A-005

DNA 計算における MAX-SAT の解法
An Algorithm for MAX-SAT with DNA strands

徳丸 雄一郎† 藤原 暁宏†
Yuichiro Tokumaru Akihiro Fujiwara

1. はじめに

近年新しい計算パラダイムの一つである DNA 計算が注目を集めている．DNA 計算では，DNA の持つ超並列性を利用して計算困難な問題に対して多項式時間で解を求めるアルゴリズムが提案されている．本研究では，この DNA 計算において NP 困難問題の一つである MAX-SAT を DNA 計算によって解くアルゴリズムを提案する．このアルゴリズムでは，論理関数の節の数が h 個，かつ変数の数が n 個の MAX-SAT に対して，各入力は一本鎖 DNA による m ビットの 2 進数で表される．この入力に対して，提案アルゴリズムは，あらかじめ必要な DNA を $O(n+m)$ ステップで準備することにより，$O(hm2^n)$ 種類の DNA を用いて，$O(1)$ ステップで解くことができる．

2. 準備

2.1 MAX-SAT

MAX-SAT とは，与えられた論理関数の充足する節の数が最大となる変数割当を求める問題であり次のように定義される．

入力 n 個の 0-1 変数と h 個の節を持つ和積標準形の論理関数 F

出力 論理関数 F の充足する節の数が最大となるような n 変数への入力割当と，そのとき充足する節の数

例として，節の数が 4 個 ($h=4$)，3 変数 ($n=3$) の MAX-SAT を考える．入力の論理関数 F を式 (1) とする．

$$F(x_0,x_1,x_2)=(x_0+x_1)(x_0+\overline{x_1})(x_1+x_2)(x_2) \quad (1)$$

ここで，各節を左から y_0，y_1，y_2，y_3 とおく．このとき，充足される節の個数 z は以下の式で表される．

$$z=y_0+y_1+y_2+y_3 \quad (2)$$

この例では，$(x_2,x_1,x_0)=(1,0,1),(1,1,1)$ のとき，$z=4$ となり，満たされる節の数が最大となる．

2.2 DNA 計算モデル

本研究では，Reif[3] によって提唱された DNA 計算モデルである RDNA モデルを使用する．この RDNA モデルは，DNA の表現および DNA の生化学的操作を抽象化し，DNA 計算のアルゴリズムの簡潔な記述を可能にするものである．

RDNA モデルでは，*Merge*，*Copy*，*Detect*，*Separation*，*Selecton*，*Cleavage*，*Annealing*，*Denaturation* という 8 つの基本操作が義されており，本研究では更に *Empty* という追加の基本操作を用いる．これらの DNA に対する操作は生化学的操作の組み合わせにより実行できるので，その計算量を $O(1)$ ステップと定義する．また，上記の基本操作に加えて，*CreateStrand_L*，*CreateStrand_R*，*Add_V_L*，*Add_V_R*，*CreateSplint*，*CreateMemory*，*CreateInput*，*ChangeMemory* という操作 [4] を用いることにより，アルゴリズムにあらかじめ必要な DNA を $O(n+m)$ ステップで生成する．

文献 [1] において，$O(mn)$ 種類の一本鎖 DNA を用いることにより n 個の m ビットの 2 進数を表現する方法が定義されている．この方法は，n 個の 2 進数を 1 ビット毎に一本鎖 DNA で表現するというアイデアに基づいており，この 1 ビットを表す一本鎖 DNA をメモリ鎖と呼ぶ．本研究ではこのメモリ鎖を用いて，DNA により各変数割当，論理関数の各節の式の係数，節の式の論理値，充足する節の数を 2 進数で表現する．

2.3 DNA による既知の演算

n 個の m ビットの 2 進数を $O(mn)$ 種類の一本鎖 DNA で表現した場合に，文献 [1] において任意の個数のメモリ鎖に値を割り当てるアルゴリズム，及び，論理演算，減算を実行するアルゴリズムについて，また，文献 [2] において任意の個数のメモリ鎖の中から最大値を求めるアルゴリズムについて，以下のような補題が証明されている．

補題 1[1] 任意の個数のメモリ鎖への値の割り当ては，$O(1)$ 種類の DNA を用いることにより $O(1)$ ステップで実行できる． □

補題 2[1] n 個の m ビットの 2 進数における $O(n)$ 個の対に対する任意の論理演算は，$O(mn)$ 種類の DNA を用いることにより $O(1)$ ステップで実行できる． □

補題 3[1] n 個の m ビットの 2 進数における $O(n)$ 個の対に対する減算は，$O(mn)$ 種類の DNA を用いることにより $O(1)$ ステップで実行できる． □

補題 4[2] n 個の m ビットの 2 進数の集合の中の最大値の計算は $O(mn^2)$ 種類の DNA を用いることにより $O(1)$ ステップで実行できる． □

3. アルゴリズムの概要

アルゴリズムは以下の 3 つのステップからなる．ただし，以下の表記において，$0\le p\le h-1$，$0\le i\le 2^n-1$，$0\le j\le h$，$0\le k\le m-1$ とし，$Y_{i,p}(V)$ は値が $V\in\{0,1\}$ であるメモリ鎖 $Y_{i,p}$ を表している．

Step 1 各節 p において，すべての変数割当 P_i と節の式の係数 Q_p の論理積演算を行い，すべての変数割当における節の式の論理値 $Y_{i,p}$ を並列に求める．例

†九州工業大学情報工学部電子情報工学科

として，変数割当が $(x_1, x_0) = (1, 1)$ であり，節の式が $x_0 + \overline{x_1}$ である場合を考える．ただし，次式中の 2 進数は最上位ビットから順に $\overline{x_1}$，$\overline{x_0}$，x_1，x_0 に対応している．

$$P_i \cap Q_p = 0011 \cap 1001 = 0001$$

ここで，論理積の結果が値 1 のビットを 1 つでも持てば，i 番目の変数割当は節 p を充足させる．従って，この例では補題 1 に示されるアルゴリズムにより $Y_{i,p}$ に 1 を割り当てる．

Step 2 すべての変数割当 P_i において並列に，各節の式の論理値 $Y_{i,p}$ の中で値 1 を持っているメモリ鎖のみを連結し，その長さによって充足する節の数 $Z_{i,j,k}$ を求める．例として i 番目の変数割当の結果を表す以下の集合を考える．

$$\{Y_{i,0}(1), Y_{i,1}(1), Y_{i,2}(1)\}$$

この集合に対し，あらかじめ，ビット昇順に連結させるために 0 と 1，0 と 2，1 と 2，というペアの添え木を用意しておくことで，次のような一本鎖 DNA が得られる．

$$\{Y_{i,0}(1)Y_{i,2}(1),\ Y_{i,0}(1)Y_{i,1}(1)Y_{i,2}(1)\}$$

この中で一番長い一本鎖 DNA の長さが値 1 を持つメモリ鎖の数を示している．したがって，$h = 3$ の場合に，一番長い一本鎖 DNA の長さを得るために，次のような一本鎖 DNA の集合を用いる．

$$\{\boxed{\alpha}\boxed{\alpha}\boxed{\alpha}Z_{i,3,0}(0)Z_{i,3,1}(0)$$
$$\boxed{\alpha}\boxed{\alpha}Z_{i,2,0}(1)Z_{i,2,1}(0)$$
$$\boxed{\alpha}Z_{i,1,0}(0)Z_{i,1,1}(1)$$
$$Z_{i,1,0}(1)Z_{i,1,1}(1)\}$$

これらの一本鎖 DNA は 2 つの部分からできており，左側の部分は，メモリ鎖の長さと等しい記号列 α を用いて，メモリ鎖の長さの定数倍の長さを構成している．右側の部分は，長さを表すメモリ鎖からできている．ここで，j を α の個数とすると，右側の部分のメモリ鎖には $h - j$ という値が 2 進数として格納されている．
これら 2 種類の一本鎖 DNA を連結し，$|\alpha| \times h + \beta$($\beta$ は追加した一本鎖の右側の部分の長さ) の長さの一本鎖 DNA を抜き出す．この例では，次のような一本鎖 DNA が抜き出される．$(|\alpha| \times 3 + 2|\alpha| = 5|\alpha|)$

$$\{\boxed{\alpha}\boxed{\alpha}\boxed{\alpha}Z_{i,4,0}(0)Z_{i,4,1}(0)$$
$$Y_{i,0}(1)\ \ Y_{i,2}(1)\boxed{\alpha}Z_{i,2,0}(0)Z_{i,2,1}(1)$$
$$Y_{i,0}(1)\ \ Y_{i,1}(1)\ \ Y_{i,2}(1)Z_{i,1,0}(1)Z_{i,1,1}(1)\}$$

これらの一本鎖 DNA から，長さが格納してある 2 つ目の部分のメモリ鎖を抽出する．この中で最も大きな値が i 番目の変数割当が充足させる節の数である．しかし，ここでそれぞれの変数割当について充足させる節の数は求める必要はなく，後で全体から，充足させる節の数の最大値を求めれば良い．

Step 3 充足する節の数が最大となるような変数割当と，そのとき充足する節の数 M を出力する．Step 2 より，各割当が充足させる節の数が求まっているので，補題 4 により示されるアルゴリズムを用いてそれらの中から最大値を求める．もし，

$$1,\ 0,\ 2,\ 3,\ 3,\ 2,\ 1,\ 0,\ 3$$

のように解が複数個あった場合は，まず，補題 4 により示されるアルゴリズムより $M = 3$ を求め，次に補題 3 により示されるアルゴリズムを用いてすべての解の候補から M を減算する．上記の例では，減算結果は以下のようになる．

$$-2, -3, -1,\ 0,\ 0, -1, -2, -3,\ 0$$

この中で値が 0，即ち符号ビットが 0 のものが解となるので，対応する変数割当を最大値 M とともに出力する．

本研究では，上記のアルゴリズムについて，下記の定理が得られている．

定理 1 論理関数の節の数が h 個，かつ，変数の数が n 個という MAX-SAT は，各入力を m ビットの 2 進数で表し，あらかじめ必要な DNA を $O(n+m)$ ステップで準備することにより，$O(hm2^n)$ 種類の DNA を用いて，$O(1)$ ステップで解を求めることができる．

4. まとめ

本研究では DNA 計算により MAX-SAT の解を求めるアルゴリズムを提案した．今回提案したアルゴリズムはあくまでも理論に基づくものであり，実際の DNA を使用した実験は行っていない．したがって，提案したアルゴリズムを実現するために，変数の個数やビット数に比例する長さの DNA 鎖を使用しないアルゴリズムへの改良や，基本操作におけるエラー率を考慮した計算精度の高いアルゴリズムの提案を行うことなどが今後の研究課題である．

参考文献

[1] A. Fujiwara, K. Matsumoto, W. Chen. Procedures for Logic and Arithmetic operations with DNA Molecules. *International Journal of Foundations of Computer Science*, Vol. 15, No. 3, pp. 461-474, 2004.

[2] S. Kamio, A. Takahara and A. Fujiwara. Procedures for computing the maximum with DNA strands. *Proceedings of the International Conference on Parallel and Distributed Processing Symposium, 2003.*

[3] J.H.Reif. Parallel biomolecular computation: models and simulations. *Algorithmica*, Vol. 25, No. 2-3, pp. 142–175, 1995.

[4] 福元 圭祐. メモリ鎖に関連する DNA 計算アルゴリズムに関する研究. 九州工業大学修士論文, 2005.