J. Opl Res. Soc. Vol. 43, No. 5, pp. 387–393

0160-5682/92 $3.50+0.00

The Elimination of Integer Variables

H. P. WILLIAMS
University of Southampton, UK

It is pointed out that the projection of a Linear Programme (LP) into a lower dimension still results in an LP. For an Integer Programme (IP) this is not generally the case. Circumstances in which the projection (after eliminating integer variables) is still an IP are given.

Key words: integer programming, projections of polyhedra

INTRODUCTION

In order to describe our results we will consider a general mixed integer programming model in the form

minimize x_0

$$\text{subject to } P: \begin{cases} \sum_{j=0}^{n} a_{ij}x_j = b_i \quad i = 0, 1, \ldots, m_1, & (1) \\ \sum_{j=0}^{n} a_{ij}x_j \geqslant b_i \quad i = m_1 + 1, m_1 + 2, \ldots, m_2 & (2) \end{cases}$$

where the x_j are members of $\mathbb{R}$ and/or $\mathbb{Z}$. If all x_j are members of $\mathbb{R}$ we have a Linear Programme (LP). If all are members of $\mathbb{Z}$ we have a Pure Integer Programme (PIP). Otherwise, we have a Mixed Integer Programme (MIP).

The most successful general method of solving such models is the branch-and-bound algorithm. It was Ailsa Land who, together with Alison Doig[1], first suggested such an approach to solving MIP methods. This approach was in contrast to the apparently more sophisticated Cutting Plane methods of Gomory[2]. There are a number of different forms of the branch-and-bound algorithm. Their description is outside the scope of this paper. All rely on choosing individual *integer* variables at successive stages in the optimization and further constraining them by means of additional inequalities and/or equations. The number of integer variables is therefore sometimes used as an indication of the difficulty of an IP model. This judgement is, of course, too simplistic since some variables will automatically be forced to be integer by virtue of the original constraints or the optimality stipulation. It is of interest to examine when this might happen. One way of doing this is to examine when it is possible to eliminate certain integer variables from a model completely and still retain an IP model. This is the subject of this paper. The paper, is however, written partly as a tribute to Ailsa Land whose pioneering work encouraged the use of IP models and the branch-and-bound approach. There is value to be gained in transforming models, when possible to make the approach more efficient.

For a Linear Programming (LP) model it is always possible to eliminate variables to produce another LP model. Geometrically this can be visualized as projecting a polyhedra down into a lower dimensional space. The result will clearly still be a polyhedron.

For an Integer Programming (IP) model an analogous situation does not generally hold. This is demonstrated by the following small example.

$$\begin{gathered} 7x - 2y \geqslant 0 \\ -3x + y \geqslant 0 \\ x, y \in \mathbb{Z} \end{gathered} \tag{3}$$

The situation is illustrated in Figure 1.

Correspondence: H. P. Williams, Faculty of Mathematical Studies, University of Southampton, Southampton, Hampshire SO9 5NN, UK.

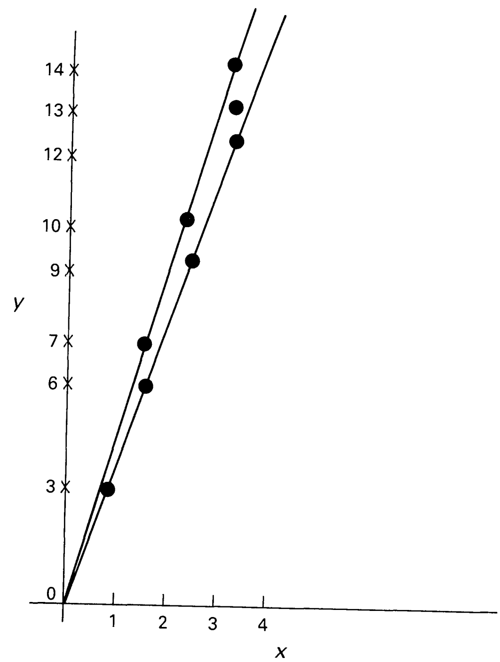

FIG. 1. *Projection of an integer program to a lower dimension.*

It can be seen that the projection onto the y axis gives the set of feasible values of y as

$$0, 3, 6, 7, 9, 10, 12, 13, \ldots \tag{4}$$

For the LP relaxation of (3) the set of feasible values of y is the set $y \geqslant 0$, i.e. a polyhedron in 1-space. Expression (4) does not, however, consist of all integral y within this polyhedron.

In this paper we demonstrate how to recognize situations in which the set of feasible solutions does consist of all the integer solutions within the projection of the polyhedron corresponding to the LP relaxation. Then it is possible to eliminate such integer variables and still have an IP model (in a smaller number of variables). Practically, this may, or may not, be a sensible tranformation. We discuss circumstances and structures in which such a transformation is possible and worthwhile.

Formally, the relations (1) and (2) can be regarded as a predicate $P(x_0, x_1, x_3, \ldots, x_n)$. By eliminating a variable x_q we seek another predicate $P'(x_0, x_1, \ldots, x_{q-1}, x_{q+1}, \ldots, x_n)$ such that

$$\exists x_q P(x_0, x_1, \ldots, x_q, \ldots, x_n) \equiv P'(x_0, x_1, \ldots, x_{q-1}, x_{q+1}, \ldots, x_n). \tag{5}$$

Notice that this equivalence must extend to the case when both P and P' are empty, i.e. the original model will be infeasible if and only if the resulting system, after the elimination of x_q, is infeasible. We are therefore looking for circumstances in which it is possible to eliminate the existential quantifier, $\exists x_q$, and the associated variable, and express the predicate P' in a similar form to (1) and (2). In order to demonstrate an equivalence (5) it is necessary to show that

(a) given $x_0, x_1, \ldots, x_{q-1}, x_q, x_{q+1}, \ldots, x_n$ satisfying P then $x_0, x_1, \ldots, x_{q-1}, x_{q+1}, \ldots, x_n$ satisfies P'; and

(b) given $x_0, x_1, \ldots, x_{q-1}, x_{q+1}, \ldots, x_n$ satisfying P' then $x_0, x_1, \ldots, x_{q-1}, x_q(x_0, x_1, \ldots, x_{q-1}, x_{q+1}, \ldots, x_n), x_{q+1}, \ldots, x_n$ satisfies P where x_q is a function of the given arguments.

There is no loss of generality in restricting the coefficients a_{ij}, b_i to integer values so long as they are rational (which is generally the case for practical models). Also, we assume that the coefficients in each constraint have been divided through by their greatest common denominator.

We have deliberately represented the objective function by a variable x_0 and equated it as one of the constraints (1). Variable x_0 will not be regarded as a variable for elimination.

Any non-negativity constraints on variables are incorporated in the constraints (2).

We will show that it is possible to eliminate an integer variable $x_j \in \mathbb{Z}$ if:

(i) it has an entry ± 1 in at least one equality row (1) and all other variables in this row are integer variables; or
(ii) it has an entry $+1$ in at least one all-integer inequality row (2) with entries in other rows (2) all being 0, negative or $+1$; or
(iii) it has an entry -1 in at least one all-integer inequality row (2) with entries in other rows (2) all being 0, positive or -1.

In case (i), or in cases (ii) or (iii) where the ± 1 entry occurs in only one row (2), then the elimination will result in a reduction in the size of the model.

THE ELIMINATION OF CONTINUOUS VARIABLES

Case 1

$$x_q \in \mathbb{R}$$

and

$$\sum_{j=0}^{n} a_{pj} x_j = b_p, \quad a_{pq} \neq 0 \tag{6}$$

i.e. x_q has an entry in an equality row.

We simply use this equation to substitute x_q out of the other equations. This is the well-known pivoting operation which we do not need to describe in detail.

Case 2

$$x_q \in \mathbb{R}$$

and

$$a_{iq} = 0 \text{ for } i = 0, 1, \ldots, m_1$$

i.e. x_q has no entry in an equality row.

Therefore no such row can be used for substitution. Instead we must eliminate x_q between each pair of inequalities in (2) in which it has coefficients of opposite sign. These inequalities in which x_q has a non-zero coefficient are conveniently partitioned into those in which it has a positive coefficient (set G) and those in which it has a negative coefficient (set L). Also, we can divide through these inequalities by the relevant non-zero coefficients giving

$$x_q \geqslant \frac{1}{a_{iq}} \left(b_i - \sum_{\substack{j=0 \\ j \neq q}}^{n} a_{ij} x_j \right) \quad i \in G \tag{7}$$

$$\frac{1}{a_{kq}} \left(b_k - \sum_{\substack{j=0 \\ j \neq q}}^{n} a_{kj} x_j \right) \geqslant x_q \quad k \in L \tag{8}$$

together with (1) and inequalities (2) in which x_q has no entry.

Eliminating x_q between (7) and (8) gives

$$\frac{1}{a_{kq}} \left(b_k - \sum_{\substack{j=0 \\ j \neq q}}^{n} a_{kj} x_j \right) \geqslant \frac{1}{a_{iq}} \left(b_i - \sum_{\substack{j=0 \\ j \neq q}}^{n} a_{ij} x_j \right) \; i \in G, \; k \in L \tag{9}$$

This elimination is possible since x_q lies in the continuum of real numbers. Note that the number of inequalities in (9) is $|G| \times |L|$. If G or L is empty then there are no inequalities in (9). After collecting terms (and remembering a_{kq} is negative) the eliminated system becomes

$$P' \begin{cases} \sum_{\substack{j=0 \\ j \neq q}}^{n} a_{ij} x_j = b_i \quad i = 0, 1, \ldots, m_1 \text{ and } a_{iq} = 0 & (10) \\ \sum_{\substack{j=0 \\ j \neq q}}^{n} a_{ij} x_j \geqslant b_i \quad i = m_1 + 1, m_1 + 2, \ldots, m_2 \text{ and } a_{iq} = 0 & (11) \\ \sum_{\substack{j=0 \\ j \neq q}}^{n} (a_{iq} a_{kj} - a_{kq} a_{ij}) x_j \geqslant (a_{iq} b_k - a_{kq} b_i) \quad i \in G, k \in L & (12) \end{cases}$$

In order to prove the equivalence of P' above, and P, we first (a) observe that the set of x_j satisfying P must also satisfy P'. This is because (10) and (11) are taken from (1) and (2), and (12) is obtained by taking non-negative multiples (a_{iq} and $-a_{kq}$) of constraints in (2).

Conversely, (b) if we have a set of $x_j\,(j \neq q)$ satisfying (10), (11) and (12) we can check, by taking non-negative multiples of (12), that the set of x_j together with

$$x_q = \max_{i \in G} \left[\frac{1}{a_{iq}} \left(b_i - \sum_{\substack{j=1 \\ j \neq q}}^{n} a_{ij} x_j \right) \right] \tag{13}$$

is a solution to P. Alternatively we could take

$$x_q = \min_{k \in L} \left[\frac{1}{a_{kq}} \left(b_k - \sum_{\substack{j=1 \\ j \neq q}}^{n} a_{kj} x_j \right) \right] \tag{14}$$

Note that either expression for x_q lies in the continuum of real numbers.

The above elimination is an example of Fourier–Motzkin elimination and is described by Williams[3]. In general the number, $|G| \times |L|$, of inequalities in (12) results in the system P' having many more constraints than P. Should, however, G or L be empty then the system P' simply remains P with those constraints involving x_q removed. This simplification of an LP resulting from an examination of sign patterns has already been pointed out by Brearley *et al.*[4] using a different argument. If G or L has only one member then P' will have exactly the same number of constraints as P but, of course, will not involve the variable x_q. Such a simplification could be worthwhile.

THE ELIMINATION OF INTEGER VARIABLES

Case 3

$$x_q \in \mathbb{Z}$$

and

$$\sum_{j=0}^{n} a_{pj} x_j = b_p \quad a_{pq} \neq 0$$

i.e. x_q has an entry in an equality row.

It is no longer sufficient simply to use this equation to substitute x_q out of the other equations (1) and inequalities (2) since the resulting expression for x_q will no longer guarantee it to be integral.

If however $a_{pq} = \pm 1$ and the expression $b_p - \sum_{\substack{j=0 \\ j \neq q}}^{n} a_{pj} x_j$ is integral then case 1 gives an equivalent system to P with the necessary integrality conditions on the variables. (This is condition (i) of the first section.)

Otherwise it is necessary to stipulate the condition

$$\sum_{\substack{j=0\\ j\neq q}}^{n} a_{pj}x_j \equiv b_p \pmod{a_{pq}} \tag{15}$$

and append it to the constraints. This system no longer takes the form of P, i.e. does not have an IP in the space of $x_0, x_1, \ldots, x_{q-1}, x_{q+1}, \ldots, x_n$.

Case 4

$$x_q \in \mathbb{Z}$$

and

$$a_{iq} = 0 \text{ for } i = 0, 1, \ldots, m_1$$

i.e. x_q has no entry in an equality row. We must therefore eliminate x_q between pairs of inequalities in (2) in which it has coefficients of opposite sign as in case 3 above. The elimination now, however, is more complicated.

The import of (7) and (8) is no longer simply that a variable in the continuum of real numbers lies between the left-hand and right-hand expressions enabling us to state (9). We also need to capture the condition that an *integer* must lie between these two expressions.

This condition is guaranteed if either all the right-hand or all the left-hand expressions are integral. This happens when $a_{iq} = +1$ for all $i \in G$ and $b_i - \sum_{\substack{j=0\\ j\neq q}}^{n} a_{ij}x_j$ is an integral expression for $i \in G$ or $a_{iq} = -1$ for all $i \in L$ and $b_k - \sum_{\substack{j=0\\ j\neq q}} a_{kj}x_j$ is an integral expression for $k \in L$. In the first instance we have condition (ii) of the first section and in the second instance condition (iii). The value of x_q is given by (13) or (14) respectively. Otherwise, a more complicated condition has to be stated. In order to derive this, and remembering that a_{kq} is negative, it is more convenient to restate (7) and (8) as

$$a_{iq}\left(-b_k + \sum_{\substack{j=0\\ j\neq q}}^{n} a_{kj}x_j\right) \geqslant a_{iq}a_{kq}x_q \geqslant a_{kq}\left(-b_i + \sum_{\substack{j=0\\ j\neq q}}^{n} a_{ij}x_j\right). \tag{16}$$

The import of (16) is that a multiple of $a_{iq}a_{kq}(=M)$ lies between the left-hand and right-hand expressions. If we represent these expressions by g and f respectively we have the situation shown in Figure 2.

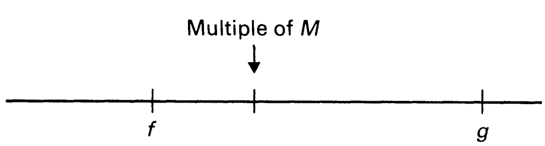

FIG. 2. *Multiple of an integer lying between two integer expressions.*

This condition can be captured by introducing an 'error' variable h and stating the constraints

$$f \leqslant g - h \tag{17}$$

$$g - h \equiv 0 \pmod{M} \tag{18}$$

h can be restricted to lie in the interval $[0, M)$.

Alternatively we could state the constraints as

$$f \leqslant g - h' \tag{19}$$

$$f + h' \equiv 0 \pmod{M} \tag{20}$$

with h' being interpreted as $M - h$ and also lying in $[0, M)$. If f and g are themselves integer expressions then h (or h') could be restricted to the values $\{0, 1, \ldots, M-1\}$ and (17) and (18) (or (19) and (20)) expressed as a disjunction of IPs. Since both expressions f and M are multiples of a_{kq} then h' must be also, and can be replaced by $a_{kq}t_i$ where t_i is restricted to the values $\{0, 1, \ldots, a_{iq}-1\}$. Dividing (23) through by a_{kq} the full condition is then captured in a finite manner by the disjunction

$$\bigvee_{t_i \in \{0, 1, \ldots, a_{iq}-1\}} \left\{ \begin{array}{ll} \sum\limits_{\substack{j=0 \\ j \neq q}}^{n} (a_{iq}a_{kj} - a_{kq}a_{ij})x_j \geqslant (a_{iq}b_q - a_{kq}b_i) + a_{kq}t_i, \quad i \in G,\ k \in L & (21) \\ \sum\limits_{\substack{j=0 \\ j \neq q}}^{n} a_{ij}x_j \equiv b_i - t_i \ (\text{mod } a_{iq}) \quad i \in G & (22) \end{array} \right.$$

A fuller discussion of this derivation is contained in Williams[5]. The alternative to considering (21) and (22) as constituting a disjunction of IPs in the variables $x_0, x_1, \ldots, x_{q-1}, x_{q+1}, \ldots, x_n$ is to regard the t_i as new integer variables. Their introduction would, however, appear to complicate rather than simplify the representation since we have introduced new integer variables t_i together with congruences (22). It should, however be recognized that the integer variables are each restricted to a finite number of values (which may not be the case for the eliminated variable). Also, if the coefficient a_{iq} is small the number of possible values for t_i (or clauses in the disjunction above) will be small.

The condition above could alternatively be represented (in the manner of (17) and (18)) by the disjunction

$$\bigvee_{t_k \in \{0, 1, \ldots, a_{iq}-1\}} \left\{ \begin{array}{ll} \sum\limits_{\substack{j=0 \\ j \neq q}}^{n} (a_{iq}a_{kj} - a_{kq}a_{ij})x_j \geqslant (a_{iq}b_k - a_{kq}b_i) + a_{iq}t_k, \quad i \in G,\ k \in L & (23) \\ \sum\limits_{\substack{j=0 \\ j \neq q}}^{n} a_{kj}k_j \equiv b_k - t_k \ (\text{mod } a_{kq}) \quad k \in L & (24) \end{array} \right.$$

In this case the term h in (17) and (18) has been replaced by $a_{iq}t_k$ where t_k is restricted to the values $\{0, 1, \ldots, a_{iq}-1\}$. Whether (21) and (22) or (23) and (24) is the more economical representation depends on the relative magnitudes of a_{iq} and a_{kq}.

These necessary extra conditions resulting from the elimination of an integer variable x_q are given for completeness. In practice it would generally not be worth performing the elimination unless special conditions held such as (i), (ii) or (iii) from the first section. Notice that in such circumstances one of the set of congruences (15), (22) or (24) becomes vacuous yielding a system expressed entirely in terms of equations and inequalities.

FURTHER OBSERVATIONS

There are some situations where the elimination of integer variables, where possible, could be well worthwhile. The 'disjunctive' formulations produced by Jeroslow[6] are sharper than conventional formulations in the sense that the LP relaxation has a smaller feasible region. They do, however, produce formulations with many more variables. In practice it is often possible to eliminate many of these variables. The 'closeness' of the LP and IP solutions is also suggestive of this possibility.

One situation where it is, of course, possible (but not usually worthwhile) to eliminate integer variables is when the IP model has a totally unimodular structure, e.g. Network Flow models and their derivatives. In such circumstances one of the conditions (i), (ii) or (iii) of the first section always holds even after some variables have been eliminated. Therefore, the distinction between integer and continuous variables for the purposes of elimination becomes irrelevant.

Finally, it should be pointed out that it may not always be desirable to eliminate integer variables that could provide a useful 'dichotomy' in the branch-and-bound solution strategy. This consideration is discussed in Williams[7].

The automatic elimination of 0–1 integer variables abiding by one of the conditions (i), (ii), or (iii) of the first section is incorporated as an option in the system described by McKinnon and Williams[8]. Two examples of when it is possible arising in a blending and a project selection problem are given there.

REFERENCES

1. A. H. Land and A. G. Doig (1960) An automatic method for solving discrete programming problems. *Econometrica* **28**, 497–520.
2. R. E. Gomory (1958) Outline of an algorithm for integer solutions to linear programs. *Bull. Am. Math. Soc.* **64**, 275–278.
3. H. P. Williams (1986) Fourier's method of linear programming and its dual. *Am. Math. Monthly* **93**, 681–695.
4. A. L. Brearley, G. Mitra and H. P. Williams (1975) Analysis of mathematical programming problems prior to applying the simplex algorithm. *Math. Prog.* **8**, 54–83.
5. H. P. Williams (1988) An alternative form of the value function of an integer programme. OR16, Faculty of Mathematical Studies, University of Southampton, U.K.
6. R. G. Jeroslow (1989) Logic-based decision support; mixed integer model formulation. *Annals of Discrete Mathematics*, Monograph 40, North Holland, Amsterdam.
7. H. P. Williams (1984) Model building in linear and integer programming. In *Computational Mathematical Programming* (K. Schittkowski, Ed.) pp 25–33. Springer-Verlag, Berlin.
8. K. I. M. McKinnon and H. P. Williams (1989) Constructing integer programming models by the predicate calculus. *Annals of OR*. **21**, 227–245.