論理回路の最大消費電力問題の近似可能性について

松田 健　岩間 一雄　マグナス・ハルダースソン*
京都大学
{matsuda,iwama,mmh}@kuis.kyoto-u.ac.jp

論理回路の最大消費電力問題とは、与えられた論理回路に対して、その消費電力を最大化する入力値のペアを求める問題である。一般の回路の場合には、この問題に対する近似度 $m^{1-\epsilon}$ 以下のアルゴリズムは存在しないことが簡単にわかる。そこで、我々の研究では対象とする回路を AND 一段のみの回路に制限する。この制限のもとでの最大消費電力問題を **0**、**1**、**u**、**d** の 4 値を使用する式の最大充足化問題として定式化した。まず、この問題が NP 困難であることを証明する。次に、近似アルゴリズムとして、変数の肯定・否定がバランスよく現れる場合には比較的よい近似度が得られることを示す。一般の場合に対する近似度 1.98 のアルゴリズムも与える。

The approximability of MAX Power Consumption Problem of Logic Circuits

Takeshi Matsuda　Kazuo Iwama　Magnús M. Halldórsson*
Kyoto University

MAX Power Consumption Problem of Logic Circuits is the problem of estimating the maximum power consumption of a given logic circuit. It is easy to see that this problem for general circuits is hard to approximate within a factor of $m^{1-\epsilon}$. So we consider restricted circuits that consist of only one level of AND gates. We formalize this problem as a kind of MAX SAT problem, using four values, **0**, **1**, **u** and **d**. We first prove that this problem is NP-hard. As for approximation algorithms, we consider two cases, the case that positive and negative appearances of each variable are well balanced and the general case. We can obtain a relatively good approximation factor for the balanced case and a factor of 1.98 for the general case.

1 はじめに

論理回路を設計する際にその最大消費電力を見積もることは重要な問題である。論理回路の消費電力は回路のトポロジー、クロック周波数、伝播遅延、入力値の組み合わせなどさまざまな要素が絡みあった複雑な関数によって決定される。さらに、容易に組合せ的爆発が生じる。そのため最大消費電力を求めることは計算困難な問題として広く認識されてきた。したがって、これまでの研究は、この計算困難さをいかに回避するかに重点が置かれていた。例えば [6] では主にシミュレーションによる手法が述べられている。また、[3] では確率モデルによる解析が研究されている。[2] では、最大充足化問題に変換して解くことを試みている。

本論文では、我々はこの問題を単純な組み合わせ問題とみなし、近年発展している性能保証のある近似アルゴリズム理論の枠組で論じることを考えた。我々のモデルでは、論理回路の電力消費はゲート出力の変化によってのみ起こり、また伝播遅延やゲートの誤動作等を一切考えないものとする。実際、CMOS で論理回路を構成した場合には、ゲート出力の変化の際に大きな電力を消費し、その他の電力消費 (例えば配線上での電力消費など) は比較的小さいため、この仮定は論理回路の電力消費の特

* 現在はアイスランド大学

徴を著しく失うものではない。

このシンプルなモデルのもとでは、回路の消費電力は入力値の変化だけで決定される。しかし、一般の回路の場合にはこの問題は非常に難しく、よい近似アルゴリズムが存在しないことがわかった。そのため、本論文では主にANDゲート一段のみの回路に対する最大消費電力問題(MAX 01UDSAT)について論じる。一段のみの回路というのは非現実的に見えるかもしれないが、必ずしもそうではない。なぜなら、多段回路の場合であっても、一段目のゲートが電力を消費するということはその出力が変化することを意味しており、より上の段の電力消費を誘発するための必要条件となっているからである。

MAX 01UDSATはMAX 2SATと関連がある。したがってよく知られたランダム割当て(およびその脱ランダム化)によって、全体の$\frac{1}{2}$以上の項を充足させることができる。つまり近似度2のアルゴリズムは容易に得られる。問題は2を切るアルゴリズムが得られるかどうかで、こちらは自明ではない。本稿の結果(1.98)は大きな進歩とは言えないかもしれないが、一応その第一歩を示したと言える。

本研究はいくつかの発展の可能性がある。(i)本稿でのMAX 01UDSATは各項に2個のリテラルを有するが、これはファンインを2と仮定しているからである。ゲートのファンインを3以上も許せば、当然各項におけるリテラル数も増える。(ii)より一般化された回路として次に考えられるのは木状回路(つまり各ゲートの出力ファンアウトを許さない)であろう。この場合第一段のゲート数は全体の半分を占めるので、本稿でのアルゴリズムは近似度を2倍程度に落とした近似アルゴリズムになっている。回路の構造を考慮しての改良を試みるのが次のステップである。(iii)**0**、**1**、**u**、**d**を2ビットで符号化すれば本問題は一種の制約問題である。したがって、半正定値計画による緩和が当然考えられる。我々も予備的数値実験を行ってみたがあまりよい結果は得られなかった。

2 論理回路の最大消費電力問題

2.1 問題の定義

論理回路の最大消費電力問題を以下のように定義する。

n個の入力変数と、2入力のAND、OR、およびNOTの3種類のゲートから構成される論理回路が与えられる。AND、ORゲートにはそれぞれのゲートの消費電力を表す重み(ただし本稿ではすべて1に限定する)が与えられている。また、論理回路中に故障や遅延等はない。時刻0においてある入力列$v \in \{0,1\}^n$が印加されているとき、回路中の各ゲートの出力は安定している。時刻t (>0)に入力列を$v' \in \{0,1\}^n$に変化させると、瞬時にいくつかのゲートの出力が$0 \to 1$あるいは$1 \to 0$の遷移(スイッチング動作)をする。論理回路の消費電力はスイッチング動作を行ったAND、ORゲートの重みの和で表わされる。

与えられた論理回路に対して消費電力が最大となる入力列の組合せ(v, v')を求める問題を論理回路の最大消費電力問題と呼ぶ。

2.2 近似困難性

次の定理が成立する。

定理 1 P=NPでない限りは論理回路の最大消費電力問題に対するどのような近似アルゴリズムもその近似度は$m^{1-\epsilon}$以上となる。($\epsilon > 0$、mはゲートの数。)

(証明) 3SAT(NP完全問題)からの変換を使って示す。3SATの例題(任意)として与えられた論理式$f = (x_1^1+x_2^1+x_3^1)(x_1^2+x_2^2+x_3^2)\cdots(x_1^M+x_2^M+x_3^M)$を図1の点線以下のような論理回路($x_k^j$はリテラル)で表現する。この回路は$2M$個のORゲートと$M-1$個のANDゲートで構成されている。この回路の出力部分(S点)の先に、図1上部のようにM^l $(\geq 3M)$個のANDゲートと新たな変数w_1、w_2、$\cdots$、w_{M^l}を追加する。このようにしてできた論理回路をf'とする。この変換は多項式時間で可能である。

3SATに対する答えがyesならば図1のS点を1にすることができる。そこで、3SAT部分の変数にはS点が1になるように値を割当てる。このときw_1、w_2、$\cdots$、w_{M^l}を一斉に$0 \to 1$としてやると、S点より上側のM^l個のゲートはすべて$0 \to 1$の遷移を行って電力を消費する。すなわち、f'の最大消費電力はM^l以上である。逆に3SATに対する答えがnoならばS点は常に0になる。このときS点より上側のゲートの出力はすべて0である。したがってS点より下側のゲート($3M-1$個以下)しか電力を消費しない。

lは定数の範囲でいくらでも大きくできるので図1のS点から上のゲート数$(= M^l)$は$3M-1$にくらべて多項式の範囲でいくらでも大きくできる。よって定理1が成立する。 □

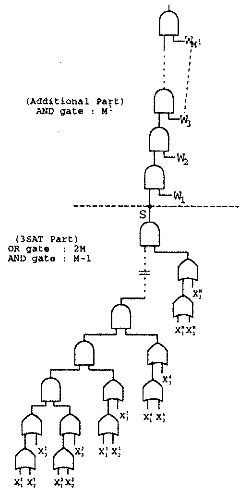

図 1: 3SAT から論理回路の最大消費電力問題へ

3 AND 一段回路の最大消費電力問題

3.1 問題の定義

一般の回路の最大消費電力問題にはよい近似アルゴリズムが存在しない。そこで、本節では対象とする回路を AND 一段回路 [図 2] に限定して考える。なぜなら、多段回路の場合でも、二段目以上のゲートが電力を消費するためには一段目のゲートが電力を消費することが必要であるため、一段目は特に重要と考えられるからである。AND 一段回路とは n 個の入力変数と m 個の 2 入力 AND ゲートからなり、入力変数はいくつかに枝分かれしてそのまま、あるいは 1 個の NOT ゲートを通過して AND ゲートへの入力となっているような回路である。

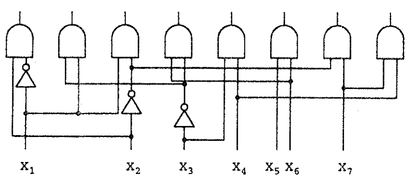

図 2: AND 一段回路

この問題を考えやすくするために、次のような最大充足化問題の形で定式化する。

MAX 01UDSAT

例題: 値として **0**、**1**、**u**、**d** のいずれかの値をとる変数 x_i $(i = 1, 2, \cdots, n)$、m 項の $(a \doteq b)$ の形の関数 (a、b はリテラル) からなる式 f。

質問: 充足される項の数を最大化する変数への割当てを求めよ。

ただし、$(a \doteq b)$ の値は表 1 のように定義される。また、項 $(a \doteq b)$ を充足するとは、$(a \doteq b)$ の値が **u** または **d** であることを言う。さらに、否定演算を $\overline{\mathbf{0}} = \mathbf{1}$、$\overline{\mathbf{1}} = \mathbf{0}$、$\overline{\mathbf{u}} = \mathbf{d}$、$\overline{\mathbf{d}} = \mathbf{u}$ と定義する。

表 1: $(a \doteq b)$ の出力

$a \backslash b$	**0**	**1**	**u**	**d**
0	**0**	**0**	**0**	**0**
1	**0**	**1**	**u**	**d**
u	**0**	**u**	**u**	**0**
d	**0**	**d**	**0**	**d**

MAX 01UDSAT が AND 一段回路の最大消費電力問題の定式化であることは以下のようにしてわかる。項 $(a \doteq b)$ を a と b を入力とする AND ゲートに対応させ、4 つの値 **0**、**1**、**u**、**d** を表 2 のように時刻 0 のときと時刻 t (> 0) のときの値のペアと考える。すると、表 1 で定義された $(a \doteq b)$ の値は時刻 0 のときと時刻 t のときの AND ゲートの出力の変化を表しており、項を充足させることはその AND ゲートがスイッチする (電力を消費する) ことに対応している。したがって、より多くの項を充足させることはより多くの AND ゲートに電力を消費させることと等価である。

表 2: 変数の置き換え

時刻 0		時刻 t (> 0)	新しい値
0	→	0	**0**
1	→	1	**1**
0	→	1	**u**
1	→	0	**d**

3.2 NP 完全性の証明

MAX 01UDSAT で、K $(\leq m)$ 項以上を充足する値の割当てが存在するかという決定問題を K-01UDSAT とすると、この問題に関して以下の定理が成立する。

定理 2 K-01UDSAT は NP 完全である。

(証明) K-01UDSAT は明らかにクラス NP に属するので、NP 完全であることが知られている MAX 2SAT の任意の例題 I が K-01UDSAT の例題 I' に多項式時間還元可能であることを示せばよい。ところで、MAX 2SAT は次のような問題である。

MAX 2SAT(2CNF 式の最大充足化問題)

例題: 変数 x_i $(i = 1, 2, \cdots, n)$ $x_i \in \{0, 1\}$、m 項か

らなる 2CNF 式 f、正の整数 K $(\leq m)$。
質問: K 項以上を充足する真理値割当てが存在するか。

MAX 2SAT の例題として論理式 $f=(x_1+y_1)(x_2+y_2)\cdots(x_m+y_m)$ と正の整数 K $(\leq m)$ が与えられたとする。ここで、x_j、y_j はリテラルを表している。これを $I=(f,K)$ と書く。f の各項 C_j $(1\leq j\leq m)$ に対して新たな変数 z_j、w_j を導入し、下のようにして 10 個の K-01UDSAT の項の積 C'_j $(1\leq j\leq m)$ を作り、これら $10m$ 項からなる式を f' とする。

$$C_j=(x_j+y_j)$$
$$\Downarrow$$
$$C'_j=$$
$$(x_j\doteq z_j)(y_j\doteq w_j)(z_j\doteq\overline{w_j})(z_j\doteq\mathbf{u})(w_j\doteq\mathbf{u})$$
$$(\overline{x_j}\doteq\overline{z_j})(\overline{y_j}\doteq\overline{w_j})(\overline{z_j}\doteq w_j)(\overline{z_j}\doteq\mathbf{d})(\overline{w_j}\doteq\mathbf{d})$$

また、$K'=6m+2K$ とすると、$I'=(f',K')$ は K-01UDSAT の例題である。この変換は多項式時間でできる。

変換が正しいことを示す。

まず、例題 I に対する答えが yes のとき例題 I' に対する答えも yes であることを示す。f を K 項充足する割当てを A_1 とする。このとき f' への割当て A'_1 を以下のように決める。f に現れる各変数 v_i に対して、A_1 のもとで $v_i=1$ のとき A'_1 では $v_i=\mathbf{u}$、A_1 のもとで $v_i=0$ のとき A'_1 では $v_i=\mathbf{d}$ とする。f' で新たに導入された変数 (各 C_j に対する z_j、w_j) への割当ては以下のように決める。

1. $x_j=\mathbf{u}$、$y_j=\mathbf{u}$ のとき
   $z_j=w_j=\mathbf{u}$
2. $x_j=\mathbf{u}$、$y_j=\mathbf{d}$ のとき
   $z_j=\mathbf{u}$、$w_j=\mathbf{d}$
3. $x_j=\mathbf{d}$、$y_j=\mathbf{u}$ のとき
   $z_j=\mathbf{d}$、$w_j=\mathbf{u}$
4. $x_j=\mathbf{d}$、$y_j=\mathbf{d}$ のとき
   $z_j=\mathbf{u}$、$w_j=\mathbf{d}$ または $z_j=\mathbf{d}$、$w_j=\mathbf{u}$

このように A'_1 を決めると、1. ～ 3. のとき (すなわち f の項 C_j が充足されているとき) には C'_j の 10 項のうち 8 項を充足でき、4. のとき (すなわち項 C_j が充足されていないとき) には C'_j のうち 6 項を充足できる。A_1 は f の K 項を充足していたので、A'_1 は f' の $8K+6(m-K)=K'$ 項を充足する。

次に、例題 I' に対する答えが yes のとき例題 I に対する答えも yes であることを示す。f' を K' 項以上充足する割当てを A'_2 とする。このとき、f への割当て A_2 を以下のように決める。f に現れる各変数 v_i に対して、A'_2 のもとで $v_i=\mathbf{u}$ のとき A_2 では $v_i=1$、A'_2 のもとで $v_i=\mathbf{d}$ のとき A_2 では $v_i=0$ とする。A'_2 のもとで $v_i=\mathbf{0}\ or\ \mathbf{1}$ のときは A_2 の側での v_i の値は任意に決めてよい。

補題 1 f' において、x_j、y_j、z_j、w_j にどのように値を割当てても、C'_j のうち最大 8 項までしか同時に充足されることはない。

補題 2 C'_j のうち 7 項以上が充足されるとき、x_j、y_j のうち少なくとも一方は $\mathbf{u}$ である。

(証明) x_j、y_j、z_j、w_j に $\mathbf{01ud}$ のすべての割当てを試してみればわかる。 □

補題 1 より、f' で K' $(=6m+2K)$ 項以上が充足されるとき、C'_j のうち 7 項以上が充足されているような j は必ず K 個以上存在する。さらに、補題 2 より、C'_j のうち 7 項以上が充足されているならば C_j も充足されている。A'_2 は f' の K' 項以上を充足していたので、A_2 は f の K 項以上を充足する。

以上より、定理 2 が示せた。 □

4 近似アルゴリズム

4.1 ud 割当てアルゴリズム

MAX 01UDSAT は当然 $\mathbf{01ud}$ の 4 値を使って充足される項数を最大化する問題であるが、$\mathbf{0}$ と $\mathbf{1}$ を使用せずに $\mathbf{u}$ と $\mathbf{d}$ のみを使ってもある程度の項は充足させることができる。したがって、$\mathbf{u}$ と $\mathbf{d}$ のみを使用する近似アルゴリズムとして以下のようなものを考えることができる。以下では $OPT_{\mathbf{01ud}}$ で MAX 01UDSAT の例題に対する最適値を、$OPT_{\mathbf{ud}}$ で変数に $\mathbf{u}$ と $\mathbf{d}$ のみを割当てる ($\mathbf{ud}$ 割当て) という制限下での最適値を表すことにする。

アルゴリズム 1 (ランダム割当て)
各変数に $\mathbf{u}$ と $\mathbf{d}$ をそれぞれ確率 $\frac{1}{2}$ でランダムに割当てたとき、項 C_j が充足される確率 s_j は

$$\begin{aligned} s_j &= Pr[(\mathbf{u}\doteq\mathbf{u})]+Pr[(\mathbf{d}\doteq\mathbf{d})] \\ &= \frac{1}{2}\cdot\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{2} \end{aligned}$$

となる。さらに充足される項数の期待値 $\widehat{W}$ は

$$\widehat{W}=\sum_{j=1}^{m}s_j=\frac{1}{2}\sum_{j=1}^{m}$$

となる。つまり $\widehat{W}$ は項の総数の $\frac{1}{2}$ である。このとき、$\widehat{W}$ 以上の値を実現する脱ランダム化算法がある。この算法をアルゴリズム 1 とし、アルゴリズム 1 によって得られる解を $RND_{\mathbf{ud}}$ と書く。

アルゴリズム 2 (半正定値計画緩和)
MAX SAT などの組合せ最適化問題に対して半正定値計画緩和を利用したよい近似アルゴリズム [4, 5, 7] が知られている。MAX 01UDSAT の例題に **ud** 割当てを行う場合にも、文献 [4] の方法を応用することによって、$OPT_{\mathbf{ud}}$ の 0.878 倍以上の解を保証する近似アルゴリズムを得ることができる。このアルゴリズムをアルゴリズム 2 とし、アルゴリズム 2 によって得られる解を $SDP_{\mathbf{ud}}$ と書く。

アルゴリズム 1 は絶対数として半分 (以上) の項を充足させることができるので、MAX 01UDSAT に対する近似度 2 のアルゴリズムになっている。アルゴリズム 2 は $OPT_{\mathbf{ud}}$ の 0.878 倍以上の項を充足させることはできるが、$OPT_{\mathbf{01ud}}$ に対してはわからないので、近似度をただちに言うことはできない。($OPT_{\mathbf{ud}} \geq \frac{1}{2}OPT_{\mathbf{01ud}}$ であるから、全体の $\frac{1}{2} \times 0.878 = 0.439$ の項が充足されることを保証しているにすぎない。) また、**u**、**d** のみでなく、**0**、**1**、**u**、**d** をランダムに割当てるようなアルゴリズムも考えられるが、その場合に充足できる項数の期待値は $\frac{6}{16} = 0.375$ で、上のアルゴリズム 1 より劣る。したがって、我々の目標は MAX 01UDSAT に対する近似度 2 を切るアルゴリズムである。

4.2 変数の肯定リテラルと否定リテラルがバランスよく現れる場合

例題式中のすべての入力変数に対して、肯定リテラルと否定リテラルがバランスよく現れるという制限を加えた場合には、**ud** 割当てのアルゴリズム 2 がそのままそのまま **01ud** 割当ての最適解に対しても有効な近似アルゴリズムとなっている。本節ではこのことを示す。

例題中に現れる変数 x_i の肯定リテラル x_i の数を k_i、否定リテラル $\overline{x_i}$ の数を l_i とする。ここで、$k_i \geq l_i$ としてよい。なぜなら、もし $k_i < l_i$ ならば変数 x_i のすべてのリテラルに否定演算を 1 つずつ付加することで $k_i > l_i$ とでき、このとき充足できる項数に影響を与えないからである。

次の定理が成立する。

定理 3 すべての i について $k_i = l_i$ であるような例題に対しては、$OPT_{\mathbf{ud}} = OPT_{\mathbf{01ud}}$ の関係が成り立つ。

(証明) 例題として与えられた式に任意に **01ud** 割当てを行ったとき、各項は下の図のいずれかで表される。

$$\underbrace{(\mathbf{0} \doteq \mathbf{0})}_{a\text{ 項}}\ \underbrace{(\mathbf{0} \doteq \mathbf{u})\ (\mathbf{0} \doteq \mathbf{d})}_{b\text{ 項}}\ \underbrace{(\mathbf{0} \doteq \mathbf{1})}_{c\text{ 項}}$$

$$\underbrace{(\mathbf{1} \doteq \mathbf{u})\ (\mathbf{1} \doteq \mathbf{d})}_{d\text{ 項}}\ \underbrace{(\mathbf{1} \doteq \mathbf{1})}_{e\text{ 項}}\ \underbrace{(\mathbf{u} \doteq \mathbf{d})}_{f\text{ 項}}\ \underbrace{(\mathbf{u} \doteq \mathbf{u})\ (\mathbf{d} \doteq \mathbf{d})}_{g\text{ 項}}$$

ここで、$a \sim g$ はそれぞれの項の数を示す。また、この定理の条件を満たす式の場合、値が **1** となっているリテラルと **0** となっているリテラルは同数ずつ存在する。したがって、$a \sim g$ は次の式を満たす。

$$a, b, c, d, e, f, g \geq 0 \tag{1}$$

$$a + b + c + d + e + f + g = m \tag{2}$$

$$2a + b + c = 2e + d + c \tag{3}$$

充足されている項数の和を $W_{\mathbf{01ud}}$ と書くと、

$$W_{\mathbf{01ud}} = d + g \tag{4}$$

である。

同じ式に対して **ud** 割当てを行った場合を考える。このとき、充足できる項数を $W_{\mathbf{ud}}$ と書くと

$$\begin{aligned} W_{\mathbf{ud}} &\geq a + e + g + \frac{1}{2}(b + d) && (5)\\ &= \frac{1}{2}(2a + b) + \frac{1}{2}(2e + d) + g && (6)\\ &= 2e + d + g && (7)\\ &\geq W_{\mathbf{01ud}} && (8) \end{aligned}$$

が言える。以下、これを証明する。

(5) 式: **01ud** 割当てで **u** または **d** を割り当てた変数には同じ **u**、**d** を割当てることができる。これらの変数をグループ 1 の変数と呼ぶことにする。**01ud** 割当てで **1** を割当てた変数と **0** を割当てた変数 (グループ 2 の変数) にそれぞれ **u**、**d** を割当ててみると、項の分布は次のようになっている。

$$\underbrace{(\mathbf{d} \doteq \mathbf{d})}_{a\text{ 項}}\ \underbrace{(\mathbf{d} \doteq \mathbf{u})\ (\mathbf{d} \doteq \mathbf{d})}_{b\text{ 項}}\ \underbrace{(\mathbf{d} \doteq \mathbf{u})}_{c\text{ 項}}$$

$$\underbrace{(\mathbf{u} \doteq \mathbf{u})\ (\mathbf{u} \doteq \mathbf{d})}_{d\text{ 項}}\ \underbrace{(\mathbf{u} \doteq \mathbf{u})}_{e\text{ 項}}\ \underbrace{(\mathbf{u} \doteq \mathbf{d})}_{f\text{ 項}}\ \underbrace{(\mathbf{u} \doteq \mathbf{u})\ (\mathbf{d} \doteq \mathbf{d})}_{g\text{ 項}}$$

このとき、明らかに $(a + e + g)$ 項は充足されている。加えて b 項のグループと d 項のグループの $\frac{1}{2}$ 以上の項が充足されている。もし充足されていなければグループ 2 の変数への割当てを **u** と **d** をすべて入れ代えたものにすることで $\frac{1}{2}(b + d)$ 項以上を充足させることができる。このことは以下のようにしてわかる。b 項のグループは b_1 項の $(\mathbf{d} \doteq \mathbf{u})$ と b_2 項の $(\mathbf{d} \doteq \mathbf{d})$ に分けることができる。また、d 項のグループは d_1 項の $(\mathbf{u} \doteq \mathbf{u})$ と d_2 項の $(\mathbf{u} \doteq \mathbf{d})$ に分けられる。もし、$b_1 + d_2 > b_2 + d_1$ であれば、グループ 2 の変数への割当てを **u** と **d** を入れ代えたものにすると、b_1 項が $(\mathbf{d} \doteq \mathbf{d})$、$b_2$ 項が (**d**

$\doteq$ **u**)、d_1 項が ($\mathbf{u} \doteq \mathbf{d}$)、$d_2$ 項が ($\mathbf{u} \doteq \mathbf{u}$) となり、$\frac{1}{2}(b+d)$ 項以上が充足されることになる。しかも、a 項、e 項、g 項のグループがすべて充足されることに変わりはない。したがって、不等式 (5) が言える。

(6) から (7): (3) 式より。

(7) から (8): (1)、(4) より。

以上より、このような例題に対しては、任意の $W_{\mathbf{01ud}}$ に対して $W_{\mathbf{ud}} \geq W_{\mathbf{01ud}}$ となる **ud** 割当てが存在する。明らかに $OPT_{\mathbf{ud}} \leq OPT_{\mathbf{01ud}}$ なので、$OPT_{\mathbf{ud}} = OPT_{\mathbf{01ud}}$ である。 □

定理 3 より

$$\begin{aligned} \frac{SDP_{\mathbf{ud}}}{OPT_{\mathbf{01ud}}} &= \frac{SDP_{\mathbf{ud}}}{OPT_{\mathbf{ud}}} \cdot \frac{OPT_{\mathbf{ud}}}{OPT_{\mathbf{01ud}}} \\ &\geq 0.878 \end{aligned}$$

が言える。これはすなわち、定理 3 のような例題に限っていえば $SDP_{\mathbf{ud}}$ が $OPT_{\mathbf{01ud}}$ の 0.878 倍以上を保証する近似解になっているということである。

バランスが多少悪くなってもこの **ud** 割当ては有効である。

定理 4 $\max_i \frac{k_i}{l_i} = \frac{k}{l}$ であるような例題に対しては次の関係が成り立つ。

$$\frac{OPT_{\mathbf{01ud}}}{OPT_{\mathbf{ud}}} < \frac{4k}{2k+l}$$

(証明) 最適な **01ud** 割当てで **0** または **1** が現れる項数の全体の項数 m に対する割合を r とする。また、($\mathbf{u} \doteq \mathbf{u}$) あるいは ($\mathbf{d} \doteq \mathbf{d}$) となっている項数の全体に対する割合を s とする。つまり、下のような分布になっている。(ただし、#は任意の値である。)

$$\underbrace{(\mathbf{0} \doteq \#)\,(\mathbf{1} \doteq \#)}_{mr\text{ 項}} \quad \underbrace{(\mathbf{u} \doteq \mathbf{u})\,(\mathbf{d} \doteq \mathbf{d})}_{ms\text{ 項}} \quad \underbrace{(\mathbf{u} \doteq \mathbf{d})}_{m(1-r-s)\text{ 項}}$$

0 または **1** が現れている mr 項のうち mx 項が充足されるとすると、残りの $m(r-x)$ 項は充足されていない。さて、**0** または **1** を含む項が充足されるのは ($\mathbf{1} \doteq \mathbf{u}$) または ($\mathbf{1} \doteq \mathbf{d}$) の形のときだけである。したがって、このような項が mx 項存在する。今、すべての i について $\frac{k_i}{l_i} \leq \frac{k}{l}$ であることより、**0** は少なくとも $mx \times \frac{l}{k} = m\frac{l}{k}x$ 個以上現れる。したがって、**0** を含む項 (絶対に充足できない項) は、すべて ($\mathbf{0} \doteq \mathbf{0}$) の形で現れるとしても $m\frac{l}{k}x \times \frac{1}{2} = m\frac{l}{2k}x$ 個以上存在する。よって、

$$r - x \geq \frac{l}{2k}x$$

あるいは

$$x \leq \frac{2k}{2k+l}r$$

が言える。このことから

$$\begin{aligned} OPT_{\mathbf{01ud}} &= m(s+x) \\ &\leq m(s + \frac{2k}{2k+l}r) \end{aligned} \tag{9}$$

であることがわかる。

一方、$OPT_{\mathbf{ud}}$ の下限を考える。**01ud** 割当てで **u** と **d** が割当てられていた変数にはそのまま同じ **ud** を割当てる。すると、ms 項は充足され、$m(1-r-s)$ 項は充足されない。さらに、残りの変数にはアルゴリズム 1 を用いる。その結果、残り mr 項のうち少なくとも半分以上の項を充足させることができる。したがって、$OPT_{\mathbf{ud}}$ に関して

$$OPT_{\mathbf{ud}} \geq m(s + \frac{1}{2}r) \tag{10}$$

が言える。(9)、(10) より

$$\begin{aligned} \frac{OPT_{\mathbf{01ud}}}{OPT_{\mathbf{ud}}} &\leq \frac{m(s+\frac{2k}{2k+l}r)}{m(s+\frac{1}{2}r)} \\ &= \frac{4k}{2k+l} - \frac{\frac{2k-l}{2k+l}s}{s+\frac{1}{2}r} \\ &< \frac{4k}{2k+l} \end{aligned}$$

となる。 □

定理 4 より

$$\begin{aligned} \frac{SDP_{\mathbf{ud}}}{OPT_{\mathbf{01ud}}} &= \frac{SDP_{\mathbf{ud}}}{OPT_{\mathbf{ud}}} \cdot \frac{OPT_{\mathbf{ud}}}{OPT_{\mathbf{01ud}}} \\ &\geq 0.878 \times \frac{2k+l}{4k} \end{aligned}$$

である。すなわち、このような例題に対しては $SDP_{\mathbf{ud}}$ が $OPT_{\mathbf{01ud}}$ の $0.878 \times \frac{2k+l}{4k}$ 以上を保証する解となっている。例えば $\frac{k}{l} = 2$ のときは、近似度約 1.82 が保証される。なお、$SDP_{\mathbf{ud}}$ が **01ud** 割当ての近似解として有効なのは $\frac{k}{l} < 3.598$ のときに限る。なぜなら、$\frac{k}{l}$ がこれより大きくなると $\frac{OPT_{\mathbf{01ud}}}{SDP_{\mathbf{ud}}}$ は 2 以上となってしまうからである。

4.3 一般の場合

前節で示したように変数の肯定、否定がバランスよく現れる例題に対しては、**ud** 割当てのアルゴリズムは有効な近似アルゴリズムとなっている。しかし、肯定と否定がアンバランスに現れる場合には $OPT_{\mathbf{01ud}} = 2OPT_{\mathbf{ud}}$ となる例が存在する。(例えば簡単な例として $(x_1 \doteq x_2)(x_1 \doteq \overline{x_2})$ がある。) したがって、**ud** 割当てのみで近似アルゴリズムを構築する場合には、近似度が 2 よりよいアルゴリズムを得ることは不可能である。近似度が 2 より小さい近似アルゴリズムを構築するためには **01ud** 割当てを行う必要がある。

01 を割当てる利点は、それによって $(\mathbf{1} \doteq \mathbf{u})$ あるいは $(\mathbf{1} \doteq \mathbf{d})$ の項を作れることである。ただ、それに伴って、$(\mathbf{0} \doteq \text{-})$ や $(\mathbf{1} \doteq \mathbf{1})$ の項 (これらの項は充足されない) も当然できてしまう。前者の項の数と後者の項の数の比を $1+\alpha$ (α は正の数) にできれば、さらにそのような項の数を全体の定数分の 1 程度にまで大きくすることができれば 2 より小さい近似度を実現することができる (つまりそれ以外の変数にはアルゴリズム 1 によって **u**、**d** を割当てればよい)。

変数の肯定、否定がバランスよく現れる場合には前に述べたアルゴリズムが有効であるので、バランスの悪い場合のみを考えればよい。また、常に肯定の方が否定より多く現れると仮定してよい。つまり、原則はアンバランスな変数に **1** を代入することによって目的を達成したい。直観的にはある変数を **1** にしたら、その変数と「重ならない」変数を次に **1** にすればよい。「重ならない」とはその 2 つの変数が (否定 ≐ 否定) 以外の形で項として現れないことを意味する。((否定 ≐ 否定) のみで現れる場合は、新しい変数の肯定のリテラルがすべて $(\mathbf{1} \doteq \mathbf{u})$ または $(\mathbf{1} \doteq \mathbf{d})$ の項を作るのに貢献するが、それ以外は一部のそのような項を壊すことになる。)

さて、アンバランスな変数の中からこのような「重ならない」変数を選ぶ問題は独立頂点集合を選ぶ問題と完全に一致する。しかし、これは近似が困難であることがよく知られている問題である。もちろん、対象となるグラフが一般のグラフよりは制限されたグラフになる可能性はあるが、十分によい近似アルゴリズムが見つかるかどうかは明らかではない。より困難な問題は、このような独立頂点集合問題では元々良質な解が存在しないかもしれないことである。つまり、重なりはあってもその重なりの度合が小さければ十分な数の $(\mathbf{1} \doteq \mathbf{u})$ または $(\mathbf{1} \doteq \mathbf{d})$ の項を作れる可能性はあるわけで、我々が本当に求めたいのはそのような割当てである。多少は枝を含んでもよい粗な部分グラフを求めるアルゴリズムも提案されてはいるが [1]、どの程度粗であればよいかなど決定しなくてはいけない問題は数多く、簡単に利用できるとは考えにくい。

他に、アンバランスな変数の一部を選ぶのではなくて、全部に対して都合のよいように **01** と **ud** の割当てを求める方法も考えられるが、易しくはなさそうである。このように 2 を切る近似アルゴリズムの構築は自明ではない。

以下のアルゴリズムでは我々は「極端に」アンバランス度の高い変数に着目する。このような変数をすべて見たとき、上で述べた重なりが小さければ原則通り **1** を割当てればよい。重なりが多い場合には **u** を割当てればよい。アンバランス度の高さが $(\mathbf{u} \doteq \mathbf{u})$ 項が数多く現れることを保証するのである。

定義 1 UB(UnBalanced) 変数とは、肯定リテラルと否定リテラルの出現回数の比 ($\frac{k_i}{l_i}$) が 17 以上であるような変数のことを言う。

アルゴリズム 3
UB 変数を見つける。
(1) UB 変数が全体の 0.43 以上の項に現れており、かつ

(1-1) 2 つのリテラルが両方 UB 変数であるような項がそのうちの 0.36 倍以上を占める場合。この場合 UB 変数に **u** を代入し、残りの変数にはアルゴリズム 1 を適用して割当てを行う。

(1-2) そうでない場合。UB 変数に **1** を代入し、残りの変数にはアルゴリズム 1 を適用する。
(2) (1) でない場合。アルゴリズム 1 とアルゴリズム 2 を適用し、よい方を解とする。

アルゴリズム 3 に関して次の定理が成立する。

定理 5 アルゴリズム 3 の近似度は 1.98 である。

(証明) 最適な割当てを次のような形で表すことができる。

$$\underbrace{(\mathbf{0} \doteq \#)\cdots(\mathbf{0} \doteq \#)}_{ma\text{ 項}}\ \underbrace{(\mathbf{1} \doteq \%)\cdots(\mathbf{1} \doteq \%)}_{mb\text{ 項}}$$
$$\underbrace{(\mathbf{u} \doteq \mathbf{u})\cdots(\mathbf{u} \doteq \mathbf{u})\ (\mathbf{d} \doteq \mathbf{d})\cdots(\mathbf{d} \doteq \mathbf{d})}_{mc\text{ 項}}$$
$$\underbrace{(\mathbf{u} \doteq \mathbf{d})\cdots(\mathbf{u} \doteq \mathbf{d})}_{md\text{ 項}}$$

ただし、#は **0**、**1**、**u**、**d** のいずれかを、%は **1**、**u**、**d** のいずれかを表している。また、a、b、c、d は全体の項数 m に対する割合を示す。以下の 4 つの場合が考えられる。

(1) $a \geq 0.01$ の場合

$OPT_{\mathbf{01ud}} \leq 0.99m$ より、アルゴリズム 1 を適用すれば $\frac{RND_{\mathbf{ud}}}{OPT_{\mathbf{01ud}}} \geq \frac{0.5}{0.99} \geq 0.505$。

(2) $d \geq 0.01$ の場合

(1) と同様に $\frac{RND_{\mathbf{ud}}}{OPT_{\mathbf{01ud}}} \geq 0.505$。

(3) $c+d \geq 0.2$ の場合

この場合、その割当ての **ud** 割当ての部分はそのまま使用し、**01** 割当ての変数をランダムに **ud** 割当てに変更する。このことによって $0.8m \times 0.5 + (0.2m - 0.01m) \geq 0.59m$ の項が充足される。したがって、$OPT_{\mathbf{ud}} \geq 0.59m$ である。アルゴリズム 2 を使うと $\frac{SDP_{\mathbf{ud}}}{OPT_{\mathbf{01ud}}} = \frac{SDP_{\mathbf{ud}}}{OPT_{\mathbf{ud}}} \cdot \frac{OPT_{\mathbf{ud}}}{OPT_{\mathbf{01ud}}} \geq 0.878 \times 0.59 =$

0.518。
(4) (1) でも (2) でも (3) でもない場合 (つまり $a < 0.01$ かつ $b > 0.79$ のとき)
次の補題が成立する。

補題 3 このとき、UB 変数が現れる項数の全体の項数に対する割合を y とすると、$y \geq 0.43$ である。

(証明) まず、最適な割当てを行ったとき、UB 変数でない変数のリテラルで値が **1** となっているものが $0.36m$ 個より少ないことを示す。仮に $0.36m$ 以上であると仮定しよう。すると、これらの変数の否定 (すなわち肯定と否定を入れ換えたリテラル) は値が **0** となっているはずであり、その個数は少なくとも $0.36m \times \frac{1}{17} = 0.021m$ である。しかし、$a < 0.01$ より、値が **0** であるリテラルは $0.02m$ 個以下 (つまり 1 つの項が共に **0** のことがあるので) しか存在し得ない。よって矛盾。したがって、値が **1** となっている UB 変数以外のリテラルが現れる項の数は $0.36m$ より小さい。一方、$b > 0.79$ より値が **1** であるようなリテラルは $0.79m$ 項以上に現れている。つまり $0.79m - 0.36m = 0.43m$ 項以上に UB 変数が出現する。 □

以下では 2 つの場合を考える。
(場合 1) 2 つのリテラルの両方が UB 変数であるような項が全体の $0.36y$ 以上を占める場合
2 つのリテラルのうち少なくとも一方が UB 変数の否定リテラルであるような項は高々

$$2y \times \frac{1}{18} < 0.112y$$

しか存在しない。(理由：UB 変数は全体の my 項にしか現れない。2 つのリテラルが共に UB 変数としてもリテラルの総数は $2my$ 以下である。そのうち最大 $\frac{1}{18}$ が否定リテラルである。) 逆に 2 つのリテラルが両方とも UB 変数の肯定であるような項は

$$0.36y - 0.112y = 0.248y$$

以上存在する。このとき、UB 変数にすべて **u** を代入し、残りの変数にはアルゴリズム 1 を適用すると、全体の

$$0.248y + (1 - 0.36y) \times 0.5 = 0.5 + 0.068y \geq 0.529$$

以上の項が充足される。
(場合 2) 2 つのリテラルが両方 UB 変数であるような項が $0.36y$ 以下の場合
UB 変数すべてに **1** を割当て、残りの変数にはアルゴリズム 1 を適用して値を割当てる。すると、$(\mathbf{1} \doteq \mathbf{1})$ となる可能性がある項は $0.36y$ 以下、$(\mathbf{0} \doteq \#)$ となる可能性がある項は $(y + 0.36y) \times \frac{1}{18}$ 以下である。これは次のような理由による。(理由：2 つのリテラルが共に UB 変数である項は $0.36my$ 項以下である。これらすべてが $(\mathbf{1} \doteq \mathbf{1})$ になったとしても当然 $0.36my$ 項以下になる。また、UB 変数のリテラルは多くても $m(y + 0.36y)$ である。**0** となるリテラルは UB 変数のリテラル全体の $\frac{1}{18}$ 以下であるため、そのすべてが $(\mathbf{0} \doteq \#)$ であっても $(y + 0.36y) \times \frac{1}{18}$ にしかならない。) よって、全体の

$$y - 0.36y - (y + 0.36y) \times \frac{1}{18} + (1 - y) \times 0.5 \geq 0.5 + 0.064y \geq 0.527$$

以上の項が充足される。

以上のようにして (1) ～ (4) いずれの場合にも最適解の 0.505 倍以上の項を充足することができる。アルゴリズム 3 は、まず UB 変数に注目することによって、(4) の場合を検出することができ、その場合には上の解析の通りにふるまう。また、(4) 以外の場合 ((1) ～ (3) の場合) にはアルゴリズム 1 とアルゴリズム 2 の両方を実行し、そのよい方をとることで全体の項数の 0.505 倍以上を充足することも上で示した通りである。よってアルゴリズム 3 の近似度は $\frac{1}{0.505} = 1.98$ である。 □

参考文献

[1] Asahiro, Y., Iwama, K., Tamaki, H. and Tokuyama, T. : Greedily Finding a Dense Subgraph. *J. Algorithms*, 34, pp. 203–221 (2000).

[2] Devadas, S., Keutzer, K. and White, J. : Estimation of Power Dissipation in CMOS Combinational Circuits Using Boolean Function Manipulation. *IEEE Trans. Computer-Aided Design*, vol.11, pp.373–383 (1992).

[3] Ding, C., Tsui, C. and Pedram, M. : Gate-Level Power Estimation Using Tagged Probabilistic Simulation. *IEEE Trans. Computer-Aided Design*, vol.17, pp.1099–1107 (1998).

[4] Goemans, M. X. and Williamson D. P. : .878-Approximation Algorithms for MAX CUT and MAX 2SAT. In *Proceedings of the 26th Symposium on Theory of Computing, Montreal, Canada*, pp. 422–431 (1994).

[5] Karloff, H. and Zwick, U. : A 7/8-approximation algorithm for MAX 3SAT? In *Proceedings of the 38rd Annual IEEE Symposium on Foundations of Computer Science, Miami Beach, Florida*, pp. 406–415 (1997).

[6] Monteiro, J., Devadas, S., Ghosh, A., Keutzer, K. and White, J. : Estimation of Average Switching Activity in Combinational Logic Circuit Using Symbolic Simulation. *IEEE Trans. Computer-Aided Design*, vol.16, pp.121–127 (1997).

[7] Zwick, U. : Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the 9th annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California*, pp. 201–210 (1998).