バルジ・内部ループを形成しない DNA 配列セットの設計

奥田　講平 *　　小林　聡*

本稿では，バルジ・内部ループを形成しない DNA 計算のための配列セットの設計問題を，超グラフの最大クリーク問題として定式化する．また，この問題を解くための分枝限定アルゴリズムを提案し，実際の DNA 配列設計に適用した結果を報告する．

Designing DNA Sequences without Bulge and Internal Loop Formation

Kohei Okuda†　　Satoshi Kobayashi †

This paper concerns the problem of designing DNA sequences for DNA computing such that any pair of them do not form bulge and internal loops. This problem is formulated as Maximum Clique Problem on Hypergraphs, and a branch and bound search algorithm is proposed. We report some experimental results of designing DNA sequences with this method.

1　はじめに

Adleman が有向ハミルトンパス問題に対して DNA 分子を用いた分子生物学的実験による解法を示して以来 ([1])，さまざまな問題に対する実験解法が提案され，DNA 計算という研究分野として定着してきた．この分野では，単に DNA 計算アルゴリズムを提示するだけでなく，実際の実験を通してその有効性を検証することも重要である．特に，実験の成功に密接に関わる重要問題である DNA 配列セットの設計においては，異なる情報を符号化した配列どうしが会合して安定な構造をとらないように配列を設計する必要がある．

従来の設計手法の中には，ハミング距離をベースとした配列間の非類似性の尺度を導入して配列セットを探索する手法が多く提案されている．しかしながら，これらの尺度では，バルジループや内部ループを形成する可能性を考慮できないので，構造形成の要因となるできるだけ少数の配列を除去する前処理が必要となる．本稿では，与えられた配列セットから，バルジ・内部ループ形成の要因となるできるだけ少数の配列を除去する問題を，超グラフの最大クリーク問題として定式化し，分枝限定法を用いた探索手法を提案する．そして，実際の DNA 配列設計に適用した結果を報告する．

*電気通信大学大学院電気通信学研究科
†Graduate School of Electro-Communications, The Univeristy of Electro-Communications

2　DNA 計算のための配列設計

DNA 計算アルゴリズムでは，個々の情報は同じ長さの DNA 配列に符号化されることが多いので，本稿もこれに従う．

DNA 計算のための配列設計で最も重要な制約は，配列どうしの想定外の会合により誤った計算結果が得られることを避けるために，

[**制約 1**] DNA 配列とその相補配列，およびそれらの連接部分との間の類似度を低く抑える

ことである．これ以外にも，各配列の融解温度を揃えることや，制限酵素を用いる場合にはその認識部位が指定された場所以外では存在しないこと等が要求される．

本稿では，DNA 配列がバルジ・内部ループを形成する可能性も考慮しながら，制約 1 を満たす配列集合を設計する問題を取り扱う．

配列間の非類似度を表すためには，一般にハミング距離を利用することが多い ([2])．しかしながら，これらの尺度では，バルジ・内部ループを形成して安定な構造をとる場合を考慮できない．この問題に対処するためには，配列どうしが会合した際の最小自由エネルギーを尺度として用いるのが適している．

配列 x，y が会合する際の自由エネルギーの最小値を $\delta(x,y)$ で表す．ただし，可能なループ構造としてはバルジ・内部ループのみを考える．配列集合 S とその相補配列集合から任意に選んだ $2m$ 個の配列 $x_1,...,x_m, y_1,...,y_m$ を考える．ただし，x_i が y_j

の相補配列となるような i, j が存在しないように選ぶものとする．このように選んだ配列に対して常に $\delta(x_1 \cdots x_m, y_1 \cdots y_m) \geq F$ が成り立つとき，配列集合 S は (m, F)-**安定**であると呼ぶことにする．本稿では，以下の問題を取り扱う．

[**問題**] DNA 配列集合 S と正の整数 m，自由エネルギー値 F が与えられたとき，(m, F)-安定な S の部分集合 S' で要素数が最大のものを求めよ．

最初の配列集合 S として，制約１以外の制約を満たしたものを用意できれば，上記の問題を解決することにより，すべての制約を満足した配列集合を設計することができる．

3 超グラフの最大クリーク問題への還元

任意の集合 X に対し，$|X|$ で X の濃度を表す．**超グラフ**は節点集合 V と辺集合 $\mathcal{E}$ の対 $H = (V, \mathcal{E})$ で与えられる．ここで，辺 $E \in \mathcal{E}$ は V の部分集合である．節点 $v \in V$ の次数 $deg(v)$ は，v を含む辺 $E \in \mathcal{E}$ の個数と定義される．V の空でない部分集合 X に対し，X の**ランク** $r(X)$ を，$r(X) = max\{|X \cap E| \mid E \in \mathcal{E}\}$ と定義する．このとき，$r(V)$ を H のランクという．任意の $E \in \mathcal{E}$ に対し，$|E| = r(V)$ となるとき，H は一様であるという．$H = (V, \mathcal{E})$ をランク h の超グラフとし，$r \leq h$ とする．V の部分集合 A は，$|A| < r$ であるかまたは，$|A| \geq r$ かつ A の任意の要素数 r の部分集合が少なくとも一つの H の辺に含まれるとき，ランク r のクリークであるという．$\omega_r(H)$ によって H におけるランク r のクリークの最大節点数を表す．

ここで，前節で定義した配列設計の問題を考える．配列集合 S を節点集合とし，要素数 $2m$ の部分集合 E が (m, F)-安定であるとき，かつそのときに限り，$E \in \mathcal{E}$ として，一様な超グラフ $(S, \mathcal{E})$ を構成する．このとき，S の節点数 $2m$ 以上の部分集合 S' が (m, F)-安定であることは，S' が H におけるランク $2m$ のクリークであることと同値である．従って，配列設計問題は，このように構成された超グラフのランク $2m$ の最大クリークを求める問題に還元できる．

より大きな m の値に対して，(m, F)-安定な配列セットを求めることが望ましいが，実際には，m の値が大きくなると，超グラフの辺の個数が指数関数的に大きくなり，自由エネルギーを計算するコストが膨大になる．そこで，以降では，$(2, F)$-安定な配列セットを求める問題を考える．このように m を制限しても，配列の連接部分での構造形成の可能性を考慮しているため，ある程度適切な配列セットが設計できると考えている．以下，特に断りがない限り，クリークとはランク 4 のクリークを指し，$\omega(H)$ は $\omega_4(H)$ を指すものとする．

4 分枝限定法による最大クリークの探索

4.1 クリークの深さ優先探索

超グラフ上の最大クリーク探索の基本探索アルゴリズムとしては, ある時点で保持しているクリーク Q に, Q に含まれない節点 (節点は順序付きである) を付け加えることによって, より大きいクリークを生成していく深さ優先探索を用いる.

初期設定として $Q := \phi$, $Q_{max} := \phi$, $R := V$(全節点集合), $E :=$ 全ての辺, とし，２つの手続き *select-E,expand-HG* からなる一連の操作を行う.

まず, *select-E* において, E 内のある辺, e を選び出し, この辺を構成する 4 節点をクリーク Q に付け加える. そして, 任意の $q_1, q_2, q_3 \in Q$ と $r \in R$ の 4 節点間に辺が存在するような節点 r の集合 R_e を求め, 新たな候補節点集合 R_e を *expand-HG* に与え, 操作 *expand-HG* により極大クリークを得る. そして e を含むクリークの探索が終了したら, $Q = \phi$ として新たな辺を選び出して同様の探索を繰り返す.

次に *expand-HG* では, 与えられた候補節点集合 R 内のある節点 p を選び出して，任意の $q_1, q_2 \in Q$ と $p, r \in R$ の 4 節点間に辺が存在するような節点 r の集合 R_p を求め, p をクリークに付け加える $(Q := Q \cup \{p\})$. そして得られた新しい候補節点集合 R_p を再び *expand-HG* に与えるという操作を再帰的に行う. そして $R_p = \phi$ となった時得られている Q が極大クリークである. 更に探索を行うため R から p を取り除いたものを R と置き直す．また，Q からも p を取り除き, R 内の新たな節点 p を選び出すことによって再び同様の探索を行う. そして $R = \phi$ となったら, その操作が呼び出された時点に戻ることにより深さ優先探索が行われる.

ここでは最大クリークを１つ抽出するということを目的としているので, より大きな極大クリークが抽出されるたびに Q_{max} を更新していけばよい.

しかしながら, この手順の操作を行うには全ての組み合わせについて評価を行うことになり, かなり

効率が悪い．そこで以下のような分枝限定アルゴリズムにより効率化を図る．

4.2 分枝限定アルゴリズム

最大クリーク抽出の過程は，全節点集合を根とし，R と，R 内の節点を Q に加えることによって得られる新たな候補節点集合 R_p を各頂点とし，それらを結んだ枝によって表現される探索木である．また，その枝を分枝と呼ぶ．何らかの上界を用いて分枝の数を減らし，解の探索領域を小さくするものが分枝限定法である．分枝限定する際には，簡素で精度の良い，何らかの上界を求めることが重要である．

基本的な分枝限定としては以下のようなものが考えられる．$|Q|+|R| \leq |Q_{max}|$ が成立する場合は，探索を続けても $|Q_{max}|$ より大きな極大クリークは明らかに存在しない．従って，分枝を行わず探索を終了する基本アルゴリズムを，図 1, 2 に示す．

また，各節点の次数について，次数の大きい節点は最大クリークに含まれる可能性が次数の小さい節点のそれより大きいと考えられる．探索の前処理として各節点を次数の非減少順に整列し，次数の大きい節点を含む辺から探索するアルゴリズムを DG とする．

4.3 計算機実験

上記基本アルゴリズムと DG とを実装し，ランダムに生成された超グラフに対して評価を行った．ここで，n は節点数，d は枝密度とする．なお，枝密度 d は $d = \frac{|E|}{\binom{n}{4}}$ として定義される．計算機実験では，$d = 0.3, 0.5, 0.7, 0.9$, $n = 30, 50, 70$ に対して，それぞれランダムに生成した超グラフを適当な数 (2〜5) 個用意し，その分枝数，実行時間の平均値を計測した．基本アルゴリズムの分枝数に対する DG の分枝数の割合 rt，および抽出された最大クリークのサイズ $\omega(H)$ も表 1, 2 に示す．

表 1，2 では，DG は基本アルゴリズムより優れた結果を示している．また，rt に注目してみると，節点数の小さいものは大きいものに比べ，より効率良く解を得ている．

4.4 分枝限定の可能性

富田らは，一般のグラフの最大クリーク問題に分枝限定法を適用する際に，彩色数が最大クリークの節点数の上界になることを利用してその有効性を示している ([3])．ここでは，この手法を超グラフの場合に一般化することを考える．

$|E| > k$ を満たす任意の $E \in \mathcal{E}$ に対して，E に属

```
procedure select-E(E)
  begin
    R : すべての節点集合;
    E : すべての辺集合;
    for all e ∈ E do
        e :=  a hyperedge in E;
        Q := e を構成する 4 節点;
        R_e := q_1, q_2, q_3 ∈ Q と r ∈ R の 4 節点に
               辺が存在するような節点 r の集合
        if |Q| > |Q_max| then Q_max := Q fi
        if R_e ≠ φ then
            expand-HG(R_e)
        fi
        Q := φ;
    od
    Exit:
    end{of select-E}
```

図 1: *select-E*

```
procedure expand-HG(R)
  begin
    while R ≠ φ do
        p:=  a vertex in R;
        if |Q| + |R| > |Q_max| then
            R_p := 任意の q_1, q_2 ∈ Q と
                   p, r ∈ R の 4 節点に
                   辺が存在するような
                   節点 r の集合
            Q := Q ∪ {p};
            if R_p ≠ φ then
                expand-HG(R_p);
            else if |Q| > |Q_max| then
                    Q_max := Q fi
            fi
            Q := Q − {p};
        else goto Exit;
        fi
        R := R − {p};
    od
  Exit:
  end {of expand-HG}
```

図 2: *expand-HG*

する節点が k 色以下で彩色されないように節点を塗分けるのに必要な色数の最小値を，$\chi_k(H)$ で表す．$\chi_1(H)$ は通常の彩色数の定義に一致する．このとき，H がランク r のクリークならば

$$|H| \leq \chi_{\lceil \frac{r}{2} \rceil}(H) + \lceil \frac{r}{2} \rceil - 1$$

が成り立つ．ここで，$r = 2$ としたとき，通常のグラフに対する彩色数と最大クリークの節点数の関係が得られる．

次に，次数を用いて分枝限定を行う手法を考える．H の節点の最小次数を g とする．このとき，H がランク r のクリークならば

$$g \geq \binom{|H|-1}{r-1}$$

が成り立つ．ここで，$|H| = n$ とすると

$$n \leq (g \times (r-1)!)^{\frac{1}{r-1}} + r - 1$$

表 1: 平均分枝数の比較

d	n	基本アルゴリズム	DG アルゴリズム	rt[%]
0.3	30	1,742	1,158	66.5
	50	26,286	20,656	78.6
	70	130,842	109,772	83.9
0.5	30	15,859	10,259	64.7
	50	240,298	186,936	77.8
	70	1,469,782	1,231,157	83.8
0.7	30	88,156	53,944	61.2
	50	1,553,375	1,214,470	78.2
	70	9,786,556	8,077,226	82.5
0.9	30	1,050,476	714,230	68.0
	50	39,946,460	30,719,019	76.9
	70	485,320,216	413,563,327	85.2

表 2: 平均実行時間の比較 [sec]

d	n	$\omega(H)$	基本アルゴリズム	DG アルゴリズム
0.3	30	5	0.01	0.01
	50	5	0.15	0.13
	70	6	0.82	0.77
0.5	30	6	0.02	0.01
	50	6	0.38	0.31
	70	6	2.17	2.02
0.7	30	7	0.06	0.05
	50	7	1.38	1.17
	70	7-8	10.31	9.48
0.9	30	9	0.76	0.59
	50	10	39.73	35.21
	70	10	586.87	566.75

が導かれる．ここで $r = 2$ としたとき，通常のグラフに対する次数と最大クリークの節点数の関係が得られる．

これらの関係を利用して分枝限定することにより，分枝数が減少することが期待される．ただし，実際に探索時間を削減するためには，$\chi_k(H)$ の上界を効率良く計算する手法の開発が必須である．また，二つ目の上界を利用する場合，節点数が多いときは，次数に対して実際のクリークサイズが小さいので効果が少ないと思われる．従って，候補節点数がある程度少なくなったところで適用することを考えている．

これらの分枝限定を効果的に行うための手法について現在検討している．

5 DNA 配列設計への応用

有田らによって提案されているテンプレート法 ([2]) を用いて設計した長さ 10 の 36 本の DNA 配列を用意した．この配列セットは，配列とその接合部におけるミスマッチ数が最小でも 4 であることが保証されており，融解温度の差は，配列の濃度 5.0 pmol, 1M NaCl の環境下で 4.2 度以内に抑えられている．

自由エネルギーの下界値 $F = -9.5 \sim -6.0$ に対する配列セットの設計結果を表 3 に示す．なお，結果は割愛したが，DG は基本アルゴリズムより少ない計算時間・分枝数で解を得ている．

表 3 と表 2 を比較すると，枝密度 d が比較的同じような値であっても，抽出された最大クリークの大きさは表 3 の方が明らかに大きい．これは各 DNA 配列どうしの関係に何らかの偏りが存在するためだと考えられる．

表 3: DNA 配列への応用

F	d	$\omega(H)$
-6.0	0.144419	7
-6.5	0.253510	8
-7.0	0.481759	9
-7.5	0.625448	10
-8.0	0.739428	12
-8.5	0.811442	14
-9.0	0.869688	16
-9.5	0.912945	19

6 おわりに

本稿では, 超グラフ上の最大クリークを抽出する分枝限定アルゴリズムを提案し, 計算機による実験を行った．そしてその結果より, 探索の前処理として, 節点を次数順に並べるというヒューリスティックが分枝限定に対して有効であることを示した.

次に本稿で提案した分枝限定アルゴリズムを DNA 配列設計へ応用した結果から，更に精度が良いと考えられる DNA 配列セットが抽出できることを示した．今後は彩色数を用いた分枝限定アルゴリズムについて議論し，効率化をはかる手法を検討していく．

謝辞

最大クリーク抽出アルゴリズムに関する文献・ソースコードを御提供頂いた富田悦次教授，関友和氏に感謝致します．また，DNA 配列データの作成に御協力頂いた長津和也氏に感謝します.

参考文献

[1] L. Adleman, Molecular Computation of Solutions to Combinatorial Problems, *Science*, **266**, pp.1021-1024, 1994.

[2] M. Arita and S. Kobayashi, DNA Sequence Design Using Templates, *New Generation Computing*, **20**, pp.263-277, 2002.

[3] 関友和，富田悦次，”分枝限定法を用いた最大クリーク抽出アルゴリズムの効率化 ”，信学技報, COMP 2001-50, pp.101-108, 2001.