Mathematical Programming 19 (1980) 255–278.
North-Holland Publishing Company

CALCULATING SURROGATE CONSTRAINTS

M.E. DYER

Mathematics Department, Teesside Polytechnic, Middlesbrough, Cleveland, Great Britain

Received 2 June 1978
Revised manuscript received 12 March 1980

Various theoretical properties of the surrogate dual of a mathematical programming problem are discussed, including some connections with the Lagrangean dual. Two algorithms for solving the surrogate dual, suggested by analogy with Lagrangean optimisation, are described and proofs of their convergence given. A simple example is solved using each method.

Key words: Surrogate Constraints, Subgradient Methods, Generalised Programming, Surrogate Duality.

1. Introduction

Surrogate constraints were introduced into mathematical programming by Glover [8]. The idea is to replace a mathematical program, by taking a suitable convex combination of the constraints, with a surrogate problem having only one constraint. Luenberger [20] has shown that for quasi-convex programs, the surrogate problem can solve the original program if the multipliers are correctly chosen. Greenberg and Pierskalla [10] treated the general surrogate problem theoretically, and established some results useful for the construction of optimal (i.e. tightest) surrogate constraints. They showed that this involves the maximisation of a quasiconcave, though possibly discontinuous, function. Some results allied to theirs are given in Section 2 below. Greenberg [13] extended the theory to a more general case involving nonlinear combinations of constraints, and Greenberg and Pierskalla [11, 12] developed a theory of quasiconvexity which they hoped would prove useful in this context. Glover [9] has given a theory of surrogate duality in the spirit of Geoffrion's [7] theory of Lagrangean duality.

Little attention seems to have been directed, however, to schemes for generating optimal surrogate constraints, although Greenberg [13] refers to a generalised programming method for this purpose. The aim in this paper is to investigate possible methods. Two algorithms are proposed, which are natural extensions of methods for the Lagrangean case, one analogous to generalised linear programming, the other to subgradient methods. The convergence of each is established.

In Section 2 some theoretical properties of the surrogate problem are discussed and connections with the Lagrangean dual are shown. Section 3, which is not central to the paper, briefly reviews a method for finding an interior point of a polytope for application in the algorithm of Section 4. Section 4 presents the first algorithm, and Section 5 the second. In Section 6 a simple example is solved using each of the methods.

A word should perhaps be said about the use of optimal surrogates. In the best cases, where there is no duality gap, the optimal surrogate problem solves the original mathematical programming problem. In other cases it merely provides a bound on the optimal objective function value in the original. However, this bound is at least as good and usually better than that obtained from optimising the Lagrangean. For integer programming, for example, the bound and the constraint itself are useful within global enumeration schemes.

The development is principally in terms of mathematical programs defined by inequalities, since this endows sufficient structure to render an optimisation scheme feasible. However, it might be noted that the work of Bradley [4] and others shows that there is always a surrogate problem for equality constrained integer programs which has no duality gap. Therefore it might be hoped that the optimum surrogate will closely bound the primal in the inequality case of integer programming also.

2. The surrogate problem

2.1. Definition and discussion

Consider the canonical primal mathematical program [9]:

$$\text{(P)} \qquad f^* = \inf\{f(x): x \in X, g(x) \leq 0\}$$

where $g(x) = (g_1(x), g_2(x), \dots, g_m(x)) \in R^m$ and $X \subseteq R^n$.

It will be assumed here that X is compact, the function f is a lower semicontinuous (l.s.c.) real-valued function on X, and the g_i are continuous on X. These are the assumptions of [10] and immediately imply that the "inf" in (P) can be replaced by "min", i.e. f^* is attained.

Now, for any $u \in R^m$ with $u \geq 0$, define [9]:

$$X(u) = \{x \in X: u^t g(x) \leq 0\} \quad \text{and} \quad w(u) = \min\{f(x): x \in X(u)\}.$$

Then

$$\text{(Q)} \qquad w^* = \sup\{w(u): u \geq 0, u \in R^m\}$$

is the surrogate dual program for (P). It is evident that $w(0) \leq w(u)$ for all $u \neq 0$ and thus zero may be excluded as a possible value for u. It is also clear that for any $k > 0$, $w(ku) = w(u)$. Thus an arbitrary normalisation of the vectors u is possible and has the advantage that they will then belong to a compact set. Thus if $\|\cdot\|$ is any norm on R^m, (Q) can be rewritten

$$\text{(Q)} \qquad w^* = \sup\{w(u): u \geq 0, \|u\| = 1, u \in R^m\}.$$

The two norms which will be used here are the L_1 and L_2 norms, i.e. either $\sum u_i = 1$ or $\sum u_i^2 = 1$. It will be observed that since the u_i are nonnegative, the former gives a linear equality constraint which will be written $e^t u = 1$ where $e^t = (1, 1, \dots, 1)$. This L_1 normalisation will be used in the remainder of this section, though the results can be obviously modified to any norm. For this purpose it is convenient to define the (compact convex) set $S = \{u \geq 0: e^t u = 1\}$.

It should be noted that even though f^* and $w(u)$ will always be attained under the assumptions made, w^* may not be equal to any $w(u)$. In fact, in the absence of convexity, it appears that very strong conditions on the problem are needed to ensure that w^* is achieved. The following example shows that it is not sufficient, for example, that f be smooth, the g_i linear and X strongly connected and compact.

Example 1.

$$\text{(P)} \qquad f^* = \min \quad 5 - x_1^2 - (x_2 - 1)^2,$$
$$\text{subject to} \quad x_1 - x_2 \leq 0, \qquad x_1 + x_2 - 2 \leq 0$$

and X the L-shaped region of R^2 shown in Fig. 1.

Letting $u_1 = 1 - u_2 = \lambda$, it follows that

$$w(\lambda) = \begin{cases} 4\lambda(2-\lambda) & \text{for } \lambda < \frac{1}{2}, \\ -8\lambda(2\lambda - 1) & \text{for } \frac{1}{2} \leq \lambda \leq \frac{3}{4}, \\ -3 & \text{for } \lambda \geq \frac{3}{4}. \end{cases}$$

The function w has a discontinuity at $\lambda = \frac{1}{2}$, and the supremum $w^* = 3$ is not attained. It is sketched in Fig. 2.

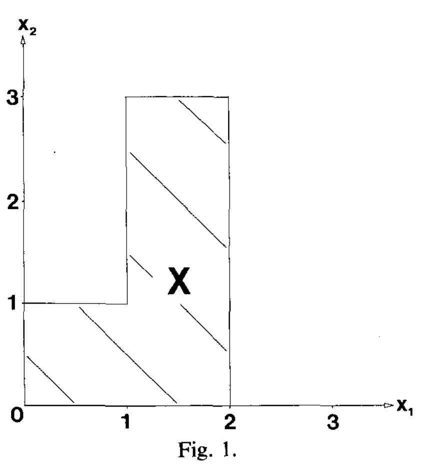

Fig. 1.

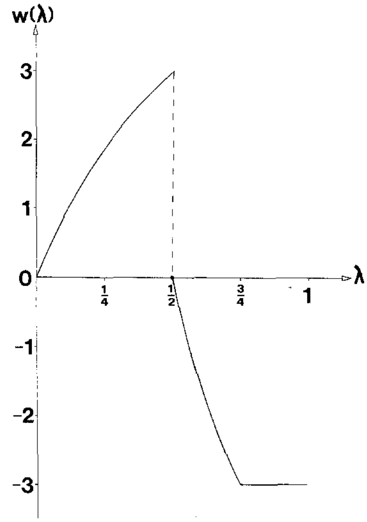

Fig. 2.

Simple examples of mixed-integer programs in which w^* is not attained can also be constructed. However, it will be shown that for a class of problems which includes pure-integer programs, w^* will necessarily be achieved.

2.2. Properties

In the following, the results stated without proof are either from, or very similar to, results of [10]. They are reproduced for completeness. The prefix "int" denotes interior (usually relative to S) and "co" the convex hull of a set.

Proposition 1. $w^* \leq f^*$ *(Weak Duality).*

Proposition 2. *w is a l.s.c. quasiconcave function on S.*

Corollary 1. *If (P) is feasible, then for each $\epsilon > 0$, the set $W(\epsilon) = \{u \in S\colon w(u) > w^* - \epsilon\}$ is a nonempty, open, convex subset of S.*

Proof. w^* is clearly finite if (P) is feasible, so $W(\epsilon)$ is nonempty. The other properties are immediate consequences of Proposition 2.

The quantity $(f^* - w^*)$ is called the duality gap. The best situation is where there is no gap i.e. $f^* = w^*$. Let $Y = \{x \in X\colon g(x) \leq 0\}$ be the feasible set of (P). Clearly $Y \subseteq X(u)$ for all u. The following simple result then follows directly.

Proposition 3. *w^* is attained and equal to f^* if and only if $w(u) = f(x)$ for some $u \in S$ and $x \in Y$. Then x solves (P) and u solves (Q).*

The case where $w^* = f^*$ but it is not attained may thus be viewed as an "infinitesimal" duality gap, since the dual will not then give a solution to the primal among the minimisers of $w(u)$ for any u.

As already observed, w^* need not be achieved on S, this being true even if there is no duality gap. Define, however, a discrete programming problem as one for which the set of real numbers $\{f(x)\colon x \in X\}$ has no cluster point. This clearly includes all pure integer programs. Then

Proposition 4. *If (P) is a feasible discrete programming problem, then w^* is attained on an open convex subset of S.*

Proof. Clearly $w^* < \infty$. If it is not attained the there is an increasing sequence $w(u_r) \to w^*$ with no $w(u_r) = w^*$. But $w(u_r) = f(x_r)$ for some $x \in X$, which implies that w^* is a cluster point of $\{f(x_r)\}$ which is impossible. Also since w^* is not a cluster point of $\{f(x)\}$ there exists an $\epsilon > 0$ such that $w^* - \epsilon < f(x) \le w^*$ implies $f(x) = w^*$ for all $x \in X$. Thus if $w^* - \epsilon < w(u) \le w^*$, then $w(u) = w^*$. The result now follows from Corollary 1.

In what follows it will be convenient to have a concise notation for certain families of sets. Let $F(\alpha) = \{x \in X\colon f(x) \le \alpha\}$, then obviously $F(\alpha) \cap X(u) \ne \emptyset$ if and only if $w(u) \le \alpha$. Denote by $M(u)$ the set of optimal solutions $X(u) \cap F(w(u))$ to the surrogate problem for u. Let $G(\alpha) = g(F(\alpha))$ and $Z(u) = g(M(u))$. Also denote by $G^*(\alpha)$ the set $\{s \in S\colon g^t s \ge 0$ for all $g \in G(\alpha)\}$, with a similar definition for $Z^*(u)$. These are sets "polar" to $G(\alpha)$, $Z(u)$. The following results then give conditions for optimality in the surrogate dual problem.

Proposition 5. *$w(u) > \alpha$ if and only if $u \in \operatorname{int} G^*(\alpha)$.*

Proof. $u \in \operatorname{int} G^*(\alpha)$ if and only if $g^t u > 0$ for all $g \in G(\alpha)$, which is true if and only if $X(u) \cap F(\alpha) = \emptyset$, which is again true if and only if $w(u) > \alpha$.

Corollary 2. *w^* is the minimum number α such that $\operatorname{int} G^*(\alpha) = \emptyset$.*

Proof. If $\alpha < w^*$, then there exists u such that $w(u) > \alpha$ and hence $\operatorname{int} G^*(\alpha) \ne \emptyset$. If $\alpha \ge w^*$, then $u \in \operatorname{int} G^*(\alpha)$ implies $w(u) > \alpha \ge w^*$, which is impossible, hence $\operatorname{int} G^*(\alpha) = \emptyset$.

Proposition 6. *$\operatorname{int} G^*(\alpha) = \emptyset$ if and only if there exists $v \in \operatorname{co}(G(\alpha))$ such that $v \le 0$.*

Proof. $\operatorname{int} G^*(\alpha) = \emptyset$ if and only if there is no $u \in S$ such that $u^t g > 0$ for all $g \in G(\alpha)$, which is true if and only if there is no u in S such that $u^t g > 0$ for all $g \in \operatorname{co}(G(\alpha))$. Since $v \le 0$ and $u \ge 0$, $u^t v \le 0$ and thus the existence of v implies $\operatorname{int} G^*(\alpha) = \emptyset$. Conversely if no such v exists, then $\operatorname{co}(G(\alpha))$ is disjoint from the nonpositive orthant N of R^m. Now since f is l.s.c., $F(\alpha)$ is closed and therefore compact. Thus since g is continuous $G(\alpha) = g(F(\alpha))$ is compact. Hence $\mathrm{co}(G(\alpha))$ is compact, and since N is closed, N and $\mathrm{co}(G(\alpha))$ can be strictly separated by a hyperplane. This implies the existence of a $u \in S$ such that $u^t g > 0$ for all $g \in \mathrm{co}(G(\alpha))$ and hence $\mathrm{int}\, G^*(\alpha) \neq \emptyset$.

Proposition 6 and Corollary 2 together immediately imply.

Proposition 7. *w^* is the minimum number α for which there exists a $v \leq 0$ in* $\mathrm{co}(G(\alpha))$.

If w^* is attained, then equally obviously.

Corollary 3. *u is a maximising point of w if and only if there is a $v \leq 0$ in* $\mathrm{co}(G(w(u)))$.

Corollary 3, which gives a necessary and sufficient condition for u to maximise w can be compared with the gap detection theorem of Greenberg and Pierskalla [10] which states that

Proposition 8. *If there is a $v \leq 0$ in* $\mathrm{co}(Z(u))$, *then u is a maximising point of w.*

This can be deduced immediately from Corollary 3 on noting that $Z(u) \subseteq G(w(u))$. Proposition 8 gives, however, only a sufficient condition for maximisation which need not be satisfied even when w^* is attained, as the following simple example shows.

Example 2.

$$\text{(P)} \qquad f^* = \min \quad 2 - x_1 - 2x_2,$$
$$\text{subject to} \quad 2x_1 + 4x_2 \leq 3, \qquad 4x_1 + 2x_2 \leq 3,$$
$$X = \{(x_1, x_2)\colon x_j \in \{0, 1\}\}.$$

If $u_1 = (1 - u_2) = \lambda$, then for $0 \leq \lambda \leq \frac{1}{2}$, $w(\lambda) = 0$ and $M(\lambda) = \{(0, 1)\}$. For $\frac{1}{2} < \lambda \leq 1$, $w(\lambda) = 1$ and $M(\lambda) = \{(1, 0)\}$. The maximum is attained on the (open, convex) interval $\frac{1}{2} < \lambda \leq 1$, but for all λ in this interval $Z(\lambda) = \{(-1, 1)\}$. Clearly $\mathrm{co}(Z(\lambda))$ does not contain a nonpositive vector for any maximising λ.

It will be observed that in this example w^* is attained on an open set of S. The following proposition shows that this is in fact the source of the difficulty in attempting to apply Proposition 8.

Proposition 9. *If w^* is attained, but not on any open set of S, then for all u such that $w(u) = w^*$, there is a $v \leq 0$ in* $\mathrm{co}(Z(u))$.

Proof. Let $W = \{u \in S\colon w(u) = w^*\}$. Then W is convex and not of full dimension in S. Thus there is a hyperplane H of R^m, other than $e^t u = 1$, which contains W. Let $K = H \cap S$. Clearly int $K = \emptyset$ in S and $W \subseteq K$. Choose any $\bar{u} \in W$ and $u \in S - K$. Thus $w(u) < w^*$. Define $u_r = (1/r)u + (1 - 1/r)\bar{u}$ for $r = 1, 2, \ldots$. Clearly $u_r \in S - K$ for all r and hence $w(u_r) < w^*$. Let $x_r \in M(u_r)$. Clearly $u_r^t g(x_r) \leq 0$ and since $f(x_r) < w^*$, $x_r \notin X(\bar{u})$ so $\bar{u}^t g(x_r) > 0$. But this implies that $u^t g(x_r) \leq 0$. Let $\bar{x}$ be any cluster point of x_r and assume that $x_r \to \bar{x}$. Now letting $r \to \infty$, $\bar{u}^t g(\bar{x}) \leq 0$ so $\bar{x} \in X(\bar{u})$ and hence $f(\bar{x}) \geq w^*$. But since f is l.s.c., $f(\bar{x}) \leq w^*$. Thus $f(\bar{x}) = w^*$ and hence $\bar{x} \in M(\bar{u})$. But from the continuity of g, it also follows that $u^t g(\bar{x}) \leq 0$. Therefore $u \notin$ int $Z^*(u)$. This implies that int $Z^*(u) = \emptyset$ in S. Then an argument almost identical to the second part of the proof of Proposition 6 establishes the existence of v.

Thus Greenberg and Pierskalla's condition would be sufficient in many cases, but in view of Proposition 4 it appears to be of restricted use for discrete programming problems. It is also of no assistance if w^* is not attained. The nearest result to this which can be guaranteed in general is a strengthened form of Proposition 7, as follows. For this purpose write $M(u, \alpha) = X(u) \cap F(\alpha)$ and $Z(u, \alpha) = g(M(u, \alpha))$. The preliminary lemma is given first.

Lemma 1. *If $A \subseteq R^m$ is any set of points such that* co(A) *contains a point of the nonpositive orthant, then there is a finite subset B of A such that* co(B) *contains a point of the nonpositive orthant but no point of the positive orthant.*

Proof. From Caratheodory's theorem all points of A can be represented as convex combinations of finite subsets of A. Let B be a minimal finite subset which contains a nonpositive point in its convex hull. Suppose this is $\sum \lambda_i b_i \leq 0$, with $\sum \lambda_i = 1$. Clearly $\lambda_i > 0$ for all i. Suppose now $\sum \mu_i b_i > 0$ with $\sum \mu_i = 1$ for any $\mu_i \geq 0$. Then, for any $\beta \geq 0$ define $\lambda_i' = (1 + \beta)\lambda_i - \beta\mu_i$. Clearly $\sum \lambda_i' = 1$ and $\sum \lambda_i' b_i \leq 0$. Now choose β to satisfy

$$\beta/(1 + \beta) = \min_i\{\lambda_i/\mu_i\colon \mu_i \neq 0\} = \lambda_k/\mu_k \quad \text{say.}$$

Then β is positive and well-defined since $\lambda_i > 0$, $\mu_i > 0$ for some i, and $\lambda_i < \mu_i$ for some i. Note that if $\lambda_i \geq \mu_i$ for all i, then $\lambda_i = \mu_i$ for all i which is clearly impossible. Now, with the chosen value of β, $\lambda_i' \geq 0$ for all i and $\lambda_k' = 0$. This contradicts the minimality of B, and the result is established.

Proposition 10. *There is $u \in S$ such that a $v \leq 0$ exists in* co$(Z(u, w^*))$. *The number w^* is minimal with respect to this property.*

Proof. Proposition 7 shows there is a $v \leq 0$ in co$(G(\alpha))$ if and only if $w^* \leq \alpha$. Thus from the lemma there is a finite subset of X, C say, such that co$\{g(x)\colon x \in C\}$ contains a nonpositive point but no positive point. Thus, from the separating hyperplane theorem, there is a $u \in S$ such that $u^t g(x) \leq 0$ for all $x \in C$. Thus

$C \subseteq X(u)$, and since clearly $C \subseteq F(\alpha), C \subseteq M(u, \alpha)$. The conclusion now follows.

Greenberg and Pierskalla's sufficient condition is met only if the u of Proposition 10 satisfies $w(u) = w^*$ also. This need not be the case and it is possible to construct problems in which all points which fulfil the condition of Proposition 10 are far from optimal.

Proposition 10 is clearly stronger than Proposition 7, but it is not exploited in the algorithms developed here. The following discussion examines the implications of these two results.

It is useful to write the content of these results in an equivalent, but more suggestive form as follows.

(Q′) $$w^* = \min \max_{\lambda(x)>0} f(x),$$

such that

$$\sum_{x \in X} \lambda(x) g(x) \leq 0,$$

$$\sum_{x \in X} \lambda(x) = 1,$$

$\lambda(x) \geq 0$ for all $x \in X$, and all but a finite number of $\lambda(x) = 0$.

Now it is clear that (Q′) is a form of generalised program. It is in fact a (generalised) bottleneck linear program. See, for example [6], which gives algorithms for the bottleneck linear program in the finite case. Note that Proposition 10 implies the existence of an optimal "dual multiplier" u such that $u^t g(x) \leq 0$ for all x such that $\lambda(x) > 0$. There is a close analogy with one of the algorithms of [6].

Suppose now that the assumption of lower semicontinuity on f is strengthened to continuity. Then it is well-known [21] that the Lagrangean dual problem to (P)

(D) $$L^* = \sup\{\min\{f(x) + d^t g(x) \colon x \in X\} \colon d \geq 0, d \in R^m\}$$

can be written as

(D′) $$L^* = \min \sum_{x \in X} \lambda(x) f(x),$$

the $\lambda(x)$ being subject to the constraints of (Q′). The coincidence of the constraint sets clearly implies the result of [10] that $w^* \geq L^*$ for any (P), since the objective function of (Q′) is at least equal to that of (D′) for any feasible $\lambda(x)$. It also implies that if $\lambda(x), x \in C$, are nonzero in any optimal solution to (D′), then $w^* \leq \max\{f(x) \colon x \in C\}$. This bound could be useful in estimating the potential improvement over the Lagrangean maximum that surrogate methods might produce. In practice, the surrogate dual would probably be used as an adjunct to Lagrangean methods in an attempt to close the Lagrangean duality gap. It might then seem that the (normalised) optimum Lagrange multiplier would be a good starting point in a search for the best surrogate multiplier. The following shows that, in a certain sense, this is true.

Proposition 11. *If d is an optimum Lagrange multiplier, and $\bar{d}$ its L_1-normalisation, then $w(\bar{d}) \geq L^*$. Moreover exactly one of the following holds:*

(i) $w(\bar{d}) > L^*$,

(ii) $w(\bar{d}) = L^*$, *but every neighbourhood of $\bar{d}$ in S contains a point u such that* $w(u) > w(\bar{d})$,

(iii) $w(\bar{d}) = L^* = w^*$.

Proof. For all $x \in X, f(x) + d^t g(x) \geq L^*$. Thus $f(x) \geq L^*$ for all $x \in X(\bar{d})$, so $w(\bar{d}) \geq L^*$. Suppose (i) does not hold, so $w(\bar{d}) = L^*$. Then for any $x \in M(\bar{d}), f(x) = L^*$. Then, for such an $x, f(x) + d^t g(x) \leq L^*$, which implies that $f(x) + d^t g(x) = L^*$ and hence $\bar{d}^t g(x) = 0$. Suppose (iii) does not hold, so there is a $u \in S$ such that $w(u) > w(\bar{d})$. Let $u_r = (1 - 1/r)\bar{d} + (1/r)u$ so $u_r \to \bar{d}$. Since w is quasiconcave, $w(u_r) \geq w(\bar{d})$ for all r. Now for any $x \in M(\bar{d}), u_r^t g(x) = (1/r)u^t g(x) > 0$ since $x \notin X(u)$. Suppose $w(u_r) = w(\bar{d})$ for any r. Then there is a $y \in X(u_r)$ with $f(y) = w(\bar{d})$. But $y \notin X(u)$, and this implies $y \in X(\bar{d})$. Thus $y \in M(\bar{d})$, which is a contradiction. Thus $w(u_r) > w(\bar{d})$ for all r, which shows that (ii) holds.

While this Proposition shows that $\bar{d}$ is a reasonable estimate for an optimal surrogate multiplier, it should be observed that it is not difficult to construct examples in which it bears no relation to the set of optimal multipliers. The following is such an example.

Example 3.

(P) $$f^* = \min \; -4x_1 - 4x_2 + x_4 + 2x_5$$

such that

$$\begin{aligned} 2x_1 + x_2 + x_3 - 3x_4 - x_5 &\leq 0, \\ x_1 + 2x_2 + x_3 + x_4 - x_5 &\leq 0, \\ x_1 + x_2 - 2x_3 + x_4 &\leq 0, \end{aligned}$$

and

$$X = \{(x_1, x_2, x_3, x_4, x_5): x_i \in \{0, 1\} \text{ and } x_1 + x_2 + x_3 + x_4 + x_5 = 1\}.$$

Here the unique optimal Lagrange multiplier $d = (1, 1, 1)$ and $L^* = 0$. Then $\bar{d} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and $w(\bar{d}) = 0$. Every neighbourhood of $\bar{d}$ contains a point at which $w(\cdot) = 1$, but $\bar{d}$ is not close to any of the points at which $w^* = 2$ is achieved, i.e. the set $\{u \in S: u_1 < \frac{1}{4}, u_3 < \frac{1}{3}\}$.

However, if the aim is to reduce the Lagrange duality gap, then Proposition 11 does suggest that $\bar{d}$ is a good starting point. This clearly justifies the traditional use of the optimum Simplex multipliers as surrogate multipliers in integer programming [8].

3. Interior points and linear programming

Nemhauser and Widhelm [23] observed that a central step of the column-generation method of Dantzig and Wolfe involves determining, by linear programming, an interior point of a polyhedron bounded by a certain set of linear inequalities. From geometric considerations they therefore suggested a normalisation of the linear program which they believed would speed convergence. O'Neill and Widhelm [24] gave some computational evidence supporting this. Mattheiss and Widhelm [22] have extended this work.

The convergence of the algorithm of Section 4 depends on finding an interior point (relative to S) of a polyhedral subset of S, U say. It is also required that the point should be sufficiently distant from the boundary of U. Such an interior point could be determined in a variety of ways. The particular form of linear program suggested here is inspired by the work cited above.

A typical polytope encountered in the algorithm is of the form $U = \{u \in S: u^t g_j \geq 0 \; (j \in J)\}$, where for each j there exists a $u \in S$ with $u^t g_j \leq 0$. Assume initially that U is nonempty and let $u_0 \in U$. Consider the Euclidean distance of u_0 from the hyperplane $H_j = \{u: u^t g_j = 0\}$ in the plane of the equality $e^t u = 1$ which contains S. It is not difficult to show that this distance is $d_j(u_0) = u_0^t g_j / \gamma_j$ where $\gamma_j = \sqrt{(g_j^t g_j - (e^t g_j)^2/m)}$. An equivalent form for γ_j is, in fairly obvious notation, $\sqrt{\sum (g_{ji} - \bar{g}_j)^2}$. Then

Proposition 12. *If* $\gamma_j = 0$, *then* $g_j \leq 0$ *and* int $U = \emptyset$.

Proof. $\gamma_j = 0$ implies that $g_j = \xi e$ for some scalar ξ. But since there is a $u \in S$ with $u^t g_j \leq 0$, $\xi \leq 0$ and thus $g_j \leq 0$. However, $g_j \leq 0$ implies that there is no $u \in S$ with $u^t g_j > 0$ and hence int $U = \emptyset$.

Since in the algorithm of Section 4, int $U = \emptyset$ is the convergence criterion, and $g_j \leq 0$ gives a feasible solution to (P), it will be true that $\gamma_j > 0$ for all $j \in J$ if U is encountered in the algorithm.

The Euclidean distance of u_0 to the boundary of U is, in view of the above, $r(u_0) = \min\{d_j(u_0): j \in J\}$. It is obvious that $u_0 \in$ int U if and only if $r(u_0) > 0$. Let $r^* = \max\{r(u): u \in U\}$. It is now clear that int $U \neq \emptyset$ if and only if $r^* > 0$. The number r^* will be called the radius of U. It is not difficult to see that

$$
\begin{aligned}
& r^* = \max y, \\
& u^t g_j - \gamma_j y \geq 0 \quad (j \in J), \qquad \text{(I)} \\
& e^t u = 1, \quad u \geq 0.
\end{aligned}
$$

This is a linear program in the variables u, y. If int $U \neq \emptyset$, it is evident that an optimal solution $\bar{u}$ to (I) will be an interior point (relative to S) since $\bar{u} \in S$ and $\bar{u}^t g > 0$ ($j \in J$). This point will be called a centre of U. If U is empty, then the maximum value of y in (I) still exists, but is negative. In this case r^* will be defined to be the optimum value of y in (I). Therefore, whether U is nonempty or empty, int $U \neq \emptyset$ if and only if $r^* > 0$. It is evident that r^* will be finite provided J is nonempty. If J is empty, as it is only at the first iteration of Algorithm I below, then any $u \in S$ is a centre of U. A suitable choice for an interior point in this case might be based on the optimum Lagrange multiplier as described in Section 2.

In the generalised programming algorithm of Section 4, g_j will be $g(x)$ for some $x \in X$. The following fact, which follows immediately from the continuity of g, is needed.

Proposition 13. *$\gamma(x)$ is a continuous function of x on X.*

When $r^* > 0$, the linear program (I) yields one interior point $\bar{u}$ of U with $r(\bar{u}) = r^*$. Now the algorithm of Section 4 successively adds constraints at each iteration to the set defining U. These are added such that, if u is the interior point of U chosen at iteration i, then $u \notin$ int U at iteration $(i+1)$. It might be considered undesirable to move the point u too drastically from one iteration to the next, and the point $\bar{u}$ might be considered too far. The remainder of this section shows how, given the point $\bar{u}$ and a point $u \notin$ int U, a wider choice of interior point can be made which still retains the property of being sufficiently distant from the boundary of U.

It is not difficult to show that $r(u)$ is a concave function of u on any nonempty U. Suppose now that u' is any boundary point of U, so $r(u') = 0$. Then, for any given $0 < \theta \leq 1$, if $\theta \leq \beta \leq 1$, then

$$u = \beta\bar{u} + (1-\beta)u' \in \text{int } U$$

and

$$r(u) \geq \beta r(\bar{u}) + (1-\beta)r(u') = \beta r^* \geq \theta r^*.$$

Now, given $\bar{u}$ and any $u'' \notin$ int U, it is easy to find a boundary point u' of U. Such a point is $\alpha\bar{u} + (1-\alpha)u''$ where $0 \leq \alpha < 1$ is given by

$$\alpha = \max\{-u''^t g_j/(\bar{u}^t g_j - u''^t g_j)\colon u''^t g_j \leq 0, j \in J\}.$$

Note that the set of constraints considered in the determination of α are those which separate u'' from the interior of U. If, in Algorithm I below, u'' is taken as the interior point of U from the previous iteration, the number of these constraints will generally be few. Thus the determination of α will not be computationally burdensome. Observe also that $u = (1-\mu)\bar{u} + \mu u''$ where $\mu = (1-\alpha)(1-\beta)$. By an appropriately small choice of the "convergence parameter"

θ, almost any interior point between u' and $\bar{u}$ may be chosen without invalidating the proofs of convergence given in Section 4. It must be emphasised, of course, that whether or not a device of the above type is used, the linear program (I) must still be solved at each iteration. The usual linear programming re-optimisation procedures can, obviously, be used to minimise the computational effort in doing this as constraints are added from one iteration to the next.

4. A generalised programming algorithm

The algorithm below uses the ideas presented in Sections 2 and 3. First it is formally stated, then a discussion given followed by proofs of convergence.

Let $0 < \theta \leq 1$ be any scalar and $\bar{w}_0$ be any number not exceeding w^*. If necessary $\bar{w}_0$ could be taken as $-\infty$.

Algorithm I

Step 0: $i \leftarrow 1$, $U_1 \leftarrow S$.

Step 1: If the radius of U_i, $r_i^* \leq 0$ then stop.
Otherwise choose $u_i \in U_i$ such that $r(u_i) \geq \theta r_i^*$.

Step 2: If $X(u_i) \cap F(\bar{w}_{i-1}) = \emptyset$, $\bar{w}_i \leftarrow w(u_i)$ otherwise $\bar{w}_i \leftarrow \bar{w}_{i-1}$.
Determine any finite, nonempty $E_i \subseteq X(u_i) \cap F(\bar{w}_i)$.

Step 3: $U_{i+1} \leftarrow U_i \cap \{u \in R^m: u^t g(x) \geq 0 \text{ for all } x \in E_i\}$.

Step 4: $i \leftarrow i+1$. Go to Step 1.

At each iteration $\bar{w}_i$ is a lower bound for w^*. It is evident that U_i is of the form discussed in Section 3, and thus the computations of Step 1 can be effected by the method proposed there. In Step 2, the surrogate sub-problem $\min\{f(x): x \in X(u_i)\}$ is considered. If this problem has feasible solutions with $f(x) \leq \bar{w}_{i-1}$, it may not be necessary to solve it to complete optimality. However, if all $x \in X(u_i)$ have $f(x) > \bar{w}_{i-1}$, $\bar{w}_i$ will be increased to $w(u_i)$ and $E_i \subseteq M(u_i)$. In Step 3 the set of new linear constraints, or cutting planes [5], implied by the sub-problem solutions E_i are added to U_i and the next iteration is begun. It will be observed, that for finiteness in practice, the convergence criterion in Step 1, $r_i^* \leq 0$, would be replaced by $r_i^* < \delta$ for some suitably small constant δ.

The following gives a general convergence result.

Proposition 14. *Algorithm I either terminates, at iteration k, with $w^* = \bar{w}_k$ or else $\lim \bar{w}_i = w^*$ as $i \to \infty$.*

Proof. Observe that $\bar{w}_i$ is a nondecreasing sequence bounded above by w^* and thus must either terminate or tend to a limit. The nondecreasing nature of the sequence also implies that $E_j \subseteq F(\bar{w}_i)$ for all $j \leq i$, and hence $U_i \supseteq G^*(\bar{w}_i)$ for all i.

If $r_k^* \leq 0$ for any k, then int $U_k = \emptyset$ and so int $G^*(\bar{w}_k) = \emptyset$. But, from Corollary 2, this implies $\bar{w}_k \geq w^*$. However $\bar{w}_k \leq w^*$ and therefore $\bar{w}_k = w^*$.

Suppose now $r_i^* > 0$ for all i. At iteration i, $u_j \notin$ int U_i for all $j < i$. This follows, from Steps 2 and 3, since for any $x \in E_j$ int $U_i \subseteq \{u: u^t g(x) > 0\}$ and $u_j^t g(x) \leq 0$. Thus the Euclidean distance between u_i and u_j, $\|u_j - u_i\| \geq r(u_i) \geq \theta r_i^*$. Now as $i \to \infty$, $\{u_i\}$ must have a cluster point since it lies in the compact set S. Also r_i^* must tend to a limit, since it is an obviously nonincreasing sequence of reals bounded below by zero. But $\{u_i\}$ having a cluster point implies that there must exist an i and a $j < i$ such that $\|u_j - u_i\| < \epsilon$ for any $\epsilon > 0$, thus $\theta r_i^* < \epsilon$. This clearly implies $r_i^* \to 0$ as $i \to \infty$. If now lim $\bar{w}_i = w' < w^*$, then $\bar{w}_i \leq w'$ for all i and hence $U_i \subseteq G^*(w')$ for all i. But Corollary 2 then implies int $G^*(w') \neq \emptyset$ and thus contains an interior point u' which is at a nonzero distance r', say, from the boundary of $G^*(w')$. Hence $r_i^* \geq r'$ for all i, and thus lim $r_i^* \geq r' > 0$. This contradiction shows that lim $\bar{w}_i = w^*$.

This result can be improved slightly in some cases. For example

Proposition 15. *If w^* is attained on any open set of S, then either Algorithm I terminates or there is a k such that $\bar{w}_i = w^*$ for $i \geq k$.*

Proof. Let $W = \{u: w(u) = w^*\}$, so int $W \neq \emptyset$. Now $w(u) = w^* > \bar{w}_i$ implies $u \in$ int $G^*(\bar{w}_i)$ by Proposition 5. Thus, if the stated result is false, $U_i \supseteq G^*(\bar{w}_i) \supseteq W$ for all i. But then $r_i^* \to 0$ implies int $W = \emptyset$, which is a contradiction.

Corollary 4. *If (P) is a feasible discrete programming problem, then either Algorithm I terminates or there is a k such that $\bar{w}_i = w^*$ for all $i \geq k$.*

Proof. Follows directly from Proposition 4.

If X is finite, a particular case of discrete programming which includes the (bounded) pure-integer program, an even stronger assertion can be made.

Proposition 16. *If X is finite, Algorithm I terminates.*

Proof. Consider any iterations $i \neq j$ of the algorithm. Suppose $i > j$, then for all $x \in E_j$, $u_i^t g(x) > 0$ since $u_i \in$ int U_j. But since $u_i^t g(x) \leq 0$ for all $x \in E_i$, it follows that $E_i \cap E_j = \emptyset$. However, $E_i \neq \emptyset$ for each i, and thus the algorithm cannot continue for more than $|X|$ iterations.

The existence was shown, in Proposition 7, of a $v \leq 0$ in co$(G(w^*))$. It will now be shown that

Proposition 17. *Algorithm I either produces, or closely approximates, a $v \in$ co$(G(w^*))$ such that $v \leq 0$.*

Proof. Let $U_i = \{u \in S: u^t g_j \geq 0, j \in J_i\}$ and $V_i = \mathrm{co}\{g_j: j \in J_i\}$. V_i is clearly a subset of $\mathrm{co}(G(w^*))$. From the discussion of Section 3,

$$\begin{aligned} &r_i^* = \max y, \\ &u^t g_j - \gamma_j y \geq 0 \quad (j \in J_i), \\ &e^t u = 1, \quad u \geq 0. \end{aligned} \tag{I}$$

The linear programming dual of this can be written

$$\begin{aligned} &r_i^* = \min z, \\ &\sum \lambda_j g_j \leq \left(\sum \lambda_j \gamma_j\right) z e, \\ &\sum \lambda_j = 1, \quad \lambda_j \geq 0 \quad (j \in J_i) \end{aligned} \tag{II}$$

where the λ_j are simply the usual dual variables of (I), but normalised to sum to unity. If $r_i^* \leq 0$, then (II) clearly implies that there is a $v \in V_i$ such that $v \leq 0$ as required. Now if $r_i^* > 0$ for all i, then

$$\sum \lambda_j \gamma_j \leq \max\{\gamma_j: j \in J_i\} \leq M \quad \text{for some constant } M,$$

since from Proposition 13 $\gamma(x)$ is continuous on the compact set X. Thus there is a $v_i \in V_i$ such that $v_i \leq (M r_i^*) e$. Because $v_i \in \mathrm{co}(G(w^*))$, a compact set, $\{v_i\}$ has a cluster point. Since $r_i^* \to 0$, however, it follows that any cluster point v of $\{v_i\}$ must satisfy $v \leq 0$ and $v \in \mathrm{co}(G(w^*))$.

An undesirable feature of algorithms such as that given above is that apparently all the defining constraints of U_i must be retained indefinitely. Eaves and Zangwill [5] have, among others, given conditions for dropping constraints in algorithms of this type, but it does not appear that these results extend immediately to Algorithm I. However, the following proposition, in the spirit of this work, shows that potentially large numbers of the constraints defining U_i may be dropped provided the algorithm has made sufficient progress. The algorithm will still converge in the sense of the above results. To this end let $0 < \mu < 1$ be any scalar.

Proposition 18. *Let $x_j \in E_j$ and suppose that, at iteration i, $0 < r_i^* \leq \mu r_j^*$. Then the constraint $u^t g(x_j) \geq 0$ can be dropped from those defining U_i, provided this will not increase r_i^*.*

Proof. The only modification needed to the proof of Proposition 14 is to show that, if $r_i^* > 0$ for all i, then r_i^* still tends towards zero. Now the conditions of the proposition will ensure that $\{r_i^*\}$ is a nonincreasing sequence, and hence it must have a limit $\bar{r}$, say. Suppose $\bar{r} > 0$, then since $\mu < 1$ there is a j_0 such that $r_j^* < \bar{r}/\mu$ for all $j \geq j_0$. But since $r_i^* \geq \bar{r}$, $\mu r_j^* < r_i^*$. Thus no constraints will be dropped from E_j for any $j \geq j_0$. The argument used in Proposition 14 to show that $r_i^* \to 0$ can now be applied to the sequence $\{u_j: j \geq j_0\}$. This implies that, still, $r_i^* \to 0$ which contradicts $\bar{r} > 0$. Thus $\bar{r} = 0$ and the result is established.

The condition that the removal of $u^t g(x_j) \geq 0$ should not increase r_i^* is equivalent, of course, to the corresponding slack variable being basic in the optimal solution to (I). This may be easily checked.

5. A subgradient-type algorithm

Subgradient methods have proved effective in Lagrangean optimisation. For example Held et al. [14] discuss, and give experience in using, such methods. In surrogate duality the quasiconcave function w will not, in general, have a subgradient at all points. There are, however, quantities which correspond to subgradients, namely the normals to the hyperplanes of support of the level sets. Greenberg and Pierskalla [12] have christened these quasi-subgradients. Thus Algorithm II below could be called a quasi-subgradient method. A statement of this algorithm is followed by a discussion and proofs of convergence. The convergence is based on the condition of Polyak [25] for subgradient step-size control.

It is more convenient in this work to normalise the surrogate multipliers with the Euclidean, rather than the L_1, norm. Thus $\|x\|$, for $x \in R^m$, denotes the Euclidean norm throughout. The distance notation $d(x, y)$ will also be used for $\|x - y\|$. For any subset C of R^m, the distance from x to C is denoted by $d(x, C) = \inf\{d(x, y): y \in C\}$. It is well-known that $d(x, C) = 0$ if and only if x is in the closure, cl C, of C. The symbol $\hat{x}$ for $x \neq 0$, is the normed version of x, $x/\|x\|$.

The results of Section 2 must be modified to this normalisation, the change needed usually being to replace notions of convexity with those of spherical convexity. The convex set $B = \{x: \|x\| \leq 1, x \geq 0, x \neq 0\}$ is the portion of the unit m-disc, minus the origin, which lies in the non-negative orthant. $S' = \{x \in B: \|x\| = 1\}$ is the intersection of B with the unit m-sphere. S' corresponds, in the new normalisation, to S in the L_1-norm but is clearly non-convex. The algorithm generates a sequence of multipliers lying in S'.

Let $\{t_i\}$ be any sequence of non-negative reals such that $t_i \to 0$ and $\sum_{i=1}^{\infty} t_i = \infty$, e.g. $t_i = 1/i$. t_i is the step-size at iteration i. Sequences of step-sizes of this type were proposed by Polyak [25].

Algorithm II

Step 0: Choose any $u_1 \in S'$. Set $i \leftarrow 1$, $\bar{w} \leftarrow -\infty$.

Step 1: Determine any $x_i \in M(u_i)$. Let $g_i = g(x_i)$ and $f_i = f(x_i) = w(u_i)$. If $f_i > \bar{w}$, $\bar{w} \leftarrow f_i$. If $g_i \leq 0$, stop.

Step 2: Determine u_{i+1} as follows:

(a) $d_i' \leftarrow g_i - (u_i^t g_i)u_i$ and $d_i \leftarrow \hat{d}_i'$.

(b) $u_i' \leftarrow u_i + t_i d_i$.

(c) $u_i'' \leftarrow u_i' + p_i$, where

$$(p_i)_j = \begin{cases} -2(u_i')_j & \text{if } (u_i')_j < 0, \\ 0 & \text{otherwise.} \end{cases}$$

(d) $u_{i+1} \leftarrow \hat{u}_i''$.

Step 3: $i \leftarrow i+1$. Go to Step 1.

Firstly, it will be observed that the algorithm, as described, will not in general terminate, since no $g_i \leq 0$ will be discovered. In practice, some criterion such as no improvement in $\bar{w}$, the best value of $w(u_i)$ obtained, having been made over a group of k iterations could be used, where k is appropriately chosen. The same problem occurs in some subgradient methods.

Clarification of the geometrical nature of the multiplier update in Step 2 may be needed. This ensures that $u_{i+1} \in S'$ when $u_i \in S'$. Thus, since $u_1 \in S'$, $u_i \in S'$ for all i. The vector g_i is the quasi-subgradient direction. The direction of change for u_i is obtained in *Step* 2(a) by projecting this direction onto the tangent plane at u_i and normalising. Note that this normalisation is always possible since $d_i' = 0$ implies $g_i \leq 0$. Then u_i is modified in *Step* 2(b) by a step of length t_i in this direction. This ensures that u_i' is outside the unit sphere, which fact is required in the convergence proofs. Note that, computationally, u_i' is simply a linear combination of u_i and g_i. The point u_i' may, however, also lie outside the non-negative orthant of R^m. If so, it is reflected back into this orthant in Step 2(c). Thus $u_i'' \geq 0$, but u_i'' will not lie in S'. Thus in Step 2(d) it is normed to give $u_{i+1} \in S'$. It will be shown that $\|u_i''\| = (1+t_i^2)^{1/2}$, so this step can be readily accomplished.

It will be noted that each one-constraint sub-problem is solved, in Step 1, to optimality. In practice this is not absolutely essential, and a sub-optimum solution might sometimes be used as in Algorithm I. The assumption is made since the result for the convergence of the multiplier sequence, in Proposition 22, depends on it.

The following preliminary results establish properties of the sequence $\{u_i\}$ which will be needed later.

Proposition 19. $\|u_i''\| = (1+t_i^2)^{1/2}$.

Proof.

$$\begin{aligned} \|u_i''\|^2 &= \|u_i'\|^2 + 2p_i^t u_i' + \|p_i\|^2 \\ &= \|u_i'\|^2 \quad (\text{since } 2p_i^t u_i' = -\|p_i\|^2) = \|u_i\|^2 + 2t_i u_i^t d_i + t_i^2 \|d_i\|^2 \\ &= 1 + t_i^2 \quad (\text{since } \|u_i\| = \|d_i\| = 1,\ u_i^t d_i = 0). \end{aligned}$$

Proposition 20. *If* $u \in B$, *then*

$$\|u_{i+1} - u\|^2 \leq \|u_i - u\|^2 - t_i(2u_i^t d_i - t_i).$$

Proof.

$$\|u_i' - u\|^2 = \|u_i - u\|^2 - t_i(2u_i^t d_i - t_i),$$
$$\|u_i'' - u\|^2 = \|u_i' - u\|^2 - 2p_i^t u \leq \|u_i' - u\|^2,$$
$$\|u_{i+1} - u\|^2 = \|u_i'' - u\|^2 - \lambda(2+\lambda) + 2\lambda u^t u_{i+1} \quad (\text{where } \lambda = \|u_i''\| - 1 \geq 0)$$
$$\leq \|u_i'' - u\|^2 - \lambda(2+\lambda) + 2\lambda \leq \|u_i'' - u\|^2.$$

Combining these inequalities gives the result.

Corollary 5. $\|u_{i+1} - u_i\| \leq t_i$.

Proof. $u_i \in B$ and $u_i^t d_i = 0$.

Corollary 6. If $u \in B$ and $2u^t \hat{g}_i \geq t_i$, then

$$\|u_{i+1} - u\|^2 \leq \|u_i - u\|^2 - t_i(2u^t \hat{g}_i - t_i) \leq \|u_i - u\|^2.$$

Proof.

$$u^t d_i' = u^t g_i - (u^t u_i)(u_i^t g_i) \geq u^t g_i \geq \tfrac{1}{2} t_i \|g_i\| \geq 0.$$

Also $\|d_i'\|^2 = \|g_i\|^2 - (u_i^t g_i)^2$ so $\|d_i'\| \leq \|g_i\|$. Thus

$$u^t d_i = u^t \hat{d}_i' \geq u^t d_i' / \|g_i\| \geq u^t g_i / \|g_i\| = u^t \hat{g}_i \geq \tfrac{1}{2} t_i.$$

The following lemma is needed for Proposition 21.

Lemma 2. *If* $u \in B$, *then* $\liminf u^t g_i \leq 0$.

Proof. Note first that the sequence $\{u^t g_i\}$ is bounded, since g is bounded. Suppose then, to the contrary, that $u^t \hat{g}_i > \delta > 0$ for all $i \geq i_0$. Then since $t_i \to 0$, for all $i \geq i_1$, say, $t_i \leq \delta$. Denoting $d(u_i, u)$ by d_i, Corollary 6 shows that $d_i^2 - d_{i+1}^2 \geq \delta t_i$ for all $i \geq k = \max(i_0, i_1)$. Summing such inequalities from k to n,

$$d_k^2 - d_n^2 \geq \delta \sum_{i=k}^{n} t_i.$$

Now, letting $n \to \infty$, the condition $\sum t_i = \infty$ implies that $d_k^2 - d_n^2 \to \infty$, which is obviously impossible. This contradiction establishes the result.

Proposition 21. $\limsup w(u_i) = w^*$.

Proof. Let $u \in B$ be such that $w(u) > w^* - \epsilon$ where $\epsilon > 0$ is arbitrary. Now suppose that $w(u_i) \leq w^* - \epsilon$ for all $i \geq i_0$, then $u^t g(x_i) > 0$ for all such i. But from Lemma 2, this implies that $\liminf u^t g(x_i) = 0$. Since the x_i lie in a compact set, a subsequence x_r can be chosen with $x_r \to x^*$ and $\lim u^t g(x_r) = 0$. Then, from continuity of g, $u^t g(x^*) = 0$ and hence $f(x^*) > w^* - \epsilon$. But the l.s.c. of f implies that

$$f(x^*) \le \limsup f(x_r) = \limsup w(u_i) \le w^* - \epsilon.$$

This is a contradiction so for all i_0 there is an $i \ge i_o$ such that $w(u_i) > w^* - \epsilon$. The arbitrary choice of ϵ then gives the conclusion.

Proposition 21 gives a form of convergence for the function values. It is natural then to enquire in what sense, if any, the sequence of multipliers converges. Proposition 22 gives such a convergence property. A few preliminary definitions and lemmas are required.

For any $\epsilon > 0$, let $W(\epsilon) = \{u \in B: w(u) > w^* - \epsilon\}$. It is clear from analogy with Corollary 1 that $W(\epsilon)$ is convex and contains an interior point relative to R^m. Let $W^* = \bigcap\{\text{cl } W(\epsilon): \epsilon > 0\} \cap S'$. W^* is nonempty since the sets $W(\epsilon)$ are nested and each set cl $W(\epsilon)$ contains points of S'. Observe that for any $u \in W^*$, every neighbourhood of u contains a point which is ϵ-maximal for w, whatever ϵ. It will be shown in Proposition 22 that all cluster points of $\{u_i\}$ lie in the set W^*, but the following lemmas are proved first.

Lemma 3. *If $C \subseteq R^m$ is a convex set, with complement $\bar{C}$, and for $x \in R^m$, $r(x) = d(x, \bar{C})$, then r is a concave function on C.*

Proof. Let $x_1, x_2, \in C$ and $0 \le \theta \le 1$, so $x = \theta x_1 + (1-\theta)x_2 \in C$. Let $r_1 = r(x_1)$, $r_2 = r(x_2)$ and $r = \theta r_1 + (1-\theta) r_2$. Suppose $y \in R^m$ is such that $d(x, y) < r$. Then $y_1 = x_1 + r_1(y-x)/r \in C$, since $d(x_1, y_1) < r_1$ and similarly $y_2 = x_2 + r_2(y-x)/r \in C$. Thus $y = \theta y_1 + (1-\theta) y_2 \in C$. Thus for all y such that $d(x, y) < r$, $y \in C$ and hence $r(x) \ge \theta r_1 + (1-\theta) r_2$ and r is concave.

Lemma 4. *Let $C \subseteq R^m$ be a bounded convex set with interior and $\alpha > 0$ be any such that $C' = \{x \in C: r(x) \ge \alpha\}$ is nonempty. Then, if $\delta(\alpha) = \sup\{d(x, C'): x \in C\}$, there is a constant $K \ge 1$ such that $\alpha \le \delta(\alpha) \le K\alpha$.*

Proof. For any x in the boundary of C it is clear that $d(x, y) \ge \alpha$ for $y \in C'$. Let $\{x_n\} \subseteq C$ be such that $x_n \to x$, then for all $y \in C'$

$$d(x_n, y) \ge d(x, y) - d(x_n, x) \ge \alpha - d(x_n, x),$$

and hence $d(x_n, C') \ge \alpha - d(x_n, x)$. Letting $n \to \infty$ it follows that $\delta(\alpha) \ge \alpha$.

Now $\delta(\alpha) < \beta$ if, for each $x \in C$, there is a $y \in C'$ such that $d(x, y) < \beta$. Let $z = (1-\theta)x + \theta y \in C$ for any $0 \le \theta \le 1$. Since r is concave

$$r(z) \ge (1-\theta) r(x) + \theta r(y) \ge \theta r(y) \ge \theta\alpha.$$

Also $d(x, z) = \theta d(x, y) < \theta\beta$. Thus $\delta(\theta\alpha) < \theta\beta$ since for all $x \in C$ there is a z such that $r(z) \ge \theta\alpha$ and $d(x, z) < \theta\beta$. This implies that $\delta(\theta\alpha) \le \theta\delta(\alpha)$ for any $0 \le \theta \le 1$. Let $\alpha_0 > 0$ be the supremum of α such that C' is nonempty. It is finite since C is bounded. It will also be attained since r, being concave, is continuous on the interior of C and $r(x) = 0$ for x in the boundary of C. Thus for any $0 \le \alpha \le \alpha_0$, the above inequality on δ implies that $\delta(\alpha) \le \alpha\delta(\alpha_0)/\alpha_0$. Letting $\delta(\alpha_0)/\alpha_0 = K$, and noting that $\delta(\alpha_0) \ge \alpha_0$, the lemma is proved.

Proposition 22. *Every cluster point of the sequence $\{u_i\}$ produced by Algorithm II lies in W^*.*

Proof. For any fixed $\epsilon > 0$, let $C = W(\epsilon)$. Let α, C' and $\delta(\alpha)$ be as defined in Lemma 4.

Since $t_i \to 0$, for any $i \ge i_0$ say, $t_i \le \alpha$. Also, from Proposition 21 there is a $k \ge i_0$ such that $u_k \in C$. Let q be the smallest $i > k$ such that $u_q \notin C$, and let $s \ge q$ be such that $u_i \notin C$ for all $q \le i \le s$. Now if $u \in C'$ is arbitrary, then $u^t\hat{g}_i \ge \alpha$ for all such i. This is because the half-space $H = \{x \in R^m: x^t\hat{g}_i \le 0\}$ does not meet C and therefore $u^t\hat{g}_i = d(u, H) \ge r(u) \ge \alpha$. Thus, from Corollary 6, $d(u_s, u) \le d(u_q, u)$. But, since $u_s \notin C$ and $u \in C'$, there will exist a point $x \in \text{cl}\, C$ on the line segment u, u_s such that $d(u_s, u) = d(u_s, x) + d(u, x)$ and $d(u, x) = \alpha$. Therefore

$$d(u_s, x) = d(u_s, u) - \alpha \le d(u_q, u) - \alpha.$$

However,

$$d(u_q, u) \le d(u_q, u_{q-1}) + d(u_{q-1}, u) \le t_{q-1} + d(u_{q-1}, u),$$

from Corollary 5, and hence $d(u_q, u) \le \alpha + d(u_{q-1}, u)$. Therefore $d(u_s, x) \le d(u_{q-1}, u)$. Note that x depends on u. But $u \in C'$ is arbitrary. Now, since $u_{q-1} \in C$, for any $\beta > \delta(\alpha)$ there is a $u \in C'$ such that $d(u_{q-1}, u) < \beta$. Thus there is an $x \in \text{cl}\, C$ such that $d(u_s, x) < \beta$. Thus it follows that $d(u_s, C) \le \delta(\alpha)$. Now using Lemma 4, it is evident that $d(u_i, C) \le K\alpha$ for all $i \ge k$, where K is a positive constant. Letting α approach zero, this shows that $d(u^*, C) = 0$ for any cluster point u^* of $\{u_i\}$. But since $C = W(\epsilon)$, $\epsilon > 0$ is arbitrary and $u_i \in S'$ for all i, the proof is complete.

Corollary 7. *If w^* is attained and w is continuous at every u such that $w(u) = w^*$, then every cluster point u^* of the sequence $\{u_i\}$ generated by Algorithm II satisfies $w(u^*) = w^*$.*

Proof. It is clear in this case that $W^* = \{u \in S': w(u) = w^*\}$.

Corollary 8. *If w has a unique maximising point u^*, then $u_i \to u^*$.*

Proof. In this case $W^* = \{u^*\}$.

Corollaries 7 and 8 give conditions under which the cluster points will maximise w. However, in general it can only be guaranteed that they are infinitesimally close to maximal points, if w^* is attained, or almost maximal points if it is not attained.

It was mentioned, in the discussion following Algorithm II, that in practice some, perhaps arbitrary, convergence criterion is needed to ensure finiteness. One possibility was suggested there. It might be asked whether there is any way, even in principle, of determining how much progress towards convergence the algorithm has made. In fact there is such a way. Let i_0 be arbitrary. Then, at any iteration $i \geq i_0$, a linear program of the form of (I) of Section 4 may be solved, where J_i is now defined to be the set $\{i_0, \dots, i\}$ and γ_j to be $\|g_j\|$. The numerical smallness of r_i^*, the optimum value of this linear program, can be used as in Algorithm I to determine the closeness to convergence. The justification for this procedure is provided by the following proposition.

Proposition 23. *If r_i^* is as defined above, then $r_i^* \to 0$ as $i \to \infty$.*

Proof. Otherwise there exists $\epsilon > 0$ such that $r_i^* \geq \epsilon$ for all $i \geq i_0$, since r_i^* is nonincreasing. Thus there exists $u_i \in S$ such that $u_i^t \hat{g}_j \geq \epsilon$ for all $j \in J_i$. Let u^* be any cluster point of $\{u_i\}$, then it follows that $u^{*t}\hat{g}_j \geq \epsilon$ for all $j \geq i_0$. For, let subsequence $u_r \to u^*$. Then for each $j \geq i_0$ and all $r \geq j$,

$$u^{*t}\hat{g}_j = (u^* - u_r)^t \hat{g}_j + u_r^t \hat{g}_j \geq (u^* - u_r)^t \hat{g}_j + \epsilon.$$

Letting $r \to \infty$ establishes the fact. But $u^* \in S \subseteq B$ and hence, from the proof of Lemma 2, $\liminf u^{*t}\hat{g}_j \leq 0$, which contradiction proves the proposition.

The linear programming duality argument used in the proof of Proposition 17 may be employed to show further that Algorithm II can be made to produce, or approximate, a vector v satisfying the optimality condition of Proposition 7.

In practice, of course, it may be doubtful whether the additional labour involved in solving the linear program would be justified in this algorithm. If linear programming is to be used, Algorithm I would seem preferable. However, Proposition 23 does open up the possibility of hybrid versions of Algorithms I and II.

6. Example

The algorithms of Sections 4 and 5 will be demonstrated on the following simple integer-programming problem:

$$\text{(P)} \qquad \begin{aligned} f^* = \min \quad & 20 - 2x_1 - 11x_2, \\ \text{subject to} \quad & 10x_1 + 2x_2 - 15 \leq 0, \\ & x_1 + 6x_2 \; - 9 \leq 0, \\ & -x_1 + x_2 \qquad \leq 0, \end{aligned}$$

$$x \in X = \{(x_1, x_2)\colon 0 \leq x_j \leq 100 \text{ and integer } (j = 1, 2)\}.$$

Note that the variable upper bounds are imposed merely to ensure that X satisfies the compactness assumption.

Both algorithms will be started with multipliers obtained from the maximisation of the Lagrangean dual. In this case, of course, this simply involves solving the usual linear-programming relaxation of (P). The optimal Lagrange multiplier can then be shown to be $d = (\frac{13}{12}, 0, \frac{106}{12})$ with $L^* = 3\frac{3}{4}$. The results of calculations in both algorithms are given to four decimals only, but were originally to greater accuracy.

6.1. Algorithm I

The algorithm is started with $u_1 = \bar{d} = (0.1092, 0, 0.8908)$. The convergence parameter θ is set to the relatively small value 0.1, since $\bar{d}$ should be near a maximising point of w. In fact a much smaller value still would work better in this particular example, but the value 0.1 was chosen for illustrative purposes. The quantities α, β, μ are as defined in Section 3, where in this example $\beta = \theta$ at each iteration. Also, since $L^* \leq w^*$, $\bar{w}_0$ can be set to $L^* = 3\frac{3}{4}$.

Iteration 1: $r_1^* = \max y$ such that $\bar{u}_1 + \bar{u}_2 + \bar{u}_3 = 1$ and $\bar{u}_1, \bar{u}_2, \bar{u}_3 \geq 0$. Since $r_1^* = \infty$, arbitrarily choose $u_1 = (0.1092, 0, 0.8908) \in U_1 = S$. The surrogate sub-problem is $w(u_1) = \min 20 - 2x_1 - 11x_2$ subject to $0.2017x_1 + 1.1092x_2 - 1.6387 \leq 0$ and $x \in X$. This has optimal solution $x_1 = 2, x_2 = 1$ and $w(u_1) = 5$. Thus $\bar{w}_1 \leftarrow 5$, $E_1 = \{(2, 1)\}$, $g(E_1) = \{(7, -1, -1)\}$. Thus $U_2 \leftarrow S \cap \{(7, -1, -1)u \geq 0\}$.

Iteration 2: $r_2^* = \max y$ such that $7\bar{u}_i = \bar{u}_2 - \bar{u}_3 - 6.5320y \geq 0$, $\bar{u}_1 + \bar{u}_2 + \bar{u}_3 = 1$, $\bar{u}_1, \bar{u}_2, \bar{u}_3 \geq 0$. So $r_2^* = 1.0717$ when $\bar{u} = (1, 0, 0)$. Then $\alpha = 0.1260/(7 + 0.1260) = 0.0177$. Thus $\mu = (1 - 0.1)(1 - 0.0177) = 0.8841$. Thus $u_2 = (1 - \mu)\bar{u} + \mu u_1 = (0.2125, 0, 0.7875)$. The surrogate sub-problem is $w(u_2) = \min 20 - 2x_1 - 11x_2$ such that $1.3375x_1 + 1.2125x_2 - 3.1875 \leq 0$. This has solution $x = (0, 2)$, and $w(u_2) = -2$. Thus $\bar{w}_2 \leftarrow 5$, $E_2 = \{(0, 2)\}$ and $g(E_2) = \{(-11, 3, 2)\}$. Hence $U_3 = U_2 \cap \{(-11, 3, 2)u \geq 0\}$.

Iteration 3: $r_3^* = \max y$ subject to $-11\bar{u}_1 + 3\bar{u}_2 + 2\bar{u}_3 - 11.0454y \geq 0$, $7\bar{u}_1 - \bar{u}_2 - \bar{u}_3 - 6.5320y \geq 0$, $\bar{u}_1 + \bar{u}_2 + \bar{u}_3 = 1$ and $\bar{u} \geq 0$. This gives $r_3^* = 0.0556$ when $\bar{u} = (0.1704, 0.8296, 0)$. Then $\alpha = 0.7625/(0.7625 + 0.6143) = 0.5538$, so $\mu = (1 - 0.1)(1 - 0.5538) = 0.4016$. Therefore $u_3 = (1 - \mu)\bar{u} + \mu u_2 = (0.1873, 0.4965, 0.3162)$. The sub-problem is $w(u_3) = \min 20 - 2x_1 - 11x_2$ subject to $2.0533x_1 + 3.6696x_2 - 7.2778 \leq 0$. This has the solution $x_1 = 1$, $x_2 = 1$ and $w(u_3) = 7$. Then $\bar{w}_3 \leftarrow 7$, $E_2 = \{(1, 1)\}$ and $g(E_2) = \{(-3, -2, 0)\}$. Thus there exists a feasible solution to (P) in $M(u_3)$, and the algorithm can be stopped here. This problem, therefore, has no surrogate duality gap and $f^* = w^* = w(u_3) = 7$.

Observe that the criterion $r_4^* \leq 0$ could equally be used to stop the algorithm after defining $U_4 = U_3 \cap \{(-3, -2, 0)u \geq 0\}$.

6.2. Algorithm II

The algorithm is started with $u_1 = \hat{d} = (0.1217, 0, 0.9926)$. The sequence $\{t_i\}$ is taken as $\{0.1/i\}$. As with θ in Section 6.1, however, a sequence of rather smaller step-sizes would actually work better in this example.

Iteration 1: $u_1 = (0.1217, 0, 0.9926)$. The surrogate problem is then $w(u_1) = \min 20 - 2x_1 - 11x_2$ subject to $0.2247x_1 + 1.2360x_2 - 1.8259 \leq 0$. This has optimal solution $x = (2, 1)$ and $w(u_1) = 5$, as in Section 6.1. Then $g_1 = g(2, 1) = (7, -1, -1)$ and $u_1^t g_1 = -0.1405$. Thus $d_1' = (7, -1, -1) + 0.1405(0.1217,\ 0,\ 0.9926)$ and $d_1 = \hat{d}_1' = (0.9828,\ -0.1401,\ -0.1205)$. Now $u_1' = u_1 + (1/10)d_1 = (0.2200, -0.0140, 0.9805)$ and $u_1'' = (0.2200, 0.0140, 0.9805)$. Hence $u_2 = \hat{u}_1'' = (0.2189, 0.0139, 0.9756)$.

Iteration 2: $u_2 = (0.2189, 0.0139, 0.9756)$. The surrogate sub-problem is $w(u_2) = \min 20 - 2x_1 - 11x_2$ such that $1.2274x_1 + 1.4971x_2 - 3.4092 \leq 0$. The optimal solution to this is $x = (0, 2)$ and $w(u_2) = -2$. Thus $g_2 = g(0, 2) = (-11, 3, 2)$ and $u_2^t g_2 = -0.4150$. Then $d_2' = (-11, 3, 2) + 0.4150(0.2189, 0{,}0139, 0.9756) = (-10.9092, 3.0058, 2.4049)$. Therefore $d_2 = \hat{d}_2' = (-0.9430, 0.2598, 0.2079)$ and $u_2'' = u_2' = u_2 + (1/20)d_2 = (0.1718, 0.0269, 0.9860)$. Now $u_3 = \hat{u}_2'' = (0.1716, 0.0269, 0.9848)$.

Iteration 3: $u_3 = (0.1716, 0.0269, 0.9848)$. The surrogate problem is $w(u_3) = \min 20 - 2x_1 - 11x_2$ subject to $0.7853x_1 + 1.4893x_2 - 2.8153 \leq 0$. This has optimal solution $x = (1, 1)$ and $w(u_3) = 7$. Therefore $g_3 = g(1, 1) = (-3,\ -2, 0)$. Since $g_3 \leq 0$, stop.

This gives the same value, 7, for w^* and the same optimal feasible solution, (1, 1), for (P) as found in Section 6.1. However, the optimal surrogate multiplier (0.1716, 0.0269, 0.9848) is rather different from that obtained using Algorithm I, i.e. (0.1873, 0.4965, 0.3162). It could, of course, have been stated a priori that the optimal multiplier would not be unique, since (P) is a discrete programming problem, even after normalisation.

7. Discussion and conclusions

In this paper some properties of the surrogate dual of a mathematical programming problem have been examined. Two algorithms for calculating strongest surrogate constraints which exploit these properties have been described, and these illustrated on a small example. No computing experience with either of the methods has yet been obtained, but it is hoped they may provide a useful adjunct to Lagrangean methods in mathematical programming. Further work, both theoretical and empirical, needs to be done to assess and compare these two methods. In particular the issue of rate of convergence needs examining for both algorithms, and the problem of step-size selection for Algorithm II requires investigation.

One obvious area of applicability is integer-programming, where surrogate constraints are already successfully established [8]. There is one small problem in this particular application which might be mentioned here, that of rounding error. In surrogate duality real-valued multipliers are used to combine the constraints, and in the presence of round-off there might be difficulty in deciding whether a particular integer solution is feasible in the surrogate constraint. Ways can be devised to minimise this difficulty, but it will be no more serious than in many other integer-programming methods, for example some cutting-plane algorithms and certain branch-and-bound approaches.

Finally it may be noted that the work of Luenberger [20] shows that the algorithms also provide one approach to general quasi-convex programming. The relationship of Algorithm I to quasi-convex programming exactly parallels the use of generalised programming for convex programs, see for example [21].

8. Note

Other recent research relevant to the content of this paper is given in references [1] to [3] and [15] to [19]. In particular, [17] gives some empirical evidence on the ability of surrogates to close the Lagrange duality gap in integer-programming. I am grateful to the editors and referees for drawing my attention to this work.

References

[1] K. Banerjee, "Generalized Lagrange multipliers in dynamic programming", Research report ORC 71-12, Operations Research Center, University of California, Berkeley, CA (1971).

[2] S. Bolwell and M.H. Karwan, "Efficient use of surrogate duality in 0–1 integer programming", Industrial Engineering report, State University of New York at Buffalo, NY (1979).

[3] S. Bolwell and M.H. Karwan, "A hybrid surrogate dual-cutting plane approach to integer programming", Industrial Engineering report, State University of New York at Buffalo, NY (1979).

[4] G.H. Bradley, "Transformation of integer programs to knapsack problems", *Discrete Mathematics* 1 (1971) 29–45.

[5] B.C. Eaves and W.I. Zangwill, "Generalised cutting plane methods", *SIAM Journal of Control* 9 (1971) 529–542.

[6] A.M. Frieze, "Bottleneck linear programming", *Operational Research Quarterly* 26 (1975) 871–874.

[7] A.M. Geoffrion, "Duality in nonlinear programming", *SIAM Review* 13 (1971) 1–37.

[8] F. Glover, "Surrogate constraints", *Operations Research* 16 (1968) 741–749.

[9] F. Glover, "Surrogate constraint duality in mathematical programming", *Operations Research* 23 (1975) 434–451.

[10] H.J. Greenberg and W.P. Pierskalla, "Surrogate mathematical programs", *Operations Research* 18 (1970) 924–939.

[11] H.J. Greenberg and W. P. Pierskalla, "A review of quasi-convex functions", *Operations Research* 19 (1971) 1553–1570.

[12] H.J. Greenberg and W.P. Pierskalla, "Quasi-conjugate functions and surrogate duality", *Cahiers du Center d'Etudes de Recherche Operationelle* 15 (1973) 437–448.

[13] H.J. Greenberg, "The generalised penalty function surrogate model", *Operations Research* 21 (1973) 162–178.

[14] M. Held, P. Wolfe and H.P. Crowder, "A validation of subgradient optimisation", *Mathematical Programming* 6 (1974) 62–88.

[15] M.H. Karwan, "Surrogate constraint duality and extensions in integer programming", Ph.D. dissertation, Georgia Institute of Technology (1976).

[16] M.H. Karwan and R.L. Rardin, "Surrogate dual multiplier search procedures", Industrial and *Systems Engineering Report Series No. J*-76-29, Georgia Institute of Technology, GA (1976).

[17] M.H. Karwan and R.L. Rardin, "Some relationships between Lagrangian and surrogate duality in integer linear programming", *Mathematical Programming* 17 (1979) 320–334.

[18] M.H. Karwan and R.L. Rardin, "Surrogate duality in a branch-and-bound procedure", *Naval Research Logistics Quarterly* (to appear).

[19] M.H. Karwan and R.L. Rardin, "Searchability of the composite and multiple surrogate dual functions", *Operations Research* (to appear).

[20] D.G. Luenberger, "Quasi-convex programming", *SIAM Journal of Applied Mathematics* 16 (1968) 1090–1095.

[21] T.L. Magnanti, J.F. Shapiro and M.H. Wagner, "Generalised programming solves the dual", *Management Science* 22 (1976) 1195–1203.

[22] T.H. Mattheiss and W.B. Widhelm, "The generalized slack variable linear program", *Management Science* 23 (1977) 859–871.

[23] G.L. Nemhauser and W.B. Widhelm, "A modified linear program for columnar methods in mathematical programming", *Operations Research* 19 (1971) 1051–1060.

[24] R.P. O'Neill and W.B. Widhelm, "Computational experience with normed and non-normed column generation procedures in nonlinear programming", *Operations Research* 23 (1975) 372–382.

[25] B.T. Polyak, "A general method of solving extremal problems", *Doklady Akademii Nauk SSSR* 174 (1967) 33–36.