

遺伝的アルゴリズムにおける交叉の Walsh 解析

古 谷 博 史[†]

遺伝的アルゴリズムにおいて最も重要な操作の 1 つである交叉の効果を用いて解析する。交叉が個体の分布に及ぼす効果は交叉行列として表される。交叉行列は 2 種類の 2×4 次元小行列の Kronecker 積を用いて分解することができ、その Walsh 変換は非常に簡単な式になる。交叉の役割の 1 つは、遺伝子間の相関を小さくし連鎖不平衡を減少させることにある。一般に、連鎖不平衡係数の総和が負の値をとると、進化の速度が遅くなるのが予想される。したがってそのような場合、交叉は連鎖不平衡係数の総和を零に近づけ、進化の遅れを回復する働きを持つ。積型適応度関数を例にとり、そのことを数値的に示す。

Walsh Analysis of Crossover in Genetic Algorithms

HIROSHI FURUTANI[†]

Crossover, which is one of the most important operators in Genetic Algorithms (GAs), is analysed by means of Walsh transformation. The effect of crossover on the distribution of individuals is described by using the crossover matrix. The crossover matrix can be decomposed into the Kronecker products of two 2×4 small matrices, and the Walsh transform of the crossover matrix has a very simple form. One of the crossover's roles is to decrease the correlation between genes and to reduce the linkage disequilibrium. In general, it is expected that when the sum of linkage disequilibrium coefficients becomes negative, it has the effect of retarding the speed of evolution. Therefore in such a case, crossover reduces the sum of linkage disequilibrium coefficients to zero, and recovers the speed of evolution. In numerical experiments, we show these effects of crossover in the GA on the multiplicative landscape.

1. はじめに

遺伝的アルゴリズム (Genetic Algorithms^{1),2})、以下 GA と略す) は、最適化など様々な分野でその有効性が認められ、現在では多くの領域における標準的手法としてその地位を確立したといえる。しかしながら、実際の GA においてどのような機構で最適解を探索しているのか、計算で用いる各種のパラメータをどう設定すればよいのか、問題をどのように表現すればよいのか、など解明されないまま残されている課題も多い^{2),3}。

現在、GA 計算における進化の過程を記述する最も標準的な理論として、Holland によるスキーマ理論¹) があげられる。しかし、GA の基礎理論としてのスキーマ定理については、多くの研究者からその有効性について疑問が投げかけられている⁴)。スキーマ定理の最大の問題は、進化のプロセスを不等式で記述した点に

あり、単に解の集団が世代の進展につれて改善していくという事実を述べた定理にすぎないという指摘がある。また、GA における代表的な遺伝的操作としては、選択、突然変異、交叉があるが、スキーマ定理では、突然変異と交叉は進化の途上で出現した近似解を破壊してしまう操作としてその効果が記述される。しかし、よく知られているように、GA では突然変異や交叉なしに最適解を得ることはほとんど不可能である。

GA の理論的基礎づけにおけるもう 1 つの有力な方法としてマルコフ連鎖解析^{5),6}) がある。Vose ら⁵) は、マルコフ過程の理論を用いて集団の進化を記述する厳密な式を導いた。しかし厳密性の代償として、彼らの進化方程式で使われている推移確率行列は非常に大きい次元を持ち、そのままの形では適用不可能である。たとえば、5 ビットの個体を 6 含む集団の進化を記述する方程式の次元は 200 万以上になってしまう。

我々は先の論文^{7),8}) において、個体数が非常に多い集団の進化を記述する数学的方法について提案し、その結果を数値実験により検討した結果を報告した。これらの論文では、Eigen の進化方程式をもとに^{9)~11})、

[†] 京都教育大学教育学部
Faculty of Education, Kyoto University of Education

GA における各遺伝子型の相対頻度の時間的変化を記述する方程式を導いた．第 1 の論文⁷⁾では，摂動論を用い突然変異率が小さい場合について，GA が収束した後の定常状態における遺伝子型の分布を導いた．また，進化方程式に表れる突然変異行列は，固有関数として Walsh 関数を持つことを示した．このように，突然変異の効果は Walsh 解析によって単純かつ直感的に分かりやすい形で表現できる．

本論文の目的は，我々が GA の解析において無視してきた交叉の効果を実験的に表現する手法について報告することである．Eigen の進化理論は DNA や RNA のような巨大分子の分子進化を記述しようとするもので，選択と突然変異に相当する効果は含んでいるが，交叉のように生物に固有な働きは取り入れられていない．交叉は，生物界における性に対応している．よく知られているように，生物になぜ性が存在するか，ということは生物学の最大の謎の 1 つである．同様に，なぜ GA において交叉が有効か，ということもよく分かっていない．交叉を単純に記述するとスキーマ定理のように，せっかく得られた最適解の近似解を破壊してしまう過程，ということになる．しかし我々は，経験的に交叉が多くの場合めざましい働きをするということも知っている．したがって交叉の効果を抑える場合に，従来のスキーマ定理とは異なった観点が必要になる．

我々は，交叉を Walsh 変換の手法を用いて解析した．交叉の効果が Walsh 変換により簡単に表現できることは，Vose らにより数値実験で明らかにされ¹²⁾，その後その解析的な表現が導かれた¹³⁾．しかし，その解析的表現が得られるまでには数年を要している．本論文では，交叉の効果を表す行列を Kronecker 積により表現し，その Walsh 変換を解析的に求める，という手順を採用する．この方法は，任意の交叉をきわめて簡単な形で表現することができ，Vose らの方法とは違った観点から交叉を解析することが可能となる．

我々の解析の特徴は，交叉と連鎖不平衡 (linkage disequilibrium) の関連に注目したところにある．集団遺伝学では，交叉は連鎖不平衡を解消し平衡状態を回復する操作としての意味を持つことはよく知られていた．しかし，GA の分野では連鎖についてあまり意識してこなかった．Holland 自身は，連鎖の重要性についてよく認識していたようで¹⁾，最近になり Goldberg ら¹⁴⁾を中心にして連鎖の意味が見直されるようになってきている．

本論文では，進化の速度と個体の Hamming 距離の分散が密接な関係にあることを指摘し，次に Ham-

ming 距離の分散において連鎖不平衡からの寄与が重要であることを明らかにする．このことから進化速度は連鎖不平衡に依存していることが分かる．そしてその結果として，交叉は進化速度に大きな影響を与えることになる．最後に，このことを種型適応度関数を用いて数値的に明らかにする．

2. 進化の数学的モデル

2.1 進化の数学的記述

まず，本論文で議論する進化モデルについて簡単にレビューし，使用する数学記号の定義を与える．ここでは，世代ごとに集団の構成要素が入れ替わる世代的 GA を採用し，進化の過程を差分方程式として記述する．また，個体数の十分多い集団を考え，確率的揺らぎを無視することができるものとする．したがって，GA の進化は決定論的方程式を用いて記述される．集団の構成要素は ℓ ビット固定長の 2 進ビット列で表され，集団の大きさは時間的に一定とする．遺伝的操作としては，選択に加えて突然変異と交叉を取り上げる．

このモデルでは全部で $n = 2^\ell$ 種類の遺伝子型が考えられるが， i 番目の遺伝子型に対応する 2 進ビット列 B_i は，非負整数 i の 2 進数表現を用いて表す．

$$B_i = \langle i_\ell, i_{\ell-1}, \dots, i_2, i_1 \rangle.$$

ここで i_k ($k = 1, \dots, \ell$) は， i の第 k ビットにおける値を表す．また

$$|i| = \sum_{k=1}^{\ell} i_k$$

でビット列 B_i に含まれるビット 1 の個数を表す．

$x_i(t)$ はビット列 B_i の世代 t における相対頻度を表す．これらをまとめてベクトル形で次のように表現することもある．

$$\mathbf{x}(t) = (x_0(t), x_1(t), \dots, x_{n-1}(t))^T,$$

ここで T はベクトルや行列の転置を表す．本論文では，遺伝的操作の効果を $x_i(t)$ の時間的変化として記述する．相対頻度 $x_i(t)$ は次の規格化条件を満足する．

$$\sum_{i=0}^{n-1} x_i(t) = 1. \quad (1)$$

また各 $x_i(t)$ は非負であり，すべての i について

$$0 \leq x_i(t) \leq 1$$

を満たす．

2.2 Kronecker 積

本論文では，Kronecker 積 (テンソル積) を用いて演算子や集団の状態を表現することが多い^{7),8),15)}．Kronecker 積については，あまりなじみのない読者が多い

と思われるので、ここで説明しておく。その数学的定義については、たとえば文献 16) のようなテキストを参照されたい。ここでは、簡単な例を示して定義に代えることにする。

次の 2 つの 2 次元ベクトル \mathbf{a} と \mathbf{b} を考える。

$$\mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}.$$

2 つのベクトルの Kronecker 積 $\mathbf{c} = \mathbf{a} \otimes \mathbf{b}$ は

$$\mathbf{c} = \mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} a_0 \mathbf{b} \\ a_1 \mathbf{b} \end{pmatrix} = (a_0 b_0, a_0 b_1, a_1 b_0, a_1 b_1)^T,$$

と定義される。積 \mathbf{c} の第 i 要素は、

$$c_i = a_{i_2} b_{i_1}, \quad i = \langle i_2, i_1 \rangle$$

により与えられ、 $c_1 = c_{01} = a_0 b_1$ 、 $c_3 = c_{11} = a_1 b_1$ のように各要素を直接計算することができる。Kronecker 積で表現することのできるベクトルや行列は、次のような関係を満たし、計算が簡単になることが多い。

$$\sum_{i=0}^3 c_i = \sum_{i_2=0}^1 \sum_{i_1=0}^1 a_{i_2} b_{i_1} = (a_0 + a_1)(b_0 + b_1).$$

2.3 Walsh 関数

次に Walsh 関数について述べる^{17),18)}。ℓ ビットのビット列 j に対する第 i Walsh 関数を次の式で定義する。

$$W_{ij} \equiv W_i(j) = \prod_{k=1}^{\ell} (-1)^{i_k \cdot j_k}, \quad (2)$$

ここで 2 つの整数 i と j について、次の 2 進数表現を用いた。

$$i = \langle i_\ell, i_{\ell-1}, \dots, i_1 \rangle, \quad j = \langle j_\ell, j_{\ell-1}, \dots, j_1 \rangle.$$

また Walsh 関数を $n = 2^\ell$ 次元の列ベクトルで表すこともある。

$$\mathbf{w}_i = (W_i(0), W_i(1), \dots, W_i(n-1))^T.$$

突然変異や交叉の効果を調べるためには、相対頻度 $x_i(t)$ そのものよりその Walsh 変換 $\tilde{x}_i(t)$ を用いた方が簡単になることが多い。

$$\tilde{x}_i(t) \equiv \sum_{j=0}^{n-1} W_{ij} x_j(t). \quad (3)$$

また、その逆変換は

$$x_i(t) = \frac{1}{n} \sum_{j=0}^{n-1} W_{ij} \tilde{x}_j(t). \quad (4)$$

で与えられる。以下、 $\tilde{x}_i(t)$ を Walsh 係数と呼ぶことにする。

Walsh 関数の定義からすべての j について $W_{0j} = 1$

となることを用い、規格化条件式 (1) から

$$\tilde{x}_0(t) = \sum_{i=0}^{n-1} x_i(t) = 1 \quad (5)$$

を得る。相対頻度のベクトル表現 $\mathbf{x}(t)$ は、Walsh 係数を用いて次のように表される。

$$\mathbf{x}(t) = \frac{1}{n} \sum_{j=0}^{n-1} \tilde{x}_j(t) \mathbf{w}_j. \quad (6)$$

以下では Walsh 係数のうち $|i| = k$ となる \tilde{x}_i を k 次の Walsh 係数と呼ぶことにする。Walsh 係数の次数は、GA 解析において重要な意味を持つことが明らかになる。そこで、 \tilde{x}_i の次数を明示的に示したほうが分かりやすい場合には

$$\tilde{x}_i \equiv \tilde{x}^{(k)}[b_1, b_2, \dots, b_k],$$

と表すことにする。ここで $k = |i|$ であり、 $b_1 < b_2 < \dots < b_k$ は 2 進ビット列 i におけるビット 1 の位置を表す。このとき

$$\tilde{x}^{(k)}[b_1, b_2, \dots, b_k] = \sum_{j=0}^{n-1} \prod_{m=1}^k (-1)^{j^{(b_m)}} x_j, \quad (7)$$

と表される。ここで $j^{(b_m)}$ はビット列 j の b_m におけるビットの値を表す。

2.4 選 択

ここでは、3 つの操作のうち最も解析が容易な選択を取り上げ、進化方程式とその解について述べる。選択には様々な方式があるが、その中でも代表的な比例選択を例にとる。

世代 $t+1$ における相対頻度は、世代 t における値から次式により与えられる。

$$x_i(t+1) = \frac{f_i}{\bar{f}(t)} x_i(t) \quad (i=0, \dots, n-1). \quad (8)$$

f_i は遺伝子型 B_i の環境に対する適応度を表し、 $\bar{f}(t)$ は世代 t における集団の平均適応度を表す。

$$\bar{f}(t) = \sum_{i=0}^{n-1} f_i x_i(t). \quad (9)$$

この連立差分方程式に対する解は容易に得られ¹⁹⁾

$$x_i(t) = \frac{f_i^t x_i(0)}{\sum_j f_j^t x_j(0)} \quad (i=0, \dots, n-1), \quad (10)$$

となることが分かる。

選択のみの進化方程式 (8) からは重要な性質が導かれる⁴⁾。式 (8) の両辺に f_i をかけ、すべての i について足し合わせれば

$$\bar{f}(t+1) = \frac{1}{\bar{f}(t)} \sum_i f_i^2 x_i(t),$$

となり、平均適応度 $\bar{f}(t)$ の時間変化

$$\Delta \bar{f}(t) = \bar{f}(t+1) - \bar{f}(t),$$

は次式で与えられる。

$$\Delta \bar{f}(t) = \frac{1}{\bar{f}(t)} \text{VAR}(f), \quad (11)$$

$$\text{VAR}(f) = \sum_i f_i^2 x_i(t) - \bar{f}(t)^2.$$

すなわち、平均適応度の世代ごとの増加（右辺は非負である）は適応度の分散 $\text{VAR}(f)$ と平均適応度の比に比例する。

3. 突然変異と交叉の表現

突然変異と交叉は代表的な遺伝的操作であるが、この2つの操作は Walsh 変換により非常に見通しよく記述できる。ここでは、まず最初に突然変異の Walsh 変換を用いた解析の結果を示し、次に交叉について述べる。交叉では Vose らによって得られた定式化をレビューし¹²⁾、そのあと次章で、Walsh 変換による我々の方法について説明する。Vose らは突然変異と交叉を一連の操作としてまとめて解析した。しかし我々は Vose らの方法と異なり、突然変異と交叉を独立した演算として別々に取り扱う。

3.1 突然変異の Walsh 係数による表現

最初に、突然変異の Walsh 係数を用いた表現について述べる⁷⁾。突然変異による相対頻度 $x_i(t)$ の世代ごとの変化は、突然変異行列 M を用いて

$$x_i(t+1) = \sum_{j=0}^{n-1} M_{ij} x_j(t), \quad (12)$$

と表すことができる。ここで M_{ij} は遺伝子型 B_j から遺伝子型 B_i への1世代あたりの変異の確率であり、ビットあたりの突然変異率 p とビット列 i と j の間の Hamming 距離 $d(i, j)$ を用いて次のように表される。

$$M_{ij} = (1-p)^{\ell-d(i,j)} p^{d(i,j)}. \quad (13)$$

差分連立方程式 (12) の解は形式的に

$$x(t) = M^t x(0), \quad (14)$$

と表されるが、以下ではその解析的表現を求めてみる。そのためには、まず行列 M の固有値と固有ベクトルを知る必要がある。後で示すように、この行列は n 個の独立な固有ベクトル

$$u_0, u_1, \dots, u_{n-1}$$

を持ち、対応する固有値を

$$\Lambda_0, \Lambda_1, \dots, \Lambda_{n-1}$$

とする。

突然変異行列 (13) は、Kronecker 積を使って

$$M = \overbrace{M_1 \otimes M_1 \otimes \dots \otimes M_1}^{\ell},$$

$$M_1 = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}. \quad (15)$$

と表されることに注意する⁷⁾。ここで M_1 は $\ell=1$ の突然変異行列である。式 (15) から突然変異行列の固有値、固有ベクトルを簡単に求めることができる。そのため、まず M_1 の固有値と固有ベクトルを求める。

$$\lambda_0 = 1, \quad u_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

$$\lambda_1 = 1-2p, \quad u_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \quad (16)$$

これらの2つの固有ベクトルをまとめて

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

と表すことにする。行列 M の Kronecker 積表現から予想できるように、その固有ベクトルもベクトル u_0 と u_1 の Kronecker 積として表現される。結果だけ示すと、固有ベクトルの順番を適当に選択することにより、 $i = \langle i_\ell, i_{\ell-1}, \dots, i_1 \rangle$ から

$$u_i = \overbrace{u_{i_\ell} \otimes u_{i_{\ell-1}} \otimes \dots \otimes u_{i_1}}^{\ell},$$

となる。固有ベクトルの第 j 成分 $u_i(j)$ を明示的に示すには行列 H_1 を用いた方が便利である。

$$u_i(j) = \prod_{k=1}^{\ell} H_{j_k, i_k} \quad (17)$$

ここで

$$H_{j_k, i_k} = (-1)^{i_k \cdot j_k}$$

に注意し、式 (2) と比較すれば

$$u_i = w_i$$

となることが分かる。すなわち突然変異行列 M の固有ベクトルは Walsh 関数であった。

u_i に対応する突然変異行列 M の固有値 Λ_i は

$$\Lambda_i = \prod_{k=1}^{\ell} \lambda_{i_k} = (1-2p)^{|i|} \quad (18)$$

となる。したがって M の最大（絶対値も）固有値は1であり、その固有ベクトルは

$$w_0 = (1, 1, \dots, 1)^T$$

である。

式 (6) と w_j が固有値 $(1-2p)^{|j|}$ を持つ M の固有ベクトルであることから、差分方程式 (12) の解析

的解は

$$\mathbf{x}(t) = \frac{1}{n} \sum_{j=0}^{n-1} (1-2p)^{|j|t} \tilde{x}_j(0) \mathbf{w}_j. \quad (19)$$

で与えられる。\$t\$ が十分大きいと Walsh 関数のうち \$|j|=0\$ の成分のみが残ることから、また規格化条件 \$\tilde{x}_0(0)=1\$ と \$\mathbf{w}_0=1\$ から、相対頻度 \$\mathbf{x}(t)\$ は \$t \to \infty\$ で \$1/n\$ となる。すなわち、突然変異だけ存在する場合は、十分時間が経過すると一様分布 \$(1/n, \dots, 1/n)^T\$ に近づくことを表している。

3.2 Vose モデル

先に述べたように世代的 GA では、遺伝的操作の効果は差分方程式によって記述される。Vose らは、突然変異と交叉の効果をもとめて定式化した¹²⁾。彼らは、GA で最も多く採用されている一点交叉について、進化方程式の具体的な形を導いた。我々の結果との比較のために、ここでその方程式について述べておく。

突然変異と交叉による相対頻度の世代 \$t\$ から世代 \$t+1\$ への変化を考える。この変化を記述する進化方程式として

$$x_k(t+1) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} r_{i,j}(k) x_i(t) x_j(t), \quad (20)$$

を採用する。ここで \$r_{i,j}(k)\$ は突然変異と交叉による遺伝子型 \$B_i\$ と \$B_j\$ から遺伝子型 \$B_k\$ への遷移の確率を表す。遺伝的操作を突然変異、交叉の順に行っても、その逆の順に行っても結論は変わらない。Vose らはまず

$$r_{i,j}(k) = r_{i \oplus k, j \oplus k}(0), \quad (21)$$

を示したうえで、一点交叉について \$r_{i,j}(0)\$ の具体的な形を導いた。ここで \$i \oplus k\$ はビット列 \$i\$ と \$k\$ について各ビットごとに排他論理和をとることを意味する。

結果は、突然変異率 \$p\$ と交叉率 \$\chi\$ を用いて

$$r_{i,j}(0) = \frac{(1-p)^\ell}{2} \times \left\{ \eta^{|i|} \left(1 - \chi + \frac{\chi}{\ell-1} \sum_{k=1}^{\ell-1} \eta^{-\Delta_{i,j,k}} \right) + \eta^{|j|} \left(1 - \chi + \frac{\chi}{\ell-1} \sum_{k=1}^{\ell-1} \eta^{\Delta_{i,j,k}} \right) \right\} \quad (22)$$

と表すことができる。ここで \$\eta = p/(1-p)\$ であり、

$$\Delta_{i,j,k} = |(2^k - 1) \odot i| - |(2^k - 1) \odot j|,$$

と定義する。式中の \$\odot\$ はビットごとの AND を意味するものとする。

4. 交叉の Kronecker 積による分解

ここでは交叉による進化方程式を示し、Walsh 変換を用いた解法について述べる。交叉としては、一点交叉と一様交叉を取り上げることにする。

4.1 交叉行列の構成

交叉による進化を形式的に

$$\mathbf{x}(t+1) = C[\mathbf{x}(t), \mathbf{x}(t)], \quad (23)$$

と表す。より具体的に表せば、\$n \times n^2\$ 交叉行列 \$C\$ を用いて

$$x_k(t+1) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} C_{k,ij} x_i(t) x_j(t). \quad (24)$$

となる。

突然変異のときと同様に、交叉の効果も Kronecker 積を使って表現することを試みる。いま親世代から第 1 親として \$B_i\$、第 2 親として \$B_j\$ を選択し、交叉の結果、子世代のビット列 \$B_k, B_{k'}\$ が生まれるものとする。一般に交叉には様々なパターンがあるが、ここでは交叉マスクとよばれる長さ \$\ell\$ のビット列 \$cm\$ を用いて交叉パターンを表すことにする。たとえば \$\ell=5\$ の場合、\$cm = \langle 1, 1, 0, 0, 1 \rangle\$ とすると、第 1 子 \$B_k\$ は \$cm\$ のビット 0 の位置のビットを第 1 親 \$B_i\$ から、ビット 1 の位置のビットを第 2 親 \$B_j\$ から遺伝するものとする。逆に第 2 子 \$B_{k'}\$ はその残りのビットを二人の親から遺伝するものとする。この例では、\$k = \langle j_5, j_4, i_3, i_2, j_1 \rangle\$、\$k' = \langle i_5, i_4, j_3, j_2, i_1 \rangle\$ となる。

交叉による進化方程式 (24) をすこし書き直し、

$$x_k(t+1) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (C_{k,ij}^{(1)} + C_{k,ij}^{(2)}) x_i(t) x_j(t)$$

とする。ここで交叉行列 \$C_{k,ij}^{(1)}\$ は第 1 子からの寄与分、\$C_{k,ij}^{(2)}\$ は第 2 子からの寄与分である。以下ではこの 2 項を別々に求め、最後に足し合わせる。

$$C_{k,ij} = C_{k,ij}^{(1)} + C_{k,ij}^{(2)}.$$

これから示すように、交叉を表すすべての行列は、2 つの \$2 \times 4\$ 行列 \$F\$ と \$G\$ を用いて Kronecker 積により表現することができる。

$$F = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad G = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}. \quad (25)$$

まず一番単純な場合、\$\ell=1\$ を考える。集団の要素としては \$B_0 = \langle 0 \rangle\$ と \$B_1 = \langle 1 \rangle\$ しかない。交叉マスク \$cm = \langle 0 \rangle\$ のとき、\$2 \times 4\$ 行列 \$C_{k,ij}\$ を次のように

表す .

$$\begin{matrix}
 k(i, j) & (0, 0) & (0, 1) & (1, 0) & (1, 1) \\
 0 & \left(\begin{matrix} C_{0,00} & C_{0,01} & C_{0,10} & C_{0,11} \end{matrix} \right) \\
 1 & \left(\begin{matrix} C_{1,00} & C_{1,01} & C_{1,10} & C_{1,11} \end{matrix} \right)
 \end{matrix}$$

$C_{k,ij}^{(1)}$ は

$$C^{(1)} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \frac{1}{2} F$$

となり, $C_{k,ij}^{(2)}$ は

$$C^{(2)} = \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} = \frac{1}{2} G$$

となる . この計算では , 交叉マスクを $cm = \langle 0 \rangle$ と選んだが , $cm = \langle 1 \rangle$ としても行列 F と G の役割が入れ替わるだけで , 本質的な結果は変わらないことに注意してほしい .

次に $\ell = 2$ の場合を考えてみる . 交叉行列を Kronecker 積を使って表現するためには , 行列 $C_{k,ij}$ の足 $\{i, j\}$ の組合せの順番に注意しなければならない . $\ell = 1$ の場合は , $\{i, j\}$ を $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$ の順番で並べたことに注意して , $\ell = 2$ の場合は

$$\begin{aligned}
 & (00, 00), (00, 01), (01, 00), (01, 01), \\
 & (00, 10), (00, 11), (01, 10), (01, 11), \\
 & (10, 00), (10, 01), (11, 00), (11, 01), \\
 & (10, 10), (10, 11), (11, 10), (11, 11)
 \end{aligned}$$

と配列する . このように並べておくと , たとえば交叉マスク $cm = \langle 0, 1 \rangle$ の場合

$$C^{(1)} = \frac{1}{2} F \otimes G, \quad C^{(2)} = \frac{1}{2} G \otimes F$$

と表すことができる . すなわち , 交叉行列 $C^{(1)}$ について , 交叉マスクのビット 0 には行列 F を , ビット 1 には行列 G を対応させて Kronecker 積を構成していけばよい . また , 交叉行列 $C^{(2)}$ については , その逆の対応を付ける .

最後に一般の ℓ については , 上記の操作を繰り返し交叉行列を構成する . $\{i, j\}$ の組合せの順番については , 2ℓ ビットのビット列 \hat{ij} を考え ,

$$\hat{ij} = \langle i_\ell, j_\ell, i_{\ell-1}, j_{\ell-1}, \dots, i_1, j_1 \rangle$$

のように i と j の各ビットを交互に組み合わせていき , これを非負整数の 2 進数表現と考えると順番を付けていけばよい . 一例として交叉マスク

$$cm = \langle \overbrace{0, 0, \dots, 0}^k, \overbrace{1, 1, \dots, 1}^{\ell-k} \rangle$$

の場合についてみると , 交叉行列は

$$\begin{aligned}
 C &= \frac{1}{2} \overbrace{F \otimes \dots \otimes F}^k \otimes \overbrace{G \otimes \dots \otimes G}^{\ell-k} \\
 &+ \frac{1}{2} \overbrace{G \otimes \dots \otimes G}^k \otimes \overbrace{F \otimes \dots \otimes F}^{\ell-k}
 \end{aligned} \tag{26}$$

で与えられる .

4.2 交叉の Walsh 解析

交叉による進化の方程式 (23) は , 直接解くよりも Walsh 変換した方が容易に解ける . しかも結果の解釈も直感的に分かりやすい . まず式 (23) の右辺にある 2 つの $x(t)$ をそれぞれ w_i と w_j で置き換える . するとその結果は付録に示すように

$$C[w_i, w_j] = nc_{ij} w_{i \oplus j}, \tag{27}$$

となる . 係数 n は c_{ij} の形を簡単にするため , あらかじめ抜き出しておく . 交叉係数 c_{ij} は交叉の種類により異なる形をとる . 先の例では式 (26) で与えられる交叉の場合

$$\begin{aligned}
 c_{ij} &= \frac{1}{2} \prod_{m=1}^k \delta(j_m) \prod_{m=k+1}^{\ell} \delta(i_m) \\
 &+ \frac{1}{2} \prod_{m=1}^k \delta(i_m) \prod_{m=k+1}^{\ell} \delta(j_m)
 \end{aligned} \tag{28}$$

となる .

次にこの式の持つ意味を考えてみたい . 式 (28) から , 任意のビット位置 m において , $i_m = j_m = 1$ となる場合は $c_{ij} = 0$ となる . したがって , 交叉係数 c_{ij} の大部分は 0 になってしまう . このことが , Walsh 変換により交叉の解析が容易になる理由の 1 つになっている . たとえば , $i = j$ の場合 , $c_{ii} \neq 0$ となるのは $i = 0$ のときだけである .

$$c_{00} = 1, \quad c_{ii} = 0 \quad (i \neq 0).$$

交叉による進化方程式 (23) に式 (6) を代入すると , Walsh 係数 $\tilde{x}_k(t)$ についての進化方程式が得られる .

$$\begin{aligned}
 \tilde{x}_k(t+1) &= \sum_{k=i \oplus j} \sum c_{ij} \tilde{x}_i(t) \tilde{x}_j(t) \\
 &= \sum_{i=0}^{n-1} c_{ii} \tilde{x}_i(t) \tilde{x}_{i \oplus k}(t).
 \end{aligned} \tag{29}$$

この式からいくつかの簡単な関係が導かれる . たとえば , $k = 0$ のとき

$$\tilde{x}_0(t+1) = \sum_{i=0}^{n-1} c_{ii} \tilde{x}_i(t) \tilde{x}_i(t) = \tilde{x}_0(t)^2.$$

しかも , 初期条件として規格化条件を満足していれば $\tilde{x}_0(0) = 1$ となるから , 規格化条件を任意の t において満足する .

$$\tilde{x}_0(t) = 1. \quad (30)$$

1 次の Walsh 係数についても重要な関係が存在する．進化方程式 (29) から，第 m ビットだけが 1 となる Walsh 係数 $\tilde{x}_i = \tilde{x}^{(1)}[m]$ は交叉によって変化しないことが容易に証明できる．

$$\tilde{x}^{(1)}[m](t+1) = \tilde{x}^{(1)}[m](t). \quad (31)$$

この式は，集団における第 m ビットのビット 1 とビット 0 の比が交叉によって変化しないことを意味している．このことは次の関係に注意すればすぐ分かる．

$$\begin{aligned} \frac{1}{2}(1 + \tilde{x}^{(1)}[m]) &= \frac{1}{2}(\tilde{x}_0 + \tilde{x}^{(1)}[m]) \\ &= \frac{1}{2} \sum_j (1 + (-1)^{j_m}) x_j. \end{aligned}$$

となるが，ビット 0 を抜き出す演算子

$$\delta(j_m) = \frac{1 + (-1)^{j_m}}{2}$$

を使うと

$$\frac{1}{2}(1 + \tilde{x}^{(1)}[m]) = \sum_j \delta(j_m) x_j$$

となる．また，ビット 1 を抜き出す演算子

$$\delta(1 - j_m) = \frac{1 - (-1)^{j_m}}{2}$$

を用いて同様の関係を得ることができる．したがって，

$$p_m^{(0)} = (1 + \tilde{x}^{(1)}[m])/2, \quad p_m^{(1)} = (1 - \tilde{x}^{(1)}[m])/2$$

とすると， $p_m^{(0)}$ と $p_m^{(1)}$ は第 m ビットにおけるビット 0 とビット 1 の相対頻度を表している．

交叉の効果は 2 次以上の Walsh 係数に現れる．2 次の Walsh 係数 $\tilde{x}^{(2)}[m, m']$ については，交叉点 $[k, k+1]$ と Walsh 係数のビット 1 の位置 m, m' の相対関係によって結果が異なる．ビット 1 が両方とも交叉点の片方にある場合 ($m, m' \leq k$ または $m, m' > k$) は，

$$\tilde{x}^{(2)}[m, m'](t+1) = \tilde{x}^{(2)}[m, m'](t),$$

交叉点の両側にある場合 ($m \leq k$ かつ $m' > k$) は，

$$\tilde{x}^{(2)}[m, m'](t+1) = \tilde{x}^{(1)}[m](t) \tilde{x}^{(1)}[m'](t),$$

と 1 次の Walsh 係数の積になる．

2 次の Walsh 係数も 1 次の場合と同様に意味づけることができる． $p_{mm'}(i_m, i_{m'})$ を第 m ビットにおける値が i_m ，第 m' ビットにおける値が $i_{m'}$ である状態の相対頻度を表すものとする． $p_{mm'}(i_m, i_{m'})$ は

$$\sum_{j=0}^{n-1} \frac{1 + (-1)^{j_m + i_m}}{2} \cdot \frac{1 + (-1)^{j_{m'} + i_{m'}}}{2} x_j,$$

と表すことができ，右辺の各項を展開して計算していくと

$$\begin{aligned} p_{mm'}(i_m, i_{m'}) &= \frac{1}{4} \{1 + (-1)^{i_m} \tilde{x}^{(1)}[m] \\ &\quad + (-1)^{i_{m'}} \tilde{x}^{(1)}[m'] + (-1)^{i_m + i_{m'}} \tilde{x}^{(2)}[m, m']\} \end{aligned}$$

を得る．

一般に L 次の Walsh 係数についても 2 次の Walsh 係数と同様の関係が成り立つ．ビット 1 の位置の集合を $S = \{b_1, b_2, \dots, b_L\}$ ，それを交叉点で 2 分してできる 2 つの部分集合を $S_1 = \{b_1, \dots, b_m\}$ ， $S_2 = \{b_{m+1}, \dots, b_L\}$ とすると，交叉による進化方程式 (29) は

$$\tilde{x}^{(L)}[S](t+1) = \tilde{x}^{(m)}[S_1](t) \tilde{x}^{(L-m)}[S_2](t) \quad (32)$$

となる．部分集合 S_1 と S_2 のいずれかが空集合のときは 0 次の Walsh 係数 $\tilde{x}_0 = 1$ を対応させるものとする．

4.3 連鎖不平衡

遺伝学において，交叉と密接な関係を持つ概念に連鎖不平衡がある^{21),22)}．これは 2 つの遺伝子座における各々の遺伝子が互いに相関を持つことを意味する．GA を例にとれば，第 m ビットの値と第 m' ビットの値とに相関があるか否かということであり，もし相関がなければ連鎖平衡，あれば連鎖不平衡の状態にあることになる．このことを定量的に示す量として連鎖不平衡係数 D ²¹⁾ があり，次式で定義される．

$$D[m, m'] \equiv p_{mm'}(0, 0) - p_m^{(0)} p_{m'}^{(0)}.$$

この式は Walsh 係数を使って書き換えることができ，

$$D = \frac{1}{4} \{ \tilde{x}^{(2)}[m, m'] - \tilde{x}^{(1)}[m] \tilde{x}^{(1)}[m'] \}, \quad (33)$$

となる．もし $D = 0$ であれば，2 つのビット位置 m, m' に相関がなく連鎖平衡にあることを意味し，

$$\tilde{x}^{(2)}[m, m'] = \tilde{x}^{(1)}[m] \tilde{x}^{(1)}[m'],$$

が成り立つ．

4.4 一点交叉と一様交叉

ここでは一点交叉と一様交叉を例に取りあげ，交叉行列の具体的な形を与える．その導出法の一例を付録に示す．交叉率 $\chi = 0$ のときの値を $c_{ij}(0)$ ， $\chi = 1$ のときの値を $c_{ij}(1)$ とすると

$$c_{ij} = (1 - \chi)c_{ij}(0) + \chi c_{ij}(1),$$

と表される． $c_{ij}(0)$ は交叉の種類によらず

$$c_{ij}(0) = \frac{\delta(i) + \delta(j)}{2}, \quad (34)$$

となる．交叉の種類による違いは $c_{ij}(1)$ に現れる．
一点交叉では，ビット 1 の位置に結果が依存する．
 $i = 0$ または $j = 0$ の場合，

$$\begin{aligned} c_{00}(1) &= 1, \\ c_{i0}(1) &= c_{0i}(1) \\ &= \frac{1}{2} - \frac{i_{\max} - i_{\min}}{2(\ell - 1)} \quad (i \neq 0), \end{aligned} \quad (35)$$

となる．ここで $i_{\max} \geq i_{\min}$ はビット列 B_i の両端のビット 1 の位置を表す．

$i \neq 0$ かつ $j \neq 0$ の場合は，

$$\begin{aligned} c_{ij}(1) &= \frac{i_{\min} - j_{\max}}{2(\ell - 1)} \quad (i_{\min} > j_{\max}) \\ &= \frac{j_{\min} - i_{\max}}{2(\ell - 1)} \quad (j_{\min} > i_{\max}) \\ &= 0 \quad (\text{otherwise}), \end{aligned} \quad (36)$$

となる．

一樣交叉の場合は，どの位置のビットも同等に寄与し，

$$c_{ij}(1) = \prod_{m=1}^{\ell} \frac{\delta(i_m) + \delta(j_m)}{2}, \quad (37)$$

となる．

式 (29) に c_{ij} の具体的な形を代入していけば，Walsh 係数 \tilde{x}_k についての進化方程式が得られる．いずれの交叉においても，0 次の Walsh 係数については規格化条件式 (30) が，1 次の Walsh 係数についてはその交叉に対する不変性の式 (31) が導かれる．

2 次の Walsh 係数の具体的な進化方程式を $\chi = 1$ の場合について求めてみる．このとき， i と j の同じビット位置の値が両方とも 1 だと $c_{ij}(1) = 0$ になることに注意する．一点交叉については

$$\begin{aligned} \tilde{x}^{(2)}[m, m'](t+1) &= \left(1 - \frac{m' - m}{\ell - 1}\right) \tilde{x}^{(2)}[m, m'](t) \\ &\quad + \frac{m' - m}{\ell - 1} \tilde{x}^{(1)}[m](t) \tilde{x}^{(1)}[m'](t), \end{aligned}$$

となる．ここで $\ell \geq m' > m \geq 1$ であり，右辺の 2 項は負にはならない．

一樣交叉の場合は

$$\begin{aligned} \tilde{x}^{(2)}[m, m'](t+1) &= \frac{1}{2} \tilde{x}^{(2)}[m, m'](t) \\ &\quad + \frac{1}{2} \tilde{x}^{(1)}[m](t) \tilde{x}^{(1)}[m'](t), \end{aligned}$$

と非常に簡単な関係が得られる．

連鎖不平衡係数 D については交叉を施した後の値を D' とすると，一点交叉では

$$D'[m, m'] = \left(1 - \frac{m' - m}{\ell - 1}\right) D[m, m'],$$

となり，一樣交叉では

$$D'[m, m'] = \frac{1}{2} D[m, m'],$$

となる．

一般的に高次の Walsh 係数については，進化方程式は複雑になっていく．しかし，交叉を繰り返していくと $t \rightarrow \infty$ において，すべての Walsh 係数について次の関係が成立する．

$$\tilde{x}_k[b_1, b_2, \dots, b_k] = \prod_{m=1}^k \tilde{x}_1[b_m]. \quad (38)$$

この式によれば， $t \rightarrow \infty$ で集団が定常状態に達すれば，その分布は連鎖平衡状態になる．すなわち，ある遺伝子型の相対頻度は各遺伝子座におけるその遺伝子型を構成する対立遺伝子の相対頻度の積になる．別の言葉でいえば，各遺伝子座における対立遺伝子の分布は遺伝子座ごとに独立になる．この結果は，すでに Geiringer の定理として知られていたものである²⁰⁾．

4.5 交叉の役割

これまで主に交叉の数学的表現についてみてきた．それでは交叉は GA においていかなる役割を果たしているのだろうか．その答えの 1 つは，交叉は連鎖不平衡の程度を弱め，各遺伝子(ビット)間の相関を小さくする，ということである．それではビットとビットの間の相関が弱められれば，GA にとってどのような利点が生ずるのであるだろうか．その完全な解答を得ることは難しいが，ここではその答えの 1 つを提出したい．

そのため，次の 2 つの統計量を取り上げる．1 つは集団中の個体と基準解の間の Hamming 距離の平均であり，他の 1 つはその分散である．いま基準解としてすべてのビットが 0 の遺伝子型 $B_0 = \langle 0, 0, \dots, 0 \rangle$ をとることにする．遺伝子型 B_i の B_0 からの Hamming 距離が $|i|$ となることに注意すると，世代 t における平均 Hamming 距離とその分散は

$$\begin{aligned} E\{|i|\} &= \sum_{i=0}^{n-1} |i| x_i(t), \\ V\{|i|\} &= \sum_{i=0}^{n-1} |i|^2 x_i(t) - E\{|i|\}^2 \end{aligned}$$

で与えられる．これらの統計量を統一的に求めるため，次の母関数

$$G(s, t) = \sum_{i=0}^{n-1} \exp\{i|s\} x_i(t)$$

を定義する．この式の右辺に $x_i(t)$ の Walsh 係数による展開式 (4) を代入すると

$$G(s, t) = \frac{1}{n} \sum_{i,j} \exp\{i|s\} W_{ij} \tilde{x}_j(t)$$

を得る．そこでまず $\sum_i \exp\{i|s\} W_{ij}$ を計算する．

$$\begin{aligned} \sum_{i=0}^{n-1} \exp\{i|s\} W_{ij} &= \sum_i \prod_{k=1}^{\ell} e^{i_k s} (-1)^{i_k j_k} \\ &= (1 + e^s)^{\ell - |j|} (1 - e^s)^{|j|}. \end{aligned}$$

この結果から母関数 G は次式で与えられる．

$$G = \sum_{j=0}^{n-1} \left(\frac{1 + e^s}{2} \right)^{\ell - |j|} \left(\frac{1 - e^s}{2} \right)^{|j|} \tilde{x}_j(t). \quad (39)$$

平均 Hamming 距離 $E\{|i|\}$ はこの母関数を用いて計算できる．

$$E\{|i|\} = \left. \frac{\partial}{\partial s} G(s, t) \right|_{s=0}$$

から， $s = 0$ のとき $1 - e^s = 0$ となることに注意して次式を得る．

$$E\{|i|\} = \sum_{m=0}^{\ell} p_m^{(1)} = \sum_{m=0}^{\ell} \frac{1 - \tilde{x}^{(1)}[m](t)}{2}, \quad (40)$$

$p_m^{(1)}$ は第 m ビットが 1 である確率であるから，これは十分予想できる結果である．1 次以下の Walsh 係数を含む項だけが残ることに注意してほしい．

Hamming 距離の分散も同様に計算することができる．

$$V\{|i|\} = \left. \frac{\partial^2}{\partial s^2} G(s, t) \right|_{s=0} - E\{|i|\}^2.$$

この場合には，結果は各ビットからの寄与を含む項 V_1 と，2 つのビットにまたがる項 V_2 の和の形に表され，

$$V\{|i|\} = V_1 + V_2, \quad (41)$$

ここで

$$\begin{aligned} V_1 &= \sum_{m=0}^{\ell} p_m^{(1)} (1 - p_m^{(1)}), \\ V_2 &= \frac{1}{2} \sum_{m < m'} \{ \tilde{x}^{(2)}[m, m'] - \tilde{x}^{(1)}[m] \tilde{x}^{(1)}[m'] \} \\ &= 2 \sum_{m < m'} D[m, m'], \end{aligned}$$

である．特に V_2 は，すべての連鎖不平衡係数からの寄与の和となっている．

V_1 は 1 次の Walsh 係数の関数であるから，交叉に

よる直接の影響はない．それに対して V_2 は交叉により大きく影響を受ける．したがって，Hamming 距離の分布には交叉の効果がみられると予想される．

交叉の効果をより詳しく調べるため，平均 Hamming 距離 $E\{|i|\}$ の世代あたりの変化率

$$v(t) \equiv E\{|i|\}(t+1) - E\{|i|\}(t), \quad (42)$$

を定義する．一般に $v(t)$ は負の量で， $-v(t)$ は定常状態へ近づく速度を表しており，GA における進化速度の指標の 1 つと考えることができる．

5. 数値 GA 実験

交叉の役割について調べるため，積型適応度関数を用いて数値実験を行った．積型適応度関数は次式で定義される．

$$f_i = \prod_{k=1}^{\ell} a^{i_k} = a^{|i|} \quad (0 < a \leq 1).$$

選択における淘汰の強さはパラメータ a で表現され， $B_0 = \langle 0, 0, \dots, 0 \rangle$ が最適解となる．この適応度関数については，集団の大きさが無限大の決定論的モデルにおいて解析的解が得られている⁸⁾．またこの積型適応度関数を用いたモデルでは，初期条件として連鎖平衡状態から出発した場合，各世代において連鎖平衡を保つ，ということも分かっている．したがってモデルの範囲では，交叉は集団に対して効果を及ぼさないことが予想される．

数値実験は， $\ell = 8$ ， $a = 0.8$ ， $p = 0.01$ として行った．交叉係数は $\chi = 0$ と 1 の 2 通りの場合について比較し，交叉の効果を調べた．集団のサイズについては $N = 10$ ， $N = 100$ ， $N = 1000$ の 3 通りの場合について計算した．乱数を変えながら同じ計算を 100 回繰り返し，各種の計算量についてその平均値を求めた．初期条件として $t = 0$ においてすべての個体を $\langle 1, 1, \dots, 1 \rangle$ とおいた．これは最適解 $\langle 0, 0, \dots, 0 \rangle$ からは最も遠いビット列である．

図 1 に $N = 10$ ，100，1000 における平均 Hamming 距離 $E\{|i|\}$ の結果を世代 t の関数として示した．実線は交叉がある場合 ($\chi = 1$)，点線は交叉がない場合 ($\chi = 0$) の結果を表している． $E\{|i|\} = 0$ はすべての個体が最適解にたどりついた状態を表すが，実際の計算では世代 t が十分大きくなると有限の $E\{|i|\}$ を持つ定常状態に到達する．これは突然変異率 p が有限の値を持つためであり，定常状態は B_0 以外の遺伝子型を含んだ分布となっている．この定常状態に到着するのに要する世代数は $N = 1000$ が最も少なく， $N = 10$ が最も多い．またいずれの場合も，交

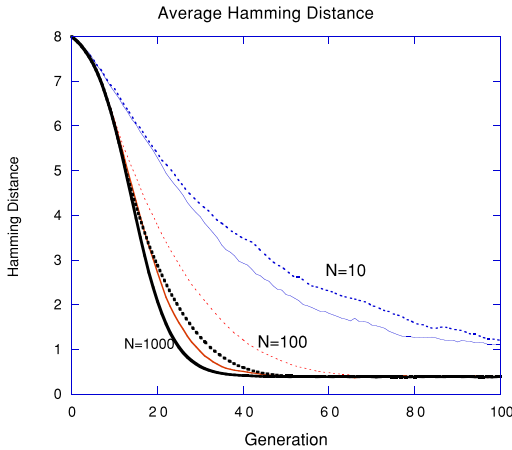


図 1 平均 Hamming 距離, $\ell = 8, a = 0.8, p = 0.01$ とし, 集団の大きさは $N = 10, 100, 1000$ の 3 通りについて計算. 100 回の計算の平均値を交叉ありの場合 ($\chi = 1$) は実線で, 交叉なしの場合 ($\chi = 0$) は点線で示す

Fig. 1 Average Hamming distance with $\ell = 8, a = 0.8, p = 0.01$ and $N = 10, 100, 1000$. The results are averaged over 100 runs. The solid lines stand for the data with crossover ($\chi = 1$), and the dotted lines for without crossover ($\chi = 0$).

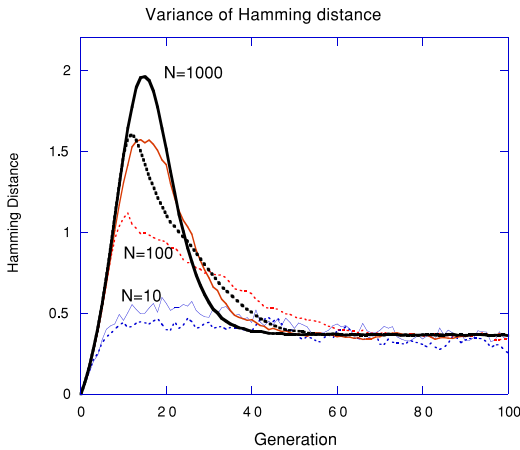


図 2 Hamming 距離の分散. 計算に使用したパラメータは図 1 と同じ

Fig. 2 Variance of Hamming distance. The calculation is the same as in Fig. 1.

又があるほうがない場合に比べて早く収束する.

一方, Hamming 距離の分散 $V\{i\}$ について調べたものが図 2 である. 初期状態では $V = 0$ だが次第に分散は大きくなり, いずれも $t = 15$ の近辺で最大になった後, 定常状態の値に近づいていく. 分散のピークの値は N の多少や交叉の有無により大きな差がでる. N が多いほどピークは高くなり, 交叉があるほうがピークが高い. この結果は, 図 1 で示した進化の速さと同じ傾向を持つ.

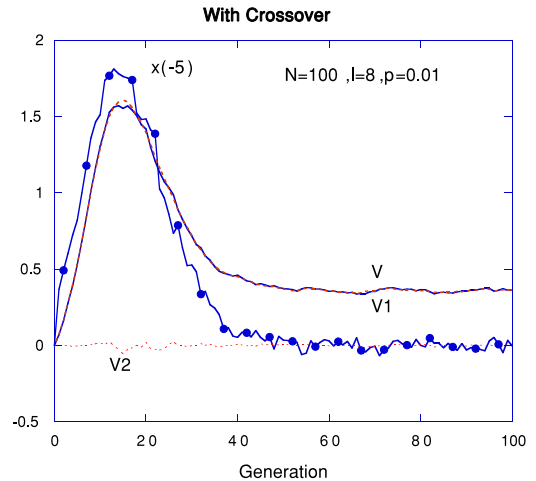


図 3 交叉ありの場合の平均 Hamming 距離の変化率 $v(t)$ と分散 $V\{i\}$ の関係. 黒丸のデータは $-5 \times v(t)$ を示す

Fig. 3 The relation between $v(t)$ and $V\{i\}$ of Hamming distance with crossover. The data of black circles stand for $-5 \times v(t)$.

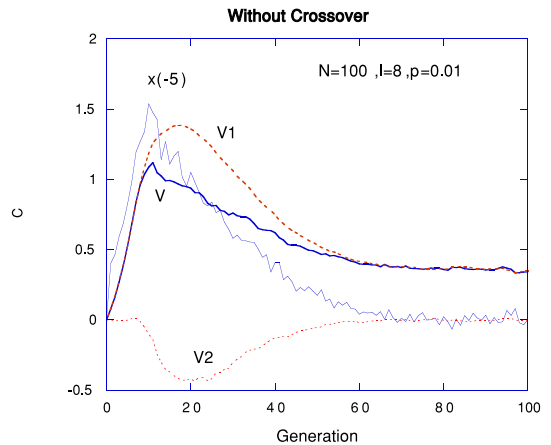


図 4 交叉なしの場合の平均 Hamming 距離の変化率 $v(t)$ と分散 $V\{i\}$ の関係

Fig. 4 The relation between $v(t)$ and $V\{i\}$ of Hamming distance without crossover.

図 3 と図 4 に集団の大きさが $N = 100$ の場合の平均 Hamming 距離の変化率 (速度) $v(t)$ と分散 $V\{i\}$ の関係を世代の関数として表した. 比較しやすいよう図には速度そのものではなく $v(t) \times (-5)$ を表示した. 図 3 が交叉を含む場合, 図 4 が含まない場合の結果である. また両方の図で, 式 (41) における分散 $V\{i\}$ への 2 種類の寄与, V_1 と V_2 , を点線で示した.

2 つの図の大きな違いは, 分散 $V\{i\}$ への連鎖不平衡からの寄与 V_2 にみられる. 交叉がある場合 (図 3) は, 連鎖不平衡からの寄与 V_2 がほぼ 0 で $V\{i\} \approx V_1$ となっている. そのため, 図 3 では $V\{i\}$ と V_1 を表

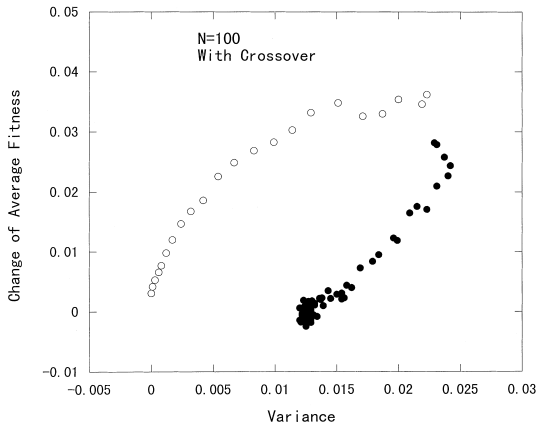


図5 $\text{VAR}(f)$ と $\Delta\bar{f}(t)$ との関係を示す散布図. $\chi = 1, N = 100$ とした. 白丸は $0 \leq t \leq 21$, 黒丸は $22 \leq t \leq 50$ における値

Fig. 5 Scatter diagram of $\Delta\bar{f}(t)$ versus $\text{VAR}(f)$ with $\chi = 1$ and $N = 100$. White circles are for $0 \leq t \leq 21$, and black circles for $22 \leq t \leq 50$.

す線が重なってほとんど区別できない. 交叉がない場合(図4)は, 進化の速度が大きい領域($t = 10 \sim 60$)で V_2 が負の有限な値をとり, 結果としてこの領域では分散 $V\{i\}$ が小さくなってしまふ. そして, このことと連動するように $v(t)$ も交叉のある場合より小さくなる. 図3, 4ともに $v(t)$ と $V\{i\}$ が大きく変化する領域では, これら2つの量はよく相関している.

先に式(11)で示したように, $p = 0, \chi = 0$ の場合, 適応度の世代ごとの変化 $\Delta\bar{f}(t)$ は適応度の分散と平均適応度の比 $\text{VAR}(f)/\bar{f}$ に比例する. 一般に p が有限な場合, もはやこの関係は成立しない. しかし, p があまり大きくなければ, 近似的に比例関係が成り立つことが予想される. 図5は, 図3と同じ計算の結果を用いて, 交叉がある場合の $\Delta\bar{f}(t)$ と $\text{VAR}(f)$ の間の関係を散布図に表したものである. 白丸は $t = 0$ から $\Delta\bar{f}(t)$ が最大値をとる $t = 21$ までの値を, 黒丸はそれ以降 $t = 50$ までの値を表している. この結果によると, $t = 21$ あたりを境にして $\Delta\bar{f}(t)$ と $\text{VAR}(f)$ の関係は異なっている. これは, \bar{f} の変化を無視したことで, 集団の構成が質的に変化したことによると予想される. しかし, いずれにしても2つの量の間には強い正の相関がみられる. そして, この相関関係の反映として $v(t)$ と $V\{i\}$ の間にも強い相関が観測されたのではないと思われる.

6. 考察とまとめ

交叉の役割の1つは, 連鎖不平衡を減少させることである. そしてこのことが結果的に集団の広がりや影

響を与える. そしてさらに, このことによって集団の進化速度も変化する. 逆に, 集団が連鎖平衡状態にあれば集団に対する交叉の効果はない. したがって, GAでも交叉という操作を有効に適用するためには, 解の集団がどのような連鎖不平衡状態にあるか, ということに注意を払う必要がある. 特に連鎖不平衡係数 D の和が全体として負の大きい量になる場合は, 交叉によりすべての D が0に近づき, 進化の速度が加速される可能性が高い. また, D の和がそれほど大きくならない場合は, 交叉率 χ を小さくにとって計算における交叉の回数を減少させても, 実際上はほとんど問題ないと思われる.

またいくつかの特定の連鎖不平衡係数 $D[m, m']$ のみが大きな負の量になることもあるだろう. その場合は, そのような特定のビット位置 m, m' の連鎖を効率良く切断する交叉の種類を選択すればよい(もし可能ならば). そしてそのような情報が得られない場合には, 一様交叉のようにすべてのビット間の連鎖を平等に切断する交叉を選択する方がよいであろう.

集団中に連鎖不平衡を生じる原因としては次の2つが考えられる.

- エピスタシスのある適応度関数による選択
- 有限の個体数による揺らぎの効果

エピスタシスは非相加的相互作用と訳され, 2つ以上の遺伝子座が互いに非線型に関係しあって適応度を決定することをいう. たとえば, ある遺伝子座の対立遺伝子 a と A の適応度が, 他の遺伝子座の対立遺伝子 b と B のどちらが存在するかによって異なるような場合である. したがって, GAで用いられているのはほとんどエピスタシスを持った適応度関数である. ただ, GAにおいてどのような関数が負の連鎖不平衡をもたらすか, ということは今後の研究に待たなければならない.

論文中で用いた積型適応度関数はその意味では特殊である. この関数はエピスタシスを持たないことが知られている. 実際, 個体数を無限大にした決定論の枠組みにおける計算では, 連鎖不平衡は生じない. しかしすでにみたように, 進化の途中においてかなりの連鎖不平衡が生じている. したがって, この連鎖不平衡の原因は個体数の有限性からくる確率的揺らぎの効果によるものと考えられる. しかし, 不平衡が具体的に確率的揺らぎのどのような機構で生じるか, ということは課題として残されている.

GAにおける代表的な適応度関数に One-Max 関数がある. その適応度は $f_i = \sum i_k = |i|$ と定義され, 関数形をみればぎりぎりエピスタシスを持たない. しかし,

選択による進化方程式 (8) にこの適応度関数を代入して調べればすぐ分かるが、選択の過程で負の連鎖不平衡を生じる。したがってこの場合、交叉は進化にとって有利に働く。この結果は意外であるが、One-Max 関数における交叉の役割については、別の機会に報告したい。

本論文では、交叉行列を Kronecker 積により分解し、それを Walsh 変換した式を導いた。交叉行列は Walsh 変換により非常に簡単かつ直感的に理解しやすい形になる。交叉の役割の 1 つである連鎖不平衡の解消に注目し、交叉による連鎖不平衡の変化を定式化した。連鎖不平衡と個体の Hamming 距離の分散 $V\{|i|\}$ の関係を示し、交叉と分散 $V\{|i|\}$ の関連を明らかにした。具体的に積型適応度関数を用いて数値実験を行い、分散 $V\{|i|\}$ と平均 Hamming 距離の世代あたりの変化率 $v(t)$ が高い相関を持つことを示した。交叉がない場合、連鎖不平衡係数の総和は負の値をとり、結果として分散 $V\{|i|\}$ が小さくなり、進化に遅れを生じた。このことから積型適応度の GA では、交叉が連鎖不平衡を減少させることにより、進化の遅れを回復させる働きを持つことが分かった。

参 考 文 献

- 1) Holland, J.H.: *Adaptation in Natural and Artificial Systems*, p.211, MIT Press, Massachusetts (1992).
- 2) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, p.412, Addison Wesley, Massachusetts (1989).
- 3) 伊庭斉志: 遺伝的アルゴリズムの基礎, p.254, オーム社, 東京 (1994).
- 4) Altenberg, L.: The Schema Theorem and Price's Theorem, *Foundations of Genetic Algorithms 3*, Whitley, L.D. and Vose, M.D. (Eds.), pp.23-49 (1995).
- 5) Nix, A.E. and Vose, M.D.: Modelling Genetic Algorithms with Markov Chains, *Annals of Mathematical and Artificial Intelligence*, Vol.5, pp.79-88 (1992).
- 6) Suzuki, J.: A Markov Chain Analysis on a Genetic Algorithm, *Proc. 5th International Conference on Genetic Algorithms*, Forrest, S. (Ed.), pp.146-153, Morgan Kaufmann (1993).
- 7) Furutani, H.: Application of Eigen's Evolution Model to Infinite Population Genetic Algorithms with Selection and Mutation, *Complex Systems*, Vol.10, pp.345-366 (1996).
- 8) Furutani, H.: Analytical Solutions for Infinite Population Genetic Algorithms on Multiplicative Landscape, *Proc. Genetic and Evolutionary Computation Conference, GECCO-99*, Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, M.H., Jakiela, M. and Smith, R.E. (Eds.), Vol.1, pp.204-211, Morgan Kaufmann (1999).
- 9) Eigen, M.: Selforganization of Matter and the Evolution of Biological Macromolecules, *Die Naturwissenschaften*, Vol.58, pp.465-523 (1971).
- 10) Nowak, M. and Schuster, P.: Error Thresholds of Replication in Finite Populations: Mutation Frequencies and the Onset of Muller's Ratchet, *Journal of Theoretical Biology*, Vol.137, pp.375-395 (1989).
- 11) Wiehe, T., Baake, E. and Schuster, P.: Error Propagation in Reproduction of Diploid Organisms: A Case Study on Single Peaked Landscape, *Journal of Theoretical Biology*, Vol.177, pp.1-15 (1995).
- 12) Vose, M.D. and Liepins, G.E.: Punctuated Equilibria in Genetic search, *Complex Systems*, Vol.5, pp.31-44 (1991).
- 13) Vose, M.D.: *The Simple Genetic Algorithms*, p.251, MIT Press, Massachusetts (1999).
- 14) Munetomo, M. and Goldberg, D.E.: Linkage Identification by Non-monotonicity Detection for Overlapping Functions, *Evolutionary Computation*, Vol.7, pp.377-398 (1999).
- 15) Rumschitzki, D.S.: Spectral Properties of Eigen Evolution Matrices, *Journal of Mathematical Biology*, Vol.24, pp.667-680 (1987).
- 16) 佐武一郎: 線形代数学, p.324, 裳華房, 東京 (1974).
- 17) Goldberg, D.E.: Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction, *Complex Systems*, Vol.3, pp.129-152 (1989).
- 18) 遠藤 靖: ウォルシュ解析, p.216, 東京電機大学出版局, 東京 (1993).
- 19) Goldberg, D.E. and Deb, K.: A Comparative Analysis of Selection Schemes Used in Genetic Algorithms, *Foundations of Genetic Algorithms*, Rawlins, G.J.E. (Ed.), pp.69-93 (1991).
- 20) Geiringer, H.: On the Probability Theory of Linkage in Mendelian Heredity, *Annals of Mathematical Statistics*, Vol.15, pp.25-57 (1944).
- 21) Maynard Smith, J.: *Evolutionary Genetics*, p.330, Oxford University Press, Oxford (1998).
- 22) Weir, B.S.: *Genetic Data Analysis II*, p.445, Sinauer Associates, Massachusetts (1996).

付 録

A.1 交叉行列の Walsh 変換

Kronecker 積を用いて構成した交叉行列は、このままでは計算上の利用価値は少ない。ここでは、それを Walsh 変換した表式を導く。進化方程式 (24) を再び書き直し

$$y_k = \sum_{i'=0}^{n-1} \sum_{j'=0}^{n-1} C_{k,i'j'} W_{i'}(i) W_{j'}(j),$$

右辺における交叉行列 C の具体的な形を与え、和を計算していく。行列 C は次式で表され、

$$C = \overbrace{\hat{F} \otimes \hat{F} \otimes \dots \otimes \hat{F}}^{\ell},$$

C の具体的な形に応じて $\hat{F} = F$ と見なしたり、 $\hat{F} = G$ と見なしたりするものとする。

$$\sum_{i'=0}^{n-1} = \sum_{i'_\ell=0}^1 \sum_{i'_{\ell-1}=0}^1 \dots \sum_{i'_1=0}^1$$

と

$$\begin{aligned} & C_{k,i'j'} W_{i'}(i) W_{j'}(j) \\ &= \prod_{m=1}^{\ell} \hat{F}_{k_m,i'_m j'_m} (-1)^{i'_m \cdot i_m} (-1)^{j'_m \cdot j_m} \end{aligned}$$

を用いて、 y_k は

$$y_k = \prod_{m=1}^{\ell} \sum_{i'_m=0}^1 \sum_{j'_m=0}^1 \hat{F}_{k_m,i'_m j'_m} (-1)^{i'_m \cdot i_m} (-1)^{j'_m \cdot j_m}$$

から簡単に計算できる。

$$y_{k_m} = \sum_{i'_m=0}^1 \sum_{j'_m=0}^1 \hat{F}_{k_m,i'_m j'_m} (-1)^{i'_m \cdot i_m} (-1)^{j'_m \cdot j_m}$$

とおく。ここで $\hat{F} = F$ とすると

$$\begin{aligned} y_{k_m} &= 2\delta(j_m) (-1)^{k_m \cdot i_m} \\ &= 2\delta(j_m) (-1)^{k_m \cdot (i_m \oplus j_m)}. \end{aligned}$$

$\delta(j_m)$ はデルタ関数であり、 $j_m = 1$ の場合は 0 となることに注意してほしい。 $\hat{F} = G$ の場合は、

$$\begin{aligned} y_{k_m} &= 2\delta(i_m) (-1)^{k_m \cdot j_m} \\ &= 2\delta(i_m) (-1)^{k_m \cdot (i_m \oplus j_m)}, \end{aligned}$$

となり、行列 F と G の違いはデルタ関数にのみ現れる。

$$\begin{aligned} y_k &= \prod_{m=1}^{\ell} y_{k_m} = 2^{\ell} W_{i \oplus j}(k) \prod_{m=1}^{\ell} \hat{\delta}_m \\ &= n W_{i \oplus j}(k) \prod_{m=1}^{\ell} \hat{\delta}_m. \end{aligned}$$

ここで $\hat{\delta}_m$ は $\hat{F} = F$ の場合は $\delta(j_m)$ と、 $\hat{F} = G$ の場合は $\delta(i_m)$ と見なすものとする。このことから式 (27) が導かれる。

$$C[\mathbf{w}_i, \mathbf{w}_j] = n c_{ij} \mathbf{w}_{i \oplus j}.$$

この例では交叉係数 c_{ij} は

$$c_{ij} = \prod_{m=1}^{\ell} \hat{\delta}_m,$$

で与えられる。

A.2 一点交叉と一様交叉

最後に、具体的な交叉の規則から交叉行列の Walsh 変換を求める。まず最初に一点交叉を取り上げる。一点交叉の交叉行列は、

$$\begin{aligned} C &= \frac{1-\chi}{2} (F \otimes F \otimes \dots \otimes F + G \otimes G \otimes \dots \otimes G) \\ &+ \frac{\chi}{2(\ell-1)} (G \otimes F \otimes \dots \otimes F + \dots \\ &+ \overbrace{G \otimes \dots \otimes G}^k \otimes \overbrace{F \otimes \dots \otimes F}^{\ell-k} \\ &+ \overbrace{F \otimes \dots \otimes F}^k \otimes \overbrace{G \otimes \dots \otimes G}^{\ell-k} \\ &+ \dots + F \otimes \dots \otimes F \otimes G), \end{aligned}$$

と表される。

この式から交叉係数 c_{ij} を求めるが、 $\chi = 0$ のときの値を $c_{ij}(0)$ 、 $\chi = 1$ のときの値を $c_{ij}(1)$ とし

$$c_{ij} = (1-\chi)c_{ij}(0) + \chi c_{ij}(1),$$

と 2 項に分けて求める。

交叉行列 C から交叉係数 c_{ij} への変換は

$$F \rightarrow \delta(j_m), \quad G \rightarrow \delta(i_m)$$

と置き換えていけばよい。たとえば

$$\overbrace{F \otimes F \otimes \dots \otimes F}^{\ell} \rightarrow \prod_{m=1}^{\ell} \delta(j_m) = \delta(j)$$

となり、同様に

$$\overbrace{G \otimes G \otimes \dots \otimes G}^{\ell} \rightarrow \delta(i)$$

と置き換える。したがって

$$c_{ij}(0) = \frac{\delta(i) + \delta(j)}{2}$$

となる。

残りの交叉に依存する項 $c_{ij}(1)$ については 3 通り

の場合に分けて考える .

(1) $i = j = 0$

このときすべてのビットは $i_m = 0, j_m = 0$ だから, 行列 F と G はあたかも単位行列のような働きをする . したがって各項を単純に足し合わせて

$$c_{00}(1) = 1$$

を得る .

(2) $i \neq 0, j = 0$ または $i = 0, j \neq 0$

$i \neq 0, j = 0$ の場合を考える . $j = 0$ については, すべてのビットが 0 なので特に考慮する必要はない . 問題は $i \neq 0$ のビット 1 の位置である . まず, 最も簡単な $|i| = 1$ について考える . この場合は, ビット 1 の位置に関係なく

$$c_{i0}(1) = \frac{\ell - 1}{2(\ell - 1)} = \frac{1}{2}$$

となる . $|i| = m \geq 2$ の場合は, ビット 1 の両端の位置 i_{\min} と i_{\max} だけが関係し,

$$\begin{aligned} c_{i0}(1) &= \frac{\{(\ell - 1) - (i_{\max} - i_{\min})\}}{2(\ell - 1)} \\ &= \frac{1}{2} - \frac{i_{\max} - i_{\min}}{2(\ell - 1)} \end{aligned}$$

となる .

(3) $i \neq 0$ かつ $j \neq 0$

この場合は, i と j のビット 1 の両端の位置 $i_{\min}, i_{\max}, j_{\min}, j_{\max}$ に依存し,

$$c_{ij}(1) = \begin{cases} (i_{\min} - j_{\max})/2(\ell - 1) & (i_{\min} > j_{\max}) \\ (j_{\min} - i_{\max})/2(\ell - 1) & (j_{\min} > i_{\max}) \\ 0 & (\text{otherwise}) \end{cases}$$

となる .

次に一様交叉の交叉係数 c_{ij} を求める . 一様交叉の交叉行列は, 次式で表される .

$$\begin{aligned} C &= \frac{1 - \chi}{2} (F \otimes F \otimes \cdots \otimes F + G \otimes G \otimes \cdots \otimes G) \\ &\quad + \chi 2^{-\ell} (F \otimes F \otimes \cdots \otimes F + \cdots + G \otimes G \otimes \cdots \otimes G). \end{aligned}$$

ここで右辺第 3 項以降はすべての F と G の組合せを含み, 全部で 2^ℓ 項ある . したがってこれらの項を整理していくと

$$\begin{aligned} C &= \frac{1 - \chi}{2} (F \otimes F \otimes \cdots \otimes F + G \otimes G \otimes \cdots \otimes G) \\ &\quad + \chi \left(\frac{F + G}{2} \otimes \frac{F + G}{2} \otimes \cdots \otimes \frac{F + G}{2} \right) \end{aligned}$$

を得る .

交叉行列 C から交叉係数 c_{ij} への変換は一点交叉と同様に $F \rightarrow \delta(j_m)$ と $G \rightarrow \delta(i_m)$ の置き換えをすればよく

$$c_{ij} = (1 - \chi) \frac{\delta(i) + \delta(j)}{2} + \chi \prod_{m=1}^{\ell} \frac{\delta(i_m) + \delta(j_m)}{2}$$

を得る .

(平成 12 年 10 月 12 日受付)

(平成 13 年 6 月 19 日採録)



古谷 博史 (正会員)

昭和 26 年生 . 昭和 49 年京都大学理学部卒業 . 昭和 51 年同大学大学院理学研究科物理学第二専攻修士課程修了 . 昭和 54 年同大学院博士課程単位修得退学 . 昭和 56 年高知医科大学助手 . 昭和 63 年同大学助教授 . 医療情報システムの研究開発に従事 . 平成 2 年より京都教育大学教授 . 遺伝子情報システム, 遺伝的アルゴリズム等の研究に従事 . 理学博士 . 医療情報学会, ソフトウェア科学会各会員 .