

捕食-被食関係にある進化型エージェントによる適応度景観の特徴量推定 Estimation of Fitness Landscapes by Evolutionary Predator and Prey Robots

摂南大学 ○ 半田 佑太, 片田 喜章

○ Y. Handa and Y. Katada, Setsunan University

Abstract Open-ended evolution is considered to be caused by several factors, one of which would be co-evolution. Competitive co-evolution can give rise to the “Red Queen effect”, where the fitness landscape of each population is continuously changed by the competing population. Therefore, if such changes are captured, co-evolutionary progress can be measured. In this paper, we estimate features of competitive co-evolutionary fitness landscapes on predator and prey robots with different sensory and motor characteristics in computer simulations and investigate the Red Queen effect on the fitness landscape.

1 はじめに

競争型共進化において、種が存続するためには進化し続ける必要があり進化をやめるとその種は絶滅する「赤の女王仮説」が提唱されている [1]。ここでは、ある種の利益は競争関係にある種の損失となり、その種が利益を得るため対抗戦略を講じることを互いに繰り返す軍拡競争として現れる。

この軍拡競争により競争型共進化は、適応度景観が絶え間なく変化し、永続的進化を導くと考えられる。したがって、適応度景観の変化を捉える、つまり、景観の特徴量が変化し続けることを観察できれば、人工進化における共進化の発生を確認できるのではないかと考えた。文献 [2] において、我々は適応度景観の推定法 [3] を用い、形態の等しい捕食-被食関係にある 2 つの個体集団における適応度景観の特徴量を推定し、その変遷を調べることで、競争型共進化の発生について検証を行った。しかしながら、その結果は上述の予想に反し、多くの場合、適応度景観の特徴量の変化が世代を経るにつれ次第に小さくなっていくというものであった。この原因は、捕食エージェントが被食エージェントに対し常に高い適応度を示し、形態を等しく設定することが捕食エージェントにとって有利となっていることにあると推測される。そこで本研究では、計算機実験において形態の異なる捕食-被食関係にある 2 つの個体集団を用いて [2] と同様の実験を行い、景観の特徴量を推定し、競争型共進化の発生について検証する。

2 適応度景観の特徴量推定

適応度景観の特徴量には凸凹度 (ruggedness) と中立度 (neutrality) があることが知られている [4]。本稿では、

エージェントのコントローラに次節で述べる進化型人工ニューラルネットワーク (Artificial Neural Networks: ANN) を用いるため、その適応度景観にはこの 2 つの特徴量が含まれると考えられる。これらを推定する指標について次に述べる。

2.1 凸凹度の指標

実問題における凸凹度は適応度景観の相関に基づいて推定される。そこで凸凹度の指標として、その傾きが適応度景観の相関を表す Smith の E_b [4] を用いる。

適応度 k の親個体から生まれる子個体の平均適応度を表す E_b は次のように定義される。

$$E_b = \frac{\sum_{g \in G(k)} V(g)}{|G(k)|} \quad (1)$$

ここで、 $G(k)$ は適応度 k の親から生まれる子の遺伝子型集合、 g は子の遺伝子型、 $V(g)$ は子の適応度である。この $E_b(k)$ は中立度の値の影響を受けず、凸凹度が増加すると k に対する E_b の傾き (\dot{E}_b) は減少することが知られている [4]。この傾きは遺伝的操作として突然変異を用いた場合の自己相関関数に比例し、凸凹度が最も小さいとき $\dot{E}_b \simeq 1.0$ に、凸凹度が最も大きいとき $\dot{E}_b \simeq 0.0$ になる [4]。

2.2 中立度の指標

中立度の指標 [3] に用いる根井の標準遺伝距離 [5] は以下のように定義される。集団 X における第 i 番目の遺伝子座の対立遺伝子 l の頻度を $x_{il} = n_{il}/M$ ($i = 1, \dots, L$, L : 遺伝子長, M : 個体数, n_{il} : 集団内の対立遺伝子 $l \in \{0, 1\}$ の数) とする。同様に集団 Y における y_{il} を考え、 $j_{xi} = x_{i0}^2 + x_{i1}^2$, $j_{yi} = y_{i0}^2 + y_{i1}^2$, $j_{xyi} = x_{i0}y_{i0} + x_{i1}y_{i1}$ とする。ここで、 j_{xi} は集団 X から任意に選んだ 2 個体

における第 i 番目の遺伝子座の対立遺伝子が等しい確率であり, j_{yi} は集団 Y のそれである. j_{xyi} は集団 X と Y のそれぞれから任意に 1 個体ずつ選んだ 2 個体における第 i 番目の遺伝子座の対立遺伝子が等しい確率である. これらは各遺伝子座に対し定義されるので, 全遺伝子座の平均は, $J_X = \sum_{i=1}^L j_{xi}/L$, $J_Y = \sum_{i=1}^L j_{yi}/L$, $J_{XY} = \sum_{i=1}^L j_{xyi}/L$ となる. このとき, 根井の標準遺伝距離は次のように与えられる.

$$D = -\log_e \frac{J_{XY}}{\sqrt{J_X} \sqrt{J_Y}} \quad (2)$$

集団遺伝学では, 突然変異率が十分に小さいことを仮定している. また, 集団 X と Y の選び方として, t 世代での個体集団と $t+1$ 世代での個体集団を用いる. そして, 世代間の遺伝距離 $D_{t,t+1}$ を求め, T 世代までの総和を GA における遺伝距離として以下のように拡張する.

$$D(T) = \sum_{t=1}^{T-1} D_{t,t+1} \quad (3)$$

この遺伝距離は中立度が大きいほど増加し, 凸凹度が大きいほど減少するという特性をもつ [3]. このことを利用し, 世代数に対する遺伝距離の変化率 α を世代毎にそれまで獲得した遺伝距離のデータを基に最小二乗法を用いて求め, 式 (1) の傾き E_b と併せて, 中立度の指標とする.

3 計算機実験

3.1 実験設定

競争関係にある個体集団として捕食者集団 (Pr) と被食者集団 (Py) を仮定する. 実験環境は Fig. 1 のように設定し, 周囲は壁に囲まれているものとする. 1. でも述べたように, 両エージェントの形態は異なるように設定する. エージェントは壁用と他エージェント用の 2

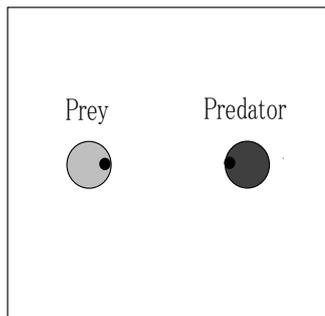
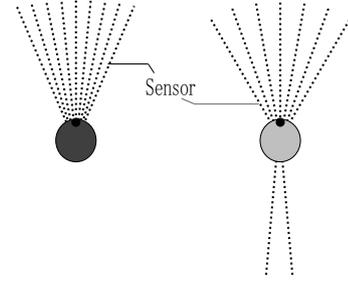
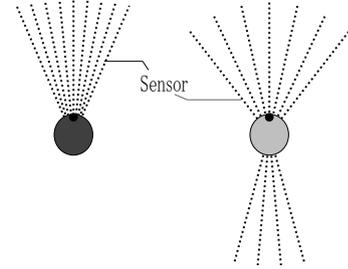


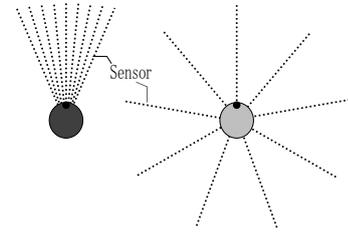
Fig. 1: Experimental setup for a pursuit-evasion contest



(a) SP1



(b) SP2



(c) SP3

Fig. 2: Simulated models of sensors for predator and prey robots

種類の距離センサをもち, その配置を SP1(Fig. 2(a)), SP2(Fig. 2(b)), SP3(Fig. 2(c)) とする. 移動機構として, 左右二輪独立駆動型を用い, 移動速度の最大値は, Pr に対し Py が 2 倍の最大速度を持つ ($V_{max,py} = V_{max,pr} \times 2$) ように設定する. エージェントの制御器として, 中間層・出力層が完全相互結合された ANN を用いる. 入力ノード数は $N_s = 18$ (センサ数), 出力ノード数は $N_o = 2$ (左右モータの周速度), 中間ノード数は $N_h = 3$ とする. Fig. 1 の初期位置から, エージェントの初期姿勢をそれぞれ 90° ずつ変えた試行を 16 回行う. Pr はより早く Py を捕えることを, Py はより長く Pr の追跡から逃れることを目的とし, 次の適応度関数を用いてそれぞれの適応度を計算する.

$$F_{pr} = \sum_{i=1}^{Trials} \frac{1 - Step_i / MaxStep}{Trials}, \quad (4)$$

$$F_{py} = \sum_{i=1}^{Trials} \frac{Step_i / MaxStep}{Trials} \quad (5)$$

ここで、*Trials* は試行回数 (16) であり、*Step* はエージェント同士が接触するまでのステップ数、*MaxStep* は最大のステップ数で 400 とする。進化手法として Simple GA (SGA) を用いる。ANN の結合荷重を変数とし、10 ビット・バイナリコーディングを行う。遺伝子長は $L = 1150$ となる。遺伝的操作として点突然変異を用い、突然変異率は、2. で述べたように、十分に小さく設定する必要があることから、 $q = 1/L$ とする。選択法としてトーナメント選択を用い、トーナメントサイズは 2 とする。競争する対戦相手の個体の選択方法は、同世代の競争関係にある集団の中からランダムに 1 個体を選択する方法 (SG), 1 世代前の最良個体を選択する方法 (BP1), 10 世代前までの各世代の最良個体と対戦しその平均を取る方法 (BP10) を用いる。すべての設定において、両集団ともに個体数を 50, 世代数を 5000 とし、独立に 10 試行行う。

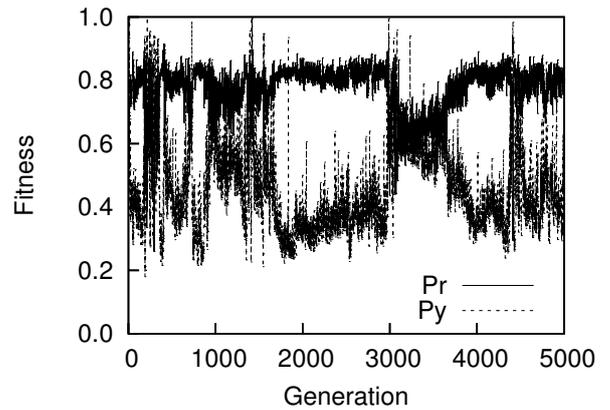
3.2 実験結果と考察

3.2.1 最大適応度の推移

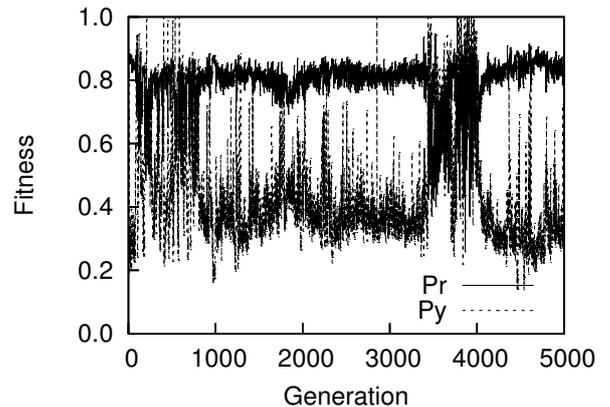
Fig. 3-5 に、対戦相手の選択方法に対する各センサパターンにおける、世代に対する最大適応度の推移の 10 試行中の典型例を示す。全ての設定において、世代を通して Pr の適応度が高くなる傾向にあるが、Py の適応度が高くなる世代も見られ、そのときは Pr の適応度が低くなっている。このときのエージェントの振舞いを観察すると、最大速度の違いから、Py が Pr を周回し、Pr に捕えられないことなく逃げ切る行動がみられる。この結果から、エージェントの形態に差を与えることで、両集団で交互に適応度が改善されることがわかった。

3.2.2 適応度景観の特徴量

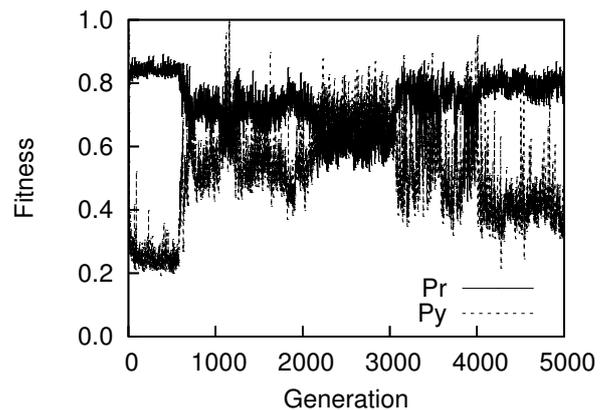
Fig. 6-8 に、Fig. 3-5 で示した試行における各世代に対する (\dot{E}_b, α) を示す。形態に差を与えることによって、両集団において最大適応度にも変化があったこと (前節参照) から、適応度景観の特徴量は変化し続けているという予想をしていた。しかし、[2] と同様に、 (\dot{E}_b, α) の値の変化は両集団ともに約 1000 世代以降でほとんど変化しなくなっている。初期 1000 世代の間では、 \dot{E}_b, α ともに大きく変化しているが 1000 世代以降になると、世代を経る毎に (\dot{E}_b, α) の値の変化が小さくなっており、約 3000 世代以降では変化はほとんど見られない。本実験設定では、遺伝距離の特性のために、突然変異率を $q = 1/L$ と非常に小さく設定している。このことから、種間に競争関係を設定し両種を進化させた場合でも、突然変異率が非常に小さいときは、世代を経るにつれて集団がそれぞれの遺伝子型空間で収束していくのではないかと考えられる。



(a) SP1



(b) SP2

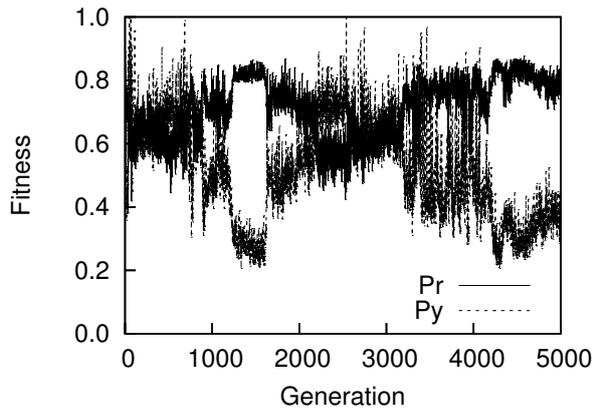


(c) SP3

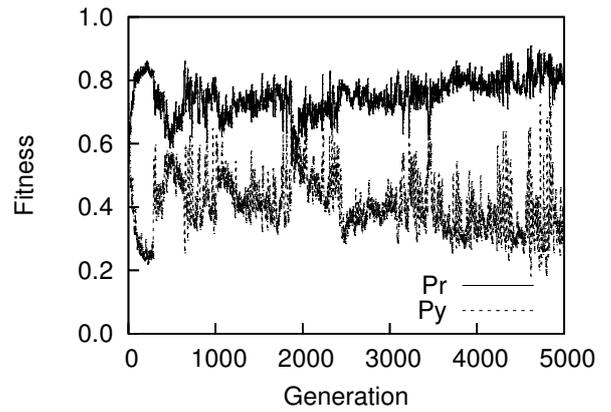
Fig. 3: Maximum fitness at each generation (SG)

3.2.3 過去の世代との比較

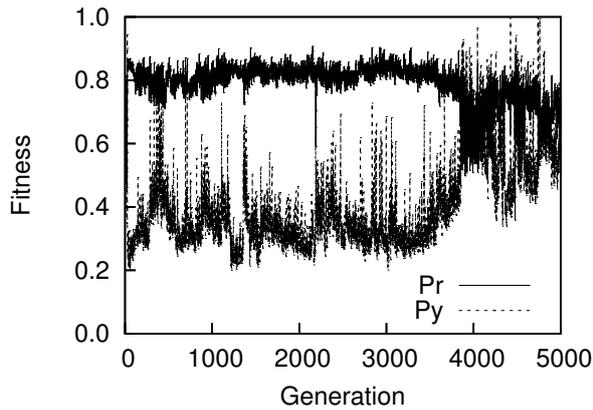
前節の実験結果では、適応度景観の特徴量は約 1000 世代以降ではほぼ変化が見られなくなっており、この結果から適応度景観が常に変化しているとは言えず、共進化の発生を十分に確認できない。これを別の方法で確認するため、Floreano ら [6] が行っている、各世代の最良個体を過去の世代の最良個体と対戦させる方法を用いる。



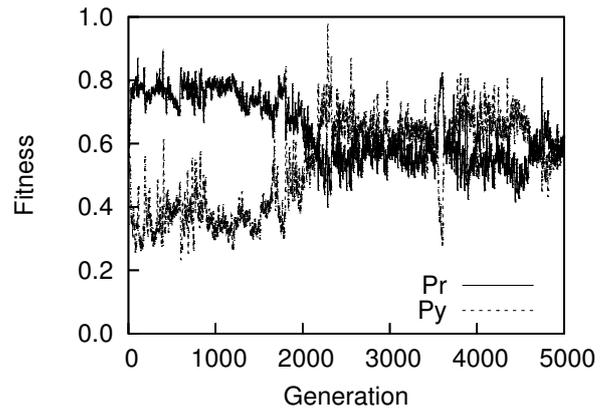
(a) SP1



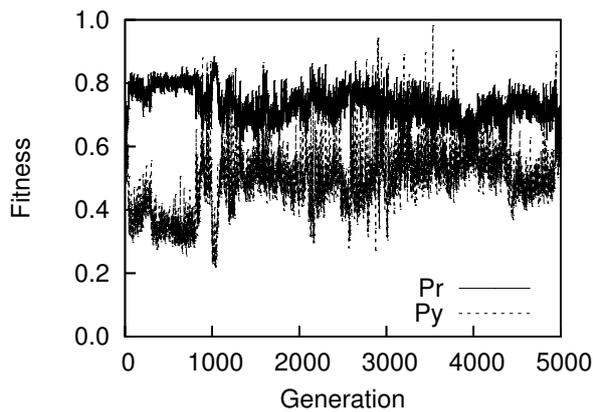
(a) SP1



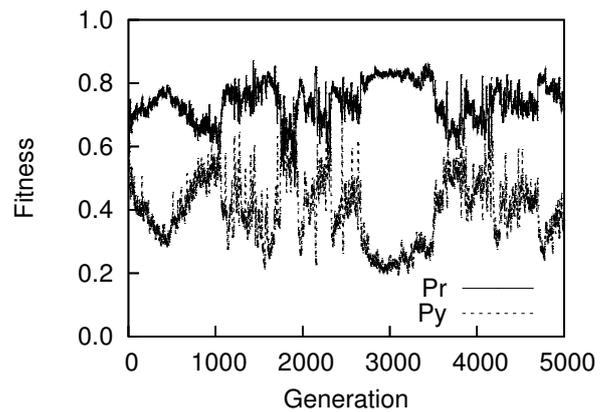
(b) SP2



(b) SP2



(c) SP3



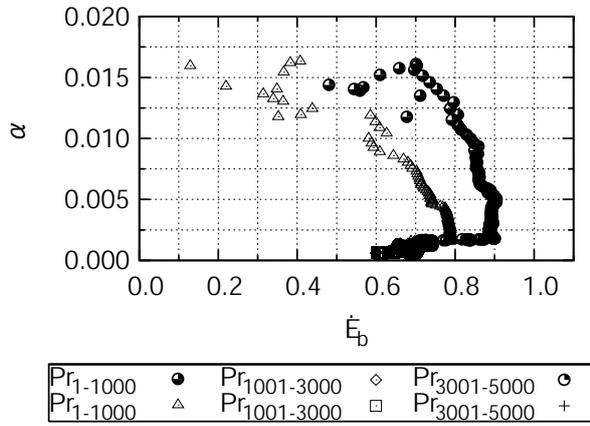
(c) SP3

Fig. 4: Maximum fitness at each generation (BP1)

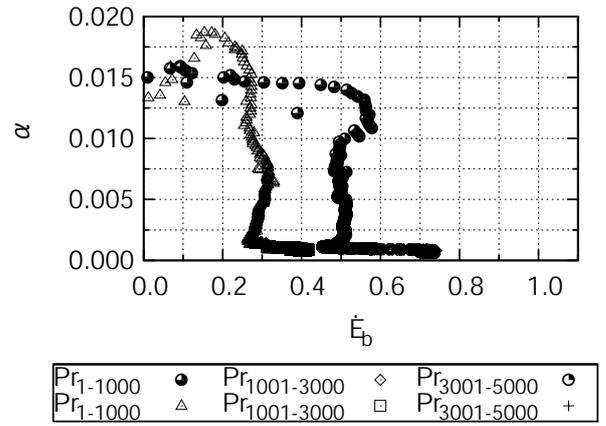
Fig. 5: Maximum fitness at each generation (BP10)

競争型共進化では、過去の対戦相手(の最良個体)に対して各世代の個体が良い適応度を示す、つまり、全世代にわたり、適応度が順次改善されていることが望ましい。Fig. 9に、(BP10, SP3)における過去の世代との対戦成績(適応度で評価)を示す(Fig. 5(c), 8(c)の試行に対応)。ここでは、色の濃淡で成績を示しており、色が明るいほど成績が良く、濃いほど成績が悪い。また、過去

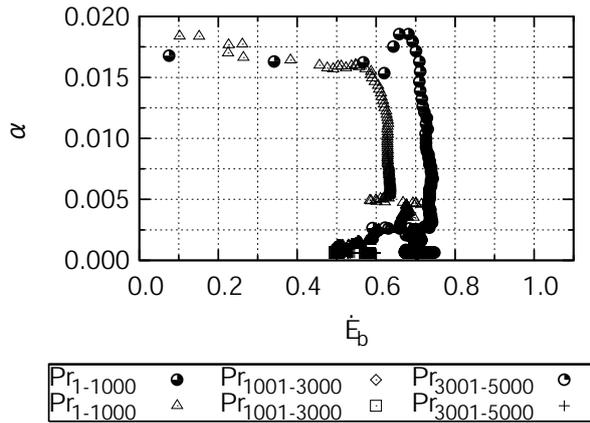
の世代との対戦であるため、ある世代(例えば10世代)の個体は、その世代以降の世代(10世代からその後)の相手とは対戦しないことから対角線の上側のみで表される。さらに、計算時間の問題から本実験では、100世代おきに最良個体を選択し対戦している。Prはほぼ全ての対戦において良い成績を示している(Fig. 9(a))。特に、2700~3500および4900世代のPrは、ほとんどの過



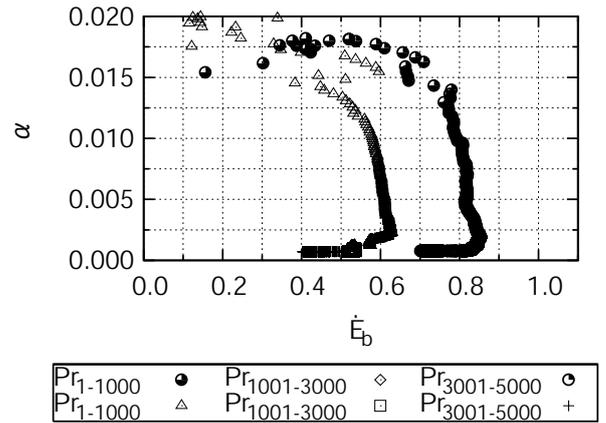
(a) SP1



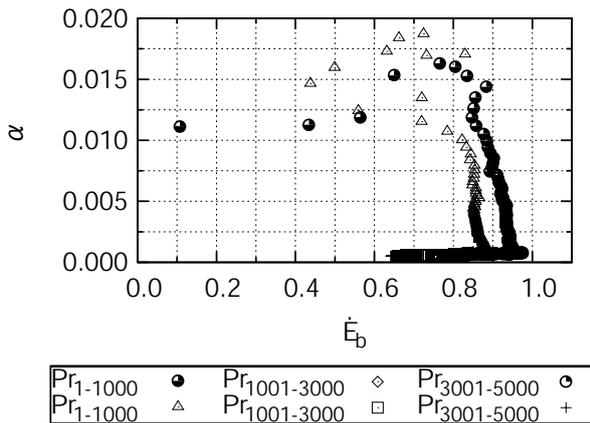
(a) SP1



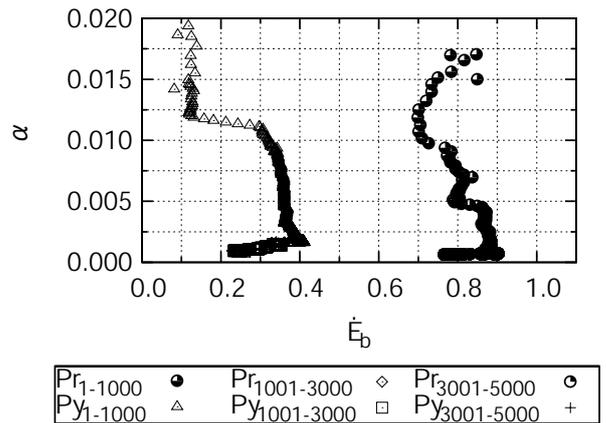
(b) SP2



(b) SP2



(c) SP3



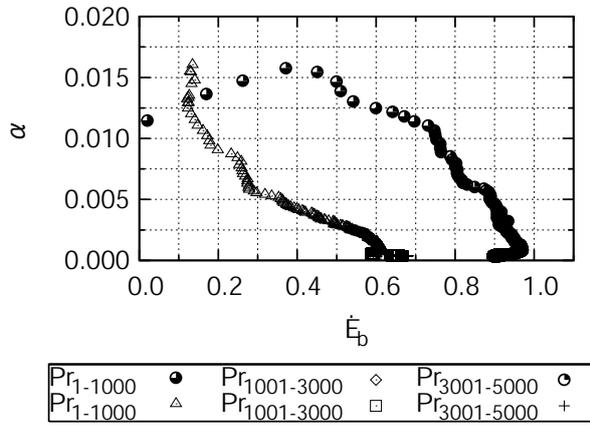
(c) SP3

Fig. 6: α as a function of the correlation (\dot{E}_b) for each generation (SG)

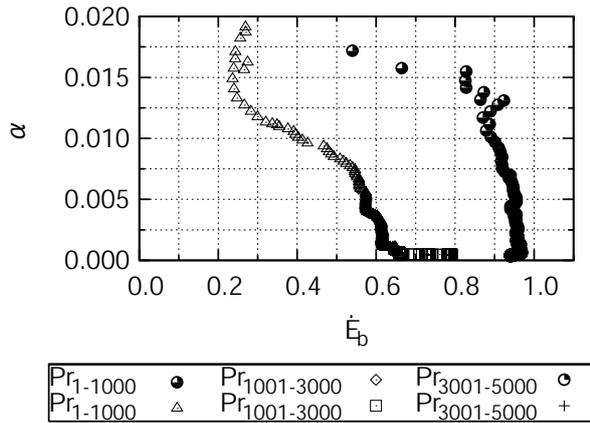
去の対戦相手に対し、良い成績を示している。これは、Fig. 5(c) で Pr の適応度が高かった世代に対応する。一方、Py はほぼ全ての対戦においてあまり良い成績を示しておらず (Fig. 9(b)), 特に、2000, 3000 および 4900 世代では、過去の対戦相手に対し、良い成績を示していない。これは、Py の適応度が低かった世代に対応する

Fig. 7: α as a function of the correlation (\dot{E}_b) for each generation (BP1)

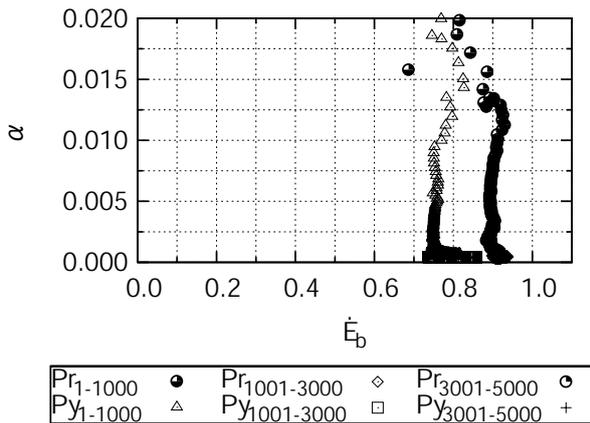
(Fig. 5(c)). 本実験では、各世代の最良個体が進化実験時に獲得した適応度にはほぼ等しい対戦成績を示し、過去の対戦相手に対して、全ての世代で必ずしも良い成績を示していないことから、理想的な共進化が起きているとはいえない。



(a) SP1



(b) SP2

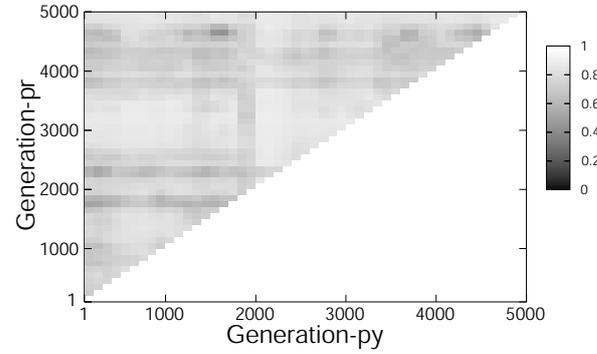


(c) SP3

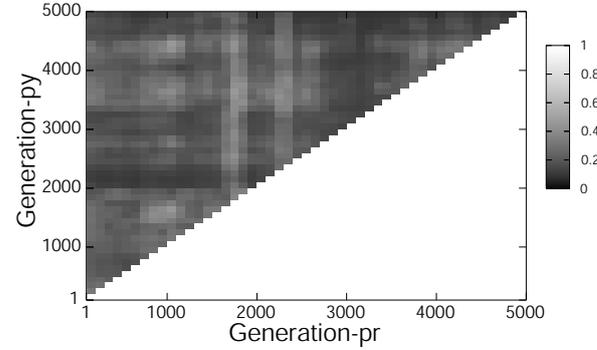
Fig. 8: α as a function of the correlation (\dot{E}_b) for each generation (BP10)

4 おわりに

本稿では、捕食-被食関係にあり形態の異なる2つの個体集団における、適応度景観の特徴量を推定しその変遷を調べ、競争型共進化の発生について検証を試みた。形態に差を与えることで、両集団の適応度が変化し、交互に適応度が改善されることを確認した。しかし、適応度



(a) Pr



(b) Py

Fig. 9: Current individual vs. ancestral opponents (BP10, SP3)

景観の特徴量は、形態が等しいときと同じく、世代を経るにつれほとんど変化しなくなった。これは種間に競争関係を設定しても (適応度関数, エージェントの形態), 突然変異率 (より一般的には遺伝的オペレータ) の設定によって集団が収束し得ることを示している。今後、形態に差を与えることに加えて、2. で述べた遺伝距離の特性が現れる範囲で突然変異率を上げ、適応度景観を観察するとともに、過去の世代の最良個体との比較と併せて、競争型共進化の発生を検証していく予定である。

参考文献

- [1] マッド・リドレー: 赤の女王, 翔泳社 (1995)
- [2] 半田佑太, 片田喜章: 競争型共進化における適応度景観の特徴量推定, 第52回システム制御情報学会研究発表講演会論文集, pp. 249-250 (2008)
- [3] 片田喜章, 大倉和博: 根井の標準遺伝距離を用いた適応度景観に含まれる neutrality の推定-進化ロボティクスへの適用, システム制御情報学会論文誌, Vol. 18, No. 8, pp. 284-291 (2005)
- [4] T. Smith, P. Husbands, P. Layzell and M. O'Shea: Fitness Landscape and Evolvability, Evolutionary Computation, Vol. 11, No. 4, pp. 381-415 (2002)
- [5] 根井正利: 分子進化遺伝学, 培風館 (1990)
- [6] D. Floreano and S. Nolfi: God Save the Red Queen! Competition in Co-Evolutionary Robotics, In Proceedings of the 2nd Annual Conference on Genetic Programming, Stanford University, pp. 398-406 (1997)