共進化に基づく先行評価の 実進化システムへの導入

中井 淳一

目次

第1章	はじめに	1
第2章	研究背景	3
2.1	進化ロボティクス	3
2.2	実進化	6
第3章	実進化への先行評価の導入	9
3.1	手法	9
3.2	基本的評価	10
	3.2.1 設定	10
	3.2.2 評価	14
	3.2.3 まとめ	18
第 4 章	2 足歩行ロボットへの適用	21
4.1	設定	21
4.2	結果.....................................	25
4.3	まとめ	30
第5章	適応度近似手法としての評価	33
5.1	概要.....................................	33
5.2	設定	34
5.3	結果.....................................	38
5.4	まとめ	42
第6章	おわりに	43
謝辞		45

参考文献 47

第1章

はじめに

近年,工場で部品の組立などを行う産業ロボット,介護用のロボット,家庭用のペットロボットやお掃除ロボットなど,様々な場所でロボットが活用されている.ショッピングセンターの案内ロボットなどの試験使用も始まり,料理を作るためのロボットや窓を拭くためのロボット等も開発されている.そして,ライントレースロボットやマイクロマウス等に加え,ホビー用の2足歩行ロボットも広く普及し,大会が行われている.

その中で,多関節多足ロボット等の,複雑な機構を持つロボットの動作を設計する場合,目的の動作を行わせるための関節の動きを人の手で設計するのは難しい.また,お掃除ロボットや介護ロボット,ショッピングセンターの案内ロボットなどは,人とのやり取りが必要とされ,動作環境が予測できないことが多く,その動作や機構を設計するのが難しい.そのため,ロボットの動作や機構の設計の方法は大きな課題である.

実社会において、ロボットを自動的に動作させるアプローチとしては、ティーチングプレイバックが、主に産業ロボット等で、行われている。これは、最初に人間がロボットを動かして教示することで、それ以降、ロボットがまったく同じ動きを再現する方法である。このようなロボットが部品を掴む場合、様々なセンサーや精度の高い画像認識を搭載することで、部品との位置のずれを微調整し、掴めるよう工夫されている。エアコン等も、人感センサー等で人の位置を感知し、人がよく居る位置を学習していく機能が搭載されている。また、介護用の会話ロボット等では、状況に応じた反応を条件として多数プログラムしておき、画像認識や音声認識などを用いて、把握した状況に応じて自律的に反応を返しているように見せているものもある。このように、部分的に人の手を借りることなく、環境に応じて自動的に動作を調整するロボットが増えている。しかし、これらの方法は、ロボットの機構や動作環境が複雑な場合のように、人が動作を予測できない場合は適用が難しい。

進化ロボティクスとは,進化的計算を用いて,ロボット自身が環境との相互作用により,自動的にその制御器を設計するものであるが,人の手を借りる必要がないことが利点である.主な手法は,シミュレーション上での評価により進化的計算を行い,その結果をロボットに転送して動作させる手法である.しかし,進化ロボティクスを用いて,動作のほとんどを自動的に獲得するロボットは,あまり実社会では活用されていない.その理由として,以下のいくつかの問題が挙げられる.シミュレーションと現実の差異により,ロボットが予期しない動作をしてしまう可能性がある.また,実社会のような,予め動作環境を把握し難い状況では,シミュレーションを設計するのが難しい.実進化はそれらの問題の解決を狙った手法であるが,ロボット行動による評価により進化的計算を行うため,特に進化初期に危険な行動をしてしまう可能性がある.また,時間的コストとロボットの疲労が増大する可能性がある.

そのような問題は,実社会において,ロボットを動作させる際に障害となる.ロボットと人とのやり取りが必要となる場合には,人に危害を与える可能性があり,災害現場や原子力発電所などの危険な場所では,事態をさらに悪化させる可能性がある.本論文では,このような場合に,進化ロボティクスの手法を基にしながらも,ロボットの自律的な行動獲得の際の危険な行動を回避し,安全な動作を可能とするための手法を提案する.

本論文の構成は次の通りである.本章に続いて第2章では,進化ロボティクスについて の基礎的な知識と背景について述べ,その問題点を挙げる.そして,それらの問題点の解 決を狙った手法である実進化について述べ,その先行研究を紹介し,実進化ロボット系全 般に付随する問題点を挙げる.第3章では,実進化ロボット系全般に付随する問題点で ある,ロボット行動で適応度評価するため,低適応度行動が弊害をもたらす可能性(特に 進化の初期段階)を解決するための手法を提案し,その基本的評価を行う.具体的には, ロボット行動の前に「先行評価」を導入し、低適応度と予測される個体のロボット行動を 回避する、さらに、先行評価の精度を適応度とした、実進化系との共進化系を構成して、 先行評価系も自動作成させる.第4章では,実進化を用いて2足歩行ロボットに歩行動 作の自律的獲得を行わせる際,初期に頻出しがちな転倒の回数を,先行評価系を導入する ことで削減することを試みる.さらに,先行評価系のユーザによる作り込みの度合いによ る,提案手法の有効性の評価の観点ごとの変化を,先行研究の手法との比較において,明 らかにする.第5章では,先行評価系を適応度評価近似手法として捉え直し,関数最大化 問題に適用することで,実進化だけでなく幅広い問題への適用可能性を検証する.それに より,実進化に限らず,適応度評価にコストやリスクが要求される問題において,幅広く 利用できることを示す.第6章では,各章の結果の概要をまとめる.そして,これらの研 究目的がどこまで達成できたかをまとめ,今後の展望について述べる.

第2章

研究背景

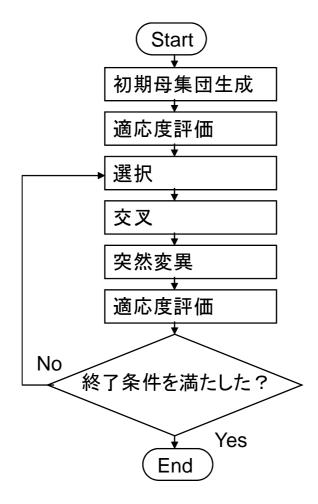
第2章では,進化ロボティクスの基礎的な知識と背景として,代表的な進化計算手法である遺伝的アルゴリズムを紹介し,進化ロボティクスの主な従来手法について述べ,その問題点を挙げる.そして,その解決を狙った実進化について述べ,その先行研究を紹介し,実進化ロボット系全般に付随する問題点を挙げる.

2.1 進化ロボティクス

進化ロボティクスは進化計算によるロボット制御器の自動設計を目指す研究分野であり,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[1].ロボット自身が環境との相互作用により,人の手を借りることなく,自動的にその制御器を設計することができる.進化ロボティクスのための手法としては,ニューラルネットワーク,遺伝的アルゴリズム,力学系など様々なものが用いられる.

その中で代表的な進化計算手法として遺伝的アルゴリズム (GA) が広く用いられている.GA とは,ダーウィンの進化論も基づいた,生物が環境に適応して進化していく過程を模した進化計算手法である.自然界における生物の進化過程においては,ある世代を形成している個体の集合,すなわち母集団の中で環境に対してより適応した個体がより高い確率で生き残り,次の世代に子を残す.このメカニズムをモデル化し,環境に対して最もよく適応した個体,すなわち目的関数に対して最適値を与えるような解を計算機上で求めようというのが GA の概念である.GA では,個体は設計変数の値がコーディングされた染色体と呼ばれる文字列として表現され,この染色体をデコーディングすることにより設計変数を読み出し,目的関数の値を計算する.このとき,染色体の内部表現のことを遺伝子型,それによって定まる個体の形質を表現型と呼ぶ.また,個体の集団のことを母集団

と呼ぶ、GA はこの母集団に対して選択,交叉,突然変異などの遺伝的操作を繰り返し行うことによって解の探索を行う、一般的な GA における処理の流れを表しているのが図 2.1 である、



初期母集団生成 ランダムに個体の集団を生成する.

適応度評価 全ての個体を目的に沿った適応度関数で評価をする.

- 選択 母集団のサイズと同じだけ,主に適応度に応じた確率で,個体を選んで次世代の集団をつくる(ルーレット選択,エリート選択,トーナメント選択等).
- 交叉 選択された 2 つの個体の染色体の内部表現を部分的に交換する (1 点交叉,多点交叉,一様交叉等).

突然変異 選択された個体の染色体の内部表現に一定の確率でランダムな変異を加える.

図 2.1 GA の概要

進化ロボティクスにおいて,適応度評価をどのように行うかが問題となる.ロボットを 用いて適応度評価を行う場合,以下のような問題があるからである.

1. 機械的な頑丈さ

進化中(特に進化初期)のロボットは,壁に高速で衝突するなど,自身を損傷させる行動を取る危険性がある.

2. エネルギーの供給

平均的なバッテリー持続時間より進化時間が長い場合,エネルギーを供給するか, 他の方法でこの問題を回避する必要がある.

3. 解析

進化したロボットの制御器が,複雑になる場合があり,理解し難くなる可能性がある.

4. 時間的効率

進化に膨大な時間が必要な場合があり,連続動作が可能な環境を作るか,他の方法で時間を短縮する必要がある.

5. 適応度関数の設計

適応度関数の設計は,進化の結果に大きな影響を与えるが,未知の環境で動作させる場合,有効な適応度関数を設計するのは難しい.

これらの問題から,主にホスト計算機での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手法 (以下, Simulation & transfer 手法と呼ぶ) が主流である.進化計算をホスト計算機上で行うため,進化中のロボットの損傷を気にする必要がなく,十分な計算能力を持つ計算機を用意できれば,時間的効率も良く,エネルギーの供給も容易である.しかし,以下のような問題点を持っている.

- 1. 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に有効とは限らず,実環境とのギャップを生みがち(以下,ギャップ問題と呼ぶ).
- 2. 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.
- 3. シミュレーションでは,ロボット台数の増加に対するスケーラビリティに乏しい.

これらの問題の解決を狙った進化ロボティクスの実現手法として次節で述べる実進化が提 案されている .

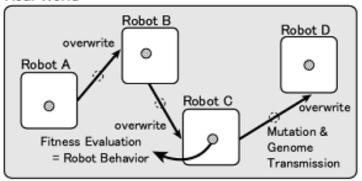
2.2 実進化

実進化とは,シミュレータ上での適応度評価を行わず,実ロボットによる適応度評価を用いて進化的計算を行う手法である.Simulation & transfer 手法のようなギャップ問題は発生しないが,適応度評価を実ロボットを用いて行うため,その時間的コストが問題となる.進化ロボティクスにおける Simulation & transfer 手法等では,前節で挙げたギャップ問題が起きていた [2,3].この問題はシミュレータを正確に設計することで回避できる場合もあるが [4,5],環境の情報を把握している場合にしか設計することができない.環境の情報を把握していても,全てを正確にシミュレータに組み込むことは難しく,シミュレータの作成に多くの時間が必要になるかもしれない.

Watson らはロボット制御器の自動設計のための方法論として,実進化を提案し,適応度評価に実ロボットを用いることで,ギャップ問題の解決を狙った [6, 7].(図 2.2)また,全ての適応度評価を実ロボットを用いて行うと時間的コストが大幅に増加するため,適応度評価のためのロボットを複数用いることでその軽減を狙っている.彼等はこの手法を,光に向かう性質を獲得する,単純な光走性タスクに実装し,その有効性を検証している.各ロボットに遺伝子を1個ずつを持たせ,その遺伝子はニューラルネットワークで制御器を表現している.そして,進化計算によりその重みを変化させている.タスクの途中で他のロボットに遭遇したとき,適応度に応じた確率で遺伝子を送信することで,高適応度の遺伝子がロボット集団中に広まり進化が進む.しかし,各ロボットが1つしか遺伝子を持たないため,他ロボットとの遭遇頻度に進化速度が依存している.また,もし優れた遺伝子が生成されても,それが集団中に広がる前に上書きされて消失してしまう可能性があった.

他にも,多くの実進化に関する研究が行われている [8, 9]. 臼井らは各ロボットに複数の遺伝子を仮想個体として持たせ,ロボット内でも進化させることにより,ロボット遭遇頻度に対する進化依存性の解決を狙った [10]. その概要を図 2.3 に示す.臼井らはこの手法を,複数の小型ロボットに実装し,障害物回避タスクや追跡タスクを行い,その有効性を検証した.各ロボットは内部に複数の仮想個体を持つため,単独で進化が可能となっており,同時に優れた遺伝子が消失してしまう問題の解決も狙っている.Elfwing 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方式を提案している [11]. 各ロボットはエネルギーを集めるタスクを行い,エネルギーが 0 になるとその個体は死に,他ロボットと個体をやり取りし新しい個体が生成される.しかし,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

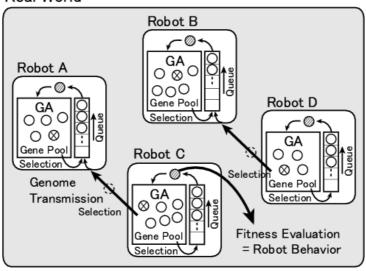
Real World



Activated Individual

図 2.2 Watson らの実進化手法

Real World



- O Virtual Individual
- Activated Individual
- ⊗ Best Individual

図 2.3 臼井らの実進化手法

ことができず、ロボット遭遇頻度に対する進化依存性は解決していない.

Perez らは各ロボットが複数の仮想個体を持ち,遺伝的プログラミングにより進化する.個体の送信頻度は適応度にかかわらず,時間ステップ毎に周りの個体に適応度に応じて個体を送信し,それを受け取った場合は最も適応度の低い個体とそれを交換していくという,"Microbial GA"に基づいた島モデルを実装している [12].Simões らは各ロボット間の個体の交換に無線を使用することで,自由に個体のやり取りが可能な実進化を提案した [13].適応度に応じて個体を集団中に送信することで進化が進む.この手法では全てのロボットが無線で繋がっているため,ロボットの台数が多い場合にも有効である.また,ロボットの遭遇頻度に対する進化依存性も解決している.

ロボットの進化計算に進化戦略を用いている研究も存在し、Walker らは、ロボットを集中管理するシステムを備え、ロボットをシミュレータ上で従来の遺伝的アルゴリズムを用いて進化させ、その結果を実ロボットに転送し、さらに進化戦略を用いて進化させることで、現実の環境にも適応させることを狙っている [14] . Nehmzow らは、2 台のロボットを用いて進化戦略による実進化を実現している [15] . お互いに個体を交換する際に、まず個体の適応度を比べ、低い適応度の個体には強い適応度の個体と交叉させ、強い適応度の個体は突然変異を加えるか、キャッシュされた最も高い適応度の個体と交叉を行うことで、少ない台数のロボットでも進化が進む.

実進化ロボット系全般に付随する問題点として,これらの研究は,主にロボット行動のみで適応度評価を行うため [7,10,12,13],特に適応度の向上していない進化初期に,弊害を生みやすい.また,時間的コストとロボットの疲労が増大する可能性があった.そこで,我々は,ロボット行動の前に先行評価を導入し,低適応度と予測される個体のロボット行動を回避することによる解決を狙っている [16].さらに,先行評価の精度を適応度とした,実進化系との共進化系を構成して,先行評価系も自動作成させる.そのため,実進化を行う前に,シミュレーションによる適応度評価により進化計算を行う場合 [14] でも,そのシミュレーションを作成する手間を軽減し,実進化を行う際のギャップ問題による予期しない動作を軽減できる可能性がある.

第3章

実進化への先行評価の導入

第3章では,先行評価系を有する提案手法の概要を述べる.具体的にはロボット行動の前に共進化により生成した先行評価系による評価を行い,低適応度とみなす個体による行動を回避する手法を提案する.そして,先行評価系のアーキテクチャを極めて単純化し,単純な問題に適用して,その性能の基本的評価を行い,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ついて明らかにする.また,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停滞するなど,この共進化系独自の問題があり,そのような問題について明らかにするとともに,実進化へ先行評価を導入する際の注意点について述べる.

3.1 手法

先行評価系を有する実進化型ロボットシステムのアーキテクチャを図 3.1 に示す.同図はロボット一体に関するコントローラの進化に関連する部分を示している.自己個体プール (CP) には制御器を表現する自己個体 (CG) が適応度 (Fr) 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ず,先行評価値 (Fp) 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,ロボット行動を司り,タスク達成度に応じて適応度評価され,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制御器が進化する.他ロボットと遭遇すると,適応度に基づいて自己個体を選択し,他ロボットに送信する.同時に,他ロボットから自己個体を受け取った場合,それは適応度評価のループにキューで合流する.

なお,本論文では,ロボット1台による基本的評価を行うので,ロボット間の送受信は扱わない.

さらに,先行評価系を環境個体(EG)として表現し,環境個体プール(EP)に格納する. 適応度(Fe)として,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る.これが1に近いほど良い個体であり,適応度値が大きくなる.環境個体で何を表現して先行評価系を構成するかについては,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探索するケースまで考えられる.

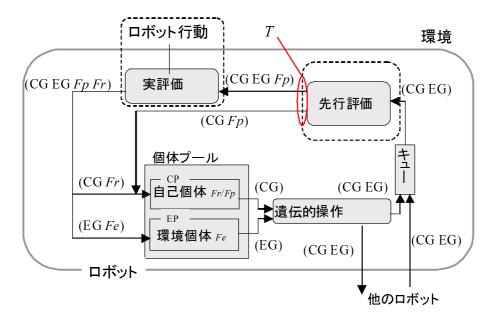
3.2 基本的評価

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.また,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停滞するなど,この共進化系独自の問題が起きることが想定される.そこで,我々は,複雑な挙動を示す先行評価系のアーキテクチャを極めて単純化し,単純な問題に適用して,性能評価を行った.

3.2.1 設定

先行評価系を有する実進化システムの基本的評価の概要を図 3.2 に示す.自己個体を 0, 1 のバイナリ列で表す.そして,環境も 0, 1 のバイナリ列で表し,ロボットが環境に対応した制御器を生成した状態を,自己個体と環境のバイナリ列が完全に一致したときとする.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.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一致率とする.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応した制御器を獲得し,先行評価系が実環境を完全に正確に再現できた状態を表している.

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



CG: 自己個体

EG: 環境個体

CP: 自己個体プール

EP: 環境個体プール

Fr: 実評価値

Fp: 先行評価値

Fe: 環境個体の適応度 (実評価値と先行評価値の一致度)

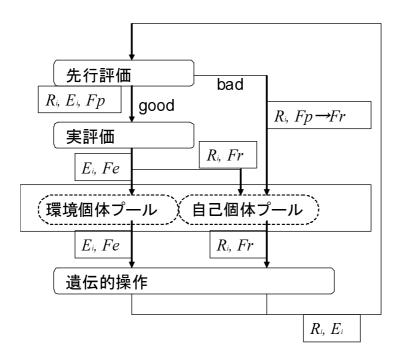
T: 実評価実行か回避の閾値

図 3.1 先行評価系を有する実進化システムのアーキテクチャ

長さ B=1000 の 0,1 のバイナリ列で表し,添え字 i は何番目のビットかを表している. 実個体の進化における適応度 Fr は自己個体と環境の一致度

$$Fr = \sum_{i=1}^{B} \delta_{R_i G_i} \tag{3.1}$$

で表す. δ はクロネッカーのデルタである.環境個体の進化における適応度 Fe は実評価



 R_i : 自己個体 E_i : 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 3.2 基本的評価の概要

値 Fr と先行評価値 Fp の小さいほうを大きいほうで割った値

$$Fe = \begin{cases} \frac{Fp}{Fr} (Fr \ge Fp) \\ \frac{Fr}{Fp} (Fr < Fp) \end{cases}$$
 (3.2)

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,固定されていない残りの遺伝子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 Fp は自己個体と環境個体

の一致度

$$Fp = \sum_{i=1}^{B} \delta_{R_i E_i} \tag{3.3}$$

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は, 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 T 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

$$T = e^{\frac{\sum_{i=1}^{P} Fr_i}{P}} \tag{3.4}$$

とし,P は個体プール内の集団サイズを表し,添え字i は個体プール内の何番目の個体かを表す.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す.

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,再びこの個体の選択,遺伝的操作,評価という一連の流れ (図 3.3) を繰り返すことで進化が進む.

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一致率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、自己個体か環境個体がそれぞれの個体プールに戻る度に、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。

個体プール内に R_i と E_i を各 30 個体ランダムに生成while (終了条件非成立)

ルーレット選択で R_i , E_i を個体プールから各 2 個体選択評価する R_i , E_i を遺伝的操作により生成

 R_i を先行評価 (Fp の算出)

if (*Fp* が *T* 以上)

 R_i を実評価 (Fr の算出)

 R_i を Fr と共に自己個体プールに返還

 E_i を Fe と共に環境個体プールに返還

else

(実評価せずに) R_i を Fp と共に自己個体プールに返還

endif

個体プールに風化操作

endwhile

 R_i : 自己個体

 E_i : 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)

T: 実評価実行か回避の閾値

図 3.3 基本的評価における進化アルゴリズムの流れ

3.2.2 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 3.4 にそれぞれ示す.どの条件でも実評価値は単調に増加し,最大値 1000 に至ったが,固定率 50 %の場合が他と比べて素早く増加した.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.

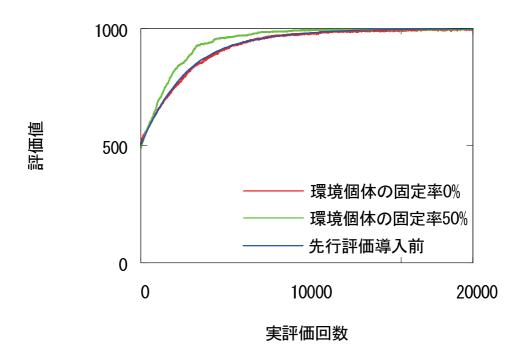


図 3.4 先行評価導入前と後 ($w=0.1\%,\ (p=0\%,\ e=0.8)$ または $(p=50\%,\ e=0.98)$) の実評価値の推移

次に,先行評価系の環境個体の適応度の推移を図 3.5 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 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.また,この両方の場合において,適応度が初めから 1 に近い値となっているが,これは環境個体は実評価値と先行評価値の一致率を適応度としているので,特定の自己個体との適応度評価のみで高適応度となっても必ずしも環境を的確に表しているとは限らないということに起因すると考えられる.この場合は,進化初期にランダムに作成された個体の適応度(実評価値)は 0.5 近辺に集中し,ランダムに作成された環境個体も 0.5 近辺の適応度(先行評価値)に集中するので,信頼度の低い高適応度の環境個体が発生してしまっている.

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 3.1 である .0% の場合は先行評価導入前よりも実評価回数は多いが ,20% 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 .

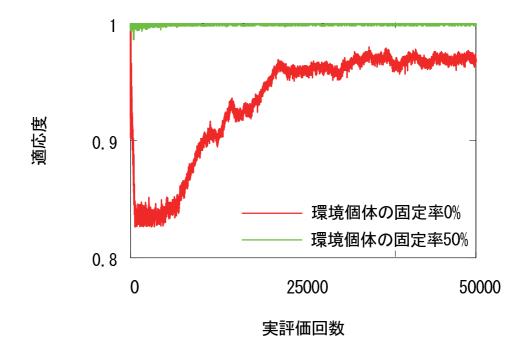


図 3.5 環境個体 (w=0.1%, (p=0%, e=0.8) または (p=50%, e=0.98))の適応度の推移

環境個体の固定率 p 実評価値 900 到達時の実評価回数 (比率) 0% 4955 (1.09) 20%4374 (0.96) 50%3683 (0.80)80%2295(0.50)100% 1652 (0.36) 先行評価系導入前

4560 (1.00)

表 3.1 実評価回数による先行評価系の性能評価

風化操作による,進化後の実評価値と環境個体の適応度の変化を表 3.2 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、環境と環境個体の一致率は 50% 程度しか一致してい ない、そのような高適応度の環境個体は、自己個体が多様化してくると良し悪しを判別で きない.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.環境と環境個体の一致率も 95% 程度一致しており,風化操作により,古い適応度は信頼度が低いとみなして,進化初期のランダム性による高適応度の環境個体を除去できたことが分かる.

表 3.2 風化操作による,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%)

	実評価値		環境個体の適応度		
風化率	最大	平均	最大	平均	環境と環境個体の一致率
0	813.9	796.0	0.99	0.99	0.509
0.1	1000	999.4	0.97	0.97	0.958

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 3.3 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.これは,自己個体プール中の各個体の先行評価値と実進化値に差が生じたことに起因すると考えられる.先行評価が実評価に比べて高い個体がプールに増加する場合(高い実評価値を持った個体がエリート保存により集団中のほとんどを占める),進化操作で選択されて先行評価されても,集団内の適応度の平均値より低い評価値になって実評価が行われず,プールにも残れないことで,進化が進まないデッドロックが発生してしまう.そこで,e=0.8 として閾値を調整すると,先行評価を通過しやすくなり,デッドロックの発生を抑制でき,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いることが分かる.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.

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 3.4 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 3.3 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%)

	実評価値		環境個	国体の適応度	
e	最大	平均	最大	平均	実評価率
1.0	785.4	758.4	0.00	0.00	0.30
0.8	1000	999.4	0.97	0.97	0.51
0.6	1000	999.3	0.97	0.97	0.71
0.4	1000	998.9	0.97	0.97	1.00
0.0	1000	999.4	0.97	0.97	1.00

表 3.4 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%)

	実評価値		環境個体の適応度	
突然変異率	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

3.2.3 まとめ

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, そのアーキテクチャを極めて単純化し,単純な問題に適用して,性能を定量的に検証した.まず,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,先行評価前よりも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き,進化を速くすることができるのが分かった.

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.

さらに,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.また,自己個体プール中の各個体の先行評価値と実進化値に差が生じた場合に進化が停滞するが,閾値の調整により,解決可能であることが分かった.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,この自動調整を行うための工夫が必要である.そのため,本論文では以降,環境個体プール内の平均適応度を閾値計算に加えることで,先行評価系の進化具合に応じた閾値の自動調整を行わせる.

第4章

2足歩行ロボットへの適用

第4章では,これまでの研究に基づき,先行評価を実進化型 2 足歩行ロボットに導入することにより,歩行動作を自律的に獲得する際の初期に頻出しがちな転倒の回数の削減を試みる。2 足歩行ロボットの転倒は損傷に直結するため,有用であろう。その際,先行評価系を 2 つの手法で構築した。1 つ目は定義された関数のパラメータ (PE1),2 つ目はニューラルネットワークの重み (PE2) をそれぞれ進化的に最適化する手法である.前者は探索空間が狭く,後者は探索空間が広くなっており,先行評価系の作り込みの度合いによる性能の変化を明らかにする.

4.1 設定

本研究では , ゼネラルロボティックス (株) の 4 自由度の腕 2 本 , 6 自由度の足 2 本を持つ人間型ロボット HRP-2m Choromet を使用した . 寸法は身長 350mm, 幅 175mm, 奥行 110mm , 重量は 1.5kg である . 両足裏に 1 個ずつ搭載した力センサを用いて適応度評価し , このロボットの歩行行動を進化させる . 図 4.1 が実験中の様子である .

歩行パターンを図 4.2 に示す.「中腰で両足を揃えた状態(状態0) 両足を傾けて重心の左移動 右足を前に出し(動作1) 両足を傾けて重心の右前方移動(動作2) 左足を前に出し 両足を傾けて重心の左前方移動 動作1 動作2 左足を前に出して状態 0へ」という一連の行動を1試行とする.

この際,足の軌跡は \cos 関数で生成し,足を傾けるときの重心の移動距離 $a(0 \le a \le 0.05)[m]$,歩幅 $c(0 \le c \le 0.05)[m]$,足を上げる高さ $d(0 \le d \le 0.02)[m]$ を自己個体の遺



図 4.1 実験中の様子

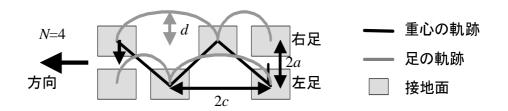


図 4.2 ロボットの歩行パターン

伝子データとする.実評価値 Fr は,歩行行動中の力センサ値 Z_i ,歩幅 c,を用い,

$$Fr = c\sum_{i=1}^{N} Z_i \tag{4.1}$$

で表す (転倒せずに前に進むほど大きい). N は 1 試行に必要なステップ数である. ${
m PE1}$ による先行評価系の場合 , 先行評価値 Fp は ,

$$Fp = crP_a^{a_1}P_c^{c_1}P_d^{d_1} (4.2)$$

$$P_a = -a'(a - a_2)^2 + 1 (4.3)$$

$$P_{a} = -a'(a - a_{2})^{2} + 1$$

$$a' = \begin{cases} 1/a_{2}^{2} & (a_{2} \ge a_{max}/2) \\ 1/(a_{max} - a_{2})^{2} & (a_{2} < a_{max}/2) \end{cases}$$

$$(4.3)$$

$$P_c = -c'(c - c_2)^2 + 1 (4.5)$$

$$c' = \begin{cases} 1/c_2^2 & (c_2 \ge c_{max}/2) \\ 1/(c_{max} - c_2)^2 & (c_2 < c_{max}/2) \end{cases}$$

$$(4.6)$$

$$P_d = -d'(d - d_2)^2 + 1 (4.7)$$

$$d' = \begin{cases} 1/d_2^2 & (d_2 \ge d_{max}/2) \\ 1/(d_{max} - d_2)^2 & (d_2 < d_{max}/2) \end{cases}$$
 (4.8)

で表し, $a_1,\,a_2,\,c_1,\,c_2,\,d_1,\,d_2,\,r$ を環境個体の遺伝子データとする.初期値として,当たりをつけた予想値 $a_1=3,\,a_2=0.03,\,c_1=3,\,c_2=0.05,\,d_1=3,\,d_2=0.015,\,r=18000$ を与えた. $a_{max},\,c_{max},\,d_{max}$ はそれぞれ $a,\,c,\,d$ の取り得る最大値である.突然変異率は 0.3 としている.

ロボットは,重心が安定した範囲を越えると転倒する.そこで,ロボットの制御器の各調整パラメータ(自己個体)に対応した先行評価系を,大まかに適応度が一番高いところを頂点とした,その周りで値が変動する山の形をした関数で表現し,頂点と勾配を先行評価系の調整パラメータ(環境個体)としたものが,この先行評価の基本的な設計方針である.これにより,山の勾配によっては,ある範囲を越えると急激に値が減少する形となり,ロボットの転倒が表現されることを期待する.

PE2 による先行評価系の場合, 先行評価値の計算のための適応度関数を表現するニューラルネットワークは:

$$y_j = \operatorname{sig}(\sum_{i=1}^3 x_i w_{i+3(j-1)} - \theta_j)$$
(4.9)

$$z = \operatorname{sig}(\sum_{i=1}^{4} y_i w_{i+12} - \theta_5)$$
 (4.10)

$$sig(x) = \frac{1}{1 + \exp(-x)}$$
 (4.11)

である. $\theta_1 \sim \theta_5$ はニューロンの閾値.ニューラルネットワークの入力は $x_1 = a/a_{max}$, $x_2 = c/c_{max}$, $x_3 = d/d_{max}$, そして,出力として先行評価値が $Fp = z \times 10^3$ で算出される.環境個体はニューラルネットワークの 21 個の重みで構成される $(w_1 \sim w_{16})$.初期値は-1.0 から 1.0 の範囲の値をランダムに設定している.突然変異率は $0.7-0.4 \cdot \frac{\sum_{i=1}^P Fe_i}{P}$ とし,環境個体の適応度の平均値に従って変化する.P は CP と EP に格納しうる最大個数である.

環境個体の適応度 Fe は,

$$Fe = \begin{cases} \frac{Fp}{Fr} (Fr \ge Fp) \\ \frac{Fr}{Fp} (Fr < Fp) \end{cases}$$
 (4.12)

とし,実評価値と先行評価値の差を縮める進化圧を与える.

個体プール内に CG と EG を各 $\operatorname{10}$ 個体ランダムに生成

while (終了条件非成立)

ルーレット選択で CG, EG を個体プールから各 2 個体選択評価する CG, EG を遺伝的操作により生成

CG を先行評価 (Fp の算出)

if (Fp が T 以上)

CG を実評価 (Fr の算出)

CG を Fr と共に自己個体プールに返還

EG を Fe と共に環境個体プールに返還

else

(実評価せずに) CG を Fp と共に自己個体プールに返還

endif

個体プールに風化操作

endwhile

CG: 自己個体

EG: 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)

T: 実評価実行か回避の閾値

図 4.3 提案手法における進化アルゴリズム (初期状態ではランダムに生成された自己個体が先行評価され,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.次に,自己個体プールからルーレット選択により 2 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同様の操作で,環境個体を作る.)

提案手法の進化のアルゴリズムを図 4.3 に示す. 閾値 T は自己個体プール内の自己個体の評価値の平均値に環境個体プール内の環境個体の適応度の平均値を掛けた値に調整パ

ラメータとして k を掛けた値

$$T = k \frac{(\sum_{i=1}^{P} Fr_i)(\sum_{i=1}^{P} Fe_i)}{P^2}$$
(4.13)

とする. 先行評価値が閾値より大きい場合は実評価し,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す.

なお,偶発的な高適応度の自己個体や進化初期のランダム性による高適応度の環境個体がプール内に残るのを防ぐために「風化」が導入されている [16].これは,自己個体プール内の自己個体,環境個体プール内の環境個体の適応度を,自己個体か環境個体がそれぞれの個体プールに戻る度に,一定の割合 w=1% で減少させるものである.

4.2 結果

提案するアーキテクチャを実進化型 2 足歩行ロボットシステムに適用し,評価した.その際,前節でも記述したように,関数の最適化 (PE1),ニューラルネットワークの最適化 (PE2) の 2 つの手法を用いた.それらを先行評価系導入前の実進化と比較する.加えて Simulation & transfer 手法との比較も行った.CP と EP のサイズは 10 とし,初期個体の数はそれぞれ 10 とした.突然変異率は 0.3,先行評価系の閾値パラメータは k=1.0,1 試行に必要なステップ数は N=4 とし,ニューロンの閾値 $\theta_1 \sim \theta_5$ は 0 とした.

自己個体の適応度の推移、 $\rm EE~(a),~\rm PE1~(b),~\rm PE2~(c)$ の場合の実評価値、先行評価値、先行評価の回数、を図 $\rm 4.4~c$ に示す.また、 $\rm x$ 軸にはロボットが転倒した実評価を示し, $\rm PE1$ と $\rm PE2$ の場合には棒グラフにより先行評価値が低く実評価を回避した回数も示している.

PE1 において,平均実評価値が約 650 に到達するのに必要な実評価回数が先行評価導入前後で約 20 から約 10 に減少したことがわかる.これは,ロボットの疲労を軽減することだけでなく,先行評価の時間が実評価に比べて無視できることから,時間の削減をも意味している.PE2 においては,EE よりも遅くなっている.実評価値においても,大きく値が振動しているが,同じ傾向が観測される.PE1 の性能が良かったのは,実評価回数約 20 の間に実評価を約 25 回回避していることから,ロボットが転倒する恐れのある低適応度行動を先行評価系により回避しているためと思われる.そのため,導入前は実評価回数約 20 までに約 10 回転倒しているのに対し,導入後は実評価回数約 20 までに約 2 回しか転倒していない.しかし,PE2 の場合,環境個体の進化は PE1 の場合よりも難しいために,自己個体の進化の速度が落ちてしまったと考えられる.今回のタスク設定では,転倒した場合としなかった場合の適応度の差が極端であるため,転倒する状況をうまく回避した場合,進化初期から実評価値が高くなる傾向がある.

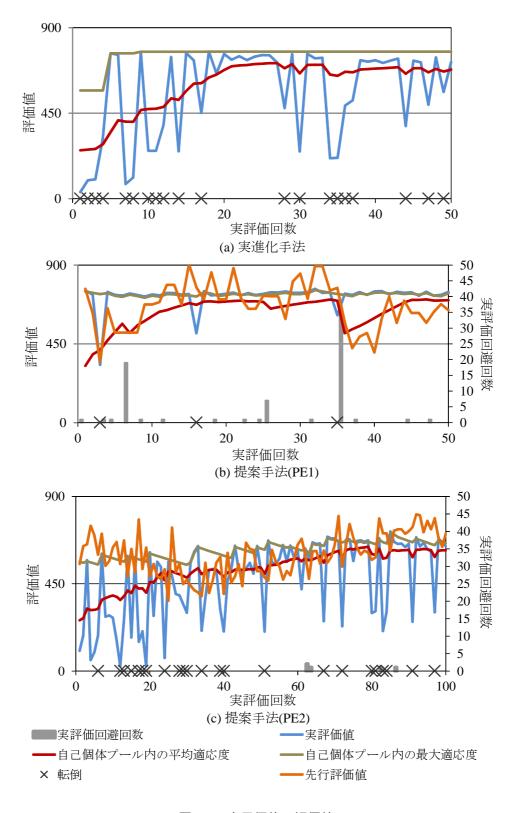


図 4.4 自己個体の評価値

先行評価値は実評価値に少しずつ近づいている.これは提案アーキテクチャにおいて共進化がうまく働いていることを意味する.しかし,それは実評価値よりも高い値を維持する傾向がある.これは環境個体の進化の特徴であり,適応度を上げる選択圧と実評価値と先行評価値の差を減少させる選択圧が常にかかっているが,先行評価値が基準値以下の場合,環境個体は環境個体プールに戻らずに破棄されるために,潜在的に先行評価値を上げる選択圧がかかっているためである.(図 3.1).

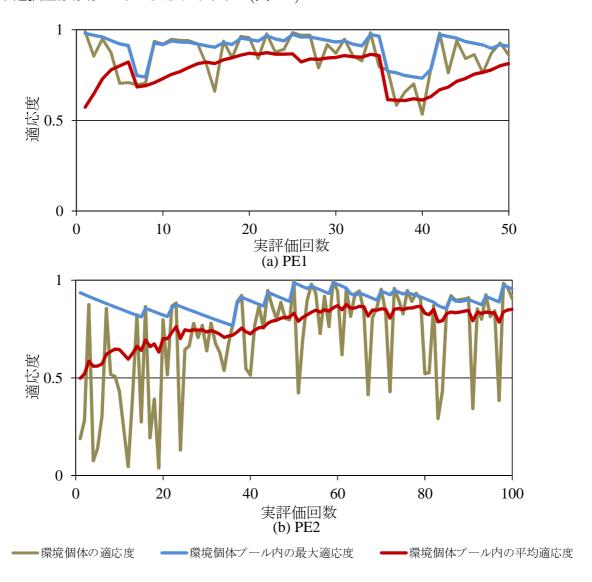


図 4.5 環境個体の適応度

PE1 (a) と PE2 (b) の場合の環境個体の平均適応度と最大適応度の推移を図 4.5 に示す、全般に比較的高い値を維持して実環境をうまく再現しているように見える、進化初期

から高適応度となることが多く,ある程度増減を繰り返しているが,これは,実評価値と 先行評価値の一致度を適応度としているので,特定の自己個体との適応度評価のみで高適 応度となっても必ずしも環境を的確に表しているとは限らないということに起因すると考 えられる.

また,PE1 の場合の評価中の自己個体の遺伝子データの推移を図 4.6 に,評価中の環境個体の遺伝子データの推移を図 4.7 に示す.横に傾ける距離 a は,ある値を超えると急に倒れるため,倒れない範囲に集まるが,倒れない範囲での変動はそれほど適応度に影響は与えないため, $0.018 \sim 0.022$ 付近で振動している.歩幅 c は値域内では大きければ大きいほど大きな前進につながり適応度に貢献するので,最大値 0.05 が多くなる.一方,高さd の適応度への影響はそれほど大きくないが,基本的には足は上げ過ぎない方が安定につながると推測され, $0.008 \sim 0.01$ 付近で振動を残しながらもほぼ収束している.

低適応度行動となる個体の実評価を先行評価系が回避し,転倒する恐れの少ない比較的近い遺伝子データを適応度評価しているため,全体的に値が大きくは変化しない.ただし,その環境個体の遺伝子データ値に依存し続けず,自己個体の進化に従って変化しており,適切に共進化していると考えられる.

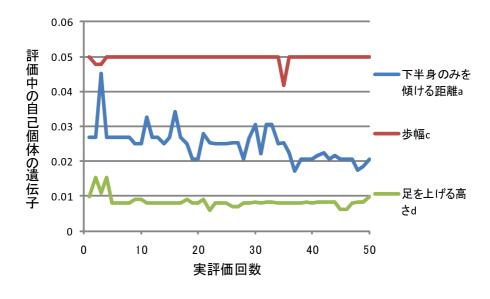


図 4.6 自己個体の遺伝子データの推移 (PE1)

さらに,PE1 の場合において,r=18000 のみ初期値を設定し,他の環境個体の遺伝子データ $(a_1,\,a_2,\,c_1,\,c_2,\,d_1,\,d_2)$ の初期値をまったくのランダムに生成した場合の環境個体の遺伝子データの推移を図 4.8 に,環境個体の適応度の推移を図 4.9 に示す.予想値を設定したときと比べて,値が大きく変動しているが,進化に従いある値に収束していること

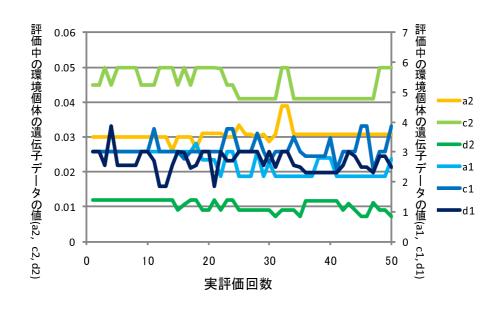


図 4.7 環境個体の遺伝子データの推移 (PE1)

がわかる.環境個体の適応度もほぼ大きい値を保っており,環境個体が実進化系との共進化が実現されていることが分かる.

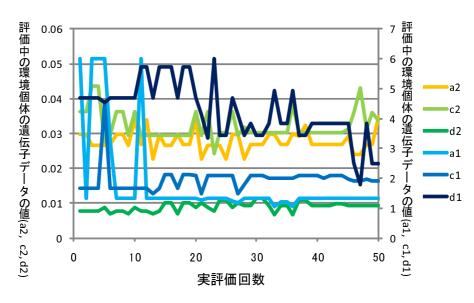


図 4.8 環境個体の遺伝子データの推移 (初期値ランダム設定) (PE1)

我々はさらに Simulation & transfer 手法の 1 つを評価した,実進化と同じ設定で,適応度評価を robotic platform OpenHRP によるロボットシミュレーションのみを用いて行い,その進化後の個体を実口ボットを用いて評価した.表 4.1 は 50 回適応度評価を行

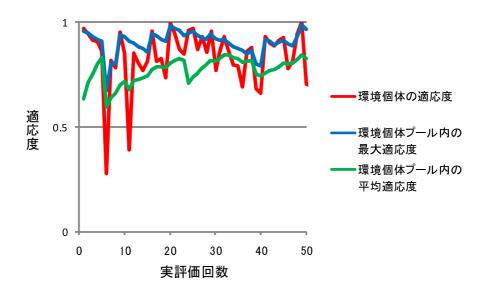


図 4.9 環境個体の適応度の推移 (初期値ランダム設定) (PE1)

い進化した後の自己個体プール内の 10 個体を全て示している . 1 列目はシミュレーション環境での適応度を示している . 2 列目は実口ボットを用いた場合の適応度を示している . 3 列目は実口ボットを用いた評価で転倒したかどうかを示している . 10 個体中 9 個体が実環境下での転倒が原因で大幅に適応度が減少している . これはシミュレーションと実環境との典型的な差異を示している .

適応度 830 814 806 797 781 769 766 751 743 735 (シミュレーション) 適応度 167 141149 149 135 149 752 182 166 166 (ロボット行動) 転倒 × X X X X X X × × (あり: x, なし:

表 4.1 シミュレーションと現実のギャップ (10 試行)

4.3 まとめ

提案アーキテクチャは、進化計算において適応度評価におけるコストやリスクを削減することを意図したものであり、そのコストやリスクが無視できないケース全般においてその枠組みは適用可能であると基本的には考えている。本章では、転倒などの弊害が伴う

ロボット行動の削減を狙って,ロボット行動を対象としている.その他にも,たとえば,ユーザの評価に時間がかかるインタラクティブ進化に適用して,その回数を減らし,ユーザの負担や解に達する時間を軽減することが期待できる.

ただし、先行評価系のユーザによる作り込みの度合いにより、提案手法の有効性は評価の観点ごとに大きく変わり、それは、先行研究の手法との比較において、明らかになる、そこで、提案手法の有効性に関する一般性と限界に関して、本研究で先行評価を導入した対象である Watson らの実進化手法、及び、実進化手法がギャップ問題解決を狙った対象である従来の進化ロボティクスの典型的手法の両者との比較に基づいて検討する、なお、以下の議論の一部は、提案アーキテクチャを、先行評価系を定義された関数のパラメータ(PE1)、ニューラルネットワークの重み(PE2)をそれぞれ進化的に最適化する手法を用い、実口ボットへ適用して得られた知見、第3章の提案アーキテクチャを可能な限り単純化して行った基本的評価から得られた知見も踏まえている。

提案手法 先行評価系の作り込み 実進化手法 従来手法 □ Max ギャップ問題 O 進化計算時間 O O[0 実評価による弊害 O 0 0 シミュレータ/先行評 \bigcirc 0 価系作成コスト

表 4.2 他手法との比較

表 4.2 は,従来二手法との比較に関して,ギャップ問題については有無で,他の評価の 観点については概ね見積もられる量をマルの大きさや線の太さで相対的に表したものであ る.先行評価系をどこまで作り込むかは第3章で述べたようにユーザ次第であり,実評価 系に関する知識をほとんど使わない場合から,先行評価系の一部のパラメータだけを探索 する場合まで考えられる.同表はそのような作り込みの度合いを線の太さで連続的に表し ている.

作り込みが大きいときは従来手法に近く,先行評価系作成コストも大きいが,進化計算時間と実評価による弊害は共に小さい.それを小さくしていくと,PE1による予め定義した関数の進化的なパラメータ調整,GPの初期個体をある程度作り込んで最適化等の場合が考えられ,先行評価系作成コストも比較的小さくなり,進化計算時間と実評価による弊害も共に小さく抑えられている.さらに,それが小さくなると,実進化に近づいていく

が,先行評価系がより良い解を得ることが難しくなり,PE2によるニューラルネットワークの重みをランダムに最適化,GPの初期個体をまったくのランダムな状態から最適化等の場合のように性能が減少する.つまり,進化計算時間と実評価による弊害の両者が先行評価系作成コストとトレードオフの関係になっていることが同表よりわかる.この点を踏まえて,先行評価系の作り込みの度合いを決めるべきである.

実進化では,進化初期の低適応度個体もロボット行動で評価するため,収束に時間を要し,不測の事態も引き起こしかねない.そこで,ロボット行動の前に先行評価し,低適応度とみなす個体による行動を回避する方式を提案し,実進化2足歩行ロボットを用いて,その基本的評価を行った.その結果,歩行の自律的獲得の初期に頻出する転倒を削減しうることが示された.このことは,ロボット疲労の軽減だけでなく,時間的コストの軽減も意味する.提案手法では,先行評価器は,その精度を適応度してロボット行動の制御器との共進化系を構成することにより,作成のコストの軽減を狙っているが,環境個体のやや特異な進化の様子も観察された.

第5章

適応度近似手法としての評価

第5章では,先行評価系を適応度評価近似手法として捉え直すことで一般化し,関数最大化問題に適用することで,実進化だけでなく幅広い問題への適用可能性を検証する.これは適応度評価近似手法の評価に関数最大化問題が使われている場合が多いからである.まず,適応度評価近似手法に関する背景を述べ,先行評価系の適応度評価近似手法としての性能を評価する.

5.1 概要

進化計算とは,ダーウィン進化のメカニズムを基にしたアルゴリズムを用いた,最適化手法であり,幅広く様々な分野で使用されている[1].しかし,最適解を得るまでに多くの適応度評価が必要で,適応度評価に高いコストやリスクが必要な問題においては,その適用が難しい.例えば,実ロボットを用いて適応度評価を行う場合や,ユーザの評価に時間がかかるインタラクティブ進化等である.そのような場合に多く用いられているのが,近似モデルを用いた適応度評価である.適応度評価の回数を減少させるために近似モデルを使うと効率が良い[17].

しかし,適応度評価に近似モデルを用いると,元の適応度評価を完全に再現した近似モデルを作成することはできないため,元の適応度評価とは異なる最適解に到達する等の問題が起きる.そのため,元の適応度評価と近似モデルによる適応度評価は同時に使う方が良い [17].そこで,近似モデルを用いて適応度評価を効率的に行う研究として進化制御が行われている.個体に基づく制御として,各世代でランダムに選択された個体のみに元の適応度評価を行う手法 [18] や各世代で適応度順に上位の個体のみを元の適応度評価を行う手法 [18] 等が行われ,世代に基づく制御として,何世代毎に元の適応度評価による

評価を行う世代を設定する手法 [20, 21] 等が行われている.

これらの手法は基本的に元の適応度評価を行う頻度は固定である.しかし,近似モデルの再現度は状況により大きく変化するかもしれず,これは実用的ではない.実際に,最適化の際,近似モデルとの差異の大きな変化が観測されている [20].そこで,適応的進化制御として,近似モデルの再現度に応じて元の適応度評価の頻度が変化する手法等の元の適応度評価を行う頻度が動的に変化する手法が提案されており [22, 23, 24],本論文で提案している先行評価系もその 1 手法として捉えることができる.

我々は元の進化系との共進化により近似モデルを自動生成することで,近似モデルの生成の手間を軽減し,先行評価器自体もその精度を適応度として進化するため,状況に応じた近似モデル生成の手間も軽減する.また,近似モデル自体も遺伝的プログラミング,ニューラルネットワーク,関数の係数を調整するなど,様々な表現が可能となっている.さらに,我々は共進化により生成した近似モデルを用いて,特に低適応度個体評価を回避するための手法を提案する.元の適応度関数による評価の前に近似モデル(先行評価系)による先行評価を行い,基準値以下の低適応度個体による評価には元の適応度関数を使わない.その際,これまでは実進化系と先行評価系の進化を同期させて行っていたが,本章では先行評価系の進化を,実進化系の進化の速度に合わせず,独立して行わせることで先行評価系の性能の向上を狙っている.これは実進化系の進化速度が,評価に時間がかかり,先行評価系の進化速度に比べて遅い場合に大幅な性能の向上が見込める.なお,先行研究において,適応度評価近似手法の評価には関数最大化問題が使われていることが多いため,本章でも適応度近似手法の評価に関数最大化問題を使用する.

先行評価系は進化計算により激しく変動するため必ずしも正確な近似はできないが,近似モデルを使った研究において,複雑な適応度地形においては,大まかに近似するほうがよいという結果が報告されている [25, 26]. モデルの設計の観点からも広範囲な近似よりも局所的な近似の方が現実的である [27]. そこで,我々の目的は低適応度個体評価を回避しつつ,最適解に到達することであり,必ずしも正確な近似を要求しない.そのため,進化的に獲得される先行評価系はある程度簡略化されたものとなることが想定される.

5.2 設定

先行評価系の概要を図 5.1 に示す.自己個体プール (CP) には自己個体 (CG) が適応度 (Fr) 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を行う.その評価結果が基準以下の個体は,元の適応度関数に

よる適応度評価を行わず,先行評価値 (Fp) 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,元の適応度評価関数により適応度評価され,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自己個体が進化する.

さらに,先行評価系を環境個体(EG)として表現し,環境個体プール(EP)に格納する. 先行評価系を環境個体(EG)として表現し,環境個体プール(EP)に格納しているが,そこから遺伝的操作により次世代環境個体を作成し,次世代環境個体プールに格納する. 適応度 (Fe)として,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る. これが 1 に近いほど良い個体であり,適応度値が大きくなる.このとき,実評価系の先行評価に使用する個体は環境個体プールの中で最も適応度が高い環境個体を使用する.そして,次世代環境個体プール内の全環境個体の先行評価値を算出し,最近行われた実評価の自己個体と実評価値の組み合わせから環境個体の適応度を算出し,その適応度と共に全環境個体を環境個体プールに戻す.この一連の流れを繰り返すことで環境個体が進化する. 環境個体で何を表現して先行評価系を構成するかについては,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探索するケースまで考えられる.

我々が第3章で提案していたアーキテクチャ(図3.1)では実進化系と先行評価系の進化の両方にオンライン進化を適用していたが,本論文の提案アーキテクチャでは先行評価系の進化が通常の遺伝的アルゴリズムによる進化の形に拡張している.これは実進化系において評価に時間がかかるタスクを行う場合でも,その終了を待つことなく先行評価系の進化を進められ効率が良い.

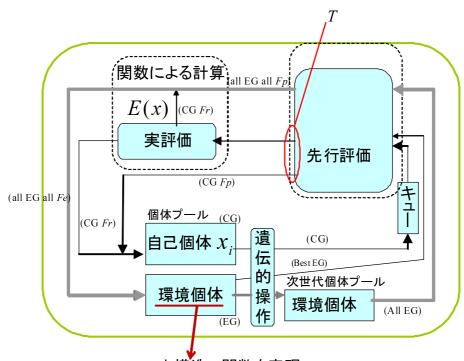
先行評価系を有した数学的なモデルを構築し解析する.遺伝的プログラミング (GP) を用いた先行評価系を構築し,関数最大化を行いその性能を比較する.その際,提案アーキテクチャの他に従来から行われている手法として最小2乗法を用いて,実評価の履歴を全て記録し,実評価関数の近似関数を作成し,それを用いた評価と実評価をランダムに行う手法とも性能を比較した.

その際,実評価系の適応度関数には,大域的単峰性関数として

$$E(x) = 71 - \frac{x^2}{4000} + a(\cos bx - 1), (-512 \le x \le 512)$$

$$max(E(x)) = E(0) = 71$$
(5.1)

を用い, $a=3,\,b=1$ とする. $a,\,b$ は関数の形を決める係数で,a が大きくなると谷が深くなり,b が小さいほど谷間の幅が広くなる.例えば, $a=1,\,b=1$ なら小さな谷が複数ある多峰性関数となり, $a=0,\,b=1$ なら単純な放物線状の 2 次関数となる.



木構造で関数を表現 適応度:実評価値と先行評価値の一致度

CG: 自己個体

EG: 環境個体

CP: 自己個体プール

EP: 環境個体プール

Fr: 実評価値

Fp: 先行評価値

Fe: 環境個体の適応度 (実評価値と先行評価値の一致度)

T: 実評価実行か回避の閾値

図 5.1 先行評価系を有する手法の拡張

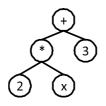


図 5.2 木構造

GP による先行評価系の場合 , 先行評価値 Fp は GP により生成された木構造で計算される . 環境個体として数値と四則演算子からなる図 5.2 のような木構造をいくつか生成する . 環境個体の適応度 Fe は ,

$$Fe = \begin{cases} \frac{Fp}{Fr} (Fr \ge Fp) \\ \frac{Fr}{Fp} (Fr < Fp) \end{cases}$$
 (5.2)

とし,実評価値と先行評価値の差を縮める進化圧を与える.これにより,実評価系により近い関数になるように環境個体が進化する.その際,突然変異として,ノードが数値の場合は値を増減させ,演算子の場合はランダムに他の演算子に変化させる.また,0.001 の確率で新しいノードを図 5.3 のように発生させる.交叉は図 5.4 のように交叉点をランダムに選択し,木構造を入れ替える.

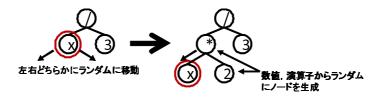


図 5.3 新しいノードの生成

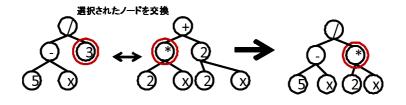


図 5.4 木構造の交叉

閾値 T は自己個体プール内の自己個体の評価値の平均値に環境個体プール内の環境個

体の適応度の平均値を掛けた値に調整パラメータとして k を掛けた値

$$T = k \frac{(\sum_{i=1}^{P} Fr_i)(\sum_{i=1}^{P} Fe_i)}{P^2}$$
 (5.3)

とする. 先行評価値が閾値より大きい場合は実評価し,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す.

なお,偶発的な高適応度の自己個体や進化初期のランダム性による高適応度の環境個体がプール内に残るのを防ぐために「風化」が導入されている [16].これは,自己個体プール内の自己個体,環境個体プール内の環境個体の適応度を,自己個体か環境個体がそれぞれの個体プールに戻る度に,一定の割合で減少させるものである.

5.3 結果

提案手法を関数最大化に適用し,拡張前の提案手法と最小 2 乗法を用いた手法とその性能を比較した.自己個体と環境個体プールに格納しうる最大個数 P は 30 とする.結果は全て 20 試行平均を示している.実評価系のパラメータ設定は,初期値は $412 \sim 512$ の乱数で生成,突然変異は- $50 \sim 50$ の乱数を加算し,突然変異率は 0.2,風化率は 0.0001 である.GP を用いた先行評価系では,初期個体の設定は,ランダム $(-30 \sim 30$ の乱数と四則演算子からランダムに作成した深さ 3 の木構造),1 次関数 $((35+35)+(-0.13\times x))$,2 次関数 $((35+35)+(-0.00025\times(x\times x)))$ の 3 種類をそれぞれ試した.設定パラメータはそれぞれ,初期個体がランダムの場合は,突然変異率 0.01,交叉率 0.02,風化率 0.01,先行評価系の閾値パラメータ k=1.1,初期個体が 1 次関数の場合は,突然変異率 0.01,交叉率 0.005,風化率 0.01,先行評価系の閾値パラメータ k=1.2,初期個体が 2 次関数の場合は,突然変異率 0.01,交叉率 0.005,風化率 0.01,先行評価系の閾値パラメータ 0.01,表行評価系の閾値パラメータ 0.01,表行評価系の 0.01,先行評価系の閾値パラメータ 0.01,表行評価系の 0.01,先行評価系の閾値パラメータ 0.01,表行評価系の 0.01,先行評価系の閾値パラメータ 0.01,表行評価系の 0.01,表行評価系の閾値パラメータ 0.01,表行評価系の 0.01,表行評価系の閾値パラメータ 0.01,表行評価系の 0.01,表行評価系の閾値パラメータ 0.01,表行評価系の 0.01,表行評価系の閾値パラメータ 0.01,表行評価系の 0.01,表別の 0.01,表別の 0.01,表行評価系の 0.01,表別の 0.01,表行評価系の 0.01,表別の 0.01,表別の

提案手法の自己個体の評価値の推移を図 5.5 に拡張前の提案手法と最小 2 乗法を用いた手法のものと共に示す.最小 2 乗法を用いた手法も先行評価系導入前に比べて適応度上昇のための実評価回数が減少し,拡張前の提案手法よりも減少しているが,提案手法の方が,初期個体が 1 次関数 , 2 次関数の場合において,大幅に適応度上昇のための実評価回数が減少している.初期個体を 1 次関数のようにアバウトな近似にした場合でも大幅な性能の向上が見られることから,厳密な近似は必ずしも必要がないことが分かる.拡張前の提案手法において初期個体がランダムな場合,先行評価導入前よりも適応度上昇のための

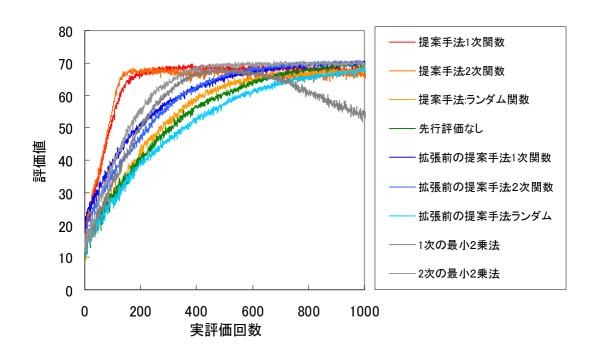


図 5.5 提案手法と拡張前の提案手法と最小 2 乗法を用いた手法の自己個体の評価値

実評価回数が増加してしまっていた.これは,環境個体がランダムな状態から自己個体を再現できるようになるのにある程度の時間を要したためと考えられる.しかし,提案手法においては初期個体がランダムな場合でも先行評価導入前よりも適応度上昇のための実評価回数が減少しており,環境個体の進化がより効率的に行われていることが分かる.

提案手法の環境個体の適応度の推移を図 5.6,拡張前の提案手法の環境個体の適応度の推移を図 5.7 に示す.初期個体がランダムな場合,拡張前の提案手法に比べて拡張後の提案手法では適応度の上昇が速くなっているのが分かる.全般に比較的高い値を維持して実環境をうまく再現しているように見える.但し,前章での図 4.5 と同様,実評価値と先行評価値の一致率を適応度としているので,特定の自己個体との適応度評価のみで高適応度となっても必ずしも環境を的確に表しているとは限らないという理由で進化初期から高適応度となることが多く,ある程度増減を繰り返していると考えられ,環境個体の特異な進化と考えられる.

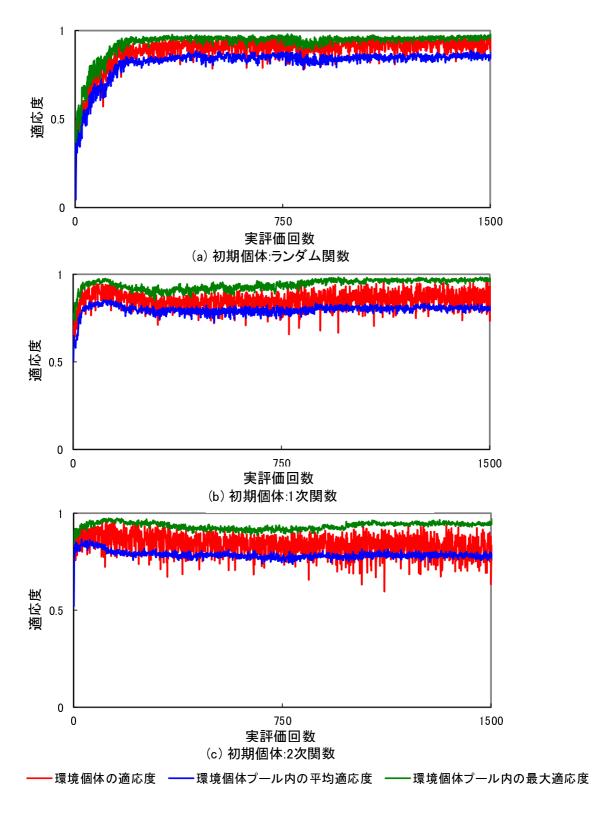


図 5.6 提案手法の環境個体の適応度

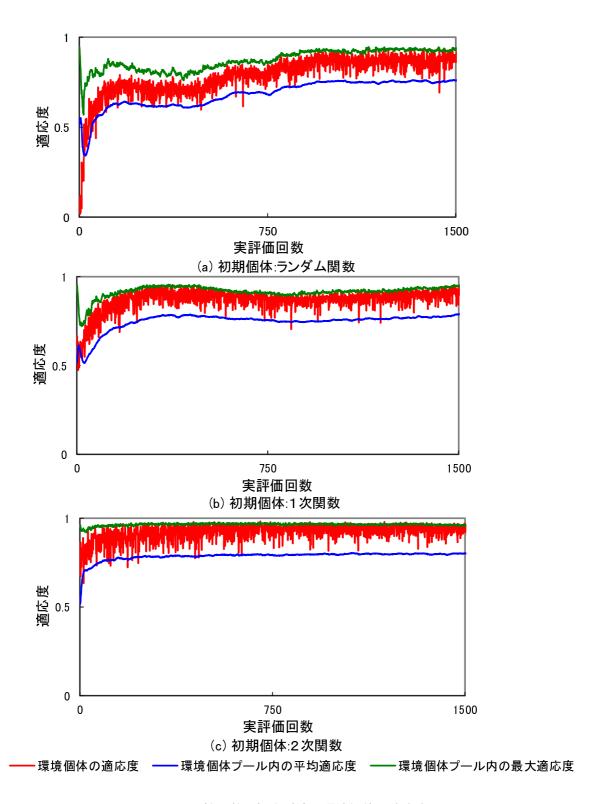


図 5.7 拡張前の提案手法の環境個体の適応度

5.4 まとめ

我々は,適応度評価に高いコストやリスクが必要な問題において,適応度近似手法を用いることで特に低適応度個体評価を回避するための手法を提案した.そのために,実評価の前に先行評価を行うことで,低適応度個体評価を回避させ,その先行評価器自体もその精度を適応度として元の進化系と共進化させた.

GPによる関数近似手法を用いた先行評価系を関数最大化問題に適用し,その性能を比較した.その際,初期個体の状態がランダム,1次関数,2次関数とそれぞれ先行評価系の作り込みの度合いが違う状態で評価した.その結果,提案手法は最小2乗法を用いた手法と以前の我々の手法よりも低適応度評価の回数を削減しうることが示された.提案手法は適応度評価近似手法としても正常に動作し,進化計算においてそのコストやリスクが無視できないケース全般において,適応度評価におけるコストやリスクを削減するために,その枠組みは適用可能であると考えられる.

第6章

おわりに

実進化方式では,進化初期の低適応度個体もロボット行動で評価するため,収束に時間を要し,不測の事態も引き起こしかねない.そこで,ロボット行動の前に先行評価し,低適応度とみなす個体による行動を回避する方式を提案した.先行評価器もその精度を適応度して制御器との共進化系を構成した.

まず、基本的評価により進化の振舞いを調べ、提案アーキテクチャの複雑な進化ダイナミクスの理解とアーキテクチャの改良を行った。その結果、先行評価系を導入することで、基準適応度に達するまでに必要な実評価回数を削減可能なことを示した。また、先行評価系の環境個体の適正な進化のために、進化初期のランダム性に基づく高適応度の環境個体を遺伝子プールから徐々に排除して再評価を促すこと(「風化」)が重要であることが分かった。さらに、自己個体プール中の各個体の先行評価値と実進化値に差が生じた場合に進化が停滞するが解決可能であることが分かった。具体的には、先行評価値が実評価値より大きい個体が増加すると、不良個体の優先的選択により進化が減速し、逆に、小さい個体が増加すると、選択され先行評価されても、集団の適応度の平均より低くなって実評価が行われずに進化が停滞する。実評価するかを決定する基準値を環境個体の適応度に応じて減少させれば、前者の場合は優遇される個体の割合を減らし、後者の場合は先行評価を通過しやすくすることで、両者の発生が抑制可能であることが分かった。

そして,先行評価系を実進化型2足歩行ロボットに導入し,歩行動作の自律的獲得の初期に頻出する転倒の回数の削減を狙った.その結果,歩行の自律的獲得の初期に頻出する転倒を先行評価導入前に比べて約8割削減した.これは,ロボット疲労の軽減だけでなく,時間的コストの軽減も意味している.環境個体の,適応度が全般に比較的高い値を維持し値が上下するという,やや特異な進化も観測された.これは先行評価系の,実評価値と先行評価値の一致率を環境個体の適応度にするなどの,特徴的な設定に起因すると推測

される.

提案手法は,実進化と比較すると実評価による弊害と進化計算時間を削減でき,Simulation & transfer 手法と比較すると,現実とのギャップによる問題の回避とシミュレーション構築のコストを軽減できると考えられるが,進化計算時間と実評価による弊害の両者と先行評価系作成コストのトレードオフの関係から先行評価系をどこまで作りこむかが重要となる.また,提案アーキテクチャは,進化計算において適応度評価におけるコストやリスクを削減することを意図したものであり,そのコストやリスクが無視できないケース全般においてその枠組みは適用可能であると基本的には考えられる.

さらに,先行評価系を適応度評価近似手法として捉え直すことで一般化し,関数最大化問題に適用することで,実進化だけでなく幅広い問題への適用可能性を検証した.その際,適応度評価に多くの時間が要求される場合を想定し,適応度評価を行う間に,環境個体の進化が効率よく行われるよう,提案手法の拡張を行った.その結果,最小2乗法をシンプルに用いた適応度評価近似手法よりも低適応度個体による適応度評価回数は減少し,より速く高適応度に到達した.このことから,適応度評価の近似手法としても正常に動作し,汎用性を持つ可能性を示した.

事前に行動を選別すると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,人の 行動獲得における脳の働きと類似した側面を持っており,人の知能の形成という観点から も興味深い.さらに,実進化に限らず,インタラクティブ進化等,適応度評価にコストや リスクが要求されるタスクへの幅広い応用も考えられる.

謝辞

本研究は、名古屋大学大学院情報科学研究科において行われたものである.指導教官ならびに主査である有田隆也教授、鈴木麗璽助教授には貴重な時間を割いて御指導をして頂きました.誠にありがとうございました.また、副査である大岡昌博教授には草稿段階で貴重な時間を割いて御指導して頂き、有益なコメントを頂きました.誠にありがとうございました.さらに、有田隆也研究室、鈴木麗璽研究室のメンバーの皆様には日々議論をさせて頂きました.感謝いたします.

参考文献

- [1] Nolfi S and Floreano D: Evolutionary Robotics, MIT Press, 2000.
- [2] Brooks R: Artificial life and real robots, Varela F and Bourgine P (eds.), *Proc.* of the First European Conference on Artificial Life, pp. 3-10, MIT Press, 1992.
- [3] Mataric M and Cliff D: Challenges in evolving controllers for physical robots, Robotics and Autonomous Systems, Special Issue on Evolutional Robotics, Vol. 19, No. 1, pp. 67-83, 1996.
- [4] Jakobi N: Evolutionary robotics and the radical envelope of noise hypothesis, *Adaptive Behavior*, Vol. 6, No. 1, pp. 131-174, 1997.
- [5] Jakobi N: Half-baked, ad hoc, and noisy: minimal simulations for evolutionary robotics, Husbands P and Harvey I (eds.), Fourth European Conference on Artificial Life, pp. 348-357, MIT Press, 1997.
- [6] Watson RA, Ficici SG and Pollack JB: Embodied Evolution: Embodying an Evolutionary Algorithm in a Population of Robots, Proc. of the 1999 Congress on Evolutionary Computation, pp. 335-342, 1999.
- [7] Watson RA, Ficici SG and Pollack JB: Embodied Evolution: Distributing an Evolutionary Algorithm in a Population of Robots, *Robotics and Autonomous Systems*, Vol. 39, No. 1, pp. 1-18, 2002.
- [8] Schut MC, Haasdijk E and Eiben AE: What is Situated Evolution?, *IEEE Congress on Evolutionary Computation*, pp. 3277-3284, 2009.
- [9] Schut MC, Haasdijk E and Prieto A: Is situated evolution an alternative for classical evolution?, *Proc. of the Eleventh conference on Congress on Evolutionary Computation*, pp. 2971-2976, 2009.
- [10] Usui Y and Arita T: Situated and Embodied Evolution in Collective Evolutionary Robotics, Proc. of the Eighth International Symposium on Artificial Life and

- Robotics, pp. 212-215, 2003.
- [11] Elfwing S, Uchibe E, Doya K and Christensen H: Biologically Inspired Embodied Evolution of Survival, Proc. of the IEEE Congress on Evolutionary Computation 2005, Vol. 3, pp. 2210-2216, 2005.
- [12] Perez ALF, Bittencourt G and Roisenberg M: Embodied evolution with a new genetic programming variation algorithm, icas, Vol. 0, pp. 118-123, 2008.
- [13] Simões EDV and Dimond KR: Embedding a distributed evolutionary system into population of autonomous mobile robots, *Proc. of the 2001 IEEE Systems*, Man., and Cybernetics Conference, 2001.
- [14] Walker JH, Garrett SM and Wilson MS: The balance between initial training and lifelong adaptation in evolving robot controllers, *IEEE Transactions on Systems*, Man, and Cybernetics, Part B, Vol. 36, No. 2, pp. 423-432, 2006.
- [15] Nehmzow U: Physically embedded genetic algorithm learning inmulti-robot scenarios: The pega algorithm, Prince CG, Demiris Y, Marom Y, Kozima H, and Balkenius C (eds.), Proc. of The Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, No. 94 in Lund University Cognitive Studies, Edinburgh, UK, Aug., LUCS, 2002.
- [16] Nakai J and Arita T: A framework for embodied evolution with pre-evaluation applied to a biped robot, Artificial Life and Robotics, Vol. 15, No. 2, pp. 156-160, 2010.
- [17] Jin Y: A comprehensive survey of fitness approximation in evolutionary computation, *Soft Computing 9*, pp. 3-12, 2005.
- [18] Jin Y, Olhofer M and Sendhoff B: On evolutionary optimization with approximate fitness functions, *Proc. of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, pp. 786-792, 2000.
- [19] Grierson DE and Pak WH: Optimal sizing geometrical and topological design using agenetic algorithm, *Structural Optimization*, Vol. 6, No. 3, pp. 151-159, 1993.
- [20] Ratle A: Accelerating the convergence of evolutionary algorithms by fitness land-scape approximation, Eiben A, Bäck Th, Schoenauer M and Schwefel H-P (eds.), Parallel Problem Solving from Nature, V, pp. 87-96, 1993.
- [21] Ratle A: Optimal sampling strategies for learning a fitness model, *Proc. of 1999 Congress on Evolutionary Computation*, Vol. 3, pp. 2078-2085, Washington DC,

- 1999.
- [22] Dennis J and Torczon V: Managing approximate models in optimization, Alexandrov N, Hussani M (eds.), *Multidisciplinary design optimization: State-of-the-art*, SIAM, pp. 330-347, 1997.
- [23] Nair PBand Keane AJ: Combining approximation concepts with Proc.algorithm-based structural optimization procedures, 39thAIAA/ASMEASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, pp. 1741-1751, 1998.
- [24] Jin Y, Olhofer M and Sendhoff B: Managing approximate models in evolutionary aerodynamic design optimization, *Proc. of IEEE Congress on Evolutionary Computation*, Vol. 1, pp. 592-599, 2001.
- [25] Liang K-H, Yao X and Newton C: Combining landscape approximation and local search in global optimization, 1999 Congress on Evolutionary Computation, pp. 1514-1520, 1999.
- [26] Liang K-H, Yao X and Newton C: Evolutionary Search of approximated n-Dimensional Landscape, International Journal of Knowledge-based Intelligent Engineering Systems, Vol. 4, No. 3, pp. 172-183, 2000.
- [27] Ong YS, Nair PB and Keane AJ: Evolutionary optimization of computationally expensive problems via surrogate modeling, *AIAA Journal*, Vol. 41, No. 4, pp. 687-696, 2003.