Belgian Journal of Operations Research, Statistics and Computer Science Vol. 36 (1) 96

A Note on Logic Cuts and Valid Inequalities for Certain Standard (0-1) Integer Programs

John M. Wilson

Loughborough University Business School
Loughborough
Leicestershire
England LE11 3TU

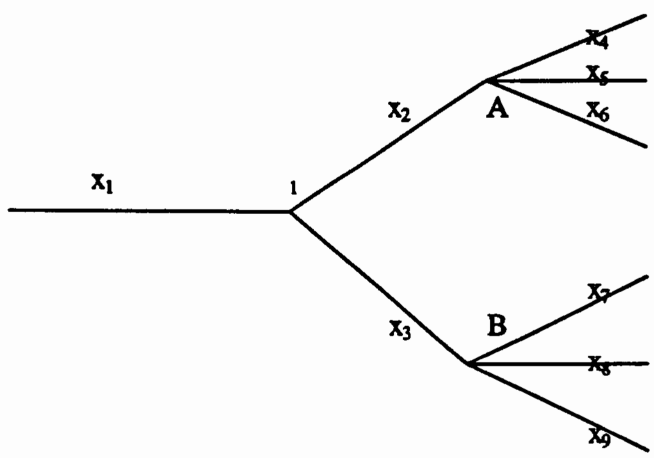
Abstract

A recent development in (0-1) integer programming has been the use of logic cuts. These are cuts that can be derived from the logic of the integer programming model and although not binding on the LP relaxation optimal solution, do improve the traversal of the branch and bound tree. This note will demonstrate how logic cuts and other valid inequalities can be developed for and appended to certain standard (0-1) integer programs.

Keywords : bin packing problem, capacitated clustering problem, generalised assignment problem, logic cut, valid inequality

1. Introduction

A paper by Hooker (1992) developed valid cuts for (0-1) integer programs (IP's) using the process of generalised resolution for a series of standard (0-1) IP's. In a later paper, Hooker et al. (1994), the idea of a logic cut was introduced and further demonstrated to be useful in various problems in Hooker (1995). An algorithm for generating logic cuts was given in Wilson (1995). Logic cuts are cuts which can be derived from the logic of a (0-1) IP model. The cuts were originally derived for network problems to avoid the IP solution process having to consider illogical flows in the network. Such network problems arise in the oil and chemical industries and a fragment of one is illustrated in Figure 1 and discussed below.



Example 1.1

In Figure 1 x_1 represents a flow of crude oil which may be processed using converters A or B (or a mixture of both) and x_2 and x_3 represent the quantities of crude sent to converters A and B, respectively.

Clearly $\quad x_1 \geq x_2 + x_3$.

After processing at A three products are produced and x_4, x_5, and x_6 represent their quantities.

$$x_4 \leq a_4x_2 ; \quad x_5 \leq a_5x_2 ; \quad x_6 \leq a_6x_2 ,$$

where a_4, a_5, a_6 are constants such that $a_4 + a_5 + a_6 < 1.0$.

A similar situation exists at B where x_7, x_8, and x_9 are produced. Set-up costs are incurred when each of processes A or B are used and consequently it may not be optimal to use both. (0,1) variables d_1, d_2, and d_3 are introduced to denote whether there is any flow to nodes 1, A or B, respectively, and

$x_j \leq Md_j \quad j = 1, 2, 3,$ where M denotes maximum flow available to node 1.

Two logic cuts can be added to the IP model

$$d_2 \leq d_1 \text{ and } d_3 \leq d_1 .$$

The cuts developed in Example 1 appear to be redundant, as processes will not be "opened", according to an optimal IP solution, unless they are needed to provide a flow. However, Hooker et al. (1994) discovered that during the solution of such network problems , branch and bound was setting d variables to the value 1 to allow for flows when the corresponding x variable was zero in the solution. Although these cuts have no effect on tightening the LP relaxation of an IP problem, they usually have useful effects on the branch and bound tree when the problem is solved. The number of nodes required to solve the problem may be reduced dramatically. In this note several standard IP will be considered and logic cuts and other valid inequalities will be derived from their models. The problem types considered in sections 2-4 are (a) bin-packing problems (b) capacitated clustering problems and (c) generalised assignment problems. In each section results on a simple case of the general problem will be given and extensions to other cases outlined. For solving many of these problems the branch and bound method of IP is not always favoured and heuristic approaches are used. However, as many heuristic approaches make use of LP relaxations and partial use of branch and bound obtaining a tighter LP formulation and aiding the branching process are both desirable features.

2. Bin-Packing Problems

2.1 The Standard Bin-Packing Problem

The bin-packing problem can be stated as, see for instance Martello and Toth (1990):

given n items and n bins, with

w_j = weight of item j

c = capacity of each bin,

assign each item to exactly one bin so that the total weight of the items in each bin is less than or equal to c and the number of bins used is a minimum i.e.

$$\text{minimise} \quad \sum_{i=1}^{n} y_i \tag{2.1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_{ij} \le c y_i \qquad i \in N = \{1, \dots, n\}, \tag{2.2}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j \in N, \tag{2.3}$$

$$y_i = 0 \text{ or } 1 \qquad i \in N \tag{2.4}$$

$$x_{ij} = 0 \text{ or } 1 \qquad i \in N, j \in N, \tag{2.5}$$

where $y_i = 1$ if bin i is used;
0 otherwise,

$x_{ij} = 1$ if item j is assigned to bin i;
0 otherwise,

$$0 \le w_j \le c, \; j \in N, c > 0. \tag{2.6}$$

Example 2.1

If n=4 and one constraint from (2.2) is

$$2x_{11} + 3x_{12} + 4x_{13} + 4x_{14} \leq 7y_1 \qquad (2.7)$$

then a logic cut is

$$y_1 \leq x_{11} + x_{12} + x_{13} + x_{14}.$$

Further cuts, which are not logic cuts, are

$$x_{11} \leq y_1, x_{12} \leq y_1, x_{13} \leq y_1, x_{14} \leq y_1. \qquad (2.8)$$

Such cuts can be added to the existing formulation.

In general, the following cuts are available for the bin-packing problem:

$$y_i \leq \sum_{j=1}^{n} x_{ij} \qquad i \in N, \qquad (2.9)$$

and $x_{ij} \leq y_i \qquad i \in N, j \in N.$ (2.10)

Example 2.2

An example was generated with n = 50 and the coefficients w_j chosen randomly from the uniform integer distribution [1,10] and $c = \sum_{j=1}^{n} w_j / 12.0$.

When cuts of the form (2.9) were added to the problem the following results were obtained using the optimisation system Sciconic.

	LP value	LP nits	nodes to opt	total nodes	total time
without cuts	12.565	363	123	231	102.8
with cuts	12.565	425	84	147	91.5

key:

nits = simplex iterations

opt = optimal solution.

All time are in CPU seconds on a Honeywell-9000.

As can be seen from this example, although the number of simplex iterations increases, the number of branch and bound nodes is reduced substantially and the total time taken drops moderately. When cuts of the type (2.10) are added, the situation becomes worse, indicating that the rise in overhead caused by the cuts at the LP relaxation stage is not worth carrying. In such circumstances it may be better to generate these cuts only when they are violated, as can be done easily with certain optimisation software such as XPRESS-MP. (See for instance Brailsford et al. (1995)).

Other variants of the bin-packing problem exist e.g. $w_j = w'_{ij}$ (differential weights on bins) and $c = c'_i$ (bins of different sizes). In these cases conditions such as $w'_{ij} \le c'_i$, $i \in N$, $j \in N$, will be unlikely to hold but will be replaced by conditions like

for each $j \in N$, there exists an $i \in N$, such that $w'_{ij} \le c'_i$.

Cuts can then be developed analogously with (2.9) and (2.10). A further generalisation involves replacing the right hand side of (2.2) with

$$\sum_{k=1}^{L} c y_{ik}$$

and the y_i variables with y_{ik} $(k = 1, \dots, L)$ in (2.2), (2.4) and (2.6) and the term 'bin' with the term 'bin-group' in (2.6). The cuts (2.9) are replaced by

$$\sum_{k=1}^{L} y_{ik} \le \sum_{j=1}^{n} x_{ij}$$

and the cuts (2.10) by

$$x_{ij} \le \sum_{k=1}^{L} y_{ik}$$

2.2 The Bin-Unpacking Problem

Associated with the problem given by (2.1)-(2.6) there will be a complementary problem. This may be stated as, place the maximum number of bins in the space occupied by the items (where the bins and items are as defined in Section 2.1). For convenience this problem will be named the **bin-unpacking problem**. Mathematically the problem may be stated as:

$$\text{maximise} \quad \sum_{i=1}^{n} y_i \tag{2.11}$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_{ij} \ge c y_i \qquad i \in N \tag{2.12}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j \in N, \tag{2.13}$$

$$y_i = 0 \text{ or } 1 \qquad i \in N \tag{2.14}$$

$$x_{ij} = 0 \text{ or } 1 \qquad i \in N, j \in N, \tag{2.15}$$

$$w_j \geq 0\,,\; j\,\varepsilon\, N,\; \sum_{j=1}^{n} w_j \geq c\,,\; c > 0. \qquad (2.16)$$

This problem is introduced to demonstrate that the cuts (2.9) and (2.10) will be reversed in role for the problem given by (2.11)-(2.16). The cuts (2.9) and (2.10) are valid for the problem given by (2.11)-(2.16), but this time (2.10) are logic cuts and (2.9) are not. If the logic cuts (2.10) are violated by the LP relaxation of (2.11)-(2.16) then the non-zero x variables would be "reallocated" into constraints where y > 0 held in the solution. Thus if x is non-zero it occurs in a constraint where y is non-zero.

2.3 The Capacitated Plant Location Problem

A problem related to the bin-packing problem is the capacitated plant location problem, the problem of deciding which plants to open to satisfy customers and assigning each customer to a plant-unit of a plant. Leung and Magnanti (1989) provide a vertex packing formulation of the constraints of the problem. The formulation is:

$y_j = 1$ if plant j is open;
0 otherwise,

and $x_{ijk} = 1$ if customer i is assigned to unit k of plant j;
0 otherwise,

with the following constraints:

$$\sum_{j\varepsilon N}\sum_{k\varepsilon K_j} x_{ijk} \leq 1 \qquad \forall i\varepsilon\ M, \qquad (2.17)$$

$$\sum_{i\varepsilon M} x_{ijk} \leq y_j \qquad \forall j\varepsilon\ N,\ \forall k\varepsilon\ K_j, \qquad (2.18)$$

$$\sum_{k\varepsilon K_j} x_{ijk} \leq y_j \qquad \forall i\varepsilon\ M,\ \forall j\varepsilon\ N, \qquad (2.19)$$

$$y_j\ \varepsilon\ \{0,1\} \qquad \forall j\varepsilon\ N, \qquad (2.20)$$

$$x_{ijk}\ \varepsilon\ \{0,1\} \qquad \forall i\varepsilon\ M,\ \forall j\varepsilon\ N,\ \forall k\varepsilon\ K_j \qquad (2.21)$$

where $M = \{1, \ldots ,m\}$, $N = \{1, \ldots ,n\}$, $K_j = \{1, \ldots k_j\}$.

Leung and Magnanti (1989) derive facets for this problem. Logic cuts of the form

$$\sum_{i \in M} \sum_{k \in K_j} x_{ijk} \geq y_j \qquad \forall j \varepsilon\ N, \qquad (2.22)$$

can also be added to the formulation given by (2.17)-(2.21).

3. Capacitated Clustering Problems

This section will discuss the capacitated clustering problem and the related p-median problem.

3.1 The Capacitated Clustering Problem

The capacitated clustering problem (CCP) is:

allocate n points uniquely to p out of m clusters such that the capacity of each cluster is not violated and the points are allocated to maximise the homogeneity of points within the cluster and, at the same time, the heterogeneity of the points between clusters (see for instance Mulvey and Peck (1984), Osman and Christofides (1994)). The problem may be formulated as:

$$\text{minimise} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} \qquad (3.1)$$

$$\text{subject to} \quad \sum_{i=1}^{m} w_i x_{ij} \leq W_j \qquad j\ \varepsilon\ N = \{1, \ldots ,n\}, \qquad (3.2)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i\ \varepsilon\ M = \{1, \ldots ,m\}, \qquad (3.3)$$

$$\sum_{i=1}^{m} y_i = p \qquad (3.4)$$

$$x_{ij} \leq y_j \qquad i\ \varepsilon\ M, j\ \varepsilon\ N, \qquad (3.5)$$

$$y_i = 0 \text{ or } 1 \qquad i\ \varepsilon\ M, \qquad (3.6)$$

$$x_{ij} = 0 \text{ or } 1 \qquad i\ \varepsilon\ M, j\ \varepsilon\ N, \qquad (3.7)$$

where $y_i = 1$ if point i is the centre of a cluster;
0 otherwise,

$x_{ij} = 1$ if point j is assigned to cluster i;
0 otherwise,

d_{ij} = distance between point i and point j,

w_i = weight of point i if allocated to cluster j

W_j = capacity of cluster j.

This problem has similarities to the bin-packing problem. Logic cuts may be deduced for the problem given by (3.1)-(3.7) as follows:

$$\sum_{i=1}^{m} x_{ij} \geq y_j \qquad j \in N. \qquad (3.8)$$

Example 3.1

A problem was generated with m =10 and n = 20 for the constraints given by (3.1)-(3.7) and w_{ij} = 1 (i ε M, j ε N), W_j = 5 (j ε N), and p = 5, and then values were chosen from the random integer distribution [1,9] for d_{ij}. The problem was solved without cuts and then the cuts (3.8) were added.

	LP value	LP nits	nodes to opt	total nodes	total time
without cuts	39.0	68	3	5	0.88
with cuts	39.0	63	1	1	0.68

Thus the addition of logic cuts to the formulation aids convergence for this problem.

3.2 The p-Median Problem

The p-median problem, see for instance Christofides and Beasley (1982), is the problem of locating p facilities (medians) on a network so as to minimise the sum of all the distances d_{ij} from each vertex i to its nearest facility j. This problem is clearly similar to CCP. It can be formulated as:

$$\text{minimise} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \qquad (3.9)$$

$$\text{subject to} \quad \sum_{i=1}^{n} x_{ij} \leq (n-1)x_{ii} \qquad i \in M = \{1, \ldots, n\}, \qquad (3.10)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i \in M \qquad (3.11)$$

$$\sum_{i=1}^{n} x_{ii} = p \qquad (3.12)$$

$$x_{ij} = 0 \text{ or } 1 \qquad i \in N, j \in N. \qquad (3.13)$$

Cuts of the form

$$\sum_{j=1}^{n} x_{ij} \geq x_{ii} \qquad (3.14)$$

may be added to the formulation given by (3.9)-(3.13).

4. Generalised Assignment and Related Problems

4.1 The Generalised Assignment Problem

The generalised assignment problem can be stated as, see for instance Martello and Toth (1990):

given n items and m containers with

p_{ij} = profit of item j if assigned to container i,

w_{ij} = weight of item j if assigned to container i,

c_i = capacity of container i,

assign each item to exactly one container so as to maximise the total profit assigned, without assigning to any container a total weight greater than its capacity i.e.

$$\text{maximise} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij} \qquad (4.1)$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_{ij} x_{ij} \leq c_i \qquad i \varepsilon\ M = \{1, \ldots ,m\}, \qquad (4.2)$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad j \varepsilon\ N, \qquad (4.3)$$

$$x_{ij} = 0 \text{ or } 1 \qquad i \varepsilon\ M, j \varepsilon\ N,$$

x_{ij} = 1 if item j is assigned to container i;

0 otherwise. (4.4)

Note: In some versions of the problem, if $w_{i'j'} = 0$ for some i',j' in the above then it is assumed that $x_{i'j'}$ is not a variable.

Cuts can be generated for the problem given by (4.1)-(4.4) as is illustrated in the following small example.

Example 4.2

Consider an instance of (4.1)-(4.4) with m =3 and n = 4 where the constraints (4.2) are

$$2x_{11} + x_{12} + 2x_{13} + 2x_{14} \leq 3 \tag{4.5}$$

$$2x_{21} + x_{22} + 2x_{23} + 2x_{24} \leq 3 \tag{4.6}$$

$$x_{31} + 3x_{32} + 2x_{33} + 3x_{34} \leq 5 \tag{4.7}$$

Adding constraints (4.5) and (4.6) together gives

$$2x_{11} + 2x_{21} + x_{12} + x_{22} + 2x_{13} + 2x_{23} + 2x_{14} + 2x_{24} \leq 6 \tag{4.8}$$

and then it follows that

$$x_{11} + x_{21} + x_{12} + x_{22} + x_{13} + x_{23} + x_{14} + x_{24} \leq 3 \tag{4.9}$$

as constraints (4.3) must hold.

Using (4.3) and (4.9) we can deduce

$$x_{31} + x_{32} + x_{33} + x_{34} \geq 1 . \tag{4.10}$$

Using the objective function given by $p_{ij} = 6.0$ ($i < 3$, $j=1,2,3,4$), $p_{ij} = 1.0$ ($i = 3$, $j=1,2,3,4$) in (4.1) with the constraints (4.3)-(4.7), the LP relaxation of the problem has the solution $x_{11} = x_{13} = x_{21} = x_{33} = 0.5$, $x_{12} = x_{24} = 1.0$, all other variables equal to zero. The cut (4.10) is clearly violated by this solution. Solving the problem with and without the cut yields the following results:

	total nodes	total time
without cut	18	0.18
with cut	1	0.02

More generally the cuts are derived as follows:

for each $i' \varepsilon$ M, let N_i be the set of values of j such that $w_{rj} > 0$ iff $j \varepsilon N_i$

Form the constraint

$$\sum_{\substack{i \in M \\ i \neq r}} \sum_{j \in N_i} w_{ij} x_{ij} \leq \sum_{\substack{i \in M \\ i \neq r}} c_i - \sum_{\substack{i \in M \\ i \neq r}} \sum_{j \in N \setminus N_i} w_{ij} \tag{4.11}$$

and solve the problem

$$\text{maximise } z_r = \sum_{\substack{i \in M \\ i \neq r}} \sum_{j \in N_i} x_{ij} \tag{4.12}$$

subject to (4.3), (4.4) and (4.11), then provided $z_r < n$, the cut

$$\sum_{j=1}^{n} x_{i'j} \geq n - z_{i'} \tag{4.13}$$

is non-trivial. (The second term on the right hand side of (4.11) is included to conform with the note following (4.4).)

Note that (4.13) is not the strongest cut of its type available, but is introduced because solving the problem given by (4.3), (4.4), (4.11) and (4.12) can be performed by a simple inspection. Gottlieb and Rao (1990a), (1990b) propose procedures for generating facets for the generalised assignment problem which will in general be more time consuming than the above, but produce stronger cuts. However, considering two examples used by them in Gottlieb and Rao (1990a) it is found that useful cuts of form (4.13) can be generated as the following shows.

Example 4.3 (from Gottlieb and Rao (1990a))

Let $m = 5, n = 5,$

$(w_{11}, w_{12}) = (1,1), c_1 = 1,$

$(w_{22}, w_{23}, w_{25}) = (1,1,2), c_2 = 2,$

$(w_{33}, w_{34}, w_{36}) = (1,1,2), c_3 = 2,$

$(w_{41}, w_{44}, w_{47}) = (1,1,2), c_4 = 2,$

$(w_{55}, w_{56}, w_{57}) = (2,1,1), c_5 = 2.$

Then $x_{55} + x_{56} + x_{57} \geq 2$ is a valid cut which cuts off the solution $x_{55} = x_{56} = x_{57} = 0.5$ as required by Gottlieb and Rao (1990a).

Example 4.4 (from Gottlieb and Rao (1990a))

Let $m = 4, n = 8,$

$(w_{11}, \ldots, w_{15}) = (1,1,1,1,1), c_1 = 2,$

$(w_{21}, w_{22}, w_{26}, w_{27}) = (1,1,2,2), c_2 = 2,$

$(w_{33}, w_{34}, w_{36}, w_{37}, w_{38}) = (1,1,2,2,2), c_3 = 4,$

$(w_{45}, w_{47}, w_{48}) = (1,1,1), c_4 = 2.$

Then $x_{45} + x_{47} + x_{48} \geq 2$ is a valid cut which cuts off the solution $x_{45} = x_{47} = x_{48} = 0.6$ as required by Gottlieb and Rao (1990a).

4.2 The Multiple Knapsack Problem

The multiple knapsack problem, see for instance Martello and Toth (1990), has similarities to the generalised assignment problem. Its principal difference from the problem given by (4.1)-(4.4) is that the equations (4.3) are replaced by inequalities

$$\sum_{i=1}^{m} x_{ij} \le 1 \qquad j \in N = \{1, \ldots ,n\}. \qquad (4.14)$$

A further modification is ususally made, namely replacing (4.2) with

$$\sum_{j=1}^{n} w_j x_{ij} \ge c_i \qquad i \in M = \{1, \ldots ,m\}. \qquad (4.15)$$

For the resulting problem cuts analogous to (4.12) to be derived.

Example 4.3

If m = 3 and n = 4 and the constraints (4.15) are given by

$$2x_{11} + x_{12} + 2x_{13} + 2x_{14} \ge 2$$
$$2x_{21} + x_{22} + 2x_{23} + 2x_{24} \ge 2$$
$$x_{31} + 3x_{32} + 2x_{33} + 3x_{34} \ge 5$$

then a cut can be derived by first solving the problem

minimise $z_3 = x_{11} + x_{12} + x_{13} + x_{14} + x_{21} + x_{22} + x_{23} + x_{24}$

subject to $2x_{11} + x_{12} + 2x_{13} + 2x_{14} + 2x_{21} + x_{22} + 2x_{23} + 2x_{24} \ge 4$,

(4.4) and (4.14).

This has solution $z_3 = 2$ and so the inequality

$x_{31} + x_{32} + x_{33} + x_{34} \le 2$ is valid.

A procedure analogous to that given by (4.11)-(4.13) can be developed for the multiple knapsack problem.

5. Conclusions

A series of standard problems involving linear 0-1 inequalities has been considered. It has been shown how cuts of two particular types may be added to such problems, taking advantage of the presence of interlinked 0-1 variables. The cuts have been demonstrated to have potential usefulness in aiding convergence of the branch and bound solution process.

References

S. C. Brailsford, P. M. Hubbard, B. M. Smith and H. P. Williams (1995) The progressive party problem: a difficult problem of combinatorial optimization, Presented at APMOD95, Brunel University, England, 1995.

N. Christofides and J.E. Beasley (1982) A tree search algorithm for the p-median problem, European Journal of Operational Research 10, 196-204.

E.S. Gottlieb and M.R. Rao (1990a) The generalized assignment problem: valid inequalities and facets, Mathematical Programming 46, 31-52.

E.S. Gottlieb and M.R. Rao (1990b) (1,k)-Configuration facets for the generalized assignment problem, Mathematical Programming 46, 53-60.

J. N. Hooker (1992) Generalized resolution for 0-1 linear inequalities, Annals of Mathematics and Artificial Intelligence 6, 271-286.

J. N. Hooker, H. Yans, I. E. Grossman and R. Raman (1994), Logic cuts for processing networks with fixed charges, Computers and Operations Research 21, 265-279.

J. N. Hooker (1995) Logic-based methods for engineering design problems, Presented at APMOD95, Brunel University, England.

J.M.Y. Leung and T.L. Magnanti (1989) Valid inequalities and facets of the capacitated plant location problem, Mathematical Programming 44, 271-291.

J.M. Mulvey and M.P. Beck (1984) Solving capacitated clustering problems, European Journal of Operational Research 18,339-348.

I.H. Osman and N. Christofides (1994) Capacitated clustering problems by hybrid simulated annealing and tabu search, International Transactions in Operational Research 1, 317-336.

Sciconic/VM, EDS-Scicon, Milton Keynes, England.

J. M. Wilson (1995) Problem specification in constraint logic programming and integer programming, Presented at INFORMS95, Singapore.

XPRESS-MP, Dash Associates, Blisworth, Northamptonshire.