

# 競合型共進化アルゴリズムにおける Focusing問題に対する適応度共有法の適用

## Application of Fitness Sharing to Focusing on Coevolutionary Algorithms

佐藤 竜也      有田 隆也  
Tatsuya Sato      Takaya Arita

名古屋大学 大学院情報科学研究科 複雑系科学専攻  
Department of Complex Systems Science, Graduate School of Information Science, Nagoya University

In the framework of competitive coevolutionary computation, Focusing refers to the ability of coevolving opponents to challenge one another by testing weak dimensions of performance. A serious issue which arises in this context is overspecialization. This paper proposes a method of behavior-based fitness sharing to solve this issue. We apply the method to the intransitive numbers game, and compare its performance with conventional fitness sharing and resource sharing methods.

### 1. はじめに

共進化とは、ある生物の種とその同種・他種間の相互作用によって現れる進化のことである。これに基づいた共進化アルゴリズムは、従来の進化的計算のように静的に与えられた適応度関数を用いるのではなく、個体同士が影響を及ぼしあうことによって適応度が決定する。ある個体の適応度はそれを評価するための集団によって違うため、適応度地形は世代に応じて動的に変化する。また、その相互作用の定義によって競合型、協調型の2種類に分けられる。協調型は個体間の関係が相利共生であり、競合型は一方の適応度の増加が他方の減少につながる。

競合型共進化アルゴリズムの代表的な成功例として、Hillisのホスト/パラサイト共進化型遺伝的アルゴリズム (HPGA) によるソーティングネットワークの設計があげられる [Hillis 92]。従来の進化的計算によりソーティングネットワークの設計を行う場合、その完全な評価を行うには膨大な計算量が必要となる。Hillisは、ホスト個体にソーティングネットワーク、そのホスト個体を評価するためのパラサイト個体としてそのテストデータをおき、より多くのテストデータを正確に処理できたソーティングネットワークの適応度が高くなり、正確に処理できたソーティングネットワークがより少ないテストデータの適応度が高くなるようにシンプルな競合関係を定義し、効率の良いソーティングネットワークを得た。

競合型共進化アルゴリズムの特長として、以下の2点があげられる。

- 両集団が互いに適度な選択圧をかけるので、局所解に陥りにくい。
- 上記ソーティングネットワークのように、解候補の完全な評価が困難な問題において、部分的な評価で実現可能。

しかし共進化アルゴリズムにおける適応度は集団に依存した相対的なものであるため、理想的に進化しているかを確認することは一般に難しい [Watson 01][Stanley 02][Bader-Natal 04]。また、共進化アルゴリズムによる探索を困難にする性質として非推移的な関係 (Intransitivity [Watson 01]) がある。これは、個体間の優劣の関係にサイクルが存在する状況を指す。Watson

連絡先: 佐藤 竜也, 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, sato@create.human.nagoya-u.ac.jp

らは、個体間の相互作用を単純な2人ゲームとして表現したナンバーズ・ゲームを提案し、その上で競合型共進化アルゴリズムに起きる問題点として以下の3つをあげた [Watson 01]。

1. 遊離問題  
集団間の進化の歩調が大幅に離れ、集団内の個体間の適応度差がなくなり選択がほぼランダムになってしまうため、進化が進みにくくなること。
2. 堂々巡り  
進化が起きているように見えるが、同じ状況の間の遷移を繰り返していること。
3. フォーカシング問題  
ある特徴を表す遺伝子にのみ進化圧がかかり、一般的に良い解が得られないこと。

遊離問題の解決方法として、たとえば、[三井 01]の逆噴射機能を有するHPGAがある。これはパラサイト集団の適応度の平均がホスト集団の適応度の平均よりある一定以上超えたときに、評価の低いパラサイトほど適応度が高くなるように適応度を変換するものである。

堂々巡りに対する解決方法として、殿堂入り [Rosin 97] や ナッシュ・メモリ [Ficici 03] などのメモリ・メカニズムがある。これらの方法は、各世代で得られた最も適応度の高い個体を適応度評価に使うことで防ごうというものである。

フォーカシング問題の解決方法として資源共有法が提案されている [Juillé 98]。これは相手個体から得られる利得を、その相手から同じ利得を得た個体に対して均等に割り当てることにより、多様性を得ようという方法である。また、適応度共有法による探索の効率化がある [Smith 94]。適応度共有法とは、従来の遺伝的アルゴリズムで用いられる多様性を制御する方法である。また、多目的最適化の概念を用いたパレート共進化 [Noble 01] がある。メモリ・メカニズムもフォーカシングの防止の機能を持ちうるが、世代数やメモリなどの計算コストがかかる。

なお、非推移的な関係を持つ問題で集団間の評価基準の偏りによって堂々巡りが起きる場合、その原因はフォーカシングに由来すると考えることができる。よってフォーカシング問題の解決が堂々巡りの解決につながる可能性があると考えられる。

本研究ではフォーカシング問題のより効率的な解決法として、遺伝子型/表現型空間での近さではなく、個体の振る舞い

の類似度を使って多様性を保つ適応度共有法を提案する．さらにナンバース・ゲームの上で、各手法の比較評価により、その有効性を示す．

## 2. フォーカシング問題

### 2.1 フォーカシングの例

図 1 に、2次元のナンバース・ゲーム (4.1 で詳述) を用いた競合型共進化アルゴリズムにフォーカシングが発生した際の個体の分布例を示す．図中、popA と popB は各集団中の個体を表す．ナンバース・ゲームの純粋戦略は 2次元平面上の各点であり、2点間の差が最小である座標の値の大小を比較し、大きい方が利得を得る．ナンバース・ゲームにおいて理想的な進化は、x 座標も y 座標も大きい値になることだが、図 1 では、x 軸方向に偏った進化をしている．

図 2 は、各集団の x 座標、y 座標の平均値の推移をプロットしたものである．380 世代付近から両集団の x 座標の平均値の差が小さく、そして y 座標の差が大きくなっており、競合のほとんどが x 座標の比較により決定している．そのため、x 座標値の大きい個体ほど適応度が高い状況になり、それらが次世代に残りやすくなる．それ以降では x 軸方向への過剰な進化が発生していることが分かる．x 軸方向へのフォーカシングが発生している間、y 座標に関する評価がされず、y 座標の値は突然変異によってランダムウォークしている．

図 3 は、各集団の x、y 座標の標準偏差の推移をプロットしたものである．フォーカシングが発生した 380 世代付近を見ると、標準偏差が小さい状況が続いているのがわかる．これは、表現型の多様性が低い状況を示している．

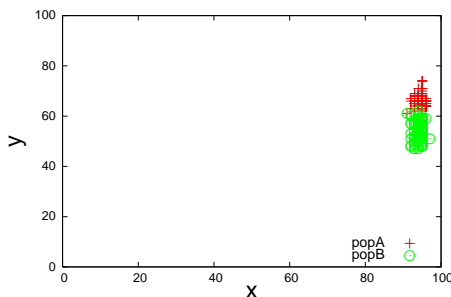


図 1: フォーカシングの例

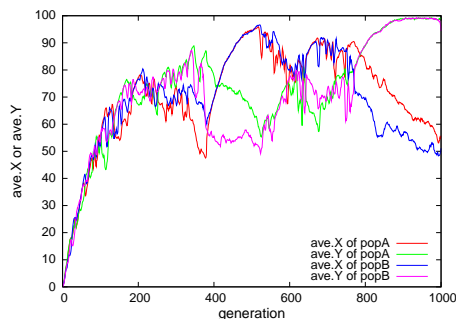


図 2: 各集団の平均値の推移

フォーカシング問題の解決方法として、評価すべき個体の特徴が偏るのを防ぐことがあげられる．共進化アルゴリズムで

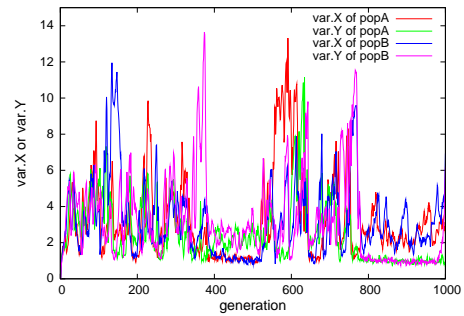


図 3: 各集団の標準偏差の推移

は、競合する集団が互いに相手の適応度地形を決定している．一方の集団が多様性を維持することにより、フォーカシングを解決できると考える．

### 2.2 適応度共有法

フォーカシング問題を解決するための多様性を制御する方法として、一般的な遺伝的アルゴリズムなどで用いられる適応度共有法 [Goldberg 87] の導入が考えられる．適応度共有法とは、集団中の個体に多様性を持たせるために、遺伝子型/表現型空間上に分布する個体の密度に比例してその部分の適応度を下げる適応度スケールリングである．個体が適応度地形のある 1 つの解候補の周辺に密集した場合、適応度共有法は一時的にその周辺の適応度を下げ、他の解候補の適応度を相対的に上げる．この操作により、それらが次世代に残る確率が上がり、多様性が生まれる．

個体  $i$  のスケールリング前の適応度を  $F(i)$ 、スケールリング後の  $F_{fs}(i)$  とした時、適応度共有法によるスケールリングを以下に定義する．

$$F_{fs}(i) = \frac{F(i)}{\sum_{j \in P} sh(d(i, j))} \quad (1)$$

ここで関数  $sh$  はシェアリング関数と呼ばれるもので、空間上で個体間が与える影響の度合いを表す．この関数の出力は、個体  $i$  と  $j$  が同一である場合には 1、そうでない場合は距離  $d$  に比例して 0 に近づく．シェアリング関数は、一般に以下のものが用いられる．

$$sh(d) = \begin{cases} 1 - (d/\sigma_{share})^\alpha & \text{if } d < \sigma_{share} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$\sigma_{share}$  は個体間が影響を与え合う距離の閾値、 $\alpha$  は距離が及ぼす影響の度合いを調整するパラメータである．

一般に距離関数  $d(i, j)$  の定義により適応度共有法は、遺伝子型 (G-FS)、表現型 (P-FS) の 2 種類に分けられる．

### 2.3 資源共有法

共進化アルゴリズムにおける多様性を制御する方法として資源共有法 (Resource Sharing, RS) がある [Juillé 98]．これは相手集団の個体から得られる評価値を、同じ評価値を得る個体同士で均等に分配するものである．その個体から得られる評価値は、同じ評価値を得る個体の普及率が高い場合は低く、普及率が低い場合には高くなる．本研究では、個体  $a$  の資源共有法による適応度  $F_{rs}(a)$  を以下のように定義した．

$$F_{rs}(a) = \sum_{b \in P_B} \frac{E(a, b)}{\sum_{a' \in P_A} \text{count}(a, a', b)} \quad (3)$$

$$\text{count}(a, a', b) = \begin{cases} 1 & \text{if } E(a, b) = E(a', b) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

ここで、 $P_A$  と  $P_B$  は各集団の個体の集合とする。関数  $\text{count}$  は、個体  $b$  による評価値が個体  $a$  と同じ個体が集団  $P_A$  中にどれだけ存在するかを表す。個体  $b$  によるその評価値は、その数に反比例した値になる。

### 3. 提案手法

本稿で提案する Behavior-based Fitness Sharing(B-FS) とは、遺伝子型/表現型空間上での個体間の近さを用いるのではなく、個体の振る舞いの類似度を用いた適応度共有法である。

本研究で扱う競合型共進化アルゴリズムでは、個体の振る舞いとして相手集団による評価結果を用いる。具体的には、集団 B による集団 A の個体  $a$  の評価結果を以下のようにベクトル  $t_a$  として表す。

$$t_a = (E(a, b_1), E(a, b_2), \dots, E(a, b_{N_B}))$$

ここで  $E(a, b)$  は、個体  $b$  による個体  $a$  の評価を表す。この評価結果のベクトル間の距離をその個体間の類似度とし、適応度共有法を用いる。本稿ではベクトルの距離関数として、ベクトルの差の 1 次ノルム<sup>\*1</sup>を用いた。

また、本手法において特徴のある個体とは、相手集団の全ての個体に対して優位な個体ではなく、ある一部の個体に対して優位な個体である。この特徴の多様性を持たせるために、個体が得た正の利得 ( $E(p, q) > 0$ ) の合計に基づいたスケールをさらに加えた。具体的には、個体  $i$  が持つ正の利得の合計を  $\text{point}(i)$  とし、適応度共有法によって得られた個体  $i$  の適応度  $F_{fs}(i)$  とした時、その個体の最終的な適応度  $F'(i)$  は以下のとおりである。

$$F'(i) = \text{weight}(\text{point}(i)) \times F_{fs}(i) \quad (5)$$

$$\text{weight}(p) = 1 - \left( \frac{p}{\text{MAX\_POINT}} \right)^\beta \quad (6)$$

ここで MAX\_POINT は、正の利得の合計の最大値である。本研究では、1 個体あたりの評価に相手集団の全個体を用いて評価するため、MAX\_POINT の値は集団中の個体数となる。

## 4. 評価

### 4.1 ナンバース・ゲーム

本研究では、2 人対称ゼロ和ゲームであるナンバース・ゲーム [Ficici 03] を用いる。ナンバース・ゲームの純粋戦略は、 $n$  次元ベクトルである。ベクトルの各要素は、区間  $[0, k]$  の値をとる。純粋戦略  $p$  が  $q$  と対戦した際の利得  $E(p, q)$  は以下の式で与えられる<sup>\*2</sup>。

$$E(p, q) = \text{sign} \left( \sum_{i=1}^n g_i \right) \quad (7a)$$

$$g_i = \begin{cases} p_i - q_i & \text{if } h_i = \min(h) \\ 0 & \text{otherwise} \end{cases} \quad (7b)$$

$$h_i = |p_i - q_i| \quad (7c)$$

\*1  $n$  次元ベクトル  $\vec{p}$  の 1 次ノルム  $\|\vec{p}\|_1 = \sum_{i=1}^n |p_i|$

\*2 参考文献 [Ficici 03] ではパラメータ  $\epsilon$  が存在するが、本稿ではそれを 0 と設定している。

ここで、関数  $\text{sign}$  は、入力が正の数ならば 1 を、負ならば -1 を、0 ならば 0 を返す関数である。

2次元のナンバース・ゲームの具体的な動作の例を、 $p = (3, 6)$ 、 $q = (1, 5)$ 、 $r = (4, 4)$  の 3 つの関係を用いて示す。 $p$  と  $q$  では、第 2 要素間の差が第 1 要素間の差よりも小さいので、第 2 要素の大小を比較する。よって  $p$  の第 2 要素が  $q$  の第 2 要素よりも大きいので  $p$  が利得  $E(p, q) = 1$  を得る。同様に  $q$  は  $r$  に対し第 2 要素の比較で大きいので  $E(q, r) = 1$  となる。しかし、 $r$  は  $p$  に対し第 1 要素の比較で大きいので、 $E(r, p) = 1$  となり、この 3 つの中でどれが最も優れた戦略かを判断することができないという問題が起こる。以上のように、ナンバース・ゲームは非推移的な関係を持つ。

ナンバース・ゲームを用いる利点は、どの戦略がどれだけ優れているかが予めわかっていることである。そのため、共進化アルゴリズムの性能評価に適している。

ナンバース・ゲームは、ベクトル間で最も差の小さい要素の大小を比較することで利得が決まる。よってベクトルの全ての要素においてより大きな値であれば、より多くの利得を得ることができる。このように世代によらない個体の評価値を絶対適応度と呼び、これに対しその世代における相手集団に対してどれだけ優れているかを表す評価値を相対適応度と呼ぶ。

本研究では、ある個体の絶対適応度をその表現型であるベクトルの全要素の和とし、相対適応度を対戦によって得られた利得から計算される適応度とする。

### 4.2 比較評価

ナンバース・ゲームを用い、提案手法である B-FS と G-FS、P-FS、RS との比較実験を行った。まず、 $n$  次元のナンバース・ゲームの場合、遺伝子として長さ  $nk$  のビット列を用意し、 $(\sum_{i=1}^k b_i, \sum_{i=k+1}^{2k} b_i, \dots, \sum_{i=k(n-1)+1}^{nk} b_i)$  として表現型であるベクトルを生成する。全ビット 0 の個体を初期集団に設定した。遺伝的操作には突然変異のみを用い、選択には適応度比例選択を用いた。共進化アルゴリズムのパラメータは、各集団の個体数は 100、突然変異確率は 0.005、世代数は 1000 を用いた。また、ナンバース・ゲームのパラメータは、次元数  $n$  は 2、各要素の区間  $k$  には 100 と設定した。

本研究では適応度共有法の距離関数として、遺伝子型にはハミング距離を、表現型にはユークリッド距離を用いた。また、シェアリング関数による影響の度合いを決定するパラメータ  $\alpha$  には、1 と設定した。各手法のパラメータ  $\sigma_{\text{share}}$  には、1 試行で得られる絶対適応度の最大値の期待値がもっとも高い値を求め、それを比較に用いる。各手法の  $\sigma_{\text{share}}$  の値は、G-FS には 10、P-FS には 7、B-FS には 90 を用いた。また、B-FS のパラメータ  $\beta$  には 2 を用いた。

以上の設定で、2 集団のうち一方に各手法を適用し、25 試行ずつ実験を行った。各手法を適用した集団を集団 A、もう一方を集団 B とする。

図 4 に、各試行で得られた集団ごとの絶対適応度の最大値の平均を示す。各手法はそれを適用した場合に何も適用しなかったもの (図中の No implement) に比べて、集団 B の値が最適値により近くなっている。これは、図 5 のように集団 A 内の個体が  $x$  軸、 $y$  軸方向へ分かれているためである。その集団 A の多様性により、集団 B には両方向への進化圧がかかり、フォーカシングが防がれている。

図 6 に、各世代において集団 B 内で得られた絶対適応度の最大値の平均を示す。何も実装しない場合に比べ、各手法の値は高い値を示している。B-FS は、他の手法に比べて安定して集団 B の絶対適応度を上げることができている。B-FS に比べて G-FS、P-FS や RS では値が途中で落ち込むが、これは

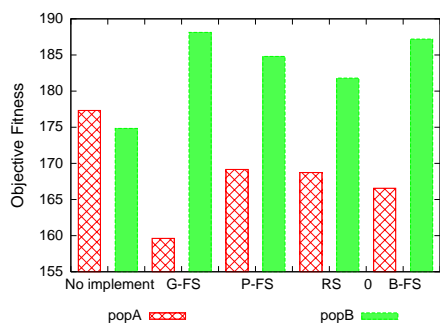


図 4: 1 試行あたりに得られる絶対適応度の最大値の平均

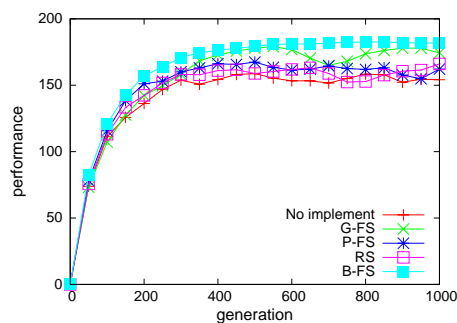


図 6: 集団 B の絶対適応度の変化

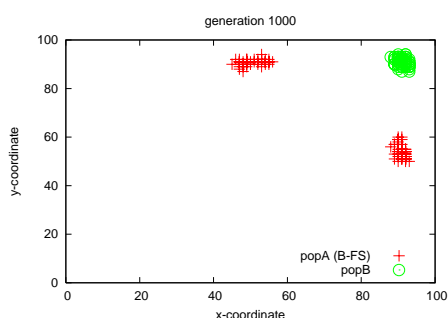


図 5: 1000 世代目における個体の分布例 (B-FS)

集団内に多様性を作り出すが、それを安定させられない場合があるためである。RS の性能が他に比べ悪かった理由として、[Juillé 98] では個体の評価値が 0 または 1 であるのに対し、用いたナンバーズ・ゲームは  $-1$  が含まれているためだと考えられる。

## 5. おわりに

競合型共進化アルゴリズムにおいて、多様性を制御する機能を一方の集団に持たせることによって、フォーカシングを解決することができた。そして、その制御機能として適応度共有法が有効であることを評価実験により示した。

本稿で提案した B-FS は、遺伝子型または表現型の距離関数を定義することが難しい場合、または不可能な場合も実装することが可能であり、有効であると考えられる。

## 参考文献

[Bader-Natal 04] Bader-Natal, A. and Pollack, J. B.: A Population-Differential Method of Monitoring Success and Failure in Coevolution, in *in Proceedings of the Genetic and Evolutionary Computation (GECCO-2004)*, Lecture Notes in Computer Science (2004).

[Ficici 03] Ficici, S. G. and Pollack, J. B.: A Game-Theoretic Memory Mechanism for Coevolution, in *Proceedings of the 2003 Genetic and Evolutionary Computation Conference (GECCO-2003)* (2003).

[Goldberg 87] Goldberg, D. E. and Richardson, J.: Genetic Algorithms with Sharing for Multimodal Function Opti-

mization, in *the Second International Conference on Genetic Algorithms* (1987).

[Hillis 92] Hillis, B.: Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure, *Artificial Life II* (1992).

[Juillé 98] Juillé, H. and Pollack, J. B.: Cevolutionary Learning: a Case Study, in *Proceedings of the Fifteenth International Conference on Machine Learning* (1998).

[Noble 01] Noble, J. and Watson, R. A.: Pareto Coevolution: Using Performance against Coevolved Opponents in a Game as Dimensions for Pareto Selection, in *Proceedings of the Genetic and Evolutionary Computation Conference* (2001).

[Rosin 97] Rosin, C. D.: *Coevolutionary Search Among Adversaries*, PhD thesis, University of California at San Diego, La Jolla, CA, USA (1997).

[Smith 94] Smith, R. E. and Gray, B.: Co-Adaptive Genetic Algorithms: An Example in Othello Strategy, in *Proceedings of The Florida Artificial Intelligence Research Symposium* (1994).

[Stanley 02] Stanley, K. O. and Miikkulainen, R.: The Dominance Tournament Method of Monitoring Progress in Coevolution, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)* (2002).

[Watson 01] Watson, R. A. and Pollack, J. B.: Coevolutionary Dynamics in a Minimal Substrate, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (2001).

[三井 01] 三井 富雄, 有田 隆也: 進化圧を自動調整するホスト/パラサイト共進化型遺伝的アルゴリズム, 人工知能学会全国大会論文集, 2C3-04 (2001).