KKU Engineering Journal

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Alternative modeling schemes for propositional calculus problems

Anupama Chanda*[1)], Bijan Sarkar[1)] and R.N. Mukherjee[2)]

[1)]Department of Production Engineering, Jadavpur University, Kolkata, West Bengal 700032, India
[2)]Department of Mathematics (Retd.) Burdwan University, Rajbati, Burwan, West Bengal: 713104, India

Received December 2015
Accepted April 2016

Abstract

The main objective of this paper is to discuss alternative approaches to convert a propositional logic problem into a 0-1 integer program. The substitution method proposed by Williams and the CNF method is used for modeling and representation of a logical inference problem. The similarity between the two fields, viz. optimization problems or mathematical programming and deductive logic are shown.

Keywords: CNF, Logical inference, Mathematical programming, 0-1 integer programming, Propositional logic

1. Introduction

It is well known that the conventional logical methods have failed to solve large inference problems. Thus, it has become important to develop an efficient method of performing deductive reasoning. Significant contributions have been made by various authors in this field. Cavelier, Pardalos and Soyster [1] have used mathematical programming for modeling a propositional calculus problem. Similarly Yager [2] also suggested means to solve a logical inference problem via the use of mathematical programming. Peysakh [3] presented an algorithm for converting any Boolean expression into CNF. Hooker [4-5] has shown the application of quantitative methods to logic have led to fast inference methods. The connection between the methods of computational logic and integer programming was shown by Williams [6] in his paper. He used IP model to overcome the computational problems that arise in logic. Blair, Jeroslow and Lowe [7] suggested ways to resolve Boolean logical problems. He related the mathematical programming methods for Boolean logic or propositional calculus problem to the Davis- Putnam method for proving theorems. The CNF method [4, 7], Williams [8] linear 0-1 method and Cavalier and Soyster [1] were compared with 0-1 polynomial methods. According to Zangwill [9], any 0-1 polynomial program can be replaced with a 0-1linear program by substituting the product of 0-1 variables with a new 0-1 variable.

This paper discusses different approaches to transform a propositional calculus problem into 0-1 integer programming using substitution method [8] and CNF method [4, 7]. The basic modeling approach and solution procedure to a logical inference problem given by [10] are used in our problem. Moreover similarity between mathematical programming and deductive reasoning has also been discussed. The main motivation of the paper is to present in such a way that some applied problem may be attempted for solving.

2. General formulation of a propositional logic and its transformation

A propositional logic problem of the form: ‘If P_1 and P_2 and $\cdots$ and P_n, then Q’

$$\begin{array}{ll}\text{Prove } Q & \\ \text{Given: } & P_1 \\ & P_2 \\ & \vdots \\ & P_{n,} \end{array} \qquad (1)$$

where each of $P_1, P_2, \dots, P_n$ is a disjunction of literals of the form $P_i = \vee D_i, i \in I$.

One can transform problem (1) into a 0-1 integer program [11] of the form:

$$\begin{array}{lll}\text{Optimize} & cx & \\ \text{Subject to} & Ax \le b & \\ & x_j \in \{0,1\}, \quad for\ j = 1, \dots, n. & \end{array} \qquad (2)$$

The solution to problem (2) will help to resolve problem (1) and shows whether Q is true given the set of boolean constraints. Hence the solution to the 0-1 linear integer program can be used to resolve the issues of problem (1).

*Corresponding author.
Email address: anupama.chanda@gmail.com

doi: 10.14456/kkuenj.2016.22

3. Different methods of representation of logical inference problem

Here we use the two approaches namely substitution method and conjunctive normal form (CNF) to formulate a propositional calculus problem into a 0-1 integer programming problem and subsequently solve the mathematical problem to arrive at a solution to the related logical problem.

4. Substitution method

William [12] suggested a mathematical modeling technique to convert a boolean logic expression into 0-1 linear program. It is known as substitution method. This method requires an additional 0-1 variable y. $Y = D_1 \wedge D_2$ can be converted into a 0-1 form using the following transformation:

$$\begin{array}{lcl} d_1 + d_2 - y & \leq & 1 \\ d_1 - y & \geq & 0 \\ d_2 - y & \geq & 0 \\ d_1, d_2, y & = & 0,1. \end{array} \tag{3}$$

Similarly, it can be shown that $D_1 \vee D_2$ is equivalent to

$$\begin{array}{lcl} d_1 + d_2 - y & \geq & 0 \\ d_1 - y & \leq & 0 \\ d_2 - y & \leq & 0 \\ d_1, d_2, y & = & 0,1. \end{array} \tag{4}$$

The set of constraints (3) and (4) are basic pillars to convert any boolean logic into a 0-1 linear form.

5. Conjunctive normal form

Consider another approach using Conjunctive normal form (CNF) [4, 7]. A boolean expression is expressed as a conjunction of disjunctions called the conjunctive normal form (CNF). CNF can be written as:

$$\wedge_{j=1}^{k} \vee_{i \in p_j} d_i, \tag{5}$$

where each d_i is either the attribute D_i or its negation $\overline{D}_i$, k is the number of attribute disjunctions and p_i is the j-th index set for j-th attribute disjunction.

For example, $Z = (D_1 \vee \overline{D}_2 \vee D_4) \wedge (D_2) \wedge (\overline{D}_1 \vee D_3)$.

Since Z is true then each of the conjunctive clauses must be true i.e.

$$\begin{array}{lcl} d_1 + (1 - d_2) + d_4 & \geq & 1 \\ d_1 & \geq & 1. \end{array}$$

6. An illustrative example [13]

Either taxes are raised or if expenditures increases, then the debt limit is raised. If taxes are raised, then the cost of tax collection increases. If an increase in expenditures implies that the government would have to borrow more money, then if the debt limit is raised, then interest rates goes up. If taxes are not raised and the cost of tax collection does not increase, then if the debt limit is raised, then government would have to borrow more money. The cost of tax collection does not increase. Either interest rates does not goes up or the government would not have to borrow more money. Therefore, either the debt limit is not raised or expenditures do not increase, thus:

D_1: Taxes are raised.
D_2: Expenditures increases.
D_3: The debt limit is raised.
D_4: The cost of tax collection increases.
D_5: The government would have to borrow more money.
D_6: Interest rates go up.

$$\begin{array}{lcl} (D_1 \vee D_2) & \rightarrow & D_3 \\ D_1 & \rightarrow & D_4 \\ (D_2 \rightarrow D_5) & \rightarrow & (D_3 \rightarrow D_6) \\ \overline{D}_1 \wedge \overline{D}_4 & \rightarrow & (D_3 \rightarrow D_5) \\ \overline{D}_4 & & \\ \overline{D}_5 \vee \overline{D}_6 & & \\ \hline \overline{D}_2 \vee \overline{D}_3, & & \end{array} \tag{6}$$

where the horizontal line signifies "therefore".
Let us introduce additional variables.

$d_{12} = D_1 \vee D_2$, $d_{25} = D_2 \rightarrow D_5$, $d_{36} = D_3 \rightarrow D_6$, $d_{14} = \overline{D}_1 \wedge \overline{D}_4$ and $d_{35} = D_3 \rightarrow D_5$.

The substitution method gives:

$$\begin{array}{ll} \text{Optimize} & \bar{d}_2 \wedge \bar{d}_3 \\ \text{Subject to} & d_{12} \rightarrow d_3 \\ & d_1 \rightarrow d_4 \\ & d_{25} \rightarrow d_{36} \\ & d_{14} \rightarrow d_{35} \\ & \bar{d}_4 \\ & \bar{d}_5 \vee \bar{d}_6 \end{array} \tag{7}$$

Using (3) and (4)

$$\begin{array}{lll} \text{Optimize} & d_2 + d_3 & \leq 1 \\ \text{Subject to} & & \\ & d_1 + d_2 - d_{12} & \geq 0 \\ & d_1 - d_{12} & \leq 0 \\ & d_2 - d_{12} & \leq 0 \\ & d_{12} - d_3 & \leq 0 \\ & d_4 - d_1 & \geq 0 \\ & d_2 - d_5 + d_{25} & \leq 1 \\ & d_2 + d_{25} & \geq 1 \\ & d_{25} - d_5 & \geq 0 \\ & d_3 - d_6 + d_{36} & \leq 1 \\ & d_3 + d_{36} & \geq 1 \\ & d_{36} - d_6 & \geq 0 \\ & d_{25} - d_{36} & \leq 0 \\ & d_1 + d_4 + d_{14} & \geq 1 \\ & d_1 + d_{14} & \leq 1 \\ & d_4 + d_{14} & \leq 1 \\ & d_3 - d_5 + d_{35} & \leq 1 \\ & d_3 + d_{35} & \geq 1 \\ & d_5 - d_{35} & \leq 0 \\ & d_{14} - d_{35} & \leq 0 \\ & d_4 & \leq 0 \\ & d_5 + d_6 & \leq 1 \\ & d_i & = 0,1. \end{array} \tag{8}$$

Let us now converting the problem to CNF.

$$\begin{aligned}
\text{Prove:}\quad & \bar{D}_2 \vee \bar{D}_3 \\
& \bar{D}_1 \vee D_3 \\
& \bar{D}_2 \vee D_3 \\
& \bar{D}_1 \vee D_4 \\
& D_2 \vee \bar{D}_3 \vee D_6 \\
& \bar{D}_3 \vee \bar{D}_5 \vee D_6 \\
& D_1 \vee \bar{D}_3 \vee \bar{D}_4 \vee D_5 \\
& \bar{D}_4 \\
& \bar{D}_5 \vee \bar{D}_6.
\end{aligned} \tag{9}$$

Williams [8] compares the two methods and concludes that on one hand CNF technique requires the propositional calculus problem to have a special structure on the other hand substitution method requires additional variables.

The corresponding linear 0-1 form:

$$\begin{aligned}
\text{Optimize}\quad & d_2 + d_3 && \le 1 \\
\text{Subject to}\quad & d_1 - d_3 && \le 0 \\
& d_1 - d_3 && \le 0 \\
& d_2 - d_3 && \le 0 \\
& d_1 - d_4 && \le 0 \\
& d_2 - d_3 + d_6 && \ge 0 \\
& d_3 + d_5 - d_6 && \le 1 \\
& d_1 - d_3 + d_4 + d_5 && \ge 0 \\
& d_4 && \le 0 \\
& d_5 + d_6 && \le 1 \\
& d_i && = 0, 1.
\end{aligned} \tag{10}$$

Here the minimal value of objective function is 0. It can be seen that CNF formulation of a propositional problem is compact as compared to the substitution method. However this advantage depends on the type of propositional calculus problem [1].

7. Conclusions

Here CNF method and substitution method have been illustrated by suitable numerical example. Moreover the example further emphasize Williams conjecture i.e. substitution method requires additional variable whereas CNF method requires a special structure. But both the method yields the same result.

8. References

[1] Cavalier T, Soyster A. Logical deduction via linear programming. IMSE Working Paper 87-147. Department of Industrial and Management Systems Engineering, Pennsylvania State University; 1987.

[2] Yager RR. A mathematical programming approach to inference with the capability of implementing default rules. Int J Man Mach Stud. 1988;29(6):685-714.

[3] Peysakh J. Fast algorithm to convert boolean expression into CNF. Technical Report IBM Computer Science RC 12913 (No. 57971). Watson, NY; 1987.

[4] Hooker JN. A quantitative approach to logical inference. Decis Support Syst. 1988;4(1):45-69.

[5] Hooker JN. Generalized resolution and cutting planes. Ann Oper Res. 1988;12(1):217-39.

[6] Williams HP. Logic applied to integer programming and integer programming applied to logic. Eur J Oper Res. 1995;81(3):605-16.

[7] Blair CE, Jeroslow RG, Lowe JK. Some results and experiments in programming techniques for propositional logic. Comput Oper Res. 1986;13(5): 633-45.

[8] Williams HP. Linear and integer programming applied to the propositional calculus. Syst Res Inform Sci. 1987;2:81-100.

[9] Zangwill WI. (1965). Media selection by decision programming. In: Ursula HF, editor. Mathematical models in Marketing. Berlin: Springer; 1976. p. 132-3.

[10] Cavalier TM, Pardalos PM, Soyster AL. Modeling an integer programming techniques applied to propositional calculus. Comput Oper Res. 1990;17(6):561-70.

[11] Chanda A, Sarkar B, and Mukherjee RN. Logical deduction for an integer programming problem and Integer Programming (IP) formulation for a logical problem. Int J Oper Res Nepal. 2015;4(1):33-40.

[12] Williams HP. Model building in mathematical programming. Chichester: John Wiley & Sons; 2013.

[13] Copi IM. Symbolic Logic. New Delhi: Prentice Hall of India Private Limited; 2006.