エピスタシス尺度に基づくリンケージ同定手法の提案

棟 朝 雅 晴†

遺伝的アルゴリズム（Genetic Algorithm，GA）はビルディングブロックを交叉により組み合わせることで効果的な探索を実現しているが，そのためにどの遺伝子座がビルディングブロックを構成しうるのかを調べるリンケージ同定が重要となる．リンケージ同定に関してはこれまでにも確率モデルに基づく方法や非線形性もしくは非単調性を基に判断する手法が提案されている．本論文では，遺伝子座間に存在する非線形性を検出することでリンケージ同定を行う LINC（Linkage Identification by Nonlinearity Check）を発展させ，それぞれの遺伝子座のペアに対してエピスタシス（非線形性）尺度を定義し，それに基づいてリンケージの同定を実現する手法を提案する．

Proposal of a Linkage Identification Method Based on Epistasis Measure

Masaharu Munetomo†

Genetic Algorithms realize effective search by exchanging building blocks through genetic recombinations. To realize effective genetic search, linkage identification becomes important which detects a set of loci tightly linked to form a building block. Several methods are already proposed to identify linkage such as linkage learning algorithms based on probabilistic models and linkage identification procedures based on nonlinearity or non-monotonicity detection. In this paper, we extend the Linkage Identification by Nonlinearity Check (LINC) which identifies linkage based on nonlinearity detection by defining an epistasis measure for each pair of loci to realize linkage identification with the epistasis measure.

1. はじめに

遺伝的アルゴリズム（Genetic Algorithm，以下 GA と略す）は広範囲の問題に対して安定した探索性能を有するとされているが，その探索の本質は交叉によるビルディングブロックの交換にある．ビルディングブロックとは最適解を生成するために有用な部分解（GA の個体文字列においては部分列）を意味し，それらビルディングブロックを正しく認識し，それを集団内の個体間で互いに交換し組み合わせることが GA の探索性能を向上させるために必要である．しかしながら，一点交叉などによる単純 GA では，ビルディングブロックが正しく認識，交換されるかは，個体を符号化する方法に強く依存してしまう．GA を現実の問題に適用した場合において,「予想以上にうまく問題を解くことができる」「問題を解くためにまったく役立たない」といった互いに相反する評価がなされることがある．これには様々な理由が考えられるが，その理由の1つとして，ビルディングブロックの交換が適切に行われるような符号化が実現できたときにはきわめて効率的な探索ができる反面，それができないような符号化が行われた場合には交叉による探索の効果がまったく働かないことが考えられる．

そこで，ビルディングブロックを正しく認識するために，遺伝子座間でのリンケージを正しく同定する必要がある．リンケージとは元々遺伝学の用語で「世代を超えて同時に継承される複数の遺伝子の組合せ」を意味しており，以下では「ビルディングブロックを構成することで世代を超えて継承される部分文字列に対応する文字位置（遺伝子座）の組合せ」として定義する．上で述べられた効果的な符号化とは，このリンケージが文字列中で密になっている（すなわち，同じリンケージに属する文字が互いに近くある）符号化を指すこととなる．問題が単純であり，その性質が探索前にある程度分かっている場合にはリンケージを密にするような符号化を事前に行うことが容易であると考えられるが，そのような条件が満たされない場合にはリンケージを同定する必要性が生じる．本論文ではこれまでに提案されてきたリンケージ同定アルゴリ

† 北海道大学情報メディア教育研究総合センター
Center for Information and Multimedia Studies, Hokkaido University

ズム LINC（Linkage Identification by Nonlinearity Check）[10]をさらに発展させ，遺伝子座間でのエピスタシス尺度を基にして同定を行うことで，より広範囲の問題に対して正しくリンケージを同定することを目指した LIEM（Linkage Identification with Epistasis Measure）を提案する．LINC では厳密な線形・非線型基準によりリンケージを同定しているが，LIEM においてはエピスタシス尺度を基準としたより柔軟な判断基準を用いているため，LINC では正確なリンケージ同定ができないような問題についても正確な同定が可能となることが期待される．本論文ではいくつかのテスト関数を例にとり，LINC と比較した LIEM の有効性について論じる．

2. リンケージ同定

分割統治法に代表されるように，問題を部分問題に分割することが問題を効率的に解くための 1 つの有効な方法である．たとえば，2 つの部分関数の和で表される関数の最大化問題を解く場合には，それぞれの部分関数の最大化を独立に行うことで問題を効率良く解くことができる．GA においては，ビルディングブロックという観点から部分問題への分割が行われ，それぞれのリンケージにおいてそのリンケージ内における近似最適解としてビルディングブロックを生成し，それらを組み合わせることで全体としての最適解を求めることを目指している．GA の場合，分割統治法とは異なり，厳密な分割を行うのではなく，ビルディングブロックという概念，交叉というオペレータにより間接的な方法で問題を大まかに分割して探索するというアプローチをとっている．

1 点交叉による単純 GA など古典的な GA ではこの問題分割が陽には取り扱われず，交叉オペレータにより間接的にビルディングブロックを処理している．この場合，問題に合わせて交叉オペレータを設計することでリンケージを保証する必要がある．また，messy GA [2]ではスキーマを部分列のリストとして直接符号化した個体を用いることで直接ビルディングブロックを操作しているが，リンケージの同定については考慮されていないため，初期個体集団としてある長さ以下の可能なすべてのスキーマを生成する必要が生じ，そのために必要な計算コストが膨大となる．一方，GEMGA（Gene Expression Messy GA）[6),7]では個体文字列中のそれぞれの遺伝子座に対し，そこでの文字変化による適応度の変化量を計算することで局所性を検出してリンケージを間接的に同定する試みを行った．さらに，LLGA（Linkage Learning GA）[5]では，円状に符号化された個体表現と 2 点交叉に類似した特殊な交叉手法を用いることで，探索の過程で動的にリンケージを生成することを可能とした．ただし，それぞれの部分関数の全体の適応度への寄与度が均一的であるような問題では，LLGA はうまくリンケージを生成できない．これは，LLGA がそれぞれの部分関数の全体への適応度への寄与度の差を間接的に検出することでリンケージ生成を行っているためである．

以上の手法では探索の過程で間接的にリンケージを生成している．一方，リンケージをより直接的な方法で同定するため，確率モデルに基づいて集団内の確率分布を基に同定を行う手法などが提案されている[3),8),14),15)]が，近年，遺伝子座間での非線形性などを検出する手法に関する研究が進められている．その最初の試みである LINC（Linkage Identification by Nonlinearity Check）[9)~11)]は，文字の変化にともなう適応度の変化量をそれぞれの遺伝子座のペアについて計算し，それが非線型の効果を生むかどうかを検出することでリンケージを同定する．リンケージが正しく同定されれば，そのリンケージを単位とした交叉を行うことで速やかに解を得ることが可能となる[9)]．

LINC はビット列で表現された個体を対象にしているため，以下，本論文ではビット列で表現された文字列についてのみ考える．一例として，部分関数の線形和として定義される関数 $f(x) = \sum_{n=1}^{N} f_n(x_n)$ を考える．x_n をそれぞれビット列として符号化してつなげたものが全体としての個体を表現する文字列 s となる．ここで，s の i 番目ビットを変化させた場合の適応度の変化量を $\Delta f_i(s)$ とし，i 番目，j 番目のビットを同時に変化させた場合の適応度の変化量を $\Delta f_{ij}(s)$ とする．もし，i と j が異なる部分関数を表すビット列から選ばれた場合，適応度の変化はそれぞれ独立であるので，$\Delta f_{ij}(s) = \Delta f_i(s) + \Delta f_j(s)$ となる．しかしながら，i と j が同じ部分関数 f_n に関するビット列内から選ばれた場合には，f_n の性質に依存して必ずしも $\Delta f_{ij}(s) = \Delta f_i(s) + \Delta f_j(s)$ となるとは限らない．このような場合，可能な個体のうち少なくとも 1 つで $\Delta f_{ij}(s) \neq \Delta f_i(s) + \Delta f_j(s)$ となる（逆に，すべての可能な個体について等号が成り立つのであれば，そのような部分関数はさらに小さな部分関数に分割可能であるといえる）．以上をまとめると，遺伝子座 i と j が同じ部分関数の中に存在する（すなわちリンケージとしてまとまっている）条件は，可能な個体のうち少なくとも 1 つにおいて $\Delta f_{ij}(s) \neq \Delta f_i(s) + \Delta f_j(s)$ となることとなる．

LINC においては，それぞれの遺伝子座のペア

```
algorithm LINC

P = initialize N strings
for each s in P
   for i = 0 to length-1
      s' = Perturb(s, i);
      df1 = f(s') - f(s);
      for j = i to length-1
         if i != j then
            s' = Perturb(s, j);
            df2 = f(s') - f(s);
            s'' = Perturb(s', i)
            df12 = f(s'') - f(s);
            if | df12 - (df1 + df2)| > e then
             /* nonlinearity detected between i and j */
               add j to the linkage_set[i];
               add i to the linkage_set[j];
           endif
         endif
      endfor
   endfor
endfor
```

図 1 LINC の実行手順
Fig. 1 Execution flow of the LINC.

(i,j) に関して以下のように適応度の変化量 $\Delta f_i(s)$, $\Delta f_j(s)$, $\Delta f_{ij}(s)$ を求める．

$$\Delta f_i(s) = f(..\bar{s_i}.....) - f(..s_i.....) \quad (1)$$

$$\Delta f_j(s) = f(.....\bar{s_j}..) - f(.....s_j..) \quad (2)$$

$$\Delta f_{ij}(s) = f(..\bar{s_i}.\bar{s_j}..) - f(..s_i.s_j...), \quad (3)$$

ここで，$f(s)$ は個体 s の適応度で，$\bar{s_i} = 1 - s_i$ ($0 \to 1$ または $1 \to 0$) は s 中の i 番目の文字の変化を示す．

ここで，すべての可能な個体集団（現実的には十分なサイズを有する個体集団中）の中で，

$$\Delta f_{ij}(s) \neq \Delta f_i(s) + \Delta f_j(s) \quad (4)$$

を満たす個体が存在する場合，現実的には誤差 $e > 0$ を許して，

$$|\Delta f_{ij}(s) - (\Delta f_i(s) + \Delta f_j(s))| > e \quad (5)$$

が満たされる場合に，遺伝子座 i と j は互いにリンケージを持つとし，それぞれをリンケージ集合に入れる．この具体的な手順を示したのが，**図 1** である．

実行手順としては，はじめに十分なサイズ N の初期個体集団を生成し，それぞれの遺伝子座のペア (i,j) に対して，その個体集団中のすべての個体について式 (5) が満たされるかどうかをチェックする．ここで本質的なのは，十分なサイズの個体集団に対して適用されなければならない，という点である．これは，たとえば，トラップ関数のような GA にとって難しいとされる問題であっても，一部には線形のアトラクタが存在するためで，1 個体についてのみチェックするなど不十分なサイズの個体集団が用いられた場合には正確な結果を得ることができない．LINC の場合，必要な個体数はビルディングブロックの最大長を k とした場合に $O(c2^k)$（c：許容誤差に依存する定数）となることが示されている[10]．

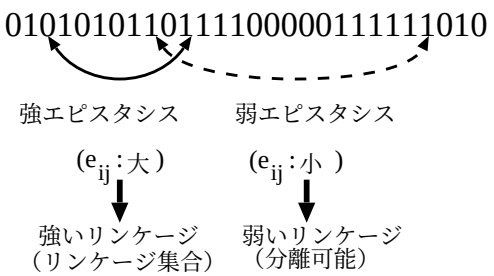

図 2 LIEM の概念
Fig. 2 An overview of the LIEM.

3. エピスタシス尺度に基づくリンケージ同定

LINC では厳密な非線型基準が満たされるか否かでリンケージを同定しているため，現実の問題では適用が難しい場合も考えられる．誤差 e を導入することである程度対応は可能ではあるが，より柔軟な基準でリンケージを同定することができれば，さらに広範囲の問題へ適用できることが期待できる．本論文で提案するエピスタシス尺度によるリンケージ同定 (Linkage Identification with Epistasis Measures, LIEM) は，遺伝子座の間の非線形性の強さを表す尺度を定義し，それをもとに統一的な基準でリンケージを同定する手法である．LIEM の概念を**図 2** に示す．

ここで，それぞれの遺伝子座ペアに対してエピスタシス（非線型性）の尺度が定義され，その値が大きい（すなわちエピスタシスが強い）遺伝子座どうしをリンケージとして認識し，その値が小さい（すなわちエピスタシスが弱い）遺伝子座はそれぞれ分離可能なものとして認識する．エピスタシス尺度に関しては，Davidor による Epistasis Variance[1]や Naudts らによる Bit Decidability[13]などの提案がなされているが，これらはある問題が GA にとって難しいかどうかを測るために導入された尺度であり，尺度が適応度関数全体としてのエピスタシスの度合いを表すため，本論文で目的としているリンケージの同定にはまったく役立たない．

ここでエピスタシス尺度をどのように設計するかが重要な課題となる．GA においては，目的関数のパラメータや微分係数など使用せず，個体の適応度としての目的関数値のみを前提として最適化を行うため，エピスタシス尺度についても目的関数値のみを用いて計

算する必要がある．尺度としては種々の可能性が考えられるが，本論文では LINC における非線形性判定条件をもととした尺度を採用し，それぞれの遺伝子座のペア (i,j) $(i \neq j)$ に関するエピスタシス尺度を以下の式で定義する．

$$e_{ij} = \max_{s \in P} |\Delta f_{ij}(s) - (\Delta f_i(s) + \Delta f_j(s))|, \quad (6)$$

ここで，$\Delta f_i(s)$，$\Delta f_j(s)$ はそれぞれ個体 s に関して i 番目，j 番目の遺伝子座の文字を変化させた場合の適応度の変化量で，$\Delta f_{ij}(s)$ は i 番目，j 番目の遺伝子座の文字を同時に変化させた場合の適応度の変化量で，LINC における定義（式 (1)，(2)，(3)）と同様である．この（6）式は，LINC における非線形条件（式 (4)）において，等号が満たされる場合からの差分（すなわち非線形の大きさ）を集団内のそれぞれの個体に対して計算し，その最大値を求めている．言い換えると，LINC では e_{ij} が 0 かどうかでリンケージを判断しており，この尺度はそれを一般化したものとしてとらえることができる．尺度の計算に最大値を用いているのは，たとえばトラップ関数のようにある 1 点できわめて強い非線形性を有しているが，それ以外では線形であるような関数の場合，2 乗平均などの平均化された尺度ではその探索における困難さが反映され難いためである．

ここで，正しくリンケージを同定するためには，LINC と同様に十分な大きさの個体集団 P を用いて e_{ij} を計算することが必要である．単に 1 個体において e_{ij} を計算した場合，その個体に関する局所的な非線形性を検出することはできるが，問題全体として正しいリンケージを求めることはできない．そこで，LINC と同様にビルディングブロックのオーダー（最大長）を k とした場合に $O(c2^k)$ 個の個体を用意する必要がある．

LIEM の実行手順を**図 3** に示す．LIEM では，LINC と同様，はじめに十分なサイズ（$O(c2^k)$）の初期個体集団をランダムに生成する．そして，それぞれの遺伝子座のペア (i,j) $(i,j = 0,1,\cdots l-1)$ に関してエピスタシス尺度 e_{ij} を計算する．この尺度の計算のため，式 (6) に基づき，初期化されたすべての個体に関して，線形条件からの離れ度合いを計算し，その最大値を求める．エピスタシス尺度の計算後，求められた尺度 e_{ij} をその値の大きい順にソートし，その順に遺伝子座を最大のオーダー k 個となるまで結合していきリンケージ集合とする．ただし，エピスタシス尺度の値が 0 に近い場合，すなわち，ある 0 に近い値 $e > 0$ よりも e_{ij} が小さい場合にはリンケージに加えない．ここで，

```
algorithm LIEM

N = c*2^difficulty;
P = initialize N strings;
/*  Calculate epistasis measure e[i][j]  */
for i = 0 to l-1
   for j = 0 to l-1
      e[i][j] = 0;
      if i != j then
         for each s in P
            s' = perturb(s, i);
            f1 = fitness(s') - fitness(s);
            s'' = perturb(s, j);
            f2 = fitness(s'') - fitness(s);
            s''' = perturb(s', j);
            f12 = fitness(s''') - fitness(s);
            ep[s] = |f12 - (f1+f2)|;
            if(ep[s] > e[i][j]) then e[i][j] = ep[s];
         endfor
      endif
   endfor
endfor
/*  Generate linkage_set[k]
    where k = 0, 1,..., difficulty-1  */
for i = 0 to l-1
   for j = 0 to l-1
      id[j] = j;
   endfor
 /*  sorting of e[i][j] with j */
   for j = 0 to l-1
      for k = j to l-2
         if e[i][j] < e[i][k]
            swap(e[i][j], e[i][k]);
            swap(id[j], id[k]);
         endif
      endfor
   endfor
 /* select linkages */
   for k = 0 to difficulty-1
      if(e[i][k] > e) add  id[k] to linkage_set[i];
      else break;
   endfor
endfor
```

図 3 LIEM の実行手順
Fig. 3 Execution flow of the LIEM.

最大のオーダー k がアルゴリズムのパラメータとなり，どの程度の長さまでリンケージを検出するか（言い換えるとどの程度までリンケージ同定のコストを負担するか）をユーザが指定することとなる．これは，GA は問題に対する仮定をおかないブラックボックス最適化であることから，k の値を問題から知ることが原理的にできないので，ユーザが推定する必要があるためである（ただし，問題によりその条件から k の値をある程度推定できる場合も多いと考えられる）．一般の GA においても，ユーザの推定によりパラメータ

が設定される場合が多いが，重要なパラメータの 1 つである初期集団サイズが理論的に 2^k に比例することが知られており[4]，初期集団サイズを決めることが交叉において取り扱うこととなるビルディングブロックの最大長（ここでの k に相当する）を間接的に決めていることになる．

4. 数値実験

ここでは，GA にとって難しい部分関数の線形和となっている関数だけではなく，従来の手法では同定が難しかった非線型関数となっている問題についても LIEM により正しいリンケージ同定ができることを数値実験により示す．テスト関数としては，GA にとって難しいとされるだまし問題のうち，典型的なトラップ関数の和をとったものを使用する．このテスト関数は GA にとって難しい部分問題であるトラップ関数とやさしい問題である線形和の組合せになっており，難しい部分は初期個体集団の多様性により部分解をビルディングブロックとして確保し，交叉によりそれぞれの部分解を組み合わせることで最適解を得ることができる．

具体的には以下に示される 5 ビットのトラップ関数の線形和である関数 $h(x)$ を使用する．

$$h(x) = \sum_{i=1}^{10} f_i(u_i). \tag{7}$$

$$f_i(u_i) = \begin{cases} 4 - u_i & \text{if } 0 \le u_i \le 4 \\ 5 & \text{if } u_i = 5 \end{cases} \tag{8}$$

ここで u_i は x を表現するビット列中の i 番目の部分列（長さ 5）に含まれる 1 の数を示す．最適解はビットすべてが 1 である場合となるが，局所解であるビットすべてが 0 となる解の方向へ探索がすすむようなアトラクタが存在する．また，この関数は多峰性関数である．この関数の大域解は 1 つであるが，局所解の数は $2^{10} - 1 = 1023$ 個存在する．よって，山登り法などの局所探索によっても最適解を発見することは困難であると考えられる．

$h(x)$ において，それぞれの部分関数に対応する部分列が密に符号化されている場合には単純 GA によっても解が得られることも期待できる．しかしながら，それぞれの部分列が疎に符号化されている場合，すなわち，それぞれの部分列を構成する 5 つのビットが互いに離れて符号化された場合には，ビルディングブロックが交叉により高い確率で切断され，最適解を得ることは絶望的となる．このような場合に，リンケージの同定が必要となる．一度正しくリンケージが同定されれば，その後はそれぞれのリンケージ内でビルディングブロックを探索し，リンケージを考慮した交叉により正しくビルディングブロックが組み合わされる．

表 1 リンケージ同定の正解率
Table 1 Percentage of linkage groups identified correctly.

$h(x)^n$	% of correct linkage	
n	LIEM	LINC
1	100	100
2	100	0
3	94	0
4	76	0
5	60	0

ここで，単なる線形和であれば LINC など従来手法であっても容易にリンケージが同定できることが予想されるので，本実験ではさらに関数全体を n 乗することで関数全体に非線形性を持たせることで，より同定が困難な関数とし，従来手法と提案手法のリンケージ同定結果の比較検討を行うこととする．以下の実験において用いられたパラメータとして，$k = 5$, $N = c2^k = 32(c = 1)$, $e = 0.001$ を用いた．この場合に必要な適応度評価の回数（計算量）は，$N \times 3 \times l^2/2 = 32 \times 3 \times 50^2 = 240{,}000$ 回となる．**表 1** に $h(x)$ を n 乗した関数 $h(x)^n$ に関して LINC および LIEM のリンケージ同定の正解率を示す．

ここで，LINC は単純な線形和の関数（$n = 1$ の場合）のみ正しくリンケージを同定している．一方 LIEM においては n の値が小さい場合にはほぼ 100% 正しく，n が比較的大きくなった場合でも 60 ∼ 70% 以上の正解率を出している．ここで，LINC，LIEM は符号化の方法に依存していないため，遺伝子座が疎に符号化されていたとしても（極端な場合として遺伝子座の位置がランダムに符号化されていても），まったく同じ結果を得ることができる．

LIEM によるリンケージ同定結果の具体例を**図 4** に示す．ここで，最初の数字（：の前の数字）はそれぞれの遺伝子座を示し，以降の数字はその遺伝子座と同じリンケージ集合に属する遺伝子座のリストを示している．テスト関数では 5 ビットごとに部分関数を符号化したので，$\{0, 1, 2, 3, 4\}$, $\{5, 6, 7, 8, 9\}$, $\cdots$, $\{45, 46, 47, 48, 49\}$ が正しいリンケージ集合となる．数字が網掛けとなっている部分は誤った結果を示している．結果を見ると，$h(x)^5$ の場合にいくつかの誤りが見られるが，$h(x)$，$h(x)^2$ ではすべて正しいリンケージ集合が得られている．

以上で得られた結果について考察するため，LIEM により求められたエピスタシス尺度を**図 5**，**図 6**，**図 7**

	h(x)		h(x)^2		h(x)^5
0:	0 1 2 3 4	0:	0 1 2 3 4	0:	0 1 2 3 4
1:	1 0 2 3 4	1:	1 0 3 2 4	1:	1 0 2 3 4
2:	2 0 1 4 3	2:	2 0 1 4 3	2:	2 0 1 3 4
3:	3 0 1 4 2	3:	3 0 1 4 2	3:	3 0 1 2 4
4:	4 0 1 2 3	4:	4 0 1 3 2	4:	4 0 1 2 3
5:	5 8 7 6 9	5:	5 8 7 6 9	5:	5 8 6 7 9
6:	6 8 7 5 9	6:	6 8 7 5 9	6:	6 5 7 8 9
7:	7 8 6 5 9	7:	7 8 6 5 9	7:	7 6 9 8 5
8:	8 5 6 7 9	8:	8 5 6 7 9	8:	8 5 6 7 9
9:	9 8 7 6 5	9:	9 8 7 6 5	9:	9 6 7 8 5
10:	10 13 12 11 14	10:	10 14 13 12 11	10:	10 14 13 11 6
11:	11 14 12 13 10	11:	11 14 13 12 10	11:	11 8 43 14 0
12:	12 10 11 13 14	12:	12 13 11 10 14	12:	12 8 18 6 0
	.		.		.
	.		.		.
43:	43 40 41 42 44	43:	43 40 41 42 44	43:	43 11 21 42 44
44:	44 40 41 42 43	44:	44 43 41 42 40	44:	44 42 43 40 41
45:	45 46 49 48 47	45:	45 46 49 47 48	45:	45 46 47 41 43
46:	46 48 45 47 49	46:	46 48 45 47 49	46:	46 45 48 47 0
47:	47 49 48 46 45	47:	47 49 48 46 45	47:	47 49 48 46 45
48:	48 46 47 49 45	48:	48 46 47 49 45	48:	48 46 47 49 0
49:	49 47 45 46 48	49:	49 47 48 45 46	49:	49 47 46 48 43

図 4 LIEM によるリンケージ同定の結果
Fig. 4 A result of linkage identification by the LIEM.

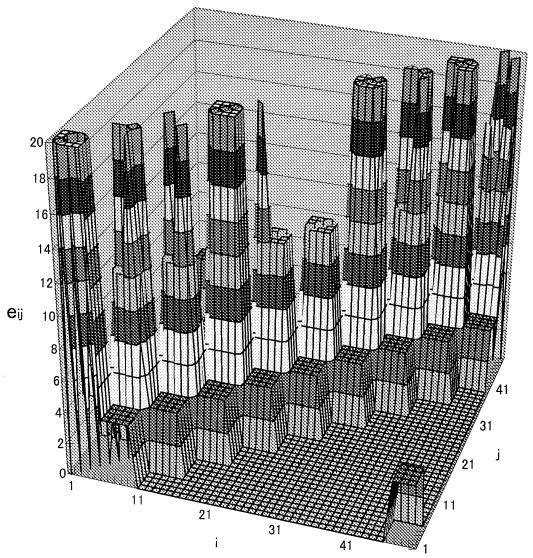

図 5 $h(x)$ に関するエピスタシス尺度の分布
Fig. 5 Distribution of epistasis measures of $h(x)$.

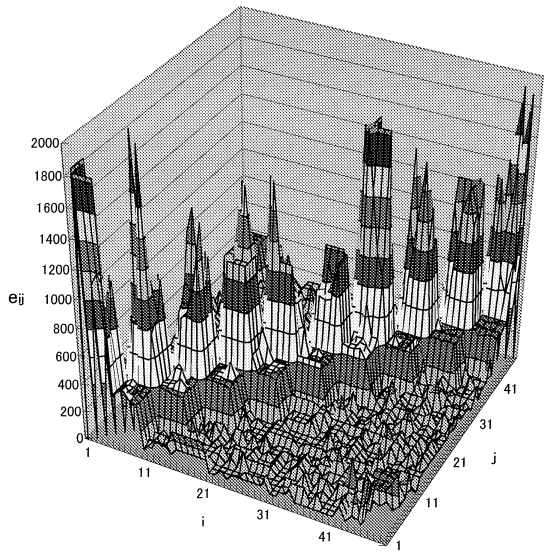

図 6 $h(x)^2$ に関するエピスタシス尺度の分布
Fig. 6 Distribution of epistasis measures of $h(x)^2$.

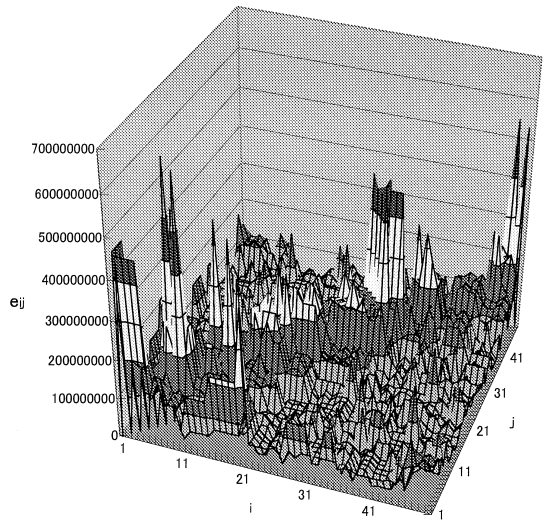

図 7 $h(x)^5$ に関するエピスタシス尺度の分布
Fig. 7 Distribution of epistasis measures of $h(x)^5$.

に示す．図の縦軸はエピスタシス尺度 e_{ij} の値を示し，残りの軸はそれぞれペアとなる遺伝子座 (i, j) の番号を示している．

図 5 に示される $h(x)$ に対応するエピスタシス尺度は，同じ部分関数に属する遺伝子座どうしで大きな値をとっており，明確にそれらの間のリンケージを同定できる基準となっている．また，図 6 における $h(x)^2$ に対応する尺度は関数全体としての弱い非線形性により若干ピークが不明瞭になってはいるが，正確にリンケージを同定できている．一方，図 7 に示される $h(x)^5$ の関数では，関数全体として持つ比較的強い非線型性により，リンケージ尺度の分布が複雑になることでリンケージ同定が不正確になっている．$h(x)$ や $h(x)^2$ など全体として線形や弱い非線型関数では，トラップ関数の持つ強い非線型性とそれら全体の関数の持つ弱い非線形性を判別することで正しくリンケージが同定できているが，$h(x)^5$ では全体としての非線型性が強くなってきていることから，トラップ関数の持つ非線型性との区別がつきにくくなり，結果としてリンケージ同定の精度が落ちているものと考えられる．

5. おわりに

本論文では，LINC など従来のリンケージ同定手法を発展させ，エピスタシス尺度に基づいたより一般的なリンケージ同定手法の枠組みを提案した．本論文で提案した LIEM は，エピスタシス尺度の値が大きい順に遺伝子座をリンケージ集合に加えていくため，部分関数に関する線形関数だけではなく，全体として弱い非線型性を有する問題に対しても正しいリンケージを生成することができ，現実の問題においても正しいリンケージ同定を実現することが期待される．

本手法においては，リンケージの同定に $O(Nl^2)$（N：個体数，l：個体長）の余分の計算コストを要する．実験で用いられているトラップ関数の和で表される関数などで，リンケージが疎に符号化されている場合（極端な場合としては遺伝子座がランダムに符号化されている場合）には，1 点交叉など従来手法を適用しても局所最適解へトラップされてしまい最適解を得ることはできない．このような場合にリンケージ同定を行うことで余分の計算コストを要するものの，正しく最適解を得ることが可能となる．

今後の課題としては，ノイズと同定精度との関連性を調べること，また，より広範囲の問題においてリンケージを正しく同定するため，問題の単調性，非単調性をも考慮したリンケージ同定手法である LIMD（Linkage Identification by non-Monotonicity Detection）[12] に基づいたエピスタシス尺度の設計があげられる．

参考文献

1) Davidor, Y.: Epistasis Variance: A Viewpoint on GA-hardness, *Foundations of Genetic Algorithms*, pp.23–35 (1991).
2) Goldberg, D.E., Korb, B. and Deb, K.: Messy genetic algorithms: Motivation, analysis and first results, *Complex Systems*, Vol.3, No.5, pp.493–530 (1989). (Also TCGA Report 89003).
3) Harik, G.: Linkage learning via probabilistic modeling in the ECGA, IlliGAL Report No.99010, University of Illinois at Urbana-Champaign, Urbana, IL (1999).
4) Harik, G., Cantú-Paz, E., Goldberg, D.E. and Miller, B.L.: The gambler's ruin problem, genetic algorithms and the sizing of populations, *Proc. 1997 IEEE Conference on Evolutionary Computation*, pp.7–12 (1997).
5) Harik, G.R. and Goldberg, D.E.: *Learning linkage*, *Foundations of Genetic Algorithms 4*, Belew, R.K. and Vose, M.D.(Eds.), San Francisco, pp.247–262, Morgan Kaufmann (1996).
6) Kargupta, H.: SEARCH, polynomial complexity and the fast messy genetic algorithm, Technical Report 95008, University of Illinois at Urbana-Champaign, Urbana, IL (1995).
7) Kargupta, H.: The gene expression messy genetic algorithm, *Proc. 1996 IEEE Conference on Evolutionary Computation*, pp.814–819 (1996).
8) Mühlenbein, H., Bendisch, J. and Voigt, H.-M.: From recombination of genes to the estimation of distributions I. Binary Parameters, *Parallel Problem Solving from Nature IV*, pp.188–197 (1996).
9) Munetomo, M. and Goldberg, D.E.: Designing a Genetic Algorithm Using the Linkage Identification by Nonlinearity Check, Technical Report IlliGAL Report No.98014, University of Illinois at Urbana-Champaign (1998).
10) Munetomo, M. and Goldberg, D.E.: Identifying Linkage by Nonlinearity Check, Technical Report IlliGAL Report No.98012, University of Illinois at Urbana-Champaign (1998).
11) Munetomo, M. and Goldberg, D.E.: Identifying Linkage Groups by Nonlinearity/Non-monotonicity Detection, *Proc. 1999 Genetic and Evolutionary Computation Conference*, pp.433–440 (1999).
12) Munetomo, M. and Goldberg, D.E.: Linkage Identification by Non-monotonicity Detection for Overlapping Functions, *Evolutionary Computation*, Vol.7, No.4, pp.377–398 (1999).
13) Naudts, B., Suys, D. and Verschoren, A.: Epistasis as a basic concept in formal landscape analysis, *Proc. 7th International Conference on Genetic Algorithms*, pp.65–72 (1997).
14) Pelikan, M., Goldberg, D.E. and Cantú-Paz, E.: Linkage Problem, Distribution Estimation, and Bayesian Networks, IlliGAL Report No.98013, University of Illinois at Urbana-Champaign, Urbana, IL (1998).
15) Pelikan, M., Goldberg, D.E. and Cantú-Paz, E.: BOA:The Bayesian optimization algorithm, *Proc. Genetic and Evolutionary Computation Conference 1999* (*GECCO-99*), pp.525–532, Morgan Kaufmann Publishers (1999).

(平成 14 年 1 月 24 日受付)
(平成 14 年 4 月 2 日再受付)
(平成 14 年 6 月 4 日採録)

棟朝 雅晴（正会員）

昭和 43 年生．平成 8 年北海道大学大学院工学研究科情報工学専攻博士後期課程修了．同年北海道大学大学院工学研究科システム情報工学専攻助手．平成 10 年〜11 年イリノイ大学基礎工学科遺伝的アルゴリズム研究室客員研究員．平成 11 年北海道大学情報メディア教育研究総合センター助教授（情報メディアシステム分野）．博士（工学）．遺伝的アルゴリズム，ネットワークシステム，分散処理システムに関する研究に従事．IEEE，ISGEC（International Society of Genetic and Evolutionary Computation）各会員．