数理解析研究所講究録
第1554巻 2007年 250-257

局所探索法に基づくDNA配列設計手法

川下 優 * 小野 廣隆 † 定兼 邦彦 † 山下 雅史 †
* 九州大学大学院システム情報科学府
† 九州大学大学院システム情報科学研究院

1 はじめに

近年，DNA分子からなる塩基配列を利用したナノ技術・ナノコンピューティングが注目されている．DNA分子はワトソン・クリック相補性に基づいた結合・乖離反応を起こすが，これらの反応は自律的・並列的に起きるため，処理速度やエネルギー効率の面からその有用性が期待されている．また，分子の微小性から莫大な情報収納量が期待されている．塩基配列集合とワトソン・クリック相補性を利用することで，Adleman [1] は，生化学実験において有向ハミルトンパス問題を解くことに成功した．また，上述の情報格納を目指した分子メモリ [7] なども考案されている．

しかしながら，塩基配列の反応は自律的であるために，意図しない反応を引き起こすこともあり，上述の技術には揺らぎが存在する．そのため，塩基配列を利用した技術にはロバストネスが求められ，これは塩基配列集合の設計により確保されることが知られている [3]．同時に，集合中の配列数は，各技術における資源の数となるため，集合のサイズは大きいことが望まれる．そこで，本研究では塩基配列を利用した技術におけるロバストネス・資源数確保のために，サイズの大きな塩基配列集合を設計することを目標とする．ロバストネス確保のための制約はさまざまな制約が考案されているが，現在もそれは研究段階にある．このため，制約が変更されても，配列設計可能な手法が求められている．

本研究では配列集合設計問題を組合せ最適化問題と捉え，現実的には解を求めることが困難である組合せ最適化問題に対して，近似解を求めることの有用性が知られている局所探索法 [14] を用いるアルゴリズムを提案する．局所探索法の中でも，可変深度探索近傍法 (VDS:Variable Depth Search) [13] と呼ばれる動的に近傍を制御する手法を適用させ，解探索の効率化を図る．また，局所探索法は制約などが変化しても，解法の枠組を変更することなく解を求めることができるため，配列設計に適した手法といえる．

動的に近傍制御を行いながらシンプルな近傍操作を適用させることで，いくつかの計算機実験において，既存の手法よりも良い配列集合を設計することに成功した．

2 準備

2.1 塩基配列とワトソン・クリック相補性

塩基配列は生体高分子であり，A,T,C,Gのアルファベットを用いて表現する塩基から構成される．また，塩基配列は方向性を持つ一本鎖であり，その両端は5′,3′のどちらかで表現される．塩

基配列を構成する塩基数を**配列長**と呼ばれる．ここで，配列長 n の塩基配列は $s = s_1 s_2 \cdots s_n$ と表現され，$s \in \{\mathrm{A,T,C,G}\}^n$ となる (図 1)．

また，塩基配列は**ワトソン・クリック相補性**と呼ばれる特性を持つ．これは，4 種類の塩基のうち，A-T 間，G-C 間でのみ水素結合が生じるというものである．このとき，塩基配列の構造上，二本の塩基配列は逆方向でなければならない．特に二本の塩基配列中のすべての塩基において塩基対が生じる場合，**相補の関係**にあるといい，二本の配列の一方を**主配列**とすると，もう一方を**相補配列**と呼ぶ．ここで，相補性に基づき $\overline{\mathrm{A}} = \mathrm{T}, \overline{\mathrm{T}} = \mathrm{A}, \overline{\mathrm{G}} = \mathrm{C}, \overline{\mathrm{C}} = \mathrm{G}$ と表すと，s の相補配列を $\bar{s} = \overline{s_n}\ \overline{s_{n-1}} \cdots \overline{s_1}$ と表現できる．このとき，A-T 間，G-C 間で常に水素結合が生じるとは限らない．

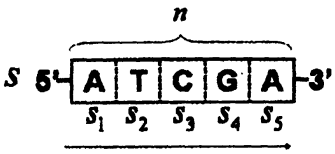

図 1: 塩基配列の図

2.2 塩基配列と形態

塩基配列はワトソン・クリック相補性による水素結合により，その形態が与えられる．例えば，図 2 と図 3 はともに同じ塩基配列であるが，生じている水素結合は異なるため，異なる形態であると考える．このように，同様の塩基配列を与えても，塩基配列が取り得る形態は数多く存在する．

図 2: 二本鎖形態

図 3: 二本鎖形態

2.3 自由エネルギー

塩基配列は，その塩基配列の形態により自由エネルギーと呼ばれる値が定められる．同一の塩基配列においても，形態が異なる場合には，自由エネルギーの値も異なる．自由エネルギーは最大値が 0 の実数で表現される．各形態のエネルギーは，その形態を構成する各ループ構造の持つエネルギーの和により近似される．

ここでループ構造とは，配列と水素結合により構成されるループのことを表す．特に，塩基配列が与えられたとき，取り得る自由エネルギーの中で最小のものを，最小自由エネルギーと呼ぶ．最小自由エネルギーは，塩基配列により一意に定められる．また，自由エネルギーが低い形態ほど安定することが知られており，塩基配列は自由エネルギーが低い形態になりやすくなる．よって，最小自由エネルギーを取る形態が最も安定した形態となる．最小自由エネルギーは動的計画法を用いることで，二本の配列長 n_1, n_2 のとき，$O((n_1+n_2)^3)$ で求めることができる [2]．

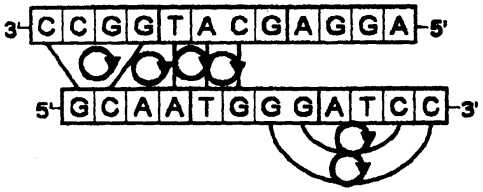

図 4: 自由エネルギー概念の図

2.4 局所探索法

局所探索法とは，現実的時間で厳密解を求めることが非常に困難（NP 困難）でありながらも，

最適性の保証は無くとも精度が十分に高い似解が求まれば良いとされるときに用いられる発見的手法である．特に組合せ最適化問題の多くはNP困難であることが知られており，そのような問題に対して高精度の近似解を求める場合には，局所探索法は有用であることが知られている [14].

組合せ最適問題は，**解** x, **目的関数** f とし，$f(x)$ を最小化（最大化）する問題と定式化できる．このとき，x に少しの変化を加えることで得られる解集合を**近傍**と呼び，$N(x)$ と表す．すべての $x' \in N(x)$ において $f(x) \leq f(x')$（もしくは $f(x) \geq f(x')$）を満たすとき，x を近傍 $N(x)$ における**局所最適解**という．局所探索法は，この局所最適解を求める解法である．ある解から近傍内の解を生成するために加える操作を**近傍操作**という．また，局所最適解は多数存在することが多いという特徴がある．

局所探索法の基本的な戦略は以下の通りである：適当な解 x の近傍 $N(x)$ の中に改善解 x' があれば $x := x'$ とし，解の移動を行う．これを繰り返し，近傍中に改善解が無くなればそれを返す．

局所探索法において，解の精度と時間はトレードオフの関係にあることが知られている．例えば，近傍のサイズが大きければ大きいほど解の改善は行われやすいが，反面，時間は増大してしまう．

一般に，近傍中には改善解が複数存在し，近傍中をどのような順序で調べ，どのような改善解に移動するのかについては，様々な戦略がある．これを**移動戦略**といい，代表的なものとしては，近傍中で改善解が見つかり次第移動する**即時戦略**がある．

以上より，性能の良い局所探索法の設計は，適当なサイズの近傍の選択，高速な近傍探索，そして解構造に適した探索制御について考慮する必要がある．

ここで，探索制御に関する重要な二つの概念である集中化と多様化について述べる．組合せ最適化問題には，「良い解同士は似通った構造を有する」ことが多いことが経験的に知られている．このことを表す概念は **POP**(proximate optimality principle) と呼ばれる．そこで，局所最適解を得た後に，構造の似ている解を集中的に探索し，その周辺に潜んでいるより良い解を発見しようとすることがある．これは**探索の集中化 (intensification)** と呼ばれている．一方，集中化に力を入れすぎると，同じ解を何度も探索したり，改善解が周辺に存在しないのに探索を続けるといった無駄が生じる．そこで，今までとは異なる構造を持つ解を探索する必要が生じる．これは**探索の多様化 (diversification)** と呼ばれる．

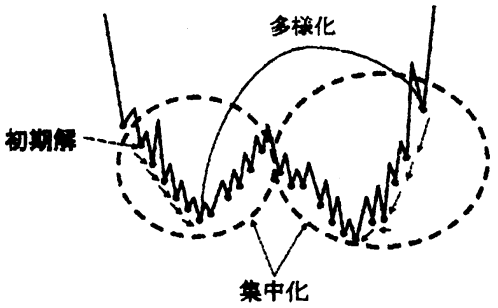

図 5: 局所探索法における集中化と多様化を表す概念図

この相反する二つの概念を，うまくバランスをとりつつ探索することが高速な局所探索設計において重要であると考えられている．

3 配列集合設計問題

塩基配列設計の研究において，配列集合設計問題が考えられており，既存の研究ではこの問題を取り扱っているものが多い．この配列集合設計問題とは，Adleman の実験 [1] に用いられた塩基配列集合に求められる条件を簡易化したものとなっている．

本研究においても，この問題を取り扱うこととする．

3.1 定式化

配列集合設計問題は，配列長 n である配列の集合 S を設計する問題である．このとき，S は以下の二点が要求される．

(a) $\forall s \in S$ は s 以外と結合して安定しない．

(b) S のサイズは大きい．

(a) の条件を満たすための制約はさまざま考案されている．多くの制約は，組合せ的制約と熱力学的制約に分類することができる．組合せ的制約は主にハミング距離を利用するものが多く [3, 4, 5, 6, 9, 10, 11]，熱力学的制約は最小自由エネルギーを利用するものが多い [6, 11]．組合せ的制約は，熱力学的制約を近似的に表したものと捉えられている．このため，組合せ的制約は熱力学的制約より簡易ではあるが，(a) の条件を満たすための指標としての精度は熱力学的制約に劣る．

既存研究の多くにおいては簡易である組合せ的制約を取り扱っている．しかしながら，ロバストネス確保の指標としての精度の高い熱力学的制約を用いることが近年重要視されてきており，熱力学的制約を利用した研究も行われるようになってきた．本研究においては，熱力学的制約を用いる．詳細は後述する．

以上より，配列集合設計問題は以下のように定式化できる．

入力 配列長 n，制約及び制約定数．

出力 制約を満たし，且つ，サイズができる限り大きい S．

3.2 制約

s, s' 二本鎖が取る最小自由エネルギーを $\Delta G(s, s')$ で表し，τ を制約定数とするとき，上記 (a) の条件は，以下の制約として表現することができる．

(a-1') $\Delta G_1(S) \stackrel{\text{def}}{=} \min_{s,s' \in S}\{\Delta G(s, s')\} > \tau$

(a-2') $\Delta G_2(S) \stackrel{\text{def}}{=} \min_{s,s' \in S, s \neq s'}\{\Delta G(s, \overline{s'})\} > \tau$

(a-3') $\Delta G_3(S) \stackrel{\text{def}}{=} \min_{s,s' \in S}\{\Delta G(\overline{s}, \overline{s'})\} > \tau$

本論文では，これら三つの制約について考える．

4 局所探索法に基づくアルゴリズム

筆者らは，過去に配列設計問題に対するアルゴリズムを提案している [?]．本論文では，過去の手法を改良・単純化した配列設計手法を提案する．本提案手法では，以下の手順を用いて配列設計を行う．

step 1. 集合サイズを定め，局所探索法により制約を満たす集合を設計.

step 2. (1) で得た集合に，新たに配列を加える.

これを繰り返すことで，制約を満たし，且つ，サイズができる限り大きい集合を設計する.

ただし，本提案手法は過去の研究と類似する部分 (具体的には，*min_related*, *farthest*-part, *join*-part, **step 2** での配列追加法) がある．これらについての説明は省略する (詳細は [8] 参照).

4.1 近傍と評価関数

効率的な探索の実現には適切な近傍定義が不可欠である．近傍については，以下のように定義する.

$$N(S) \stackrel{\text{def}}{=} \{S' | \text{ Sets that flip 1 letter in a sequence belongs to } S\}$$

評価関数は以下のように定義する.

$$\Delta G_{min}(S) \stackrel{\text{def}}{=} \min\{\Delta G_1(S), \Delta G_2(S), \Delta G_3(S)\}$$

$\Delta G_{min}(S) > \tau$ の時，制約を満たしていることとなる.

4.2 提案アルゴリズム

本研究では探索の集中化実現に向けて**可変深度近傍探索法 (Variable Depth Search:VDS)** [13] を利用する．これにより局所最適解を得る．また探索の多化実現を図るために，二種類の**反復局所探索法 (Iteration Local search:ILS)** を利用する．反復局所探索法は *farthest* と *join* と定義する二つの動作により実現する.

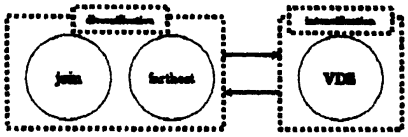

図 6: アルゴリズム概念図

4.2.1 *VDS* による集中化

制約を満たす集合を設計を効率的に行うために，**可変深度近傍探索法 (Variable Depth Search : VDS)** [13] を用いて探索を行う.

これは単純な近傍操作を連鎖的に複数回適用することで生成され得る新たな解集合を近傍と定義する方法である．この手法の特徴は，改悪があってもその先を探索することにより，より良い解を発見しようとする点にある．このような近傍は，現在の解を根とした木として表現でき，これを**探索木**と呼ぶ．VDS では，解空間上に仮想的な探索木を深さ優先で生成しながら探索を行う．探索木上においてあるノードから子ノードへ移動することは，近傍操作を一回行うことを意味し，ここでは，文字を一つ書き換えることを意味する．制限を加えなければ，探索木の形状が爆発的に増えるので問題に応じて形状を変えたり，形状あらかじめ限定しておく必要がある.

そこで，本研究では探索木を以下のように，深さに応じて *shallow* 部と *deep* 部に分けた探索木を用いる．このとき深さは，根の解からの文字書き換え回数と等しくなる.

但し，*shallow* 部と *deep* 部には，基本的に以下の制約を設ける．

Shallow-Part:

shallow 部のノードは改悪を許しながら，近傍操作により得られる全ての解を子として持つ．

Deep-Part:

deep 部のノードは，自身より評価の高い解を一つだけ即時戦略にて探索し，唯一の子とする．ただし，改善解が無い場合は子を作らない．このようにして，探索木の深さは変化する．

基本的には，上記の指針に従い探索木を構成しながら深さ優先で探索を行う．改善解を見付けた場合には，その解を根とし再び探索を行う．改善解が見付からない場合は，現在の探索木の根を局所最適解として返す．

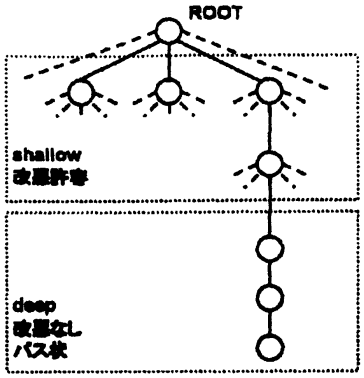

図 7: 提案手法の探索木

上述の方針のみでは木の制限が十分でないため，以下に追加の木の制限方法を示す．

shallow 部では，文字の変化順番が異なるだけで同じ解が現れることがあるので，*shallow* 部で既に生成した解については二度目以降は生成をしない．さらに，深さ二以降では既に文字変化を行った配列の文字変化のみを許容する．過去の研究において，「複数の文字を変化させる場合，一つの配列のみを変化させる方が，多数の配列の文字を少しずつ変化させるより高効率」という事実を組合せ的制約にて示しているためである [8]．また，これにより木のサイズを大幅に制限することができる．

deep 部では，根への逆戻りを防ぐために，文字変化を行うノードから根へのパス上のノードで既に文字変化を行っている文字を変化することを禁じる．しかし，パス上で既に二本以上の配列で文字変化が起きている場合には根への逆戻りが生じにくいと判断し，この禁止は行わない．

deep 部の深さは，状況に応じて定まるが，*shallow* 部の深さについては決めておく必要がある．ここで，*shallow* 部で最も深いノードの数に着目する．これが多い場合には，その後の *deep* 部での探索数は増えるため，改善解が見付かる可能性は高くなるが，探索時間は増えてしまう．これは，局所探索法における解の精度と時間はトレードオフを表している．*shallow* 部で最も深いノードの数は *shallow* 部の深さに依存する．そこで，*shallow* 部で最も深いノードの数がある数以下になるように，*shallow* 部の深さを決定する．

本論文では，$min\_related = 0$ である配列の文字の書き換えに基づく探索は実行しない．$m_{\mathrm{R}}(S)$ を $min\_related \neq 0$ である配列の数を表すとする．ここで，$f_i(S)$ を *shallow* 部で深さ i のノード数とすると，以下のように記述できる．

$$f_i(S) \;=\; m_{\mathrm{R}}(S)\begin{pmatrix} n \\ i \end{pmatrix} 3^i.$$

shallow 部で深さは，*shallow* 部で深さ 1 かつ $m_{\mathrm{R}}(S) = m$ の場合の $f_{1\{m_{\mathrm{R}}(S)=m\}}(S) = m\binom{n}{1}3 = 3mn$ と比較して定めるとする．ここで，P_i を比較のためのパラメータとし，MAX_N を *shallow* 部の最大の深さとする．$f_i(S) \leq 3mn \cdot P_i$ かつ $i \leq MAX\_N$ が満たされる i より 1 小さな値を *shallow* 部で深さとする．これを書き換えると，以下の値が *shallow* 部の深さとなる．

$$\max\left\{ i-1 \mid m_{\mathrm{R}}(S)\binom{n}{i}3^{i-1} \leq mn \cdot P_i \text{ and } i \leq MAX\_NUM \right\}.$$

4.2.2 *ILS* による多様化

反復局所探索法 (Iteration Local search:ILS) とは，局所最適解に辿り着いた際にその解の評価が不十分である場合には，近傍操作より大きな解変化を改悪を許して与え，局所最適解から脱出し，新たな解を探索する方法である．反復局所探索法を用いる場合には，条件を定めなければ探索が終了しない．本論文では，実行時間が定められた制限時間を越えた時，終了とした．

反復を行う際に注意すべきことがある．多様化実現のために大きな解変化を与える必要があるが，その場合には場合には改悪が大きくなる．また，評価の悪い解から局所最適解を探索する場合には，評価の良い解からよりも時間がかかる．つまり，適切な改変化を与えることが重要である．

上記の理由より，本論文においては二種類の改変化をを使い分ける．*farthest* では小さな改変化を与え，*join* ではより大きな解変化を与える．通常は *farthest* による解変化を利用するが，解変化が不十分と判断した場合は *join* を利用する．

5 実験結果

提案手法を実装し，計算機実験を行った．実験環境は，CPU-Pentium4 3.3GHz,MEM-512MB,OS-VineLinux-3.2, コンパイラ-gcc3.3.2 である．最小自由エネルギー計算には `PairFold Package` [2] を利用した．

提案手法とほぼ同一の制約を用いて配列設計を行った Garzon [6] らの結果と比較を行う．同一の制約を用いている場合，設計できた配列数が多い方が好ましい結果となる．実験結果は以下の表．“Garzon ら”，“提案手法” と表している部分は，設計できた配列の本数を表している．time は提案手法による実行時間を示している．

length	τ	Garzon ら	提案手法	time
8	-5.4	132	158	2 hours
8	-6.5	173	250	2 hours
12	-9.9	617	395	24 hours
12	-11.2	1424	439	24 hours
15	-14.9	42	152	2 hours
15	-15.5	96	152	2 hours

表 1: 実験結果

配列長 8，15 の時には同一制約において，Garzon らの結果より大きな配列集合を設計することに成功した．これらの結果は提案手法の有効性を表している．ただし，配列長 12 で集合サイズが大きなものと比較すると，Garzon らの結果に劣っている．最小自由エネルギーの計算には $O(n^3)$ が必要であり，また，それを集合の配列の全ての組合せにおいて行う必要がある．このため，集合の評価には $O(m^2n^3)$ と大きな時間が必要となる．このため，提案手法では，近傍操作ごとに評価を行うために，時間を多く必要とし，良い結果が得られにくいと考えられる．しかしながら，Garzon らの手法で得られる集合を初期解とし提案手法を適用することは可能である．そのため，これらの結果によって，提案手法の有用性が失われるわけではない．

6 まとめと今後の課題

本論文では，DNAテクノロジーにおいて必要となる塩基配列集合設計の設計に取り組んだ．塩基配列集合設計問題を組合せ最適化問題と捉え，局所探索法に基づく手法を提案した．制約には熱力学的指標を採り入れ，動的に近傍操作を行う可変深度近傍探索法を適用することで，効率的な探索を目指した．

計算機実験では，配列集合にサイズが比較的小さいものに関しては，既存の研究と比較し良い結果を得ることができた．評価計算において多くの時間が消費されている．この部分の改良も課題となる．

参考文献

[1] L.Adleman, "Molecular computation of solutions to combinatorial problems", Science 226, pp.1021–1024, 1994.

[2] M.Andronescu, Z.C.Zhang, and A.Condon, "Secondary Structure Prediction of Interacting RNA Molecules", Journal of Molecular Biology, Vol 345/5 pp.987-1001. Web page: www.rnasoft.ca/download.html

[3] M.Arita, A.Nishikawa, M.Hagiya, K.Komiya, H.Gouzu,and K.Sakamoto, "Improving Sequence Design for DNA Computing", Proceedings of 5th Gnenetic and Evolutionary Computation Conference (GECCO'00), pp.875–882, 2000.

[4] M.Arita, and S.Kobayashi, "DNA Sequence Design Using Templates", New Generation Computing, Vol.20(3), pp.263–273, 2002.

[5] Y.Asahiro, "Simple Greedy Methods for DNA Word Design", Proc. 9th World Multi-Conference on Systemics, Cybernetics and Informatics, Vol.III, pp.186–191, 2005.

[6] M.H.Garzon, V.Phan, S.Roy, and A.J.Neel, "In Search of Optimal Codes for DNA Computing", Proceedings of 12th DNA Based Computers, Springer LNCS vol.4287 pp.143–156, 2006.

[7] S.Kashiwamura, M,Yamamoto, A.Kameda, T.Shiba, and A.Ouchi, "Potential for Enlarging DNA Memory: The Validity of Experimental Operations of Scaled-up Nested Primer Molecular Memory", BioSystems 80, pp.99–112, 2005.

[8] S.Kawshimo, H.Ono, K.Sadakane, and M.Yamashita, "DNA Sequence Design by Dynamic Neighborhood Searches", Proc. 12th DNA Computing, LNCS(4287) pp.157–171, 2006.

[9] S.Kobayashi, T.Kondo, and M.Arita, "On Template Method for DNA Sequence Design", Proceedings of 8th DNA Based Computers, Springer LNCS vol.2568 pp.205–214, 2002.

[10] D.Tulpan, H.Hoos, and A.Condon, "Stochastic Local Search Algorithms for DNA Word Design", Lecture Notes in Computer Science 2568, pp.229–241, 2003.

[11] D.Tulpan, and H.Hoos, "Hybrid Randomised Neighborhoods Improve Stochastic Local Search for DNA Code Design", Springer LNCS vol.2671, 2003, pp.418–433.

[12] D.Tulpan, M.andronescu, S.Changf, M.Shortreed, A.Condon, H.Hoos, and L.Smith, "Thermodynamically based DNA strand design", Nucleic Acids Res. 2005 33(15), pp.4951–4964,2005.

[13] M.Yagiura, T.Yamaguchi, and T.Ibaraki, "A Variable Depth Search Algorithm for the Generalized Assignment Problem", in: S.Voss, S.Martello, I.H. Osman and C.Roucairol,eds., Meta-Heuristics: Advances and ztrends in Local Sarch Paradigms for Optimization, Kluwer Acad.Publ., pp.459–471, 1999.

[14] 柳浦 睦憲, 茨木 俊秀, "組合せ最適化－メタ戦略を中心として", 朝倉書店, 2001.