進化型人工神経回路網の構造進化のための一手法 —二重倒立振子問題への適用*

大倉 和博[†]・岩波 悟史[‡]

An Approach to Topology and Weight Evolving Artificial Neural Networks and the Validation with the Double Pole Balancing Problems*

Kazuhiro OHKURA[†] and Satoshi IWANAMI[‡]

A novel approach to Topology and Weight Evolving Artificial Neural Networks (TWEANN) is presented. This comes from the following two considerations. (1) Artificial evolution is designed without recombination which mostly generates an offspring whose fitness value is considerably reduced than those of parents. Instead of recombination, two types of topological mutations are provided as genetic operations. One is for inserting a new neuron and the other is for generating a new synaptic connection. The both of them are designed carefully to retain the current evaluation value. (2) A new encoding technique such that a string is defined as a set of substrings called *operons* is introduced. This is based on our idea that each potential function should be encoded into a different part of genetic information. In order to validate our approach, computer simulations are conducted by using one of the standard reinforcement learning problems called the double pole balancing problems without velocity information. The results obtained are compared with the results by NEAT, which is recognized as one of the most powerful techniques in this problem domain. It is found that our proposed approach yields competitive results, especially when the problem is tuned to be difficult.

1. はじめに

生物脳の仕組みを模倣した情報処理機構として人工神経回路網 (Artificial Neural Networks: ANN) がある。この分野には、生物脳の仕組みの解明を目指す生理学的アプローチと、人工物における機械学習等を目的とする工学的アプローチがあるが、本論文では後者の立場をとる。

ANN には進化型計算を用いて学習を試みる研究領域があり、EANN (Evolving Artificial Neural Networks) とよばれている [14]。一般に、EANN ではシナプス結合荷重値のみを進化対象とし、ANN の性能を左右するその他の要素、とくに、ネットワーク構造 (隠れ層ノードの数・シナプス結合の形態) を固定とすることが多かつ

た。これは、(1) ネットワーク構造を複雑化、すなわち、ノード数・シナプス結合数を増加させるに従い、十分な探索が行われないまま解探索空間が止めどなく広がってしまう、(2) 逆に、必要以上にネットワーク構造を単純化させておくと入出力間の写像精度が低下してしまい所望の性能を獲得できない、という問題が存在するため、ネットワーク構造も同時に進化させることは非常に困難であると考えられていたからである。それゆえ、多くの場合、EANN の使用者が問題ごとにいく度かの予備実験を通してネットワーク構造を事前に決定していた。

そのため、この状況を打破すべく、結合荷重値だけでなく隠れ層ノード数やシナプス結合の形態をも進化対象とする研究が行われており (たとえば, [1,7,8,10,11,14]), TWEANN (Topology and Weight Evolving Artificial Neural Networks) とよばれている [12]。TWEANN では、(1) 交叉が有効に働きにくい [11]、(2) 構造が異なるもの同士に対して (無理に) 交叉させると神経回路網の構造が変化するとともに適応度が急激に低下して淘汰されることが非常に多くなり、結果として、タスクに適したネットワーク構造を進化的に得ることが難しい、(3) 初期構造をどう設定するか、といった結合荷重値のみの

* 原稿受付 2005 年 1 月 25 日

[†] 神戸大学 工学部 Faculty of Engineering, Kobe University; 1-1 Rokkodai-cho, Nada-Ku, Kobe 657-8501, JAPAN

[‡] 神戸大学 大学院 自然科学研究科 Graduate School of Science and Technology, Kobe University; 1-1 Rokkodai-cho, Nada-Ku, Kobe 657-8501, JAPAN

Key Words: evolutionary artificial neural networks, evolutionary robotics, double pole balancing problems.

進化とは異なる TWEANN 特有の問題も現れる。なお、同様に、ネットワーク構造の遺伝子型へのエンコード方式も現在においても決定的なものではなく、重要な課題である [2,3,14]。

これに対し、近年、NEAT (Neuro-Evolution of Augmenting Topologies) [13] とよばれる TWEANN の一手法が提案され多くの注目を集めている。NEAT では、前述の (1) に対して、交叉が有効に働く遺伝的エンコード手法を提唱し、(2) に対して、“種分化”によって構造が変化した個体を保護する、(3) に対して、中間層ノードや再帰シナプス結合のない最小構造のネットワークから進化的探索を開始させる、ことを特徴としている。そして、強化学習のベンチマーク問題である二重倒立振り子問題に適用し、EANN として傑出した成果をあげている。

しかし、我々が行った NEAT の追加実験によると、次章で示すようないくつかの深刻な問題が生じていることがわかっている。これは、交叉を主要な遺伝的操作とする方式では避けられない特徴ではないかと思われる。そのため、本研究では、交叉を用いずに突然変異のみ、すなわち、一個体上でのみ遺伝的操作が施される構造進化手法 (Mutation-Based Evolving Artificial Neural Networks: MBEANN) を提案する。MBEANN は、実数で表現されるシナプス結合荷重値の突然変異だけでなく、構造変化による適応度低下をほとんど起こさない隠れ層ノード付加突然変異とシナプス結合付加突然変異を備えることを特徴とする。

以下は次のように構成される。2. では、NEAT の概略を説明するとともに、MBEANN を提案する動機となった NEAT の問題点について述べる。3. では、本論文で提案する構造進化手法である MBEANN について説明する。そして 4. で二重倒立振り子制御問題を取り上げ、計算機実験を通して NEAT との比較から、MBEANN の特徴とその有効性を検証する。最後に、まとめを示す。

2. NEAT

ここでは、TWEANN に関して非常に有効であると報告されている NEAT [12,13] について概略説明を行う。その際、新たに発生する問題点についても言及する。

2.1 遺伝子型へのエンコード手法と交叉

個体の遺伝情報は、ノードを表す情報とシナプス結合を表す情報をそれぞれ表す二本のリスト構造として保持されている。ノードを表す遺伝情報には、そのノード番号とそれが入力層・中間層・出力層のいずれに属するかが記載されている。ノード番号は、各試行において探索集団全体かつ全世代を通してユニークな番号であり、交叉の際のノード識別に用いられている。

各シナプス結合を表す遺伝情報には、そのシナプス結合の両端に接続しているノードへのポインタ、結合荷重

値、その結合が利用可能であるかどうかのフラグ、刷新番号の合計五つの情報が保持されている。ここで、刷新番号とは、ノード番号と同様に、各シナプス結合の識別に使われる番号のことであり、各試行において探索集団全体かつ全世代を通してユニークな番号である。この刷新番号は、集団中の個体で新しいシナプス結合が生成されるたびにインクリメントされて新しい番号が付与される。これは、交叉を用いて子孫を生成するときに、両親の遺伝情報を比較して同じ刷新番号を持つ部位の重複を避けるのに使用される。

このような工夫により、子孫は両方の両親に共通な遺伝情報を重複もしくは欠失せずに受け継ぐことができる。しかし、NEAT における交叉では子孫の遺伝情報は親のそれと比較して大きくなる傾向にある。また、親個体同士のネットワーク構造の差異が大きくなるに従い、子孫のネットワーク構造は大きくなる傾向を持つ。もし、差が小さくても数世代のうちに急速に肥大化する傾向を示すようになる。

2.2 種分化

TWEANN では、ネットワーク構造が変化すると適応度が急激に悪化することが多いが、NEAT での交叉における構造変化の場合もその例外ではない。そのため、(1) 新しく生成されたネットワーク構造を持つ適応度の低い子孫を自然淘汰から保護するとともに、(2) ネットワーク構造の差異がある程度大きな個体同士は交叉させない、ことを目的として“種分化”機構を採用している。ここで種分化とは、探索集団を個体間の類似度をもとに部分集団に分割し、それらを独立した種と考えて異種間の交叉を制限するとともに自然淘汰をそれぞれの種内に限定することにより、異なるネットワーク構造を持つ個体を保護しながら種内で進化させていく方式を取っている。

探索集団を種に分割するためには、個体間の類似度の指標となる適応性距離とよぶ値を用いる。この適応性距離が一定の閾値を越えると新たな種を生成し、それ以下ならば、適応性距離の最も小さい個体と同じ種として分類する。ここで、適応性距離は個体間で類似度を計測するもので、似ているほど小さくなり、似ていないほど大きくなる指標である。そして、*explicit fitness sharing* [4] を用いて、個体の適応度を同種内の個体数で割って算出しておし、適応度の高い種に属する個体数が過度に増えないようにしている。

種分化は上記のような新しい構造を持つ個体の保護だけでなく、交叉を同種内の個体同士でのみ行うことで、交叉を行う親個体の構造差異があまり大きくならないようにしている。しかし、このためには種分化の基準となる適応性距離の閾値を小さくして細かく種に分ける必要があるが、これをあまりに小さくすると、新しい構造を持った個体を過剰に保護し、ほぼ確実に淘汰されずに生き残ってしまうという問題が生じる。

2.3 最小ネットワーク構造からの進化

直接エンコード手法を用いた TWEANN の多くでは、ランダムな形態で初期個体を生成することが多い。しかし、NEAT では、隠れ層を持たない入力層と出力層をシナプス結合で接続した最小のネットワーク構造から進化的探索を開始する。これは、遺伝情報が最も少ない状態から進化が始まるため、十分長い計算世代にわたり低い次元の探索空間でシナプス結合荷重値の探索を十分に行わせてから次第に次元を高くしていく方法が、結果的に効率的な探索ができるためと説明されている。また、集団をネットワーク構造も含めてランダムに初期化するという従来よく使用された方式よりも、良い結果を生成することも NEAT の予備実験からわかっている [12]。

3. 提案手法：MBEANN

TWEANN において非常に効率的とされる NEAT の交叉にも前章で述べたような深刻な問題点がある。また、これらを解消できるような方法も容易には見つかりそうにもない。そこで、本論文では交叉を用いずに突然変異のみで行う手法 (MBEANN) を提案する。MBEANN では、ネットワーク構造進化のために、その適用によって適応度低下がほとんど起きないように配慮した 2 種類の構造突然変異を用いる。そのため、新しい構造を持つ適応度の低い個体の保護を目的として種分化を用いる必要はなくなるため、NEAT のように種分化は使用しない。

3.1 遺伝型へのエンコーディング法

TWEANN では、表現型となるネットワーク構造は、階層型人工神経回路網のように整然と表現されることはほとんどなく、再帰的な結合を含むなどして多様であるため、遺伝子型へのエンコーディング法も表現豊かである必要がある。しかし、意味のないもしくは意味をなしにくい表現を含んでしまうと探索空間が過度に広がって効率的な進化的探索が困難になる。これまでの TWEANN の研究では、エンコーディング法がシステムティックに研究されてきたわけではなく、おのおの独自の方法を提案するとともに、それに対し有効と思われる遺伝的操作を新規に設計することがほとんどであり [9,14]、いまだ標準となりうるようなエンコーディング方法はないといえる。

本研究でも、以下に示す考察のもと、人工神経回路網進化により適していると考える新しいエンコーディング法を提案する。まず、各個体が所持してほしい特性は何かについて考える。EANN において、ある程度集団が局所解に収束してくると、ネットワーク構造を変更する遺伝的操作は、ほとんどが大きな負の効果を及ぼすのみになってしまう傾向がある。それゆえ、遺伝的操作が効果的でなくなり十分な進化的探索が行われなくなることが多い。そこで、我々は、これが軽減さらには解消できるエンコーディング法が望ましいのではないかと考え

た。これに基づけば、遺伝的操作に基づく遺伝子型の変異によってもたらされる表現型 (ネットワーク構造) への影響は、表現型がもつ一部の機能に限定的に作用するだけであることが望ましい。すなわち、遺伝子型上の小さな変異は、表現型の上でも小さく局所的に影響を及ぼすのみであってほしい。したがって、逆に、各個体が獲得するであろう各機能は遺伝子型の上でもそれぞれにモジュール化されて表現されることが望ましいのではないかと、という着想に至った。

以上のことを踏まえ、本論文では Fig. 1 に示す遺伝子型を採用する。string は (1) 式に示すように、遺伝子座部分集合である operon で構成されている。operon を構成する遺伝子座集合は、(2) 式に示す通り、その operon 内に含まれるノードとそれに関連するシナプス結合から構成されている。各 operon は表現型において部分ネットワークを構成し、異なる operon に属するノード間を接続するシナプス結合は生成されないとする。そのため、上述したように operon ごとに独立した機能を持つことが期待される。

Fig. 1 Genotype

$$string = \{operon_1, operon_2, \dots, operon_l\} \quad (1)$$

$$operon_i = \{link_j | j \in O_i, node_k | k \in O_i\} \quad (2)$$

ここで、各 node は各 operon 内でユニークであり、ノード番号・ノード形式 (入力ノード/隠れ層ノード/出力ノードのいずれか) の値を持つ。また、link はシナプス結合を表しており、結合荷重値、(入力/出力) ノード番号、結合番号の値を持つ。結合荷重値 w は実数値がそのままおかれる。 l は string に含まれる operon の最大添字、 O_i は operon_{*i*} が含む遺伝子座の番号を要素とする集合である。

3.2 構造進化のための突然変異

初期構造は、NEAT にならい、隠れ層ノードを持たない入力層と出力層を直接結合した最小構造の個体で実験を開始し、世代が進むにつれて隠れ層ノード、シナプス結合が付加されていくことによって構造が進化していくものとする。参考文献 [12] と同じく、我々の予備実験からもランダムな構造を持つ初期集団よりも最小構造をもつ初期集団を与えた方が結果的に良いパフォーマンスが得られることを確認している。さらに、遺伝情報の部分

的削除は、適応度に与える影響が非常に大きく、総じて使用しない方がよいことも確認している。

初期個体を持つネットワーク構造を具体的に示せば、Fig. 2の左側のネットワーク構造のように、一つの *operon*₀ がすべての入力・出力ノードを含み、また、各入力ノードからすべての出力ノードへ直接シナプス結合があるものとなる。

構造進化操作を適用した際、多くの場合に適応度は低下する。NEATにおいては適応度の低下は黙認し、代わりに適応度が低下した個体が自然淘汰で失われないように、種分化によって新しい構造を持つ個体を保護していた。本研究では、構造進化操作による適応度への影響がない、もしくは非常に軽微となるように設計された、二つの新しい構造突然変異(ノード付加突然変異とシナプス結合付加突然変異)を用いて構造が進化していくとする。

3.2.1 ノード付加突然変異

ノード付加突然変異は、*operon* ごとに一定の確率で適用する。これが作用するときには、その *operon* 内の一つのシナプス結合を任意に選択しそれを取り除くとともに、新しいノードとそれに付随する新しい二つのシナプス結合を付加する。たとえば、Fig. 2における結合荷重値 w_1 を持つシナプス結合が選択されると、その結合を削除した後、結合荷重値 w_3 と w_4 を持つシナプス結合をその *operon* 内に生成する、という操作を行う。もし、その際に、Fig. 2のように、削除されるシナプス結合が入力ノードと出力ノードを直接結合させているものであるならば、新しく付加するノードとそれから伸びる二つのシナプス結合とで新しい *operon* を生成する。

Fig. 2 Add node mutation

また、ノード付加前とノード付加後で適応度が変化しないためには、ノード付加前後で同じ入力に対して等しい出力を行うことが必要となる。これはFig. 2において、ある値の入力に対して a_1 と a_2 が等しい値を示すことである。 a_1, a_2 を式で表すと(3)式となる。 a_i はノードからの出力値、 x_i はノードへの各入力値、 w_i は各結合荷重の結合荷重値、 $S_{[\]}$ はシグモイド関数を表す。

$$\begin{aligned} a_1 &= S_{[w_1 x_1 + w_2 x_2]} \\ a_2 &= S_{[w_3 S_{[w_4 x_1]} + w_2 x_2]} \end{aligned} \quad (3)$$

a_1 と a_2 が等しくなるためには適切な w_3, w_4 の値およびシグモイド関数 $S_{[\]}$ を用いる必要がある。(3)式に対して $a_1 - a_2$ が最小となるためには(3)式より、

$w_1 x_1 - w_3 S_{[w_4 x_1]}$ が最小となればよいことがわかる。さらにここで、 $w_1 = w_3$ という条件を加え、シグモイド関数が(4)式であることを考慮すると最終的に(5)式の値を各 x_1 に対して小さくすることが必要であるとわかる。

$$S_{[x]} = \frac{1}{1 + e^{\beta(\alpha - x)}} \quad (4)$$

$$f_{[x_1]} = x_1 - \frac{1}{1 + e^{\beta(\alpha - w_4 x_1)}} \quad (5)$$

ここで予備実験を通して、いくつかの w_4, α, β の値の組を用いて、 x_1, x_2 の値を $[0, 1]$ まで、 $w_1 (= w_3)$ の値を $[-10, 10]$ まで変化させたときの $f_{[x_1]}$ の値を計算した結果、 β の値が大きくなるにつれて α の値を小さくすることが必要であることがわかった。本論文では、 $\alpha = 0.5, \beta = 4.9, w_4 = 1$ の値を用いる。

3.2.2 シナプス結合付加突然変異

シナプス結合付加は、*operon* ごとに一定のある確率で適用される。これが作用すると、Fig. 3に示すように、その *operon* 内でランダムに選択されたノード一つに対し、その結合相手ノードを自身の *operon* 内のノードもしくは *operon*₀ 内のノードを候補としてランダムに一つ選び出しシナプス結合を生成する。このようなシナプス結合付加突然変異では、ある *operon*_{*i*} ($i > 0$) に属する部分ネットワークから *operon*₀ 以外の他の *operon*_{*j*} ($i \neq j > 0$) に接続することは決してない。そのため、各部分ネットワークが互いに独立な構造を構成するようになるため、モジュール性を持つ遺伝子型-表現型写像となることが期待できる。

なお、付加される際のシナプス結合の結合荷重値は0とする。これはシナプス結合付加突然変異による適応度の低下を発生させないためである。

Fig. 3 Add-connection mutation

3.3 シナプス結合荷重突然変異

本論文では、前述の二種類の突然変異から所望の構造を進化的に獲得させることに焦点を当てるため、結合荷重突然変異には、標準的な実数値GAでよく用いられる突然変異法を採用する。すなわち、平均0、標準偏差 σ (本論文では $\sigma = 0.05$) のガウス乱数を現在の結合荷重値に加えることによって行う。

4. 二重倒立振子制御問題

この章では3.で説明したMBEANNを用い、タスクとして速度情報を用いない二重倒立振子制御問題(Fig. 4(a))を取り扱う。この問題の困難性を増加させながら、TWEANNの代表的手法であるNEATとの比較を

行い、提案手法の有効性を検証する。また、3.1で述べたエンコーディング手法の有効性を検証するため、*operon* 構造を用いない、すなわち、ノード付加突然変異において新しく *operon* を生成せずにすべて *operon*₀ 内のノードとし、実質 *operon* 構造のない MBEANN (MBEANN Without Operons : MBEANN-WO) との比較も行う。

4.1 問題の特徴

速度情報を用いない二重倒立振子制御問題は、強化学習の分野においてよく取り扱われている。近年では、TWEANN のテスト問題としてもしばしば用いられている [5,6,12,13]。この問題では、長振子と短振子の長さ比が大きくなるにつれて制御の困難性が増大し、1:1 になると安定制御が不可能となる。

なお、文献 [6] にあるように、隠れ層ノードや再帰結合を持たないネットワークであっても、前後に台車を揺り動かすことでシステムを振動状態にして振子のバランスをとることができることもある。本研究では、この状態が出現するのを抑制する効果を持つ文献 [6] にある適応度関数を採用した。

4.2 実験設定

同様に、実験設定も文献 [6] と同じように行った。ネットワークへの入力値には中心位置から台車の距離 x 、垂直からの各振子の角度 θ_1 , θ_2 、そして、バイアスの計 4 入力を用いた。なお、バイアス入力として常に 1 の値を用いた。また、出力として台車に $[-10,10]$ N の範囲で力を加えた。用いた適応度関数は (6)-(8) 式である。

Fig. 4 Setting of the double pole balancing problem without velocity information

$$f_1 = \frac{t}{1000} \quad (6)$$

$$f_2 = \begin{cases} 0 & \text{if } t < 100 \\ \left(\frac{K}{\sum_{i=t-100}^t (|x^i| + |\dot{x}^i| + |\theta_1^i| + |\dot{\theta}_1^i|)} \right) & \text{otherwise} \end{cases} \quad (7)$$

$$f = 0.1f_1 + 0.9f_2 \quad (8)$$

ここで、 t は振子がバランスをとることができた時間ステップ、 K は 0.75 の値を持つ定数である。分母は台車位置、台車速度、長振子の角度、角速度の絶対値の 100

ステップ間の和である。

集団中の各個体は、各エピソードで 1,000 ステップ間バランスをとるか振子が 36° 以上倒れるか台車が前後 2.4m の範囲外に出た場合にシミュレーションを終了し、適応度値が計算される。

全部で 10 回行う各試行では、打ち切り世代 (1,000 世代) に到達するか、もしくは、その世代の最良個体が以下の二つの条件を満たした場合に終了とした。

- (1) 初期状態から 100,000 ステップ間バランスをとることができる。
- (2) 相異なる 625 通り¹の初期状態から評価を開始し、1,000 ステップ間制御可能であった初期状態数を汎化初期状態数とよび、この汎化初期状態数が 200 以上である。

本論文では、長い振子と短い振子の長さ比を 1:0.1, 1:0.2, 1:0.3, 1:0.4 と変更し、困難性の異なる四つの問題に対して、NEAT, MBEANN-WO および MBEANN の三つを比較した。

NEAT は Stanley のウェブサイト² からソースコードをダウンロードして用いた。NEAT ではこの問題に対して集団サイズを 1,000 としていたので、これを全実験で用いた。その他、MBEANN, MBEANN-WO で使用するパラメータは Table 1 に示すものを用いた。初期構造は NEAT, MBEANN-WO, MBEANN 共通に Fig. 4(b) に示す最小構造から開始した。

Table 1 Parameters for MBEANN

population size	1,000	weight mutation rate	1.0
tournament size	20	weight value in add-node	1
add-node rate	0.007	α in sigmoid function	0.5
add-connection rate	0.3	β in sigmoid function	4.9
maximal generation	1,000		

4.3 実験結果

Fig. 5 に短振子長さが 0.1, 0.2, 0.3, 0.4 の場合の NEAT, MBEANN-WO, MBEANN を用いた場合の実験における 10 試行の平均結果を示す。Fig. 5(a) はタスク達成率、Fig. 5(b) はタスク達成に要した世代数、Fig. 5(d) は汎化初期状態数、Fig. 5(f) は個体集団の平均ノード数、Fig. 5(h) は個体集団の平均シナプス結合数を表し、Fig. 5(c), (e), (g), (i) はそれぞれの標準偏差を表している。ただし、タスク達成世代数、汎化初期状態数はタスク達成可能であった試行のみの平均であるため、これらの値が直接的に手法の優劣を示すとは限らない。また、Fig. 6 にそれぞれの振子長さ比において、MBEANN を用いて獲得した構造の一例を示す。以下、

¹各変数 $x, \dot{x}, \theta_1, \dot{\theta}_1$ にそれぞれ 0.05, 0.25, 0.5, 0.75, 0.95 の値をとらせて $5^4 = 625$ 通りの初期状態を作成する。

²<http://nn.cs.utexas.edu/>

Fig. 5 Experimental results averaged over ten trials for NEAT, MBEANN-WO and MBEANN

タスク達成率, タスク達成世代数, 汎化初期状態数, ネットワークの規模の四つの観点から, 各振子長さにおける三つの手法の比較を行う。

【タスク達成率】 Fig. 5(a) より, 短振子長さが 0.1, 0.2 の実験においては三つの手法による違いはほとんど大差なく, タスク達成率は 0.9 もしくは 1.0 となっている。しかし, 短振子長さが 0.3, 0.4 と長くなるにつれ, NEAT

のタスク達成率は次第に低下し, 短振子長さ 0.4 では 1 試行しか成功していない。いっぽう, MBEANN-WO では短振子長さ 0.3, 0.4 の実験においても, タスク達成率は 0.8 となり, さほど低下していない。さらに, MBEANN では, 常に高いタスク達成率を維持できている。

【タスク達成世代数】 成功した試行のみにおけるタスク達成世代数の平均を Fig. 5(b) に示す。これから, すべての短振子長さにおいて NEAT が他に比して非常に小さい世代数でタスクを達成していることがわかる。しかし, 短振子長さ 0.4 では短振子長さ 0.1 の 7 倍近い世代数が必要となっている。また Fig. 5(c) より, NEAT では短振子長さの増加につれて標準偏差の値も増加し, 短振子長さ 0.3 の実験においては, 三つの手法の中で最も高くなっている¹。このことから, NEAT では簡単な問題においては非常に高速にタスク達成可能だが, 問題の困難さが増大するにつれてタスク達成率は低下し, タスク達成不可能もしくはタスク達成に必要な計算世代数の増大が顕著になることがわかる。MBEANN-WO においても, 問題の困難さが増大するとタスク達成に必要な世代数は多くなる傾向にあるが, 標準偏差に関しては短振子長さとの強い関連性は見られない。いっぽう, MBEANN では, その他に比較して, タスク達成世代数への多少の影響は見られるものの, すべての場合において MBEANN-WO よりも安定してタスクを達成していることがわかる。

【汎化初期状態数】 Fig. 5(d) より, 汎化初期状態数に関しても, 短振子長さ 0.1 の実験では NEAT が 300 と最も高い値を示しているが, 短振子長さ 0.4 の実験では 200 まで減少している。いっぽう, MBEANN-WO ではすべての振子長さにおいて 210 から 230 程度, MBEANN では 230 から 250 程度であり, 問題の困難さによる影響は NEAT より少ない傾向にある。

【ネットワーク構造の規模】 Fig. 5(f) および 5(h) より, 短振子長さ 0.1 の実験では NEAT を用いた場合, 獲得したネットワークのノード数の平均値が 6, シナプス結合の数の平均値が 10 と最も小さい。しかし, 短振子長さが増加するにつれて平均ノード数, 平均シナプス結合数は急激に増加し, 短振子長さ 0.4 の実験では平均ノード数は 53, 平均シナプス結合数は 332 までに増加している。同様に Fig. 5(g) および (i) より, 標準偏差の値も増加傾向にある。MBEANN-WO でも短振子長さ 0.2 から 0.4 の実験では, NEAT と同様に増加傾向が見られる。しかし, 短振子長さ 0.1 の実験では, 平均ノード数は 24, 平均シナプス結合数は 112 となっており, 簡単な問題であっても構造が肥大化している。いっぽう, MBEANN ではすべての振子長さにおいて, 平均ノード数は 7 前後,

¹ 短振子長さ 0.4 の場合で NEAT による実験の標準偏差が 0 となっているが, これは 10 試行中 1 試行のみしかタスク達成していないためである。

Fig. 6 Network topologies at the last generation for MBEANN

平均シナプス結合数は20前後となっており、また標準偏差値も小さいことから、安定して比較的小さな構造でタスク達成させている。

これらをまとめると以下のようにいえる。

- NEAT では、問題の難易度がそれほど高くない場合には非常に高速に解く。しかし、問題の難易度が増加するにつれてタスク達成率は落ちる。それとともに、獲得されるネットワーク構造の規模は急速に肥大化する傾向にある。
- MBEANN-WO では、常に安定してタスク達成が可能ではあるが、問題難易度の増加につれて、必要な計算世代数は増加し、ネットワーク構造の規模は増大する傾向を示す。
- MBEANN では、常に安定してタスク達成が可能である。問題の難易度の増加による必要な計算世代数やネットワーク構造の規模への影響は他に比較して小さい。

5. おわりに

TWEANN の代表的手法である NEAT には、いくつか深刻な問題点があることを指摘した。その根源は主要な遺伝的操作として交叉を用いることにありと考える。これを使用せずに、適応度値にほとんど影響の出ない新たな二つの構造突然変異（ノード付加、シナプス結合付加）を用いる手法 MBEANN を提案した。また、遺伝子型上の小さな変異は表現型の上でも局所的に影響が及ぶ程度であることが望ましいのではないかと考えるから operon 構造を持つエンコーディング手法を提案した。二重倒立振子問題において、NEAT との比較実験を行った結果、問題が容易な条件では、NEAT のほうが優れた結果を出す。問題の難易度が増大するにつれ、NEAT に比べて MBEANN のほうが安定により良い性能を示すことを確認した。

本論文における MBEANN の計算機実験では、予備実験と通しておおまかに適切であると思われる設定やパラメータ値を用いた。自然淘汰は進化アルゴリズムでは標準的なトーナメント戦略を用いた。また、結合荷重値の実数値突然変異法にも最も標準的で簡便な Gauss 分布

に基づく乱数を加えるという方法を用いた。それゆえ、さらに洗練された自然淘汰戦略や実数値突然変異法を用いてシステムチューニングをすれば、MBEANN の性能を更に向上させることが可能であると思われる。

今後、MBEANN の実問題での有効性を検証するために、進化ロボティクス (ER) の実機実験に適用する予定である。また、今回用いた程度のベンチマーク問題による実験では、入力ノードや出力ノードの数が少ないため比較的小さな構造のままタスク達成が可能であった。しかし、問題規模がさらに大きくなりがちな ER の問題などでは、タスク達成なネットワーク構造が大規模かつ複雑になると思われる。この場合には、本論文で MBEANN に用いたような基本的な淘汰戦略では十分な性能が発揮されない可能性もある。そのために、ネットワークの類似性に基づいた適応度シェアリングの導入や、遺伝情報の統計量など新たな指標として取り上げて多目的 GA のような高度な淘汰戦略を採用する必要も出てくるかもしれない。これらは今後の課題としたい。

謝 辞

福森淳一氏（神戸大学）に、膨大な計算機実験データの管理および解析をしていただきました。ここに謝意を表します。

参 考 文 献

- [1] P. J. Angeline, G. M. Saunders and J. B. Pollack: An evolutionary algorithm that constructs recurrent neural networks; *IEEE Transactions on Neural Networks*, Vol. 5, pp. 54-65 (1994)
- [2] R. D. Beer: Toward the evolution of dynamical neural networks for minimally cognitive; *From Animals to Animals 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pp. 421-429 (1996)
- [3] D. Floreano and C. Mattiussi: Evolution of spiking neural controllers; *Proceeding of Evolutionary Robotics -From Intelligent Robots to Artificial Life. ER'01*, T.Gomi(Eds.), Springer-Verlag, pp. 38-61 (2001)

- [4] D. E. Goldberg and J. Richardson: Genetic algorithms with sharing for multimodal function optimization; *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms* (J. J. Grefenstette, ed.), pp. 41–49 (1987)
 - [5] F. J. Gomez and R. Miikkulainen: Solving non-Markovian control tasks with neuroevolution; *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Denver (1999)
 - [6] F. Gruau and D. Whitley: A comparison between cellular encoding and direct encoding for genetic neural networks; *Larry Pyeatt Genetic Programming 1996: Proceedings of the First Annual Conference* (1996)
 - [7] Y. Liu and X. Yao: A population-based learning algorithm which learns both architectures and weights of neural networks; *Chinese Journal of Advanced Software Research* (Allerton Press, Inc., New York, NY 10011), Vol. 3, No. 1, pp. 54–65 (1996)
 - [8] S. Nolfi and D. Parisi: Growing neural networks, Technical Report, Roma: Institute of Psychology, CNR (1992)
 - [9] S. Nolfi and D. Floreano: Encoding, mapping, and development; *Evolutionary Robotics(2002)*, Vol. 9, pp. 223–239 (2002)
 - [10] J. C. F. Pujol and R. Poli: Dual network representation applied to the evolution of neural controllers; *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming* (1998)
 - [11] N. J. Radcliffe: Genetic set recombination and its application to neural network topology optimisation; *Neural Computing and Applications*, Springer-Verlag, 1:1, pp. 67–90 (1993)
 - [12] K. Stanley and R. Miikkulainen: Evolving neural networks through augmenting topologies; *Evolutionary Computation*, Vol. 10, No. 2, pp. 99–127 (2002)
 - [13] K. Stanley and R. Miikkulainen: Efficient reinforcement learning through evolving neural network topologies; *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, Morgan Kaufmann (2002)
 - [14] Xin Yao: Evolving artificial neural networks; *Proceedings of the IEEE*, Vol. 87, No. 9, pp. 1423–1447 (1999)
-