

# 実進化型ロボットシステムへの先行評価導入に関する基本的検討

## A Fundamental Study on Introducing Pre-evaluation into a Multiple Robot System Realizing Embodied Evolution

中井淳一 有田隆也

Junichi Nakai Takaya Arita

名古屋大学大学院情報科学研究科

Department of Complex Systems Science, Graduate School of Information Science, Nagoya University

“Embodied Evolution (EE)” is a methodology in evolutionary robotics, in which, without simulations on a host computer, real robots evolve based on the interactions with actual environment. However, we had to accept robot behavior with low fitness especially in the early generations when adopting the EE framework. We introduced pre-evaluation into the EE framework so as to restrain robot behavior whose fitness is predicted to be low. The pre-evaluation mechanism was adopted as a coevolutionary system in the proposed architecture. This paper reports on the results of the quantitative evaluation of a simplified model for the architecture.

### 1. はじめに

進化ロボティクス [Nolfi 2000] はロボット制御器を進化計算で自動設計する方法論であり、近年のロボット応用範囲の広がりに応じて注目を集めている。従来のほとんどの進化ロボット研究ではホスト計算機上でシミュレーションによる適応度評価に基づいて制御器を進化させ、それをロボットに転送して動作させる“Simulation & Transfer”手法が主流であったが、シミュレーション環境に適応した制御器が実世界のロボットで必ずしも同様の行動が可能とは限らず、シミュレーションによる適応度評価は実環境とのギャップを生む。また、環境やタスクを変更するたびに、シミュレータを変更する必要がある。

Watson らは Embodied Evolution を提案し [Watson 99] [Watson 2002]、ホスト計算機なしに実環境でのロボット行動により適応度評価することで、これらの問題を解決した (図 1)。制御器を構成するニューラルネットワークの結合重みを表す遺伝子セットは、ロボット行動によって評価され、他ロボットに遭遇したときに適応度に応じた確率で送信されることで、ロボットの行動は進化する。しかし、Watson らのモデルでは進化は他ロボットと遺伝子交換することのみで行っており、ロボットの遭遇頻度に強く依存してしまう。

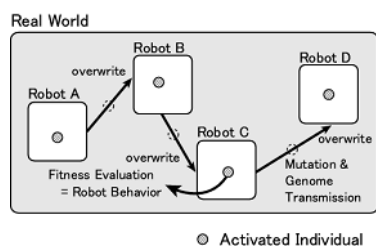


図 1: Watson らの実進化手法

そこで、臼井らは各ロボットに複数の仮想個体を持たせ、個別に進化できるようにすることで、他ロボットとの遭遇頻度に左右されない実進化を実現した [Usui 2003] (図 2)。また Elfving

連絡先: 中井淳一, 名古屋大学 大学院情報科学研究科 複雑系科学専攻, 〒464-8601 愛知県名古屋市千種区不老町, nakai@alife.cs.is.nagoya-u.ac.jp

らも臼井らと同様に複数の仮想個体をロボット内に持つモデルを提案しているが [Elfving 2005]、汎用的なタスク実行を狙っておらず、生物的妥当性に重点を置いて設計されたものと考えられ、ロボット単体では進化することができず、ロボット遭遇回数に対する進化依存性は解決していない。

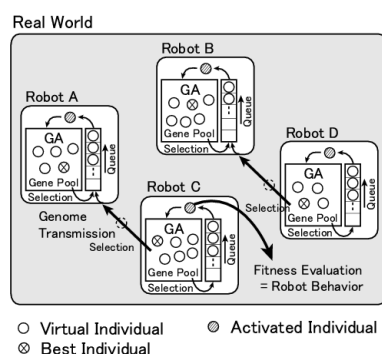


図 2: 臼井らの実進化手法

これらの研究では、ロボット行動だけで適応度評価するため、特に進化初期に低適応度動作が弊害を生みやすい。そこで、笠井らはロボット行動の前に先行評価し、低適応度とみなす個体は行動を回避してこれの解決を狙った [笠井 2007]。また、先行評価器はその精度を適応度として進化させ、制御器の進化系との共進化系を構成している。

しかし、先行評価系の導入が制御器の進化に対して与える影響には未解明な点が多い。さらに、タスクによっては先行評価値が実評価値よりも高くなりすぎて進化が進まなくなるなど、この共進化系独自の問題がある。そこで、本研究は複雑な挙動を示す笠井らのアーキテクチャを極めて単純化したモデルを作成し、性能を定量的に検証する。それにより、どのような場合に制御器の進化系と先行評価系が適正に共進化することができるのかを明らかにすることを目的とする。

まず、2 節で笠井らによるアーキテクチャの説明をする。次に、3 節で先行評価系を導入した単純なモデルを説明し、4 節でその評価結果を示す。そして、5 節で本論文をまとめる。

## 2. アーキテクチャの概要

先行評価系を有する実進化システムのアーキテクチャを図3に示す。同図はロボット体に関するコントローラの進化に関連する部分を示したものである。自己個体プール (Self-genome Pool) の中には制御器を表現する自己個体 (Self-genome) が適応度とともに格納されている。

まず、自己個体が自己個体プール内から適応度を用いて選択され、交叉、突然変異の処理がなされた後、キュー (Queue) に保持される。キューから取り出された自己個体は先行評価 (Fitness PreEvaluation) を行う。その評価結果が基準以下の個体は、ロボット行動による適応度評価を行わない。その場合は、その個体は先行評価値とともに自己個体プールへ戻される。適応度評価を行う場合は、一定時間、ロボットの行動を司り、タスクの達成度に応じて適応度評価 (Fitness Evaluation) され、その値とともに再び自己個体プールに戻される。この一連の流れを繰り返すことで進化が進む。

ロボットが他ロボットと遭遇すると、適応度を用いて自己個体を選択し、他ロボットに送信する。同時に、もし他ロボットから自己個体を受け取った場合、それはキューで適応度評価のループに合流する。

さらに、先行評価系を環境個体 (Environment-genome) として表現し、それは環境個体プール (Environment-genome Pool) に格納する。適応度としては、実評価値と先行評価値の小さいほうを大きいほうで割った値を用いている。両者が近いほど良い環境個体であり、高い適応度となる。

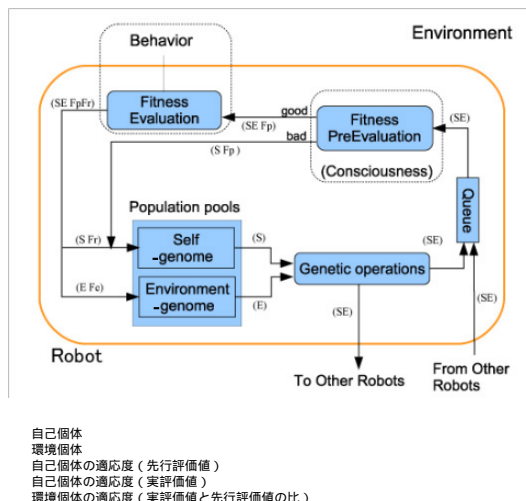
なお、先行評価系をどう構成し、環境個体で何を表現するかということに関しては、問題に応じて様々に設定可能である。タスクを忠実に再現し、不確実なパラメータのみを環境個体として表すケースから、タスクを完全にブラックボックスとして扱い、自己個体を引数として実評価の値を出力する関数を遺伝的プログラミングのようにして探索するケースまで考えられる。

また、このアーキテクチャの進化圧の特徴として、自己個体プール側では、実評価値が上がるが、先行評価値もバイパス時に適応度として使用されるため上がる。そして、環境個体側では、両評価値の差を縮めるが、先行評価値も閾値以下だと個体選択が無効化されるため上がる。つまり、総じて先行評価値が上がりやすいが、先行評価値が先行評価系の閾値を超え実評価が行われた場合、先行評価値は適応度として使用されないで、果てしなく上がり続けることはない。

## 3. 基本モデル

### 3.1 設定

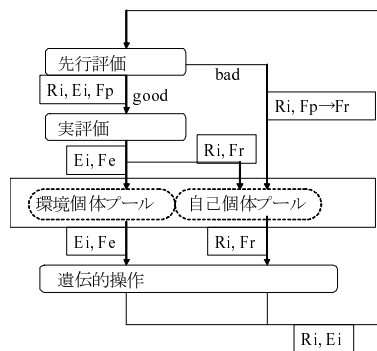
先行評価系を有する実進化システムの基本モデルの概要を図4に示す。自己個体、環境をそれぞれ0, 1のバイナリ列で表す。そして、自己個体と環境とのバイナリ列を比較し、その一致度を実評価値として算出することが、実環境でロボットを動かし評価することに対応するものとする。つまり、自己個体と環境のバイナリ列が完全に一致したとき、ロボットは環境に対応した制御器を生成した状態である。また、環境個体も0, 1のバイナリ列で表す。自己個体と環境個体のバイナリ列を比較し、その一致度を先行評価値として算出することが、先行評価系でのロボットの評価に対応するものとする。そして、環境個体の適応度は実評価値と先行評価値の差を表す値とする。両者が近く、差が少ないほど良い環境個体であり、高い適応度となる。環境個体が環境と直接比較されることなしに、環境に近づくことも本モデルの特徴である。したがって、自己個体と



S: 自己個体  
E: 環境個体  
Fp: 自己個体の適応度 (先行評価値)  
Fr: 自己個体の適応度 (実評価値)  
Fe: 環境個体の適応度 (実評価値と先行評価値の比)

図3: 先行評価系を有する実進化システムのアーキテクチャ

環境のバイナリ列が完全に一致し、環境と環境個体のバイナリ列が一致することで実評価値と先行評価値の差がなくなれば、ロボットが環境に適した制御器を獲得し、先行評価系が実環境をより正確に再現できた状態を表している。



Ri: 自己個体  
Ei: 環境個体  
Fp: 自己個体の適応度 (先行評価値)  
Fr: 自己個体の適応度 (実評価値)  
Fe: 環境個体の適応度 (実評価値と先行評価値の比)

図4: 基本モデル

次に、具体的な流れを説明していく。自己個体  $R_i$ 、環境  $G_i$ 、環境個体  $E_i$  をそれぞれ長さ  $B = 1000$  の0, 1のバイナリ列で表し、実個体の進化における適応度  $F_r$  は自己個体と環境の一致度

$$F_r = \sum_{i=1}^B \delta_{R_i G_i} \quad (1)$$

で表し、環境個体の進化における適応度  $F_e$  は実評価値  $F_r$  と先行評価値  $F_p$  の小さいほうを大きいほうで割った値

$$F_e = \begin{cases} \frac{F_p}{F_r} (F_r \geq F_p) \\ \frac{F_r}{F_p} (F_r < F_p) \end{cases} \quad (2)$$

とし、環境個体は実評価値と先行評価値の差を縮めるように進化する。 $G_i$  は便宜上全て1とする。そして、環境個体  $E_i$  は部分的に環境と同じ値1に固定し、ある割合を遺伝的アルゴリズムのループの中で変化させていく。これは、実際のロボット

を動かした場合を想定しており、先行評価系として最低限のシミュレータ環境を与え、部分的にシミュレータ環境のパラメータを変化させていくことに対応している。1 に固定する割合を環境個体の固定率  $p$  で表す。また、先行評価値  $F_p$  は自己個体と環境個体の一致度

$$F_r = \sum_{i=1}^B \delta_{R_i E_i} \quad (3)$$

である。自己個体プールには自己個体、環境個体プールには環境個体が入り、それぞれ最大個数  $P = 30$  の個体を格納することができる。

まず、ランダムに生成された自己個体を先行評価し、先行評価値が閾値以下の場合、実評価を行わずに先行評価値とともに自己個体プールへ戻される。閾値  $K$  は自己個体プール内の自己個体の評価値の平均値にある係数  $e$  を掛けた値

$$K = e \frac{\sum_{i=1}^P F_{ri}}{P} \quad (4)$$

とする。そして、先行評価値が閾値より大きい場合は実評価を行い、自己個体は実評価値と共に自己個体プールに戻し、環境個体は環境個体の適応度と共に環境個体プールに戻る。

次に、自己個体プールからルーレット選択により 2 つの個体を選び、突然変異、交叉を行い、次に実行する自己個体を作り、環境個体プールからも自己個体プールと同様の操作で、次に実行する環境個体を作る。突然変異率は 0.001 としている。そして、再びこの一連の流れを繰り返すことで進化が進む。

### 3.2 「風化」の導入

環境個体は基本的には環境と環境個体のバイナリ列が一致しているほど高い適応度をとるが、環境個体と環境のバイナリ列は直接比較することなく、実評価値と先行評価値の差を適応度として扱うため、必ずしも環境と環境個体のバイナリ列が一致している個体のほうが適応度が高くなるわけではない。特に、進化初期のランダム性により、実評価値、先行評価値は共に低い値にもかかわらず、その差が小さくなることで高適応度の環境個体が生成される可能性がある。その場合、そのような環境個体が環境個体プールに残ることにより、進化の停滞を引き起こす恐れがある。本研究では、それを防ぐために「風化」操作を遺伝的操作の前に導入する。具体的には、古い適応度を持つ環境個体は信頼度が低いとみなし、環境個体プール内の環境個体の適応度を時間経過に応じて一定の割合で減少させる。その割合を風化率と呼び、 $w$  で表す。この操作により、進化初期のランダム性による高適応度の環境個体が環境個体プールに残ることを防ぐ。

## 4. シミュレーション評価

まず、先行評価系導入前と先行評価系導入後の実評価値の推移を図 5 にそれぞれ示す。実評価値が約 900 に到達するのに必要な実評価回数は、先行評価導入前の約 4500 に対して、導入後は環境個体固定率 50% で約 3500 に減少することがわかる。ただし、環境個体固定率 0% の場合、共進化がそれほどうまく進まないため、減少しない。

次に、先行評価系の環境個体の適応度の推移を図 6 に示す。環境個体の固定率が 50%、0% のときは共に環境個体の適応度が 1 付近まで上昇していることから、実評価系と先行評価系の共進化が実現していることが分かる。環境個体の固定率が

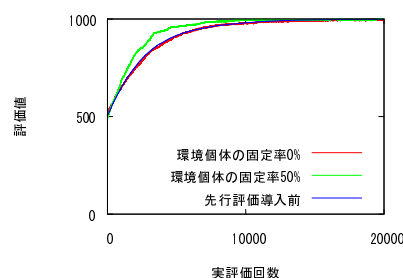


図 5: 先行評価導入前と後 ( $w = 0.1\%$ , ( $p = 0\%$ ,  $e = 0.8$ ) または ( $p = 50\%$ ,  $e = 0.98$ )) の実評価値の推移

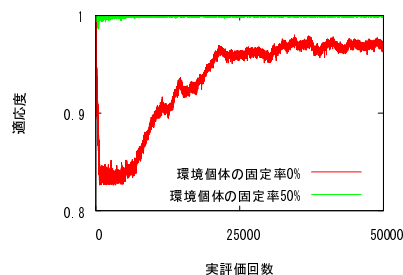


図 6: 環境個体 ( $w = 0.1\%$ , ( $p = 0\%$ ,  $e = 0.8$ ) または ( $p = 50\%$ ,  $e = 0.98$ )) の適応度の推移

0% のときは、とくに低い評価値から 1 付近に上昇しており、環境個体のほぼ全てをうまく自動生成できていることが分かる。

実評価値 900 到達までに要した実評価回数を環境個体の固定率別に示したのが表 1 である。0%、20% の場合は先行評価導入前よりも実評価回数は多いが、50% から先行評価導入前よりも実評価回数は少なくなり、80% からはほぼ半分の実評価回数で実評価値 900 に到達することができている。このことから、先行評価系を導入することで、低適応度の個体の実評価をはじき、実評価回数を少なく抑えることができていることが分かる。

表 1: 実評価回数による先行評価系の性能評価

環境個体の固定率 $p$	実評価値 900 到達時の実評価回数
0%	4955
20%	4374
50%	3683
80%	2295
100%	1652
先行評価系導入前	4560

風化率の違いによる、進化後の実評価値と環境個体の適応度の変化を表 2 に示す。風化率が 0%、つまり風化を行わないときは、実評価値は 800 付近で停滞しており、進化がうまく進んでいないことが分かる。このとき、環境個体の適応度はほぼ最大値に上昇しているように見えるが、これは進化初期のランダム性による高適応度の環境個体が環境個体プールに残っているからだと考えられる。風化率が 0.1% として、風化を行ったときは実評価値、環境個体の適応度は共に最大値に上昇しており、実評価系と先行評価系が共にうまく進化できていることが分かる。風化率が 0.4% になると、風化率が高すぎて、環境個体の進化が停滞している。つまり、風化率は高すぎずは駄目

だが、進化初期のランダム性による高適応度の環境個体に対処するためには必須であることが分かる。

表 2: 風化率の違いによる、進化後の実評価値と環境個体の適応度の変化 ( $e=0.8, p=0\%$ )

風化率	実評価値		環境個体の適応度	
	最大	平均	最大	平均
0	813.9	796.0	0.99	0.99
0.1	1000	999.4	0.97	0.97
0.2	1000	999.3	0.91	0.91
0.3	1000	999.3	0.87	0.87
0.4	999.4	972.1	0.16	0.16

閾値の調整パラメータ  $e$  の違いによる、進化後の実評価値と環境個体の適応度の変化を表 3 に示す。 $e = 1.0$  のときは実評価値、環境個体の適応度は共にある値で停滞してしまう。これは、閾値が高すぎるために、先行評価系を通ることができる個体がなく、実評価がまったく行われなくなってしまうからである。そこで、 $e = 0.8$  として閾値を調整すると、実評価値、環境個体の適応度は共にほぼ最大値に上昇しており、うまく進化が進むことが分かる。しかし、ここで閾値を下げすぎると、評価値は最大値まで上昇するが、全ての個体が先行評価系を通り実評価が行われてしまうので注意が必要である。

表 3:  $e$  の違いによる、進化後の実評価値と環境個体の適応度の変化 ( $w = 0.1\%, p = 0\%$ )

e	実評価値		環境個体の適応度	
	最大	平均	最大	平均
1	785.4	758.4	0.00	0.00
0.8	1000	999.4	0.97	0.97
0.6	1000	999.3	0.97	0.97
0.4	1000	998.9	0.97	0.97
0	1000	999.4	0.97	0.97

突然変異率の違いによる、進化後の実評価値と環境個体の適応度の変化を表 4 に示す。突然変異率が 0.001 以下のときは実評価値、環境個体の適応度は共にほぼ最大値になっており、実評価系、先行評価系が共にうまく進化できていることが分かる。しかし、突然変異率が 0.002 以上のときは実評価値、環境個体の適応度は共にある値までで停滞している。このことから突然変異率は 0.001 以下が適正な値である。

表 4: 突然変異率の違いによる、進化後の実評価値と環境個体の適応度の変化 ( $e = 0.8, w = 0.1\%, p = 0\%$ )

突然変異率	実評価値		環境個体の適応度	
	最大	平均	最大	平均
0.0005	1000	999.5	1.00	1.00
0.001	1000	999.4	0.97	0.97
0.002	905.3	887.2	0.36	0.36
0.003	819.2	789.9	0.00	0.00

## 5. おわりに

実進化型ロボットシステムへの先行評価系の導入が与える影響を明らかにするために、複雑な挙動を示す笠井らのアーキテクチャを極めて単純化したモデルを作成し、性能を定量的に検証した。その際、序盤のランダム性による高適応度の環境個体の発生を想定し、生成されてから時間が経った環境個体の適

応度は信頼度が低いとして、時間に応じて一定率の適応度を減少させていく、風化を導入した。その結果、序盤のランダム性による高適応度の環境個体の発生を抑制することができた。また、先行評価系の閾値を適正に調整する必要があることが分かった。

先行評価系の閾値に調整パラメータ  $e$  を加え、いくつかの値で検証を行った。その結果、適正な閾値の値は与えるタスクや突然変異率、環境個体の状態などで大きく変動することが分かった。よって、タスクの状態に応じて適正に調整してやる必要があり、この自動調整は今後の課題である。さらに、先行評価系を導入する前と後の進化速度を比較するために、高適応度到達時の実評価回数を調べた。その結果、環境個体の固定率の値によっては、先行評価前より少ない実評価回数で高適応度に到達することが可能であることが分かった。このことから、先行評価系を導入することで低適応度の実評価をはじめ、進化を速くすることができるのが分かった。

現在、2 足歩行ロボットによる歩行制御の実進化に先行評価系を導入する実験を行っている。進化初期の稚拙な歩行の軽減を目的とするものである。歩行行動の獲得において、事前に行動を選別しながら、同時に環境モデルを内部モデルとして作っていく処理は、人による歩行動作の獲得における脳の働きと対応しており、人の知能の形成という観点からも興味深いものであろう。

## 参考文献

- [Nolfi 2000] Nolfi, S. and Floreano, D.: *Evolutionary Robotics*, MIT Press (2000).
- [Watson 99] Watson, R. A. and Ficici, S. G. and Pollack, J. B.: Embodied Evolution: Embodying an Evolutionary Algorithm in a Population of Robots, *Proc. of the 1999 Congress on Evolutionary Computation* (1999).
- [Watson 2002] Watson, R. A. and Ficici, S. G. and Pollack, J. B.: Embodied Evolution: Distributing an Evolutionary Algorithm in a Population of Robots, *Robotics and Autonomous Systems* (2002).
- [Usui 2003] Usui, Y. and Arita, T.: Situated and Embodied Evolution in Collective Evolutionary Robotics, *Proc. of the Eighth International Symposium on Artificial Life and Robotics* (2003).
- [笠井 2007] 笠井大輔, 有田隆也: 先行評価系を有する実進化型群ロボットシステム, 第 34 回知能システムシンポジウム講演論文集 (2007).
- [Elfwing 2005] Elfwing, S. and Uchibe, E. and Doya, K. and Christensen, H.: Biologically Inspired Embodied Evolution of Survival, *Proc. of the IEEE Congress on Evolutionary Computation 2005* (2005).