Integer Programming Duality

J. N. Hooker

MSC2000:90C10, 90C46

Article Outline

Keywords Integer programming; Duality

One of the more elegant and satisfying ideas in the theory of optimization is linear programming duality. The dual of a linear programming problem is not only interesting theoretically but has great practical value, because it provides *sensitivity analysis*, bounds on the optimal value, and marginal values for resources.

It is natural to want to extend duality to integer programming in order to obtain these same benefits. The matter is not so simple, however. Linear programming duality actually represents several concepts of duality that happen to coincide in the case of linear programming but diverge as one moves to other types of optimization problems. The benefits also decouple, because each duality concept provides some of them but not others.

J.N. Hooker
Tepper School of Business, Carnegie Mellon University, Pittsburgh, USA

Five types of integer programming duality are surveyed here. None is clearly superior to the others, and their strengths and weaknesses are summarized in at the end of the article.

Linear Programming Duality

A brief summary of *linear programming duality* will provide a foundation for the rest of the discussion. Consider the linear programming (primal) problem,

$$\begin{cases} \max & cx \\ \text{s.t.} & Ax \leq b \\ & x \geq 0, \end{cases} \tag{1}$$

where A is an $m \times n$ matrix. The dual problem may be stated

$$\begin{cases} \min & ub \\ \text{s.t.} & uA \geq c \\ & u \geq 0, \end{cases} \tag{2}$$

where u is a vector of *dual variables*. This is a *strong dual* because its optimal value is the same as that of the primal problem, unless both primal and dual are infeasible. (An unbounded or infeasible maximization problem is regarded as having optimal value ∞ or $-\infty$, respectively, and analogously for minimization problems.)

The linear programming dual brings at least three important benefits.

(a) (Bounds) The value of any feasible dual solution provides an upper bound on the optimal value of the primal problem. For any x and y that are primal and dual feasible, respectively, $ub \geq uAx \geq cx$. The first inequality is due to the fact that $Ax \leq b$ and $u \geq 0$, and the second is due to the fact that $uA \geq c$ and $x \geq 0$. By finding a dual feasible solution, one can estimate how much a primal feasible solution falls short of optimality. Although this property is less important for linear programming, where robust solution algorithms are available, it is essential for integer programming.

(b) (Sensitivity analysis) Due to (a), the dual solution provides a partial sensitivity analysis. Let $\bar{u}$ be an optimal solution of the dual problem (2), so that $\bar{u}b$ is the optimal value of both primal and dual. If the right-hand side of the constraint in (1) is perturbed by Δb, so that it becomes $Ax \leq b+\Delta b$, then the dual (2) becomes

$$\begin{cases} \min & u(b + \Delta b) \\ \text{s.t.} & uA \geq c \\ & u \geq 0. \end{cases} \tag{3}$$

Because only the objective function changes, $\bar{u}$ is feasible in (3) as well as (2). So $\bar{u}(b + \Delta b)$ is an upper bound on the optimal value of the perturbed primal problem. The (possibly negative) change in the optimal value $\bar{u}b$ of the original problem is bounded above by $\bar{u}\Delta b$. The change is in fact equal to $\bar{u}\Delta b$ if the perturbation Δb lies within easily computable ranges.

(c) (Complementary slackness) Due to (b), the marginal values of resources are readily available. If the right-hand side b_i of a particular constraint of (1) represents a resource constraint, then a change Δb_i in the amount of resource available raises the optimal profit by at most $\bar{u}_i \Delta b_i$. In particular there is a *complementary slackness* property, which says that a surplus resource has no marginal value. More precisely, if $\bar{x}$ is optimal in (1), then $\bar{u}(b - A\bar{x}) = 0$.

Consider for example the linear programming problem

$$\begin{cases} \max & 20x_1 + 10x_2 \\ \text{s.t.} & x_1 + 4x_2 \leq 8 \\ & 2x_1 - 2x_2 \leq 3 \\ & x_1, x_2 \geq 0, \end{cases} \tag{4}$$

which is graphed in Fig. 1. The optimal dual solution is $(u_1, u_2) = (6, 7)$. If the two constraints represent two resource limitations, then the resources have marginal values of at most 6 and 7, respectively. If one less unit of each resource is available (represented by dashed lines in Fig. 1.), then the change in the objective function value is bounded above by $-6 - 7 = -13$. In fact, the profit decreases by exactly 13.

Integer Programming

Integer programming modifies the linear programming problem (1) by requiring the variables to take integral values:

$$\begin{cases} \max & cx \\ \text{s.t.} & Ax \leq b \\ & x \geq 0 \text{ and integer.} \end{cases} \tag{5}$$

In *mixed integer/linear programming* (MILP) some variables are continuous and some are integral. For ease of exposition, the discussion here is restricted to pure (unmixed) integer programming.

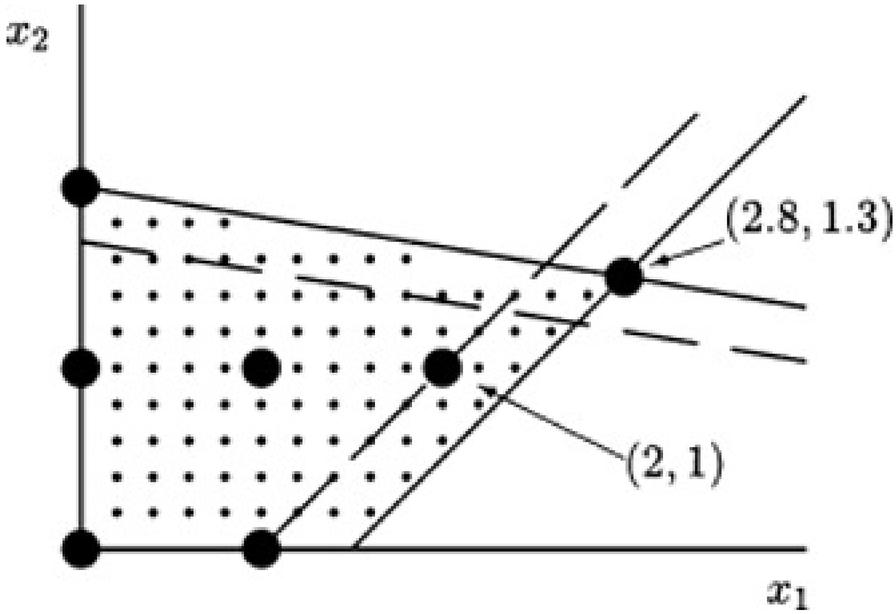

Figure 1 The shaded polyhedron is the feasible set of a linear programming problem with optimal solution $(x_1, x_2) = (2.8, 1.3)$ and optimal value 69. The dashed lines represent a perturbation of the right-hand sides. The black dots represent feasible solutions of the corresponding integer programming problem, which has optimal solution $(x_1, x_2) = (2, 1)$ and optimal value 50.

The example problem (4) may be modified to obtain the integer programming problem,

$$\begin{cases} \max & 20x_1 + 10x_2 \\ \text{s.t.} & x_1 + 4x_2 \leq 8 \\ & 2x_1 - 2x_2 \leq 3 \\ & x_1, x_2 \geq 0 \text{ and integers.} \end{cases} \tag{6}$$

Figure 1 illustrates this problem as well.

In linear programming, a constraint with slack at the optimal solution is redundant. It may be omitted without changing the optimal solution. The example shows that this is untrue for integer programming. Both constraints in (6) contain slack at the solution $(x_1, x_2) = (2, 1)$. Yet removing either would result in a different solution.

The concept of a marginal value is problematic in integer programming. Let the *value function* $v^*(b)$ indicate the optimal value of (1) or (5) for a given right-hand side b. In linear programming the marginal value of resource i is essentially the partial derivative of $v^*(b)$ with respect to b_i. Yet in integer programming $v^*(b)$ is a step function with respect to any b_i, as illustrated by

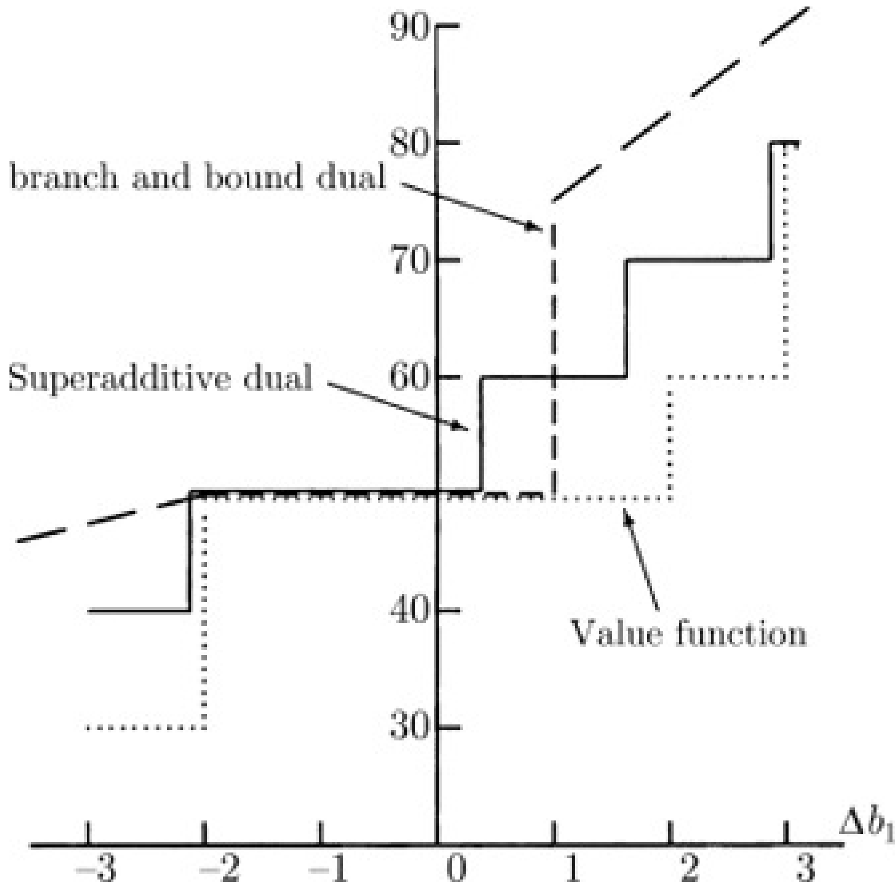

Figure 2 Upper bounds on the optimal value of an integer programming problem provided by the superadditive and branch-and-bound duals, as a function of the right-hand side perturbation Δb_1. The value function indicates the exact optimal value for each Δb_1.

the dotted line in Fig. 2. So it is unclear what would be meant by a marginal value. However, there may be a complementary slackness property of some kind, depending on the duality in question.

Surrogate Duality

One general scheme for formulating integer programming duals is to define a family of relaxations of the original problem that are parameterized by dual variables. The dual problem is then the problem of finding the tightest relaxation. It will be seen that the linear programming dual does exactly this.

One instance of this scheme is *surrogate duality* [8, 9, 10]. The integer programming problem (5) can be relaxed by replacing the constraints $Ax \leq b$ with a *surrogate constraint*, i. e., a nonnegative linear combination of the inequalities in $Ax \leq b$. This yields a *surrogate relaxation* of (1):

$$\begin{cases} \max & cx \\ \text{s.t.} & uAx \leq ub \\ & x \geq 0 \text{ and integer.} \end{cases} \tag{7}$$

This is a relaxation in the sense that its feasible set contains that of (5). Its optimal value $\sigma(u)$ is therefore an upper bound on that of (5) for any $u \geq 0$. The surrogate relaxation may be much easier to solve than the original problem because it has only one constraint (other than nonnegativity). The surrogate dual problem is to find a u that gives the best bound:

$$\begin{cases} \min & \sigma(u) \\ \text{s.t.} & u \geq 0. \end{cases} \tag{8}$$

The surrogate relaxation of a linear programming problem (1) is (7) without the integrality constraint. From strong linear programming duality, its optimal value is $\sigma(u) = \min_\alpha \{\alpha ub : \alpha uA \geq c,\ \alpha \geq 0\}$, where scalar α is the dual variable associated with the first constraint of (7). So the surrogate dual (8) becomes precisely the linear programming dual (2).

Surrogate duality can be illustrated with the integer programming problem (6). Because there are only two constraints and only the ratio u_2/u_1 matters, we can set u_1 to 1 and replace u_2 with u. The surrogate relaxation is therefore

$$\begin{cases} \min & 20x_1 + 10x_2 \\ \text{s.t.} & (1+2u)x_1 + (4-2u)x_2 \leq 8 + 3u \\ & x_1, x_2 \geq 0 \text{ and integers.} \end{cases}$$

A plot of $\sigma(1, u)$ appears in Fig. 3.

The primary utility of the surrogate dual is to provide an upper bound on the optimal value of the original problem. In the example, the dual attains its optimal value of 60 when $1 < u \leq 20/17$. This is better than the bound of 69 provided by the linear programming relaxation. But there is a duality gap of $60 - 50 = 10$.

One might speculate that the surrogate multipliers indicate the relative importance of the two constraints, but it is unclear what this means. One can say, however, that omitting a constraint with a vanishing multiplier does not raise the optimal value above that of the surrogate dual. Vanishing multipliers therefore identify redundant constraints when there is no duality gap.

The surrogate dual (8) must be solved by a search method that does not require gradient or subgradient information. Possible algorithms are discussed

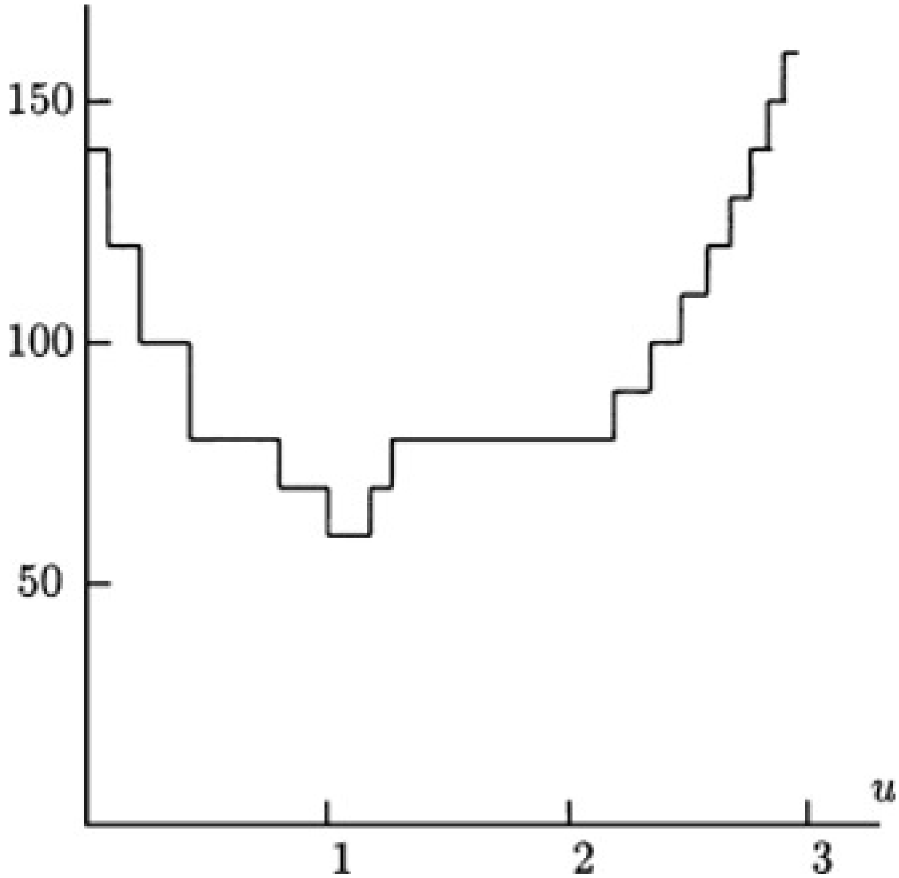

Figure 3 Plot of $\sigma(1, u)$ for a surrogate dual problem

in [18]. The dual problem need not be solved to optimality, because only an upper bound is sought in any case.

Lagrangian Duality

Another form of relaxation duality, *Lagrangian duality* [5**?** , 6], removes some of the more troublesome constraints from (5) but inserts into the objective function a penalty for violating them. Thus the constraints are partitioned into 'hard' constraints $A^1x \leq b^1$ and 'easy' constraints $A^2x \leq b^2$:

$$\begin{cases} \max & cx \\ \text{s.t.} & A^1x \le b^1 \\ & A^2x \le b^2 \\ & x \ge 0 \text{ and integer.} \end{cases} \tag{9}$$

The hard constraints are *dualized* to obtained the *Lagrangian relaxation*:

$$\begin{cases} \max & cx + u(b^1 - A^1x) \\ \text{s.t.} & A^2x \le b^2 \\ & x \ge 0 \text{ and integer.} \end{cases} \tag{10}$$

This is a relaxation in the sense that its optimal value $\theta(u)$ is an upper bound on the optimal value of (5). For any x that is feasible in (9), we have $cx \le cx + u(b^1 - A^1x)$ because $u \ge 0$ and $b^1 - A^1x \ge 0$. The *Lagrangian dual problem* is

$$\begin{cases} \min & \theta(u) \\ \text{s.t.} & u \ge 0. \end{cases} \tag{11}$$

If all the constraints of (9) are dualized, then the Lagrangian dual is no improvement over the linear programming dual. In this case $\theta(u) = \max\{(c - uA)x + ub : x \ge 0,\ x \text{ integer}\}$. So $\theta(u)$ is ub when $c - uA \le 0$ and is ∞ otherwise. The Lagrangian dual problem (2) is now the problem of minimizing ub subject to $c - uA \le 0$ and $u \ge 0$, which is precisely the linear programming dual (2). (It follows that linear programming duality is a special case of Lagrangian duality.)

The Lagrangian dual is therefore useful only when some constraints are not dualized. These constraints must be carefully chosen so that the integer programming problem (10) is easy to solve. It may, for example, decouple into smaller problems or have other special structure.

As an example of Lagrangian duality, suppose that the first constraint of (6) is dualized. The Lagrangian relaxation is

$$\begin{cases} \max & 20x_1 + 10x_2 + u(8 - x_1 - 4x_2) \\ \text{s.t.} & 2x_1 - 2x_2 \le 3 \\ & x_1, x_2 \ge 0 \text{ and integers.} \end{cases}$$

The optimal solution of the Lagrangian dual is $u = 6$, with value $\theta(u) = 62$, slightly worse than the surrogate bound of 60 but still better than the linear programming bound of 69.

The Lagrangian and surrogate duals can be compared in general if the surrogate relaxation dualizes the same constraints as the Lagrangian relaxation. In this case it can be shown that the surrogate duality gap is never larger than the Lagrangian duality gap [6], and it tends to be smaller. The

Lagrangian relaxation has the advantage, however, that it is often easier to solve than the surrogate relaxation that dualizes the same constraints. Moreover, $\theta(u)$ is convex and piecewise linear. A subgradient optimization method can be used to find a global minimum of $\theta(u)$ by finding a local minimum. In fact, if $\theta(u) = cx_u + u(b - Ax_u)$, then $b - Ax_u$ is a subgradient of $\theta(u)$ at u.

When there is no duality gap, the Lagrangian multipliers u_i can be viewed as sensitivities to right-hand side perturbations, with respect to at least one optimal solution. It can be shown that there is no duality gap if and only there exists a feasible solution $\bar{x}$ of (5) and $\bar{u} \geq 0$ that satisfy $\theta(\bar{u}) = c\bar{x}+\bar{u}(b^1-A^1\bar{x})$ and complementary slackness: $\bar{u}(b^1 - A^1\bar{x}) = 0$. However, solution of the Lagrangian dual does not necessarily yield a solution $\bar{x}$ with these properties. Further search may be required.

Superadditive Duality

So far the linear programming dual has been viewed as a relaxation dual, of either the surrogate or Lagrangian type. It can also be viewed as representing the classical duality of vectors and linear functionals. For this purpose the dual of (1) is written:

$$\begin{cases} \min & f(b) \\ \text{s.t.} & f(A) \geq c \\ & f \in F. \end{cases} \tag{12}$$

Here f is a linear functional defined by a nonnegative row vector u, so that $f(b) = ub$ and $f(A) = uA$. The minimization is over the set F of all such functionals.

A similar dual of the integer programming problem (5) can be written as (12), but with minimization over a broader class F of functions. It can be shown that if F is the class of superadditive nondecreasing functions f with $f(0) = 0$, then (12) is a strong integer programming dual. This *superadditive dual* [2, 17, 22, 27] provides sensitivities to right-hand side perturbations. It is also possible, at least in principle, to construct a function f that solves the dual, by means of a cutting plane algorithm.

A *superadditive function* f is one that satisfies $f(a + b) \geq f(a) + f(b)$ for all vectors a, b. The superadditive dual satisfies weak duality because if x is feasible in (5) and f is feasible in (12), then

$$cx \leq \sum_j f(a^j)x_j \leq \sum_j f(a^j x_j) \leq f(Ax) \leq f(b),$$

where a^j is row j of A. The first inequality follows from $f(A) \geq c$. The second is due to superadditivity of f and the fact that multiplication by a nonnegative integer x_j creates a sum of zero or more terms (also $f(0) = 0$). The third is due to superadditivity. The fourth follows from the fact that f is nondecreasing and $Ax \le b$.

Strong duality can be established by exhibiting a dual feasible solution f for which there is no duality gap. Let a *rounding function* be a function of the form

$$R(d) = \lfloor M_k \lfloor M_{k-1} \cdots \lfloor M_1 d \rfloor \cdots \rfloor \rfloor, \tag{13}$$

where each M_i is a nonnegative matrix and $\lfloor \alpha \rfloor$ is α rounded down. A *Chvátal function* has the form uR, where $u \ge 0$ and R is a rounding function. Because Chvátal functions clearly belong to F, it suffices to exhibit a Chvátal function uR for which $uR(b)$ is the optimal value of (5).

This is done by generating Chvátal-Gomory cuts. A rank 1 *Chvátal-Gomory cut* for $Ax \le b$, $x \ge 0$ is an inequality of the form $\lfloor mA \rfloor x \le \lfloor mb \rfloor$, where $m \ge 0$ defines a linear combination of the rows of $Ax \le b$. Rank 2 cuts are obtained by applying the same operation to rank 1 cuts, and so forth. Chvátal showed that the integer hull of any polyhedron (i. e., the convex hull of its integral points) is described by finitely many cuts of finite rank.

This implies that for some rounding function $\bar{R}$, $\bar{R}(A)x \le \bar{R}(b)$ and $x \ge 0$ describe the integer hull of $P = \{x \ge 0 : Ax \le b\}$. So the optimal value of the integer programming problem (5) is the optimal value of

$$\begin{cases} \max & cx \\ \text{s.t.} & \bar{R}(A)x \le \bar{R}(b) \\ & x \ge 0. \end{cases} \tag{14}$$

The linear programming dual of (14) is

$$\begin{cases} \min & u\bar{R}b \\ \text{s.t.} & u\bar{R}(A) \ge c \\ & u \ge 0. \end{cases} \tag{15}$$

If $\bar{u}$ solves problem (15), then its optimal value $\bar{u}\bar{R}(b)$ is the optimal value of (14) and therefore of the original integer programming problem (5). Thus $f = \bar{u}\bar{R}$ is a Chvátal function that solves the dual problem (12).

The dual solution provides sensitivity analysis with respect to right-hand sides. Due to weak duality, $\bar{u}\bar{R}(b + \Delta b)$ is an upper bound on the optimal value when the right-hand side in (5) is perturbed to $b + \Delta b$. There is a form of complementary slackness, because for any optimal solution $\bar{x}$ of (5), $(\bar{u}\bar{R}(A) - c)\bar{x} = 0$.

Consider again the example problem (6). It will be seen below that $\bar{R}$ is

$$\bar{R}(d) = \left\lfloor \begin{pmatrix} \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{pmatrix} \left\lfloor \begin{pmatrix} \frac{4}{5} & \frac{1}{10} \\ \frac{1}{5} & \frac{2}{5} \\ 0 & \frac{1}{2} \end{pmatrix} d \right\rfloor \right\rfloor . \tag{16}$$

Problem (14) is

$$\begin{cases} \max & 20x_1 + 10x_2 \\ \text{s.t.} & x_1 + 2x_2 \le 4 \\ & x_1 - x_2 \le 1 \\ & x_1, x_2 \ge 0. \end{cases}$$

The solution of (15) is $\bar{u} = (10\ \ 10)$. So if the right-hand side of (6) is perturbed by $\Delta b = (\Delta b_1, \Delta b_2)$, the new optimal value of (6) is bounded above by

$$(10\ 10) \left\lfloor \begin{pmatrix} \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{pmatrix} \left\lfloor \begin{pmatrix} \frac{4}{5} & \frac{1}{10} \\ \frac{1}{5} & \frac{2}{5} \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 8 + \Delta b_1 \\ 3 + \Delta b_2 \end{pmatrix} \right\rfloor \right\rfloor .$$

For instance, if each resource is reduced by one ($\Delta b = (-1, -1)$), then the new optimal value is at most 50. In fact it is exactly 50. Figure 2 plots $\bar{u}\bar{R}(b + \Delta b)$ against Δb_1 for comparison with the value function $v^*(b + \Delta b)$. Note that there is complementary slackness, because $(\bar{u}\bar{R}(A) - c)\bar{x} = [(20\ \ 10) - (20\ \ 10)](2, 1) = 0$.

Solving the Superadditive Dual

A solution of the dual problem (12) can be constructed in stages that correspond to Chvátal ranks [3, 15, 26]. It is assumed without practical loss of generality that the components of A, b and c are rational numbers.

The first stage proceeds as follows. Let $x^1, \ldots, x^p$ be the vertices of $P_0 = P$. For each x^k consider the cone C_k of directions d for which x^k maximizes dx subject to $x \in P_0$. To describe C_k, let $\bar{A}x \le \bar{b}$ be the constraints of $Ax \le b$ that are *active* at x^k (i.e., the constraints $a^i x \le b_i$ for which $a^i x^k = b_i$), and let $-\bar{I}x \le 0$ be the active constraints of $-x \le 0$ (the constraints $-x_j \le 0$ for which $x_j^k = 0$). Then C_k is the cone spanned by the rows of $\bar{A}$ and $-\bar{I}$.

It suffices to identify a *Hilbert basis* [7] for C_k; i.e., a set of directions $d^1, \ldots d^q$ such that every integer vector in C_k is a nonnegative integer combination of $d^1, \ldots, d^q$. Assume without loss of generality that the components of $\bar{A}$ are integers (the inequalities $\bar{A}x \le \bar{b}$ can be multiplied by appropriate integers to achieve this). Then the integer vectors $d^1, \ldots, d^q$ in the set

Table 1 Hilbert basis vectors d^j and rank 1 cuts $d^j x \leq \lfloor d^j x^k \rfloor$ corresponding to vertices x^k of an integer programming problem

x^k	d^j	$d^j x \leq \lfloor d^j x^k \rfloor$
$(2.8, 1.3)$	$(1, -1)$	$x_1 - x_2 \leq 1$
	$(1, 0)$	$x_1 \leq 2$
	$(1, 1)$	$x_1 + x_2 \leq 4$
	$(1, 2)$	$x_1 + 2x_2 \leq 5$
	$(1, 3)$	$x_1 + 3x_2 \leq 6$
	$(1, 4)$	$x_1 + 4x_2 \leq 8$
	$(2, 3)$	$2x_1 + 3x_1 \leq 9$
$(1.5, 0)$	$(0, -1)$	$-x_2 \leq 0$
	$(1, -1)$	$x_1 - x_2 \leq 1$
	$(1, -2)$	$x_1 - 2x_2 \leq 1$
$(0, 2)$	$(-1, 0)$	$x_1 \leq 0$
	$(0, 1)$	$x_2 \leq 2$
	$(0, 2)$	$2x_2 \leq 4$
	$(0, 3)$	$3x_2 \leq 6$
	$(1, 3)$	$x_1 + 3x_2 \leq 6$

$$\left\{ \bar{A} - \mu \bar{I} \colon \; 0 \leq \lambda \leq e, \; 0 \leq \mu \leq e \right\}$$

form a Hilbert basis for C_k, where e is a row vector of ones.

The next step is to generate rank 1 Chvátal-Gomory cuts associated with x^k. First note that each inequality of the form $d^j x \leq d^j x^k$ supports P_0 at x^k and is therefore a nonnegative linear combination of the rows of $\bar{A}x \leq \bar{b}$, $-\bar{I}x \leq 0$. Thus one can write

$$d^j = m^j \bar{A} - p^j \bar{I}. \tag{17}$$

The multipliers m^j and p^j can be obtained by solving (17). The valid inequalities $d^j x \leq \lfloor d^j x^k \rfloor$ are clearly rank 1 cuts for $Ax \leq b$, $-x \leq 0$. Rank 1 cuts are generated in this fashion for all the vertices x^k.

Now let P_1 be defined by all of the rank 1 cuts generated, plus $x \geq 0$. Let the rows of M_1 be the vectors m^j corresponding to rank 1 cuts that define facets of P_1. Then $P_1 = \{x \geq 0 : \lfloor M_1 A \rfloor x \leq \lfloor M_1 b \rfloor\}$.

This same procedure is now applied to the vertices of P_1 to obtain M_2 and P_2, and so forth until all the vertices of P_k are integral. At this point (13) is the desired rounding function $\bar{R}$, and $f = \bar{u}\bar{R}$ solves the dual (12).

The Hilbert bases and inequalities $d^j x \leq \lfloor d^j x^k \rfloor$ for problem (6) appear in Table 1. (The origin need not be considered as a vertex.) At vertex $x^1 = (2.8, 1.3)$, for example, the two constraints $x_1 + 4x_2 \leq 8$ and $2x_1 - 2x_2 \leq 3$ are active, and so the cone C_1 is spanned by $(1, 4)$ and $(1, -1)$. The Hilbert basis consists of the integer vectors of the region depicted in Fig. 4.

The polyhedra P_1 and P_2 are shown by dashed and dotted lines, respectively, in Fig. 5. Their facets (other than $x \geq 0$) and the corresponding vectors m^j appear in Table 2. The vectors M_1 and M_2 that appear in the rounding function (16) can be read from Table 2.

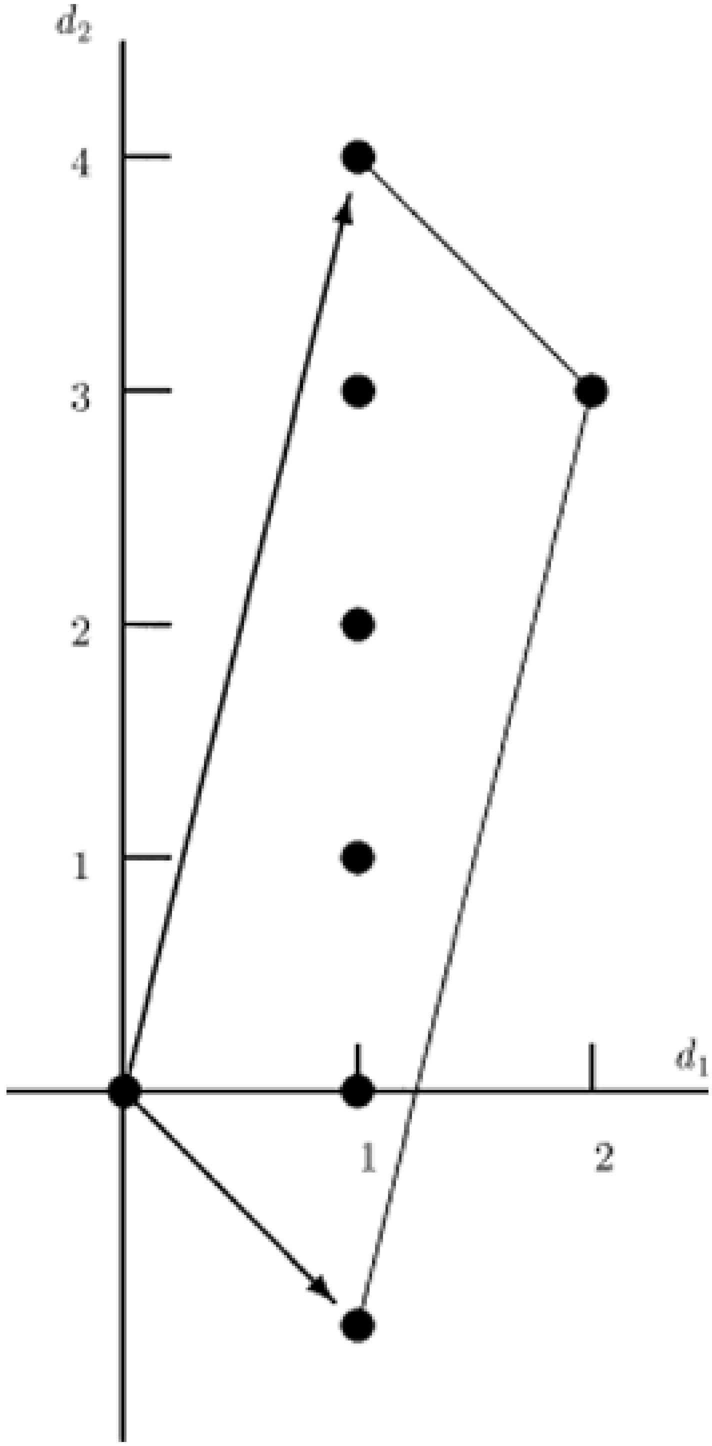

Figure 4 The black dots indicate a Hilbert basis for the cone spanned by $(1,4)$ and $(1,-1)$.

Table 2 Polyhedra P_1, P_2 and vectors m^j corresponding to their facets

P_i	Facet	m^j
P_1	$x_1 + 3x_2 \leq 6$	$(\frac{4}{5}\ \frac{1}{10})$
	$x_1 \leq 2$	$(\frac{1}{5}\ \frac{2}{5})$
	$x_1 - x_2 \leq 1$	$(0\ \frac{1}{2})$
P_2	$x_1 + 2x_2 \leq 4$	$(\frac{2}{3}\ \frac{1}{3}\ 0)$
	$x_1 - x_2 \leq 1$	$(0\ 0\ 1)$

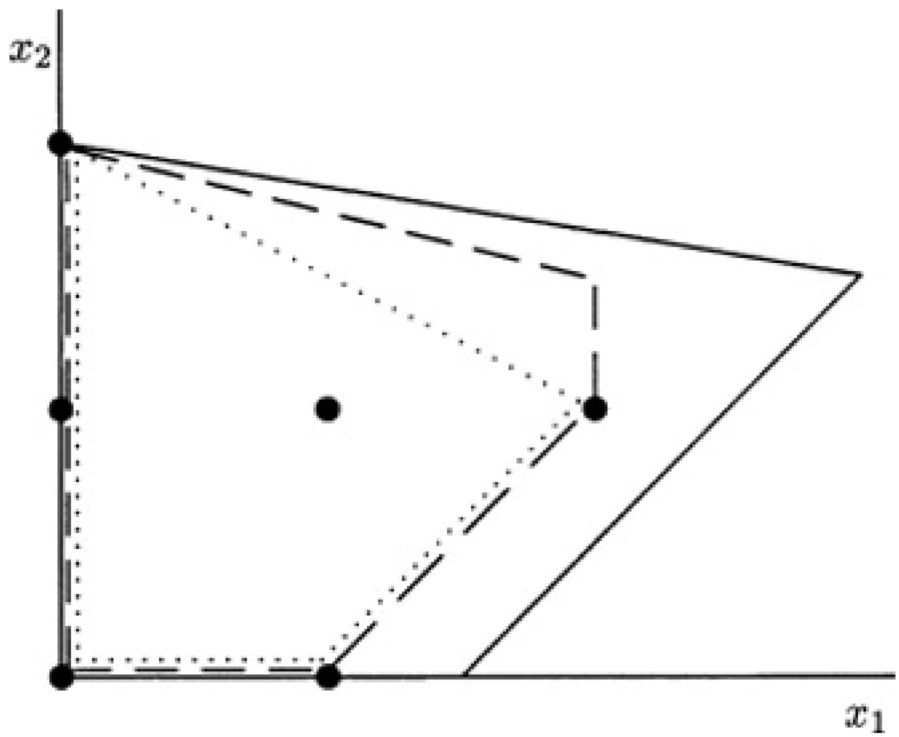

Figure 5 The polyhedra P_0 (solid line), P_1 (dashed line), and P_2 (dotted line) for an integer programming problem

Another Functional Dual

It is practical to solve the superadditive dual only when the problem is small or has special structure. An alternative is to derive a dual solution for (12) from the branch-and-bound tree that solves the primal problem, as proposed in [20] on the basis of work in [27]. This maneuver sacrifices an independently computed upper bound on the optimal value, but it provides useful sensitivity analysis in a more practical fashion than the superadditive dual. It might be called a 'branch-and-bound dual'.

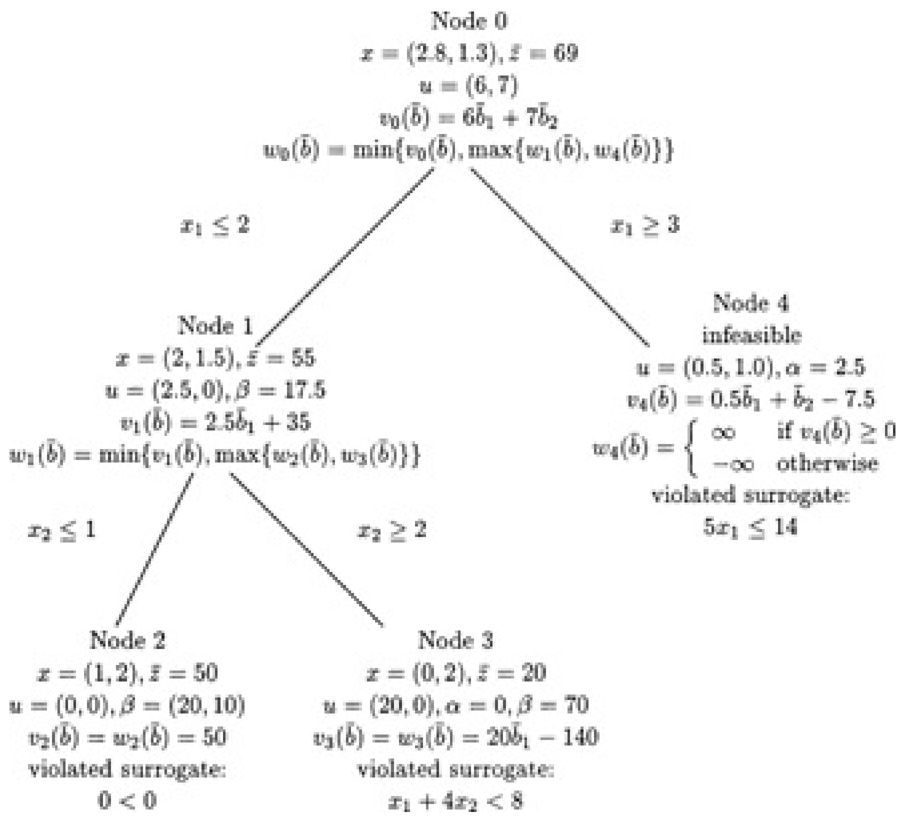

Figure 6 A branch and bound tree with information relevant to the branch-and-bound dual and the inference dual

Rather than superadditive functions, the feasible set F in (12) will contain functions of the form

$$f(d) = \min\{yd + y_0,\ \max\{f_1(d), f_2(d)\}\}, \tag{18}$$

where $y \geq 0$ and f_1 and f_2 are either identically zero or of the form (18). Weak duality is easily shown. Strong duality is shown by constructing a solution as follows.

At each node t of the branch-and-bound tree for (5), one solves the linear relaxation

$$\begin{cases} \max\ cx & \\ \text{s.t.}\quad Ax \leq b & (u) \\ \qquad -x \leq -L^t & (\alpha) \\ \qquad x \leq U^t & (\beta), \end{cases} \tag{19}$$

where the lower and upper bounds L^t, U^t are defined by branching, and associated dual variables are shown on the right. By weak linear programming duality, $v_t(\bar{b}) = u\bar{b} - \alpha L^t + \beta U^t$ is an upper bound on the optimal value of (19) with perturbed right-hand side $d = b + \Delta b$. If (19) is infeasible, let (u, α, β) be the dual solution of the phase I problem in which the objective function is the sum of negative constraint violations. In this case $v_t(\bar{b})$ is $-\infty$ if $u\bar{b} - \alpha + \beta < 0$ and is ∞ otherwise.

Now if t_1, t_2 are the successor nodes of node t in the search tree,

$$w_t(\bar{b}) = \min \left\{ v_t(\bar{b}), \max\{w_{t_1}(\bar{b}), w_{t_2}(\bar{b})\} \right\}$$

is an upper bound on the optimal value of (19) with right-hand side $\bar{b} = b + \Delta b$ and integral x. (At leaf nodes, the max expression is omitted.) The recursively computed function w_0 associated with the root node solves the dual problem (12) because $w_0 b$ is the optimal value of (5).

The dual solution w_0 for the example problem (6) is indicated in the branch-and-bound tree for this problem depicted in Fig. 6. A plot of $w_0(b + \Delta b)$ as a function of Δb_1 appears in Fig. 2.

Inference Duality

Still another interpretation of the linear programming dual views it as an inference problem. It wishes to find the smallest upper bound z^* on the objective function that can be inferred from the constraints. This dual of (1) can be written

$$\begin{cases} \min & z \\ \text{s.t.} & \begin{pmatrix} Ax \le b \\ x \ge 0 \end{pmatrix} \text{ imply } cx \le z. \end{cases} \tag{20}$$

A corollary of the classical Farkas lemma states that the constraints $Ax \le b$, $x \ge 0$ imply $cx \le z$ if and only if they are infeasible or some surrogate $uAx \le ub$ dominates $cx \le z$; i.e., $uA \ge c$ and $ub \le z$. So the inference dual (20) seeks the smallest ub for which $uA \ge c$ and $u \ge 0$ (assuming the constraints are feasible), which is precisely the linear programming dual.

The inference dual can be generalized to integer programming if the implication in (20) is interpreted differently [4, 13, 14]. Constraints $Ax \le b$, $x \ge 0$ imply $cx \le z$ if and only if all *integer* (rather than all real) vectors x that satisfy the former also satisfy the latter. There is obviously no duality gap, because the maximum value z^* of cx is the smallest upper bound on cx implied by the constraints. As will be seen, inference duality allows calculation of sensitivity ranges for all problem data (not just right-hand sides) by solving linear programming problems.

To solve the dual (20) is in effect to exhibit a proof that the value of cx is at most z^*. In linear programming, a proof is a nonnegative linear combination of constraints, and the optimal dual multipliers u encode the desired proof. A method of proof suitable for integer programming is developed in [12], but for present purposes it suffices to reconstruct a proof from the branch-and-bound tree that solves the primal problem. Actually it will be proved that cx is at most $z^* + \Delta z$ (for any $\Delta z \geq 0$), to provide a more flexible analysis.

The proof is by contradiction. Assume, contrary to the claim, that the optimal value of (5) is strictly more than $z^* + \Delta z$. Then each branch of the tree can be seen as leading to a contradiction. At any given leaf node t let z_{LB} be the value of the best integral solution found so far ($z_{\text{LB}} = -\infty$ if none has been found). One of the following cases obtains.

(a) The linear relaxation (19) is infeasible. Then the dual solution (u, α, β) proves infeasibility; i.e., $u^tA - \alpha + \beta \geq 0$ and $u^tb - \alpha^tL^t + \beta^tU^t < 0$. So the constraints $u^tAx \leq u^tb$, $L^t \leq x \leq U^t$, $x \geq 0$ are also infeasible. In other words, the bounds $L^t \leq x \leq U^t$ are inconsistent with the surrogate $u^tAx \leq u^tb$.

(b) The solution of (19) is integral with value $\bar{z}_t$, where $\bar{z}_t > z_{\text{LB}}$. So the constraints

$$\begin{cases} -cx < -\bar{z}_t - \Delta z \\ Ax \leq b \\ -x \leq -L^t \\ x \leq U^t \end{cases} \tag{21}$$

are infeasible. If (u^t, α^t, β^t) is the dual solution of (19), the multipliers $(1, u^t, \alpha^t, \beta^t)$ prove infeasibility of (21). This means that the bounds $L^t \leq x \leq U^t$ are inconsistent with the surrogate $(u^tA - c)x < u^tb - \bar{z}_t - \Delta z$.

(c) The optimal value $\bar{z}$ of (19) satisfies $\bar{z} \leq z_t^{\text{LB}}$, where z_t^{LB} is the current lower bound (the tree is pruned at this node). Here the bounds $L^t \leq x \leq U^t$ are inconsistent with the surrogate $(u^tA - c)x < u^tb - z_t^{\text{LB}} - \Delta z$.

Thus there is a contradiction at every leaf node, because the bounds $L^t \leq x \leq U^t$ are inconsistent with some surrogate at every leaf node.

The key to sensitivity analysis is that a contradiction remains at every leaf node, and the proof remains valid, so long as the bounds remain inconsistent with the surrogates after perturbation of the data. To analyze how much perturbation is possible, the following observation is helpful. The bounds $L \leq x \leq U$ are inconsistent with inequality $dx \leq \delta$ if and only if there exists a vector $\bar{d} \geq 0$ such that

$$\begin{cases} dL - \bar{d}(U - L) > \delta \\ \bar{d} \geq d, \quad \bar{d} \geq 0. \end{cases} \tag{22}$$

Now let (5) be perturbed as follows:

Table 3 Properties of five integer programming duals

Type of dual	Strong duality?	Bounds on objective fcn?	Sensitivity analysis?	Complementary slackness?
Surrogate	No	Yes	Very limited, even if no duality gap	No
Lagrangian	No	Yes	For RHS, if no duality gap	Yes, if no duality gap
Superadditive	Yes	Not practical	For RHS only	Yes
Branch & bound	Yes	No	For RHS only	No
Inference	Yes	No	Bounds for all problem data	No

$$\begin{cases} \max & (c + \Delta c)x \\ \text{s.t.} & (A + \Delta A)x \leq b + \Delta b \\ & x \geq 0 \text{ and integer.} \end{cases} \tag{23}$$

Thus the violated surrogate in case (a) becomes $u^t(A + \Delta A)x \leq u^t(b + \Delta b)$, and similarly in cases b) and c). Using (22), the optimal value of (23) rises no more than Δz ($\Delta z \geq 0$) if the perturbation satisfies the following for some $\bar{q}^t \geq 0$ at every leaf node t,

$$\begin{cases} (q^t + \Delta q^t)L^t - \bar{q}^t(U^t - L^t) \\ \qquad \geq u^t(A + \Delta A) + z_t \\ \bar{q}^t \geq q^t + \Delta q^t, \quad \bar{q}^t \geq 0. \end{cases} \tag{24}$$

Here

$$\begin{aligned} q^t &= u^t A - u_0^t c, \\ \Delta q^t &= u^t \Delta A - u_0^t \Delta c, \\ (u_0^t, z_t) &= \begin{cases} (0, \epsilon) & \text{in case (a),} \\ (1, \bar{z}_t + \Delta z) & \text{in case (b),} \\ (1, z_t^{LB} + \Delta z) & \text{in case (c).} \end{cases} \end{aligned}$$

This can be checked by linear programming. Note that the perturbations $\Delta A, \Delta b, \Delta c$ are not restricted to be nonnegative. Ranges for any perturbation can be computed by minimizing and maximizing it subject to (24) with all other perturbations set to zero.

The dual solutions in Fig. 6 suffice to generate the inequalities (24) for the example problem (6). Leaf nodes 2, 3 and 4 respectively illustrate cases (b), (c) and (a). For instance, the inequalities for leaf node $t = 2$ are

$$\begin{aligned} &-2\Delta c_1 - \Delta c_2 - 2\bar{q}_1^2 - \bar{q}_2^2 \geq 0, \\ &\bar{q}_1^2 \geq -20 - \Delta c_1, \quad \bar{q}_1^2 \geq 0, \\ &\bar{q}_2^2 \geq -10 - \Delta c_2, \quad \bar{q}_2^2 \geq 0. \end{aligned}$$

At leaf nodes 3 and 4 one must assume some large but finite upper bound on variables x_j for which U_j is otherwise infinite. The resulting sensitivity range for b_1 is given below, along with the ranges yielded by the superadditive dual, the branch-and-bound dual, and the true value function (the last three from Fig. 2).

- inference dual: $-\infty < \Delta b_1 < 1$;
- superadditive dual: $-\infty < \Delta b_1 < 0.375$;
- branch-and-bound dual: $-\infty < \Delta b_1 < 1$;
- maximum range: $-\infty < \Delta b_1 < 2$.

No perturbation within the maximum range causes the optimal value to rise above 50. The various forms of sensitivity analysis generally provide more conservative ranges (the same is true of classical linear programming). This example shows that the superadditive dual, although the hardest to compute, does not necessarily provide the sharpest analysis. The inference dual, unlike the others, provides ranges for all problem data:

$$\begin{pmatrix} -\infty \\ -\infty \end{pmatrix} < \Delta b < \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix},$$
$$\begin{pmatrix} -\frac{1}{3}+\epsilon & -\frac{1}{3}+\epsilon \\ 0 & 0 \end{pmatrix} \leq \Delta A < \begin{pmatrix} \infty & \infty \\ \infty & \infty \end{pmatrix},$$
$$(0 \quad 0) \leq \Delta c < (\infty \quad \infty).$$

By setting Δz to 10 rather than zero in (24), one obtains ranges within which perturbations do not increase the optimal value more than 10, and so forth.

Like branch-and-bound duality, inference duality is computationally impractical if the branch-and-bound tree is too large, although it requires fewer data from the tree. It does not provide an explicit approximation of the value function as superadditive and branch-and-bound duality do. However, only inference duality provides easily computable sensitivity ranges, not only for right-hand sides but for all problem data.

Conclusions

Table 3 summarizes the properties of the various duals. The surrogate and Lagrangian duals are used primarily for computational purposes, because they provide independent bounds on the optimal value. The remaining duals are useful for sensitivity analysis. The superadditive and branch-and-bound duals provide a more complete analysis of right-hand side sensitivity. The latter requires a branch-and-bound solution of the problem but considerably less computation. Inference duality requires a branch-and-bound solution and provides only sensitivity ranges, but ranges can be obtained for all problem data by solving linear programming problems.

One can also formulate a dual based on congruence relations, as described in [23, 25]. It is similar to the superadditive dual described above, except that the nested roundings in the value function are roundings down to the nearest multiple of an appropriate modulus rather than to the nearest integer. The matrices M_i are computed by a projection operation that based on a generalized Chinese remainder theorem, rather than by the generation of Chvátal-Gomory cuts. One feature of this dual is that the nesting depth of roundings in the value function is bounded above by the number of integer variables, while there is no upper bound in the superadditive dual. This dual is also defined for a generalization of integer programming that is solved over a subset of the integer lattice defined by a given set of congruence relations, rather than over the entire integer lattice. Unlike conventional integer programming, this larger problem class is closed under projection. A comparison of this and other duals (surrogate, Lagrangian, superadditive) appears in [24].

General treatments of Lagrangian and superadditive duality may be found in [19, 21], with a brief discussion of surrogate duality in the former. Classic discussions of the value function appear in [1, 2, 3], and a more recent survey of value functions in [11].

See also

Decomposition Techniques for MILP: Lagrangian Relaxation
Integer Programming
Integer Programming: Algebraic Methods
Integer Programming: Branch and Bound Methods
Integer Programming: Branch and Cut Algorithms
Integer Programming: Cutting Plane Algorithms
Integer Programming: Lagrangian Relaxation
Simplicial Pivoting Algorithms for Integer Programming
Time-dependent Traveling Salesman Problem

References

1 Blair CE, Jeroslow R (1977) The value function of a mixed integer program I. Discret Math 19:121–138
2 Blair CE, Jeroslow R (1979) The value function of a mixed integer program II. Discret Math 25:7–19
3 Blair CE, Jeroslow R (1982) The value function of a mixed integer program. Math Program 23:237–273
4 Dawande M, Hooker JN (2000) Inference-based sensitivity analysis for mixed integer linear programming. Oper Res 48:623-634

5 Everett III H (1963) Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. Oper Res 11:399–417
6 Geoffrion AM (1974) Lagrangean relaxation for integer programming. Math Program Stud 2:82–114
7 Giles FR, Pulleyblank WR (1979) Total dual integrality and integer polyhedra. Linear Alg Appl 25:191–196
8 Glover F (1968) Surrogate constraints. Oper Res 16:741–749
9 Glover F (1975) Surrogate constraint duality in mathematical programming. Oper Res 23:434–451
10 Greenberg H, Pierskalla WP (1970) Surrogate mathematical programming. Oper Res 18:924–939
11 Guzelsoy M, Ralphs TK (2010) Integer programming duality. Tech rep, Lehigh University
12 Hooker JN (1992) Generalized resolution for 0–1 linear inequalities. Ann Math Artificial Intelligence 6:271–286
13 Hooker JN (1996) Inference duality as a basis for sensitivity analysis. In: Freuder EC (ed) Principles and Practice of Constraint Programming-CP96. no. 1118 of Lecture Notes Computer Sci. Springer, Berlin, pp 224–236
14 Hooker JN (2012) Integrated Methods for Optimization, 2nd ed. Springer, Berlin
15 Jeroslow RG (1978) Cutting plane theory: Algebraic methods. Discret Math 23:121–150
16 Blair CE, Jeroslow RG (1982) The value function of an integer program. Math Prog 23:237–273
17 Johnson EL (1973) Cyclic groups, cutting planes and shortest paths. In: Hu TC, Robinson S (eds) Mathematical Programming. Acad. Press, New York, pp 185–211
18 Karwan MH, Rardin RL (1979) Some relationships between Lagrangean & surrogate duality in integer programming. Math Program 17:320–334
19 Nemhauser GL, Wolsey LA (1988) Integer and combinatorial optimization. Wiley, New York
20 Schrage L, Wolsey LA (1985) Sensitivity analysis for branch and bound integer programming. Oper Res 33:1008–1023
21 Schrijver A (1986) Theory of linear and integer programming. Wiley, New York
22 Tind J, Wolsey LA (1981) An elementary survey of general duality theory in mathematical programming. Math Program 21
23 Williams HP (1984) A duality theorem for linear congruences. Discrete Appl Math 7:93–103
24 Williams HP (1996) Duality in mathematics and linear and integer programming. J Optim Th Appl 90:257–278
25 Williams HP, Hooker JN (2016) Integer programming as projection. Discret Optim 22:291-311

26 Wolsey LA (1981) The b-hull of an integer program. Discrete Appl Math 3:193–201
27 Wolsey LA (1981) Integer programming duality: Price functions and sensitivity analysis. Math Program 20:173–195