

DNA を用いた 0-1 整数計画問題の解法 An Algorithm for Computing 0-1 Integer Problem with DNA strands

厚山 耕太[†]
Kouta Atsuyama

藤原 暁宏[†]
Akihiro Fujiwara

1. はじめに

近年新しい計算パラダイムの 1 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 (NP 完全問題) 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。本論文では、この DNA 計算において NP 完全問題の 1 つである n 変数の 0-1 整数計画問題 (制約条件数 h 個) に対して、各入力を n 個の m ビットの 2 進数で表すことにより、 $O((m+2^n)n)$ 種類の DNA を用いて、 $O(n + \log h)$ ステップで解くアルゴリズムを提案する。

2. 準備

2.1 0-1 整数計画問題

0-1 整数計画問題とは、制約条件を満たし目的関数の値が最大となる変数の組み合わせを求める問題である。この問題では、1 つの目的関数と h 個の制約条件が与えられたとき、以下の様に目的関数の値を最大化する変数の組み合わせとそのときの目的関数の値を出力する。

$$\text{目的関数 } z = \sum_{j=0}^{n-1} c_j x_j \Rightarrow \text{最大化}$$

$$\text{制約条件 } y_g = \sum_{j=0}^{n-1} a_{g,j} x_j - b_g \geq 0 \quad (0 \leq g \leq h-1)$$

$$x_j \in \{0, 1\}$$

ここで、各 $a_{g,j}, b_g, c_j$ は与えられた整数である。

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*, *Selection*, *Cleavage*, *Annealing*, *Denaturation* という 8 つの基本操作が定義されており、本研究では *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。

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

各入力の制約条件の係数と定数項、及び、目的関数の係数を 2 進数で表現するものとする。

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、加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 $O(n)$ 個の対に対する加算は、 $O(mn)$ 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^[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 $O(mn)$ 種類の DNA を用いることにより $O(\log n)$ ステップで実行できる。□

3. アルゴリズムの概要

今回提案する 0-1 整数計画問題の解法は、以下の 4 ステップで構成されている。

Step1 全解探索を行うため、全ての変数の組み合わせ ($0 \leq i \leq 2^n - 1$) における各制約条件 g ($0 \leq g \leq h-1$) の値 $y_{i,g}$ ($0 \leq i \leq 2^n - 1, 0 \leq g \leq h-1$) を並列に求める。また、全ての変数の組み合わせにおける目的関数の値 z_i ($0 \leq i \leq 2^n - 1$) も並列に求める。

Step2 各制約条件 g ($0 \leq g \leq h-1$) において、制約条件式の値 $y_{i,g}$ ($0 \leq i \leq 2^n - 1, 0 \leq g \leq h-1$) が制約条件 ($y_g \geq 0$) を満たす組み合わせを抜き出す。抜き出した組み合わせに制約条件番号 g を表す一本鎖 DNA を繋ぐことで、他の制約条件を満たす組み合わせと区別をする。

Step3 各制約条件を満たす変数の組み合わせを 2 分木状に組み合わせることにより、 $O(\log h)$ ステップで全制約条件を満たす変数の組み合わせを求める。更に、制約条件を満たさない目的関数の値 z_i (m ビットで表現) を最小値である -2^{m-1} にする。

Step4 全制約条件を満たす変数の組み合わせの中から、目的関数の値が最大値をとるものを求める。求めた最大値と、そのときの変数の組み合わせを解として出力する。

[†]九州工業大学情報工学部電子情報工学科

4. アルゴリズムの詳細

0-1 整数計画問題の入力は、次のような一本鎖 DNA により与えられる。各変数の値の組み合わせは、0, 及び、1 の 2 進数の数列で表すことにする。さらに、各制約条件の係数、及び、定数項、目的関数の係数などの値も 2 進数表現であるとする。これらの 2 進数は前述のメモリ鎖を用いて表すものとする。

初期状態として、全ての変数の組み合わせ (000...0 から 111...1 までの 2^n 通り) を表すメモリ鎖を試験管 $T_{ans,g}$ ($0 \leq g \leq h-1$)、及び、 T'_{ans} に準備する。また、入力として制約条件の定数項、目的関数の値、制約条件の係数、目的関数の係数を表すメモリ鎖もそれぞれ $T_{const,g}$ ($0 \leq g \leq h-1$)、 T_{value} 、 $T_{subto,g}$ ($0 \leq g \leq h-1$)、 T_{obj} という試験管に準備する。ただし、 T_{value} 中の目的関数の値を表すメモリ鎖の初期値は 0 である。また、出力試験管 T_{output} は初期状態では何も入っていないものとする。

上記の入力、及び、初期状態に対する 0-1 整数計画問題を解く手続き *0-1 Integer Program* を以下に示す。

Step1 $j = 0$ とし、 $j = n - 1$ まで以下の (1-1) から (1-4) を j を 1 増加させながら繰り返す。

(1-1) 試験管 $T_{ans,g}$ において、組み合わせを表すメモリ鎖のうち、 j 番目の変数の値が 1 であるものを試験管 $T_{const,g}$ に抜き出す。

(1-2) 試験管 $T_{const,g}$ において、(1-1) で抜き出したメモリ鎖を用いて、 j 番目の係数を加える制約条件の値を表すメモリ鎖を試験管 $T_{tmp,g}$ に抜き出す。

(1-3) 試験管 $T_{subto,g}$ において、制約条件の j 番目の係数を表すメモリ鎖を $T_{tmp,g}$ に抜き出す。

(1-4) 試験管 $T_{tmp,g}$ 中で、抜き出した係数と制約条件の値を並列に加算し、その結果を試験管 $T_{const,g}$ に出力する。

Step2 以下の (2-1)、(2-2) を実行する。

(2-1) 全ての変数の組み合わせにおける制約条件の値を表すメモリ鎖が入った試験管 $T_{const,g}$ から、符合を表す $k = m - 1$ ビット目が 0 を表すもの (値が正のメモリ鎖) を試験管 $T_{tmp,g}$ に抜き出す。

(2-2) 試験管 $T_{tmp,g}$ において、(2-1) で抜き出したメモリ鎖を用いて、制約条件を満たす変数の組み合わせ番号を表す一本鎖 DNA と、制約条件番号を表す一本鎖 DNA を繋いだ一本鎖 DNA を生成する。

Step3 以下の (3-1) から (3-3) を実行する。

(3-1) 試験管 $T_{ans,g}$ ($0 \leq g \leq h-1$) に含まれる制約条件を全て満たす組み合わせ番号を表す一本鎖 DNA を求める。この操作は、2 つの試験管を 2 分木状に組み合わせることにより $O(\log h)$ ステップで実現される。制約条件を全て満たす組み合わせ番号を表す一本鎖 DNA は試験管 $T_{ans,0}$ に保存される。

(3-2) 試験管 $T_{ans,0}$ に格納されている全ての制約条件を満たす組み合わせ番号を表す一本鎖 DNA と、試験管 T_{value} に格納されている目的関数の値を表すメモリ鎖を、試験管 $T_{tmp,0}$ に移す。

(3-3) 試験管 $T_{tmp,0}$ において、全ての制約条件を満たす組み合わせ番号を表す一本鎖 DNA を用いて、制約条件を満たさない組み合わせの目的関数の値 (m ビットで表現) を最小値である -2^{m-1} にする。

Step4 以下の (4-1)、(4-2) を実行する。

(4-1) 試験管 T_{value} において、目的関数の値の中から最大値を求め試験管 T'_{ans} に抜き出す。

(4-2) 試験管 T'_{ans} において、試験管 T_{value} から抜き出した解となる目的関数の値を表すメモリ鎖を用いて、解となる組み合わせを表す一本鎖 DNA を求める。これらの解となる一本鎖 DNA を出力試験管 T_{output} に出力する。

この手続きについて、以下の定理を得ることができた。

定理 1 手続き *0-1 Integer Program* は、 n 変数 (m ビット) の 0-1 整数計画問題 (制約条件数 h 個) に対して、 $O((m+2^n)n)$ 種類の DNA 鎖を用いて $O(n + \log h)$ 時間で問題を解くことができる。□

5. まとめ

本論文では DNA 計算による 0-1 整数計画問題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 n の個数やビット数 m に正比例する長さの DNA 鎖を生成する点の解決や、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.