

擬似ブール関数およびブール関数の 数式処理システム (BALOC)

名古屋大学 吉田 雄二

§1. まえがき.

整数計画問題の定式化の一方法である擬似ブール計画法は
扱える問題の範囲が極めて広く、種々の応用が期待される。
しかし、これ迄に開発されている種々の解法は、^{(1), (2), (3)}
その多くがブール関数の数式処理に負う所が大きく、現在の
電子計算機のプログラミングシステムではコーディングの困
難なものになっている。

筆者は整数計画問題への近接の一方法として、擬似ブール
関数およびブール関数の数式処理のためのプログラミングシ
ステムの開発を試みている。

§2. 擬似ブール計画法とその解法

$(x_1, \dots, x_n)$ を 0 または 1 をとる N 個の 2 値変数とし、
 $f_i(x_1, \dots, x_n)$ ($i=1, \dots, k$) をこれらを変数とする N 変
数ブール関数とする。このとき、

$$F(x_1, \dots, x_N) = \sum_{i=1}^K a_i f_i(x_1, \dots, x_N) \quad (1)$$

(a_i : 実数)

のように定義される関数 $F(x_1, \dots, x_N)$ を N 変数擬似ブール関数と呼ぶ。一般に、擬似ブール計画法の問題は、2つの擬似ブール関数、 $F(x_1, \dots, x_N)$, $G(x_1, \dots, x_N)$ をもちいて次のように定式化される。

「 $G(x_1, \dots, x_N) \geq 0$ をみたす $x_1, \dots, x_N$ の値の組合せのうちで、 $F(x_1, \dots, x_N)$ を最小にする組合せを求めよ。」

この問題の解法は次のようにして与えられる。

初めに、 $G(x_1, \dots, x_N)$ をみたす x_1 を次のような形で求める。

$$x_1 = u_1 \varpi_{11}(x_2, \dots, x_N) + \varpi_{12}(x_2, \dots, x_N) \quad (2)$$

u_1 : 0 または 1 をとる任意パラメータ。

(2) を $F(x_1, \dots, x_N)$ の x_1 に代入する。 $F(u_1) \geq F(u_1)$ の条件より $F(x_1, \dots, x_N)$ を最小にする u_1 を次の様な形で求める。

$$u_1 = v_1 \varpi_{11}(x_2, \dots, x_N) + \varpi_{12}(x_2, \dots, x_N) \quad (3)$$

v_1 : 0 または 1 をとる任意パラメータ。

(3) を $F(x_1, \dots, x_N)$, $G(x_1, \dots, x_N)$ に代入して v_1 を適当な値にきめれば、変数 x_1 に関係しない2つの関数 $F_2(x_2, \dots, x_N)$, $G_2(x_2, \dots, x_N)$ がえられる。

F_2, G_2 について同様の手順で x_2 を求め、以下順に、 $x_3, \dots$, x_n を求めれば問題の条件を満たす解 $(x_1, \dots, x_n)$ が $u_1, \dots, u_n, v_1, \dots, v_n$ をパラメータとして与えられるから、これらに適当に値を代入することにより全ての解が与えられる。

以上の手順は擬似ブール関数およびブール関数の数式処理に負う所が極めて大きく、現在の計算機のプログラミングシステム、例えば FORTRAN でコーディングする事は極めて困難である。そこで筆者は、次節以下に述べるようなプログラミングシステムの開発を行なった。なお、ここで述べるシステムはすべて BALOC という名前と数字の組合わせで呼ばれるが、前者は Boolean Algorithm Oriented Compiler の略であり、後者はシステムを区別する番号である。

§3. 擬似ブール関数の数式処理システム.

筆者はまず初めに、擬似ブール関数の数式処理を行なうためのシステムを構成した。BALOC-0 および BALOC-1 と呼ぶものがそれである。前者はマクロアセンブラレベルの言語で、後者は BALOC-0 を基礎にしたコンパイラレベルのものである。BALOC-0 は擬似ブール関数の数式処理に必要な基本操作をマクロ化して FORTRAN にうめこんだ形式をとり、BALOC-1 はやはり FORTRAN にうめこまれているが、一旦 BALOC-1 コンパイラを通して BALOC-0 に翻訳させてのち

BALOC-0 アセンブラにより翻訳されてFÖRTRAN ^プログラムとなる。BALOC-0 および BALOC-1 の詳細については、文献(4), (5) に詳しく述べてあるのでここでは省略する。

§4. ブール関数の数式処理システム.

BALOC-0, BALOC-1 は対象が特殊でありシステムとしても能率のよいものでない。BALOC-2 では、対象をブール関数に限定すると共に能率的で実用的なシステムとすることを主眼にした。擬似ブール関数で扱われる問題の殆んどがブール関数の処理だけで扱うことが出来る。したがって、このように対象を限定しても、その応用性が狭くなることはないといつてよく、むしろ能率の点でもシステム構成の点でも利点が多いと期待される。

4.1 システム構成上の問題点

ここではBALOC-2 を構成するに当って問題となった点をいくつかあげ、それらの解決法について述べる。

(a) 記号処理によるか数値処理によるか。

ブール関数は2値変数により表わされ、しかもそれらの変数は多項式で言えば、たかだか1次迄である。この特徴をうまく生かすには(すくなくとも内部的には)数値処理による方がはるかに能率的である。このことは(b)にあげるブール関数の表現法の問題にも関連する。

(b) ブール関数の表現法

ブール関数の表現法はいくつかあり、それぞれ一長一短がある。人間にとって最も分かり易い積和形は表現が一意でないという欠点がある。ここでは、表現の一意性、処理の容易さという点を考慮して内部表現として環和標準形をもちいた。

この形式によれば、1つのブール変数に対して語内の1ビットを割りあてることにより1つの(否定を含まない)積項を1語で表わし、その配列としてブール関数を表わすことが出来る。このように表現することにより、ブール関数同士の演算は、Bitwiseの機械命令を巧みにもちいることにより、極めて能率的に行なわれる。

(c) データ構造

数式処理のために数式を記憶するデータ構造としてはリスト構造がよくもちいられる。ブール関数の数式処理を(b)で述べた表現法をもちいて処理する場合には、数式の長さが動的に変わるために、データ構造も動的な形式によらなければならないことは言う迄もないが、さらに、重要なこととして、1回の関数同士の演算において何度も数式を走査する必要があるという特徴がある。このような場合には、リスト構造は必ずしも能率的ではない。ここではその解決策として、ある程度の領域をまとめて使用することにより出来る限り、数

式の内容を直接参照出来る方法，ちなみち，ページング方式をとった。また，実際の演算に当っては，アキュムレータに相当する2000語の連続した領域を確保し，演算結果を一旦その中に書き込み，演算終了後その内容をページに書きこむという方式をとった。

(d) 処理内容

ブール関数の数式処理に必要な演算は出来る限りとり入れるのが望ましい。BALOC-2では，FORTRANの算術式のような形式でブール式による数式処理を行うことが出来るが，このための演算子として， $=$ （代入）， $.EQV.$ （対等）， $+$ （環和）， $.IMP.$ （合意）， $/$ （論理和）， $*$ （論理積）， $!$ （否定）をもちいることができる。

また，専用のステートメントを用いて，あるブール変数へのブール式の代入，変数の否定，変数の置換，ある種のチェック（恒等性等）を行うことができる。

なおブール関数の入出力については，表現法の問題等から現在の所は，内部表現ほぼそのままの形式になっている。

4.2 言語構成について。

BALOC-2の言語は実用性，処理能率，翻訳能率を考慮してFORTRANプログラム中に，BALOC-2専用のステートメントを， C1 コラムにBを付した形で挿入する形式をとる。

BALOC-2専用のステートメントはブール関数名を宣言するFORMAL文, ブール変数名を宣言するBASIC文, FORTRANの算術代入文と同じ形式の数式によるブール関数の数式処理ステートメントおよび特殊な演算のための各種ステートメント, ブール関数の入出力ステートメントに分れる。

これらの詳細については文献⁽⁶⁾に述べられている。

4.3 Translatorの構成

BALOC-2システムは, 内部表現された数式をもちいて, 種々の演算を行うサブルーチンパッケージが作成されていてTranslatorはBALOC-2のソースプログラムを読みこみブール関数名を, 対応する式番号に, ブール変数を対応するビットポジションに割りつけを行い, さらに, 数式処理のためのステートメントについては, これらのサブルーチンをもちいて数式処理を行うFORTRANプログラムを作成する。

Translatorおよびサブルーチンパッケージは実験的に構成されたもので, その殆んどが, FORTRANにより記述されている。なお, 本システムの実現に当っては, 東京大学大型電子計算機センターのHITAC 5020Eを使用した。

§ 5. むすび。

ここに述べたシステムはいずれも基本的なものでまだ実験的段階にすぎない。ブール関数の数式処理システムは, 余り

とりあげられることはないが、その応用はかなり広いばかりでなく、その実現にあたって興味深い問題が多いと思われる。ここでは扱われなかったが、実際のシステムとして構成するためには、やはり、入出力の点でかなり問題が多く、またそのような点からはやはり、能率が多少落ちても、記号処理にもとづくブール関数の数式処理システムの開発が必要である。筆者は現在この見地から、BALOC-3を構成することを考え、それを実現するための準備段階として、FORTRANを拡張した形の新しい記号処理システムを開発している。(?)

謝辞 本研究集会に参加された皆様より種々有益な御討論を戴いた。ここにあらためて謝意を表する。

文献

- (1) P. L. Ivănescu, et al: "Aspra deteminarii minimelor functiilor pseudo Boolean", Studii i Cercetar Matematice, vol.14 p. 359 (1963)
- (2) 稲垣, 福村: "制約条件をもつ擬似ブール計画法について", 電子通信学会誌, 50巻6号, P.1005 (昭42)
- (3) 吉田, 稲垣, 福村: "Branch and Bound法にもとづく擬似ブール計画法のアルゴリズム", 電子通信学会誌, 50巻10号, P.1995 (昭42)
- (4) 吉田, 福村: "擬似ブール関数の数式処理用言語の

マクロアセンブラ”，昭和44年電気四学会連合大会，
講演番号3298.

- (5) 吉田，福村：“擬似ボール関数の数式処理用言語
(BALOC-1)”，昭和44年電子通信学会全国大会，
講演番号886.
- (6) 吉田，福村：“ボール関数の数式処理言語(BALOC
-2)”，昭和44年情報処理学会大会，講演番号13.
- (7) 吉田，福村：“FORTRANを拡張した記号処理言語”，
昭和45年電気四学会連合大会，講演番号2619.