

進化的計算手法における効率的な解探索についての分析

Analysis of the Efficient Solution Search for Evolutional Computation Methods

今福 啓^{*1}

Kei Imafuku

e-mail:k03082@dokkyo.ac.jp

keywords: 進化的計算手法、個体学習、集団進化

近年、組み合わせ最適化やコンピュータプログラムの自動生成手法を目的として、さまざまな進化的計算手法が提案されている。しかし、それらの手法でどのようにして望ましい解を得られるのかについての考察はあまり行われていない。本研究では、進化的計算手法において解を効率的に探索するために必要となる点を、進化的計算手法で一般的に解かれる問題を単純化したモデルを用いたコンピュータシミュレーションにより明らかにした。得られた結果から、個体の多様化には形質の学習を制御することが有効であること、望ましい解探索には形質の多くを一度に狭い範囲で探索するか、あるいは同時に変化させることのできる形質を適度に持ち、各形質を適度な範囲で変化させることが必要であることを明らかにした。

Recently, various evolutional computation methods are proposed for the purpose of solving combinatorial optimization or creating computer programs automatically. However, it is not apparent how to obtain the desired solution efficiently. The purpose of this paper is to clarify what is required to obtain the solution efficiently. From the simulation results, we disclosed that it is effective to control the learning of the traits to keep the diversification of the individual. Furthermore, to obtain the desirable solution, we clarified that it is required to search the solution by changing the elements of the traits widely in the small range, or to have several traits that can be changed in the moderate range.

*1: 獨協大学 経済学部

1. はじめに

遺伝的アルゴリズムをはじめとする多くの進化的計算手法は、組み合わせ最適化問題やコンピュータプログラムの自動生成といった、解を得ることが難しい問題に広く応用され有効性が示されている。これらの手法では、生物の個体が学習する過程と、集団の世代交代による進化からなる過程を問題解決の手法として取り入れている。解の探索において、それらの過程がどのような効果をもたらしているのかについては、遺伝的アルゴリズムではスキーマ定理⁽¹⁾のように数理的な分析を行ったものや、選択方法や交叉の違いがもたらす効果を分析した研究⁽²⁾⁽³⁾がおこなわれている。

具体的な進化的計算手法として、近年、粒子群最適化法⁽⁴⁾やアントコロニー最適化法⁽⁵⁾、遺伝的ネットワークプログラミング⁽⁶⁾、GRAPE⁽⁷⁾といったさまざまな手法が提案されている。しかし、それらの手法において、何が良い解の獲得につながるのかという点についての分析はあまり進められておらず、明確になっていないといえる。

進化的計算手法において、何が集団の進化を促進するのかを分析した研究として、文献⁽⁸⁾がある。この文献では、生物の遺伝子とその良し悪しから適応度地形を作り、適応度が上昇するごとに高い適応度の獲得が難しくなる問題をモデル化している。そしてモデルを用いたコンピュータシミュレーションで局所探索をおこない、生物が学習によって獲得した形質が次第に先天的に生まれ持つ形質に変化していく Baldwin 効果⁽⁹⁾が生じることを示し、個体の学習と集団の進化が相互におよぼす関係を明らかにしている。

生物では、個体の生涯の学習が地形上の近傍を局所探索することに相当する⁽⁸⁾。そのため、生物の学習を表現するには、局所探索をおこなった計算で十分といえる。しかし、コンピュータ内で実行する進化的計算手法では、生物とは異なり必ずしも局所探索に限定する必要はない。たとえば遺伝的アルゴリズムでは、解を表現する個体に交叉や突然変異といった操作がおこなわれる。交叉では一つの個体内にある複数の要素が一度に別の要素に置き換えることから、個体が表現する解を中心として大域的な探索を行うことに等しい。それに対して突然変異は、個体のもつ一つの要素を別の要素に置き換えるため、局所的な探索の実行に相当する。このように、進化的計算手法では必ずしも生物と同じ探索のみを行っているわけではない。また、進化的計算手法でプログラムを自動生成するには、多様な表現をもつ個体が必要であるとの議論もおこなわれている⁽⁷⁾。そのため、進化的計算手法でより良い解を得るには、解探索の効率化だけではなく、解を表現する個体の多様性を維持することが重要である。

本研究では、進化的計算手法で効率良く解を求める際に必要となる探索をどのようにおこなえばよいのか、個体の多様性を維持するためには何が効果的であるのかを明らかにする。そこで、文献⁽⁸⁾のモデルを局所探索から大域的な探索まで可能となるよう修正し、それを用いてコンピュータシミュレーションを実行する。シミュレーションは、学習と進化を制御するためのパラ

メータと解の探索範囲をさまざまに変更して実行し、得られた結果からどのようなパラメータ設定が効率的な解の獲得につながるのかを示す。

2. モデル

本研究では、文献⁽⁸⁾で提案されているモデルに変更をくわえたものを用いる。文献⁽⁸⁾のモデルでは、複数の個体からなる集団が、与えられた環境において適応度を高めるために学習と進化をくり返す。1つの個体は、 M 個の形質 $t_i (i = 0, \dots, M - 1)$ 、遺伝子 g_i 、可塑性（学習の可否）を決める遺伝子 p_i から構成される。 t_i および g_i は初期値を1として、1から M までの整数値をとる。 p_i は0か1のいずれかの値をもち、1の場合のみ t_i が学習する。このように、学習可能な形質を制御する点がモデルの特徴となっている。そして、個体を N 個集めた集団により、モデルは構築される。

1つの個体は、図1のような構造となっている。

図1は、 $M = 8$ とした場合の具体例である。1世代からはじまる各世代において、形質 t_i は遺伝子 g_i を初期値として学習を開始する。そして、決められた回数 L だけ次式にしたがい学習する。文献⁽⁸⁾のモデルでは、 g_i のうち学習できるのは可塑性を決める遺伝子 p_i が1となっている形質のみで、 p_i が0の形質は学習しない。ただし本研究では、 p_i によらず全形質が学習できるシミュレーションも実行し、 p_i の有無が結果にあたる影響についての分析をおこなう。

形質 t_i は、次式により学習する。

$$t_i = \begin{cases} g_i + \text{rand}(\Delta T_i) & \text{if } p_i = 1 \\ g_i & \text{otherwise} \end{cases} \quad (1)$$

$\text{rand}(\Delta T_i)$ は、 $-\Delta T_i$ から $+\Delta T_i$ の範囲の整数をランダムに発生する関数である。文献⁽⁸⁾では、式(1)の学習は全ての個体において $\Delta T_i = 1$ に固定されているが、本論文では個体ごとに異なり、 $1 \leq \Delta T_i \leq \Delta T_{MAX}$ ($1 \leq \Delta T_{MAX} \leq M$)の範囲で変更できるよう変更をくわえた。また文献⁽¹⁾では、1回の学習時に $t_i (i = 1, \dots, M)$ のうちいずれか1つのみが学習を行うよう制御しているが、本論文では個体ごとに異なる T_{NUM_i} 個 ($1 \leq T_{NUM_i} \leq T_{NUMMAX}$, $T_{NUMMAX} \leq M$)の t_i が同時に学習できるよう変更した。

各個体は1回の学習を行うごとに、獲得した形質 t_i の適応度を次式により計算する。そして L 回の学習後、各個体の適応度は、学習を通じて得られた適応度の最大値とする。

$$\text{fitness} = \text{argmax } f(n) \quad (2)$$

$$f(n) = \begin{cases} n & \text{if } \text{num}(n) \geq n \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

式(3)において、 $\text{num}(n)$ は、 t_i の値が n 以上の総数である。図1の例では t_1, t_2, t_5, t_6, t_7 が4となっており、式(3)において $\text{num}(4) = 5 > 4$ となることから、適応度は4となる。

この適応度関数では、適応度が高くなるほど多くの形質が同一の高い値とならなければ、より高い適応度を得ることができない。たとえばある時点で4つ以上

の形質が 4 となり、適応度が 4 となったとする。この状態から適応度が 5 となるためには、多くの形質が変化できる場合、その値を同時に 5 に変化させる必要がある。少ない形質しか変化できない場合は、すでに得ている形質の 4 をいったん捨てて適応度を下げなければならない。そのため、低い適応度にくらべて高い適応度は獲得しづらくなっている。このような状況は、進化的計算手法が用いられるナップサック問題や巡回セールスマン問題といったクラス NP の問題を解く際にも生じることであり、それを単純化したものとなっている。

各個体が L 回の学習を行った後、集団の中から適応度の高いものを選択し、それを次世代に残す世代交代をおこなう。本研究では、適応度の値に応じたルーレット選択により、重複を許して N 個の個体を選択し、それを次世代の新しい個体とする。また世代交代の際、選ばれた個体の $g_i, p_i, \Delta T_i, T_{NUM_i}$ を、それぞれ確率 $p_{mg}, p_{mp}, p_{m\Delta T}, p_{mTNUM}$ で突然変異させる。突然変異の際、 g_i には +1 または -1 のいずれかをランダムに加え、 p_i は現在の値が 0 ならば 1 に、1 ならば 0 に変更する。 $\Delta T_i, T_{NUM_i}$ は、シミュレーション実行時に設定する $\Delta T_{MAX}, T_{NUMMAX}$ を使い、それぞれ $1 \leq \Delta T_i \leq T_{MAX}, 1 \leq T_{NUM_i} \leq T_{NUMMAX}$ の範囲で別の整数に変更する。

i	0	1	2	3	4	5	6	7
p_i	0	1	1	0	0	1	1	1
g_i	3	2	4	1	5	3	1	3
t_i	3	4	4	1	5	4	4	4

図 1 個体の構造 ($M = 8$ の場合)

3. シミュレーション

2 章のモデルを用いてシミュレーションを行った。1 回のシミュレーションにおけるパラメータの値は、1 世代分の学習回数 $L = 100$ 、世代交代を 16000 回として、個体数 $N = 400$ 、1 個体の形質数 $M = 12$ とした。また、 $g_i, p_i, \Delta T_i, T_{NUM_i}$ を突然変異させる確率は、それぞれ $p_{mg} = 0.002$, $p_{mp} = 0.001$, $p_{m\Delta T} = 0.002$, $p_{mTNUM} = 0.002$ とした。 $\Delta T_{MAX}, T_{NUMMAX}$ はシミュレーションごとに設定を変更し、値の違いが結果にどのような違いをもたらすのかを考察する。そして p_i の値に応じて学習できる形質を制御する場合と、全形質が学習できる場合の結果を比較する。なお、以下の結果では $p_i = 1$ の形質のみ学習した結果を「制御あり」、全形質が学習した結果を「制御なし」とよぶ。

以下の結果は、各パラメータでのシミュレーションを 10 回くり返し、その中で最終世代の適応度が最大となった結果である。シミュレーションは乱数を発生しておこなっているため、常に同一の結果が生じるわけではないが、いずれのシミュレーションでも結果の傾向は同様となる。また、図 30 および図 31 で考察している形質の多様性の傾向には違いはない。

3.1 $\Delta T_{MAX} = 1, T_{NUMMAX} = 1$ の結果

最初に、文献⁸⁾と同じく $\Delta T_{MAX} = 1, T_{NUMMAX} = 1$ として、 p_i の値に応じて学習できる形質を制御するシミュレーションを実行した。 $\Delta T_i, T_{NUM_i}$ は常に 1 となるため、文献⁸⁾と全く同じ条件である。また、1 回の学習時に 1 つの t_i を 1 ずつしか変更しないため、この条件は局所的な探索を行っていることに相当する。

図 2 の上段において、全個体で学習前の適応度を計算してその平均値を世代ごとに示したものが「学習前の適応度平均値」である。「学習後の適応度平均値」は、 L 回の学習後に学習前の適応度平均値と同じく平均値の計算を行った結果である。図 2 の下段では、全ての個体で全形質 M のうち学習できる形質 p_i の割合を計算し、その平均値を世代ごとに「学習できる形質の割合」として図示している。「変化した形質の割合」は、全個体で学習開始前の t_i と L 回の学習終了時の t_i を比較し、異なっている割合を計算して、その平均値を世代ごとに示したものである。

図 3 は、全形質が学習できるとした以外、図 2 と同じ条件でシミュレーションを行った結果である。図 3 の結果では p_i によらず全形質が学習できるため、図の下段に学習できる形質の割合を表示していない¹⁾。

どちらの結果も、世代が進むにつれて学習後の適応度が 1 程度ずつ上昇し、最終的には適応度が 8 となっている。しかし変化した形質の割合を比べると、図 2 では 6000~7000 世代にみられるように、適応度が上昇し始めてから上昇が終了するまでの間に形質の変化が上昇するのに対し、図 3 では 5000~6500 世代のように適応度が上昇する前に多くの形質が変化し、上昇後は変化した形質の割合が急に減少している。また、図 2 では全ての世代において学習できる形質の割合が 0.5 程度となっており、半数の形質を変化させる状況を維持することが望ましいことがわかる。

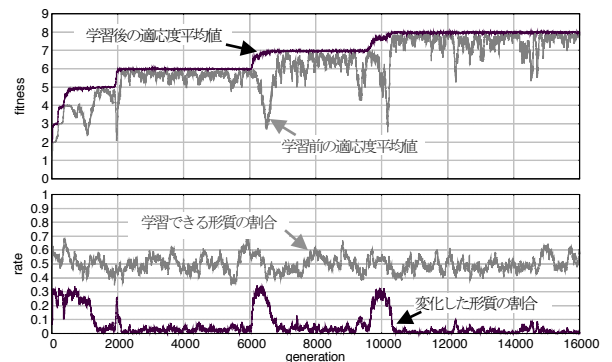


図 2 $\Delta T_{MAX} = 1, T_{NUMMAX} = 1$ (制御あり)

¹⁾ 他の (番号の末尾に a が付いている) 全形質が学習できる結果でも同じく、学習できる形質の割合は表示していない。

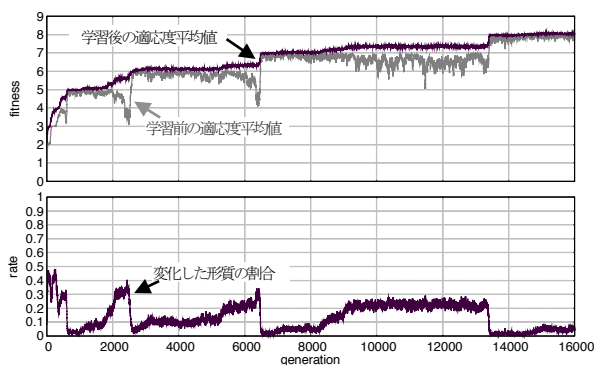


図 3 $\Delta T_{MAX} = 1, T_{NUMMAX} = 1$ (制御なし)

3.2 $\Delta T_{MAX} = 3, 5, 12, T_{NUMMAX} = 1$ の結果

$T_{NUMMAX} = 1$ に固定し、 ΔT_{MAX} を3, 5, 12と変更してシミュレーションを行った。3.1節と比較すると1回の学習時に1つの t_i しか変化しないが、上昇する t_i の値が大きいため、1回の学習で適応度が大幅に上昇することが期待できる。

図 4~図 6 は、 p_i の値に応じて学習できる形質を制御した結果である。図 4 では約 4000 世代まで、図 5 では約 8000 世代まで、図 6 では約 7000 世代まで、学習できる形質と変化した形質の割合が高いまま維持され、適応度も学習前と比較して学習後に大きく上昇することがみてとれる。そして、どの結果でも図 2 と同様に、世代が進むと適応度が上昇しはじめてから終了するまでの間にかけて、変化する形質の割合が高くなっている。最終的な学習後の適応度は図 4~図 6 のそれぞれで8, 7, 7となり、 ΔT_{MAX} が大きくなると減少する。

$\Delta T_{MAX} = 3$ における ΔT_i の平均値は図 7 のようになった。世代とともに大きく変動している。これは、適応度の高い個体の ΔT_i が特定の値に収束しないためと考えられる。他の ΔT_{MAX} でも同様となったことから、以下では結果を省略する。

図 8~図 10 は、図 4~図 6 と同じ条件としたときの、全形質が学習できる場合の結果である。図 8 では、図 4~図 6 と同様に約 6000 世代まで変化した形質の割合が高いまま維持されている。そして世代が進むにつれて形質が進化し、最終的な適応度は 8 となっている。しかし図 9, 図 10 では形質が進化せず、変化した形質の割合も 0.8 程度と高いままであり、どの世代でも学習のみで多くの形質を変化させて適応度を上げていることがわかる。適応度は初期世代から上昇せず、5 を少々上回る程度である。 ΔT_i の平均値は、 $\Delta T_{MAX} = 3$ の場合には図 7 と同様に大きく変動したが、 $\Delta T_{MAX} = 5, 12$ の場合にはそれぞれ 3 世代、40 世代で 5 に収束した。これは、全形質が学習する結果の全てで同様であった。また適応度も、 $\Delta T_{MAX} \geq 5$ では図 9, 図 10 と同様に初期世代から進化しなくなった。そのため、以下では $\Delta T_{MAX} \geq 5$ の結果は省略する。

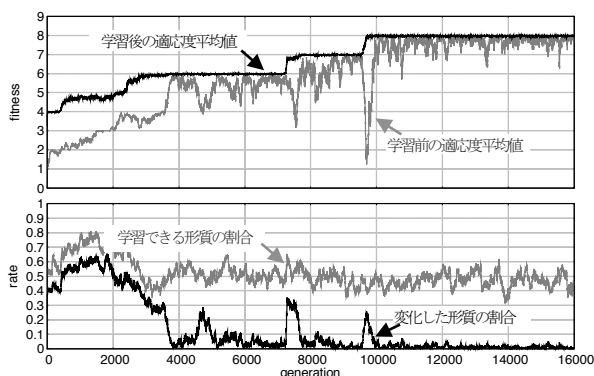


図 4 $\Delta T_{MAX} = 3, T_{NUMMAX} = 1$ (制御あり)

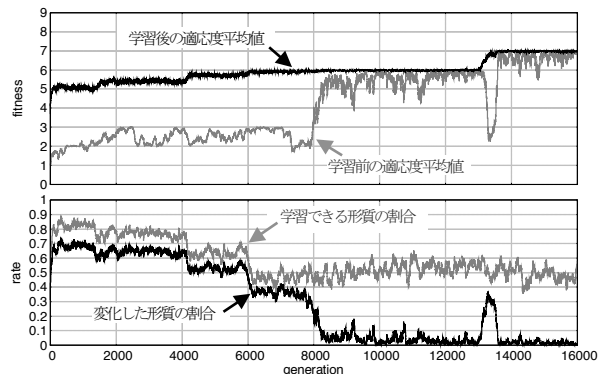


図 5 $\Delta T_{MAX} = 5, T_{NUMMAX} = 1$ (制御あり)

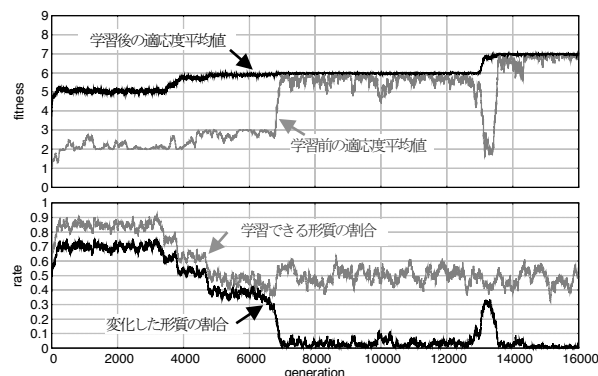


図 6 $\Delta T_{MAX} = 12, T_{NUMMAX} = 1$ (制御あり)

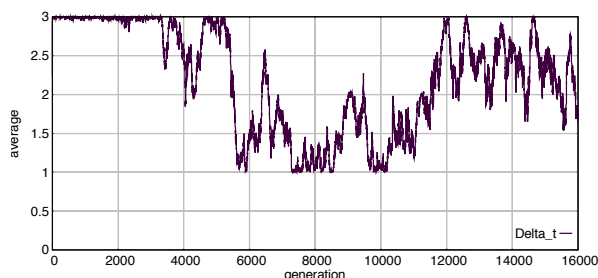
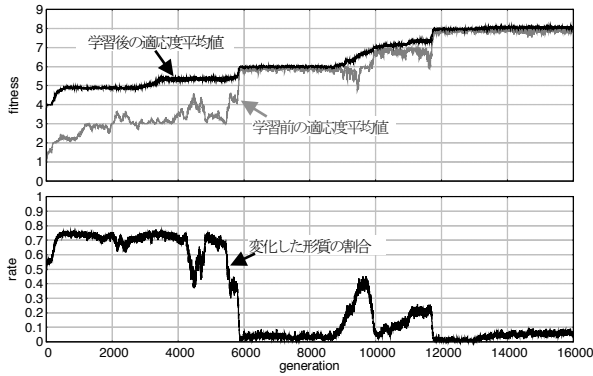
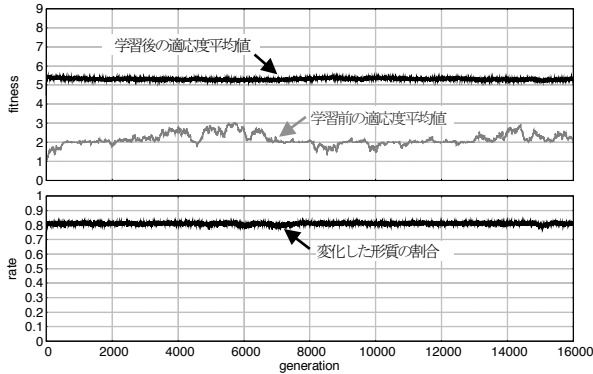
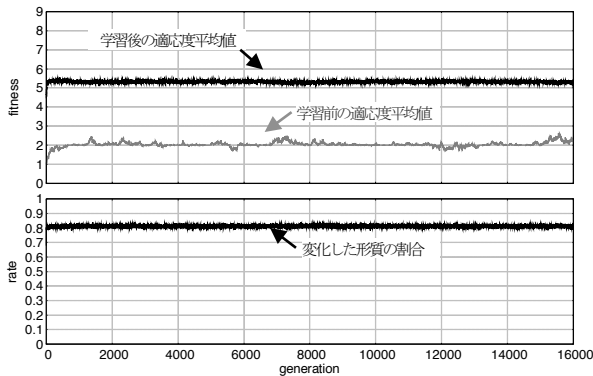


図 7 $\Delta T_{MAX} = 3, T_{NUMMAX} = 1, \Delta T_i$ の平均値 (制御あり)

図 8 $\Delta T_{MAX} = 3, T_{NUMMAX} = 1$ (制御なし)図 9 $\Delta T_{MAX} = 5, T_{NUMMAX} = 1$ (制御なし)図 10 $\Delta T_{MAX} = 12, T_{NUMMAX} = 1$ (制御なし)

3.3 $\Delta T_{MAX} = 1, T_{NUMMAX} = 3, 5, 12$ の結果

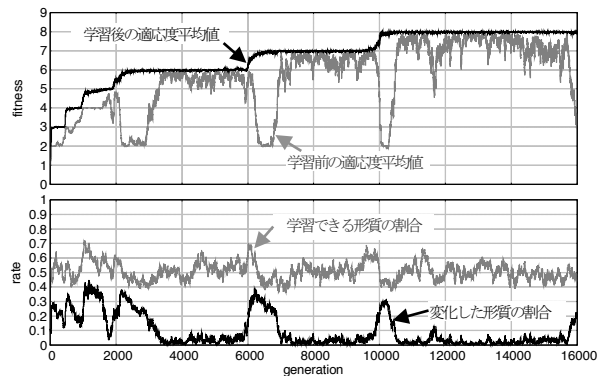
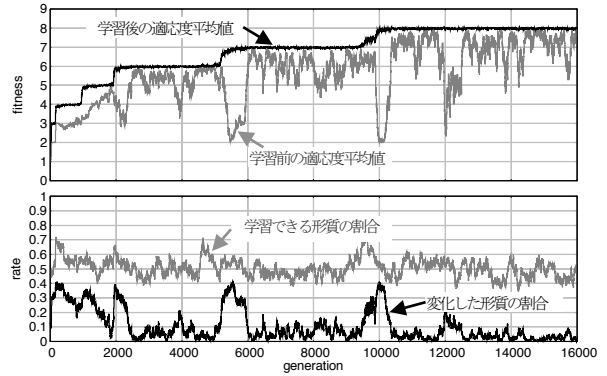
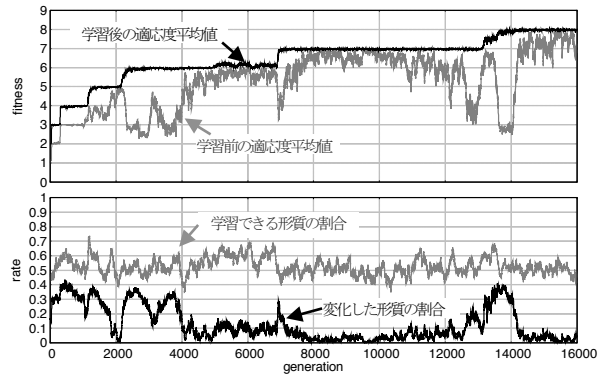
$\Delta T_{MAX} = 1$ に固定し、 T_{NUMMAX} をそれぞれ 3, 5, 12 と変更したシミュレーションを行った。3.1 節と比較すると、1 回の学習時に t_i は 1 ずつしか上昇しないが、同時に更新できる t_i の個数が増えている。これは、1 回の学習で複数の形質が変化し、形質が同時に一致することのできる状況となっている。

図 11~図 13 に、 p_i の値に応じて学習できる形質を制御した結果を示す。いずれの結果でも、すべての世代にわたって変化できる形質の割合が 0.5 程度を維持している。そして 6000~7000 世代では、学習後の適応度が上昇しはじめると学習前の適応度が一時的に減少し、その後学習前と学習後の適応度が一致する。これは、図 2 と同様の傾向である。

$T_{NUMMAX} = 3$ における T_{NUMMAX_i} の平均値は図 14

のようになった。 ΔT_i と同様に、世代とともに大きく変動している。 T_{NUMMAX} を他の値に変更した場合も同様の結果となったため、以下では結果を省略する。

図 15~図 17 に、図 11~図 13 と同じ条件で全形質が学習できる場合の結果を示す。 ΔT_{MAX} を大きくした場合と異なり、世代とともに学習前と学習後の適応度が上昇している。また、図 15 の 3000~4200 世代にみられるとおり、学習後の適応度が上昇する前から変化した形質の割合が上昇し、その後減少するという図 3 と同様の傾向をみてとれる。しかし図 17 の 14000 世代以降のように、学習後の適応度が一定値となった後も、変化した形質の割合が 0 に近い値とはならず、高い値が維持されている点は図 3 と異なる特徴である。最終的な学習後の適応度も図 15~図 17 では 8.5~9 程度となり、図 3 の適応度 8 を上回っている。

図 11 $\Delta T_{MAX} = 1, T_{NUMMAX} = 3$ (制御あり)図 12 $\Delta T_{MAX} = 1, T_{NUMMAX} = 5$ (制御あり)図 13 $\Delta T_{MAX} = 1, T_{NUMMAX} = 12$ (制御あり)

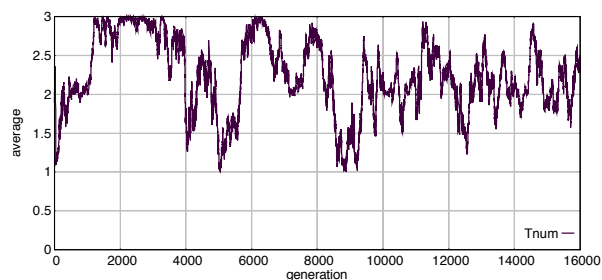


図 14 $\Delta T_{MAX} = 1, T_{NUMMAX} = 3, T_{NUM_i}$ の平均値 (制御あり)

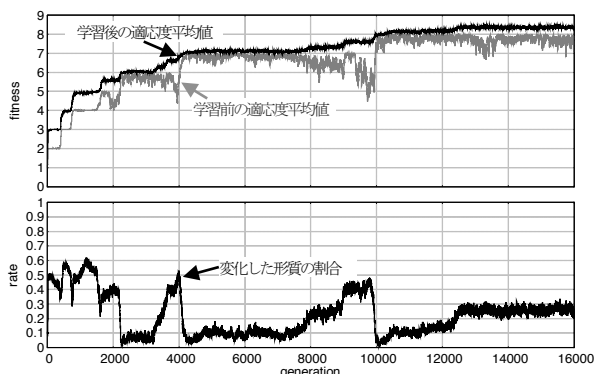


図 15 $\Delta T_{MAX} = 1, T_{NUMMAX} = 3$ (制御なし)

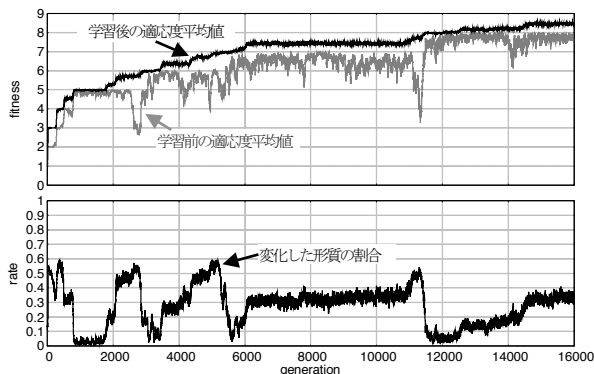


図 16 $\Delta T_{MAX} = 1, T_{NUMMAX} = 5$ (制御なし)

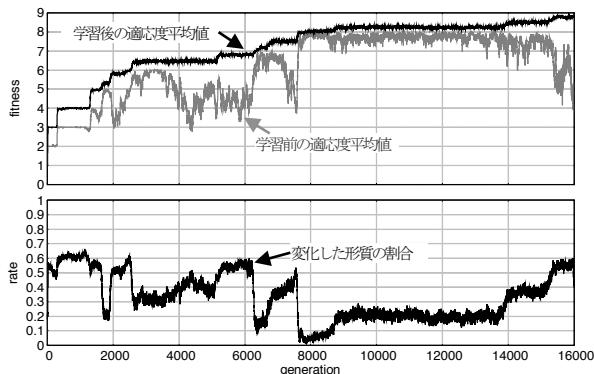


図 17 $\Delta T_{MAX} = 1, T_{NUMMAX} = 12$ (制御なし)

3.4 $\Delta T_{MAX} = 3, 5, 12, T_{NUMMAX} = 3, 5, 12$ の結果

ΔT_{MAX} と T_{NUMMAX} の両方をさまざまに変更したシミュレーションを実行した。この設定では 3.1~3.3 節と

比較して、より大域的な探索を行うことができる。

最初に、 p_i の値に応じて学習できる形質を制御した場合の結果を示す。図 18~図 20 は $\Delta T_{MAX} = 3$ に固定し、 $T_{NUMMAX} = 3, 5, 12$ と変化させた結果である。同様に、図 21~図 23 は $\Delta T_{MAX} = 5, T_{NUMMAX} = 3, 5, 12$ の結果、図 24~図 26 は $\Delta T_{MAX} = 12, T_{NUMMAX} = 3, 5, 12$ の結果である。図 18 の 0~3000 世代付近までに見られるとおり、いずれの結果においても、初期世代では学習できる形質と変化した形質の割合が 0.6~0.7 程度と高い値まで上昇している。そして世代が進むにつれて学習できる形質の割合は 0.5 前後に、変化した形質の割合は学習後の適応度が上昇した後を除いて 0 近くになっていることがみてとれる。これは、3.2 節で ΔT_{MAX} を 3, 5, 12 に変化させた場合と同様の結果である。

学習後の適応度は、 $\Delta T_{MAX} = 3, T_{NUMMAX} = 5, 12$ とした図 19、図 20 において、全ての結果中で最大の 9 となっている。しかし、 $T_{NUMMAX} = 5, 12$ のまま $\Delta T_{MAX} = 5$ に変更した図 22、図 23 と、 $\Delta T_{MAX} = 12$ に変更した図 25、図 26 では適応度が 7 に減少した。

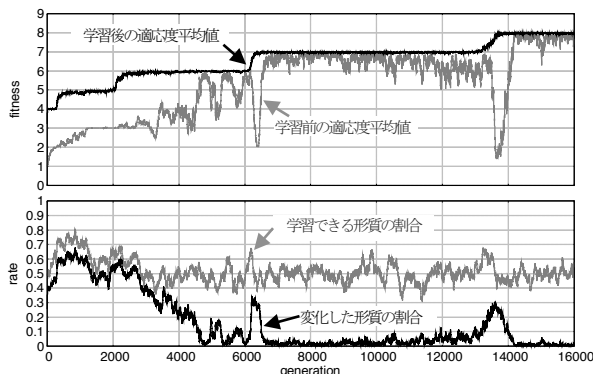


図 18 $\Delta T_{MAX} = 3, T_{NUMMAX} = 3$ (制御あり)

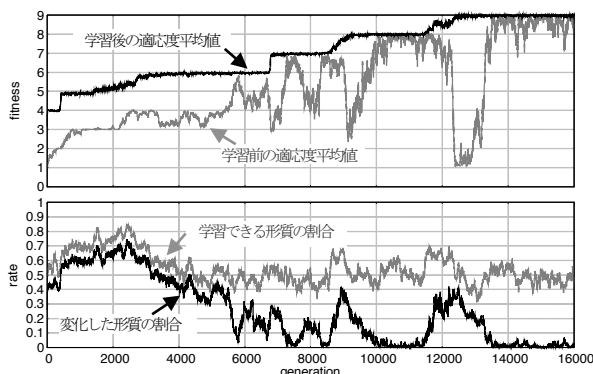
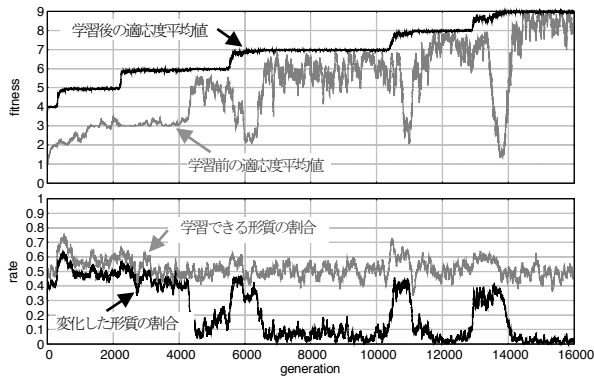
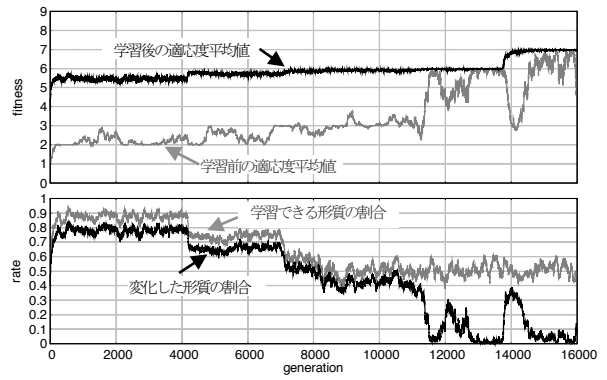
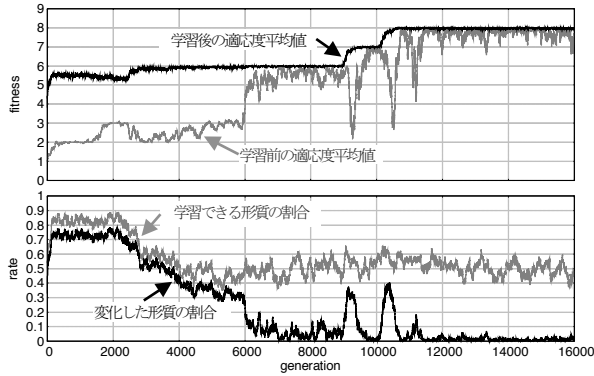
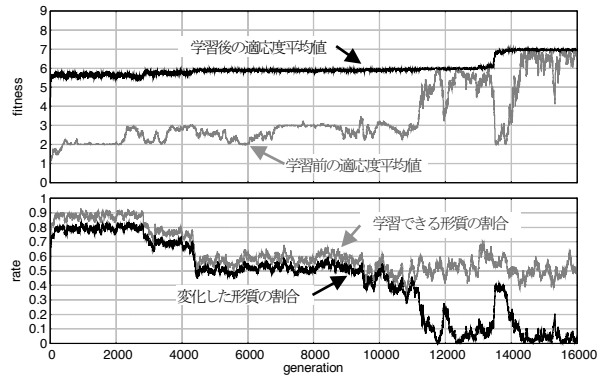
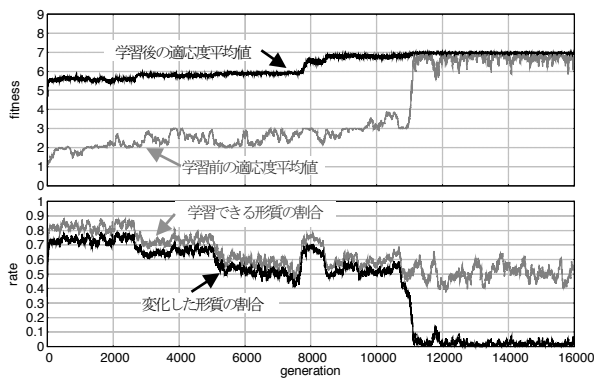
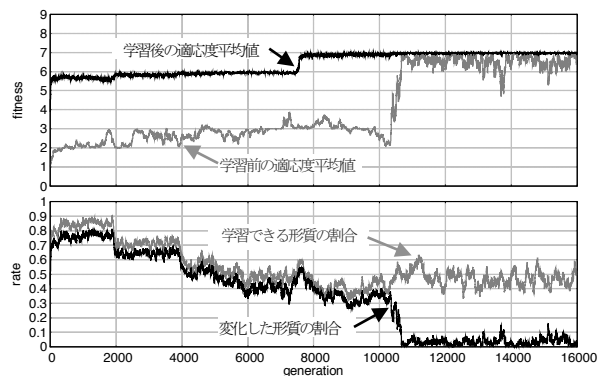
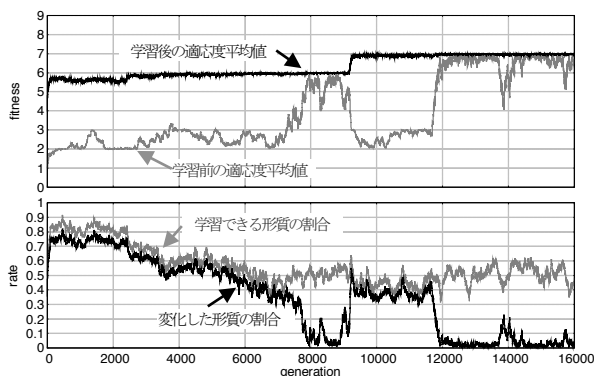


図 19 $\Delta T_{MAX} = 3, T_{NUMMAX} = 5$ (制御あり)

図 20 $\Delta T_{MAX} = 3, T_{NUMMAX} = 12$ (制御あり)図 24 $\Delta T_{MAX} = 12, T_{NUMMAX} = 3$ (制御あり)図 21 $\Delta T_{MAX} = 5, T_{NUMMAX} = 3$ (制御あり)図 25 $\Delta T_{MAX} = 12, T_{NUMMAX} = 5$ (制御あり)図 22 $\Delta T_{MAX} = 5, T_{NUMMAX} = 5$ (制御あり)図 26 $\Delta T_{MAX} = 12, T_{NUMMAX} = 12$ (制御あり)図 23 $\Delta T_{MAX} = 5, T_{NUMMAX} = 12$ (制御あり)

次に、全形質が学習できる場合の結果を示す。図 27～図 29 は、 $\Delta T_{MAX} = 3$ として、 T_{NUMMAX} を 3, 5, 12 と変化させた結果である。いずれの結果でも、5000～6000 世代までは変化した形質の割合が最大で 0.8 程度と高い状態が続いている。そして、図 27 の 7000～8000 世代にみられるように、図 3 の $\Delta T_{MAX} = 1, T_{NUMMAX} = 1$ の結果と同様に学習後の適応度が上昇する直前に変化した形質の割合が高くなり、適応度が一定の値となると形質の変化が急に減少している。最終的な学習後の適応度は、いずれの図でも 9 となっており、図 20 と同じく全ての結果の中で最も高い値となった。

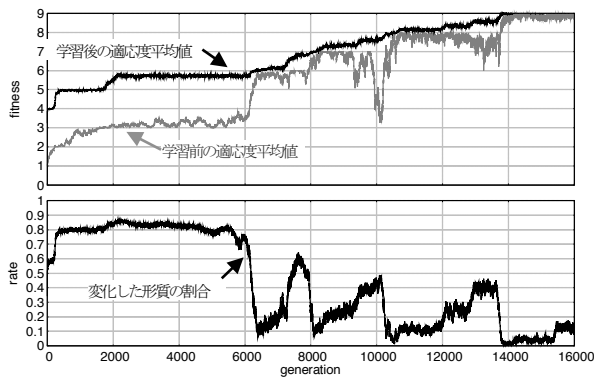


図 27 $\Delta T_{MAX} = 3, T_{NUMMAX} = 3$ (制御なし)

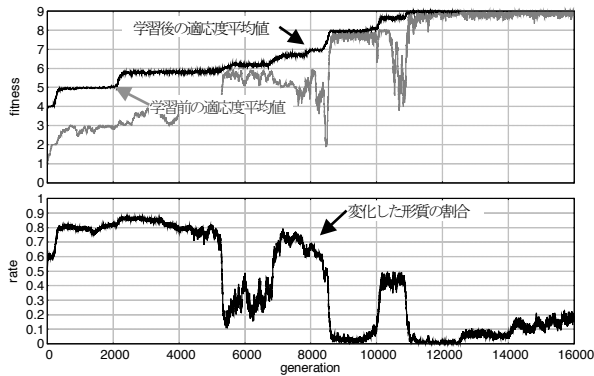


図 28 $\Delta T_{MAX} = 3, T_{NUMMAX} = 5$ (制御なし)

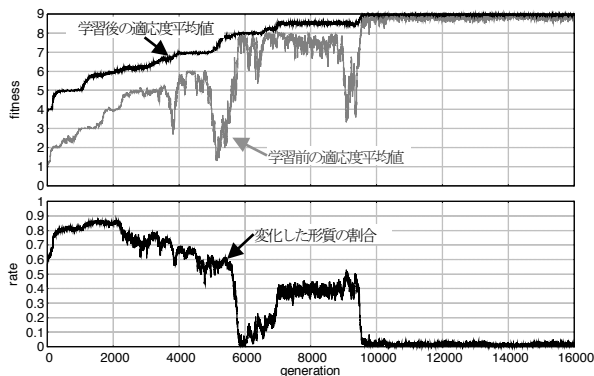


図 29 $\Delta T_{MAX} = 3, T_{NUMMAX} = 12$ (制御なし)

各シミュレーション結果で、適応度が最大値に近づき安定した世代と最大値を表 1 と表 2 にまとめた。適応度は世代ごとに異なり一定値に安定するわけではないため、適応度が安定した世代は、図から適応度が一定となった見受けられる期間に初めて 5 世代にわたって適応度の値の変化が 0.05 以内となった世代とした。 $p_i = 1$ の形質のみ学習する結果では、 $\Delta T_{MAX} = 3, T_{NUMMAX} = 5$ と $\Delta T_{MAX} = 3, T_{NUMMAX} = 12$ の結果で適応度が 9 を超えている。また全形質が学習する結果では、 $\Delta T_{MAX} = 3$ のすべての結果で適応度が 9 を超えていることがわかる。高い適応度を得るには、 $\Delta T_{MAX} = 3, T_{NUMMAX} = 5, 12$ が良い結果を生み出すといえる。

表 1 適応度が安定した世代と最大値 (制御あり)

ΔT_{MAX}	T_{NUMMAX}	安定した世代	最大値
1	1	10159	8.06
3	1	9686	8.0125
5	1	13666	7.055
12	1	13567	7.005
1	3	10097	8.0225
1	5	10006	8.02
1	12	13802	8.01
3	3	13731	8
3	5	12355	9.0125
3	12	13715	9
5	3	10513	8.0075
5	5	9028	7
5	12	9446	7
12	3	14141	7.0125
12	5	13746	7.0025
12	12	7718	7.0225

表 2 適応度が安定した世代と最大値 (制御なし)

ΔT_{MAX}	T_{NUMMAX}	安定した世代	最大値
1	1	13405	8.1375
3	1	11913	8.1425
5	1	11	5.5425
12	1	21	5.555
1	3	12399	8.5175
1	5	14552	8.5825
1	12	15378	8.8875
3	3	13746	9.2575
3	5	11174	9.3225
3	12	9655	9.02

3.5 考察

得られた結果から、形質の学習を制御の有無が結果に与える違い、 ΔT_{MAX} および T_{NUMMAX} が結果にあたる影響を考察する。

最初に、学習できる形質の制御を制御する場合と全形質が学習できる場合で、集団内で個体がどのように進化するかを検討し、多様性の維持に必要な要素を明確にする。ここでは図 2 と図 3 において、学習後の適応度が 6 から 7 に上昇する 5000~7000 世代を詳細に分析する。なお、以下で考察する点と同様の状況は、学習後の適応度が 7 から 8 に上昇する際にもみられた。

図 30 の上段は、学習を制御した場合の学習前と学習後の適応度変化を示したもので²、中段は学習前の個体もつ適応度の個体割合の変化、下段は学習後の個体もつ適応度の個体割合の変化を示している。5500 世代では、学習前には適応度 6 の個体が最も多くなっているが、適応度 3, 4 の個体も一定数だけ存在している。ただし、この世代ではどの個体も学習後は適応度 6 にしかならず、学習前の適応度が 6 の個体が最も多いことから、世代交代で学習前の適応度が 6 の個体が選択されることが多くなる。

² これは図 2 と全く同じである。

しかし 6000 世代になると、学習前の適応度が 4 の個体は 3 つしかないものの、学習後にはすべて適応度 7 となる。それに対して、学習前の適応度が 6 の個体は 338 あるが、学習後に適応度が 7 となる個体は 1 つしか存在しない。そのため、適応度 4 の個体が徐々に次世代に残るようになると同時に、突然変異により学習前の適応度が 2, 3 に減少した個体も発生している。

6100 世代では、学習前の適応度が 6 の個体数 338 のうち、約 56% の 181 個体が学習後に適応度 7 になるが、学習前の適応度が 3 の個体は、個体数 30 のうち約 63% の 19 個体が学習後には適応度 7 に達する。そのため、適応度 3 の個体のほうが適応度 6 の個体より次世代に残りやすくなり、徐々に数を増やしていく。学習前の適応度が 2 の個体も同様である。この状況が、6500 世代まで継続する。

6600 世代になると、突然変異により学習前も学習後も適応度が 7 の個体が出現する。この個体は学習後に適応度が必ず 7 以上になるが、学習前の適応度が 6 以下の個体の中には学習後に適応度 7 にならなかった個体が存在している。よって、学習前の適応度が 7 の個体が徐々に集団を占めるようになり、6700 世代以降はその数が最大となる。しかし、7000 世代をみると学習前の適応度が 1, 2, 3 の個体のうち、学習後に適応度 7 に到達する割合はそれぞれ 100%, 83%, 100% となっており、これらの個体も消滅することなく、少数ながら後の世代に残っていく。

このように、さまざまな個体が集団中に多様に残るという特徴を、個体の学習を制御する結果ではみることができる。学習を制御する場合には、個体ごとに t_i だけでなく p_i も異なる値をもつことができる。そのため、学習後の適応度が同じ値となるとしても、学習前には多様な要素をもつ個体が混在することが可能である。それが、集団内での個体の多様性を生み出すことにつながったといえる。

全形質が学習できる場合の結果を検討する。図 31 は、図 30 と同様に学習後の適応度が 6 から 7 に上昇する 5000~7000 世代における適応度の変化と学習前、学習後での適応度の個体割合の変化を示したものである。

5200 世代では、学習前の適応度が 6 の個体数は 379 と全体の約 95% であるが、学習後に適応度 7 に到達するのはそのうちの約 15% である。この世代を境目に、学習後の適応度が 6 の割合が減少すると同時に、適応度 7 の個体が増加しはじめる。実際 6000 世代では、学習後に適応度 7 に達する割合は 28% まで増加する。

6000~6400 世代にかけては、学習前の適応度が 6 の個体が数を減少させ、代わりに適応度 3 の個体が数を増やしている。学習前の適応度が 3 の個体のうち、学習後に適応度 7 に達する個体の割合が 6000 世代では 0% であるが、6100 世代では 5%、6400 世代では 54% と大きく増加させている。それに対し、学習前に適応度 6 となる個体は 6100 世代で 28%、6400 世代では 44% と増加の速度が劣っている。それが影響し、学習前の適応度が 3 と 6 の個体数が変化している。また、6400 世代には学習前の適応度が 7 の個体が出現しているが、この個体は 6500 世代になると、全体の 94% にあたる 375

の個体を占めると同時に、他の個体はほとんど集団内に存在しなくなる。学習できる形質を制御した場合と同様に、最初から高い適応度をもつ個体のほうが、学習で高い適応度を得る個体よりも生存に有利なためである。全形質が学習する場合は、学習後の適応度が上昇した後は、このようにして個体の多様性が急速に失われる。

全形質が学習する場合には、 t_i のみが適応度の計算に影響する。そのため、ある世代において学習後に適応度が最大となる形質は t_i に依存し、多様な形質を持ちづらいついえる。また適応度を上げるには、その世代で学習後に最大となる適応度をもつ個体が、さらなる突然変異を経て適応度を上げる形質を獲得することが必要となる。

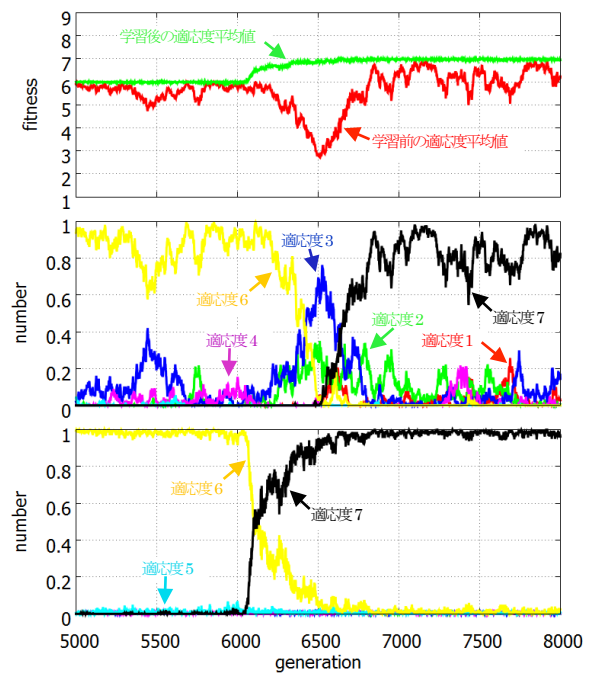


図 30 適応度の変化と個体数 (制御あり)

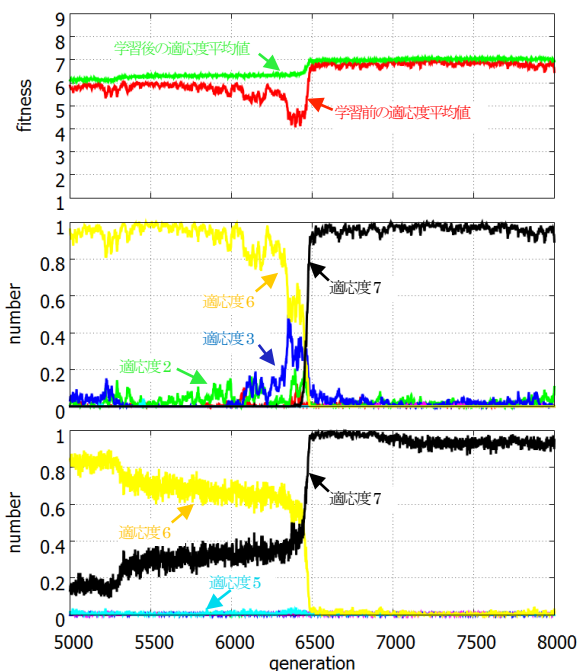


図 31 適応度の変化と個体数 (制御なし)

ΔT_{MAX} が解探索にあたる影響を検討する。全形質が学習する結果では、 $T_{NUMMAX} = 1, \Delta T_{MAX} = 3$ とした図 8 で世代とともに学習後の適応度が上昇し、最終的に適応度 8 となる。しかし $\Delta T_{MAX} = 5, 12$ とした図 9、図 10 では、世代が進んでも初期の適応度 5.5 程度から全く進化していない。また、学習前の適応度はいずれも 1.5 程度と等しいが、 $\Delta T_{MAX} = 3$ での学習後の適応度は 4 であるのに対し、 $\Delta T_{MAX} = 5, 12$ では 5.5 と高くなっている。これは、 ΔT_{MAX} を大きくすると学習の影響が強くなり、形質を進化させられずに学習のみで高い適応度を獲得するためと考えられる。1 つの t_i のみを大きく変更しても、初期の形質から到達しうる局所解を探すだけとなることがわかる。

図 4~図 6 は、 $p_i = 1$ の形質のみ学習する場合に $T_{NUMMAX} = 1, \Delta T_{MAX} = 3, 5, 12$ とした結果であるが、世代が進むと適応度も上昇しており、全形質が学習する場合のように学習の効果は強く出ている。これは、学習を制御したことで探索範囲が限定されるためと考えられる。しかし、図 4 では最終的な適応度が 8 であるのに対し、 ΔT_{MAX} を大きくした他の 2 つでは 7 と減少しており、1 つの形質のみを大きく変更することは効率的な解の探索に貢献していない。

ΔT_{MAX} にくわえて、 T_{NUMMAX} を変更した際における解探索の性能を検討する。 $p_i = 1$ の形質のみ学習する場合の結果をみると、 $\Delta T_{MAX} = 1$ で $T_{NUMMAX} = 3, 5, 12$ と変更した図 11~図 13 では、最終的な適応度が 8 となっている。これを $\Delta T_{MAX} = 3$ に増やすと、図 18~図 20 でみられるように適応度は $T_{NUMMAX} = 3, 5, 12$ のそれぞれで 8, 8, 9 となり、等しいか上昇している。しかし、さらに ΔT_{MAX} を増やして 5, 12 とすると、図 21 が示す $\Delta T_{MAX} = 5, T_{NUMMAX} = 3$ の結果で適応度 8 となる以外は、図 22~図 25 のとおり 7 に減少する。これらの結果から、 $\Delta T_{MAX} = 3$ のように 1 つの形質を変化させる

範囲を適度に限定した上で、 T_{NUMMAX} を大きくとり同時に多くの形質を変化させることが効率的な解探索に効果的といえる。

図 15~図 17 は、全形質が学習する場合に $\Delta T_{MAX} = 1$ として T_{NUMMAX} を 3, 5, 12 と増やした結果であるが、最終的な学習後の適応度が 8.5, 8.5, 8.9 となっている。 ΔT_{MAX} を 3 に増やすと、図 27~図 29 の $T_{NUMMAX} = 3, 5, 12$ のいずれの結果でも最終的な適応度が 9 に増加する。しかし、 ΔT_{MAX} をそれ以上に増やすと、3.2 節でのべたとおり図 9、図 10 のように形質が進化せず適応度は減少する。そのため全形質が学習する場合も、1 つの形質の変化は $\Delta T_{MAX} = 3$ と限定し、同時に多くの形質を変化させることが探索には効果的といえる。

表 1 と表 2 を比較すると、同じパラメータ設定のほとんどで最終世代に獲得する適応度は同様であるが、 $T_{NUMMAX} = 1, \Delta T_{MAX} = 12$ の結果のみ両者が異なる。 $p_i = 1$ の形質のみ学習する図 13 では 8 となっているのに対し、全形質が学習する図 17 では適応度が 8.8 と、より高い適応度を獲得している。変化した形質の割合にも違いがみられ、図 13 では適応度がほぼ最大値となった 14000 世代以降の割合は 0 に近いのに対し、図 17 では 15000 世代以降も 0.6 と高い割合を維持している。これは、図 13 における個体は図 30 と同様に多様な個体から構成されているが、図 17 の個体は多くが学習前の適応度として 1 か 2 を持ち、学習後に適応度 9 または 10 を獲得しているためである。図 17 での形質の例を図 32 に示す。この形質の学習前における適応度は 2 である。もし形質 9 となっている箇所が学習時に同時に 10 に変化すれば、学習後の適応度は 10 となる。これが実現すると変化した形質の割合は $8/12 = 0.75$ となる。また、形質 10 の箇所のうち 1 つでも 9 に変化すれば、学習後の適応度は 9 となる。このときは、変化した形質の割合は $1/12 = 0.08$ である。このような変化を平均した結果が 0.6 程度となっていると予想される。

i	0	1	2	3	4	5	6	7	8	9	10	11
g_i	9	9	9	9	9	9	9	9	10	10	2	2

図 32 個体の例

以上の考察から、進化的計算手法で効率的に解探索をおこなうには $\Delta T_{MAX} = 3, T_{NUMMAX} = 5, 12$ として、同時に変化する形質数を適度にもち、かつ各形質が適度な大きさで変化させることが必要であるといえる。これは探索方向を限定しつつ、ある程度離れた範囲まで探索を実行することを意味している。加えて全形質が学習する場合は、 $\Delta T_{MAX} = 1, T_{NUMMAX} = 12$ として多くの形質を同時に局所的に変更することも、効率的な解の探索に有効である。これは、現在得られている暫定的な最適解を中心として、全方向に狭い範囲での探索をくり返すことである。また解の多様性を維持するには、 $p_i = 1$ の形質のみ学習することで学習を制御することが有効である。

4. おわりに

本研究では、進化的計算手法において望ましい解を得るために必要となる個体の多様性の維持と、どのように効率良く解探索を実行すればよいのかを明らかにした。そのため、進化的計算手法で一般的に解かれている問題を単純化した文献⁽⁸⁾のモデルを局所探索から大域的な探索まで可能となるよう修正し、それを用いたコンピュータシミュレーションを行った。得られた結果を分析することで、個体の多様性を維持するには、形質の学習を制御することが必要という知見が得られた。解探索では、1つの形質のみを大きく変化させると性能が低下すること、そしてより高い適応度を得るうえで多数の形質を一度に狭い範囲で変更するか、同時に変化できる形質数を適度にもち、かつ各形質が適度な範囲で変化することが必要であることを明らかにした。

参考文献

- (1) Holland, J. H.: *Adaptation in Natural and Artificial Systems*, MIT Press, Massachusetts (1992)
- (2) 小林重信: “実数型 GA のフロンティア”, 人工知能学会誌, Vol. 24, No. 1, pp.128-143 (2009)
- (3) 白川真一、矢田紀子、長尾智晴: “遺伝的プログラミングによる実数型 GA の性能差を協調する探索空間の生成”, 新価計算学会論文誌, Vol. 1, No. 1, pp.54-64 (2010)
- (4) J. Kennedy and R. Eberhart: *Particle Swarm Optimization*, Proceedings of IEEE International Conference on Neural Networks, IV, pp.1942-1948 (1995)
- (5) Dorigo, M. and Gambardella, L. M. *Ant Algorithms for Discrete Optimization*, Artificial Life, Vol.5, No. 2, pp.137-172 (1999)
- (6) S. Mabu, K. Hirasawa and J. Hu: *A graph-based evolutionary algorithm: Genetic Network Programming (GNP) and its extension using reinforcement learning*, Evolutionary Computation, Vol. 15, No. 3, pp. 369-398 (2007)
- (7) Shinichi Shirakawa, Shintaro Ogino, and Tomoharu Nagao: *Graph Structured Program Evolution*, Proceedings of the Genetic and Evolutionary Computation Conference 2007 (GECCO '07), Vol.2, pp.1686--1693, London, England, 7-11 July (2007)
- (8) 鈴木麗璽、有田隆也: “凸凹した適応度地形において繰り返し生じる Baldwin 効果”, 第21回人工知能学会全国大会論文集, 3E7-1, pp. 1-4 (2007)
- (9) Baldwin, J. M.: *A New Factor in Evolution*, American Naturalist, Vol. 30, pp.441-451 (1896)

(2014年9月30日受付)

(2014年12月3日採録)