H. Paul Williams

An alternative form of the value function of an integer programme

Discussion paper [or working paper, etc.]

Original citation:
Williams, H. Paul (1988) *An alternative form of the value function of an integer programme.* Faculty of Mathematical Studies working papers, OR16. Faculty of Mathematical Studies, University of Southampton, Southampton, UK.
This version available at: http://eprints.lse.ac.uk/31735/

Faculty of Mathematical Studies, University of Southampton

Available in LSE Research Online: June 2011

UNIVERSITY OF SOUTHAMPTON

Faculty of Mathematical Studies

preprint series

OR16

AN ALTERNATIVE FORM FOR THE VALUE FUNCTION OF AN INTEGER PROGRAMME

By

H.P. Williams

October 1988

Southampton SO9 5NH England

AN ALTERNATIVE FORM FOR THE VALUE FUNCTION OF AN INTEGER PROGRAMME

H.P. Williams

University of Southampton, U.K.

Abstract

The value function of an Integer Programme is the optimal objective value expressed as a function of the right-hand-side coefficients. A method of expressing the value function is described which involves non-negative "correction terms" applied to the right-hand-side coefficients, which must satisfy a series of linear congruences and are restricted to a finite set of values.

A method of devising such a form of the value function is given, based on the successive elimination of integer variables. The method uses results from Elementary Number Theory such as the Generalised Chinese Remainder Theorem.

Although the method solves an Integer Programme as a function of the right-hand-side it could be specialised to specific right-hand-sides giving a new integer programming algorithm.

Key words: Integer Programming, Linear Congruences, Chinese Remainder Theorem, Duality, Sensitivity Analyses.

1. INTRODUCTION

We consider the Pure Integer Programming Model in the form

$$\begin{aligned} &\text{Minimise} \quad \underline{c}\,\underline{x} \\ &\text{Subject to:} \quad A\underline{x} \geq \underline{b} \\ &\qquad\qquad \underline{x} \geq \underline{0} \ \text{ and integer} \end{aligned} \tag{1.1}$$

We assume throughout that all coefficients are integral unless stated otherwise. The <u>value function</u> $G(\underline{b})$ is the optimal value of the objective as a function of the right-hand-side $\underline{b}$.

It has been shown by Blair and Jeroslow [1] that there is a function $F(\underline{b})$ such that

(i) (1.1) is feasible so long as $F(\underline{b}) \leq 0$

(ii) F and G are constructed from $\underline{b}$ by a finite number of applications of the operations of taking non-negative linear combinations of the b_i , integer round-up, and maximum.

1.1 An Example

The value function of

$$\begin{aligned} &\text{Minimise} \quad 18x_1 - 3x_2 \\ &\text{Subject to:} \quad 4x_1 + x_2 \geq b_1, \\ &\qquad\qquad 9x_1 - 2x_2 \geq b_2 \\ &x_1, x_2 \geq 0 \ \text{ and integer} \end{aligned} \tag{1.2}$$

is

$$\text{Max}\left\{ 3\left\lceil \frac{b_2}{2} \right\rceil,\ 3\left\lceil \frac{1}{2}b_2 + \frac{3}{2}\left\lceil \frac{b_2}{9} \right\rceil \right\rceil,\ 3\left\lceil \frac{b_2}{2} + \frac{3}{2}\left\lceil \frac{1}{17}(2b_1+b_2) \right\rceil \right\rceil \right\} \tag{1.3}$$

where "$\lceil\ \rceil$" represents the integer round-up operation. (1.2) is feasible for all b_1, b_2 hence in this case $F(\underline{b})$ is vacuous.

It is straightforward to show that the "maximum" operations can all be moved to the "top level" as in (1.3.)

Functions of the form 1.3 are known as Gomory functions since the fact that value functions take this form follows from applying Gomory's cutting planes algorithm (Gomory [3]). The expressions resulting from excluding the maximum operation are known as Chvátal [2] functions.

For a <u>specific</u> value of $\underline{b}$ one of the Chvátal functions within the maximum expression will provide the optimal value of the integer programme. This expression may also provide the optimal value within some neighbourhood of $\underline{b}$. We will call this expression the <u>perturbation function</u> corresponding to the solution.

For example if $b_1 = 25$, $b_2 = 20$ (1.3) becomes $\mathrm{Max}\{30, 45, 54\} = 54$.

The Chvátal function $3\left\lceil \frac{b_2}{2} + \frac{3}{2}\left\lceil \frac{1}{17}(2b_1+b_2)\right\rceil\right\rceil$ also represents the optimal value of (1.2) if, for b_1 fixed at 25, b_2 is altered in the range $-\infty$ to 63 or for b_2 fixed at 20 b_1 is altered in the range 18 to ∞ . Hence there is interest in studying value and perturbation functions from the point of view of sensitivity analysis in integer programming.

It is instructive to compare the value and function for an integer programme with the well known form it takes in linear programming derived from the dual.

1.2 The Value Function of a Linear Programme.

For a linear programme expressed in the form of the relaxation (dropping the integrality stipulation) of (1.1) the value function $G(\underline{b})$ and feasibility function $F(\underline{b})$ take the same form as that for the integer case if the "$\lceil\ \rceil$" operation is dropped. For example the value function of

$$\begin{aligned} \text{Minimise} \quad & 18x_1 - 3x_2 \\ \text{Subject to:} \quad & 4x_1 + x_2 \geq b_1 \\ & 9x_1 - 2x_2 \geq b_2 \\ & x_1, x_2 \geq 0 \end{aligned} \tag{1.4}$$

is

$$\text{Max}\left\{ \frac{3b_2}{2},\ 2b_2,\ \frac{9}{17} b_1 + \frac{30}{17} b_2 \right\} \tag{1.5}$$

The coefficients of b_1 and b_2 in each of the three expressions in brackets arise from the <u>vertices</u> of the <u>dual</u> to 1.4. Should a feasibility condition $F(\underline{b}) \leq 0$ arise (which is not the case in this example) then the corresponding coefficients of the b_i arise from the <u>extreme</u> <u>rays</u> of the dual polytope. The <u>dual</u> <u>values</u> (coefficients of b_i) in the expression corresponding to the optimum do, of course, give <u>shadow</u> <u>prices</u> much used in sensitivity analysis so long as they apply over a neighbourhood.

The value function of an integer programme provides the only really satisfactory corresponding dual. Wolsey [6] surveys a number of different structures it might take as well as proving the corresponding duality results.

Blair and Jeroslow refer to the expressions obtained from Gomory and Chvátal functions by dropping the "$\lceil\ \rceil$" operation as the <u>carriers</u> of the corresponding functions. They show that the carrier of the value function of an integer programme provides the value function of its linear programming relaxation as is instanced by examples (1.3) and (1.5) above.

1.3 Alternative Forms for the Value Function of an Integer Programme

The Gomory function form of a value function is not a particularly convenient one for sensitivity analysis. One difficulty is that it may be impossible to collect all the instances of a particular coefficient b_i in a Chvátal function together. Hence the effect of a change in the value of b_i may be obscured. It is also difficult to establish a satisfactory "canonical form". In addition it is often difficult to demonstrate the identity of two Gomory functions other than by complete enumeration over a finite number of values of the b_i. The depth of nesting of the "$\lceil\ \rceil$" operation can be an exponential function of the size of the coefficients.

It is also difficult, systematically, to construct the value function in this form.

We suggest an alternative form for the value function and, in Section 3, give a procedure for obtaining this form.

The general form for model (1.1) is

$$G(b) = \underset{h_k}{\mathrm{Min}}\ \underset{i}{\mathrm{Max}} \left\{ \sum_j \Pi_{ij} b_j + \sum_k \alpha_{ik} h_k \right\}$$

where

$$h_k \in S_k = \{0, 1, \ldots, n_k\} \quad \text{for all} \quad k \in K \tag{1.6}$$

and

$$\sum_j \lambda_{\ell j} b_j + \sum_k \beta_{\ell k} h_k \equiv 0 \pmod{m_\ell} \quad \text{for all } \ell$$

$$L(\underline{b}) = \underset{i}{\mathrm{Max}} \left\{ \sum_j \gamma_{ij} b_j + \sum_k \mu_{ik} h_k \right\}$$

where the h_k are those giving rise to $G(\underline{b})$ above. Π_{ij} and α_{ik} are non-negative rational coefficients and $\lambda_{\ell j}, \beta_{\ell h}, \lambda_{ij}$ and μ_{ik} are non-negative integer coefficients. S_k are finite sets of values for the new integer variables h_k.

This apparently complicated form for the value function is best demonstrated by an example. The h_k can be regarded as "correction terms" for the expressions under each "$\lceil\ \rceil$" operation. These correction terms enable us to remove the "$\lceil\ \rceil$" operation so long as necessary congruence relations are introduced.

For example the term $\dfrac{\lceil b_2 \rceil}{2}$ becomes $\dfrac{b_2}{2} + \dfrac{h_1}{2}$ so long as $b_2 + h_1 \equiv 0 \pmod 2$ and $h_1 \in \{0,1\}$.

Introducing correction terms h_k for each of the expressions under round up operations in (1.3) we obtain (after simplification) the value function in the form

$$\underset{h_k}{\text{Min}}\ \text{Max} \left\{ \frac{3}{2} b_2 + \frac{3}{2} h_1 ,\ 2b_2 + \frac{1}{6} h_2 + \frac{1}{2} h_3 ,\ \frac{9}{17} b_1 + \frac{30}{17} b_2 + \frac{9}{34} h_4 + \frac{3}{2} h_5 \right\}$$

where $h_1 \in \{0,1\},\ h_2 \in \{0,1,\dots,8\}\ ,\ h_3 \in \{0,1\}$

$$h_4 \in \{0,1,\dots,17\}\ ,\ h_5 \in \{0,1\} \qquad (1.7)$$

and

$$b_2 + h_1 \equiv 0 \pmod 2$$
$$b_2 + h_2 \equiv 0 \pmod 9$$
$$4b_2 + h_2 + 3h_3 \equiv 0 \pmod 6$$
$$2b_1 + b_2 + h_4 \equiv 0 \pmod{17}$$
$$6b_1 + 20b_2 + 3h_4 + 17h_5 \equiv 0 \pmod{34}.$$

Note that if the congruences are ignored we can set each h_k to 0 giving the value function of the linear programming relaxation. For the integer case, however, the h_k are restricted to a finite set of values (dependent on the b_i) , within the lattice defined by the congruences.

For the numerical case of $b_1 = 25$ and $b_2 = 20$ it is easy to see that $h_1 = 0$, $h_2 = 7$, $h_3 = 1$, $h_4 = 15$ and $h_5 = 1$, giving the optimal objective value of 54.

There are a number of alternative standard forms in which the congruences could be expressed. This is discussed in Section 2.2.

In Sections 3.1 and 3.2 we give a systematic procedure for constructing the value function of an integer programme in the form above. This is illustrated by a numerical example in Section 4. It is necessary, at various stages, to use a result from Elementary Number Theory known as the Generalised Chinese Remainder Theorem. Since this result is not widely described we do so in the next section.

2. THE GENERALIZED CHINESE REMAINDER THEOREM (GCRT)

Theorem: Given two congruences

$$\begin{aligned} x &\equiv f \pmod{m_1} \\ \text{and} \quad x &\equiv g \pmod{m_2} \end{aligned} \qquad (2.1)$$

there exist multipliers λ_1 and λ_2 such that (2.1) is equivalent to

$$\begin{aligned} x &\equiv \lambda_1 f + \lambda_2 g \pmod{[m_1, m_2]} \\ \text{and} \quad 0 &\equiv f - g \pmod{(m_1, m_2)} \end{aligned} \qquad (2.2)$$

where $[m_1, m_2]$ is the ℓ.c.m. of m_1 and m_2 and (m_1, m_2) is the g.c.d. of m_1 and m_2.

If $(m_1, m_2) = 1$ then the second congruence of (2.2) is vacuous and we obtain the more familiar Chinese Remainder Theorem demonstrating that congruences to coprime moduli may be aggregated into one congruence.

Proof: Since $\frac{[m_1,m_2]}{m_1}$ and $\frac{[m_1,m_2]}{m_2}$ are coprime, by the Euclidean Algorithm, we can find μ_1 and μ_2 such that

$$\mu_1 \frac{[m_1,m_2]}{m_1} + \mu_2 \frac{[m_1,m_2]}{m_2} = 1$$

We take $\lambda_1 = \mu_1 \frac{[m_1 m_2]}{m_1}$, $\lambda_2 = \mu_2 \frac{[m_1,m_2]}{m_2}$

Multiplying the two congruences in (2.1) by λ_1 and λ_2 respectively gives

$$\begin{aligned} \lambda_1 x &\equiv \lambda_1 f \pmod{\mu_1 [m_1,m_2]} \\ \lambda_2 x &\equiv \lambda_2 g \pmod{\mu_2 [m_1,m_2]} \end{aligned} \qquad (2.3)$$

Adding gives

$$x \equiv \lambda_1 f + \lambda_2 g \pmod{[m_1,m_2]}$$

since $\lambda_1 + \lambda_2 = 1$ and $[m_1,m_2]$ is a common factor of the two moduli in (2.3).

Subtracting the two congruences in (2.1) gives the second congruence in (2.2) since (m_1,m_2) is a common factor of both.

Therefore (2.1) $\Longrightarrow$ (2.2)

In order to show that (2.2) $\Longrightarrow$ (2.1) we can write the first congruence of (2.2) as

$$x \equiv f - \lambda_2(f-g) \pmod{[m_1,m_2]}$$

since $\lambda_1 + \lambda = 1$

$$\lambda_2 = \mu_2 \frac{[m_1, m_2]}{m_2} = \mu_2 \frac{m_1}{(m_1, m_2)}$$

Since $f-g$ is a multiple of (m_1, m_2) by the second congruence of (2.2) $\lambda_2(f-g)$ is a multiple of m_1.

Therefore $x \equiv f \pmod{m_1}$.

By a similar argument $x \equiv g \pmod{m_2}$ demonstrating that (2.2) $\Longrightarrow$ (2.1). □

If desired, non-negative values of λ_1 and λ_2 can be found by replacing them by non-negative residues modulus $[m_1 m_2]$.

The GCRT is of use in eliminating a variable x occurring in more than one congruence (2.1) from all but one congruence as in (2.2).

2.1 An Example of Applying the GCRT

$$\text{Suppose } \begin{array}{l} x \equiv f \pmod{12} \\ x \equiv g \pmod{30} \end{array} \qquad (2.4)$$

$[12,30] = 60$, $(12,30) = 6$, $\frac{[12,30]}{12} = 5$, $\frac{[12,30]}{30} = 2$.

Since $1\times5 - 2\times2 = 1$ we take $\lambda_1 = 5$, $\lambda_2 = -4$ giving the equivalent congruences

$$\begin{array}{l} x \equiv 5g - 4g \pmod{60} \\ 0 \equiv f - g \pmod{6} \end{array} \qquad (2.5)$$

If desired we could replace the coefficients of f and g by non-negative (modular equivalent) coefficients to give

$$x \equiv 5f + 56g \pmod{60}$$
$$x \equiv f + 5g \pmod{6} \qquad (2.6)$$

Note that after applying to GCRT the second modulus will divide into the first. Applying the GCRT theorem repeatedly to pairs in a set of congruences we achieve the form in which such modulus divides the previous one. This is, of course, the number theoretic equivalent of the result that any finite abelian group can be expressed as the direct sum of cyclic groups. It is illustrated in Section 2.2.

2.2 Reexpressing Linear Congruences in Alternative Forms

The GCRT can also be used to reexpress a set of linear congruences in a form where the modulus of each congruence divides that of the last. This may be done by successively considering each pair of congruences. It may be illustrated by taking the congruences given with the example in Section 1.3. These are

$$b_2 + h_1 \equiv 0 \pmod{2} \qquad (2.7)$$

$$b_2 + h_2 \equiv 0 \pmod{9} \qquad (2.8)$$

$$4b_2 + h_2 + 3h_3 \equiv 0 \pmod{6} \qquad (2.9)$$

$$2b_1 + b_2 + h_4 \equiv 0 \pmod{17} \qquad (2.10)$$

$$6b_1 + 20b_2 + 3h_4 + 17h_5 \equiv 0 \pmod{34} \qquad (2.11)$$

Combining (2.8) and (2.11) by the GCRT gives

$$108b_1 + 190b_2 + 136h_2 + 207h_4 + 153h_5 \equiv 0 \pmod{306} \qquad (2.12)$$

Combining (2.9) and (2.10) gives

$$36b_1 + 52b_2 + 85h_2 + 51h_3 + 18h_4 \equiv 0 \pmod{102} \qquad (2.13)$$

These two congruences together with (2.7) gives this alterative representation.

Another possible form is to reduce each congruence to a series of congruences whose moduli are primes or powers of primes. This gives (after eliminating redundant congruences).

$$b_2 + h_1 \equiv 0 \pmod{2}$$
$$h_2 + h_3 \equiv 0 \pmod{2}$$
$$h_4 + h_5 \equiv 0 \pmod{2}$$
$$b_2 + h_2 \equiv 0 \pmod{9}$$
$$2b_1 + b_2 + h_4 \equiv 0 \pmod{17}$$

3. THE ELIMINATION OF AN INTEGER VARIABLE BETWEEN LINEAR INEQUALITIES AND CONGRUENCES

In order to construct the value function we successively eliminate each variable between the inequalities of the integer program. The result may be to create linear congruences as well as inequalities involving the remaining variables. Subsequent eliminations may have to take into account these congruences. Therefore the general elimination step, considered here, will assume the integer variable to be eliminated also occurs in a congruence. By means of the GCRT described in the last section we can aggregate all congruences involving the variable into one congruence. The variable will be absent from the other congruences created.

Therefore there is no loss of generality in assuming the variable to occur in, at most, one congruence.

It is also convenient to convert the congruence into a form where the coefficient of the variable to be eliminated divides the modulus of the congruence. This may always be done, for suppose

$$vx \equiv w(\text{mod } m) \tag{3.1}$$

where v is a positive integer. By the Euclidean Algorithm we can find μ and η such that

$$\mu v + \eta m = (v,m)$$

Representing vx by y we have

$$\begin{aligned} y &\equiv w(\text{mod } m) \\ y &\equiv 0(\text{mod } v) \end{aligned} \tag{3.2}$$

By the GCRT we have

$$\begin{aligned} y &\equiv \mu \frac{v}{(v,m)} w\left(\text{mod } \frac{mv}{(v,m)}\right) \\ 0 &\equiv w(\text{mod } (v,m)) \end{aligned} \tag{3.3}$$

Hence from the first of the above congruences

$$(v,m)vx \equiv \mu vw(\text{mod } mv)$$

Dividing through by v gives (together with the second congruence from (3.3))

$$\begin{aligned} (v,m)x &\equiv \mu w(\text{mod } m) \\ 0 &\equiv w(\text{mod } (v,m)) \end{aligned} \tag{3.4}$$

Clearly (v,m) divides m. Therefore there is no loss of generality in assuming that in congruences of the form (3.1) v divides m.

3.1 The General Elimination Step

We will always express the inequalities in the "≥" form. Therefore, if x is the integer variable to be eliminated, we will assume, in general, that it occurs (with non-zero coefficient) in relations of the following three types. (non-negativity conditions are included if applicable).

$$\begin{array}{lrcl} (G) & p_1x & \geq & s \\ (L) & -p_2x & \geq & t \\ (C) & p_3x & \equiv & u(\text{mod } kp_3) \end{array} \qquad (3.5)$$

p_1, p_2, p_3, k are positive coefficients. s, t, u are expressions involving the other variables and right-hand-side coefficients. Following the discussion above we are assuming (C) has at most one member and that the modulus in a multiple of p_3.

In eliminating x we must produce a system of inequalities and congruences in the remaining variables with the same solution set as that implied by the original system. Geometrically we are projecting our system down into a lower dimension.

There are two cases to consider, the first of which is comparatively trivial. We deliberately leave out detailed proofs of the equivalence of the systems before and after elimination. These are easy to construct but cause a diversion from the main procedural explanation. References are given in Section 6 from which such proofs could be obtained.

Case (i) (G) or (L) (or both) empty.

The above relations reduce to

$$0 \equiv u(\text{mod } p_3)$$

If (C) is also empty the above relations reduce to nothing.

Case (ii) (G) and (L) both non-empty

Every pair of inequalities, one from (G) and one from (L) must be considered together with the congruence from (C) if it exists. We take the relations in (3.5) as typical instances. They can be rewritten as

$$\begin{aligned} p_2p_3\,s \;\le\; p_1p_2p_3x \;&\le\; -\,p_1p_3t \\ p_1p_2p_3x \;&\le\; p_1p_2u(\text{mod } kp_1p_2p_3) \end{aligned} \qquad (3.6)$$

Representing $p_1p_2p_3x$ by y we have

$$\begin{aligned} p_2p_3s \;\le\; y \;&\le\; -p_1p_3t \\ y \;&\equiv\; p_1p_2u(\text{mod } kp_1p_2p_3) \\ y \;&\equiv\; 0(\text{mod } p_1p_2p_3) \end{aligned} \qquad (3.7)$$

The congruences can be rewritten, by the GCRT, as

$$\begin{aligned} y \;&\equiv\; p_1p_2u(\text{mod } kp_1p_2p_3) \\ 0 \;&\equiv\; u(\text{mod } p_3) \end{aligned} \qquad (3.8)$$

Taking p_1p_2u from both sides of the inequalities in (3.7) we have

$$\begin{aligned} p_2p_3s - p_1p_2u \le \text{Multiple of } \; kp_1p_2p_3 \le -p_1p_3t - p_1p_2u \\ 0 \;\equiv\; u(\text{mod } p_3) \end{aligned} \qquad (3.9)$$

The import of the inequalities in (3.9) is that a multiple of $kp_1p_2p_3$ lies between the outer two expressions. Therefore there exists a "correction term" h', which can be subtracted from the right-hand-side expression, to give the required multiple.

i.e

$$\begin{aligned} p_2p_3s - p_1p_2u \le -\,p_1p_3t - p_1p_2u - h' \\ -\,p_1p_3t - p_1p_2u - h' \equiv 0(\text{mod } kp_1p_2p_3) \end{aligned} \qquad (3.10)$$

Moreover h' can be restricted to the values $\{0, 1, \ldots, kp_1p_2p_3 - 1\}$.

Since all other terms in the congruence of (3.10) are multiples of p_1 we may replace h' by $p_1 h$ with h suitably restricted.

Reexpressing (3.10) and including the second congruence of (3.9) gives

$$
\begin{aligned}
0 &\geq p_3(p_2 s + p_1 t) + p_1 h \\
0 &\equiv p_3 t + p_2 u + h (\mathrm{mod}\ k p_2 p_3) \\
0 &\equiv u\ (\mathrm{mod}\ p_3) \\
h &\in \{0, 1, \ldots, k p_2 p_3 - 1\}
\end{aligned}
\qquad (3.11)
$$

as the full system resulting from the elimination of x between a set of relations (3.5). Should (C) be empty in (3.5) then some simplification is possible and the system reduces to

$$
\begin{aligned}
0 &\geq p_2 s + p_1 t + p_1 h \\
0 &\equiv t + h (\mathrm{mod}\ p_2) \\
h &\in \{0, 1, \ldots, p_2 - 1\}
\end{aligned}
\qquad (3.12)
$$

It should be reemphasised that the elimination of x must be carried out between all pairs of inequalities from (G) and (L) together with any congruence from (C). The resulting system (3.11) will therefore contain a number of inequalities together with a congruence and correction term corresponding to each member of (L)

3.2 Constructing the Value Function

We express our model (1.1) in the form

$$
\text{Minimise } x_0 \text{ s.t.} \quad
\begin{aligned}
x_0 - \sum_j c_j x_j &\geq 0 \\
\sum_j a_{ij} x_j &\geq b_i \qquad i \in I \\
x_j &\geq 0 \qquad j \in J \\
x_0, x_1, x_2 &\quad \text{integer}
\end{aligned}
\qquad (3.13)
$$

The variables x_1, x_2, etc can then be eliminated in any order using the procedure of Section (3.1). At each elimination the resultant inequalities (3.11) arise from non-negative integral combinations of the inequalities in (3.13) together with non-negative correction terms (taking a finite number of possible values). The resultant congruences also arise from non-negative combinations of the expressions in (3.13) together with non-negative correction terms.

Hence the resultant system, after combining and transforming congruences containing x_0, takes the form

$$p_{i1}x_0 \geq s_i \qquad i \in I_1 \qquad (3.14)$$

$$0 \geq t_i \qquad i \in I_2 \qquad (3.15)$$

$$p_2 x_0 \equiv u \pmod{kp_2} \qquad i \in I_3 \qquad (3.16)$$

$$0 \equiv v_i \pmod{p_{i3}} \qquad i \in I_4 \qquad (3.17)$$

where p_{i1}, p_2, p_{i3} and k are positive integers and s_i, t_i, u and v_i are non-negative linear combinations of the original right-hand-sides and the h_k.

In order to obtain the value function in the desired form we carry out the following steps

(i) Replace a suitable multiple of x_0 by y so as to express (3.14) and (3.16) in the form

$$y \geq s_i' \qquad i \in I_1 \qquad (3.18)$$

$$y \equiv u' \pmod{p'} \qquad i \in I_3 \qquad (3.19)$$

(ii) By adding a correction term h_i we express (3.18) as

$$y - u' \geq s_i' - u' + h_i \qquad i \in I_1 \qquad (3.20)$$

where $h_i \in \{1, 2, \ldots, p' - 1\}$.

Since the left-hand-side of (3.20) is a multiple of p' we have

$$0 \equiv s_i' - u' + h_i \pmod{p'} \qquad i \in I_1 \qquad (3.21)$$

These new congruences (3.21), together with those in (3.17), combined with the inequalities (3.20) give the relations for x_0 in the following form

$$x_0 \geq \sum_j \Pi_{ij} b_j + \sum_k \alpha_{ik} h_k \qquad i \in I_1$$

where $h_k \in S_k = \{0, 1, \dots, n_h\}$ for all $k \in k$

$$0 \geq \sum_j \gamma_{ij} b_j + \sum_k \mu_{ik} h_k \qquad i \in I_2$$

$$0 \equiv \sum_j \ell_{ij} b_j + \sum_k \beta_{ik} h_k \pmod{m_i} \qquad i \in I_3'$$

The expression (1.6) for the value function follows. A numerical example demonstrates the full procedure.

4. AN EXAMPLE

We construct the value function for the numerical example given in Section 1.1 using the methods described in Sections 3.1 and 3.2.

The model is expressed in the form

Minimise x_0

such that

$$x_0 - 18x_1 + 3x_2 \geq 0 \qquad (4.1)$$

$$4x_1 + x_2 \geq b_1 \qquad (4.2)$$

$$9x_1 - 2x_2 \geq b_2 \qquad (4.3)$$

$$x_1 \geq 0 \qquad (4.4)$$

$$x_2 \geq 0 \qquad (4.5)$$

$$x_0,\ x_1,\ x_2 \quad \text{integer}$$

The constraints involving x_1 can be expressed as

$$\left.\begin{array}{r} 9(b_1 - x_2) \\ 4(b_2 - 2x_2) \\ 0 \end{array}\right\} \le 36x_1 \le 2(x_0 + 3x_2) \qquad (4.6)$$

Introducing a correction term $2h_1$ subtracted from the right-hand expression of (4.6) allows us to eliminate x_1 by giving

$$\left.\begin{array}{r} 9(b_1 - x_2) \\ 4(b_2 + 2x_2) \\ 0 \end{array}\right\} \le 36x_1 \le 2(x_0 + 3x_2) - 2h_1 \qquad (4.7)$$

$$0 \equiv x_0 + 3x_2 - h_1 (\mathrm{mod}\ 18) \qquad (4.8)$$

where $h_1 \in \{0,1,\ldots,.17\}$.

This reduces the original system to

$$2x_0 + 15x_2 \ge 9b_1 + 2h_1 \qquad (4.9)$$

$$x_0 - x_2 \ge 2b_2 + h_1 \qquad (4.10)$$

$$x_0 + 3x_2 \ge h_1 \qquad (4.11)$$

$$x_2 \ge 0 \qquad (4.12)$$

$$x_0 + 3x_2 \equiv h_1 (\mathrm{mod}\ 18) \qquad (4.13)$$

where $h_1 \in \{0,1,\ldots,17\}$.

In order to eliminate x_2 this system can be expressed as

$$\left.\begin{array}{r} 9b_1 - 2x_0 + 2h_1 \\ 5(-x_0 + h_1) \\ 0 \end{array}\right\} \le 15x_2 \le 15(-2b_2 + x_0 - h_1) \qquad (4.14)$$

$$3x_2 \equiv x_0 + h_1 (\mathrm{mod}\ 18) \qquad (4.15)$$

Representing $15x_2$ by y (4.15) can be written

$$y \equiv 5(-x_0 + h_1)(\mathrm{mod}\ 18) \qquad (4.16)$$

$$y \equiv 0(\mathrm{mod}\ 15) \qquad (4.17)$$

which can be rewritten using the GCRT as

$$y \equiv 35(-x_0 + h_1)(\text{mod } 90) \tag{4.18}$$

$$0 \equiv -x_0 + h_1 \ (\text{mod } 3) \tag{4.19}$$

The inequalities (4.14) can now be written as

$$\left.\begin{matrix} 9b_1 - 2x_0 + 2h_1 - 35(-x_0 + h_1) \\ 5(-x_0 + h_1) - 35(-x_0 + h_1) \\ - 35(-x_0 + h_1) \end{matrix}\right\} \le \begin{matrix} \text{Multiple} \\ \text{of} \\ 90 \end{matrix} \ \begin{matrix} 15(-2b_2 + x_0 - h_1) \\ \le -35(-x_0 + h_1) \\ \end{matrix} \tag{4.20}$$

A correction term $5h_2$ is subtracted from the right-hand expression in (4.20). The system can now be reduced, after some simplification, to

$$17x_0 \ge 9b_1 + 30b_2 + 17h_1 + 5h_2 \tag{4.21}$$

$$4x_0 \ge 6b_2 + 4h_1 + h_2 \tag{4.22}$$

$$3x_0 \ge 6b_2 + 3h_1 + h_2 \tag{4.23}$$

$$10x_0 \equiv 6b_2 + 10h_1 + h_2(\text{mod } 18) \tag{4.24}$$

$$x_0 \equiv h_1(\text{mod } 3) \tag{4.25}$$

where $h_1, h_2 \in \{0, 1, \ldots, 17\}$.

The congruences can be rearranged into the form

$$x_0 \equiv 6b_2 + h_1 + h_2(\text{mod } 9) \tag{4.26}$$

$$0 \equiv h_2(\text{mod } 2) \tag{4.27}$$

If $204x_0$ is represented by y we have

$$y \ge 12(9b_1 + 30b_2 + 17h_1 + 5h_2) \tag{4.28}$$

$$y \ge 51(6b_2 + 4h_1 + h_2) \tag{4.29}$$

$$y \ge 68(6b_2 + 3h_1 + h_2) \tag{4.30}$$

$$y \equiv 204(6b_2 + h_1 + h_2)(\text{mod } 1836) \tag{4.31}$$

$$y \equiv 0(\text{mod } 204) \tag{4.32}$$

Congruence (4.32) is redundant.

The inequalities can now be written as

$$y - 204(6b_2+h_1+h_2) \geq \begin{cases} 12(9b_1+30b_2+17h_1+5h_2) - 204(6b_2+h_1+h_2) \\ 51(6b_2+4h_1+h_2) - 204(6b_2+h_1+h_2) \\ 68(6b_2+3h_1+h_2) - 204(6b_2+h_1+h_2) \end{cases} \quad (4.33)$$

Since the left-hand-side is a multiple of 1836 we can add correction terms of $12h_3$, $51h_4$, and $68h_5$ respectively to the three right-hand expressions. After some simplification, and including congruence (4.27), this gives the value function as

$$\underset{h_k}{\text{Min}}\ \text{Max} \left\{ \frac{9}{17} b_1 + \frac{30}{17} b_2 + \frac{5}{17} h_2 + \frac{1}{17} h_3 ,\quad \frac{3}{2} b_2 + \frac{1}{4} h_2 + \frac{1}{4} h_4 ,\quad 2b_2 + \frac{1}{3} h_2 + \frac{1}{3} h_5 \right\} \quad (4.34)$$

where $h_2 \in \{0,1,\ldots,17\}$, $h_3 \in \{0,1,\ldots,152\}$,

$h_4 \in \{0,1,\ldots,35\}$, $h_5 \in \{0,1,\ldots,26\}$

and

$$9b_1 + 81b_2 + 141h_2 + h_3 \equiv 0 (\text{mod } 153)$$

$$33h_2 + h_4 \equiv 0 (\text{mod } 36)$$

$$15b_2 + 25h_2 + h_5 \equiv 0 (\text{mod } 27)$$

$$h_2 \equiv 0 (\text{mod } 2) .$$

Alternatively the congruences in (4.34) can be expressed in the following form, with moduli dividing previous moduli.

$$756b_1 + 276b_2 + 1423h_2 + 1104h_3 + 663h_4 + 1768h_5 \equiv 0(\text{mod } 1836)$$

$$9h_2 + 10h_3 + 8h_4 \equiv 0(\text{mod } 18) \quad (4.35)$$

$$3b_2 + 5h_2 + 7h_3 + h_4 + 8h_5 \equiv 0(\text{mod } 9)$$

or expressing to prime and powers of prime moduli giving (after eliminating redundant congruences)

$$h_2 \equiv 0 \pmod{2}$$
$$h_2 + h_4 \equiv 0 \pmod{4}$$
$$6h_2 + h_3 \equiv 0 \pmod{9}$$
$$6h_2 + h_4 \equiv 0 \pmod{9}$$
$$9b_1 + 13b_2 + 5h_2 + h_3 \equiv 0 \pmod{17}$$
$$15b_2 + 25h_2 + h_5 \equiv 0 \pmod{27}$$

For the trial values $b_1 = 25$, $b_2 = 20$ the optimal solution to (4.34) is given when $h_2 = 6$, $h_3 = 63$, $h_4 = 18$ and $h_5 = 9$. This gives the optimal objective value of 54.

Although this form of the value function is the same as that given for the example in Section 1.3 there is no obvious canonical form in which both can be expressed to demonstrate their identity.

5. FURTHER CONSIDERATIONS

The method used to construct the form of the value function of an Integer Programme described here is a development of a method described in Williams [5]. In that paper a disjunction of linear programmes is created. Since each linear programme in the disjunction has the same structure it seems clearer to represent the parameters in each clause of the disjunction as integer variables (h_k). The result is, of course, to create another integer programme. This integer programme (i) restricts each of the integer variables h_k to a finite number of values and (ii) gives them all non-negative coefficients.

While the method will not be computationally tractible, in general, it could be applied in more specialist forms e.g. fixing the b_i

at specific values or restricting the method to specialist models. In such cases streamlining may be possible leading to, computationally feasible integer programming algorithms. The major aim of this paper is, however, to demonstrate the structure of the value function and dual of an integer programme in a more transparent form.

It should be pointed out that the original method described derives from that of Presburger [4] for Additive Arithmetic.

6. REFERENCES

1 C.E. Blair and R.G. Jeroslow, "The value function of an integer programme, Mathematical Programming, 23(1982) 237-273.

2. V. Chvátal, "Edmonds polytopes and a hierarchy of combinational problems", Discrete Mathematics, 4(1973) 305-337.

3. R.E. Gomory, "An algorithm for integer solutions to linear programs", in R.L. Graves and P. Wolfen, Eds., Recent advances in Mathematical programming (McGraw-Hill, New York, 1963) pp269-302.

4. M. Presurger, "Über die Vollsträndigkeit eines gewissen Systems der Arithmetik gabzer Zagkeb ub wekcgen due Addition als einzige Operation hervortrith, C.-R. I. Congres des Math.des Pays Slaves, Warsaw (1930), pp92-101.

5. H.P. Williams, "Fourier-Motzkin Elimination extended to Integer Programming problems, Journal of Combinational Theory, 21(1976) 118-123.

6. L. Wolsey, "Integer programming duality:Price functions and sensitivity analysis", Mathematical Programming, 20(1981) 173-195.