

Duality in Optimization and Constraint Satisfaction

J. N. Hooker

Carnegie Mellon University, Pittsburgh, USA
john@hooker.tepper.cmu.edu

Abstract. We show that various duals that occur in optimization and constraint satisfaction can be classified as inference duals, relaxation duals, or both. We discuss linear programming, surrogate, Lagrangean, superadditive, and constraint duals, as well as duals defined by resolution and filtering algorithms. Inference duals give rise to nogood-based search methods and sensitivity analysis, while relaxation duals provide bounds. This analysis shows that duals may be more closely related than they appear, as are surrogate and Lagrangean duals. It also reveals common structure between solution methods, such as Benders decomposition and Davis-Putnam-Loveland methods with clause learning. It provides a framework for devising new duals and solution methods, such as generalizations of mini-bucket elimination.

1 Two Kinds of Duals

Duality is perennial theme in optimization and constraint satisfaction. Well-known optimization duals include the linear programming (LP), Lagrangean, surrogate, and superadditive duals. The constraint satisfaction literature discusses constraint duals as well as search methods that are closely related to duality.

These many duals can be viewed as falling into two classes: inference duals and relaxation duals [12]. The two classes represent quite different concepts of duality. This is perhaps not obvious at first because the traditional optimization duals just mentioned can be interpreted as both inference and relaxation duals.

Classifying duals as inference or relaxation duals reveals relationships that might not otherwise be noticed. For instance, the surrogate and Lagrangean duals do not seem closely related, but by viewing them as inference duals rather than relaxation duals, one sees that they are identical except for a slight alteration in the type of inference on which they are based.

A general analysis of duality can also unify some existing solution methods and suggest new ones. Inference duals underlie a number of nogood-based search methods and techniques for sensitivity analysis. For instance, Benders decomposition and Davis-Putnam-Loveland methods with clause learning, which appear unrelated, are nogood-based search methods that result from two particular inference duals. Since any inference method defines an inference dual, one can

in principle devise a great variety inference duals and investigate the nogood-based search methods that result. For example, filtering algorithms can be seen as inference methods that define duals and give rise to new search methods, such as decomposition methods for planning and scheduling.

Relaxation duals underlie a variety of solution methods that are based on bounding the objective function. A relaxation dual solves a class of problem relaxations that are parameterized by “dual variables,” in order to obtain a tight bound on the objective function value. The LP, surrogate, Lagrangean, and superadditive duals familiar to the optimization literature are relaxation duals. A constraint dual is not precisely a relaxation dual but immediately gives rise to one that generalizes mini-bucket elimination methods.

Inference and relaxation duals are precise expressions of two general problem-solving strategies. Problems are often solved by a combination of search and inference; that is, by searching over values of variables, which can yield a certificate of feasibility for the original (“primal”) problem, and by simultaneously drawing inferences from constraints, which can yield a certificate of optimality by solving the dual problem. A problem belongs to NP when the primal solution has polynomial size and to co-NP when the dual solution has polynomial size.

Problems can also be solved by a combination of search and relaxation; that is, by enumerating relaxations and solving each. The relaxation dual is one way of doing this, since it searches over values of dual variables and solves the relaxation corresponding to each value.

2 Inference Duality

An *optimization problem* can be written

$$\min_{x \in D} \{f(x) \mid \mathcal{C}\} \tag{1}$$

where $f(x)$ is a real-valued function, \mathcal{C} is a constraint set containing variables $x = (x_1, \dots, x_n)$, and D is the *domain* of x . A solution $\bar{x} \in D$ is *feasible* when it satisfies \mathcal{C} and is *optimal* when $f(\bar{x}) \leq f(x)$ for all feasible x . If there is no feasible solution, the optimal value of (1) is ∞ . If there is no lower bound on $f(x)$ for feasible x , the problem is *unbounded* and the optimal value is $-\infty$.¹ A value \bar{z} is a feasible value of (1) if $f(x) = \bar{z}$ for some feasible x , or if $\bar{z} = \infty$, or if $\bar{z} = -\infty$ and the problem is unbounded.

A *constraint satisfaction problem* can be viewed the problem of determining whether the optimal value of

$$\min_{x \in D} \{0 \mid \mathcal{C}\}$$

is 0 or ∞ .

¹ We exclude problems that have no optimal value, such as $\min_{x \in \mathbb{R}} \{x \mid x > 0\}$.

2.1 The Inference Dual

The *inference dual* of (1) is the problem of finding the greatest lower bound on $f(x)$ that can be inferred from \mathcal{C} within a given proof system. The inference dual can be written

$$\max_{P \in \mathcal{P}} \left\{ v \mid \mathcal{C} \stackrel{P}{\vdash} (f(x) \geq v) \right\} \quad (2)$$

where $\mathcal{C} \stackrel{P}{\vdash} (f(x) \geq v)$ indicates that proof P deduces $f(x) \geq v$ from \mathcal{C} . The domain of variable P is a family \mathcal{P} of proofs. A pair (\bar{v}, \bar{P}) is a feasible solution of (2) if $P \in \mathcal{P}$ and $\mathcal{C} \stackrel{\bar{P}}{\vdash} (f(x) \geq \bar{v})$, and it is optimal if $\bar{v} \geq v$ for all feasible (v, P) . If $f(x) \geq v$ cannot be derived from \mathcal{C} for any finite v , the problem is infeasible and has optimal value $-\infty$. If for any v there is a feasible (v, P) , the problem is unbounded and has optimal value ∞ . A value \bar{v} is a feasible value of (2) if (\bar{v}, P) is feasible for some $P \in \mathcal{P}$, or if $\bar{v} = -\infty$, or if $\bar{v} = \infty$ and (2) is unbounded.

The original problem (1) is often called the *primal* problem. Any feasible value of the dual problem is clearly a lower bound on any feasible value of the primal problem, a property known as *weak duality*. The difference between the optimal value of the primal and the optimal value of the dual is the *duality gap*.

The constraint set \mathcal{C} *implies* $f(x) \geq v$ when $f(x) \geq v$ for all $x \in D$ satisfying \mathcal{C} . The proof family \mathcal{P} is *complete* if for any v such that \mathcal{C} implies $f(x) \geq v$, there is a proof $P \in \mathcal{P}$ that deduces $f(x) \geq v$ from \mathcal{C} . If \mathcal{P} is complete, then there is no duality gap. This property is known as *strong duality*.

Solution of the inference dual for a complete proof family \mathcal{P} solves the optimization problem (1), in the sense that a solution (\bar{v}, \bar{P}) of the dual provides a proof that \bar{v} is the optimal value of (1). If \bar{P} always has polynomial size, then the dual belongs to NP and the primal problem belongs to co-NP. Solution of the inference dual for an incomplete proof family may not solve (1) but may be useful nonetheless, for instance by providing nogoods and sensitivity analysis.

2.2 Nogood-based Search

Nogoods are often used to exclude portions of the search space that have already been explicitly or implicitly examined. The inference dual can provide a basis for a nogood-based search.

Suppose we are solving problem (1) by searching over values of x in some manner. The search might proceed by splitting domains, fixing variables, or by adding constraints of some other sort. Let \mathcal{B} be the set of constraints that have been added so far in the search. The constraint set has thus been enlarged to $\mathcal{C} \cup \mathcal{B}$. The inference dual of this restricted problem is

$$\max_{P \in \mathcal{P}} \left\{ v \mid \mathcal{C} \cup \mathcal{B} \stackrel{P}{\vdash} (f(x) \geq v) \right\} \quad (3)$$

If (\bar{v}, \bar{P}) solves the dual, we identify a subset \mathcal{N} of constraints that include all the constraints actually used as premises in the proof \bar{P} . That is, \bar{P} remains a

valid proof when $\mathcal{C} \cup \mathcal{B}$ is replaced by \mathcal{N} . Then by weak duality we can infer the *nogood*

$$\mathcal{N} \rightarrow (f(x) \geq \bar{v})$$

This nogood is a valid constraint and can be added to \mathcal{C} , which may accelerate the search. For instance, if \mathcal{N} contains only a few variables, then restricting or fixing only a few variables may violate the nogood, allowing us to avoid a dead end earlier in the search process.

An important special case of this idea identifies a subset $\bar{\mathcal{B}} \subset \mathcal{B}$ of the search constraints that preserves the validity of \bar{P} . That is, \bar{P} remains a proof when $\mathcal{C} \cup \bar{\mathcal{B}}$ replaces $\mathcal{C} \cup \mathcal{B}$. Then we can use the nogood

$$\bar{\mathcal{B}} \rightarrow (f(x) \geq \bar{v})$$

as a side constraint that guides the search, rather than adding it to \mathcal{C} . Suppose for example that the search proceeds by splitting domains; that is, by adding bounds of the form $L_j \leq x_j \leq U_j$ to \mathcal{B} . Suppose further that at some point in the search we obtain a solution (\bar{v}, \bar{P}) of the inference dual and find that the only bounds used as premises in \bar{P} are $L_j \leq x_j$ and $x_k \leq U_k$. Then we can write the nogood

$$(L_j \leq x_j, x_k \leq U_k) \rightarrow (f(x) \geq \bar{v})$$

To obtain a solution value better than \bar{v} , we must avoid all future branches in which $x_j < L_j$ and $x_k > U_k$.

We can equally well apply this technique when we branch by fixing a variable x_j to each of the values in its domain. Suppose that at some point in the search the variables in x_F have been fixed to values \bar{x}_F , and the variables in x_U remain unfixed, where $x = (x_F, x_U)$. Thus $\mathcal{B} = \{x_F = \bar{x}_F\}$. We obtain a solution (\bar{v}, \bar{P}) of the inference dual and identify a subset x_J of variables in x_F such that \bar{P} is still valid when $x_F = \bar{x}_F$ is replaced by $x_J = \bar{x}_J$. The resulting nogood

$$(x_J = \bar{x}_J) \rightarrow (f(x) \geq \bar{v})$$

tells us that if we want a solution value better than \bar{v} , the remainder of the search should exclude solutions x in which $x_J = \bar{x}_J$.

2.3 Sensitivity Analysis

Sensitivity analysis determines the sensitivity of the optimal value of (1) to perturbations in the problem data. Suppose that we have solved (1) and found its optimal value to be z^* . A simple form of sensitivity analysis relies on an optimal solution (\bar{v}, \bar{P}) of the inference dual [11]. Let $\bar{\mathcal{C}}$ be a subset of \mathcal{C} for which \bar{P} remains a valid proof of $f(x) \geq \bar{v}$. Then changing or removing the premises in $\mathcal{C} \setminus \bar{\mathcal{C}}$ has no effect on the bound and therefore cannot reduce the optimal value of (1) below \bar{v} . If there is no duality gap, then $z^* = \bar{v}$, and changing or removing these constraints has no effect on the optimal value of (1).

A sharper analysis can often be obtained by observing how much the individual constraints in \mathcal{C} can be altered without invalidating the proof \bar{P} . One can

also observe whether a proof having the same form as \bar{P} would deduce $f(x) \geq v'$ for some $v' < v$ when the constraints in \mathcal{C} are altered in certain ways. Both of these strategies have long been used in linear programming, for example. They can be applied to integer and mixed integer programming as well [6].

From here out we focus on nogood-based search rather than sensitivity analysis.

2.4 Linear Programming Dual

A linear programming (LP) problem has the form

$$\min_{x \geq 0} \{cx \mid Ax \geq b\} \quad (4)$$

The inference dual is

$$\max_{P \in \mathcal{P}} \left\{ v \mid Ax \geq b \stackrel{P}{\vdash} (cx \geq v) \right\} \quad (5)$$

The proofs in family \mathcal{P} are based on nonnegative linear combination and domination. Let a *surrogate* of a system $Ax \geq b$ be any linear combination $uAx \geq ub$, where $u \geq 0$. An inequality $ax \geq \alpha$ *dominates* $bx \geq \beta$ when $a \leq b$ and $\alpha \geq \beta$. There is a proof $P \in \mathcal{P}$ of $f(x) \geq v$ when some surrogate of $Ax \geq b$ dominates $f(x) \geq v$. The proof P is encoded by the vector u of *dual multipliers*. Due to the classical Farkas lemma, the proof family \mathcal{P} is complete, which means that strong duality holds.

The inference dual (5) of (4) is essentially the same as the classical LP dual of (4). A solution (v, P) is feasible in the dual problem (5) when some surrogate $uAx \geq ub$ dominates $cx \geq v$, which is to say $uAx \leq c$ and $ub \geq v$. So when the dual is bounded (i.e., the primal is feasible), it can be seen as maximizing v subject to $uAx \leq c$, $ub \geq v$, and $u \geq 0$, or equivalently

$$\max_{u \geq 0} \{ub \mid uA \leq c\} \quad (6)$$

which is the classical LP dual. Strong duality holds for the classical dual unless both the primal and dual are infeasible.

When the LP dual is used in nogood-based search, the well-known method of *Benders decomposition* results [2]. It is applied to problems that become linear when certain variables x_F are fixed:

$$\min_{x_U \geq 0} \{f(x_F) + cx_U \mid g(x_F) + Ax_U \geq b\} \quad (7)$$

Suppose that when x_F is fixed to \bar{x}_F , (7) has optimal value \bar{z} and optimal dual solution \bar{u} . By strong duality $\bar{z} - f(\bar{x}_F) = \bar{u}(b - g(\bar{x}_F))$, which means that

$$f(x_F) + \bar{u}(b - g(x_F)) \quad (8)$$

is the largest possible lower bound on the optimal value of (7) when $x_F = \bar{x}_F$. But since \bar{u} remains dual feasible when \bar{x}_F in (7) is replaced by any x_F , weak

duality implies that (8) remains a valid lower bound for any x_F . This yields the nogood

$$z \geq f(x_F) + \bar{u}(b - g(x_F)) \quad (9)$$

where z represents the objective function of (7). This nogood is known as a *Benders cut*. If the dual of (7) is unbounded, there is a direction or ray \bar{u} along which its solution value can increase indefinitely. In this case the Benders cut (9) simplifies to $\bar{u}(b - g(x_F)) \leq 0$.

In the Benders algorithm, the set x_F of fixed variables is static. The algorithm searches over values of x_F by solving a *master problem* in each iteration of the search. The master problem minimizes z subject to the Benders cuts obtained in previous iterations. The optimal solution of the master problem becomes the next \bar{x}_F . The search terminates when the optimal value of the master problem is equal to the previous \bar{z} . The master problem can be solved by any desired method, such as branch and bound if it is mixed integer programming problem.

2.5 Surrogate Dual

The surrogate dual results when one writes the inference dual of an inequality-constrained problem, again using nonnegative linear combination and domination as an inference method. When the inequalities and objective function are linear, the surrogate dual becomes the linear programming dual. When a slightly stronger form of domination is used, we obtain the Lagrangean dual, as is shown in the next section.

The surrogate dual [10] is defined for a problem of the form

$$\min_{x \in D} \{f(x) \mid g(x) \leq 0\} \quad (10)$$

where $g(x)$ is a vector of functions. A surrogate of $g(x) \leq 0$ is any linear combination $ug(x) \leq 0$ with $u \geq 0$. Let $P \in \mathcal{P}$ deduce $f(x) \geq v$ from $g(x) \leq 0$ when some surrogate $ug(x) \leq 0$ dominates $f(x) \geq v$. We will use the weakest possible form of domination: $ug(x) \leq 0$ dominates $f(x) \geq v$ whenever the former implies the latter. This family \mathcal{P} of proofs is generally incomplete.

Under this definition of \mathcal{P} , the inference dual of (10) finds the largest v such that $ug(x) \leq 0$ implies $f(x) \geq v$ for some $u \geq 0$. The inference dual therefore becomes the *surrogate dual*

$$\max_{u \geq 0} \left\{ \min_{x \in D} \{f(x) \mid ug(x) \leq 0\} \right\} \quad (11)$$

A difficulty with the surrogate dual is that it is generally hard to solve. Yet if the problem (10) has special structure that allows easy solution of the dual, the resulting nogoods could be used in a search algorithm.

2.6 Lagrangean Dual

Like the surrogate dual, the Lagrangean dual is defined for inequality-constrained problems of the form (10). Again the proofs in \mathcal{P} consist of nonnegative linear

combination and domination, but this time a stronger form of domination is used.

In the surrogate dual, $ug(x) \leq 0$ dominates $f(x) \geq v$, which can be written $v - f(x) \leq 0$, when $ug(x) \leq 0$ implies $v - f(x) \leq 0$. In the Lagrangean dual, $ug(x) \leq 0$ dominates $v - f(x) \leq 0$ when $ug(x) \geq v - f(x)$ for all $x \in D$.

Under this definition of \mathcal{P} , the inference dual of (10) finds the largest v such that $ug(x) \geq v - f(x)$ for some $u \geq 0$. Since $ug(x) \geq v - f(x)$ can be written $f(x) + ug(x) \geq v$, the inference dual becomes the *Lagrangean dual*

$$\max_{u \geq 0} \left\{ \min_{x \in D} \{f(x) + ug(x)\} \right\} \quad (12)$$

The Lagrangean dual has the nice property that the optimal value $\theta(u)$ of the minimization problem in (12) is a concave function of u . This means that $\theta(u)$ can be maximized by a hill-climbing search. Subgradient optimization techniques are often used for this purpose [1, 19].

When the inequalities $g(x) \leq 0$ and the objective function $f(x)$ are linear, both the surrogate and Lagrangean duals become the linear programming dual, since the two types of domination collapse into implication.

Nogoods can be obtained from the Lagrangean dual much as from the LP dual. If at some point in the search x_F is fixed to \bar{x}_F and \bar{u} solves the Lagrangean dual problem

$$\max_{u \geq 0} \left\{ \min_{x \in D} \{f(\bar{x}_F, x_U) + ug(\bar{x}_F, x_U)\} \right\} \quad (13)$$

then we have the nogood

$$z \geq f(\bar{x}_F, x_U) + \bar{u}g(\bar{x}_F, x_U) \quad (14)$$

Nogoods of this sort could be generated in a wide variety of search methods, but they have apparently been used only in the special case of *generalized Benders decomposition* [9]. This method can be applied when there is a set x_F of variables for which (13) has no duality gap when x_F is fixed. The set x_F is therefore static, and x_F is fixed by solving a master problem that contains the Benders cuts (14). In practice the classical method obtains the multipliers \bar{u} by solving

$$\min_{x \in D} \{f(\bar{x}_F, x_U) \mid g(\bar{x}_F, x_U) \leq 0\}$$

as a nonlinear programming problem and letting \bar{u} be the Lagrange multipliers that correspond to the optimal solution. However, the multipliers could be obtained by solving the Lagrangean dual directly. as a nonlinear programming problem and letting \bar{u} be the Lagrange multipliers that correspond to the optimal solution. However, the multipliers could be obtained by solving the Lagrangean dual directly.

2.7 Superadditive/Subadditive Dual

The *subadditive dual* [17] has been studied in connection with integer programming problems, which can be written

$$\min_{x \in D} \{cx \mid Ax \geq b\} \quad (15)$$

where D is the set of n -tuples of nonnegative integers. (The *superadditive dual* is used when one maximizes cx subject to $Ax \leq b$.) The subadditive dual can be viewed as an inference dual, using a form of inference that generalizes inference by nonnegative linear combination and domination. Let a real-valued function $h(\cdot)$ be *subadditive* when $h(d + d') \leq h(d) + h(d')$ for all d, d' . We will say that a proof in \mathcal{P} derives $cx \geq v$ from $Ax \geq b$ when $h(Ax) \geq h(b)$ dominates $cx \geq v$ for some nondecreasing, subadditive function h , and that $h(Ax) \geq h(b)$ dominates $cx \geq v$ when $h(b) \geq v$ and $h(Ax) \leq cx$ for all $x \in D$. This inference method can be shown to be complete for linear integer inequalities, based on cutting plane theory developed in [3, 4, 16, 20].

When \mathcal{P} is defined in this way, the inference dual (2) becomes

$$\max_{h \in H} \{g(b) \mid g(Ax) \leq cx, \text{ all } x \in D\}$$

where H is the set of subadditive, nondecreasing functions. This is the subadditive dual of (15). Since \mathcal{P} is complete, it is a strong dual.

The subadditive dual has been used primarily for sensitivity analysis in integer programming (e.g. [5]). It has apparently not been used in the context of nogood-based search. Since the form of domination used to define \mathcal{P} is that used in the surrogate dual, one could obtain a Lagrangean analog of the subadditive dual by substituting the form of domination used in the Lagrangean dual.

2.8 Duals for Propositional Satisfiability

Propositional satisfiability (SAT) problems are often solved by a *Davis-Putnam-Loveland* (DPL) method with *clause learning* (e.g., [18]). These methods can be seen as nogood-based search methods derived from an inference dual.

The SAT problem can be written

$$\min_{x \in \{0,1\}^n} \{0 \mid \mathcal{C}\} \quad (16)$$

where \mathcal{C} is a set of logical clauses. To formulate an inference dual, let \mathcal{P} consist of *unit resolution* proofs (i.e., repeated elimination of variables that occur in unit clauses until no unit clauses remain). The dual problem (2) has optimal value ∞ when unit resolution proves unsatisfiability by deriving the empty clause. Since unit resolution is not a complete inference method, there may be a duality gap: the dual may have optimal value zero when the primal is unsatisfiable.

Now suppose we solve (16) by branching on the propositional variables x_j . At each node of the search tree, certain variables x_F are fixed to \bar{x}_F . Let \mathcal{U}

contain the unit clause x_j when $\bar{x}_j = 1$ and $\neg x_j$ when $\bar{x}_j = 0$. We now solve the inference dual of (16) with the clause set $\mathcal{C} \cup \mathcal{U}$. If the optimal value is ∞ , we generate a nogood and backtrack; otherwise we continue to branch. To generate the nogood, we identify a subset $\bar{\mathcal{U}}$ of \mathcal{U} for which some portion of the unit resolution proof obtains the empty clause from $\mathcal{C} \cup \bar{\mathcal{U}}$. Then

$$\bigvee_{x_j \in \bar{\mathcal{U}}} \neg x_j \vee \bigvee_{\neg x_j \in \bar{\mathcal{U}}} x_j$$

is a nogood or “learned” clause that can be added to \mathcal{C} before backtracking. This results in a basic DPL algorithm with clause learning.

Similar algorithms can be developed for other types of resolution, including full resolution, which could be terminated if it fails to derive the empty clause in a fixed amount of time.

2.9 Domain Filtering Duals

A domain filtering algorithm can be viewed as an inference method and can therefore define an inference dual. For concreteness suppose that the domain of each variable is an interval $[L_j, U_j]$, and consider a filtering algorithm that tightens lower and upper bounds. The inference dual is most straightforward when we assume the objective function $f(x)$ is monotone nondecreasing. In this case we can let \mathcal{P} contain all possible proofs consisting of an application of the filtering algorithm to obtain reduced bounds $[L_j, U_j]$, followed by an inference that $f(x) \geq f(L_1, \dots, L_n)$. This defines an inference dual (2).

If the filtering method achieves bounds consistency, $f(L_1, \dots, L_n)$ is the optimal value of (1), there is no duality gap, and the problem is solved.

If bounds consistency is not achieved, the dual can be useful in nogood-based search. Suppose we search by domain splitting, and let \mathcal{B} contain the bounds currently imposed by the branching process. We can examine the filtering process to identify a subset $\bar{\mathcal{B}} \subset \mathcal{B}$ of bounds that are actually used to obtain the lower bounds L_j that affect the value of $f(L_1, \dots, L_n)$. The resulting nogood

$$\bar{\mathcal{B}} \rightarrow (f(x) \geq f(L_1, \dots, L_n))$$

can be used as described earlier.

A related idea has proved very useful in planning and scheduling [12, 15, 13]. Let each variable x_j in x_F indicate which facility will process job j . The jobs assigned to a facility i must be scheduled subject to time windows; the variables in x_U indicate the start times of the jobs. The processing times of job j may be different on the different facilities. For definiteness, suppose the objective $f(x)$ is to minimize the latest completion time over all the jobs (i.e., minimize makespan). We solve each scheduling problem with a constraint programming method that combines branching with edge finding. This can be viewed as a complete inference method that defines an inference dual with no duality gap. Let \bar{v}_i be the minimum makespan obtained on facility i for a given assignment

\bar{x}_F . By examining the edge finding and branching process, we identify a subset x_J of job assignments for which the minimum makespan on each facility i is still \bar{v}_i . Then we have the nogood

$$(x_F = \bar{x}_F) \rightarrow \left(f(x) \geq \max_i \{\bar{v}_i\} \right)$$

Nogoods of this sort are accumulated in a master problem that is solved to obtain the next \bar{x}_F , thus yielding a generalized form of Benders decomposition [14]. The assignments in x_J are identified by noting which jobs play a role in the edge finding at each node of the search tree; details are provided in [13].

3 Relaxation Duality

A *parameterized relaxation* of the optimization problem (1) can be written

$$\min_{x \in D} \{f(x, u) \mid \mathcal{C}(u)\} \quad (17)$$

where $u \in U$ is a vector of *dual variables*. The constraint set $\mathcal{C}(u)$ is a relaxation of \mathcal{C} , in the sense that every $x \in D$ that satisfies \mathcal{C} satisfies $\mathcal{C}(u)$. The objective function $f(x, u)$ is a lower bound on $f(x)$ for all x feasible in (1); that is, $f(x, u) \leq f(x)$ for all $x \in D$ satisfying \mathcal{C} .

Clearly the optimal value of the relaxation (17) is a lower bound on the optimal value of (1). The *relaxation dual* of (1) is the problem of finding the parameter u that yields the tightest bound [12]:

$$\max_{u \in U} \left\{ \min_{x \in D} \{f(x, u) \mid \mathcal{C}(x, u)\} \right\} \quad (18)$$

Let z^* be the optimal value of (1), and $\theta(u)$ be the optimal value of the minimization problem in (18). Since $\theta(u)$ is a lower bound on z^* for every $u \in U$, we have weak duality: the optimal value \bar{v} of the relaxation dual (18) is a lower bound on z^* .

The lower bound \bar{v} can abbreviate the search, as for example in a branch-and-relax (branch-and-bound) scheme. The parameterized relaxation is chosen so that $\theta(u)$ is easy to compute. The dual problem of maximizing $\theta(u)$ over $u \in U$ may be solved by some kind of search procedure, such as subgradient optimization in the case of Lagrangean relaxation. The maximization problem need not be solved to optimality, since any $\theta(u)$ is a valid lower bound.

3.1 Equivalence to an Inference Dual

Although inference and relaxation duality are very different concepts, a relaxation dual is always formally equivalent to an inference dual, provided there exists a solution algorithm for the parameterized relaxation. There does not seem to be a natural converse for this proposition.

To formulate the relaxation dual (18) as an inference dual, suppose that an algorithm $P(u)$ is available for computing $\theta(u)$ for any given $u \in U$. We can regard $P(u)$ as a proof that $f(x) \geq \theta(u)$ and let $\mathcal{P} = \{P(u) \mid u \in U\}$. The resulting inference dual (2) is $\max_{u \in U} \{\theta(u)\}$, which is identical to the relaxation dual (18).

3.2 Linear Programming and Surrogate Duals

A simple parameterized relaxation for the inequality-constrained problem (10) uses a surrogate relaxation of the constraints but leaves the objective function unchanged. The relaxation therefore minimizes $f(x)$ subject to $ug(x) \leq 0$, where $u \geq 0$. The resulting relaxation dual is the surrogate dual (11) of (1). Since the surrogate dual of an LP problem is the LP dual, the relaxation dual of an LP problem is likewise the LP dual.

3.3 Lagrangean Dual

Another parameterized relaxation for (10) removes the constraints entirely but “dualizes” them in the objective function. The parameterized relaxation minimizes $f(u, x) = f(x) + ug(x)$ subject to $x \in D$. The function $f(x, u)$ is a lower bound on $f(x)$ for all feasible x since $ug(x) \leq 0$ when $u \geq 0$ and $g(x) \leq 0$. The resulting relaxation dual is precisely the Lagrangean dual (13).

The close connection between surrogate and Lagrangean duals, conceived as inference duals, is much less obvious when they are reinterpreted as relaxation duals.

3.4 Superadditive/Subadditive Dual

The subadditive dual discussed earlier can be viewed as a relaxation dual that generalizes the surrogate dual. We can give the integer programming problem (15) a relaxation parameterized by subadditive, nondecreasing functions h , in which we minimize cx subject to $h(Ax) \geq h(b)$ and $x \in D$. (In the surrogate dual, the function h is multiplication by a vector u of nonnegative multipliers.) This yields the relaxation dual

$$\max_{h \in H} \left\{ \min_{x \in D} \{cx \mid h(Ax) \geq h(b)\} \right\}$$

which is equivalent to the subadditive dual.

3.5 Constraint Dual

The constraint dual is related to a relaxation dual. More precisely, the constraint dual can be given a parameterized relaxation that yields a relaxation dual. A special case of the relaxation has been applied in mini-bucket elimination and perhaps elsewhere.

It is convenient to let x_J denote the tuple of variables x_j for $j \in J$. Given a constraint set \mathcal{C} , the *constraint dual* of \mathcal{C} is formed by “standardizing apart” variables that occur in different constraints and then equating these variables. So if $x_{j_1}, \dots, x_{j_{n_i}}$ are the variables in constraint $C_i \in \mathcal{C}$, let $y^i = (y_1^i, \dots, y_{n_i}^i)$ be a renaming of these variables. Also let J_{ik} be the index set of variables that occur in both C_i and C_k . The constraint dual associates the dual variable y^i with each constraint C_i , where the domain D_i of y^i is the set of tuples that satisfy C_i . The dual constraint set consists of the binary constraints $y_{J_{ik}}^i = y_{J_{ik}}^k$ for each pair i, k .

The constraint dual can be relaxed by replacing each $y_{J_{ik}}^i = y_{J_{ik}}^k$ with $y_{J'_{ik}}^i = y_{J'_{ik}}^k$ where $J'_{ik} \subset J_{ik}$. It is helpful to think about the constraint graph G corresponding to the dual, which contains a vertex for each variable y_j^i and an edge between two variables when they occur in the same tuple y^i or in the same equality constraint. Removing equality constraints deletes the corresponding edges from G , resulting in a sparser graph $G(E)$, where E is the set of edges corresponding to the equality constraints that remain. The relaxation is therefore parameterized by the subset E of edges that defines $G(E)$. This relaxation also serves as a parameterized relaxation $\mathcal{C}(E)$ of the original constraint set \mathcal{C} . Thus if the constraint satisfaction problem is written

$$\min_{y^i \in D_i, \text{ all } i} \{0 \mid \mathcal{C}\} \quad (19)$$

then we can write the relaxation dual

$$\max_{E \in \mathcal{E}} \{0 \mid \mathcal{C}(E)\}$$

where \mathcal{E} is some (generally incomplete) family of subsets E . To solve the dual, we check the feasibility of $\mathcal{C}(E)$ for each $E \in \mathcal{E}$. The family \mathcal{E} normally would be chosen so $G(E)$ has small induced width for $E \in \mathcal{E}$, since in this case $\mathcal{C}(E)$ is easier to solve by nonserial dynamic programming.

One way to construct \mathcal{E} is to define sets of “mini-buckets” [7, 8]. We consider various partitions of the constraints in \mathcal{C} , where the k th partition defines disjoint subsets or mini-buckets $\mathcal{C}_{k1}, \dots, \mathcal{C}_{km_k}$. For each k and each $t \in \{1, \dots, m_k\}$ we let E_{kt} contain the edges corresponding to equality constraints between variables occurring in \mathcal{C}_{kt} , so that $\mathcal{C}(E_{kt}) = \mathcal{C}_{kt}$. Now \mathcal{E} is the family of all sets E_{kt} . Thus, rather than solve the relaxations $\mathcal{C}_{k1}, \dots, \mathcal{C}_{km_k}$ corresponding to a single set of mini-buckets as in [7], we solve relaxations \mathcal{C}_{kt} for all E_{kt} . Other relaxation duals based on reducing the induced width are discussed in [12]. All of these approaches can be applied to problems (19) with a general objective function $f(x)$, as is done in mini-bucket elimination schemes.

References

1. F. Barahona and R. Anbil. The volume algorithm: Producing primal solutions with a subgradient algorithm. *Mathematical Programming*, 87:385–399, 2000.

2. J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.
3. C. E. Blair and R. G. Jeroslow. The value function of a mixed integer program. *Mathematical Programming*, 23:237–273, 1982.
4. V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.
5. W. Cook, A. M. H. Gerards, A. Schrijver, and E. Tardos. Sensitivity results in integer programming. *Mathematical Programming*, 34:251–264, 1986.
6. M. Dawande and J. N. Hooker. Inference-based sensitivity analysis for mixed integer/linear programming. *Operations Research*, 48:623–634, 2000.
7. R. Dechter. Mini-buckets: A general scheme of generating approximations in automated reasoning. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI 97)*, pages 1297–1302, 1997.
8. R. Dechter and I. Rish. Mini-buckets: A general scheme for bounded inference. *Journal of the ACM*, 50:107–153, 2003.
9. A. M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 1972.
10. F. Glover. Surrogate constraint duality in mathematical programming. *Operations Research*, 23:434–451, 1975.
11. J. N. Hooker. Inference duality as a basis for sensitivity analysis. *Constraints*, 4:104–112, 1999.
12. J. N. Hooker. *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*. Wiley, New York, 2000.
13. J. N. Hooker. A hybrid method for planning and scheduling. *Constraints*, 10:385–401, 2005.
14. J. N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming*, 96:33–60, 2003.
15. V. Jain and I. E. Grossmann. Algorithms for hybrid MILP/CP models for a class of optimization problems. *INFORMS Journal on Computing*, 13:258–276, 2001.
16. R. G. Jeroslow. Cutting plane theory: Algebraic methods. *Discrete Mathematics*, 23:121–150, 1978.
17. E. L. Johnson. Cyclic groups, cutting planes and shortest paths. In T. C. Hu and S. Robinson, editors, *Mathematical Programming*, pages 185–211. Academic Press, 1973.
18. M. Moskewicz, C. F. Madigan, Ying Zhao, Lintao Zhang, and S. Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference (DAC01)*, pages 530–535, 2001.
19. A. Nedic and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12:109–138, 2001.
20. L. A. Wolsey. The b-hull of an integer program. *Discrete Applied Mathematics*, 3:193–201, 1981.