THE REFORMULATION OF TWO MIXED INTEGER PROGRAMMING PROBLEMS

H.P. WILLIAMS

University of Edinburgh, Scotland, UK

Received 27 October 1975
Revised manuscript received 24 October 1977

Two practical problems are described, each of which can be formulated in more than one way as a mixed integer programming problem. The computational experience with two formulations of each problem is given. It is pointed out how in each case a reformulation results in the associated linear programming problem being more constrained. As a result the reformulated mixed integer problem is easier to solve. The problems are a multi-period blending problem and a mining investment problem.

Key words: Integer Programming, Formulation, Models.

1. Introduction

In [1] Williams describes how five practical integer programming problems can all be reformulated to make the associated linear programming problem more constrained. As a result the reformulated integer programming problem is easier to solve using the Branch and Bound Algorithm. All five of those problems can be regarded as pure 0–1 integer programming problems. The purpose of this paper is to show how such ideas can be extended to two mixed integer programming problems.

Firstly a multiperiod blending problem is considered. It is wished to impose extra logical restrictions on the combinations of ingredients which can be included in any period’s blend. These extra restrictions are imposed by adding 0–1 variables to the original linear programming model. The logical restrictions can then be modelled by inequalities involving the 0–1 variables. Alternative ways of modelling these logical restrictions lead to the different formulations.

Secondly a mining problem involving decisions over which of a number of mines to work in which of future years is considered. The ores mined in each year are blended together. In the first formulation a mixed integer model with both 0–1 and general integer variables is generated. A reformulation leads to a more easily solvable mixed integer model with 0–1 variables only. It is shown that this second formulation has a more constrained associated linear programming model. The fact that this is the case is not immediately apparent.

Both of these problems are based on much larger original problems. They
have been reduced in size to facilitate experimentation but the essential structures of the original problems have been preserved.

Many practical Operational Research workers would agree on the need for a wider understanding of modelling principles in integer programming. One of the purposes of this paper, as with (1), is to present easily understood, but practical, problems with which to experiment.

2. Problem 1, Multiperiod Blending

Five ingredients are to be blended together to produce a given product in each of six successive months. The proportions in which the ingredients can be blended are restricted by processing and quality restrictions. For each period of the original problem the following continuous variables and constraints appear.

- 5 ingredient variables,
- 1 product variable,
- 4 blending and processing constraints,
- 1 material balance constraint.

To obtain a multiperiod model the ingredient variables are distinguished into three categories, buying, using and storing. Consecutive months are linked together by the following relations for each ingredient:

\[
\text{Quantity put into store in month } t - 1 = \text{Quantity used in month } t + \text{Quantity bought in month } t + \text{Quantity put into store in month } t
\]

The objective, to be maximised, involves negative (cost) coefficients associated with the buying and storing variables and positive (profit) coefficients associated with the product variables.

It is now necessary to impose the extra logical restrictions for each month:

- No more than 3 out of the 5 ingredients can be used. \(1\)
- If ingredient \(i\) is used at least \(m_i\) tons must be used. \(2\)
- If either ingredient 1 or ingredient 2 (or both) are used, then ingredient 5 must also be used. \(3\)

For each period:

- \(x_i\) = Quantity of ingredient \(i\) used;
- \(\delta_i = 1\) if ingredient \(i\) used, \(= 0\) otherwise.

The 0–1 "indicator" variables \(\delta_i\) are linked to the corresponding \(x_i\) variables by