsection 2, to check whether this system is feasible, we solve the following LP with the artificial variable x_{n+1}. Let $x^0 > 0$ be any vector.

$$\begin{array}{ll} \text{minimize} & x_{n+1} \\ \text{subject to} & Ax - x_{n+1}(Ax^0 - b) = b \\ & x \geqq 0, \quad x_{n+1} \geqq 0 \end{array} \tag{11.19}$$

$(x^0, 1) > 0$ is a feasible solution to this problem. The original problem is feasible iff the optimum objective value in this problem is zero. Even though the exact optimum objective value in this problem is unknown, we know that it lies between 0 and 1. Using it, this problem could be solved by Karmarkar's algorithm with the sliding objective value approach discussed below.

Now consider the general LP

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & Ax = b \\ & x \geqq 0 \end{array} \tag{11.20}$$

This problem can be solved in two stages. First we check whether it is feasible, as discussed above. If a feasible solution $\bar{x}$ is obtained, $c\bar{x}$ is an upper bound on the optimum objective value in (11.20). We could then check whether the dual problem is feasible. If the dual is infeasible, from the duality theory of linear programming we know that cx is unbounded below in (11.20) (since (11.20) has already been verified to be feasible). If the dual is feasible, the dual objective value at the dual feasible solution obtained is lower bound on the optimum objective value in (11.20).

Now, consider the LP in the form discussed in (11.11)

$$\begin{array}{ll} \text{minimize} & dx \\ \text{subject to} & x \in \Omega \cap \mathbf{S} \\ \text{where} & \Omega = \{x : Ax = 0\} \\ & \mathbf{S} = \{x : x \geqq 0, \sum_{j=1}^{n} x_j = 1\} \end{array} \tag{11.21}$$

where A is a matrix of order $m \times n$ and rank m. We assusme that an optimum solution exists and that the optimum objective value is known to be between the given lower and upper bound l_0, u_0 (if the original problem is transformed directly into this form using the techniques discussed in Subsections 2, 3, we could take $l_0 = -2^{\mathbf{L}}$ and $u_0 = 2^{\mathbf{L}}$, where L is the size of the problem, under the assumption that an optimum solution exists). The difference between the current lower and upper bounds on the objective value is called the range. The sliding objective value approach is divided into several phases. At the end of each phase the range reduces to at least $\frac{2}{3}$ of its length at the beginning of the phase and takes no more than $n(k+\log(n))$ steps where k is a constant satisfying

$$\left(1 - \frac{\delta}{n}\right)^{\mathbf{kn}} \leqq \frac{1}{2} \ .$$

Let z^* denote the unknown optimum objective value in (11.21). We run the algorithm pretending that a selected value, $\bar{z}$ is the minimum objective value (the value of $\bar{z}$ is updated at the beginning of each phase), that is, we try to minimize $dx - \bar{z} = (d - \bar{z}e^T)x$. This leads to the problem

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & x \in \Omega \cap \mathbf{S} \end{array}$$

with $c = d - \bar{z}e^T$. We need to modify the computation of the vector g' in each step of the algorithm as follows. Compute g' as in the subroutine discussed in Subsection 6. Check if $\hat{c}g' < 0$. If so, choose the point g'' on the line segment joining a^0 and g' which satisfies $\hat{c}g'' = 0$, and make the point g'' the output of the subroutine instead of g'.

If $z^* \leqq \bar{z}$, let x^m be the point where $\hat{c}x$ achieves its minimum over $\Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$. If $\hat{c}x^m < 0$, then define x^* to be the point on the line segment joining a^0 and x^m satisfying $\hat{c}x^* = 0$. Then all the proofs go through, and each step of the algorithm leads to a reduction of δ in the potential function or finds a point where the original objective function is $\bar{z}$.

Now a phase in the sliding objective value approach consists of the following. Let l, u be the current lower and upper bounds for the objective value dx at the beginning of the phase. Let

$$\bar{l} = l + \frac{1}{3}(u - l)$$
$$\bar{u} = l + \frac{2}{3}(u - l)$$

Figure 11.6

Run the algorithm as described above with $\bar{z}$ = pretended minimum objective value = $\bar{l}$.

If we obtain a feasible solution x which satisfies $dx < \bar{u}$, then terminate the phase, make dx the new upper bound u, and go to the next phase with the new bounds for the objective value.

Suppose after $n(k + \log(n))$ steps we have not reached a solution x with $dx < \bar{u}$. If $z^* \leqq \bar{l}$, we must have achieved a reduction δ in the associated potential function in each step, forcing the objective value dx to be $< \bar{u}$. So, if after $n(k + \log(n))$ steps we have not reached a solution x with $dx < \bar{u}$, we must have $z^* \geqq \bar{l}$. So make $\bar{l}$ the new lower bound l, and go to the next phase with the new bounds for the objective value.

Thus the length of the range gets multiplied by a factor $\frac{2}{3}$ or less during each phase. So after $\mathcal{O}(L)$ phases (i. e., after $\mathcal{O}(nL \log n)$ steps) we narrow the range to within $2^{-\mathcal{O}(L)}$ of the optimum objective value, and then obtain the exact optimum solution from the solution at that stage.

9 Implementation Difficulties

Consider the LP in standard form, find $y \in \mathbf{R}^n$ to

$$\begin{array}{ll} \text{minimize} & gy \\ \text{subject to} & Gy = d \\ & y \geqq 0 \ . \end{array} \tag{11.22}$$

The primal simplex algorithm for solving this problem processes the problem as it is in (11.22). It performs a sequence of operations on the data G, d, g until the problem is solved.

To solve (11.22) by Karmarkar's algorithm in the form discussed in Subsection 6, we have to first convert the problem into the form (11.11). As pointed out in Subsection 3, we add the additional constraint

$$\sum_{j=1}^{n} y_j + y_{n+1} = M \ .$$

Mathematically, taking M to be $2^{\mathbf{L}}$ where L is the size of the LP (11.22), would suffice; but in practical implementations M could be any practically reasonable upper bound for $\sum_{j=1}^{n} y_j$ in the problem. Using this additional constraint, (11.22) is transformed into the form

$$\begin{array}{ll} \text{minimize} & gy \\ \text{subject to} & Gy - \left(\frac{1}{M}\right) d \left(\sum\limits_{j=1}^{n+1} y_j \right) = 0 \\ & \sum\limits_{j=1}^{n+1} y_j = 1 \\ & y \geqq 0, j = 1 \text{ to } n+1 \end{array} \tag{11.23}$$

which is in Karmarkar's form.

LP models arising in practical applications lead to problems of the form (11.22) in which the coefficient matrix G is very sparse, that is, most of the entries in it are zero. Commercial implementations of the primal simplex algorithm exploit this sparsity and are able to take tremendous advantage of it. When the problem is transformed into the form (11.23) as discussed above, the resulting coefficient matrix A is usually totally dense, that is, almost all the entries in it are nonzero. This makes it very difficult to produce a practically viable implementation of Karmarkar's algorithm, at least for the algorithm in the form that is stated above. One may be able to overcome this problem by not computing A explicitly, but storing it as $G - \left(\frac{1}{M}\right)de^T$.

Now, consider the LP in the following form

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & Ax = 0 \\ & e^T x = 1 \\ & x \geqq 0 \end{array} \tag{11.24}$$

The primal simplex algorithm would solve (11.24) by performing operations on the constraint matrix A directly. Karmarkar's algorithm operates on AA^T or $AD^{\mathbf{2}}A^T$ where D is a positive diagonal matrix. The computation of this matrix product is an additional burden in Karmarkar's algorithm. In fact an implementation of Karmarkar's algorithm which maintains $(AD^{\mathbf{2}}A^T)^{-1}$ in any form and updates it exactly from step to step in the algorithm, is not likely to be competitive with efficient implementations of the primal simplex algorithm.

Let D_r denote the diagonal matrix in step $r+1$ of Karmarkar's algorithm applied to (11.24). The computations (as discussed in Subsections 6, 7) in this step of the algorithm can be carried out by doing the following.

First solve the following system of equations for the row vector of variables $u = (u_1, \ldots, u_m)$

$$u(AD_r^{\mathbf{2}}A^T) = cD_r^{\mathbf{2}}A^T \ . \tag{11.25}$$

Let u^r denote the exact solution of this system. Then compute the $1 \times n$ row vector $\hat{c}_p^r$ from

$$\hat{c}_p^r = cD_r - u^r AD_r - cD_r ee^T \ .$$

This $(\hat{c}_p^r)^T$ is the direction for moving from a^0 to the boundary of the sphere $\mathbf{B}(a^0, \alpha\underline{\rho})$ in this step. It provides the steepest descent direction for minimizing the linear function $\hat{c}x'$ over $\Omega'' \cap \mathbf{B}(a^0, \alpha\underline{\rho})$ in this step. See Figure 11.7. In reality, we dot not need $\hat{c}_p^r$ exactly. Any approximate vector $\bar{c}_p^r$ that makes a strict acute angle will be adequate (the closer this angle is to 0 the better), it produces a decrease in objective function which may suffice in practice.

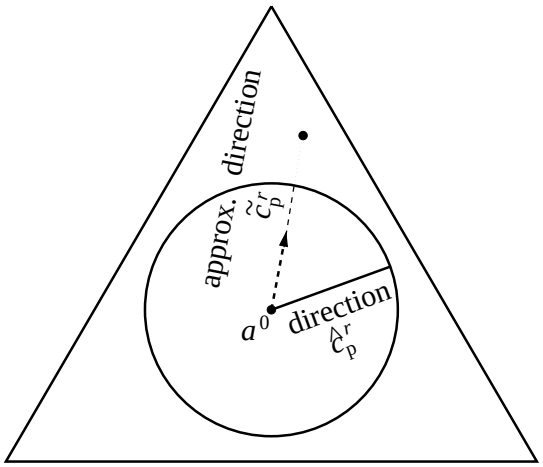

Figure 11.7 Steepest descent direction $\hat{c}_p^r$ for linear objective funtion $\hat{c}x$ in step r. Approximate descent direction $\tilde{c}_p^r$.

The key point is to get an approximate solution $\bar{u}^r$ for (11.25) efficiently, so that if

$$\tilde{c}_p^r = cD_r - \bar{u}^r AD_r - (cD_r ee^T)$$

that would satisfy $\tilde{c}_p^r \hat{c}_p^r > 0$ (acute angle condition). We also need $AD_r \tilde{c}_p^r = 0$ and $e^T \tilde{c}_p^r = 0$, so that moving from a^0 in the direction $\tilde{c}_p^r$ keeps the point within Ω''. Also given the approximate $\bar{u}^r$, how to update it into $\bar{u}^{r+1}$ that works for the $(r+1)$ in step the same way, when D_r changes to D_{r+1}. Some iterative methods for solving linear equations that produce approximate solutions efficiently may provide the key to this computation, and these are being investigated.

Also, once the direction of movement $\tilde{c}_p^r$ is obtained, in practical implementations one may want to move all the way closer to the boundary of the simplex, rather than to the boundary of the insphere $\mathbf{B}(a^0, \alpha\underline{\rho})$ as indicated in Figure 11.7. Since the simplex is determined by linear constraints, this can be done efficiently through a minimum ratio computation to determine how far you can move in this direction while retaining feasibility, and you can stop just a little bit short of it.

These and various other ideas are being explored for producing a practically useful implementation of Karmarkar's algorithm.

10 Solving Quadratic and Convex Programs by Karmarkar's Approach

It should be possible to extend Karmarkar's algorithm to solve convex quadratic programs and LCPs associated with PSD matrices, and possibly even smooth nonlinear convex programming problems. These extensions, and the best implementations of them, are now active research topics.

11.4.2 Tardos' New Strongly Polynomial Minimum Cost Circulation Algorithm

Consider a directed single commodity flow capacitated network with n nodes and m arcs. In [11.7] E. Tardos developed an algorithm for finding a minimum cost circulation in this network, with worst case computational complexity of $\mathcal{O}(m^{\frac{3}{2}}n^5)$ or $\mathcal{O}(m^2n^3 \log m)$ with some improvements. She has applied the idea of this algorithm and developed an algorithm to solve the general linear programming problem

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & Ax = b \\ & x \geqq 0 \end{array}$$

in time polynomial in the size of A.

It remains to be investigated whether this approach can be extended to solve LCPs (q, M) when M is PSD, in time polynomial in the size of M.

11.4.3 The Ellipsoid Method for Linear Programming

A version of the ellipsoid method for solving linear programming problems is presented in Chapter 15 of [2.26]. The approach outlined there, uses a scheme suggested by P. Gacs and L. Lovász in a terminal step in order to obtain an optimum solution of the LP. Here we show how that terminal step can be replaced by a much more efficient scheme similar to the one discussed in Subsection 5 of Section 11.4.1. This has been suggested by R. Chandrasekaran and K. Truemper.

Consider an LP with rational data. By the techniques discussed in Section 1.2 and by scaling, this LP can be transformed into the problem

$$\min\ cu, \quad Fu \geqq g, \quad u \geqq 0 \tag{11.26}$$

where F, g, c are integer matrices. Let v denote the column vector of dual variables. By the duality theorem of linear programming (also see Subsection 1 of Section 11.4.1) solving this LP is equivalent to solving the system of linear inequalities (11.27).

$$\begin{array}{rr} -Fu & \leqq -g \\ F^Tv & \leqq c^T \\ cu\ -g^Tv & \leqq 0 \\ -u & \leqq 0 \\ -v & \leqq 0 \end{array} \tag{11.27}$$

Let $x = \begin{pmatrix} u \\ v \end{pmatrix}$. The system (11.27) is a system of linear inequalities in which all the coeficients are integer. Let D, b denote the coefficient matrix and right hand

side constants vector in (11.27), including the sign restrictions on the variables. Then (11.27) can be written as

$$Dx \leqq b \ . \tag{11.28}$$

Let D be of order $m \times n$, and L be the size of (11.28) (that is, L is the total number of binary digits in all the data in (11.28), see Section 8.3). Let $L_1 = 3\big((m+1)(n+1)+1\big)L$. As in Section 15.3 of [2.26], consider the perturbed system

$$2^{\mathbf{L_1}}(D_i.x) < 2^{\mathbf{L_1}}b_i + 1, \quad i = 1 \text{ to } m \tag{11.29}$$

(11.29) is now an open system of linear inequalities with integer data, and hence it can be solved by the ellipsoid method discussed in Section 15.2 of [2.26] in polynomial time. The method begins with an arbitrary point $x^0 \in \mathbf{R}^n$, and the matrix $A_0 = 2^{\mathbf{2(1+L_1)}}I$, where I is the unit matrix of order n, and generates the sequence (x^r, A_r), $r = 1, 2, \ldots$ using the iterative scheme (8.7) discussed in Section 8.4. For some r, if x^r satisfies (11.29), define $\tilde{x}$ to be that feasible x^r and go to the terminal step discussed below. If x^r violates (11.29), find a constraint in (11.29) violated by x^r, suppose it is the p^{th} constraint in (11.29). Then define $a = 2^{\mathbf{L_1}}D_p.$ and $d = 1 + 2^{\mathbf{L_1}}b_p$, and compute γ_{r+1} as in (8.7) using this a, d, x^r and A_r. If $\gamma_{r+1} \leqq -1$, (11.29) is infeasible, terminate the ellipsoid algorithm. If $\gamma_{r+1} > -1$, compute x^{r+1}, A_{r+1} as in (8.7), and continue.

If the ellipsoid algorithm continues for $r = 0$ to $6(n+1)^{\mathbf{2}}(m+1)(n+1)(L+L_1)$ steps and all the points x^r obtained in the algorithm are infeasible to (11.29), terminate with the conclusion that (11.29) has no feasible solution. The proofs of this and the other infeasibility conclusion stated earlier, are given in Chapter 15 of [2.26]. Under this infeasibility termination, (11.28), that is, (11.27), has no feasible solution, this implies that either the LP (11.26) is infeasible, or it is feasible and the objective function is unbounded below on its set of feasible solutions, we terminate.

Otherwise, let $\tilde{x}$ be the feasible solution for (11.29) obtained by the ellipsoid algorithm discussed above. If $\tilde{x} = \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix}$ is feasible to (11.28), then $\tilde{u}$ is an optimal solution of (11.26) and $\tilde{v}$ is an optimal dual solution, terminate. If not, consider the following system.

$$Dx^+ - Dx^- + It = b + e2^{-\mathbf{L_1}} \tag{11.30}$$

where $x^+ = (x_1^+, \ldots, x_n^+)$, $x^- = (x_1^-, \ldots, x_n^-)$, $t = (t_i) \in \mathbf{R}^m$ and e is the column vector in $\mathbf{R}^m$ of all 1's. Define for $j = 1$ to n

$$\tilde{x}_j^+ = \begin{cases} 0 & \text{if } \tilde{x}_j \leqq 0 \\ \tilde{x}_j & \text{if } \tilde{x}_j \geqq 0 \end{cases}$$

$$\tilde{x}_j^- = \begin{cases} |\tilde{x}_j| & \text{if } \tilde{x}_j \leqq 0 \\ 0 & \text{if } \tilde{x}_j \geqq 0 \end{cases}$$

$$\tilde{t}_i = -D_i.\tilde{x} + b_i + 2^{-\mathbf{L_1}}$$

$$\tilde{x}^+ = (\tilde{x}_j^+), \quad \tilde{x}^- = (\tilde{x}_j^-), \quad \tilde{t} = (\tilde{t}_i)$$

Then $(\tilde{x}^+, \tilde{x}^-, \tilde{t})$ is feasible to (11.30). Using the method discussed in Subsection 5 of Section 11.4.1, or Section 3.5.4 of [2.26] (here there is no objective function involved, so

we just apply this method without worrying about the objective value), obtain a BFS $(\hat{x}^+, \hat{x}^-, \hat{t}) \in \mathbf{R}^{2n+m}$ to (11.30). Denote the vector $(x^+, x^-, t) \in \mathbf{R}^{2n+m}$ by y and let $\hat{y} = (\hat{x}^+, \hat{x}^-, \hat{t})$. Since $\hat{y}$ is a BFS of (11.30), there exists a basis B, a square submatrix of $(D \vdots -D \vdots I)$ of order m, so that $\hat{y} = (\hat{y}_B, \hat{y}_E)$ is given by

$$\begin{aligned} \hat{y}_E &= 0 \\ \hat{y}_B &= B^{-1}(b + e2^{-\mathbf{L_1}}) \ . \end{aligned} \tag{11.31}$$

Here E is the submatrix of $(D \vdots -D \vdots I)$ consisting of all the columns other than those in B, and y_B, y_E are the basic, nonbasic vectors of variables y_j corresponding to the basic, nonbasic partition $(B \vdots E)$ of $(D \vdots -D \vdots I)$. Now define the vector $y^* = (y_B^*, y_E^*)$ by

$$\begin{aligned} y_E^* &= 0 \\ y_B^* &= B^{-1}b \end{aligned} \tag{11.32}$$

and let $y^* = (x^{*+}, x^{*-}, t^*)$ in terms of the original variables. Let

$$x^* = x^{*+} - x^{*-} \ .$$

The vector y^* is the basic solution of the system

$$Dx^+ - Dx^- + It = b \tag{11.33}$$

corresponding to the basis B. By Theorem 15.1 of [2.26], |determinant of B| $< 2^{\mathbf{L}}$, and hence using an argument similar to that in Theorem 15.2 of [2.26] we have, for $i = 1$ to m

$$\text{either} \quad t_i^* = 0 \quad \text{or } |t_i^*| > 2^{-\mathbf{L}} \ . \tag{11.34}$$

Let $\mathbf{J} = \{i : 1 \leqq i \leqq m$, and i such that t_i is a basic variable corresponding to the basis $B\}$. So, from the definition of $\hat{y}$, and from (11.31), (11.32), we have

$$\hat{t}_i = \begin{cases} 0, & \text{for all } i \notin \mathbf{J} \\ t_i^* + (B^{-1}e2^{-\mathbf{L_1}})_i, & \text{for } i \in \mathbf{J} \ . \end{cases} \tag{11.35}$$

From well known results in the theory of determinants, B^{-1} is the adjoint of B multiplied by a scalar, which is the inverse of the determinant of B. The determinant of the basis B is a nonzero integer and hence has absolute value $\geqq 1$. Each entry in the adjoint of B is the determinant of a square submatrix of B, by Theorem 15.1 of [2.26] its absolute value is $\leqq \frac{2^{\mathbf{L}}}{n}$. So $|(B^{-1}e2^{-\mathbf{L_1}})_i| \leqq \frac{m2^{\mathbf{L}}}{n2^{\mathbf{L_1}}} \leqq \frac{2^{\mathbf{L}}}{2^{\mathbf{L_1}}} \leqq 2^{-\mathbf{nL}}$. But $\hat{y}$ is a BFS of (11.30), so $\hat{t}_i \geqq 0$ for all i. Using this and (11.34) in (11.35) we conclude that t_i^* must be $\geqq 0$ for all $i \in \mathbf{J}$. We already know that $t_i^* = 0$ for all $i \notin \mathbf{J}$. So $t^* \geqq 0$. This clearly implies that x^* is feasible to (11.28). Therefore if $x^* = \begin{pmatrix} u^* \\ v^* \end{pmatrix}$, u^* is an optimum solution of (11.26) and v^* is an optimum dual solution. From u^*, a

basic feasible optimum solution of (11.26) can be obtained by the method described in Subsection 5 of Section 11.4.1.

The ellipsoid method is the first mathematical device used to prove that linear programming is in the class $\mathcal{P}$ of problems solvable in polynomial time. The modified terminal step given above is not adequate to make the ellipsoid method practically useful. However, the ellipsoid method remains a very important mathematical tool in the study of computational complexity of optimization problems.

11.4.4 The Gravitational Method for Linear Programming

Here we briefly describe an interior point variant of the gradient projection method for linear programming proposed by K. G. Murty [11.5, 11.6]. We consider the LP in the following form

$$\begin{array}{ll} \text{minimize} & z(x) = cx \\ \text{subject to} & Ax \geqq b \end{array} \tag{11.36}$$

where A is a matrix of order $m \times n$. Sign restrictions on the variables and any other lower or upper bound conditions on the variables, if any, are all included in the above system of constraints. Clearly every LP can be put in this form by well known simple transformations discussed in most LP textbooks (for example, see [2.26]).

Note: In practical applications, it usually turns out that the LP model for a practical problem is in standard form

$$\begin{array}{c} \min\ p\chi \\ \text{subject to } B\chi = d \\ \chi \geqq 0. \end{array} \tag{11.37}$$

The dual of this model is directly in form (11.36) and the gravitational method can be applied to solve the dual of (11.37) directly. As it will be shown later on, when the gravitational method is applied on the dual of (11.37), at termination, it will produce an optimum solution for (11.37), if one exists.

Assumptions

Let $\mathbf{K}$ denote the set of feasible solutions of (11.36). We assume that $\mathbf{K} \neq \emptyset$, and that $\mathbf{K}$ has a nonempty interior in $\mathbf{R}^n$, and that an initial interior feasible solution x^0 (this is a point x^0 satisfying $Ax^0 > b$) of (11.36) is available.

If these assumptions are not satisfied, introduce an artificial variable x_{n+1} and modify the problem as follows

$$\begin{array}{ll} \text{minimize} & cx + vx_{n+1} \\ \text{subject to} & Ax + ex_{n+1} \geqq b,\ x_{n+1} \geqq 0 \end{array} \tag{11.38}$$

where $e = (1, \ldots, 1)^T \in \mathbf{R}^m$ and v is a large positive number. For any $\hat{x} \in \mathbf{R}^n$, let $\hat{x}_{n+1} > \max\{|\min\{0, A_i.\hat{x} - b_i\}| : i = 1 \text{ to } m\}$, then $(\hat{x}, \hat{x}_{n+1})$ satisfies the constraints in (11.38) as strict inequalities. Thus the modified problem (11.38) satisfies all the assumptions made in the above paragraph.

We also assume that $c \neq 0$, as otherwise x^0 is optimal to (11.36), and we can terminate.

The Gravitational Method

The Euclidean distance of x^0 from the hyperplane $\{x : A_i.x = b_i\}$ is $(A_i.x^0 - b_i)/\|A_i.\|$.

The gravitational approach for solving (11.36) is the following. Assume that the boundary of $\mathbf{K}$ is an impermeable layer separating the inside of $\mathbf{K}$ from the outside. Introduce a powerful gravitational force inside $\mathbf{K}$ pulling everything down in the direction $-c^T$. Choose $0 < \varepsilon < \min\{(A_i.x^0 - b_i)/\|A_i.\| : i = 1 \text{ to } m\}$. Release a small spherical n-dimensional drop of mercury of diameter 2ε with its center at the initial interior feasible solution $x^0 \in \mathbf{K}$. The drop will fall under the influence of gravity. During its fall, the drop may touch the boundary, but the center of the drop will always be in the interior of $\mathbf{K}$ at a distance $\geqq \varepsilon$ from the nearest point to it on the boundary. Whenever the drop touches a face of $\mathbf{K}$, it will change direction and will continue to move, if possible, in the gravitational direction that keeps it within $\mathbf{K}$. If the objective function is unbounded below in (11.36), after changing direction a finite number of times, the drop will continue to fall forever along a half-line in $\mathbf{K}$ along which the objective function diverges to $-\infty$. If $z(x)$ is bounded below on $\mathbf{K}$, after changing direction a finite number of times, the drop will come to a halt. The algorithm tracks the path of the center of the drop as it falls in free fall under the influence of gravity. Let $\mathbf{P}$ denote this path of the center of this drop in its fall.

The Gravitational Direction at an Interior Point $\overline{x} \in \mathbf{K}$

Suppose a drop of radius ε, with its center at $\overline{x}$ is inside $\mathbf{K}$. So

$$(A_i.\overline{x} - b_i)/\|A_i.\| \geqq \varepsilon, \quad i = 1 \text{ to } m. \tag{11.39}$$

At every point $\overline{x}$ on the locus $\mathbf{P}$ of the center of the drop in the gravitational method, (11.39) will always be satisfied. Given a point $\overline{x}$ on $\mathbf{P}$, define

$$\mathbf{J}(\overline{x}) = \{i : (A_i.\overline{x} - b_i)/\|A_i.\| = \varepsilon\}. \tag{11.40}$$

The hyperplane $\{x : A_i.x = b_i\}$ is touching the drop of radius ε when its center is at the interior point $\overline{x} \in \mathbf{K}$ only if $i \in \mathbf{J}(\overline{x})$. Now, define

$$y^0 = -c^T/\|c\|. \tag{11.41}$$

If $\mathbf{J}(\overline{x}) = \emptyset$ (i. e., if $(A_i.\overline{x} - b_i)/\|A_i.\| > \varepsilon$ for all $i = 1$ to m), when the drop is in a position with its center at $\overline{x}$, it will move in the gravitational direction y^0. The distance that it will move in this direction is

$$\theta = \text{ minimum } \left\{ \frac{(A_i.\overline{x} - b_i) - \varepsilon\|A_i.\|}{-A_i.y^0} : 1 \leqq i \leqq m \text{ and } i \text{ such that } A_i.y^0 < 0 \right\} \tag{11.42}$$

where we adopt the convention that the minimum in the empty set is $+\infty$. If $\theta = +\infty$ in (11.42), then the drop continues to move indefinitely along the half-line $\{\overline{x} + \lambda y^0 : \lambda \geqq 0\}$, and $z(x)$ is unbounded below on this feasible half-line, terminate. If θ is finite in (11.42), at the end of this move, the drop will be in a position with its center at $\overline{x} + \theta y^0$, touching the boundary of $\mathbf{K}$, and it will either halt (see the conditions for this, discussed later on) or change direction into the gravitational direction at $\overline{x} + \theta y^0$ and move in that direction.

When $\overline{x}$ is such that $\mathbf{J}(\overline{x}) \neq \emptyset$, that is,

$$\min\{(A_i.\overline{x} - b_i)/\|A_i.\| : i = 1 \text{ to } m\} = \varepsilon \tag{11.43}$$

the direction that the drop will move next, called the **gravitational direction** at $\overline{x}$, can be defined using many different principles. One principle to define the gravitational direction at $\overline{x}$, where $\overline{x}$ is an interior point of $\mathbf{K}$ satisfying (11.43) is by the following procedure, which may take several steps.

Step 1 : If the drop moves in the direction y^0 from $\overline{x}$, the position of its center will be $\overline{x} + \lambda y^0$ for some $\lambda > 0$. Since (11.39) holds, the ith constraint will block the movement of the drop in the direction y^0, only if $i \in \mathbf{J}(\overline{x})$ and $A_i.y^0 < 0$. Define

$$\mathbf{J}_1 = \{i : i \in \mathbf{J}(\overline{x}), \text{ and } A_i.y^0 < 0\}.$$

Case 1 : $\mathbf{J}_1 = \emptyset$: If $\mathbf{J}_1 = \emptyset$, y^0 is the gravitational direction at $\overline{x}$, and the distance it can move in this direction is determined as in (11.42).

Case 2 : $\mathbf{J}_1 \neq \emptyset$: If $\mathbf{J}_1 \neq \emptyset$, each of the constraints $A_i.x \geqq b_i$ for $i \in \mathbf{J}_1$, is currently blocking the movement of the drop in the direction y^0.

Define $\mathbf{T}_1 = \mathbf{J}_1$, and let D_1 be the matrix of order $|\mathbf{T}_1| \times n$ whose rows are $A_i.$ for $i \in \mathbf{T}_1$. Let E_1 be the submatrix of D_1 of order (rank of D_1) $\times\ n$, whose set of rows is a maximal linearly independent subset of row vectors of D_1. Let $\mathbf{I}_1 = \{i : A_i.$ is a row vector of $E_1\}$. So $\mathbf{I}_1 \subset \mathbf{T}_1$. Let $\mathbf{F}_1$ be the subspace $\{x : D_1x = 0\} = \{x : E_1x = 0\}$, F_1 is the subspace corresponding to the set of all constraints which are blocking the movement of the drop in the direction y^0. Let ξ^1 be the orthogonal projection of y^0 in the subspace F_1, that is

$$\xi^1 = (I - E_1^T(E_1E_1^T)^{-1}E_1)y^0. \tag{11.44}$$

Subcase 2.1 : $\xi^1 \neq 0$: If $\xi^1 \neq 0$, let $y^1 = \xi^1/\|\xi^1\|$, go to Step 2.

Subcase 2.2 : $\xi^1 = 0$: If $\xi^1 = 0$, let the row vector $\mu = (\mu_i : i \in \mathbf{I}_1) = -\|c\|((E_1E_1^T)^{-1}E_1y^0)^T$. Then $\mu E = c$.

Subcase 2.2.1 : $\xi^1 = 0$ and $\mu \geqq 0$: If $\mu \geqq 0$, define the row vector $\overline{\pi} = (\overline{\pi}_i)$ by

$$\begin{aligned}\overline{\pi}_i &= 0, \text{ if } i \notin \mathbf{I}_1\\ &= \mu_i, \text{ if } i \in \mathbf{I}_1.\end{aligned}$$

Then $\overline{\pi}$ is a basic feasible solution to the dual of (11.36). In this case, as will be shown later on, the drop halts in the current position, it cannot roll any further, under the gravitational force.

Subcase 2.2.2 : $\xi^1 = 0$, $\mu \not\geqq 0$: If $\xi^1 = 0$ and $\mu \not\geqq 0$, delete the i corresponding to the most negative μ_i from the set $\mathbf{I}_1$ (any other commonly used rule for deleting one or more of the i associated with negative μ_i from $\mathbf{I}_1$ can be applied in this case). Redefine the matrix E_1 to be the one whose rows are $A_i.$ for i in the new set $\mathbf{I}_1$, compute the new orthogonal projection ξ^1 as in (11.44) using the new E_1 and repeat Subcase 2.1 or 2.2 as appropriate with the new ξ^1.

General Step r : Let y^{r-1} be the direction determined in the previous step. Define

$$\mathbf{J}_r = \{i : i \in \mathbf{J}(\overline{x}) \text{ and } A_i.y^{r-1} < 0\}.$$

Case 1 : $\mathbf{J}_r = \emptyset$: If $\mathbf{J}_r = \emptyset$, y^{r-1} is the gravitational direction at $\overline{x}$, and the distance the drop can move in this direction is determined as in (11.42) with y^{r-1} replacing y^0.

Case 2 : $\mathbf{J}_r \neq \emptyset$: Define $\mathbf{T}_r = \bigcup_{s=1}^{r} \mathbf{J}_s$ and let D_r be the matrix of order $|\mathbf{T}_r| \times n$ whose rows are $A_i.$ for $i \in \mathbf{T}_r$. Let E_r be the submatrix of D_r of order (rank of D_r) $\times n$, whose set of rows is a maximal linearly independent subset of row vectors of D_r. Let $\mathbf{I}_r = \{i : A_i. \text{ is a row vector of } E_r\}$. Let $\mathbf{F}_r$ be the subspace $\{x : D_rx = 0\} = \{x : E_rx = 0\}$. Let ξ^r be the orthogonal projection of y^0 in the subspace $\mathbf{F}_r$, that is

$$\xi^r = (I - E_r^T(E_rE_r^T)^{-1}E_r)y^0.$$

Subcase 2.1 : $\xi^r \neq 0$: Let $y^r = \xi^r/\|\xi^r\|$, go to Step $r+1$.

Subcase 2.2 : $\xi^r = 0$: Let $\mu = (\mu_i : i \in \mathbf{I}_r) = -\|c\|((E_rE_r^T)^{-1}E_ry^0)^T$.

Subcase 2.2.1 : $\xi^r = 0$, and $\mu \geqq 0$: Define $\overline{\pi} = (\overline{\pi}_i)$ by

$$\begin{aligned}\overline{\pi}_i &= 0, \text{ for } i \notin \mathbf{I}_r\\ &= \mu_i, \text{ for } i \in \mathbf{I}_r.\end{aligned}$$

$\overline{\pi}$ is a basic feasible solution to the dual of (11.36). In this case the drop halts, it cannot roll any further under the gravitational force.

Subcase 2.2.2 : $\xi^r = 0$, and $\mu \not\geqq 0$: If $\xi^r = 0$ and $\mu \not\geqq 0$, proceed exactly as under Subcase 2.2.2 described under Step 1, with $\mathbf{I}_r$ replacing P_1.

It can be shown that this procedure does produce the gravitational direction at $\overline{x}$, finitely, if the drop can move at all. Currently work is being carried out on developing efficient methods for choosing the index set $\mathbf{I}_r$ of maximal linearly independent subset of row vectors of D_r, in Case 2, and on the best strategies for deleting a subset of constraints associated with negative μ_i in Subcase 2.2.2. Other principles for defining the gravitational direction at the interior point $\overline{x}$ of $\mathbf{K}$, are also being investigated.

Conditions for the Halting of the Drop

Let ε be the radius of the drop and $\overline{x} \in \mathbf{K}$ satisfy (11.39). We have the following theorem.

Theorem 11.6 *When the center of the drop is at $\overline{x}$, it halts iff $\mathbf{J}(\overline{x})$ defined in (11.40) is $\neq \emptyset$, and there exists a dual feasible solution $\overline{\pi} = (\overline{\pi}_i)$ for the dual of (11.36) satisfying*

$$\overline{\pi}_i = 0, \text{ for all } i \notin \mathbf{J}(\overline{x}). \tag{11.45}$$

Proof. The drop will halt when its center is at $\overline{x}$, iff there exists no direction at $\overline{x}$ along which the drop could move within the interior of $\mathbf{K}$, that will slide its center on a line of decreasing objective value for some positive length. That is, iff there exists no y satisfying

$$cy < 0$$
$$(A_i.(\overline{x} + \lambda y) - b_i)/\|A_i.\| \geqq \varepsilon, \ i = 1 \text{ to } m$$

for $0 \leqq \lambda < \alpha$, for some $\alpha > 0$. Since $\overline{x}$ satisfies (11.39), and from the definition of $\mathbf{J}(\overline{x})$ in (11.40), this implies that the drop will halt when its center is at $\overline{x}$ iff the system

$$A_i.y \geqq 0, \text{ for all } i \in \mathbf{J}(\overline{x})$$
$$cy < 0$$

has no solution y. By the well known Farkas' lemma, Theorem 3 in Appendix 1, this holds iff there exists a $\overline{\pi} = (\overline{\pi}_i : i = 1 \text{ to } m)$ feasible to the dual of (11.36) satisfying (11.45).

□

What to Do When the Drop Halts?

Theorem 11.7 *Suppose the drop of radius ε halts with its center at $\overline{x} \in \mathbf{K}$. Then the LP (11.36) has a finite optimum solution. Let z^* be the optimum objective value in (11.36). Let $\overline{\pi} = (\overline{\pi}_i)$ be the dual feasible solution satisfying (11.45) guaranteed to exist by Theorem 11.6. Then*

$$c\overline{x} = \overline{\pi}b + \varepsilon \sum_{i \in \mathbf{J}(\overline{x})} \overline{\pi}_i \tag{11.46}$$

and

$$c\overline{x} \leqq z^* + \varepsilon \sum_{i \in \mathbf{J}(\overline{x})} \overline{\pi}_i. \tag{11.47}$$

Proof. If the drop halts, by Theorem 11.6, the dual of (11.36) is feasible. So, the LP (11.36) has a finite optimum solution by the duality theory of LP. Consider the perturbed LP

$$\begin{array}{ll} \text{minimize} & z(x) = cx \\ \text{subject to} & A_{i.}x \geqq \begin{cases} b_i, & \text{for } i \notin \mathbf{J}(\overline{x}) \\ b_i + \varepsilon, & \text{for } i \in \mathbf{J}(\overline{x}). \end{cases} \end{array} \tag{11.48}$$

The hypothesis in the theorem implies that $\overline{x}$, $\overline{\pi}$, together satisfy the primal, dual feasibility and the complementary slackness optimality conditions for (11.48) and its dual. Hence, by the duality theorem of LP, (11.46) holds. Also, by the weak duality theorem of LP, (11.47) holds.

Hence, if the drop halts with its center at position $\overline{x}$, and a $\overline{\pi}$ satisfying (11.45) is found, and $\varepsilon \sum_{i \in \mathbf{J}(\overline{x})} \overline{\pi}_i$ is small, then $\overline{x}$ can be taken as a near optimum solution to (11.36) and the algorithm terminated. Also, in this case $\overline{\pi}$ is an optimum solution for the dual of (11.36), and the true optimum solution of (11.36) can be obtained by well known pivotal methods that move from $\overline{x}$ to an extreme point without increasing the objective value (see Subsection 5 in Section 11.4.1).

Theorem 11.8 *Suppose the drop of radius ε halts with its center at $\overline{x} \in \mathbf{K}$. If the system of equations*

$$A_{i.}x = b_i, \ i \in \mathbf{J}(\overline{x}) \tag{11.49}$$

has a solution $\tilde{x}$ which is feasible to (11.36), then $\tilde{x}$ is an optimum feasible solution of (11.36).

Proof. Let $\overline{\pi}$ be the dual feasible solution satisfying (11.45) guaranteed by Theorem 11.6. It can be verified that $\tilde{x}$, $\overline{\pi}$ together satisfy the complementary slackness optimality conditions for (11.36) and its dual, so $\tilde{x}$ is an optimum solution for (11.36). In this case $\overline{\pi}$ is optimum to the dual of (11.36).

□

If the drop of radius ε halts with its center at $\overline{x} \in \mathbf{K}$, and there exists no solution to the system of equations (11.49) which is feasible to (11.36), then this drop is unable to move any further down in $\mathbf{K}$ under the gravitational force, even though it is not close to an optimum solution for (11.36). See Figure 11.8.

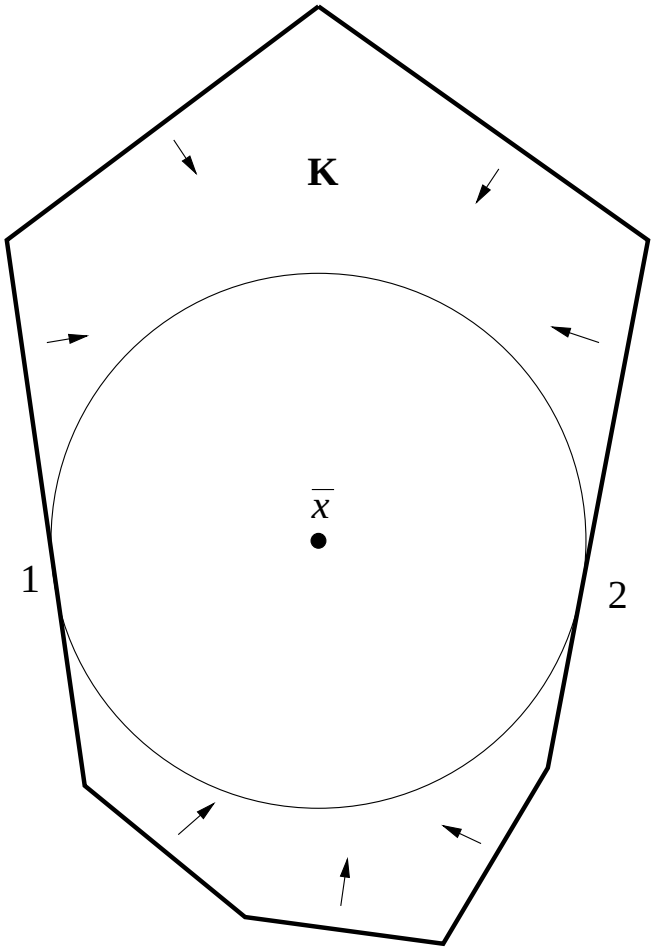

Figure 11.8 The set **K** is on the side of the arrow marked on each constraint. The gravitational force is pulling the drop straight down, but it cannot move any further, because it is squeezed between hyperplanes 1 and 2.

Suppose the drop of radius ε halts with its center at $\overline{x}$. If the system

$$A_i.x = b_i,\ i \in \mathbf{J}(x) \tag{11.50}$$

has no feasible solution, the gravitational method reduces the radius of the drop, see below, keeping the center at $\overline{x}$, and continues.

On the other hand, suppose the drop of radius ε halts with its center at $\overline{x}$, and the system (11.50) is feasible. Let E be the matrix whose rows form a maximal linearly independent subset of rows of $\{A_i. : i \in \mathbf{J}(\overline{x})\}$. Then the nearest point to $\overline{x}$ in the flat $\{x : A_i.x = b_i, i \in \mathbf{J}(\overline{x})\}$ is $\hat{x} = \overline{x} + E^T(EE^T)^{-1}(d - E\overline{x})$ where d is the column vector of b_i for i such that $A_i.$ is a row of E. If $\hat{x}$ is feasible to (11.36), then by Theorem 11.8, $\hat{x}$ is an optimum feasible solution for (11.36) and the method terminates. Otherwise, at this stage the gravitational method reduces the radius of the drop (for example, replace ε by $\varepsilon/2$), keeping the center at $\overline{x}$, and traces the locus of the center of the new

drop as it now begins to fall under the influence of gravity again. The same process is repeated when the new drop halts.

See Figure 11.9 for an illustration of the path of the drop in a convex polyhedron in $\mathbf{R}^3$.

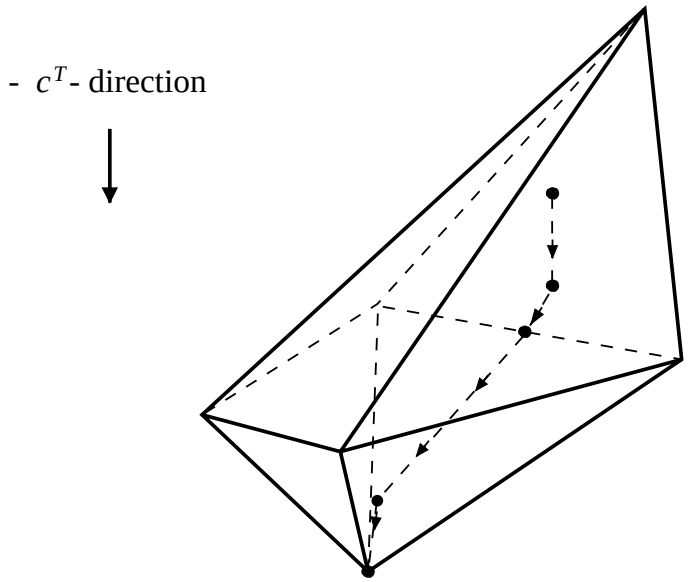

Figure 11.9 Path of the drop in the gravitational method in a convex polyhedron in $\mathbf{R}^3$.

The theoretical worst case computational complexity of this algorithm is currently under investigation. Initial computational trials with the method are very encouraging. The practical efficienty of this algorithm is also being studied via a computational project.

11.5 References

11.1 R. Chandrasekaran, S. N. Kabadi and K. G. Murty, "Some $\mathcal{NP}$-complete problems in linear programming", *Operations Research Letters*, 1, 3 (July 1982) 101–104.

11.2 N. Karmarkar, "A new polynomial-time algorithm for linear programming", *Combinatorica*, 4, no. 4 (1984) 373–395.

11.3 K. G. Murty, "Algorithm for finding all the feasible complementary bases for a linear complementarity problem", Technical Report 70-7, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109–2117.

11.4 K. G. Murty and Y. Fathi, "A feasible direction method for linear programming", *Operations Research Letters*, 3, no. 3 (August 1984) 121–127.

11.5 K. G. Murty, "A new interior variant of the gradient projection method for linear programming", Technical Report 85-18, Department of IOE, University of Michigan, Ann Arbor, MI 48109–2117, USA, 1985.

11.6 K. G. Murty, "The gravitational method for linear programming", *Opsearch*, 23, no. 4 (1986) 206–214.

11.7 E. Tardos, "A strongly polynomial minimum cost circulation algorithm", *Combinatorica*, 5 (1985) 247–255.