

# 遺伝的アルゴリズムによる動的環境下での最適化

林 貴宏<sup>†,††</sup> 木村 春彦<sup>††</sup> 白山 政敏<sup>†</sup>

遺伝的アルゴリズム (GA) は、解の表現形式や評価方式に制約がないため広範囲な分野に適用可能であるが、静的環境を前提としたアルゴリズムであるため、適用可能な問題のクラスが限定されるという問題点がある。この問題点を解消するために本論文では動的環境下での最適化を目的とした GA を提案する。提案手法は、(1) 適応度ランドスケープの形状が変化する環境、(2) 表現型に与える各遺伝子座の影響力が時間によって変化する環境、(3) リンケージを構成する遺伝子座の組合せが時間によって変化する環境を取り扱う。本論文では、これらの動的環境に対処するために必要となる探索スケール制御とブロック同定という処理を説明する。また、変動ナップザック問題や変動トラップ関数の最適化問題を用いた 3 種類の実験によって提案手法の有効性を示す。

## Optimization under a Changing Environment by Genetic Algorithms

TAKAHIRO HAYASHI,<sup>†,††</sup> HARUHIKO KIMURA<sup>††</sup>  
and MASATOSHI SIRAYAMA<sup>†</sup>

Genetic algorithms (GAs) can be applied to problems in various fields because there are not restricted in terms of solution expression or evaluation, however, GAs have a serious disadvantage in that they cannot be applied to changing environments. In this paper, a novel type GA for optimization under a dynamic environment is proposed. The method deal with (1) change of fitness landscape shape, (2) change of effect on phenotype at each locus and (3) change of locus-groups which organize linkages. Search scale control and block identification mechanisms, which are for treating these changes, is explained. Effectiveness of the method is confirmed by three simulations using changing knapzack problems and changing trap function optimization problems.

### 1. はじめに

遺伝的アルゴリズム (GA) は、解の表現形式、評価方式に制約がないため広範囲な分野で適用され成果をあげている<sup>1)~3)</sup>。GA は実行中評価関数を固定しておくことから静的環境の問題 (探索空間の形状が変化しない問題) を前提に作られたアルゴリズムと考えられる。しかし、現実世界には動的環境の問題 (探索空間の形状が変化する問題) も多く存在することから、静的環境を前提としていることは応用上大きな制約になると考えられる。そこで、本研究では動的環境を前提とした GA を提案することを目的とする。

これまでも動的環境を扱った研究が報告されている。Yamazaki<sup>4)</sup> は動的環境における適応は、環境に持続的な性質があつて初めて可能になり、環境の持続

的性質そのものが適応の標的になっている。また、そのような性質の例として再現性、連続性、希少性の 3 つをあげている。再現性は過去に表れた適応度ランドスケープの何らかの特徴が再び表れる性質、連続性は適応度ランドスケープの形状変化が 1 時刻前と比べて小さい性質、希少性は適応度ランドスケープの変化が大きく不規則であっても変化が希少である性質である。再現性に対してはメモリベースの GA (たとえば文献 5)~8) など) が有効であると考えられる。連続性に対しては適応度ランドスケープ上の近傍探索を行う手法 (たとえば文献 9)~11) など) が有効であると考えられる。また、希少性に対しては多様性維持あるいは瞬時的に多様性を生成する手法 (たとえば文献 12)~16) など) が有効であると考えられる。

このように動的環境の性質によって問題をクラス分けできれば、各クラスに適した手法を適用することで効果的な探索が期待できる。しかしながら、事前に環境の性質が分からない場合や、複数の性質が組み合わされている場合なども考えられる。そこで本論文では

† 石川工業高等専門学校  
Ishikawa National College of Technology  
†† 金沢大学  
Kanazawa University

実行しながらリアルタイムに動的環境の性質を同定し、状況に応じて適応的に探索方針を変えていくような手法を提案する。

## 2. 動的環境下での GA の探索方針

性質が未知の動的環境下での探索に必要とされる性能として、環境の急激な変化に対処できる能力（大域的な環境追従性）と、連続的な変化に対処できる性能（局所的な環境追従性）が考えられる。大域的な環境追従性と局所的な環境追従性の間には探索スケール（表現型空間における個体の分散）に関してトレードオフがあるためこれらの性能を同時に向上させることには限界がある。そこで、各時点で環境変動がないか監視し、環境変動が発生した場合はその規模を同定し、変動規模に応じて適切に探索スケールを制御していくことを考える。たとえば、大きな環境変動を検知した場合は探索スケールが大きくなるように制御し、小さな環境変動を検知した場合は探索スケールが小さくなるように制御する。

ここで探索スケール制御の実現方法について考察する。交叉や突然変異などの遺伝子操作によって探索スケールを制御するためには、各遺伝子座が表現型にどの程度影響を与えるか（遺伝子座の影響度）を考慮することが不可欠と考える。

各遺伝子座の影響度が判明していれば、適切な影響度を持った遺伝子座に対し多様性制御などの遺伝子操作を行うことで、影響度に応じた探索スケールが実現できると考える。たとえば、遺伝子座の影響度が大きな遺伝子座群において多様性を発生させれば個体は表現型空間上に大きく分散し探索スケールが大きくなり、反対に影響度が小さな遺伝子座群において多様性を発生させれば狭い範囲で分散し探索スケールが小さくなると考えられる。

## 3. 提案手法

### 3.1 概要

著者らはこれまでに遺伝子座の影響度を考慮して探索スケール制御を行う GA を開発しており、これを実際に動的環境の問題に適用してその有効性を確認している<sup>16)</sup>。しかし、これまでの手法は、探索スケール制御に必要な遺伝子座の影響度の情報を前処理によって同定してから、GA ループを実行する方法を採用しているため、影響度が GA ループ実行中に変化しないことが前提となっていた（図 1(a)）。そのため、環境変動にともない遺伝子座の影響度が変化する問題には適用できないという問題点があった。そこで、本研

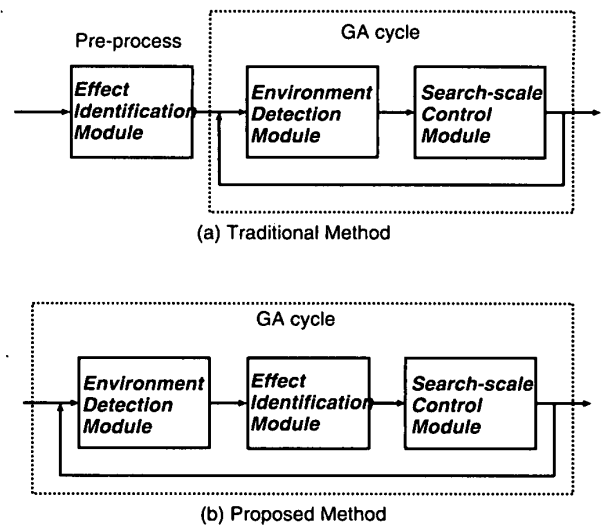


図 1 従来手法<sup>16)</sup>と提案手法の違い  
Fig. 1 Difference between traditional and proposed methods.

究では実行中に環境変動があった場合は、遺伝子座の影響度を同定しなおすという処理を GA ループ内に組み込むことを考える（図 1(b)）。これにより GA ループ実行中のオーバーヘッドが大きくなるが、従来対処できなかったクラスの問題にも対処できるようになる。

また、これまでの手法は急激な環境変動が発生したときに一時的に急激な適応度低下を引き起こす可能性があった。そこで、提案手法では個体集団を大きく 2 つのグループに分けて、一方のグループを GA とは無関係につねにランダムに個体を作り出すことにする。解の改善には貢献しないが、GA のみの集団に比べて環境変動時の急激な適応度低下をある程度抑える効果がある。

提案する GA は、(i) 環境状態検出モジュール、(ii) 影響度同定モジュール、(iii) 探索スケール制御モジュールで構成されており、GA ループ中にこれらのモジュールが順次呼び出されて実行される。(i) では環境状態を調べ、環境状態に応じて適切な探索スケールを設定する。(ii) では (i) で環境変動を検出した場合に遺伝子座の影響度を同定する。(iii) では (i) で設定された探索スケールを実現するように、遺伝子座の影響度を考慮した GA 操作を適用する。以下に各モジュールの処理内容の詳細を示す。

### 3.2 (i) 環境状態検出モジュール

このモジュールは、環境状態を調べ、環境状態に応じて適切な探索スケールを設定することを目的とする。

時刻  $t$  での探索スケールを  $S(t)$  で表す。 $S(t)$  は段階的に定義され、 $1 \sim R$  のいずれかの整数値をとる。値が大きくなるほどより大域的に探索を行うことを表す。ここで、 $R$  は探索スケールの段階数を表すパ

表1 各環境状態における操作

Table 1 Operations in each state of environment.

Conditions	Operations
$r > T_{inc}, S(t) > 1$	$S(t) \leftarrow S(t) - 1$
$ r  > T_{dec}, S(t) < L$	$S(t) \leftarrow S(t) + 1$
$r = 0$	if $f(i_{Rmax}) < f(i_{GAmax})$ then $swap(i_{Rmax}, i_{GAmax})$

Here,  $i_{Rmax}$  and  $i_{GAmax}$  are the best individuals in Random and GA Populations,  $f(i)$  is the fitness of individual  $i$ ,  $swap(i, j)$  is an operator for swapping individuals  $i$  and  $j$ .

ラメータであり提案手法で加わる GA パラメータの 1 つである。

環境状態を検出するために以下の指標値  $r$  を用いる。

$$r = \frac{f_{max}(t) - f_{max}(t-1)}{f_{max}(t-1)} \quad (1)$$

ここで  $f_{max}(t)$  は世代  $t$  での GA 集団における最良個体の適応度である。ただし、適応度は 0 以上になるようにスケールされているものとする。

$r > 0$  は最良個体の適応度が上昇している状態を表している。 $S(t) > 1$  かつ  $r > T_{inc}$  ( $T_{inc}$  はある閾値で正の値) のときは、大域的な探索によって適応度ランドスケープ上の「山」をとらえた可能性がある。このとき、とらえた山の近傍探索を行うために探索スケールを現在よりも 1 段階小さく設定する。すなわち、 $S(t) \leftarrow S(t) - 1$  とする。

$r = 0$  は解が改善しなかった状態を表している。このとき、最適解か局所解に到達した可能性がある。局所解の場合は脱出する必要があるのでランダム集団における最良個体の適応度が GA 集団における最良個体の適応度よりも大きければ、両者を入れ換える。

$r < 0$  は最良個体の適応度が低下している状態を表している。エリート保存戦略を併用した場合、環境が変化しない限りは  $r < 0$  の状態になることはない。よって  $r < 0$  を検出した場合は環境変動が発生したと考えられる。ここで  $S(t) < R$  かつ  $|r| > T_{dec}$  ( $T_{dec}$  はある閾値で正の値) のときは、集団が適応度ランドスケープ上の山の位置から外れた可能性がある。そこで、新たな山の位置の発見に向けて大域的な探索を行うため探索スケールを 1 段階大きくする。すなわち、 $S(t) \leftarrow S(t) + 1$  とする。

以上をまとめると表 1 のようになる。

### 3.3 (ii) 影響度同定モジュール

このモジュールは、(i) において環境変動を検出した場合に各遺伝子座の影響度の再同定処理を行うことを目的とする。

第  $i$  ビット目の遺伝子座の影響度  $e_i$  を以下のように

procedure locus-effect identification

```

for i=0 to L-1
  e[i]=0;
  for each s in P
    si=perturb(s,i);
    di=|p(si)-p(s)|;
    if di>e[i] then e[i]=di;
  endfor
endfor

```

図 2 遺伝子座の影響度の同定処理

Fig. 2 Locus-effect identification.

に定義する。

$$e_i = \max_{s \in P} |p(s_i) - p(s)| \quad (2)$$

ここで、 $P$  は個体集団、 $s$  は  $P$  における任意の個体の遺伝子列、 $s_i$  は  $s$  の第  $i$  ビット目を反転させた遺伝子列。 $p(s)$  は  $s$  の表現型値を表す。 $e_i$  を求めるための具体的な手順を図 2 に示す。図で  $perturb(s, i)$  は  $s$  の第  $i$  ビットを反転する処理を表す。表現型を 1 つの数値として表現しない場合、何らかの形で表現型を数値にマッピングする必要がある。マッピング戦略次第で遺伝子座の影響度の正確さが決定されるためマッピング戦略の設計は重要である。問題に依存しないマッピング戦略として適応度をそのまま表現型値として使う方法が考えられる。ただし、問題によっては誤差が大きくなる可能性があるので使用には十分な注意が必要となる。

式 (2) で定義した指標は、バイナリーコーディングなど遺伝子座が独立して表現型の形成に貢献する場合に有効であるが、遺伝子座間に相互作用 (エピスタシス) がある場合は問題が生じる。以下、エピスタシスが存在する場合の対処法について説明する。

エピスタシスが存在する場合は、1 つ 1 つの遺伝子座の影響度は一意に決められないので、エピスタシス関係にある遺伝子座群 (リンケージ) に対する統計的な影響度を求めることにする。そこで、まず初めにどの遺伝子座間で相互作用があるかを見つけるリンケージ同定を行う。リンケージ同定の手法として LINC (Linkage Identification by Nonlinearity Check)<sup>17)</sup> が提案されている。本論文でも基本的にはこの方法を採用する。ただし、LINC はリンケージ同定の際に適応度の変化量を目安にしているが本研究では表現型値の変化量を用いる点が異なる。なお、表現型を数値で表現しない場合は前述したようにマッピング戦略を適用し数値化するものとする。

遺伝子列  $s$  の表現型値を  $p(s)$  とする。 $s$  の第  $i$  ビット目を反転させた遺伝子列  $s_i$  とする。 $s$  の  $i$  ビット目と  $j$  ビット目を同時反転させた遺伝子列を  $s_{ij}$  とす

```

procedure LINC
for each s in P
  for i=0 to L-1
    si=perturb(s,i);
    for j=0 to L-1
      sj=perturb(s,j);
      di=p(si)-p(s);
      dj=p(sj)-p(s);
      dij=p(sij)-p(s);
      if dij = di + dj then
        adding j to the linkage_set[i].
        adding i to the linkage_set[j].
      endif
    endfor
  endfor
endfor

```

図3 LINCによるリンケージ同定処理  
Fig. 3 Linkage identification by LINC.

```

procedure linkage-effect identification
for i=0 to N-1
  le[i]=0;
  for each s in P
    dp=0;
    for each p in pattern_set[i]
      sp=purturb(s,i,p);
      dp=p(sp)-p(s);
      if dp>ep then ep=dp;
    endfor
    if ep>le[i] then le[i]=ep;
  endfor
endfor

```

図4 リンケージの影響度の同定処理  
Fig. 4 Linkage-effect identification.

る。このとき以下のように、表現型値変化量  $\Delta p(s_i)$ ,  $\Delta p(s_j)$ ,  $\Delta p(s_{ij})$  を求める。

$$\Delta p(s_i) = p(s_i) - p(s) \quad (3)$$

$$\Delta p(s_j) = p(s_j) - p(s) \quad (4)$$

$$\Delta p(s_{ij}) = p(s_{ij}) - p(s) \quad (5)$$

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

$$\Delta p(s_{ij}) \neq \Delta p(s_i) + \Delta p(s_j) \quad (6)$$

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てそれぞれをリンケージ集合に入れる。具体的な手順を図3に示す。

次に、各リンケージが表現型に与える影響度を計算する。リンケージ同定によって遺伝子列中に  $N$  個の独立したリンケージが見つかったとする。このとき  $i$  番目のリンケージの影響度  $le_i$  を次式により定義する。

$$le_i = \max_{s \in P} \max_{p_i \in G_i} |p(s_{p_i}) - p(s)| \quad (7)$$

ここで、 $P$  は個体集団、 $s$  は  $P$  における任意の個体の遺伝子列、 $G_i$  は  $i$  番目のリンケージがとりうる遺伝子パターンの集合、 $p_i$  は  $G_i$  中の任意の遺伝子パターン、 $s_{p_i}$  は  $s$  の第  $i$  リンケージを遺伝子パターン  $p_i$  に置き換えた遺伝子列、 $p(s)$  は  $s$  の表現型値を表す。 $le_i$  を求めるための具体的な手順を図4に示す。図で  $perturb(s, i, p)$  は  $s$  の第  $i$  リンケージにおける遺伝子パターンを遺伝子パターン  $p$  によって置き換える処理を表す。

### 3.4 (iii) 探索スケール制御モジュール

このモジュールは、GA操作（評価、選択、交叉・突然変異など）によって (i) で設定された探索スケールを実現することを目的とする。

準備として (ii) で求めた影響度を基準に遺伝子座（あるいはリンケージ）を  $R$  段階にグループ分けしておく（ $R$  は制御したい探索スケールの段階数）。たとえば、影響度の小さい遺伝子座（リンケージ）のグ

ループ、影響度が次に小さい遺伝子座（リンケージ）のグループといった具合にグループ化しておく。また、影響度の小さいグループから大きいグループの順に  $1 \sim R$  までの通し番号を付けておく。ここで、各グループをブロックと呼ぶことにする。

このとき、ブロックの番号が大きくなるほど、そのブロック内で多様性を発生させたときに表現型空間中で個体が大きく広がる。そこで、(i) によって設定された探索スケールを  $S$  とする。このとき  $S$  番のブロックを選びそのブロック内で多様性を発生させる処理を行うことで所望の探索スケールを実現する。

特定のブロック内で多様性制御を行い、探索スケール制御を行うためのGA操作について以下に述べる。

#### 適応度評価

適応度評価の段階で、目的関数値のほか、対象ブロック内の遺伝子パターンの希少性を考慮して適応度を計算すればブロック内で多様性が発生する<sup>14),15)</sup>。そこで、適応度計算を次式のように定義する。

$$f = a_1 f_1 + a_2 f_2 \quad (8)$$

ここで、 $f_1$  および  $f_2$  はそれぞれ目的関数値および希少度で、 $a_1$ ,  $a_2$  は  $f_1$ ,  $f_2$  のバランスをとるための係数である。個体の希少度  $f_2$  は次式のように定義する。

$$f_2 = \frac{1}{M(i)} \quad (9)$$

ここで、 $M(i)$  は、対象ブロック内で自分の遺伝子パターン  $i$  と同じ遺伝子パターンを持つ個体数である。

#### 交叉・突然変異

GAにおいて、交叉は親個体の形質の一部を子個体へ伝えるための重要な操作である。特にエピスタシスがある場合、交叉戦略を工夫しないと形質保存がうまく作用しない可能性がある。そこで、ブロック単位で遺伝子列の交換を行うブロック交叉と呼ばれる交叉方法を用いる。ブロック交叉では各ブロックを個体間に入れ換えるかどうかを表すマスクビット列と呼ばれるビット列を用いる。マスクビット列の各ビット位置は

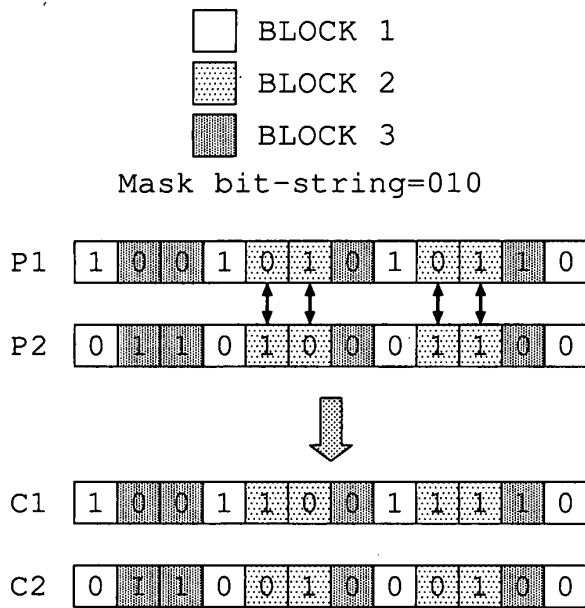


図5 ブロック交叉  
Fig. 5 Block crossover.

各ブロックに対応しており、1の値をとるブロックが遺伝子の交換の対象となる。具体例として図5を用いて、親個体  $P_1$ ,  $P_2$  から子個体  $C_1$ ,  $C_2$  が生成される様子を説明する。各遺伝子座は1~3のいずれかのブロックに属しており、マスクビット列として010が与えられているとする。このとき遺伝子交換の対象となるブロックは、マスクビット列で1の値をとる位置に対応するブロック2である。そこで、 $P_1$  と  $P_2$  のブロック2に属する遺伝子座の遺伝子を交換し、 $C_1$  および  $C_2$  が得られる。ブロック交叉によってブロック内の形質は損なわれずに次の世代に受け継がれる。

次にブロックごとに突然変異率を制御し意図的に多様性を発生させることを考える。ブロック内の遺伝子座の突然変異確率  $p'_m(i)$  を次式のように設定する。

$$p'_m(i) = M(i) \cdot p_m \quad (10)$$

ここで、 $p_m$  は通常の突然変異確率、 $M(i)$  はブロック内で遺伝子パターン  $i$  を持つ個体の数である。この方法を使えば、希少性が高くない個体は突然変異率は高くなる。

#### 4. 実験

本手法の有効性を確認するために3種類の実験を行う。実験1として複数の環境変化パターンが混在するクラスの問題、実験2として環境変化によって遺伝子座の影響度が変化するクラスの問題、実験3として遺伝子座間にリンケージが存在するクラスの問題を用いる。また、これらの実験においてシェアリング法<sup>14)</sup>と、超突然変異 (Hypermutation) を導入したGA<sup>12)</sup> (本論文ではHGAと呼ぶ) との比較を行う。これら

の手法は、これまでに動的環境への対処法とされてきた多様性維持・発生的手法としてよく知られている。

##### 4.1 実験 1

ここでは複数の環境変化パターンが混在するクラスの問題に対する提案手法の有効性を確認する。取り扱う問題は変動ナップザック問題とする。変動ナップザック問題とはナップザック問題における様々なパラメータが時間によって変化する問題であり、以下のように定式化できる<sup>15)</sup>。

$$\begin{aligned} \max_{x_i(t) \in 0,1} z &= \sum_{i=1}^N v_i(t)x_i(t) \\ \text{subject to} & \sum_{i=1}^N w_i(t)x_i(t) \leq W(t) \quad (11) \\ & t = 1, 2, \dots \end{aligned}$$

ここで  $N$  は荷物の数、 $w_i(t)$ ,  $v_i(t)$  はそれぞれ各荷物の重量と価値、 $W(t)$  はナップザック問題の制限重量、 $t$  は時刻である。

ナップザック問題におけるGA個体の評価関数は、選んだ荷物の重さの合計が制限重量以内であれば価値の合計値、さもなければ制限重量から選んだ荷物の重さの合計を引いた値(負の値になる)とする。

実験1では上記の変動ナップザック問題で  $W(t)$  を変化させる。 $W(t)$  の変化パターンとして断続的で急激な変化と連続的で緩やかな変化の2つのパターンが表れるとする。断続的で急激な変化として100世代経過ごとに  $W(t)$  を直前の世代の80%に減少させる。また連続的で緩やかな変化としては各世代ごとに  $W(t)$  を1ずつ減らしていく。

荷物の数は  $N = 15$  とする。各荷物の重さ  $w_i$ , 価値  $v_i$  は1-500の範囲でランダムに設定する。制限重量の初期値は  $W(0) = 2000$  とする。また、提案手法、シェアリング法、HGAのパラメータを表2に示す。

ナップザック問題では選んだ荷物の重量と価値の2つの表現型がある。提案手法において各遺伝子座の影響度を求めるためには、表現型の変化量を求める必要があるが、2つの表現型がある場合はこれを1つの表現型値としてマッピングする必要がある。そこで、本実験では選んだ荷物の重量と価値の積の総和を表現型値  $P(t)$  とするマッピング戦略を用いることにした。すなわち、

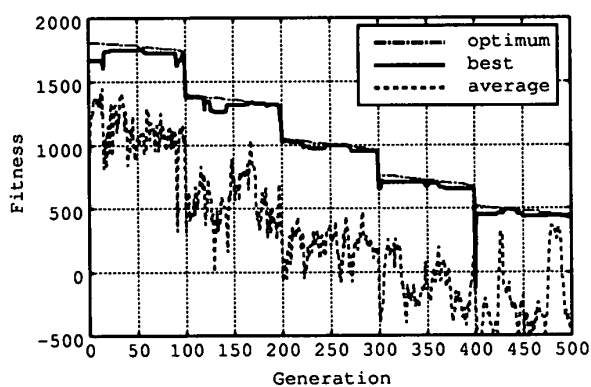
$$P(t) = \sum_{i=1}^N w_i(t)v_i(t)x_i(t) \quad (12)$$

とした。

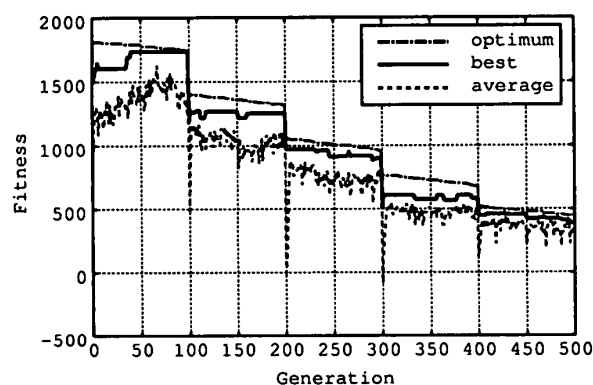
実験結果を図6に示す。提案手法、シェアリング法、

表2 GA パラメータ  
Table 2 GA parameters.

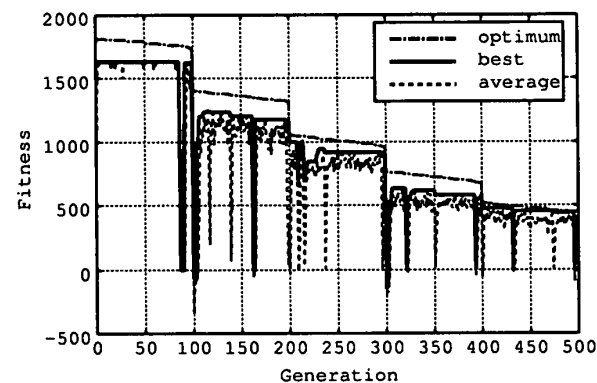
	proposed method	sharing method	HGA
generation		500	
population size	GA population 20 random population 10		30
bit length		15	
block size	3		-
sharing parameters <sup>14)</sup>	-	$\sigma_{share} = 10, \alpha = 0.1$	-
mutation rate	basic rate=0.01	0.01	0.01 (hyper mutation rate=0.1)
selection Strategy	tournament strategy (tournament size=2), elite selection (size=2)		
crossover Strategy	block crossover	simple crossover	



(a) Proposed-GA



(b) Sharing GA



(c) Hyper-mutation GA

図6 適応度の比較 (実験1)

Fig. 6 Comparison of fitness (Experiment 1).

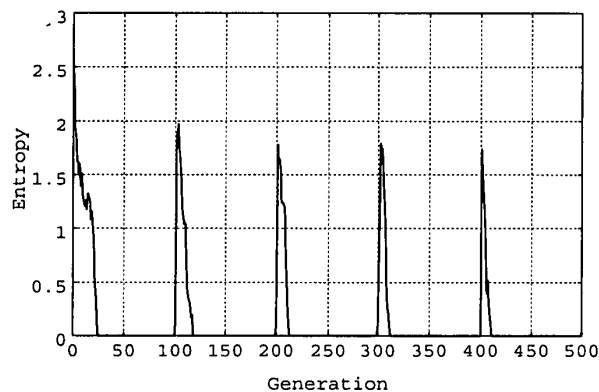
HGAのそれぞれを適用したときの最適解 (optimum), 各時点での最良個体の適応度 (best), 各時点での平均適応度 (average) の推移を示している。

100世代ごとに発生する急激な環境変動時における3手法の振舞いを見てみる。HGAでは100世代ごとに一時的に大きく最良個体の適応度が減少しているが、このとき超突然変異が適用されるためその後の回復スピードは比較的良好であった。シェアリング法は適応度の低下は比較的小さい。シェアリング法では恒常的に多様性を維持するため個体が大域的に分散されやすく、その結果急激な変化にも対応できたと考えられる。提案手法は適応度低下が比較的小さく、その後の追従スピードも良好であった。急激な適応度低下を防ぐことができたことからランダム集団の導入効果が認められる。また、追従スピードが良好であったことから、GA集団に対する探索スケール制御がうまく作用したと考えられる。GA集団は急激な環境変動を検出した場合、影響度の大きなブロックで多様性を持たせようとする。このことが個体を大域的に分散させ、環境の追従スピードを高める原因であると考えられる。図7は提案手法における各ブロックの多様性の推移を示したものである。多様性の評価は情報エントロピー<sup>18)</sup>を用いた。情報エントロピーは次式で定義される。

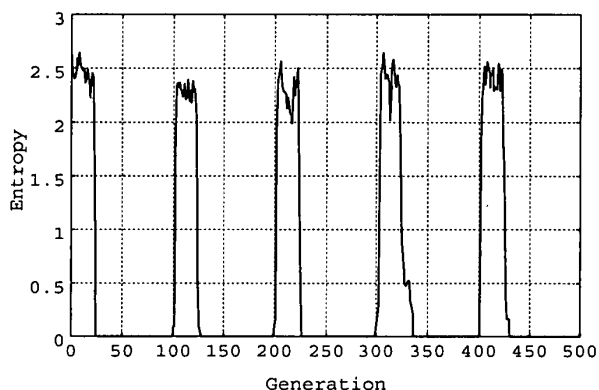
$$H_i = - \sum_{j=1}^{M_i} p_{ij} \log p_{ij} \quad (13)$$

ここで、 $H_i$ はブロック*i*の情報エントロピー、 $M_i$ はブロック*i*がとりうるパターンの総数、 $p_{ij}$ はブロック*i*に占める遺伝子パターン*j*の割合である。図から100世代ごとに影響度が最も大きいブロック3やその次に大きいブロック2で多様性が発生していることが分かる。

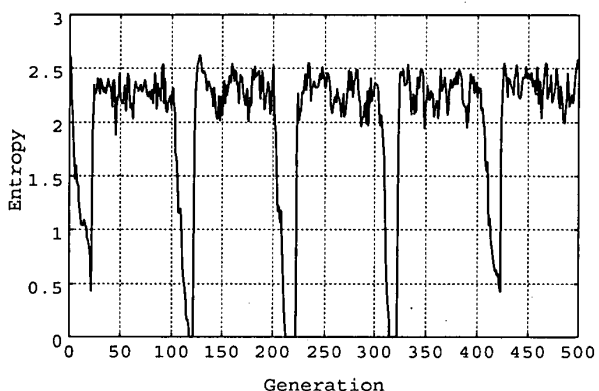
次に、環境が連続的に変化しているときの3手法の振舞いを見てみる。図6から連続変化時において、シェアリング法、HGAは最適解までたどり着いてお



(a) block 3



(b) block 2



(c) block 1

図7 各ブロックのエントロピー (実験1)

Fig. 7 Entropy in each block (Experiment 1).

らず進化的に停滞していることが分かる。シェアリング法は遺伝子座の影響度を考慮せずに全遺伝子座に対し均質な多様性維持を行うため個体が大域的に分散しやすく、その結果、近傍探索の性能が劣ってしまうことが原因と考えられる。また、HGAは最良個体の適応度が下がった段階で超突然変異を行うがどの遺伝子座も均質な確率で突然変異を行うため、やはり大域的に個体が分散してしまい近傍探索が行えないことが原因と考えられる。また、HGAは通常時は単純GAと同様、最良個体の遺伝子が集団中に急速に広がり多様性を失ってしまうことが多い。この状態になると緩や

かな環境変動によってそれまでの最良個体が制限重量をオーバーしてしまうと急激に集団の平均適応度が下がる。図でHGAが所々急激に平均適応度を下げているのもこのことが原因と考えられる。一方、提案手法では小さな環境変動を検出したときは、影響度の小さな遺伝子座において多様性維持を図るため、最良個体の近傍に他の個体が分布し近傍探索が行われる。その結果、連続的な変化に対する追従性も他の2手法に比べて良好な結果を示したと考えられる。図7において環境が連続変化している間は最も影響度が小さなブロック1で多様性が発生していることが分かる。

最後に、提案手法と従来手法<sup>16)</sup>との関係について考察する。これらの手法の違いは遺伝子座の影響度同定をGAループの外側で行うか内側で行うかという点にある。この実験では全体を通じて遺伝子座の影響度は変化しないので、影響度の同定をGAループの外側で行っても内側で行っても結果は変わらない。よって、このクラスの問題に関しては従来手法を用いても同様の結果が得られると考えられる。

#### 4.2 実験 2

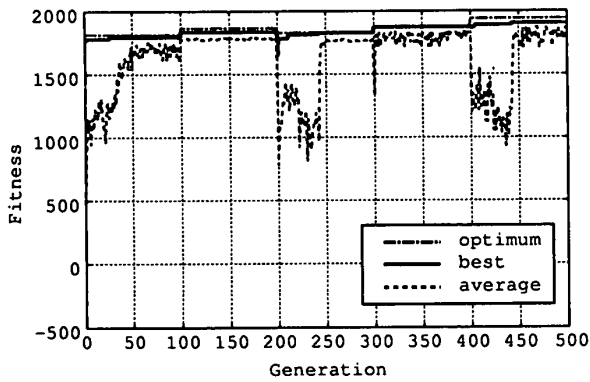
実験2では遺伝子座の影響度が時間によって変化するクラスの問題を扱う。このクラスは、従来手法<sup>16)</sup>(遺伝子座の影響度の同定をGAのループの外側で行う方法)では扱えなかったクラスである。

遺伝子座の影響度が変化するクラスの問題として、変動ナップザック問題において各荷物の重さと価値 $w(t)$ ,  $v(t)$ を時間によって変化させる。具体的には、環境変動によって各荷物の重さと価値は $\pm 10\%$ の間で変化するものとする。また、環境変動のタイミングは100世代ごととする。したがって100世代ごとに遺伝子座の影響度が変化するようになる。

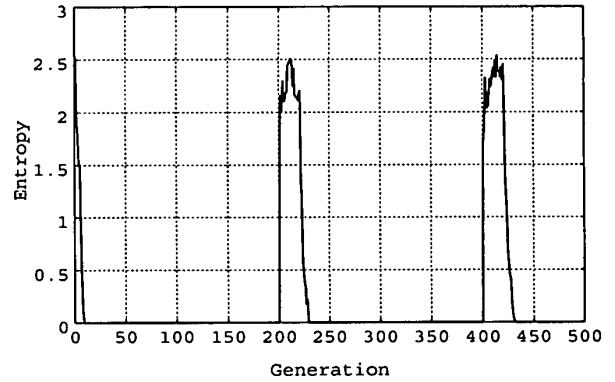
荷物の数は $N = 15$ とする。各荷物の重さ、価値の初期値 $w_i(0)$ ,  $v_i(0)$ は1-500の範囲でランダムに設定する。制限重量の初期値は $W(0) = 2000$ とする。また、2つの実験で用いた提案手法、シェアリング法、HGAのパラメータは実験1と同様、表2のものを用いた。

実験結果を図8に示す。図は提案手法、シェアリング法、HGAのそれぞれを適用したときの最適解(optimum)、各時点での最良個体の適応度(best)、各時点での平均適応度(average)の推移を示している。

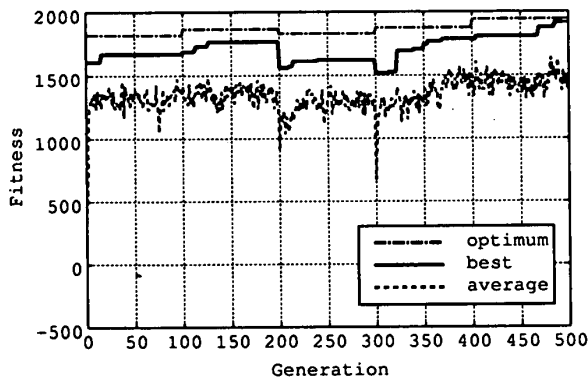
図から提案手法だけが最良個体が最適解に追従していることが分かる。図9に提案手法における各ブロックの多様度の推移を示した。200世代目と400世代目でブロック3やブロック2などの影響力が大きなブロックで多様性が発生していることから、最良個体の



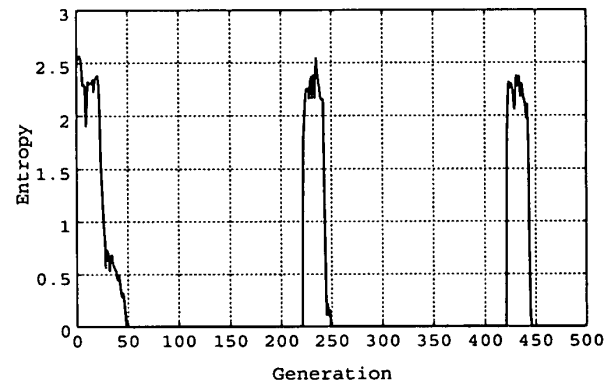
(a) Proposed-GA



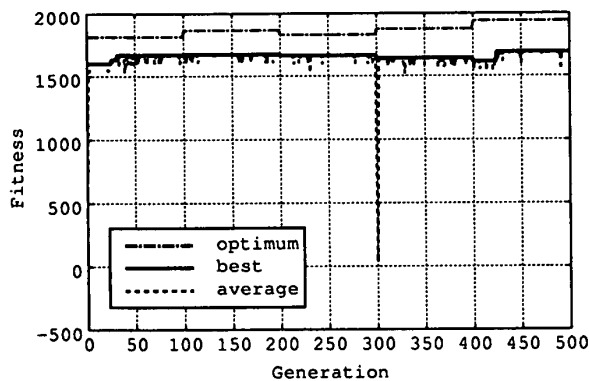
(a) block 3



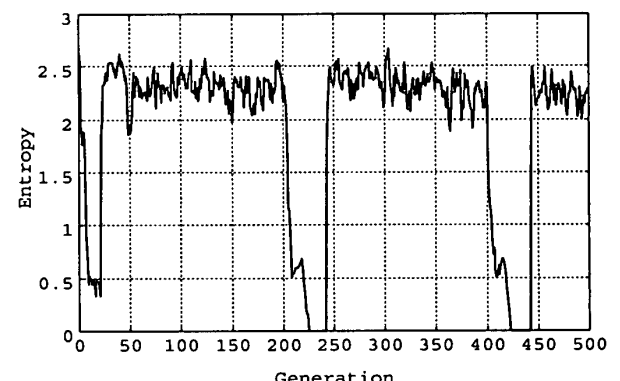
(b) Sharing GA



(b) block 2



(c) Hyper-mutation GA



(c) block 1

図8 適応度の比較 (実験2)

Fig. 8 Comparison of fitness (Experiment 2).

図9 各ブロックのエントロピー (実験2)

Fig. 9 Entropy in each block (Experiment 2).

適応度が大きく減少したと考えられる。しかし、図8を見ると最良個体の適応度低下はあまり起きていない。このことから、ランダム集団による適応度低下の防止効果があったと考えられる。また、環境変化後も最適解への追従性が持続していることからブロックの再同定による効果が表れたと考えられる。もしブロックの再同定を行わなかったとしたら、各ブロックを構成する遺伝子座の影響度は環境変動によって大きくばらついてしまい、探索スケール制御が正確にできなくなる可能性があるからである。シェアリング法は環境変動時の適応度低下は比較的小さいが、最良個体が最適解

にたどり着けず進化的停滞の状態に陥ってしまっていることが分かる。これは大域的な探索に強いが、近傍探索性能に劣ることが原因と考えられる。HGAも最良個体が最適解に追従できていない。しかし、環境変動時の適応度低下はほとんど見られない。これは、各荷物の重さと価値が $\pm 10\%$ という比較的小さな範囲で変化していることと、HGAの局所解に陥りやすい性質によるものと考えられる。最適解付近の解は荷物の総重量が制限重量に近く、わずかな荷物の重さの変動で制限重量をオーバーしてしまい適応度が急激に低下する可能性があるのに対し、局所解付近では比較的制限



重量に余裕がある場合が多く、荷物の重さに多少の変化があったとしても現在の組合せのまま制限重量内に収まる確率が高いことが原因ではないかと考える。

### 4.3 実験 3

実験3では遺伝子座間にリンケージが存在する場合を扱う。動的環境として、リンケージを構成する遺伝子座の組合せが時間によって変化するものとする。リンケージが実行中に変化することから、従来手法<sup>16)</sup>のようにリンケージの同定をGAループの外側で行ったのでは対処できない。本手法のようにGAループの内側で適宜リンケージの再同定を行うことが必要になる。

遺伝子列を、リンケージ部分（リンケージを構成する遺伝子座群）と、非リンケージ部分（リンケージを構成しない遺伝子座群）に分ける。このとき、時刻  $t$  で、リンケージ部分が与える表現型値を  $P_L(t)$ 、非リンケージ部分が与える表現型値を  $P_N(t)$  とする。また、遺伝子列中に  $N_L(t)$  個のリンケージが存在するものとし、 $i$  番目のリンケージが与える表現型値を  $P_{L_i}(t)$  とする。このとき、この遺伝子列の表現型値  $P(t)$  は次式のように定義できる。

$$\begin{aligned} P(t) &= P_L(t) + P_N(t) \\ &= \sum_{i=1}^{N_L(t)} P_{L_i}(t) + P_N(t) \end{aligned} \quad (14)$$

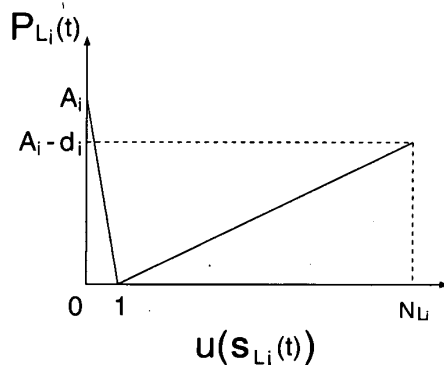


図10 トラップ関数  
Fig. 10 Trap function.

本実験の設定として、 $P_{L_i}(t)$  はトラップ関数<sup>19)</sup>の出力値、 $P_N(t)$  は OneMax 関数の出力値を用いる。また、適応度は表現型値  $P(t)$  をそのまま用いる。よって、この実験はGAによって  $P(t)$  の最大値を探索することが目的となる。以下に  $P_{L_i}(t)$  および  $P_N(t)$  の定義を示す。

$$P_{L_i}(t) = \begin{cases} \frac{(A_i - d_i) \{u(\mathbf{S}_{L_i}(t)) - 1\}}{N_{L_i}} & \text{if } 1 \leq u(\mathbf{S}_{L_i}(t)) \leq N_{L_i} \\ A_i & \text{if } u(\mathbf{S}_{L_i}(t)) = 0 \end{cases} \quad (15)$$

$$P_N(t) = u(\mathbf{S}_N(t))$$

$\mathbf{S}_{L_i}(t)$  は時刻  $t$  で、 $i$  番目のリンケージを構成する遺伝子（ビット）の集合、 $N_{L_i}$  は  $i$  番目のリンケージを構成する遺伝子座数、 $u(\mathbf{S})$  はビット集合  $\mathbf{S}$  に含まれる 1 の数を示す。  $A_i$ ,  $d_i$  は定数で  $0 \leq d_i \leq A_i$  の関係がある。  $\mathbf{S}_N(t)$  は時刻  $t$  において非リンケージ部分を構成する遺伝子（ビット）の集合である。  $P_{L_i}(t)$  と  $u(\mathbf{S}_{L_i}(t))$  の関係を図10に示す。

実験のその他の設定として、リンケージの数は時間によらず2とする。また、各リンケージを構成する遺伝子座数を5、非リンケージを構成するの遺伝子座数を5とする。よって、遺伝子列長は15となる。動的環境として、各リンケージおよび非リンケージを構成する遺伝子座の組合せが100世代ごとに変化するものとする。2つのリンケージに対応する定数  $A_i$ ,  $d_i$  (式(15))の値は、 $A_1 = 10$ ,  $A_2 = 20$ ,  $d_1 = 2$ ,  $d_2 = 4$  とする。また、実験で用いるGAパラメータは表3のとおりである。

以上の設定の下で、提案手法、シェアリング法、HGAを適用した結果について述べる。図11は各時点での最良個体の適応度 (best)、集団の平均適応度 (average) および最適解 (optimum) の推移を示したものである。図から提案手法は、最良個体の適応度が最適解に追従できていることが分かる。一方、シェアリング法、HGAは局所解に陥っていることが分かる。この実験では

表3 GAパラメータ (実験3)

Table 3 GA parameters (Experiment 3).

	proposed method	sharing method	HGA
generation		500	
population size	GA population 30 random population 20		50
bit length		15	
block size	3		-
sharing parameters <sup>14)</sup>	-	$\sigma_{share} = 10, \alpha = 0.1$	-
mutation rate	basic rate=0.01	0.01	0.01 (hyper mutation rate=0.2)
selection Strategy	tournament strategy (tournament size=2), elite selection (size=2)		
crossover Strategy	block crossover	simple crossover	

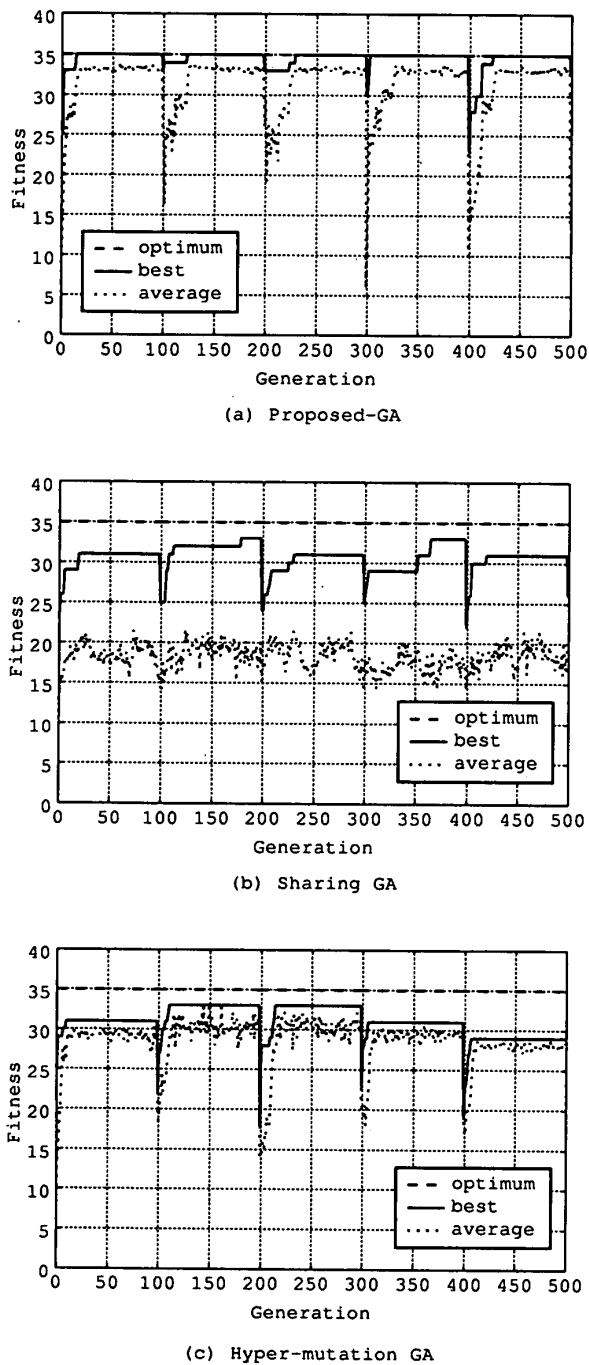


図 11 適応度の比較 (実験 3)

Fig. 11 Comparison of fitness (Experiment 3).

ンケージ部分に集中して多様性発生を行わないと  
 ラップ関数によって局所解に陥りやすくなる。シェア  
 リング法はどの遺伝子座も対等な立場で多様性維持が  
 行うので、集中して多様性発生を行う場合に比べてリ  
 ンケージ部分の多様性が小さい。このことがシェア  
 リング法が最適解を得られなかった原因と考えられる。  
 同様のことは HGA の超突然変異の適用時においても  
 いえる。提案手法は、各リンケージを構成する遺伝子  
 座群を 1 つのブロックとしてグループ化する。探索  
 スケール制御の過程で特定のブロック内で多様性維持

が行われるので、リンケージ部分に集中して多様性を  
 発生させることができ、局所解を回避できたと考えら  
 れる。また、図から、100 世代ごとの環境変化の後も  
 提案手法は最適解へ追従できていることが分かる。こ  
 れは、GA ループの内側でリンケージの再同定処理を  
 行っているため、環境変動によってリンケージを構成  
 する遺伝子座の組合せが変更されたとしてもただちに  
 同定されるからである。ただし、一般にリンケージ同  
 定処理は計算コストが大きい。そのため、リンケージ  
 同定をいかに効率的に行うかが実用において重要とな  
 る。実験 3 は単純な実験であるので問題ないが、問題  
 の規模が大きくなったときにリンケージ同定による計算  
 コストが問題となる可能性がある。しかし、本論文で  
 用いているリンケージ同定処理部分は他の処理と独立  
 しているため、より効率の良いリンケージ同定処理ア  
 ルゴリズムと入れ換え可能である。今後リンケージ同  
 定アルゴリズムが改善されればそれらを用いることで  
 この問題を解決できると考えられる。

次に、提案手法の探索スケール制御の動作の検証を  
 行う。各ブロックの多様性の推移を図 12 に示す。最  
 も影響力の大きなブロック 3 で多様性が発生するのは  
 実行直後および 100 世代ごとの環境変化直後であるこ  
 とが分かる。つまり、新しい環境に変わった直後に大  
 域的な探索が行われていることになる。このことから、  
 提案手法の環境変化の検出、探索スケール制御が効果  
 的に働いていることが分かる。また、環境変動後しば  
 らくすると多様性の発生がブロック 3 からブロック 2  
 へ移っている。さらに時間がたつと今度はブロック 2  
 からブロック 1 へと多様性の発生が移っていることが  
 分かる。これは大域的な探索が終了し、解の質の向上  
 を求めてより局所的な探索に移っていていることを  
 示している。このことは、探索スケール制御によって  
 提案手法の探索方針がうまく実現されていると考えら  
 れる。

## 5. おわりに

本論文では GA による動的環境下での最適化につ  
 いて検討した。特に、環境の変化パターンが事前に分  
 からない場合に必要と考えられる探索スケール制御  
 について検討した。また、探索スケール制御を行う際  
 に必要となるブロック同定処理について検証した。特  
 に、ブロック同定を GA ループの内側に組み込むこ  
 とで、遺伝子座の影響度が実行中に変化するようなク  
 ラスの問題でも扱えるようになることを示した。提案  
 手法の有効性の確認として 3 種類の実験を行った。実  
 験 1 では、適応度ランドスケープ上において複数の

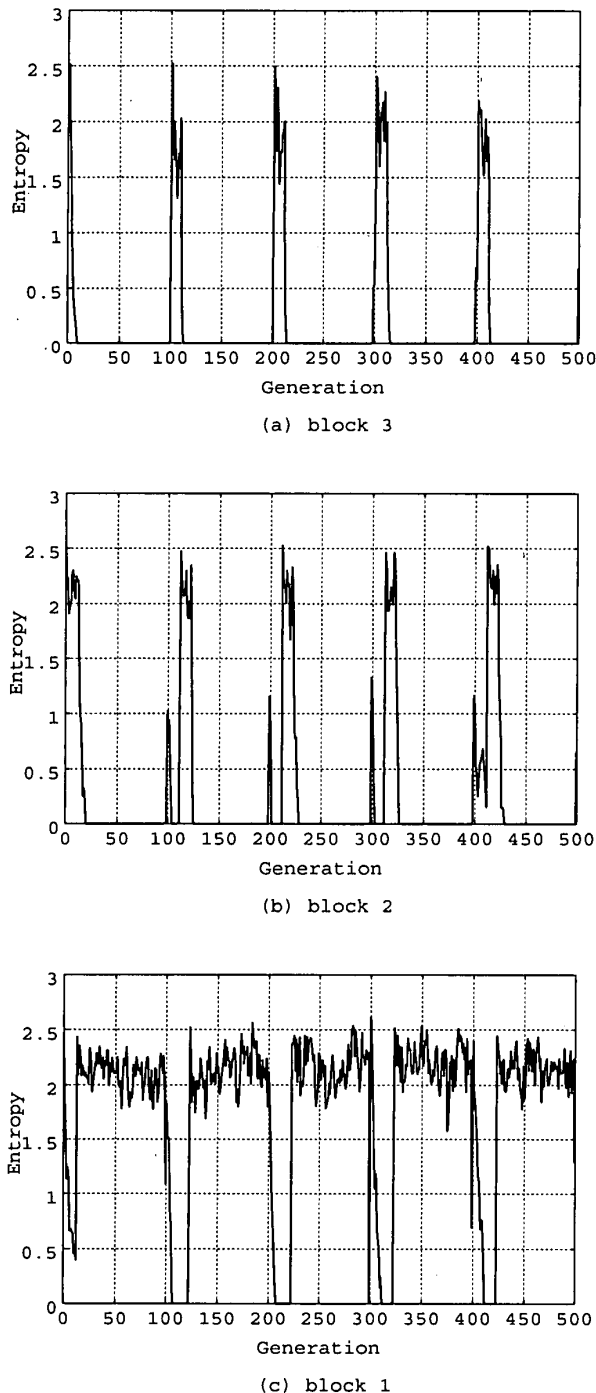


図12 各ブロックのエントロピー (実験3)

Fig.12 Entropy in each block (Experiment 3).

変化パターンが表れるクラスの問題，実験2では遺伝子座の影響度が環境によって変化するクラスの問題，実験3ではリンケージが存在するクラスの問題を扱った。これらの実験において提案手法がシェアリング法やHGAに比べて，大域的追従性，局所的追従性の双方に優れた特性を示すことが確認できた。

提案手法はバイナリーコーディングの問題を前提としている。今後，実数値GA，スケジューリング，TSPなど，より複雑な問題に対する拡張可能性について検討することが必要である。実数値GAでは，遺伝子座

の影響度同定アルゴリズムを再設計する必要があるが，影響度を定義できればそれを元に探索スケール制御をそのまま適用できると考えられる。スケジューリングやTSPなど複雑な問題に関しては，遺伝子型空間と表現型空間の関係を分析し，探索スケール制御が可能となるオペレータ（遺伝子型空間での操作）を定義するという原点に立ち返って考える必要があると思われる。その際に，必ずしも遺伝子座を単位として影響度を定義できるとは限らないと考えられる。しかし，いずれにしても，探索スケール制御という基本的なアイデアは動的環境に対処するための一般的な枠組みとして用いることができると考える。

### 参考文献

- 1) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley (1989).
- 2) 北野宏明：遺傳的アルゴリズム1~4，産業図書(1993~2000)。
- 3) Karr, C.L. and Freeman, L.M.: *Industrial Applications of Genetic Algorithms*, CRC Press (1999).
- 4) Yamazaki, K.: Dynamic Pareto Optimum GA against the changing environments, *Genetic and Evolutionary Computation Conf. Workshop (GECCO2001)*, pp.47-50 (2001).
- 5) Goldberg, D.E. and Smith, R.E.: Nonstationary function optimization using genetic algorithms with dominance and diploidy, *Proc. 2nd Int. Conf. on Genetic Algorithms*, Grefenstette, J.J. (Ed.), pp.59-68 (1987).
- 6) Ramsey, C.L. and Grefenstette, J.J.: Case-based initialization of genetic algorithms, *5th Conf. on Genetic Algorithms*, Forrest, S. (Ed.), pp.84-91, Morgan Kaufmann (1993).
- 7) Mori, N., Imanishi, S., Kita, H. and Nishikawa, Y.: Adaptation to Changing Environments by Means of the Memory Based Thermodynamical Genetic Algorithm, *Proc. 7th Int. Conf. on Genetic Algorithms*, pp.299-306 (1997).
- 8) Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems, *Proc. Congress on Evolutionary Computation CEC99* (1999).
- 9) Vavak, F., Jukes, K. and Fogarty, T.C.: Learning the Local Search Range for Genetic Optimization in Nonstationary Environment, *Proc. 1997 IEEE Int. Conf. on Evolutionary Computation*, pp.355-360 (1997).
- 10) Vavak, F. and Jukes, K.: Adaptive Combustion Balancing in Multiple Burner Boiler Us-

ing a Genetic Algorithm with Variable Range of Local Search, *7th Int. Conf. on Genetic Algorithm*, Bäck, T. (Ed.), Morgan Kaufmann (1997).

- 11) 相沢彰子：スキーマ処理に基づく集団型探索アルゴリズムの構成，電子情報通信学会論文誌 D-II, Jol. J78-D-II, No.1, pp.94-104 (1995).
- 12) Cobb, H.G.: An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments, *NRL Memorandum Report*, 6760 (1990).
- 13) Grefenstette, J.J.: Genetic Algorithms for changing environments, *Proc. 2nd Parallel Problem Solving from Nature*, pp.137-144 (1992).
- 14) Deb, K. and Goldberg, D.E.: An Investigation of Niche and Species Formation, *Proc. 3rd Int. Conf. on Genetic Algorithms* (1989).
- 15) Mori, N., Kita, H. and Nishikawa, Y.: Adaptation to a changing environment by means of the feedback thermodynamical genetic algorithm, *Parallel Problem Solving from Nature*, Eiben, et al. (Eds.), No.1498 in LNCS, Springer (1998).
- 16) 林 貴宏, 木村春彦, 広瀬貞樹：動的環境に対処するための遺伝的アルゴリズムの制御方法，電子情報通信学会論文誌 D-II, Vol.J85-D-II, No.4, pp.649-658 (2002).
- 17) Munetomo, M. and Goldberg, D.E.: Identifying linkage groups by nonlinearity/non-monotonicity detection, *Journal of Evolutionary Computation*, Vol.7, No.4, MIT Press (1999).
- 18) 森 一之, 築山 誠, 福田豊生：多様性を持つ免疫的アルゴリズムの提案と負荷割当て問題への応用，電気学会論文誌 C, Vol.113, No.10, pp.872-878 (1993).
- 19) Bosman, P.A.N. and Thierens, D.: Linkage Information Processing In Distribution Estimation Algorithms, *Proc. Genetic and Evolutionary Computation Conf.*, pp.60-67, Morgan Kaufmann (1999).

(平成 14 年 2 月 5 日受付)

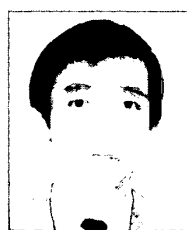
(平成 14 年 4 月 9 日再受付)

(平成 14 年 5 月 15 日採録)



林 貴宏

平成 10 年金沢大学工学部電気・情報工学科卒業。平成 12 年金沢大学大学院自然科学研究科電子情報システム専攻前期博士課程修了。同年同数理情報科学専攻後期博士課程入学。平成 13 年石川工業高等専門学校電子情報工学科助手。現在に至る。その間、人工知能、進化計算に関する研究に従事。IEEE, 電子情報通信学会, 人工知能学会各会員。



木村 春彦 (正会員)

昭和 49 年東京電機大学工学部応用理化学科卒業。昭和 54 年東北大学大学院博士課程 (情報工学専攻) 修了。工学博士。同年、富士通 (株) 入社。昭和 55 年金沢女子短期大学講師。昭和 56 年同短大助教授，昭和 59 年金沢大学経済学部助教授，平成 4 年同大学工学部電気・情報工学科助教授，平成 6 年同学科教授，平成 11 年同学部情報システム工学科教授，平成 13 年同大学大学院自然科学研究科教授，現在に至る。その間、最適コード変換，推論の高速化の研究に従事。電子情報通信学会，人工知能学会各会員。



白山 政敏

昭和 42 年富山大学工学部生産機械学科卒業。昭和 45 年富山大学大学院工学研究科生産機械工学専攻修士課程修了，工学博士 (大阪大学)，昭和 58 年富山工業高等専門学校助教授，平成元年石川工業高等専門学校教授。現在に至る。その間，ランダム表面形状の ARMA モデル解析，ファジィ制御やニューラルネットワークによる視覚移動ロボットの走行制御の研究に従事。電子情報通信学会，ファジィ学会，精密工学会各会員。