11.4 Integer programming duality

In this section, we develop the duality theory of integer programming. This in turn leads to a method for obtaining tight bounds, that are particularly useful for branch and bound. The methodology is closely related to the subject of Section 4.10, but our discussion here is self-contained.

We consider the integer programming problem

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{D}\mathbf{x} \geq \mathbf{d} \\ & \mathbf{x} \text{ integer}, \end{aligned} \tag{11.5}$$

and assume that $\mathbf{A}$, $\mathbf{D}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$ have integer entries. Let Z_{IP} the optimal cost and let

$$X = \{\mathbf{x} \text{ integer} \mid \mathbf{D}\mathbf{x} \geq \mathbf{d}\}.$$

In order to motivate the method, we assume that optimizing over the set X can be done efficiently; for example X may represent the set of feasible solutions to an assignment problem. However, adding the constraints $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ to the problem, makes the problem difficult to solve. We next consider the idea of introducing a dual variable for every constraint in $\mathbf{A}\mathbf{x} \geq \mathbf{b}$. Let $\mathbf{p} \geq \mathbf{0}$ be a vector of dual variables (also called *Lagrange multipliers*) that has the same dimension as the vector $\mathbf{b}$. For a fixed vector $\mathbf{p}$, we introduce the problem

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x} + \mathbf{p}'(\mathbf{b} - \mathbf{A}\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in X, \end{aligned} \tag{11.6}$$

and denote its optimal cost by $Z(\mathbf{p})$. We will say that we *relax* or *dualize* the constraints $\mathbf{A}\mathbf{x} \geq \mathbf{b}$. For a fixed $\mathbf{p}$, the above problem can be solved efficiently, as we are optimizing a linear objective over the set X. We next observe that $Z(\mathbf{p})$ provides a bound on Z_{IP}.

Lemma 11.1 *If the problem (11.5) has an optimal solution and if $\mathbf{p} \geq \mathbf{0}$, then $Z(\mathbf{p}) \leq Z_{\text{IP}}$.*

Proof. Let $\mathbf{x}^*$ denote an optimal solution to (11.5). Then, $\mathbf{b} - \mathbf{A}\mathbf{x}^* \leq \mathbf{0}$ and, therefore,

$$\mathbf{c}'\mathbf{x}^* + \mathbf{p}'(\mathbf{b} - \mathbf{A}\mathbf{x}^*) \leq \mathbf{c}'\mathbf{x}^* = Z_{\text{IP}}.$$

Since $\mathbf{x}^* \in X$,

$$Z(\mathbf{p}) \leq \mathbf{c}'\mathbf{x}^* + \mathbf{p}'(\mathbf{b} - \mathbf{A}\mathbf{x}^*),$$

and therefore, $Z(\mathbf{p}) \leq Z_{\text{IP}}$. □

Since problem (11.6) provides a lower bound to the integer programming problem (11.5) for all $\mathbf{p} \geq \mathbf{0}$, it is natural to consider the tightest such

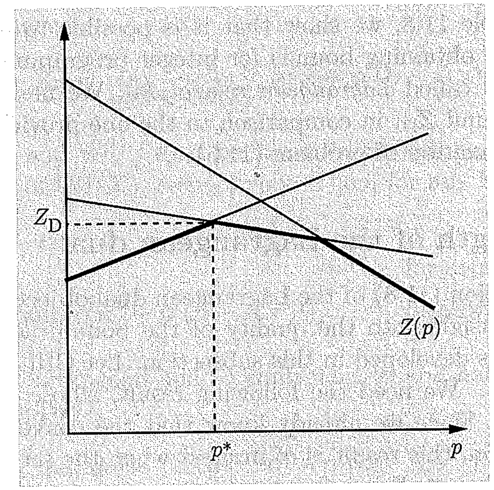

Figure 11.6: The function $Z(\mathbf{p})$ is concave and piecewise linear.

bound. For this reason, we introduce the problem

$$\begin{aligned} \text{maximize} \quad & Z(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{p} \geq \mathbf{0}. \end{aligned} \tag{11.7}$$

We will refer to problem (11.7) as the *Lagrangean dual.* Let

$$Z_{\text{D}} = \max_{\mathbf{p} \geq \mathbf{0}} Z(\mathbf{p}).$$

Suppose for instance, that $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^m\}$. Then $Z(\mathbf{p})$ can be also written as

$$Z(\mathbf{p}) = \min_{i=1,\ldots,m} \left(\mathbf{c}'\mathbf{x}^i + \mathbf{p}'(\mathbf{b} - \mathbf{A}\mathbf{x}^i)\right). \tag{11.8}$$

The function $Z(\mathbf{p})$ is concave and piecewise linear, since it is the minimum of a finite collection of linear functions of $\mathbf{p}$ (see Theorem 1.1 in Section 1.3 and Figure 11.6). As a consequence, the problem of computing Z_{D} [namely, problem (11.7)] can be recast as a linear programming problem, but with a very large number of constraints.

It is clear from Lemma 11.1 that weak duality holds:

Theorem 11.2 *We have $Z_{\text{D}} \leq Z_{\text{IP}}$.*

The previous theorem represents the weak duality theory of integer programming. Unlike linear programming, integer programming does not have a strong duality theory. (Compare with Theorem 4.18 in Section 4.10.)

Indeed in Example 11.8, we show that it is possible to have $Z_D < Z_{IP}$. The procedure of obtaining bounds for integer programming problems by calculating Z_D is called *Lagrangean relaxation.* We next investigate the quality of the bound Z_D, in comparison to the one provided by the linear programming relaxation of problem (11.5).

On the strength of the Lagrangean dual

The characterization (11.8) of the Lagrangean dual objective does not provide particular insight into the quality of the bound. A more revealing characterization is developed in this subsection. Let CH(X) be the convex hull of the set X. We need the following result, whose proof is outlined in Exercise 11.8. Since we already know that the convex hull of a finite set is a polyhedron, this result is of interest when the set $\{\mathbf{x} \mid \mathbf{Dx} \geq \mathbf{d}\}$ is unbounded and the set X is infinite.

> **Theorem 11.3** *We assume that the system of linear inequalities* $\mathbf{Dx} \geq \mathbf{d}$ *has a feasible solution, and that the matrix* $\mathbf{D}$ *and the vector* $\mathbf{d}$ *have integer entries. Let*
>
> $$X = \{\mathbf{x} \text{ integer} \mid \mathbf{Dx} \geq \mathbf{d}\}.$$
>
> *Then* CH(X) *is a polyhedron.*

The next theorem, which is the central result of this section, characterizes the Lagrangean dual as a linear programming problem.

> **Theorem 11.4** *The optimal value* Z_D *of the Lagrangean dual is equal to the optimal cost of the following linear programming problem:*
>
> $$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x} \\ \text{subject to} \quad & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{x} \in \mathrm{CH}(X). \end{aligned} \tag{11.9}$$

Proof. By definition,

$$Z(\mathbf{p}) = \min_{\mathbf{x} \in X} \big(\mathbf{c}'\mathbf{x} + \mathbf{p}'(\mathbf{b} - \mathbf{Ax})\big).$$

Since the objective function is linear in $\mathbf{x}$, the optimal cost remains the same if we allow convex combinations of the elements of X. Therefore,

$$Z(\mathbf{p}) = \min_{\mathbf{x} \in \mathrm{CH}(X)} \big(\mathbf{c}'\mathbf{x} + \mathbf{p}'(\mathbf{b} - \mathbf{Ax})\big),$$

and hence, we have

$$Z_D = \max_{\mathbf{p} \geq \mathbf{0}} \min_{\mathbf{x} \in \mathrm{CH}(X)} \big(\mathbf{c}'\mathbf{x} + \mathbf{p}'(\mathbf{b} - \mathbf{Ax})\big).$$

Let $\mathbf{x}^k$, $k \in K$, and $\mathbf{w}^j$, $j \in J$, be the extreme points and a complete set of extreme rays of CH(X), respectively. Then, for any fixed $\mathbf{p}$, we have

$$Z(\mathbf{p}) = \begin{cases} -\infty, & \text{if } (\mathbf{c}' - \mathbf{p}'\mathbf{A})\mathbf{w}^j < 0, \\ & \text{for some } j \in J, \\ \min\limits_{k \in K} \big(\mathbf{c}'\mathbf{x}^k + \mathbf{p}'(\mathbf{b} - \mathbf{Ax}^k)\big), & \text{otherwise.} \end{cases}$$

Therefore, the Lagrangean dual is equivalent to and has the same optimal value as the problem

$$\begin{aligned} \text{maximize} \quad & \min_{k \in K} \big(\mathbf{c}'\mathbf{x}^k + \mathbf{p}'(\mathbf{b} - \mathbf{Ax}^k)\big) \\ \text{subject to} \quad & (\mathbf{c}' - \mathbf{p}'\mathbf{A})\mathbf{w}^j \geq 0, \qquad j \in J, \\ & \mathbf{p} \geq \mathbf{0}, \end{aligned}$$

or equivalently,

$$\begin{aligned} \text{maximize} \quad & y \\ \text{subject to} \quad & y + \mathbf{p}'(\mathbf{Ax}^k - \mathbf{b}) \leq \mathbf{c}'\mathbf{x}^k, \qquad k \in K, \\ & \mathbf{p}'\mathbf{Aw}^j \leq \mathbf{c}'\mathbf{w}^j, \qquad j \in J, \\ & \mathbf{p} \geq \mathbf{0}. \end{aligned}$$

Taking the linear programming dual of the above problem, and using strong duality, we obtain that Z_D is equal to the optimal cost of the problem

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\left(\sum_{k \in K} \alpha_k \mathbf{x}^k + \sum_{j \in J} \beta_j \mathbf{w}^j\right) \\ \text{subject to} \quad & \sum_{k \in K} \alpha_k = 1 \\ & \mathbf{A}\left(\sum_{k \in K} \alpha_k \mathbf{x}^k + \sum_{j \in J} \beta_j \mathbf{w}^j\right) \geq \mathbf{b} \\ & \alpha_k, \beta_j \geq 0, \qquad k \in K,\ j \in J. \end{aligned}$$

Since,

$$\mathrm{CH}(X) = \left\{ \sum_{k \in K} \alpha_k \mathbf{x}^k + \sum_{j \in J} \beta_j \mathbf{w}^j \;\middle|\; \sum_{k \in K} \alpha_k = 1,\ \alpha_k, \beta_j \geq 0,\ k \in K,\ j \in J \right\},$$

the result follows. □

Example 11.8 (Illustration of Lagrangean relaxation) Consider the problem

$$\begin{array}{ll} \text{minimize} & 3x_1 - x_2 \\ \text{subject to} & x_1 - x_2 \geq -1 \\ & -x_1 + 2x_2 \leq 5 \\ & 3x_1 + 2x_2 \geq 3 \\ & 6x_1 + x_2 \leq 15 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \text{ integer.} \end{array}$$

We relax the first constraint $x_1 - x_2 \geq -1$, and we let X be the set of integer vectors that satisfy the remaining constraints. The set X, shown in Figure 11.7, is then

$$X = \{(1,0),(2,0),(1,1),(2,1),(0,2),(1,2),(2,2),(1,3),(2,3)\}.$$

For $p \geq 0$, we have

$$Z(p) = \min_{(x_1,x_2)\in X} \big(3x_1 - x_2 + p(-1 - x_1 + x_2)\big),$$

which is plotted in Figure 11.8.

Since there are nine points in X, $Z(p)$ is the minimum of nine linear functions. The function $Z(p)$ turns out to be equal to

$$Z(p) = \begin{cases} -2 + p, & 0 \leq p \leq 5/3, \\ 3 - 2p, & 5/3 \leq p \leq 3, \\ 6 - 3p, & p \geq 3. \end{cases}$$

The Lagrangean dual is maximized for $p = 5/3$, and the optimal value is $Z_D = Z(5/3) = -1/3$. For $p = 5/3$, the corresponding elements of X are $(1,0)$ and $(0,2)$.

In order to illustrate Theorem 11.4, we find first the convex hull $\text{CH}(X)$ of X, and intersect it with the constraint $x_1 - x_2 \geq -1$, forming the shaded polyhedron in Figure 11.7. Optimizing the original objective function $3x_1 - x_2$ over this polyhedron, we obtain that the optimal solution is $(1/3, 4/3)$ with value $-1/3$, which is the same as Z_D.

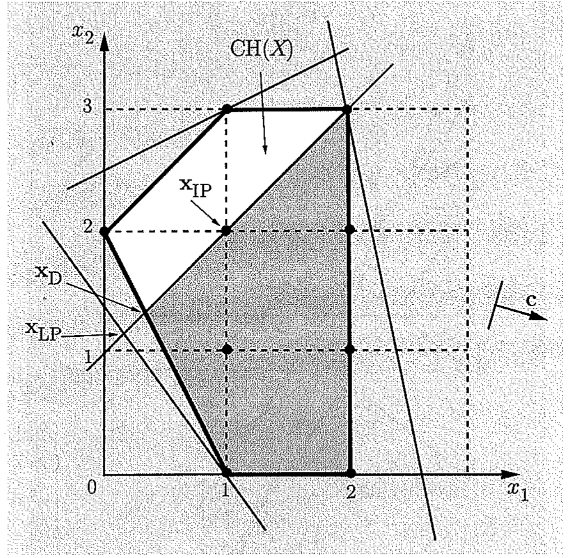

Figure 11.7: The points shown are elements of X. The convex hull of X is the set outlined by the thicker lines. The shaded polyhedron represents the intersection of $\text{CH}(X)$ with the set of vectors that satisfy $x_1 - x_2 \geq -1$. The optimal solution to problem (11.9) is $\mathbf{x}_D = (1/3, 4/3)$, and its cost Z_D is equal to $-1/3$. Note that the optimal solution to the linear programming relaxation is the vector $\mathbf{x}_{LP} = (1/5, 6/5)$, resulting in a lower bound $Z_{LP} = -9/5$. The optimal solution to the integer programming problem is $\mathbf{x}_{IP} = (1,2)$, and $Z_{IP} = 1$. Note that $Z_{LP} < Z_D < Z_{IP}$.

Although we presented the method for the case where the relaxed constraints were inequalities, the method is exactly the same even if we have equality constraints. The only difference is that the corresponding Lagrange multipliers are unrestricted in sign.

Having characterized the optimal value of the Lagrangean dual as the solution to a linear programming problem, it is natural to compare it with the optimal cost Z_{IP} and the optimal cost Z_{LP} of the linear programming relaxation

$$\begin{array}{ll} \text{minimize} & \mathbf{c}'\mathbf{x} \\ \text{subject to} & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{Dx} \geq \mathbf{d}. \end{array}$$

In general, the following ordering holds among Z_{LP}, Z_{IP}, and Z_D:

$$Z_{LP} \leq Z_D \leq Z_{IP}.$$

The first inequality follows from Theorem 11.4, because

$$\text{CH}(X) \subset \{\mathbf{x} \mid \mathbf{Dx} \geq \mathbf{d}\},$$

and the second inequality follows from Theorem 11.2. In the next example, we show that, depending on the objective function, these inequalities can be strict.

Example 11.9 We refer again to Example 11.8. It can be verified that we have the following possibilities:

(a) For the original objective function $3x_1 - x_2$, we have $Z_{LP} < Z_D < Z_{IP}$.

(b) If we change the objective function to $-x_1 + x_2$, we have $Z_{LP} < Z_D = Z_{IP}$.

(c) For the objective function $-x_1 - x_2$, we have $Z_{LP} = Z_D = Z_{IP}$.

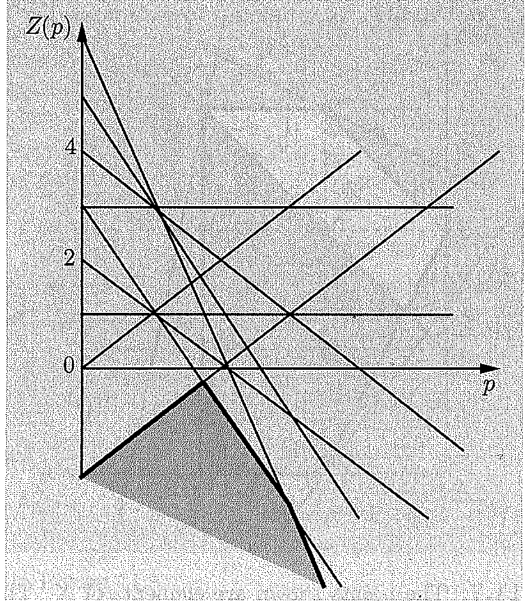

Figure 11.8: The function $Z(p)$. Each line is the plot of the function $3x_1 - x_2 + p(-1 - x_1 + x_2)$, where (x_1, x_2) is set to some particular element of X. The lower envelope of these lines is the function $Z(p)$. The maximum of $Z(p)$ is $-1/3$ and is attained for $p = 5/3$.

One can also construct an example, in which the relation $Z_{\mathrm{LP}} = Z_{\mathrm{D}} < Z_{\mathrm{IP}}$ holds. In this example, however, such an ordering is not possible.

Using Theorem 11.4, we can make the following observations:

Corollary 11.1

(a) *We have $Z_{\mathrm{IP}} = Z_{\mathrm{D}}$ for all cost vectors* **c**, *if and only if*

$$\mathrm{CH}\Big(X \cap \{\mathbf{x} \mid \mathbf{Ax} \geq \mathbf{b}\}\Big) = \mathrm{CH}(X) \cap \{\mathbf{x} \mid \mathbf{Ax} \geq \mathbf{b}\}.$$

(b) *We have $Z_{\mathrm{LP}} = Z_{\mathrm{D}}$ for all cost vectors* **c**, *if*

$$\mathrm{CH}(X) = \{\mathbf{x} \mid \mathbf{Dx} \geq \mathbf{d}\}.$$

It is interesting to observe that if the polyhedron $\{\mathbf{x} \mid \mathbf{Dx} \geq \mathbf{d}\}$, has integer extreme points, then $\mathrm{CH}(X) = \{\mathbf{x} \mid \mathbf{Dx} \geq \mathbf{d}\}$, and therefore Z_{D} is equal to the optimal cost of the linear programming relaxation.

Example 11.10 (Improved bounds for the traveling salesman problem) The following set of constraints for the traveling salesman problem on an undirected graph $G = (\mathcal{N}, \mathcal{E})$, was introduced in Section 10.3:

$$\begin{aligned} \sum_{e \in \delta(\{i\})} x_e &= 2, & i \in \mathcal{N}, \\ \sum_{e \in E(S)} x_e &\leq |S| - 1, & S \subset \mathcal{N},\ S \neq \emptyset, \mathcal{N}, \\ x_e &\in \{0,1\}. \end{aligned}$$

[Recall that x_e indicates whether edge e participates in the tour. Also, $\delta(\{i\})$ is the set of edges incident to node i, and $E(S)$ is the set of edges with both endpoints in S.] We choose node 1 as a special node, called the root node, and add the redundant equality

$$\sum_{e \in E(\mathcal{N} \setminus \{1\})} x_e = |\mathcal{N}| - 2.$$

The formulation can be then written as follows.

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in \mathcal{E}} c_e x_e \\ \text{subject to} \quad & \sum_{e \in \delta(\{i\})} x_e = 2, & i \in \mathcal{N} \setminus \{1\}, \\ & \sum_{e \in \delta(\{1\})} x_e = 2, \\ & \sum_{e \in E(S)} x_e \leq |S| - 1, & S \subset \mathcal{N},\ S \neq \emptyset, \mathcal{N}, \\ & \sum_{e \in E(\mathcal{N} \setminus \{1\})} x_e = |\mathcal{N}| - 2, \\ & x_e \in \{0,1\}. \end{aligned}$$

Next, we apply the Lagrangean relaxation methodology to the above formulation, by dualizing the constraints

$$\sum_{e \in \delta(\{i\})} x_e = 2, \qquad i \in \mathcal{N} \setminus \{1\}. \tag{11.10}$$

The binary vectors satisfying all the constraints except for (11.10) constitute the set X. We define an 1-*tree* to be a tree involving all nodes in $\mathcal{N} \setminus \{1\}$, and two additional edges incident to node 1 (see Figure 11.9). It is not hard to show that X is the set of vectors that correspond to 1-trees. As a result, we can optimize over X efficiently, by using the greedy minimum spanning tree algorithm on $\mathcal{N} \setminus \{1\}$, and then adding the two smallest cost edges from node 1. Moreover, it is known that $\mathrm{CH}(X)$ is the polyhedron described by all inequalities except (11.10), and where we replace the integrality constraints with $\mathbf{x} \geq \mathbf{0}$.

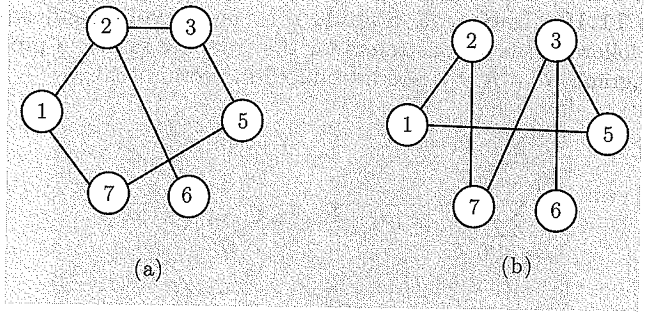

Figure 11.9: Two 1-trees.

We consider the linear programming relaxation of the original formulation, which is

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in \mathcal{E}} c_e x_e \\ \text{subject to} \quad & \sum_{e \in \delta(\{i\})} x_e = 2, \qquad i \in \mathcal{N}, \\ & \sum_{e \in E(S)} x_e \leq |S| - 1, \quad S \subset \mathcal{N},\ S \neq \emptyset, \mathcal{N}, \\ & x_e \geq 0. \end{aligned}$$

Notice that the constraint $x_e \leq 1$ is implied by the subtour elimination constraints for $S = e = \{i, j\}$, and is therefore omitted.

By Corollary 11.1(b), $Z_{\mathrm{D}} = Z_{\mathrm{LP}}$. The optimal value Z_{D} of the Lagrangean dual is called the *Held-Karp lower bound.* As we mentioned, the calculation of $Z(\mathbf{p})$ for a fixed vector $\mathbf{p}$ can be done efficiently. This leads to an effective algorithm for computing the Held-Karp lower bound.

In general, the combination of branch and bound and Lagrangean relaxation yields some of the most effective methods for many classes of integer programming problems.

Solution of the Lagrangean dual

In this subsection, we outline a method for finding the optimal Lagrange multipliers $\mathbf{p}^*$, that solve the Lagrangean dual problem (11.7). To keep the presentation simple, we assume that X is finite and $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^m\}$. Given a particular value of $\mathbf{p}$, we assume that we can calculate $Z(\mathbf{p})$, which we have defined as follows:

$$Z(\mathbf{p}) = \min_{i=1,\ldots,m} \left(\mathbf{c}'\mathbf{x}^i + \mathbf{p}'(\mathbf{b} - \mathbf{A}\mathbf{x}^i)\right).$$

Let $\mathbf{f}_i = \mathbf{b} - \mathbf{A}\mathbf{x}^i$ and $h_i = \mathbf{c}'\mathbf{x}^i$. Then,

$$Z(\mathbf{p}) = \min_{i=1,\ldots,m} \left(h_i + \mathbf{f}_i'\mathbf{p}\right),$$

which is piecewise linear and concave, as discussed earlier.

In order to motivate the following discussion, let us assume for the moment that the function $Z(\mathbf{p})$ is also differentiable. Then, the classical steepest ascent method for maximizing $Z(\mathbf{p})$ is given by the sequence of iterations

$$\mathbf{p}^{t+1} = \mathbf{p}^t + \theta_t \nabla Z(\mathbf{p}^t), \quad t = 1, 2, \ldots$$

In our case, the function $Z(\mathbf{p})$ is not differentiable and thus $\nabla Z(\mathbf{p}^t)$ does not always exist. For this reason, we need to generalize the notion of the gradient to nondifferentiable concave functions. The following alternative characterization of concave functions is helpful in this respect. The proof is based on the supporting hyperplane theorem, which is an extension of the separating hyperplane theorem, and is omitted; see Exercise 11.9.

Lemma 11.2 *A function $f : \Re^n \mapsto \Re$ is concave if and only if for any $\mathbf{x}^* \in \Re^n$, there exists a vector $\mathbf{s} \in \Re^n$ such that*

$$f(\mathbf{x}) \leq f(\mathbf{x}^*) + \mathbf{s}'(\mathbf{x} - \mathbf{x}^*),$$

for all $\mathbf{x} \in \Re^n$.

The vectors $\mathbf{s}$ in Lemma 11.2 provide the required generalization.

Definition 11.1 *Let f be a concave function. A vector $\mathbf{s}$ such that*

$$f(\mathbf{x}) \leq f(\mathbf{x}^*) + \mathbf{s}'(\mathbf{x} - \mathbf{x}^*),$$

for all $\mathbf{x} \in \Re^n$, is called a **subgradient** *of f at $\mathbf{x}^*$. The set of all subgradients of f at $\mathbf{x}^*$ is denoted by $\partial f(\mathbf{x}^*)$ and is called the* **subdifferential** *of f at $\mathbf{x}^*$.*

If the function f is differentiable at $\mathbf{x}^*$, then it can be verified that $\partial f(\mathbf{x}^*) = \{\nabla f(\mathbf{x}^*)\}$. If f is not differentiable, then Lemma 11.2 establishes that the subdifferential is nonempty at every point. Figure 11.10 shows an example of a subgradient. Definition 11.1 is the same as Definition 5.1 in Section 5.3, except that the direction of the inequality is reversed; the reason is that here we are dealing with concave as opposed to convex functions.

Note that the inequality $f(\mathbf{x}) \leq f(\mathbf{x}^*)$ for all $\mathbf{x} \in \Re^n$, is equivalent to saying that $\mathbf{0}$ is a subgradient of f at $\mathbf{x}^*$. This observation is formally recorded in the following lemma, which establishes a necessary and sufficient condition for the maximum of a concave function.

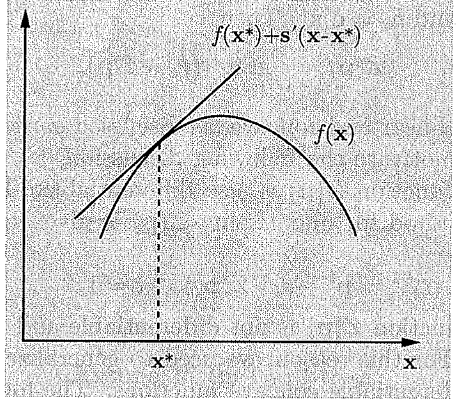

Figure 11.10: A concave function $f(\mathbf{x})$ and a subgradient $\mathbf{s}$ of f at $\mathbf{x}^*$. Note that $f(\mathbf{x}) \le f(\mathbf{x}^*) + \mathbf{s}'(\mathbf{x} - \mathbf{x}^*)$.

Lemma 11.3 *Let $f : \Re^n \mapsto \Re$ be a concave function. A vector $\mathbf{x}^*$ maximizes f over $\Re^n$ if and only if $\mathbf{0} \in \partial f(\mathbf{x}^*)$.*

We next characterize exactly the subdifferential of a piecewise linear concave function; see Figure 11.11 for an illustration. The proof is based on Farkas' lemma and is omitted (Exercise 11.10).

Lemma 11.4 *Let*

$$Z(\mathbf{p}) = \min_{i=1,\ldots,m} (h_i + \mathbf{f}_i'\mathbf{p}),$$
$$E(\mathbf{p}) = \{i \mid Z(\mathbf{p}) = h_i + \mathbf{f}_i'\mathbf{p}\}.$$

Then:

(a) *For every $i \in E(\mathbf{p}^*)$, $\mathbf{f}_i$ is a subgradient of the function $Z(\cdot)$ at $\mathbf{p}^*$.*

(b) *$\partial Z(\mathbf{p}^*) = \mathrm{CH}(\{\mathbf{f}_i,\ i \in E(\mathbf{p}^*)\})$, i.e., a vector $\mathbf{s}$ is a subgradient of the function $Z(\cdot)$ at $\mathbf{p}^*$ if and only if $Z(\mathbf{p}^*)$ is a convex combination of the vectors $\mathbf{f}_i$, $i \in E(\mathbf{p}^*)$.*

The following algorithm generalizes the steepest ascent algorithm and can be used to maximize a nondifferentiable concave function $Z(\cdot)$.

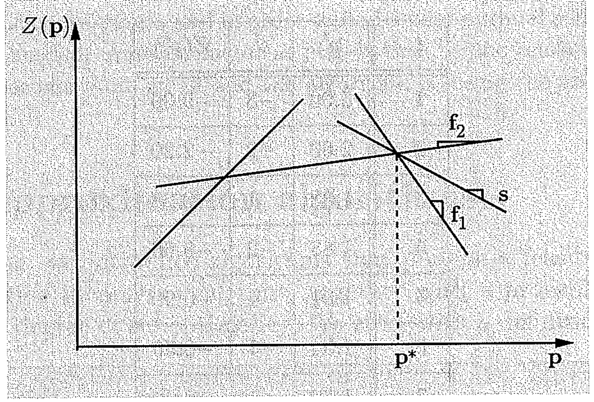

Figure 11.11: The subdifferential of $Z(\mathbf{p})$ at $\mathbf{p}^*$ is the set of all vectors that can be written as convex combinations of $\mathbf{f}_1$ and $\mathbf{f}_2$.

The subgradient optimization algorithm

1. Choose a starting point $\mathbf{p}^1$; let $t = 1$.
2. Given $\mathbf{p}^t$, choose a subgradient $\mathbf{s}^t$ of the function $Z(\cdot)$ at $\mathbf{p}^t$. If $\mathbf{s}^t = \mathbf{0}$, then $\mathbf{p}^t$ is optimal and the algorithm terminates. Else, continue.
3. Let $\mathbf{p}^{t+1} = \mathbf{p}^t + \theta_t \mathbf{s}^t$, where θ_t is a positive stepsize parameter. Increment t and go to Step 2.

We have characterized the subdifferential of a piecewise linear concave function in Lemma 11.4. Typically, however, only the extreme subgradients $\mathbf{f}_i$ are used.

We next specify the stepsize θ_t. It can be proved that $Z(\mathbf{p}^t)$ converges to the unconstrained maximum of $Z(\cdot)$, assuming it is finite, for any stepsize sequence θ_t such that

$$\sum_{t=1}^{\infty} \theta_t = \infty, \qquad \text{and} \qquad \lim_{t\to\infty} \theta_t = 0.$$

An example of such a sequence is $\theta_t = 1/t$. For practical purposes, however, this leads to slow convergence, and other choices for the stepsizes θ_t are often used. An example is

$$\theta_t = \theta_0 \alpha^t, \qquad t = 1, 2, \ldots,$$

where α is a scalar satisfying $0 < \alpha < 1$. A more sophisticated and popular rule is to let

$$\theta_t = \frac{\hat{Z}_D - Z(\mathbf{p}^t)}{\|\mathbf{s}^t\|^2} \alpha^t,$$

t	p^t	s^t	$Z(p^t)$
1	5.00	−3	−9.00
2	2.60	−2	−2.20
3	1.32	1	−0.68
4	1.83	−2	−0.66
5	1.01	1	−0.99
6	1.34	1	−0.66
7	1.60	1	−0.40
8	1.81	−2	−0.62
9	1.48	1	−0.52
10	1.61	1	−0.39

Table 11.1: An example of the subgradient optimization algorithm.

where α satisfies $0 < \alpha < 1$, and $\hat{Z}_D$ is an estimate of the optimal value Z_D. In practice, the stopping criterion $\mathbf{0} \in \partial Z(\mathbf{p}^t)$ is rarely met. Typically, the algorithm is stopped after a fixed number of iterations.

Notice that we are interested in maximizing $Z(\mathbf{p})$ subject to $\mathbf{p} \geq \mathbf{0}$. However, with the algorithm that we have presented, the property $\mathbf{p}^t \geq \mathbf{0}$ is not guaranteed to hold. In order to enforce this condition, we replace Step 3 of the subgradient optimization algorithm by

$$p_j^{t+1} = \max\left\{p_j^t + \theta_t s_j^t, 0\right\}, \qquad \forall\, j.$$

Example 11.11 We apply subgradient optimization to find Z_D in Example 11.8. In this case,

$$Z(p) = \min\left\{3-2p,\ 6-3p,\ 2-p,\ 5-2p,\ -2+p,\ 1,\ 4-p,\ p,\ 3\right\},$$

corresponding to the points in the set

$$X = \left\{(1,0),(2,0),(1,1),(2,1),(0,2),(1,2),(2,2),(1,3),(2,3)\right\}.$$

We let $\theta_t = 0.8^t$. We start with $p^1 = 5$. Then, the minimum in the formula for $Z(p)$ is obtained for the piece $6-3p$ corresponding to $(2,0)$. The new Lagrange multiplier is $p^2 = 5 + 0.8(-3) = 2.6$. The results of the first ten iterations are reported in Table 11.1. The optimal solution is $p^* = 5/3 = 1.66$ and $Z_D = -1/3 = -0.33$. In ten iterations, the best value obtained was in iteration 8, with value -0.39. The example is typical of the behavior of the algorithm. It does not have monotonic convergence, and in order to find a near-optimal solution, several iterations are needed. Another factor at play is that by the tenth iteration, the stepsize has become quite small and the algorithm is losing the ability to make rapid progress.

11.5 Approximation algorithms

In this section, we introduce algorithms that provide a feasible, but suboptimal solution in polynomial time, together with a provable guarantee regarding its degree of suboptimality. We start with a definition.

Definition 11.2 *Algorithm H constitutes an* **ϵ-approximation** *algorithm for a minimization problem with optimal cost Z^*, if for each instance of the problem, algorithm H runs in polynomial time, and returns a feasible solution with cost Z_H, such that*

$$Z_H \leq (1+\epsilon)Z^*. \tag{11.11}$$

Symmetrically, for a maximization problem, we require

$$Z_H \geq (1-\epsilon)Z^*. \tag{11.12}$$

Given an optimization problem, a natural question is whether there exists an ϵ-approximation algorithm for every $\epsilon > 0$. For some problems, this is indeed the case, but there is no general methodology for coming up with such algorithms. An example is provided next.

An ϵ-approximation algorithm for the zero-one knapsack problem

Let us recall the zero-one knapsack problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^n c_j x_j \\ \text{subject to} \quad & \sum_{j=1}^n w_j x_j \leq K \\ & x_j \in \{0,1\}. \end{aligned}$$

We have observed in Section 11.3, that the zero-one knapsack problem can be solved in time $O(n^2 c_{\max})$, and becomes polynomially solvable if $c_{\max} \leq n^d$. This motivates an algorithm in which the coefficients c_i are replaced by smaller values and which produces approximately optimal solutions to the original problem. The key idea is the following. Consider a problem