I.2

Linear Programming

1. INTRODUCTION

The general linear programming problem is

$$\text{(LP)} \qquad z_{\text{LP}} = \max\{cx\colon Ax \leqslant b,\ x \in R_+^n\},$$

where the data are rational and are given by the $m \times n$ matrix A, the $1 \times n$ matrix c, and the $m \times 1$ matrix b. This notation is different from that of Section I.1.1 but is preferable here because of its widespread use in linear programming. Recall that, as we observed in Section I.1.1, equality constraints can be represented by two inequality constraints.

Problem LP is well-defined in the sense that if it is feasible and does not have unbounded optimal value, then it has an optimal solution.

A good understanding of the theory and algorithms of linear programming is essential for understanding integer programming for several reasons that can be summed up by the statement that "one has to learn to walk before one can run". Integer programming is a much harder problem than linear programming, and neither the theory nor the computational aspects of integer programming are as developed as they are for linear programming. So, first of all, the theory of linear programming serves as a guide and motivating force for developing results for integer programming.

Computationally, linear programming algorithms are very often used as a subroutine in integer programming algorithms to obtain upper bounds on the value of the integer program. Let

$$\text{(IP)} \qquad z_{\text{IP}} = \max\{cx\colon Ax \leqslant b,\ x \in Z_+^n\}$$

and observe that $z_{\text{LP}} \geqslant z_{\text{IP}}$ since $Z_+^n \subset R_+^n$. The upper bound z_{LP} sometimes can be used to prove optimality for IP; that is, if x^0 is a feasible solution to IP and $cx^0 = z_{\text{LP}}$, then x^0 is an optimal solution to IP.

A deeper connection between linear and integer programming is that corresponding to any integer programming problem there is a linear programming problem $\max\{cx\colon Ax \leqslant b,\ A^1x \leqslant b^1,\ x \in R_+^n\}$ that has the same answer as IP.

Our presentation of linear programming is by necessity very terse and is not intended as a substitute for a full treatment. The reader who has already studied linear programming is advised to scan this section to become familiar with our notation or, perhaps, to review an unfamiliar topic.

In the next section, we consider the duality theory of linear programming which, among other things, provides necessary and sufficient optimality conditions. In the following two sections, we present algorithms for solving linear programs.

The simplex algorithms are used to prove the main duality theorem and also to show that every feasible instance of LP that is not unbounded has an optimal solution. But, more importantly, they are the practical algorithms that are part of linear programming software systems and many integer programming software systems as well. The performance of simplex algorithms, observed over years of practical experience, shows that they are very robust and efficient. Typically the number of iterations required is a small multiple of m. Although there exist simplex algorithms that converge finitely, these are inefficient; and the ones used in practice can fail to converge. Moreover, there are examples which show that finitely convergent simplex algorithms may require an exponential number of iterations. But this bad behavior does not seem to occur in the solution of practical problems.

Section 4 deals with subgradient optimization. There are convergent subgradient algorithms, but, as described, they are not finite. However, on certain classes of linear programs that arise in solving integer programs, they tend to produce good solutions very quickly.

In Chapter I.6, we consider two other linear programming algorithms. These have been deferred to a later chapter because some of the motivation for considering them concerns the theoretical complexity of computations, which is studied in Chapter I.5.

2. DUALITY

Duality deals with pairs of linear programs and the relationships between their solutions. One problem is called the primal and the other the dual.

We state the *primal* problem as

(P) $$z_{\mathrm{LP}} = \max\{cx\colon Ax \leqslant b,\ x \in R^n_+\}.$$

Its *dual* is defined as the linear program

(D) $$w_{\mathrm{LP}} = \min\{ub\colon uA \geqslant c,\ u \in R^m_+\}.$$

It does not matter which problem is called the primal because:

Proposition 2.1. *The dual of the dual is the primal.*

Proof. To take the dual of the dual, we need to restate it as a maximization problem with equal-to-or-less-than constraints. Once this is done, the result follows easily. We leave the details to the reader. ■

Feasible solutions to the dual provide upper bounds on z_{LP} and feasible solutions to the primal yield lower bounds on w_{LP}. In particular:

Proposition 2.2 *(Weak Duality). If x^* is primal feasible and u^* is dual feasible, then $cx^* \leqslant z_{\mathrm{LP}} \leqslant w_{\mathrm{LP}} \leqslant u^*b$.*

Proof. $cx^* \leqslant u^*Ax^* \leqslant u^*b$, where the first inequality uses $u^*A \geqslant c$ and $x^* \geqslant 0$, and the second uses $Ax^* \leqslant b$ and $u^* \geqslant 0$. Hence $w_{\mathrm{LP}} \geqslant cx$ for all feasible solutions x to P, and $z_{\mathrm{LP}} \leqslant ub$ for all feasible solutions u to D, so that $w_{\mathrm{LP}} \geqslant z_{\mathrm{LP}}$. ■

Corollary 2.3. *If problem* P *has unbounded optimal value, then* D *is infeasible.*

Proof. By weak duality, $w_{LP} \geqslant \lambda$ for all $\lambda \in R^1$. Hence D has no feasible solution. ■

We now come to the fundamental result of linear programming duality, which says that if both problems are feasible their optimal values are equal. A constructive proof will be given in the next section.

Theorem 2.4 *(Strong Duality). If z_{LP} or w_{LP} is finite, then both* P *and* D *have finite optimal value and $z_{LP} = w_{LP}$.*

Corollary 2.5. *There are only four possibilities for a dual pair of problems* P *and* D.

i. *z_{LP} and w_{LP} are finite and equal.*
ii. *$z_{LP} = \infty$ and* D *is infeasible.*
iii. *$w_{LP} = -\infty$ and* P *is infeasible.*
iv. *Both* P *and* D *are infeasible.*

A problem pair with property iv is $\max\{x_1 + x_2\colon x_1 - x_2 \leqslant -1, -x_1 + x_2 \leqslant -1, x \in R_+^2\}$ and its dual.

Another important property of primal–dual pairs is *complementary slackness*. Let $s = b - Ax \geqslant 0$ be the vector of slack variables of the primal and let $t = uA - c \geqslant 0$ be the vector of surplus variables of the dual.

Proposition 2.6. *If x^* is an optimal solution of* P *and u^* is an optimal solution of* D, *then $x_j^* t_j^* = 0$ for all j, and $u_i^* s_i^* = 0$ for all i.*

Proof. Using the definitions of s^* and t^*, we have

$$\begin{aligned} cx^* &= (u^*A - t^*)\,x^* = u^*Ax^* - t^*x^* \\ &= u^*(b - s^*) - t^*x^* = u^*b - u^*s^* - t^*x^*. \end{aligned}$$

By Theorem 2.4, $cx^* = u^*b$. Hence $u^*s^* + t^*x^* = 0$ with $u^*, s^*, t^*, x^* \geqslant 0$ so that the result follows. ■

Example 2.1. The dual of the linear program

$$\text{(P)} \qquad \begin{aligned} z_{LP} = \max\ & 7x_1 + 2x_2 \\ & -x_1 + 2x_2 \leqslant 4 \\ & 5x_1 + x_2 \leqslant 20 \\ & -2x_1 - 2x_2 \leqslant -7 \\ & x \in R_+^2 \end{aligned}$$

is

$$\text{(D)} \qquad \begin{aligned} w_{LP} = \min\ & 4u_1 + 20u_2 - 7u_3 \\ & -u_1 + 5u_2 - 2u_3 \geqslant 7 \\ & 2u_1 + u_2 - 2u_3 \geqslant 2 \\ & u \in R_+^3. \end{aligned}$$

It is easily checked that $x^* = (\frac{36}{11} \;\; \frac{40}{11})$ is feasible in P, and hence $z_{LP} \geqslant cx^* = 30\frac{2}{11}$. Similarly, $u^* = (\frac{3}{11} \;\; \frac{16}{11} \;\; 0)$ is feasible in D, and hence, by weak duality, $z_{LP} \leqslant u^*b = 30\frac{2}{11}$. The two points together yield a proof of optimality, namely, x^* is optimal for P and u^* is optimal for D.

Note also that the complementary slackness condition holds. The slack variables in P are $(s_1^*, s_2^*, s_3^*) = (0 \;\; 0 \;\; 6\frac{9}{11})$, and the surplus variables in D are $(t_1^*, t_2^*) = (0 \;\; 0)$. Hence $x_j^* t_j^* = 0$ for $j = 1, 2$ and $u_i^* s_i^* = 0$ for $i = 1, 2, 3$. ■

It is important to be able to verify whether a system of linear inequalities is feasible or not. Duality provides a very useful characterization of infeasibility.

Theorem 2.7 *(Farkas' Lemma). Either* $\{x \in R_+^n: Ax \leqslant b\} \neq \emptyset$ *or (exclusively) there exists* $v \in R_+^m$ *such that* $vA \geqslant 0$ *and* $vb < 0$.

Proof. Consider the linear program $z_{LP} = \max\{0x: Ax \leqslant b, x \in R_+^n\}$ and its dual $w_{LP} = \min\{vb: vA \geqslant 0, v \in R_+^m\}$. As $v = 0$ is a feasible solution to the dual problem, only possibilities i and iii of Corollary 2.5 can occur.

i. $z_{LP} = w_{LP} = 0$. Hence $\{x \in R_+^n: Ax \leqslant b\} \neq \emptyset$ and $vb \geqslant 0$ for all $v \in R_+^m$ with $vA \geqslant 0$;

iii. $z_{LP} = w_{LP} = -\infty$. Hence $\{x \in R_+^n: Ax \leqslant b\} = \emptyset$ and there exists $v \in R_+^m$ with $vA \geqslant 0$ and $vb < 0$. ■

There are many other versions of Farkas' Lemma. Some are presented in the following proposition.

Proposition 2.8. *(Variants of Farkas' Lemma)*

a. Either $\{x \in R_+^n: Ax = b\} \neq \emptyset$, or $\{v \in R^m: vA \geqslant 0, vb < 0\} \neq \emptyset$.
b. Either $\{x \in R^n: Ax \leqslant b\} \neq \emptyset$, or $\{v \in R_+^m: vA = 0, vb < 0\} \neq \emptyset$.
c. If $P = \{r \in R_+^n: Ar = 0\}$, either $P \setminus \{0\} \neq \emptyset$, or $\{u \in R^m: uA > 0\} \neq \emptyset$.

3. THE PRIMAL AND DUAL SIMPLEX ALGORITHMS

Here it is convenient to consider the primal linear program with equality constraints:

(LP) $$z_{LP} = \max\{cx: Ax = b, x \in R_+^n\}.$$

Its dual is

(DLP) $$w_{LP} = \min\{ub: uA \geqslant c, u \in R^m\}.$$

We suppose that $\text{rank}(A) = m \leqslant n$, so that all redundant equations have been removed from LP.

Bases and Basic Solutions

Let $A = (a_1, a_2, \ldots, a_n)$ where a_j is the jth column of A. Since $\text{rank}(A) = m$, there exists an $m \times m$ nonsingular submatrix $A_B = (a_{B_1}, \ldots, a_{B_m})$. Let $B = \{B_1, \ldots, B_m\}$ and let $N =$

$\{1, \ldots, n\} \setminus B$. Now permute the columns of A so that $A = (A_B, A_N)$. We can write $Ax = b$ as $A_Bx_B + A_Nx_N = b$, where $x = (x_B, x_N)$. Then a solution to $Ax = b$ is given by $x_B = A_B^{-1}b$ and $x_N = 0$.

Definition 3.1

a. The $m \times m$ nonsingular matrix A_B is called a *basis*.
b. The solution $x_B = A_B^{-1}b$, $x_N = 0$ is called a *basic solution* of $Ax = b$.
c. x_B is the vector of *basic variables* and x_N is the vector of *nonbasic variables*.
d. If $A_B^{-1}b \geqslant 0$, then (x_B, x_N) is called a *basic primal feasible solution* and A_B is called a *primal feasible basis*.

Now let $c = (c_B, c_N)$ be the corresponding partition of c, that is, $cx = c_Bx_B + c_Nx_N$, and let $u = c_BA_B^{-1} \in R^m$. This solution is complementary to $x = (x_B, x_N)$, since

$$uA - c = c_BA_B^{-1}(A_B, A_N) - (c_B, c_N) = (0, c_BA_B^{-1}A_N - c_N)$$

and $x_N = 0$. Observe that u is a feasible solution to the dual if and only if $c_BA_B^{-1}A_N - c_N \geqslant 0$. This motivates the next definition.

Definition 3.2. If $c_BA_B^{-1}A_N \geqslant c_N$, then A_B is called a *dual feasible basis*.

Note that a basis A_B defines the point $x = (x_B, x_N) = (A_B^{-1}b, 0) \in R^n$ and the point $u = c_BA_B^{-1} \in R^m$. A_B may be only primal feasible, only dual feasible, neither, or both. Bases that are both primal and dual feasible are of particular importance.

Proposition 3.1. *If A_B is primal and dual feasible, then $x = (x_B, x_N) = (A_B^{-1}b, 0)$ is an optimal solution to* LP *and $u = c_BA_B^{-1}$ is an optimal solution to* DLP.

Proof. $x = (A_B^{-1}b, 0)$ is feasible to LP with value $cx = c_BA_B^{-1}b$. $u = c_BA_B^{-1}$ is feasible in DLP and $ub = c_BA_B^{-1}b$. Hence the result follows from weak duality. ■

Changing the Basis

We say that *two bases A_B and $A_{B'}$ are adjacent* if they differ in only one column, that is $|B \setminus B'| = |B' \setminus B| = 1$. If A_B and $A_{B'}$ are adjacent, the basic solutions they define are also said to be adjacent. The simplex algorithms to be presented in this section work by moving from one basis to another adjacent one.

Given the basis A_B, it is useful to rewrite LP in the form

$$\text{LP}(B) \qquad \begin{aligned} z_{\text{LP}} = c_BA_B^{-1}b + \max(c_N - c_BA_B^{-1}A_N)x_N \\ x_B + A_B^{-1}A_Nx_N = A_B^{-1}b \\ x_B, x_N \geqslant 0. \end{aligned}$$

It is simple to show that problems LP(B) and LP have the same set of feasible solutions and objective values.

We now define some additional notation that allows us to state things more concisely. Let $\bar{A}_N = A_B^{-1}A_N$, $\bar{b} = A_B^{-1}b$, and $\bar{c}_N = c_N - c_BA_B^{-1}A_N$ so that

$$\text{LP}(B) \qquad \begin{aligned} z_{\text{LP}} &= c_B\bar{b} + \max \bar{c}_N x_N \\ x_B &+ \bar{A}_N x_N = \bar{b} \\ x_B&, x_N \geqslant 0. \end{aligned}$$

Also, for $j \in N$, we let $\bar{a}_j = A_B^{-1}a_j$ and $\bar{c}_j = c_j - c_B\bar{a}_j$ so that

$$\text{LP}(B) \qquad \begin{aligned} z_{\text{LP}} &= c_B\bar{b} + \max \sum_{j\in N} \bar{c}_j x_j \\ x_B &+ \sum_{j\in N} \bar{a}_j x_j = \bar{b} \\ x_B \geqslant 0, &\quad x_j \geqslant 0 \quad \text{for } j \in N. \end{aligned}$$

Finally, we sometimes write the equations of LP(B) as

$$x_{B_i} + \sum_{j\in N} \bar{a}_{ij}x_j = \bar{b}_i \quad \text{for } i = 1, \ldots, m,$$

that is, $\bar{a}_j = (\bar{a}_{1j}, \ldots, \bar{a}_{mj})$ and $\bar{b} = (\bar{b}_1, \ldots, \bar{b}_m)$.

Let $\bar{c}_N = c_N - c_B\bar{A}_N$ be the *reduced price* vector for the nonbasic variables. Then, by Definition 3.2, dual feasibility of basis A_B is equivalent to $\bar{c}_N \leqslant 0$.

Now given the representation LP(B), we show how to move from one basic primal feasible solution to another in a systematic way.

Definition 3.3. A primal basic feasible solution $x_B = \bar{b}$, $x_N = 0$ is *degenerate* if $\bar{b}_i = 0$ for some i.

Proposition 3.2. *Suppose all primal basic feasible solutions are nondegenerate. If A_B is a primal feasible basis and a_r is any column of A_N, then matrix (A_B, a_r) contains, at most, one primal feasible basis other than A_B.*

Proof. We consider the system

$$\text{(3.1)} \qquad \begin{aligned} x_B + \bar{a}_r x_r &= \bar{b} \\ x_B \geqslant 0, \quad x_r &\geqslant 0, \end{aligned}$$

that is, all components of x_N except x_r equal zero.

Case 1. $\bar{a}_r \leqslant 0$. Suppose $x_r = \lambda > 0$. Then for all $\lambda > 0$ we obtain

$$x_B = \bar{b} - \bar{a}_r\lambda \geqslant \bar{b} > 0.$$

Thus for every feasible solution to (3.1) with $x_r > 0$, we have $x_B > 0$ so that A_B is the only primal feasible basis contained in (A_B, a_r).

Case 2. At least one component of $\bar{a}_r$ is positive. Let

$$\text{(3.2)} \qquad \lambda_r = \min\left\{\frac{\bar{b}_i}{\bar{a}_{ir}} : \bar{a}_{ir} > 0\right\} = \frac{\bar{b}_s}{\bar{a}_{sr}}.$$

Hence $\bar{b} - \bar{a}_r\lambda_r \geqslant 0$ and $\bar{b}_s - \bar{a}_{sr}\lambda_r = 0$. So we obtain an adjacent primal feasible basis $A_{B^{(r)}}$ by deleting B_s from B and replacing it with r, that is, $B^{(r)} = B \cup \{r\} \setminus \{B_s\}$. Note that the nondegeneracy assumption implies that $\bar{b}_i - \bar{a}_{ir}\lambda_r > 0$ for $i \neq s$ so that the minimum in (3.2) is unique. Consequently, any basis $A_{\hat{B}}$ with $\hat{B} = B \cup \{r\} \setminus \{k\}$ for $k \in B \setminus \{B_s\}$ is not primal feasible. ∎

The new solution is calculated by:

1. Dividing

$$x_{B_S} + \bar{a}_{sr}x_r + \sum_{j \in N \setminus \{r\}} \bar{a}_{sj}x_j = \bar{b}_s$$

by $\bar{a}_{sr}$, which yields

$$\frac{1}{\bar{a}_{sr}}x_{B_S} + x_r + \sum_{j \in N \setminus \{r\}} \left(\frac{\bar{a}_{sj}}{\bar{a}_{sr}}\right)x_j = \frac{\bar{b}_s}{\bar{a}_{sr}}. \tag{3.3}$$

2. Eliminating x_r from the remaining equations by adding $-\bar{a}_{ir}$ multiplied by (3.3) to

$$x_{B_i} + \bar{a}_{ir}x_r + \sum_{j \in N \setminus \{r\}} \bar{a}_{ij}x_j = \bar{b}_i \quad \text{for } i \neq s$$

and eliminating x_r from the objective function.

This transformation is called a *pivot*. It corresponds precisely to a step in the well-known Gaussian elimination technique for solving linear equations. The coefficient $\bar{a}_{sr}$ is called the *pivot element*.

Corollary 3.3. *Suppose A_B is a primal feasible nondegenerate basis that is not dual feasible and $\bar{c}_r > 0$.*

a. *If $\bar{a}_r \leqslant 0$, then $z_{\text{LP}} = \infty$.*
b. *If at least one component of $\bar{a}_r$ is positive, then $A_{B^{(r)}}$, the unique primal feasible basis adjacent to A_B that contains a_r, is such that $c_{B^{(r)}}x_{B^{(r)}} > c_Bx_B$.*

Proof

a. $x_B = \bar{b} - \bar{a}_r\lambda$, $x_r = \lambda$, $x_j = 0$ otherwise is feasible for all $\lambda > 0$ and

$$c_Bx_B + c_rx_r = c_B\bar{b} + \bar{c}_r\lambda \to \infty \quad \text{as} \quad \lambda \to \infty.$$

b.

$$c_{B^{(r)}}x_{B^{(r)}} = c_B\bar{b} + \bar{c}_r\lambda_r > c_B\bar{b} = c_Bx_B,$$

where the inequality holds since λ_r defined by (3.2) is positive and $\bar{c}_r > 0$ by hypothesis. ∎

Primal Simplex Algorithm

We are now ready to describe the main routine of the primal simplex method called *Phase 2*. It begins with a primal feasible basis and then checks for dual feasibility. If the basis is not dual feasible, either an adjacent primal feasible basis is found with (in the absence of degeneracy) a higher objective value or $z_{LP} = \infty$ is established.

Phase 2

Step 1 (Initialization): Start with a primal feasible basis A_B.

Step 2 (Optimality Test): If A_B is dual feasible (i.e., $\bar{c}_N < 0$), stop. $x_B = \bar{b}$, $x_N = 0$ is an optimal solution. Otherwise go to Step 3.

Step 3 (Pricing Routine): Choose an $r \in N$ with $\bar{c}_r > 0$.

a. *Unboundedness test.* If $\bar{a}_r \leqslant 0$, $z_{LP} = \infty$.
b. *Basis change.* Otherwise, find the unique adjacent primal feasible basis $A_{B^{(r)}}$ that contains a_r. Let $B \leftarrow B^{(r)}$ and return to Step 2.

Note that in Step 3, we can choose any $j \in N$ with $\bar{c}_j > 0$. A pricing rule commonly used is to choose $r = \arg(\max_{j \in N} \bar{c}_j)$, since it gives the largest increase in the objective function per unit increase of the variable that becomes basic. But this computation can be time consuming when n is large, so that various modifications of it are used in practice.

Theorem 3.4. *Under the assumption that all basic feasible solutions are nondegenerate, Phase 2 terminates in a finite number of steps either with an unbounded solution or with a basis that is primal and dual feasible.*

Proof. At each step the value of the basic feasible solution increases. Thus no basis can be repeated. Because there is only a finite number of bases, this procedure must terminate finitely. ■

When basic solutions are degenerate, and this happens often in practice, Proposition 3.2 and Corollary 3.3 are not true. Consequently, the finiteness argument given in the proof of Theorem 3.4 does not apply.

Note that when the basic feasible solution is degenerate, the arg(min) of (3.2) may not be unique. In this case, (A_B, a_r) contains more than one primal feasible basis adjacent to A_B, and in Step 3b of the algorithm an arbitrary choice is made. A complication arises when $\lambda_r = 0$ in (3.2) since each primal feasible basis in (A_B, a_r) defines the same solution, namely, $x_B = \bar{b}$ and $x_N = 0$. A sequence of such degenerate changes of basis can, although it rarely happens in practice, lead back to the original basis. This phenomenon is called *cycling*.

Two methods for eliminating the possibility of cycling are known. One involves a lexicographic rule for breaking ties in (3.2), and the other involves both the choice of r in Step 3 and a tie-breaking rule for (3.2). By eliminating cycling, these algorithms establish the finiteness of Phase 2 for any linear programming problem. Hence there are primal simplex methods for which Theorem 3.4 holds without a nondegeneracy assumption.

Example 3.1

$$
\begin{aligned}
z_{LP} = \max \quad & 7x_1 + 2x_2 \\
& -x_1 + 2x_2 + x_3 \qquad = 4 \\
& 5x_1 + x_2 \qquad + x_4 = 20
\end{aligned}
$$

$$-2x_1 - 2x_2 \qquad\qquad + x_5 = -7$$

$$x \geqslant 0.$$

Step 1 (Initialization): The basis $A_B = (a_3, a_4, a_1)$ with

$$A_B^{-1} = \begin{pmatrix} 1 & 0 & -\frac{1}{2} \\ 0 & 1 & \frac{5}{2} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

yields the primal feasible solution

$$x_B = (x_3, x_4, x_1) = A_B^{-1}b = \bar{b} = \begin{pmatrix} 7\frac{1}{2} & 2\frac{1}{2} & 3\frac{1}{2} \end{pmatrix}$$

and $x_N = (x_2, x_5) = (0 \quad 0)$,

Iteration 1

Step 2:

$$\bar{A}_N = (\bar{a}_2, \bar{a}_5) = A_B^{-1}A_N = \begin{pmatrix} 3 & -\frac{1}{2} \\ -4 & \frac{5}{2} \\ 1 & -\frac{1}{2} \end{pmatrix},$$

$$\bar{c}_N = c_N - c_B\bar{A}_N = (2 \quad 0) - (0 \quad 0 \quad 7)\bar{A}_N = \begin{pmatrix} -5 & \frac{7}{2} \end{pmatrix}.$$

Thus LP(B) can be stated as

$$z_{LP} = 24\frac{1}{2} + \max - 5x_2 + \frac{7}{2}x_5$$

$$3x_2 - \frac{1}{2}x_5 + x_3 \qquad\qquad = 7\frac{1}{2}$$

$$-4x_2 + 2\frac{1}{2}x_5 \qquad + x_4 \qquad = 2\frac{1}{2}$$

$$x_2 - \frac{1}{2}x_5 \qquad\qquad + x_1 = 3\frac{1}{2}$$

$$x \geqslant 0.$$

Step 3: The only choice for a new basic variable is x_5. By (3.2),

$$\lambda_5 = \min\left\{ -, \frac{2\frac{1}{2}}{2\frac{1}{2}}, - \right\} = 1.$$

Hence x_4 is the leaving variable.

$$A_B \leftarrow A_{B^{(r)}} = (a_3, a_5, a_1).$$

Iteration 2

Step 2: $A_B^{-1} = \begin{pmatrix} 1 & \frac{1}{5} & 0 \\ 0 & \frac{2}{5} & 1 \\ 0 & \frac{1}{5} & 0 \end{pmatrix}$, $\quad x_B = (x_3, x_5, x_1) = \bar{b} = (8 \quad 1 \quad 4),$

$$\bar{c}_N = (\bar{c}_2, \bar{c}_4) = \begin{pmatrix} \frac{3}{5} & -\frac{7}{5} \end{pmatrix}.$$

x_2 is the entering variable.

Step 3: $\bar{a}_2 = (\frac{11}{5} \quad -\frac{8}{5} \quad \frac{1}{5})$. By (3.2), $\lambda_2 = \min(\frac{8}{11/5}, -, \frac{4}{1/5}) = \frac{40}{11}$. Hence x_3 is the leaving variable. $A_B \leftarrow (a_2, a_5, a_1)$.

Iteration 3

Step 2: $A_B^{-1} = \begin{pmatrix} \frac{5}{11} & \frac{1}{11} & 0 \\ \frac{8}{11} & \frac{6}{11} & 1 \\ -\frac{1}{11} & \frac{2}{11} & 0 \end{pmatrix}$, $\quad x_B = (x_2, x_5, x_1) = \begin{pmatrix} \frac{40}{11} & \frac{75}{11} & \frac{36}{11} \end{pmatrix},$

$$\bar{c}_N = (\bar{c}_3, \bar{c}_4) = \begin{pmatrix} -\frac{3}{11} & -\frac{16}{11} \end{pmatrix} \leqslant 0.$$

Hence $x = (x_1, x_2, x_3, x_4, x_5) = (\frac{36}{11} \quad \frac{40}{11} \quad 0 \quad 0 \quad \frac{75}{11})$ is an optimal solution to LP, and $u = c_B A_B^{-1} = (\frac{3}{11} \quad \frac{16}{11} \quad 0)$ is an optimal solution to DLP.

We have shown that if LP has a basic primal feasible solution, it either has unbounded optimal value or it has an optimal basic solution. It remains to show that if it has a feasible solution, then it has a basic feasible solution. This is accomplished by Phase 1 of the simplex algorithm.

Phase 1. By changing signs in each row if necessary, write LP as $\max\{cx: Ax = b, x \in R^n_+\}$ with $b \geqslant 0$. Now introduce *artificial variables* x_i^a for $i = 1, \ldots, m$, and consider the linear program

$$(\text{LP}^a) \qquad z_a = \max\left\{-\sum_{i=1}^{m} x_i^a : Ax + Ix^a = b, (x, x^a) \in R_+^{n+m}\right\}.$$

1. *LP*^{a} is a feasible linear program for which a basic feasible solution $x^a = b$, $x = 0$ is available. Hence LP^a can be solved by the Phase 2 simplex method. Moreover $z_a \leqslant 0$ so that LP^a has an optimal solution.
2. i) A feasible solution (x, x^a) to LP^a yields a feasible solution x to LP if and only if $x^a = 0$. Thus if $z_a < 0$, LP^a has no feasible solution with $x^a = 0$ and hence LP is infeasible.
   ii) If $z_a = 0$, then any optimal solution to LP^a has $x^a = 0$ and hence yields a feasible solution to LP. In particular, if all the artificial variables are nonbasic in some basic optimal solution to LP^a, a basic feasible solution for LP has been found.

On the other hand, if one or more artificial variables are basic, it may be possible to remove them from the basis by degenerate basis changes. When this is not possible it can be shown that certain constraints in the original problem are redundant, and the equations with basic artificial variables can be dropped. Again this leads to a basic feasible solution to LP.

By combining Phases 1 and 2, we obtain a finite algorithm for solving any linear program. This establishes Theorem 2.4 and also Theorem 3.5:

Theorem 3.5

a. *If* LP *is feasible*, it *has a basic primal feasible solution*.
b. *If* LP *has a finite optimal value*, it *has an optimal basic feasible solution*.

Example 3.1 (continued). We will use Phase 1 to construct the initial basis (a_3, a_4, a_1) that we used previously. The Phase 1 problem is

$$\begin{aligned} z_a = \max \quad & \qquad\qquad\qquad\qquad\qquad - x_1^a - x_2^a - x_3^a \\ & -x_1 + 2x_2 + x_3 \qquad\qquad + x_1^a \qquad\qquad = 4 \\ & 5x_1 + x_2 \qquad + x_4 \qquad\qquad + x_2^a \qquad = 20 \\ & 2x_1 + 2x_2 \qquad\qquad - x_5 \qquad\qquad + x_3^a = 7 \\ & x, x^a \geqslant 0. \end{aligned}$$

Observe, however, that because x_3, x_4 are slack variables and b_1 and b_2 are nonnegative, the artificial variables x_1^a and x_2^a are unnecessary. Hence we can start with (x_3, x_4, x_3^a) as basic variables. Since $-x_3^a = -7 + 2x_1 + 2x_2 - x_5$, the Phase 1 problem is

$$\begin{aligned} z_a = \max - 7 + 2x_1 + 2x_2 \qquad\qquad & - x_5 \\ -x_1 + 2x_2 + x_3 \qquad\qquad & \qquad = 4 \\ 5x_1 + x_2 \qquad + x_4 \qquad & \qquad = 20 \\ 2x_1 + 2x_2 \qquad\qquad & - x_5 + x_3^a = 7 \\ x \geqslant 0, x_3^a \geqslant 0. & \end{aligned}$$

Using the simplex algorithm (Phase 2) we introduce x_1 into the basis, and x_3^a leaves. The resulting basis (a_3, a_4, a_1) is a feasible basis for the original problem.

Dual Simplex Algorithm

The primal simplex algorithm works by moving from one primal feasible basis to another. In contrast, the dual simplex algorithm works by moving from one dual feasible basis to another. This latter approach is useful when we know a basic dual feasible solution but not a primal one. This occurs, for example, when we have an optimal solution to a linear programming problem that becomes infeasible because additional constraints have been added.

Proposition 3.6. *Let A_B be a dual feasible basis with $\bar{b}_s < 0$.*

a. *If $\bar{a}_{sj} \geqslant 0$ for all $j \in N$, then* LP *is infeasible.*
b. *Otherwise there is an adjacent dual feasible basis $A_{B^{(r)}}$, where $B^{(r)} = B \cup \{r\} \setminus \{B_s\}$ and $r \in N$ satisfies $\bar{a}_{sr} < 0$ and*

$$r = \arg\min_{j \in N}\left\{\frac{\bar{c}_j}{\bar{a}_{sj}} : \bar{a}_{sj} < 0\right\}.$$

Proof.

a. $x_{B_s} + \Sigma_{j\in N}\,\bar{a}_{sj}x_j = \bar{b}_s < 0$. Hence if $\bar{a}_{sj} \geqslant 0$ for all $j \in N$, every solution to $Ax = b$ with $x_j \geqslant 0$ for all $j \in N$ has $x_{B_s} < 0$.

b. If x_r enters the basis and x_{B_s} leaves we have

$$z = c_B\bar{b} + \sum_{j\in N}\bar{c}_j x_j - \lambda(x_{B_s} + \sum_{j\in N}\bar{a}_{sj}x_j) + \lambda\bar{b}_s$$

$$= c_B\bar{b} + \lambda\bar{b}_s + \sum_{j\in N}(\bar{c}_j - \lambda\bar{a}_{sj})x_j - \lambda x_{B_s},$$

where $\lambda = \dfrac{\bar{c}_r}{\bar{a}_{sr}} \geqslant 0$. The basis $A_{B^{(r)}}$ is dual feasible since $\lambda \geqslant 0$, $\bar{c}_j - \lambda\,\bar{a}_{sj} \leqslant \bar{c}_j$ for all j with $\bar{a}_{sj} \geqslant 0$, and $\bar{c}_j - \lambda\,\bar{a}_{sj} \leqslant 0$ for all j with $\bar{a}_{sj} < 0$ by the choice of r. ■

Dual Simplex Algorithm (Phase 2)

Step 1 (Initialization): A dual feasible basis A_B.

Step 2 (Optimality Test): If A_B is primal feasible, that is, $\bar{b} = A_B^{-1}\,b \geqslant 0$, then $x_B = \bar{b}$ and $x_N = 0$ is an optimal solution. Otherwise go to Step 3.

Step 3 (Pricing Routine): Choose an s with $\bar{b}_s < 0$.

a. *Feasibility Test.* If $\bar{a}_{sj} \geqslant 0$ for all $j \in N$, LP is infeasible.

b. *Basis change.* Otherwise let

$$r = \arg\min_{j\in N}\left\{\frac{\bar{c}_j}{\bar{a}_{sj}} : \bar{a}_{sj} < 0\right\}$$

and $B^{(r)} = B \cup (r) \setminus (B_s)$. Return to Step 2 with $B \leftarrow B^{(r)}$.

In contrast to the primal algorithm, in the dual simplex algorithm the objective function is nonincreasing. The magnitude of the decrease at each step is $|\bar{c}_r\bar{b}_s/\bar{a}_{rs}|$. In the absence of dual degeneracy, $\bar{c}_r < 0$ and the decrease is strict. As with the primal algorithm, it is possible to give more specific rules that guarantee finiteness. Such an algorithm is presented in Section II.4.3. A Phase 1 may be required to find a starting dual feasible basic solution.

Example 3.1 (continued). We apply the dual simplex algorithm.

Step 1 (Initialization): Consider the basis $A_B = (a_3, a_2, a_5)$, which is dual feasible since $\bar{c}_N = (\bar{c}_1, \bar{c}_4) = (-3 \quad -2)$.

Iteration 1

Step 2: The basis is not primal feasible since $x_B = (x_3, x_2, x_5) = (-36 \quad 20 \quad 33)$.

Step 3: The only possible choice is $s = 1$. We have $\bar{a}_{11} = -11$, $\bar{a}_{14} = -2$, and $\min(\frac{3}{11}, \frac{2}{2}) = \frac{3}{11}$. Hence $x_{B_1} = x_3$ leaves the basis, x_1 enters the basis, and $A_B \leftarrow (a_1, a_2, a_5)$.

Iteration 2. We have seen earlier that A_B is primal and dual feasible and hence optimal.

The Simplex Algorithm with Simple Upper Bounds

It is desirable for computational purposes to distinguish between upper-bound constraints of the form $x_j \leqslant h_j$ and other more general constraints. Hence we consider the problem

(ULP) $\qquad z_{\text{LP}} = \max\{cx\colon Ax = b, 0 \leqslant x_j \leqslant h_j \text{ for } j \in \{1, \ldots, n\}\}.$

Whereas the primal simplex algorithm described earlier would treat ULP as a problem with $m + n$ constraints, the *simplex algorithm with upper bounds* treats it as a problem with m constraints.

Now the columns of A are permuted so that $A = (A_B, A_{N_1}, A_{N_2})$, where A_B is a basis matrix as before, but the index set of the nonbasic variables N is partitioned into two sets N_1 and N_2. N_1 is the index set of variables at their lower bound ($x_j = 0$), and N_2 is the index set of variables at their upper bound ($x_j = h_j$).

Now we need to modify Definition 3.1.

Definition 3.4

a. The $m \times m$ nonsingular matrix A_B is called a *basis*.
b. For each partition N_1, N_2 of N, we associate the *basic solution* $x_B = A_B^{-1}(b - A_{N_2}h_{N_2}) = \bar{b} - \bar{A}_{N_2}h_{N_2}$, $x_{N_1} = 0$, $x_{N_2} = h_{N_2}$.
c. If $0 \leqslant \bar{b} - \bar{A}_{N_2}h_{N_2} \leqslant h_B$, then (x_B, x_{N_1}, x_{N_2}) is a *basic primal feasible solution*, and (B, N_1, N_2) indexes a primal feasible basis.

Now consider the dual of ULP,

$$\begin{aligned} \min\ & ub + vh \\ & uA + v \geqslant c \\ & v \geqslant 0, \end{aligned}$$

and let $v = (v_B, v_{N_1}, v_{N_2})$ and $c = (c_B, c_{N_1}, c_{N_2})$. The dual basic solution complementary to (x_B, x_{N_1}, x_{N_2}) is $(u, v_B, v_{N_1}, v_{N_2}) = (c_BA_B^{-1}, 0, 0, c_{N_2} - c_B\bar{A}_{N_2})$. Observe that (u, v) is a feasible solution to the dual if and only if $\bar{c}_{N_1} = c_{N_1} - c_B\bar{A}_{N_1} \leqslant 0$ and $\bar{c}_{N_2} = c_{N_2} - c_B\bar{A}_{N_2} \geqslant 0$.

Proposition 3.7. *If (A_B, A_{N_1}, A_{N_2}) is primal and dual feasible, then $x = (x_B, x_{N_1}, x_{N_2}) = (\bar{b} - \bar{A}_{N_2}h_{N_2}, 0, h_{N_2})$ is an optimal solution to ULP and $(u, v_{B_1}, v_{N_1}, v_{N_2}) = (c_BA_B^{-1}, 0, 0, c_{N_2} - c_B\bar{A}_{N_2})$ is an optimal solution to its dual.*

The modifications to the simplex algorithms are straightforward. Bases A_B and $A_{B'}$, are adjacent if (i) $|B \setminus B'| = |B' \setminus B| = 1$ or (ii) $B = B'$, and in both cases $|N_1' \setminus N_1| + |N_2' \setminus N_2| = 1$. In the latter case, one nonbasic variable changes from its lower to its upper bound, or vice versa. It is then easy to write out the rules for the choice of entering and leaving variable, leading to primal and dual simplex algorithms for ULP. Note that these algorithms choose the same pivots as the standard simplex algorithms, so the advantage lies in handling a basis that is $m \times m$ rather than $(m + n) \times (m + n)$.

Addition of Constraints or Variables

After solving LP to optimality, it is common that one or more new constraints or columns have to be added. In Part II, we will discuss cutting-place algorithms that add a constraint cutting off the optimal solution of LP; we will also discuss problems having such a large number of variables that we do not wish to introduce them all a priori.

If LP has been solved by a simplex algorithm, there is a straightforward way to use the current optimal basis A_B to solve the new problem. Suppose an inequality $\Sigma_{j=1}^{n} d_j x_j \leqslant d_0$ is added that is violated by the optimal solution $(x_B, x_N) = (A_B^{-1}b, 0)$. Now if x_{n+1} is the slack variable of the new constraint, then $B' = B \cup \{n+1\}$ indexes a new basis, and we obtain LP(B'):

$$\begin{aligned} z'_{\text{LP}} = \max\ & c_B\bar{b} + \bar{c}_N x_N \\ & x_B + \bar{A}_N x_N = \bar{b} \\ & x_{n+1} + (d_N - d_B\bar{A}_N)x_N = d_0 - d_B\bar{b} \\ & x_B, x_N, x_{n+1} \geqslant 0 \end{aligned}$$

since

$$dx + x_{n+1} = d_B x_B + d_N x_N + x_{n+1} = d_B\bar{b} + (d_N - d_B\bar{A}_N)x_N + x_{n+1}.$$

We see immediately that this basis is dual feasible and that it is primal feasible in all but the last row, that is, $d_B\bar{b} > d_0$. It is therefore desirable to reoptimize using the dual simplex algorithm. Since the current solution is "nearly" primal feasible, it is likely that only a few iterations will be required.

The procedure to be followed in adding new columns is dual to that described above. Given a new variable x_{n+1} with column $\binom{c_{n+1}}{a_{n+1}}$, we calculate its reduced price $\bar{c}_{n+1} = c_{n+1} - c_B A_B^{-1} a_{n+1}$ to check if the basis A_B remains optimal. If $\bar{c}_{n+1} \leqslant 0$, A_B is still optimal and the solution is unchanged. If $\bar{c}_{n+1} > 0$, we can use the primal simplex algorithm as A_B remains primal feasible.

Example 3.1 (continued). We add the upper-bound constraint $x_1 \leqslant 3$, cutting off the optimal solution $x = (\frac{36}{11}\ \ \frac{40}{11}\ \ 0\ \ 0\ \ \frac{75}{11})$. Let $x_1 + x_6 = 3$, so that x_6 is the new basic variable. Starting from the optimal basis $A_B = (a_2, a_5, a_1)$, we have $d_B = (0\ \ 0\ \ 1)$, $d_N = (0\ \ 0)$, $d_0 = 3$, and $A_{B'} = (a_2, a_5, a_1, a_6)$.

Iteration 1

Step 2: $x_{B'} = (\frac{40}{11}\ \ \frac{75}{11}\ \ \frac{36}{11}\ \ -\frac{3}{11})$.
Step 3: x_6 leaves the basis

$$\bar{c}_N = \left(-\frac{3}{11}\ \ -\frac{16}{11}\right),\quad d_N - d_B\bar{A}_N = \left(\frac{1}{11}\ \ -\frac{2}{11}\right)$$

$$\min\left\{-, \frac{16}{11}\Big/\frac{2}{11}\right\} = 8.$$

Hence x_4 enters the basis. $A_{B'} \leftarrow (a_2, a_5, a_1, a_4)$.

Iteration 2

Step 2: $x_{B'} = (\frac{7}{2}\ \ 6\ \ 3\ \ \frac{3}{2}) \geqslant 0$. Hence $x = (3\ \ \frac{7}{2}\ \ 0\ \ \frac{3}{2}\ \ 6\ \ 0)$ is an optimal solution to the revised problem.

Noting that the added constraint is an upper-bound constraint means that we can also reoptimize without increasing the size of the basis by using the dual simplex algorithm with upper bounds. In this case we have:

Iteration 1

$$A_B = (a_2, a_5, a_1), \quad A_{N_1} = (a_3, a_4), \quad N_2 = \emptyset$$

$$c_{N_1} = \begin{pmatrix} -\frac{3}{11} & -\frac{16}{11} \end{pmatrix} \leqslant 0,$$

so the basis is dual feasible.

$$x_B = \begin{pmatrix} \frac{40}{11} & \frac{75}{11} & \frac{36}{11} \end{pmatrix}.$$

Because $x_{B_1} = x_1 > h_1$, the basis is not primal feasible.

The dual simplex algorithm then removes x_1 from the basis at its upper bound and calculates (as above) that x_4 enters the basis.

Iteration 2

$$A_B = (a_2, a_5, a_4), \quad A_{N_1} = (a_3), \quad A_{N_2} = (a_1).$$

$$\bar{c}_{N_1} = (-1) \leqslant 0, \quad \bar{c}_{N_2} = (8) \geqslant 0,$$

so the basis remains dual feasible.

$x_B = (\frac{7}{2} \quad 6 \quad \frac{3}{2})$. Because $0 \leqslant x \leqslant h_B$, the basis is primal feasible and hence optimal.

4. SUBGRADIENT OPTIMIZATION

Here we consider an algorithm for solving linear programs whose roots are in nonlinear, nondifferentiable optimization. Consider the linear program

$$\zeta = \min \sum_{i=1}^{l} \lambda_i d_i$$

$$\sum_{i=1}^{l} \lambda_i g_{ij} = c_j \quad \text{for } j = 1, \ldots, n$$

$$0 \leqslant \lambda_i \leqslant h_i \quad \text{for } i = 1, \ldots, l.$$

By duality it can be shown (see Section II.3.6) that this problem can be restated as

$$\zeta = \max_{x \in R^n} \min_{0 \leqslant \lambda \leqslant h} \left[\sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{l} \lambda_i \left(d_i - \sum_{j=1}^{n} g_{ij} x_j \right) \right].$$

Now to solve the inner optimization problem for fixed x, we can set $\lambda_i = 0$ if $d_i - \sum_{j=1}^{n} g_{ij} x_j > 0$, and $\lambda_i = h_i$ otherwise. Thus there are a finite number of candidate solutions $\lambda^k \in R_+^l$, $k \in K$, where $\lambda_i^k \in \{0, h_i\}$. So we can rewrite the problem as

$$\zeta = \max_{x \in R^n} \min_{k \in K} [(c - \lambda^k G)x + \lambda^k d],$$

or more generally as

$$\zeta = \max_{x \in R^n} f(x), \tag{4.1}$$

where

$$f(x) = \min_{i \in I} (a^i x - b_i) \quad \text{and} \quad I = \{1, \ldots, m\} \quad \text{is a finite set.} \tag{4.2}$$

In other words a general linear program can be transformed to the nonlinear optimization problem (4.1), where typically m is much larger than n. In this section, we present an algorithm for problem (4.1).

Figure 4.1 illustrates f given by (4.2) for $n = 1$. The heavy lines give $f(x)$, and point B is the optimum solution x^* with value $\zeta = f(x^*)$.

We now develop an important property of the function f.

Definition 4.1. A function g: $R^n \to R^1$ is *concave* if

$$g(\alpha x^1 + (1 - \alpha)x^2) \geq \alpha g(x^1) + (1 - \alpha)\, g(x^2) \quad \text{for all } x^1, x^2 \in R^n \text{ and all } 0 \leq \alpha \leq 1.$$

Note that the definition simply states that the function is underestimated by linear interpolation (see Figure 4.2).

This suggests the following proposition.

Proposition 4.1. *Let* $f(x) = \min_{i=1,\ldots,m} (a^i x - b_i)$. *Then* $f(x)$ *is concave.*

Proof. Let $x^3 = \alpha x^1 + (1 - \alpha)x^2$ and $f(x^j) = a^{i(j)}x^j - b_{i(j)}$ for $j = 1, 2, 3$. Then

$$\begin{aligned} f(x^3) = a^{i(3)}x^3 - b_{i(3)} &= a^{i(3)}(\alpha x^1 + (1 - \alpha)x^2) - b_{i(3)} \\ &= \alpha(a^{i(3)}x^1 - b_{i(3)}) + (1 - \alpha)(a^{i(3)}x^2 - b_{i(3)}) \\ &\geq \alpha(a^{i(1)}x^1 - b_{i(1)}) + (1 - \alpha)(a^{i(2)}x^2 - b_{i(2)}) \\ &= \alpha f(x^1) + (1 - \alpha)f(x^2). \end{aligned}$$

■

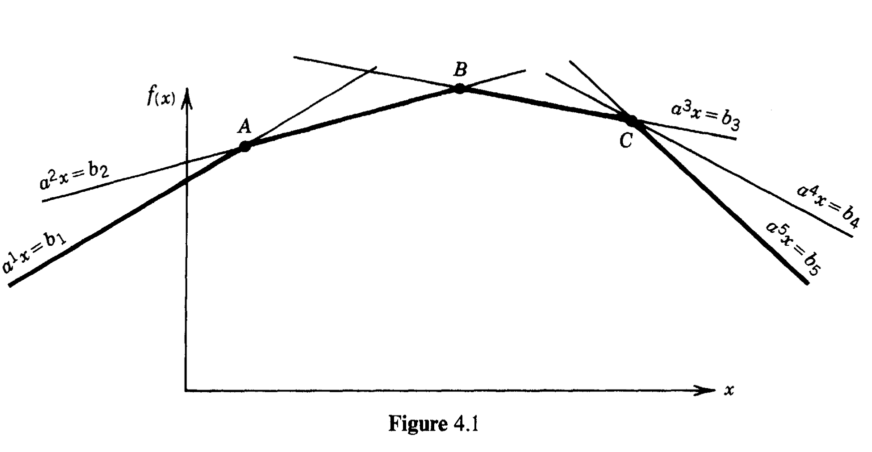

Figure 4.1

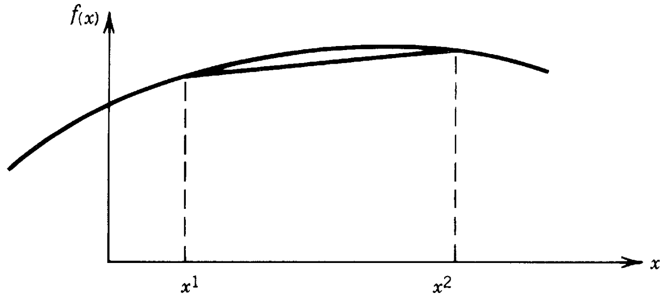

Figure 4.2

An alternative characterization of concave functions is given by the following proposition.

Proposition 4.2. *A function g: $R^n \to R^1$ is concave if and only if for any $x^* \in R^n$ there exists an $s \in R^n$ such that $g(x^*) + s(x - x^*) \geqslant g(x)$ for all $x \in R^n$.*

The characterization is illustrated in Figure 4.3. Note that s is the slope of the hyperplane that supports the set $\{(x, z) \in R^{n+1}: z \leqslant g(x)\}$ at $(x, z) = (x^*, g(x^*))$.

Comparing Figures 4.1 and 4.3, we see that in Figure 4.1 there is not a unique supporting hyperplane at the points A, B, and C, while for the smooth function g in Figure 4.3, the supporting hyperplane is unique at each point.

Figure 4.4 illustrates Proposition 4.2 for $x \in R^2$. Contours of $\{x: g(x) = c\}$ are shown for different values of c along with the supporting hyperplane given by $s(x - x^*) = 0$. By Proposition 4.2, if x satisfies $s(x - x^*) \leqslant 0$, then $g(x) \leqslant g(x^*)$. In other words, if $g(x) > g(x^*)$, then $s(x - x^*) > 0$. Thus if we are at the point x^* and want to increase $g(x)$, we should move to a point x' with $s(x' - x^*) > 0$. One possibility is to move in a direction normal to the hyperplane $s(x - x^*) = 0$. This direction is given by the vector s, which is, when g is differentiable at x^*, the *gradient vector* $\nabla g(x^*) = (\partial g(x^*)/\partial x_1, \ldots, \partial g(x^*)/\partial x_n)$ at $x = x^*$. It is well known that the gradient vector is the local direction of maximum increase of $g(x)$, and $\nabla g(x^*) = 0$ implies that x^* solves $\max\{g(x): x \in R^n\}$.

The classical steepest ascent method for maximizing $g(x)$ is given by the sequence of iterations

$$x^{t+1} = x^t + \theta^t \nabla g(x^t), \quad t = 1, 2, \ldots.$$

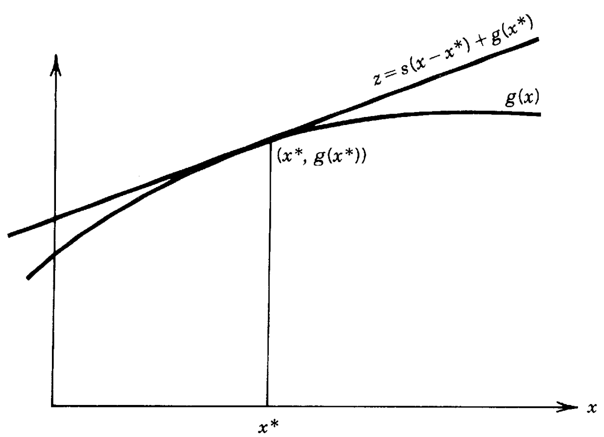

Figure 4.3

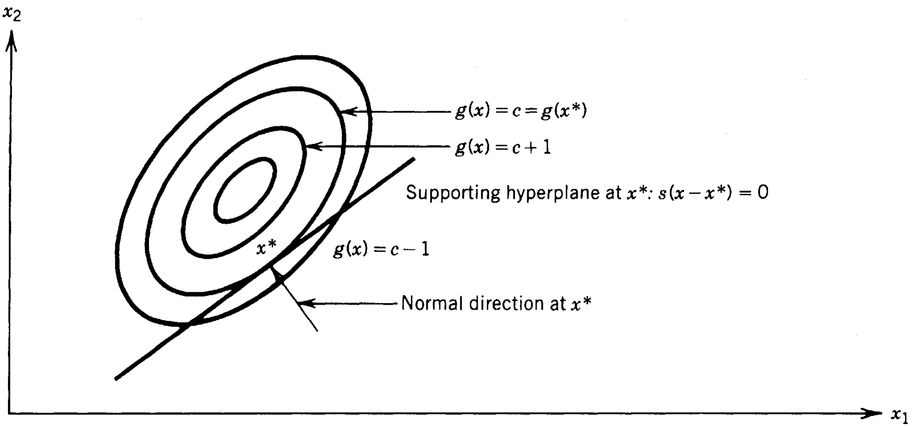

Figure 4.4

With appropriate assumptions on the sequence of step sizes $\{\theta^t\}$, the iterates $\{x^t\}$ converge to a maximizing point.

The potential problems that arise in applying this idea to a nondifferentiable concave function are illustrated in Example 4.1.

Example 4.1

$$f(x_1, x_2) = \min\{-x_1, x_1 - 2x_2, x_1 + 2x_2\}.$$

The contours $f(x) = c$ for $c = 0, -1$, and -2 are shown in Figure 4.5.

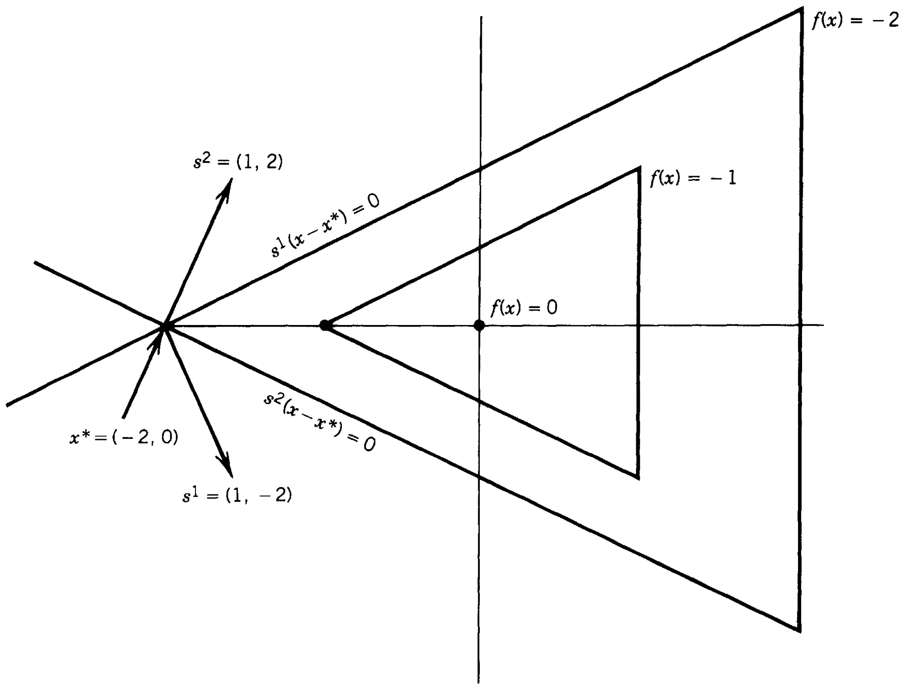

Figure 4.5

In addition, at the point $x^* = (-2 \quad 0)$ we show the supporting hyperplanes $s^i(x - x^*) = 0$, for $i = 1, 2$ where $s^1 = (1 \quad -2)$ and $s^2 = (1 \quad 2)$.

Now consider what happens when we move from x^* in the direction s^1. We have

$$f(x^* + \theta s^1) = f(-2 + \theta, 0 - 2\,\theta) = \min\{2 - \theta, -2 + 5\,\theta, -2 - 3\,\theta\}$$
$$= -2 - 3\,\theta \quad \text{for all } \theta \geqslant 0.$$

Hence $f(x^* + \theta s^1) < f(x^*)$ for all $\theta > 0$. Similar behavior is observed for s^2.

The example illustrates the nonuniqueness of the supporting hyperplanes and also shows that a direction normal to a supporting hyperplane may not be a direction of increase.

There is, however, an alternative point of view, which provides the intuitive justification for moving in a direction normal to any supporting hyperplane at x^*. As we have already noted, if $s(x - x^*) = 0$ is *any* supporting hyperplane at x^*, then any point with a larger objective value than x^* is contained in the half-space $s(x - x^*) > 0$. Now it is a simple geometric exercise to show that if $\hat{x}$ is an optimal solution, a small move in the direction s gives a point that is closer to $\hat{x}$. In particular, there exists $\hat{\theta}$ such that for any $0 < \theta < \hat{\theta}$,

$$\|\hat{x} - (x^* + \theta s)\| < \|\hat{x} - x^*\|$$

(see Figure 4.6.). The notation $\|u\|$, $u \in R^n$, represents the euclidean distance from 0 to u, that is, $\sqrt{u^T u}$.

We now formalize the discussion given above.

Definition 4.2. If g: $R^n \to R^1$ is concave, $s \in R^n$ is a *subgradient* of g at x^* if $s(x - x^*) \geqslant g(x) - g(x^*)$ for all $x \in R^n$.

Definition 4.3. The set $\partial g(x) = \{s \in R^n\colon s \text{ is a subgradient of } g \text{ at } x\}$ is called the *subdifferential* of g at x.

Note that by Proposition 4.2, $\partial g(x) \neq \emptyset$.

Proposition 4.3. *If g is concave on R^n, x^* is an optimal solution of $\max\{g(x)\colon x \in R^n\}$ if and only if $0 \in \partial g(x^*)$.*

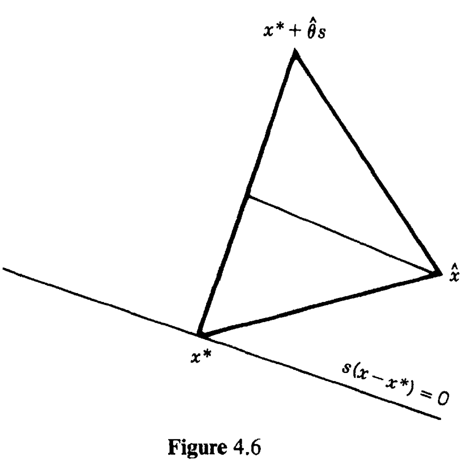

Figure 4.6

Proof. $0 \in \partial g(x^*)$ if and only if $0(x - x^*) \geqslant g(x) - g(x^*)$ for all $x \in R^n$ if and only if $g(x) \leqslant g(x^*)$ for all $x \in R^n$. ■

Now we characterize the subdifferential of $f(x)$ given by (4.2).

Proposition 4.4. *Let* $f(x) = \min_{i=1,\ldots,m} (a^i x - b_i)$ *and let* $I(x^*) = \{i: f(x^*) = a^i x^* - b_i\}$.

1. a^i *is a subgradient of f at* x^* *for all* $i \in I(x^*)$.
2. $\partial f(x^*) = \{s \in R^n: s = \Sigma_{i\in I(x^*)} \lambda_i a^i, \Sigma_{i\in I(x^*)} \lambda_i = 1, \lambda_i \geqslant 0$ *for* $i \in I(x^*)\}$.

Proof.

1. If $i \in I(x^*)$, then $a^i(x - x^*) = (a^i x - b_i) - (a^i x^* - b^i) \geqslant f(x) - f(x^*)$ for all $x \in R^n$, so that $a^i \in \partial f(x^*)$.
2. A proof is obtained by using statement 1 of Proposition 4.4 along with the Farkas lemma. ■

The following algorithm can use any subgradient at each step, but for computational purposes one of the extreme directions a^i will be chosen.

The Subgradient Algorithm for (4.1)

Step 1 (Initialization): Choose a starting point x^1 and let $t = 1$.

Step 2: Given x^t, choose any subgradient $s^t \in \partial f(x^t)$. If $s^t = 0$, then x^t is an optimal solution. Otherwise go to Step 3.

Step 3: Let $x^{t+1} = x^t + \theta_t s^t$ for some $\theta_t > 0$. (Procedures for selecting θ_t are given below.) Let $t \leftarrow t + 1$ and return to Step 2.

Two schemes for selecting $\{\theta_t\}$ are the following:

i. A divergent series: $\Sigma_{t=1}^{\infty} \theta_t \to \infty$, $\theta_t \to 0$ as $t \to \infty$.
ii. A geometric series: $\theta_t = \theta_0 \rho^t$, or $\theta_t = [\bar{f} - f(x^t)]\rho^t/\|s^t\|^2$ where $0 < \rho < 1$ and $\bar{f}$ is a target, or upper bound on the optimal value ζ of (4.1).

Series i is satisfactory theoretically, since it converges to an optimal point. But in practice the convergence is much too slow. Series ii, which is recommended in practice, is less satisfactory theoretically. The convergence is "geometric", but the limit point is only an optimal point if the initial choices of (θ_0, ρ) or $(\bar{f}, \rho)$ are sufficiently large. In practice, appropriate values can typically be found after a little testing, and step sizes closely related to a geometric series of type ii will be used in our applications of the subgradient algorithm in Part II.

Ideally the subgradient algorithm can be stopped when, on some iteration t, we find $s^t = 0 \in \partial f(x^t)$. However, in practice this rarely happens, since the algorithm just chooses one subgradient s^t and has no way of showing $0 \in \partial f(x^t)$ as a convex combination of subgradients. Hence the typical stopping rule is either to stop after a fixed number of iterations or to stop if the function has not increased by at least a certain amount within a given number of iterations.

Example 4.2. Consider $\max\{f(x): x \in R^2\}$, where

$$f(x) = \min\{f_i(x): i = 1, \ldots, 5\}$$

and

$$\begin{aligned} f_1(x) &= x_1 - 2x_2 + 4 \\ f_2(x) &= -5x_1 - x_2 + 20 \\ f_3(x) &= 2x_1 + 2x_2 - 7 \\ f_4(x) &= x_1 \\ f_5(x) &= x_2. \end{aligned}$$

We apply the subgradient algorithm with $\theta_t = (0.9)^t$ and initial point $x^1 = (0 \quad 0)$. The results of 25 iterations are shown in Table 4.1, in which the last column, $i(t)$, gives the index of the function that defines the subgradient. The best solution of value 2.30 is found at iteration 13. The optimal solution is $(x_1 \quad x_2) = (\frac{52}{17} \quad \frac{40}{17})$ of value $\frac{40}{17} = 2.353$.

Table 4.1.

t	x_1^t	x_2^t	$f(x^t)$	ρ^t	$i(t)$
1	0.000	0.000	−7.000	0.900	3
2	1.800	1.800	0.200	0.810	3
3	3.420	3.420	−0.520	0.729	2
4	−0.225	2.691	−2.068	0.656	3
5	1.087	4.003	−2.919	0.590	1
6	1.678	2.822	0.033	0.531	1
7	2.209	1.759	0.937	0.478	3
8	3.166	2.716	1.455	0.430	2
9	1.013	2.285	−0.402	0.387	3
10	1.788	3.060	−0.332	0.349	1
11	2.137	2.363	1.411	0.314	1
12	2.451	1.735	1.372	0.282	3
13	3.016	2.300	2.300	0.254	5
14	3.016	2.554	1.907	0.229	1
15	3.244	2.097	1.681	0.206	2
16	2.215	1.891	1.212	0.185	3
17	2.585	2.262	2.062	0.167	1
18	2.752	1.928	1.928	0.150	5
19	2.752	2.078	2.078	0.135	5
20	2.752	2.213	2.213	0.122	5
21	2.752	2.335	2.083	0.109	1
22	2.862	2.116	2.116	0.098	5
23	2.862	2.214	2.214	0.089	5
24	2.862	2.303	2.256	0.080	1
25	2.941	2.144	2.144	0.072	5

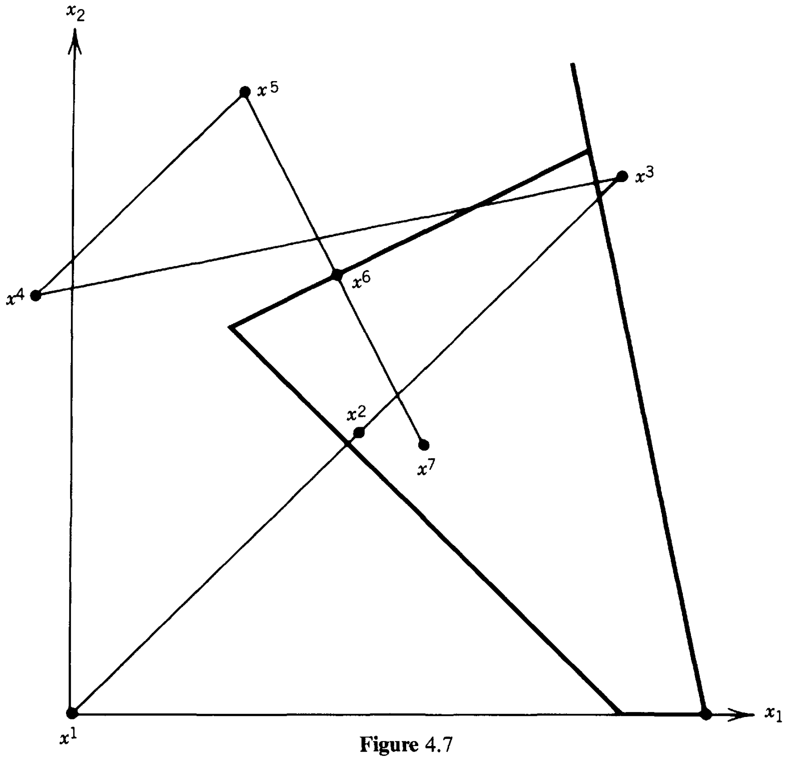

Figure 4.7

We can also view the problem as one of finding (x_1, x_2) such that the smallest slack variable y_i of the constraints

$$\begin{aligned}
-x_1 + 2x_2 + y_1 \qquad\qquad\qquad &= 4\\
5x_1 + x_2 \quad + y_2 \qquad\qquad &= 20\\
-2x_1 - 2x_2 \qquad + y_3 \qquad &= -7\\
-x_1 \qquad\qquad\qquad + y_4 \quad &= 0\\
-x_2 \qquad\qquad\qquad + y_5 &= 0
\end{aligned}$$

is as large as possible (see Figure 4.7). With this geometry, each subgradient step is in the direction of the normal to the constraint whose slack variable is smallest.

Because the magnitudes of the constraint coefficients are different, the five subgradients have different magnitudes which can substantially bias the progress of the algorithm. This suggests the use of normalized subgradients $s/\|s\|$ in the subgradient algorithm. For Example 4.2, this gives the iterations shown in Table 4.2. Note that more rapid convergence is achieved using normalized subgradients.

Finally suppose that $x \in R^n$ must satisfy some linear constraints, say $x \in C$. Thus we have the problem

(4.3) $$\eta = \max\{f(x)\colon x \in C\}, \quad \text{where } f(x) = \min_{i=1,\ldots,m} (a^i x - b_i).$$

The subgradient algorithm for (4.3) is as before, except that Step 3 is modified to maintain feasibility.

Table 4.2.

t	x_1^t	x_2^t	$f(x^t)$	ρ^t	$i(t)$
1	0.000	0.000	−7.000	0.900	3
2	0.636	0.636	−4.454	0.810	3
3	1.209	1.209	−2.163	0.729	3
4	1.725	1.725	−0.101	0.656	3
5	2.189	2.189	1.754	0.590	3
6	2.606	2.606	1.394	0.531	1
7	2.844	2.131	2.131	0.478	5
8	2.844	2.609	1.626	0.430	1
9	3.036	2.224	2.224	0.387	5
10	3.036	2.611	1.813	0.349	1
11	3.192	2.300	1.739	0.314	2
12	2.885	2.238	2.238	0.282	5
13	2.885	2.520	1.844	0.254	1
14	2.998	2.293	2.293	0.229	5
15	2.998	2.522	1.954	0.206	1
16	3.090	2.338	2.211	0.185	2
17	2.909	2.301	2.301	0.167	5
18	2.909	2.468	1.972	0.150	1
19	2.976	2.334	2.308	0.135	1
20	3.036	2.213	2.213	0.122	5
21	3.036	2.335	2.335	0.109	5
22	3.036	2.444	2.148	0.098	1
23	3.080	2.356	2.243	0.089	2
24	2.993	2.339	2.316	0.080	1
25	3.029	2.267	2.267	0.072	5

Step 3′: Let $y^{t+1} = x^t + \theta_t s^t$ for some $\theta_t > 0$ and let $x^{t+1} = \arg\min_{x \in C} \|x - y^{t+1}\|$.

In other words, x^{t+1} is the projection of y^{t+1} onto the feasible region C. A typical application is to have $C = R_+^n$, in which case $x_j^{t+1} = \max(x_j^t + \theta_t s_j^t, 0)$ for $j = 1, \ldots, n$

5. NOTES

Sections I.2.1–I.2.3.

Chvátal (1983) gave a modern and comprehensive treatment of linear programming, with the exception of the significant post-1983 developments covered in Sections I.6.2–I.6.4. Some earlier books are Charnes and Cooper (1961), Dantzig (1963), Gass (1975), Hadley (1962), and Murty (1976).

Section I.2.4

The use of subgradient directions in the solution of large-scale linear programs that arise from combinatorial optimization problems was instigated by Held and Karp (1970, 1971) in a study of the traveling salesman problem. Held et al. (1974) investigated the behavior of a subgradient algorithm in a variety of combinatorial problems. A theoretical analysis of the convergence of subgradient algorithms is given by Goffin (1977). Subgradients and subgradient algorithms are also discussed by Grinold (1970, 1972), Camerini et al. (1975), Shapiro (1979a, b), and Sandi (1979).