

DNA 計算を用いた SAT の解法

坂本典子[†] 村上和樹[†] 大倉慎平[†] 青山真之[‡] 會澤邦夫[‡]
 島根大学大学院総合理工学研究科[†] 島根大学総合理工学部[‡]

1 はじめに

DNA 計算とは、DNA 分子の構造と分子生物学的な操作を活用して計算を行うことである。現在の我々が使用している計算機では計算量的に解を求めることが困難な問題を DNA 計算により解く方法が研究されている。

Lipton は、論理式の充足可能性問題 (satisfiability problem; SAT) の特殊形である 3-SAT について DNA 計算を用いて解いた [1]。全探索手法を用いており、最初に全ての解の候補を表現する DNA 分子を 2^n 本生成するが、これは問題のサイズに対して解の候補のサイズが指数関数的に増大するというスケール問題を抱えている。これを克服するアルゴリズムとして、Suyama らは幅優先探索法を用いて大幅に少ない分子量で 3-SAT 問題を DNA 計算で解いた [2]。

本研究では、Suyama らの幅優先探索法を用いて k-SAT, SAT を解くアルゴリズムを提案し、問題を解くために必要となる分子量をシミュレートした。

2 k-SAT アルゴリズム (図 1)

n 個の真理値割当 (x_1, x_2, \dots, x_n) , m 節から成る下のような連言標準形 (CNF) 式の k-SAT 問題を扱う。

$$(v_{1,1} \vee v_{1,2} \vee \dots \vee v_{1,k}) \wedge \dots \wedge (v_{m,1} \vee v_{m,2} \vee \dots \vee v_{m,k})$$

1) Initial Solution

部分問題の解の候補 (初期解) を生成する。論理変数の 1 番目から $k-1$ 番目それぞれが $\{0, 1\}$ を表現する全通りを DNA 分子で生成する。

2) Main Loop

ループ中の h 番目では x_1 から x_h の論理変数とその否定から構成される節に対して、その節の命題論理式と 1 番目から $k-1$ 番目の論理変数

割当から x_h の真理値割当を決定し、それを表現する DNA 鎖を追加する。ループを n まで繰り返すと、最終的に n 個の論理変数に対する割当を表す DNA 分子が生成され、解が求められる。

```
Function dna_ksat
(v1,1, v1,2, ..., v1,k, ..., vm,1, vm,2, ..., vm,k)
begin
/* Initial Solution */
begin
for i=1 to k-1
Amplify(T, TT, TF);
Apped(TT, XiT, Xi-1T/F XiT);
Append(TF, XiF, Xi-1T/F XiF);
T = merge(TT, TF);
end
end
/* Main Loop */
for h=k to n do
Amplify(T, TT, TF);
for j=1 to m do
if vj,k = xh then
TF = getvsat(TF, vj,1, ..., vj,k-1);
end
if vj,h = -xh then
TT = getvsat(TT, vj,1, ..., vj,k-1);
end
end
TT = Append(TT, XhT, Xh-1T/F XhT);
TF = Append(TF, XhF, Xh-1T/F XhF);
T = Merge(TT, TF);
end
return Detect(T);
end
function getvsat(T, v1, ..., vk-1):
begin
for p=1 to k-1
TT = Get(T, +XvpT);
end
return TT;
end
```

図 1. 提案する k-SAT アルゴリズム

3 DNA 計算を構成する分子演算

分子演算は、試験管の中にある DNA 分子を生物学的に操作するモデルのことであり、DNA 計算における“プログラミング言語”の役割を担う。以下の基本演算がある。

Merge (混合), Detect (検出), Amplify (増幅), Append (追加), Get (抽出)

A method to Reduce the Number of DNA molecules in SAT

Noriko SAKAMOTO[†], Kazuki MURAKAMI[†],

Shinpei Ohkura[†], Masayuki AOYAMA[‡] and

Kunio AIZAWA[‡]

Interdisciplinary Faculty of Science and Engineering,
 Shimane University

4 分子量のシミュレーション(1)

提案するアルゴリズムが k -SAT 問題を解くために必要とする分子量を見積もるため、 k -SAT 問題を以下の条件でそれぞれ 1000 例ずつ作成し、シミュレーションを行った (図 2, 図 3)。

リテラル数 $k = 2, 3, 4$
 変数 $N = [20, 24]$
 節数 $M = [4.3, 4.7] \times N$

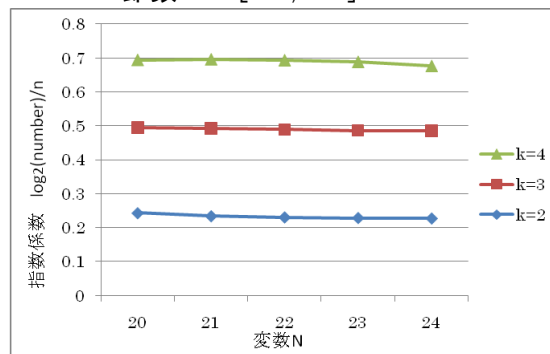


図 2. 必要な DNA 分子の数 (最大)

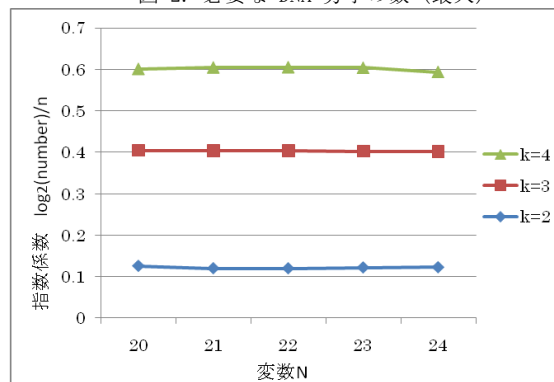


図 3. 必要な DNA 分子の数 (平均)

結果から、理論値では 2^n 必要な分子量を実験的には $k=2, 3, 4$ それぞれのとき、最大で $2^{0.25n}$, $2^{0.5n}$, $2^{0.7n}$ 程度で問題が解けることが分かった。

5 SAT アルゴリズム

SAT の任意の節 $C_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,k}\}$ は k の値に応じてリテラル数 3 であるいくつかの節に分解することができる [3]。SAT 問題にこの変換を用いて 3-SAT 問題に分解することで SAT 問題を解くアルゴリズムを提案した。

6 分子量のシミュレーション(2)

提案するアルゴリズムが SAT 問題 (random k -SAT) を解くために必要とする分子量を見積もるため、SAT 問題を以下の条件でそれぞれ 1000 例ずつ作成し、シミュレーションを行った (図 4, 図 5)。

リテラル数 $k = 2, 3, 4, 5, 10$
 変数 $N = [20, 24]$
 節数 $M = [4.3, 4.7] \times N$

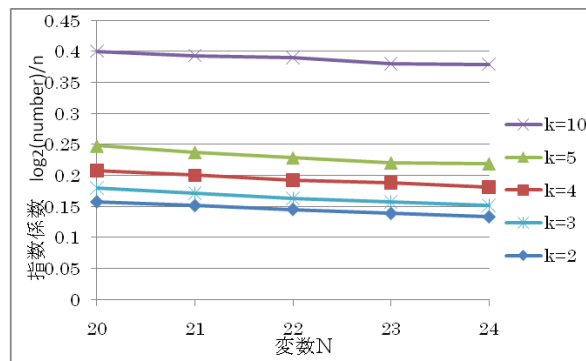


図 4. 必要な DNA 分子の数 (最大)

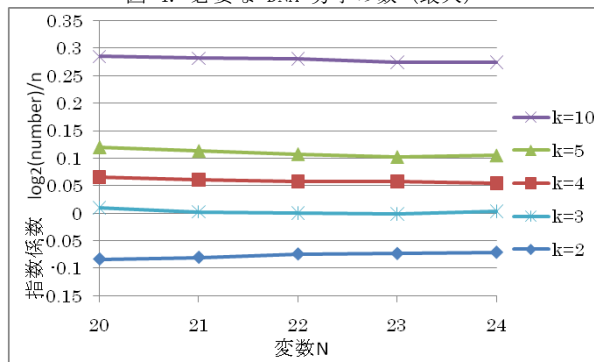


図 5. 必要な DNA 分子の数 (平均)

結果から、理論値では 2^n 必要な分子量を実験的には $k=2, 3, 4, 5, 10$ それぞれのとき最大で $2^{0.16n}$, $2^{0.18n}$, $2^{0.2n}$, $2^{0.25n}$, $2^{0.4n}$ 程度で問題が解けることが分かった。

7 まとめ

Suyama らが提案した 3-SAT を解く幅優先探索アルゴリズムを用いて k -SAT, SAT を解くアルゴリズムを提案し、必要となる分子量をシミュレートした。

参考文献

- [1] R. J. Lipton. DNA Solution of Hard Computational Problems. Science, Vol.268, 1995, pp.542--545.
- [2] H. Yoshida and A. Suyama. Solution to 3-SAT by breadth first search. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol.54, 2000, pp.9--22.
- [3] 竹内外史. P と NP —計算量の根本問題—. 日本評論社, 1996, 250p.