

競合型共進化における多様性維持手法による適応度こう配の提供の効果

佐藤 竜也[†] 有田 隆也[†]

The Methods for Maintaining Diversity in Evaluation Criteria Solve the Loss of Gradient Problem in Competitive Coevolutionary Algorithms

Tatsuya SATO[†] and Takaya ARITA[†]

あらまし 2 集団の競合型共進化アルゴリズムにおけるこう配の喪失とは、一方の集団が他方に対して圧倒的に優位な個体群へ進化し、各集団内の適応度差がなくなることであり、選択の効果がなくなる。これを解決するには、優位な個体群へ進化しやすい集団の進化を相対的に遅くする必要がある。また、別の問題点とされる過度の特殊化とは、個体の特徴の一部に関する競合だけが集団間に生じ、その部分だけが最適化されてしまうことであり、最適解の探索を妨げる。振舞いに基づいた適応度共有法とエリート抑制は、この問題点を集団間の競合が生み出す評価基準の多様性の消失とみなし、その多様性維持により解決を図る。この 2 手法を含めた過度の特殊化の解決手法は、集団内の最良個体の適応度を下げ、それよりも若干劣る個体の適応度を高くすることで、進化を遅くすることが期待できることから、こう配の喪失を解決できると考えられる。そこで本研究は、この推測を検証するために両問題点の解決手法に関する比較評価実験を行った。その結果、過度の特殊化の解決手法がこう配の喪失を解決し得ること、及び、上記 2 手法が他の解決手法よりも良い解を発見し得ることを示した。

キーワード 競合型共進化アルゴリズム, こう配の喪失, 過度の特殊化, 遺伝的アルゴリズム

1. ま え が き

共進化アルゴリズムは、解きたい問題に対する絶対的な評価値を用いずに、個体間相互作用に基づく相対的な適応度評価を行うのが特徴である。そして、この相互作用の性質によって協調型と競合型に分類できる。協調型では、個体の表現型が部分解を表しており、それらを複数個組み合わせられた解を評価することで、それを構成する各個体の適応度を決める [1]。一方、本研究で取り扱う競合型では、個体間の相互作用に基づいて優劣を比較することで適応度を評価する。

競合型共進化アルゴリズムが従来の進化的計算手法よりも効率的に探索できる問題として、解評価にテストデータを要するテストベース問題が挙げられる。テストデータの総数が膨大である場合、絶対的な適応度の評価は計算コストが大きく、テストデータの数を限定するにしても効率的な探索に必要なものを選ぶのが

必ずしも容易でないので、従来の進化的計算手法は適用しづらい。それに対して、解を表す個体（ホスト個体）集団とテストデータを表す個体（パラサイト個体）集団とを競合させることで、そのような困難なしに競合型共進化アルゴリズムが適用できる。

Hillis の先駆的研究 [2] では、ソーティングネットワーク設計問題において、ホスト個体をソーティングネットワーク、パラサイト個体をテストデータとし、正確に処理できたパラサイト個体数をホスト個体の適応度、正確に処理できなかったホスト個体の数をパラサイト個体の適応度として、競合型共進化アルゴリズムを適用した。その結果、遺伝的アルゴリズムを適用した場合よりも最適解に近い解を得た。その後、セルオートマトン密度分類問題 [3] や関数推定問題 [4] などの他のテストベース問題に対しても競合型共進化アルゴリズムが効率的に適用可能であることが示されている。

競合型共進化アルゴリズムの大きな特徴は、相手集団よりも優位になろうという進化圧を相互にかけあうことにより、それぞれの集団内に適応度のこう配を提

[†] 名古屋大学大学院情報科学研究科, 名古屋市
Graduate School of Information Science, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya-shi, 464-8601 Japan

供し続けることである(こう配の提供)。その際、まだ最適化されていない部分を最適化する方へ集団の進化の方向を集中させること(フォーカシング)が可能である。これらによって、自然界に見られるようなオープンエンドな進化を目指す。

しかし、Watsonらは、個体間の競合関係を単純な2人ゲームとして表現したナンバーズゲームを用いて、これらの利点が常に得られる保証がなく、逆に、適切な進化の阻害要因になり得ることを指摘している[5]。本論文では、そこで列挙されている二つの問題点、過度の特殊化とこう配の喪失に注目する。

過度の特殊化とは、個体の特徴の一部に関する競合だけが集団間に生じ、その部分だけが最適化されてしまうことであり、最適解の探索を妨げる。集団間の競合が個体の表現型に依存することから、この問題点の解決には、集団内に個体の多様性をもたせることが必要であると指摘されている[6],[7]。その多様性維持手法として、従来の進化的計算で用いられてきた適応度共有法[8]や、この問題点に特化した解決手法として資源共有法[9]、多目的最適化の概念を用いるパレート共進化[10]がある。

過度の特殊化の解決手法として、振舞いに基づいた適応度共有法及びエリート抑制を我々は提案した[11]。評価実験の結果から、これらの手法が、単に表現型の多様性ではなく、一方の集団内に異なった特徴に特化した個体群を生み出すことにより、他方の集団に適切な進化圧を与え、問題点を解決し得ること、特に、従来研究よりも少ない個体数と短い世代数の設定であったにもかかわらず、それらに匹敵する良解を発見し得ることが示された。

こう配の喪失とは、一方の集団が他方に対して圧倒的に優位な個体へ進化し、各集団内の適応度差がなくなることであり、選択の効果がなくなる[5],[12]。そして、選択が機能しないことで、良解が淘汰されたり、交叉や突然変異によるスキーマの破壊などの問題を引き起こす。また、この問題点は、単調性^(注1)が保証されているパレート共進化においても生じることが報告されている[14],[15]。一方の集団が他方に対して優位なものへ進化しやすい原因として、ソーティングネットワーク設計問題[2],[16]などのテストベース問題や捕食者-被食者問題[17]などのように、両集団で探索空間の難易度や性質が異なることが挙げられる。例えば、テストベース問題では、遺伝的操作によって良い解候補が発見されにくいのに対し、テストデータはそ

れらに対して難しいものへ進化しやすい。

この問題点を解決するには、一方の集団が相手に対して圧倒的優位なものへ進化するのを防がなければならない。その方法として、圧倒的優位になりやすい集団の進化の速さを抑えるために、選択圧、及び、遺伝的操作の調整が考えられる。しかし、ソーティングネットワーク設計問題(3.2参照)のように、初期世代において集団間の優劣に大差がある場合、劣る個体を積極的に次世代へ残すことが必要であるが、選択圧や遺伝的操作の調整では実現できない。そのため、集団の状況に応じて最良個体よりも劣る個体を次世代に残すことができる手法が提案されてきた。そのような従来手法として、最良個体よりも若干劣る個体の適応度を最大にするパラサイト毒性緩和[12]や、集団間の平均適応度に大差が生じたときに進化圧を逆向きにする逆噴射機能[18]がある。

これらの問題点はそれぞれ独立したものとして扱われ[5],[19]、上述に挙げた手法も含め、各問題点に特化した解決手法が提案されてきた[3],[9],[12],[18],[20]~[22]。しかし、本研究では、上で紹介した、過度の特殊化に対象を絞って提案されてきた解決手法はこう配の喪失に対しても有効であると考えた。なぜなら、相手集団に対して優位な個体が集団内に増加した際に、これらの手法は、集団中の他個体との類似度に応じて適応度を下げることで、最良個体よりも若干劣る個体の方が適応度が高い状況を作ることが期待できるためである。

そこで、本研究は上記仮説を検証することを第1の目的とし、計算機実験によって各手法の探索性能を定量的に評価した。本実験では、こう配の喪失が生じ、なおかつ解候補の良さの絶対的な評価が可能な問題に、過度の特殊化の解決手法及びこう配の喪失の解決手法のそれぞれを適用し、得られた解候補の絶対的な評価値に基づいて手法の比較評価を行う。そして、その際に我々が前研究で提案した振舞いに基づいた適応度共有法及びエリート抑制の有効性を調べることを第2の目的とする。取り扱う解決手法には、こう配の喪失の解決手法として逆噴射機能とパラサイト毒性緩和を、過度の特殊化の解決手法として適応度共有法と資源共有法、振舞いに基づいた適応度共有法、エリート抑制を用いた。問題には、一次元ナンバーズゲームとソー

(注1): 単調性とは、新しく得られた最良個体がそれまでに得られた最良個体よりも絶対的な評価において常に良いことを指す[13]。

ティングネットワーク設計問題を用いた．競合型共進化アルゴリズムでは適応度が相対的であり，また挙動が複雑化しやすいが，これらの問題は解の絶対的な評価が可能であり，また複雑すぎないので，解析が容易であり，手法の比較評価に適している [5], [12], [21] ．

2. 両問題点に対する解決手法

2.1 こう配の喪失に対する解決手法

2.1.1 パラサイト毒性緩和 (RPV)

パラサイト毒性緩和 (Reducing Parasite Virulence) は，集団内に適応度の多様性を維持することでこう配の喪失を解決する手法であり，パラサイト集団の最良個体ではなく，それよりも若干劣る個体に最大適応度を与える [12] ．

パラサイト毒性緩和による適応度 $F_{rpv}(i)$ を，以下のように定義した．

$$x(i) = \frac{f(i)}{\max_j f(j)} \quad (1)$$

$$F_{rpv}(i) = \frac{2x(i)}{\lambda} - \frac{x(i)^2}{\lambda^2} \quad (0.5 \leq \lambda \leq 1.0) \quad (2)$$

ここで， $f(i)$ はパラサイト個体 i の適応度を表し， $x(i)$ はパラサイト個体 i の適応度をパラサイト集団中の最大適応度で正規化した値を表し，これをスコアと呼ぶ．パラメータ λ は，最大の適応度を得るスコアの値を決定する．

2.1.2 逆噴射機能 (TR)

逆噴射機能 (Thrust Reverse) は，集団間の平均適応度の差がパラメータ T_{level} 以上になったとき，優位になっている集団 (パラサイト側を想定) 中の適応度の評価基準を逆転させるように適応度を変換する手法である [18] ．

逆噴射機能による個体の適応度 F_{tr} を以下のように定義した．

$$F_{tr}(i) = \begin{cases} f_{\max} - f(i) & \text{if } \bar{f}_p - \bar{f}_h \geq T_{level} \\ f(i) & \text{otherwise} \end{cases} \quad (3)$$

ここで， f_{\max} はパラサイト個体がとり得る最大適応度， \bar{f}_p と \bar{f}_h はそれぞれパラサイト個体，ホスト個体の平均適応度を表す．

2.2 過度の特殊化に対する解決手法

2.2.1 適応度共有法 (G-FS, P-FS)

適応度共有法 (Fitness Sharing) とは，集団中の個体の多様性維持のために，各個体の適応度を集団中の

他個体との類似度に応じて下げる適応度計算手法である [8], [23] ．

適応度共有法による個体 i の適応度 $F_{fs}(i)$ を以下のように定義した．

$$F_{fs}(i) = \frac{f(i)}{\sum_{j \in P} sh(d_{i,j})} \quad (4)$$

$$sh(d) = \max\{0, 1 - d/\sigma_{share}\} \quad (5)$$

ここで， $d_{i,j}$ は個体 i, j 間の距離， P は集団を表す．関数 $sh(d)$ は，個体間の距離からそれらの類似度を計算する． σ_{share} は，類似しているとみなされる範囲を表すパラメータである．

一般に，距離 $d_{i,j}$ の定義により，遺伝子型適応度共有法 (G-FS) と表現型適応度共有法 (P-FS) に分類される．

2.2.2 資源共有法 (RS)

競合型共進化アルゴリズムにおける多様性維持手法である資源共有法 (Resource Sharing) [9] は，ある相手個体による評価から得られる利得を，その相手個体から同じ評価を得た個体同士で分配する．

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．

$$F_{rs}(a) = \sum_{b \in P_B} \frac{E(a,b)}{\sum_{a' \in P_A} c(a,a',b)} \quad (6)$$

$$c(a,a',b) = \begin{cases} 1 & \text{if } E(a,b) = E(a',b) \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

ここで， $E(a,b)$ は個体 a が個体 b との相互作用により得る利得， P_A と P_B は各集団の個体の集合を表す．関数 c は，個体 b から得た利得が個体 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 からの利得は，その数に反比例した値となる．

2.2.3 評価基準の多様性維持手法

過度の特殊化を解決する手法として我々が提案した，振舞いに基づいた適応度共有法とエリート抑制 [11] を以下に紹介する．これらの手法は，過度の特殊化を一方の集団が他方に対して与える評価基準の多様性の喪失によるものとみなし，また相手集団に与える評価基準は相手集団による評価結果に反映されることが期待できることから，評価結果の多様性維持による解決を目的としている．

(a) 振舞いに基づいた適応度共有法 (B-FS)

振舞いに基づいた適応度共有法 (Behavior-based Fitness Sharing) は，相手集団による評価結果を個体

の振舞いと定義し、その振舞いの類似度を用いた適応度共有法である [11] .

具体的には、評価結果を表すベクトル t_a を

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

と定義し、このベクトル間の距離を用いて適応度共有法を行う。ここで、 N_B は集団 B の個体数を表す。

テストベース問題におけるテストデータの集団のように、一方的に進化しやすい集団に対して B-FS を適用した場合、エリート個体が集団内に増加した際に、それらの評価結果が互いに類似することから、他の若干劣る個体よりもそれらの適応度を下げる効果が期待できる。そのため、B-FS は、本来対象としていた過度の特殊化に加えて、こう配の喪失も解決し得るものと期待できる。

(b) エリート抑制 (ES)

エリート抑制 (Elite Suppression) は、個体が相手集団にどれだけ優位であるかに応じて、もとの適応度を割り引く適応度計算手法である [11]。相手に多様な評価基準を与える集団とは、複数の評価基準において優れた個体が多数いる集団ではなく、単一の評価基準において優れた個体が多様に存在する集団であると考へ、エリート抑制は前者のような個体を淘汰することを目的としている。具体的には、エリート抑制による適応度 $F_{es}(i)$ を以下のように定義した。

$$F_{es}(i) = w(p(i)) \times f(i) \quad (8)$$

$$w(p) = 1 - \left(\frac{p}{p_{\max}}\right)^\beta \quad (9)$$

$$p(i) = \sum_{j \in P_{\text{opponent}}} \max\{E(i, j), 0\} \quad (10)$$

$$p_{\max} = \text{win_payoff} \times |P_{\text{opponent}}| \quad (11)$$

ここで、 win_payoff は得られる正の利得の最大値、 P_{opponent} は相手集団中の個体の集合、 $p(i)$ は個体 i が得た正の利得の合計、 p_{\max} は正の利得の合計の最大値を表す。関数 w は、もとの適応度に対する割り引き後の適応度の割合を表す単調減少関数である。この関数により、得た正の利得の合計が高い個体ほど適応度が多く割り引かれ、 $p(i) = p_{\max}$ である個体の場合は 0 となる。また、 β は p に対する割り引きの割合を調整するパラメータである。

集団内の良い個体の適応度を下げ、それよりも若干劣る個体の適応度が高い状況を作れることから、RPV

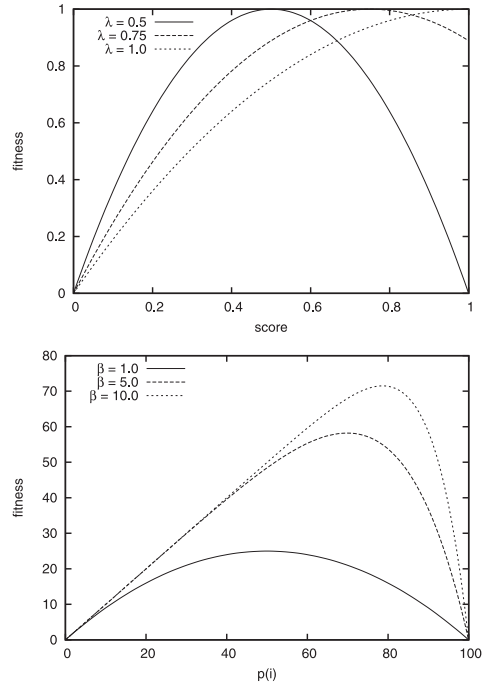


図 1 RPV (上) と ES (下) の適応度関数。ここでは $|P_{\text{opponent}}| = 100, \forall i, j E(i, j) \in \{0, 1\}, p_{\max} = 100, \text{win_payoff} = 1$ と仮定。

Fig. 1 The fitness function of RPV (top) and of ES (bottom).

に似ている (図 1 参照)。そのため、エリート抑制もまたこう配の喪失を解決し得ると考えられる^(注2)。RPV が集団内の最良個体の適応度 (式 (1) 中、 $\max_j f(j)$) を基準にしているのに対し、エリート抑制は得られる利得の最大値 (p_{\max}) を基準にしている。そのため、RPV は常に進化の速さを遅くするが、エリート抑制は相手集団に対してある程度優位になるまでは通常の共進化を許す。

3. 比較評価

3.1 一次元ナンバーズゲームによる評価実験

3.1.1 概要

本論文で取り扱う一次元ナンバーズゲームは、戦略は整数値 1 個であり、その値の大小比較により利得の計算を行う。戦略 a が戦略 b と対戦して得る利得 $E(a, b)$ は、 a の値が b よりも大きい場合は 1、等しい場合は $1/2$ 、小さい場合は 0 となる。戦略が表す値が

(注2): エリート抑制は、B-FS などの多様性維持手法と組み合わせることを目的としていたが、本論文では組み合わせずに評価する。

大きいほど、より多くの他の戦略から利得を得ることができるため、戦略が表す値自身を戦略の良さを表す絶対的な指標（絶対適応度）として、各手法の比較評価の際に用いる。このように絶対的評価ができるため、競合型共進化アルゴリズムにおける手法の評価によく用いられる [24] ~ [26]。

3.1.2 実験設定

本評価実験では、単に競合型共進化アルゴリズムを適用した際に生じるこう配の喪失を各手法によって改善できるのかを評価するため、進化アルゴリズムの設定を統一した。文献 [12] に従い、以下のように設定した。ホスト集団とパラサイト集団中の個体数は 25 ずつ、ホスト個体とパラサイト個体の遺伝子はともに長さ 100 の 2 進数列とし、その中に含まれる 1 の個数を表現型とする。初期集団中の全個体の全遺伝子座の遺伝子を 0 とする。遺伝的操作は突然変異のみを用い、突然変異確率は 0.03 とした。こう配の喪失を意図的に発生させ、またその発生しやすさを連続的に調整できるようなタスクを用いた。ここでは、タスクにおいてではなく、パラサイト個体において突然変異の起こり方を調整することでそれを実現している。具体的には、パラサイト個体に対してのみ、突然変異が起きた遺伝子座において遺伝子が 1 に変化する確率 $B_p (> 0.5)$, 0 に変化する確率を $1 - B_p$ として調整可能とした。こうすることにより、パラサイト集団がホスト集団に対して優位に進化することができる。なお、このようなやり方は Cartlidge らの研究 [12] においても採用されている。パラメータ B_p の値に、0.55, 0.60, 0.65, ..., 0.95 を用いた。選択には、トーナメント選択（トーナメントサイズ 5）を用いた。各集団中の個体の適応度の評価は、相手集団中から無作為に選んだ 5 個体と対戦して得られた利得の総和とする。

適応度共有法のパラメータ σ_{share} は、G-FS と P-FS には 1.0, 5.0, 10.0, 30.0, 50.0, 100.0 を、B-FS には 0.0, 1.0, ..., 5.0 を用いた。逆噴射機能のパラメータ T_{level} には、0.0, 1.0, ..., 5.0 を用いた。パラサイト毒性緩和のパラメータ λ には、0.5, 0.75, 1.0 を、エリート抑制のパラメータ β には、1.0, 2.0, ..., 10.0, 50.0, 100.0 を用いた。適応度共有法で用いる距離関数には、G-FS ではハミング距離を、P-FS では個体の遺伝子中に含まれる 1 の数の差の絶対値を用いた。

以上の設定で、パラサイト集団に対して各手法を適用し、1,000 世代の試行を 30 試行ずつ実験を行った。

3.1.3 結果

各手法の性能を表す値として、全世代の中でこう配の喪失が発生した世代の割合と、ホスト集団とパラサイト集団の絶対適応度の平均を計算した。各集団の適応度の分散が 0 となった世代をこう配の喪失が発生した世代と定義した。各手法のパラメータは、ホスト集団の絶対適応度の平均が最大となった値を比較に用いた。

図 2 にパラメータ B_p の各値におけるこう配の喪失が発生した世代数の割合を示す。この値は、全世代数に対するこう配の喪失が生じた世代の割合を各試行について計算し、全試行について平均したものである。また、図 3 にホスト集団の絶対適応度の平均を図 4 にパラサイト集団の絶対適応度の平均を示す。競合型共進化アルゴリズムを単に適用した場合（図中 NONE）、 $B_p \leq 0.7$ ではこう配の喪失がほとんど起きず、ホスト個体及びパラサイト個体の絶対適応度の平

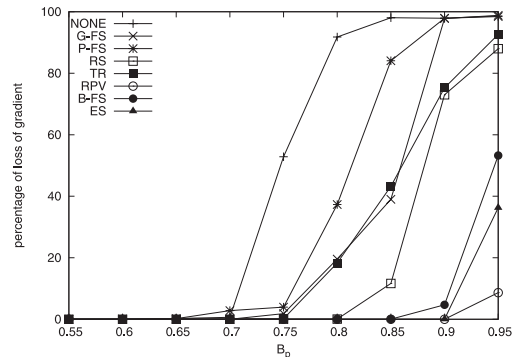


図 2 こう配の喪失が発生した世代の割合
Fig. 2 Percentage of generations at which loss of gradient occurred.

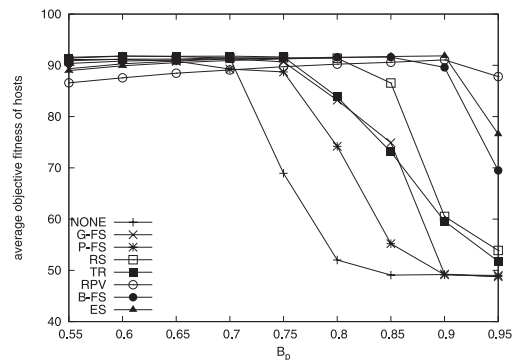


図 3 ホスト個体の絶対適応度の平均値
Fig. 3 Average of objective fitness in the host population.

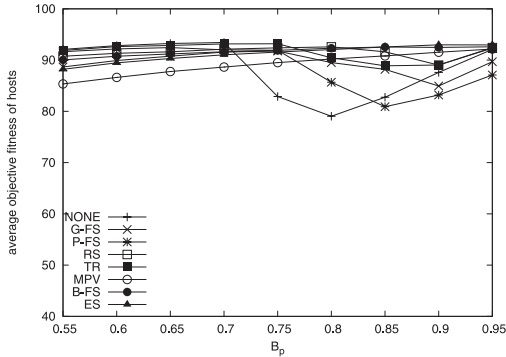


図 4 パラサイト個体の絶対適応度の平均値

Fig. 4 Average of objective fitness in the parasite population.

均は高かった。 $B_p \geq 0.75$ では B_p の値に応じてこう配の喪失が発生した世代数は増加し、それとともに宿主集団の絶対適応度の平均は減少した。それに対し、 $B_p \geq 0.75$ において、パラサイト集団の絶対適応度は B_p に比例して上昇している。しかし、これは単拮競争が成功した結果ではなく、この状況ではこう配の喪失が頻繁に生じたことにより探索が確率 B_p に従ったランダムウォークとなったためである。

NONE と比べて、各手法を適用した場合は、こう配の喪失の発生が抑えられ、宿主個体の絶対適応度は改善されていた。しかし、G-FS、P-FS、RS、TR の四つは、 B_p の値が高くなるにつれて性能が下がり、 $B_p \geq 0.9$ の場合ではこう配の喪失をほとんど防ぐことができず、宿主集団の絶対適応度の平均も大きく減少していた。このとき、パラサイト集団の絶対適応度が上昇しているが、上述の NONE の場合と同様、ランダムウォークによる結果である。RPV については、宿主個体の絶対適応度の平均値が B_p の値に比例して上昇し、特に $B_p = 0.95$ においては最も高い値を示した。しかし、 $B_p \leq 0.7$ において、RPV はこう配の喪失を防いでいたものの、得られた宿主個体の絶対適応度は NONE よりも下回っており、比較した手法の中で最も低かった。これは、RPV により常に集団中の最良個体が淘汰されてしまうことから、パラサイト集団が宿主集団に対して若干優位になりやすい程度では進化の速さを過剰に下げってしまうためだと考えられる。

B-FS と ES は、他の過度の特殊化に対する解決手法とは異なり、 $B_p \leq 0.9$ までは性能が高く、RPV に準じる性能であった。他の過度の特殊化に対する解決

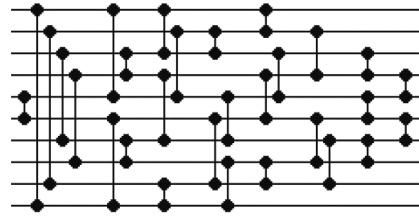


図 5 ソーティングネットワーク (入力サイズ 10)

Fig. 5 A sorting network (input size 10).

手法と性能に差が出た原因は、相手個体すべてから利得を得る最良個体の増え方の違いにあると考えられる。適応度共有法を用いた場合には、最良個体はその遺伝子(表現型)が珍しいければ、適応度が高いため増加する。RS を用いた場合、相手個体すべてから利得を得た個体の適応度が最も高くなるため、NONE と同様である。それに対して、B-FS と ES は、どのような状況であっても、最良個体の適応度をそれらの個体数に応じて下げ、繁殖を抑えることができる。

3.2 ソーティングネットワーク設計問題による評価実験

3.2.1 概要

ソーティングネットワークとは、任意の順序で並んだデータに対して比較交換を順に適用することによりソートを行うアルゴリズムである(図 5)。入力データ中の各値は対応する横線の上を左から右へ流れ、縦線で表される比較交換が順に適用され、右側へ出力される。流れている値が縦線の端点に到達すると、上端の値が下端の値より大きければ、それらは交換される。

ソーティングネットワーク(SN)設計問題とは、任意の順序のデータを正確にソートできるソーティングネットワークを探索する問題である。SN 設計問題の先駆的研究 [2], [21] では、比較交換回数の最小化もしていたが、本評価実験の目的は比較評価であるため、比較交換回数を固定した上で比較交換する位置と順番の設計に限定した。本実験では、入力サイズ n は 13、比較交換回数 l には、入力サイズに対して現在知られている最小値である 45 とした [27]。

3.2.2 実験設定

SN 設計問題を、ソーティングネットワークを表す個体の集団(宿主集団)とそれを評価するためのテストデータの集団(パラサイト集団)との競合型共進化アルゴリズムにより行う。先の実験と同様の理由から、進化アルゴリズムの設定を統一した。文献 [12] に従い、以下のように設定した。

入力サイズ n , 比較交換回数 l のソーティングネットワークを表すために, 1 から n までの範囲の整数値からなる長さ $2l$ のリストを宿主個体の遺伝子型として定義する. 遺伝子座 i の遺伝子が表す整数を g_i としたとき, 表現型であるソーティングネットワークの j 番目の比較交換の位置は, g_{2j-1} と g_{2j} との組によって表される. 遺伝的操作は突然変異のみを用いる. 突然変異確率は遺伝子座当り 0.002 とし, ランダムな値に置き換える.

テストデータを表すパラサイト個体の遺伝子型は長さ n の順列とし, そのまま表現型として用いる. ホスト個体と同様に, 遺伝的操作は突然変異のみとする. 突然変異確率は遺伝子座当り 0.002 とし, 発生した遺伝子座の遺伝子とランダムに選ばれた他の遺伝子座の遺伝子を交換する. 正確にソートすることができたパラサイト個体の数を宿主個体の適応度とし, その個体を正確にソートできなかった宿主個体の数をパラサイト個体の適応度とした. 両集団ともに, 選択にはトーナメント選択 (トーナメントサイズ 5) を用い, 個体数は 100 とし, 初期集団中の個体をランダムに生成する.

パラサイト個体の遺伝子型はそのまま表現型を表しているため, 適応度共有法は遺伝子型適応度共有法のみ適用し, 距離関数には偏差距離 [28] を用い, パラメータ σ_{share} には 1.0, 2.0, 3.0, 4.0, 5.0, 7.0, 13.0 を用いた. 逆噴射機能のパラメータ T_{level} には 5, 10, 20, 30, 40, 50, 75, 90, パラサイト毒性緩和のパラメータ λ には 0.5, 0.75, 1.0 を用いた. B-FS のパラメータ σ_{share} には 2.0, 5.0, 10.0, 25.0, 50.0, 75.0, 100.0 を, エリート抑制のパラメータ β には 1.0, 2.0, 3.0, 4.0, 5.0, 10.0 を用いた.

以上の設定で, パラサイト集団に各手法を適用し, 各パラメータごとに 30 試行ずつ実験を行った.

3.2.3 結果

本論文では, 0-1 原理^(注3)に従い, 0 と 1 からなる 2^n 個のテストデータのうちソーティングネットワークが正確にソートできたものの割合を正答率と定義し, これを各手法の比較評価に用いる. 各手法のパラメータは, 最終世代における宿主個体の正答率の平均が最大となった値を比較に用いた. また, パラサイト集団の進化を見るために絶対的な指標として, テストデータの難度を用いた. 文献 [18] と同様に, テストデータ d の難度 $D(d)$ を以下のように定義した.

$$D(d) = \frac{1}{n} \sum_{i=1}^n \frac{|i - a(d_i)|}{n - 1} \quad (12)$$

ここで, d_i は i 番目の位置にあるデータを, $a(d_i)$ は正しくソートされたときのデータ d_i の位置を表す.

図 6 に, 各世代でこう配の喪失が発生した試行数を示す. ここでは, 先の実験と同様に, 各集団の適応度の分散が 0 となった世代をこう配の喪失が発生した世代と定義した. 集団をランダムに初期化したため, どの宿主個体もパラサイト集団のほとんどのテストデータを処理できていなかった. そのため, 適用した手法によらず, 試行の多くは初期世代からこう配の喪失が発生していた. 競合型共進化アルゴリズムを単に適用した場合 (図中 NONE), こう配の喪失が発生した試行数は, どの世代においても 20 試行以上と高い頻度であったが, すべての試行で起きていた世代はなかった. これは, パラサイト集団中のわずかな個体を正確にソートできる宿主個体が突然変異によって現れたことで, 適応度のこう配が一時的に生じたためである. しかし, その突然変異体によって正確にソートされたパラサイト個体は, 他の個体よりも適応度が低いので, 淘汰されやすい. そのため, 再びこう配の喪失が発生し, ホスト集団に現れた突然変異体もまた失われる.

それに対して, RS 以外では, こう配の喪失が発生した試行回数は世代が進むにつれて減少しており, 特に, TR, RPV, B-FS と ES の四つについては, 35

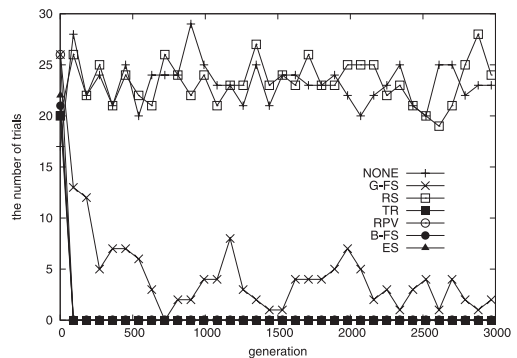


図 6 こう配の喪失が発生した試行数
Fig. 6 The number of trials in which loss of gradient occurred at a generation.

(注3): 0 と 1 からなる長さ n の入力データすべてを正確に処理できるソーティングネットワークは長さ n の任意のデータについても正確に処理することができることが証明されており, これを 0-1 原理という [29].

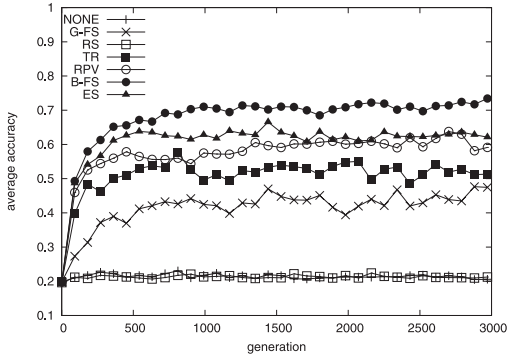


図 7 平均正答率の推移

Fig. 7 Average accuracy of sorting networks obtained by each method.

世代目以降では完全に防いでいた。これは、いくつかの宿主個体に正確にソートされたパラサイト個体の適応度が、宿主個体すべてに対して難しいパラサイト個体よりも高くなる状況を作り出したためである。このような状況では、若干簡単なパラサイト個体が増え、それによりそのパラサイト個体をソートできる宿主個体も増え、宿主集団の進化が促される。RSはこの状況を作ることができなかったために、NONEと同様の結果であった。

図 7 に、各手法により得られたソーティングネットワークの平均正答率の推移を示す。NONE、及びRSでは、ほとんどの世代でこう配の喪失が発生していたために正答率は0.2付近で停滞していたが、その他の手法では、こう配の喪失が抑えられたことで正答率が上昇していた。G-FSは、こう配の喪失を完全に抑えられなかったが、平均正答率は0.47まで上昇していた。B-FS、ES、RPVとTRはすべて、およそ35世代以降にこう配の喪失を完全に防いでいたが、B-FSを除く三つは、およそ500世代から正答率が0.65未満で停滞していた。これは、TRについては、難度を上昇させられず軍拡競争が実現できなかったためであり(図8)、ESとRPVについては、簡単なテストデータが淘汰されやすく、宿主集団に提示するテストデータの難度の多様性が低くなっていたことが原因と考えられる(図9)。それに対し、B-FSは、序盤に簡単なテストデータへ進化させたあと、難度の多様性を保ちながら難しいテストデータへと進化させることができていた。それにより、500世代目には既に0.65を超え、その後、最終世代までに0.730へ緩やかに上昇した。

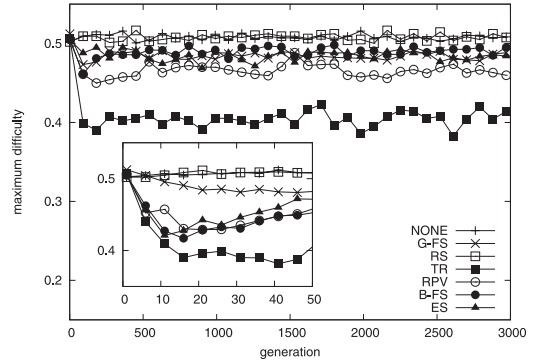


図 8 最大難度の推移。各世代の値は、各試行においてその世代で得られたテストデータの難度の最大値を全試行について平均化したもの

Fig. 8 Maximum difficulty of test data obtained by each method (averaged over 30 runs).

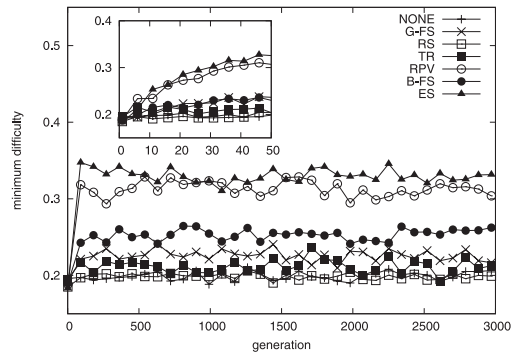


図 9 最小難度の推移。各世代の値は、図 8 と同様の方法で、最小値について計算したのもの

Fig. 9 Minimum difficulty of test data obtained by each method (averaged over 30 runs).

4. 考 察

前章では、競合型共進化アルゴリズムの性能評価によく用いられる二つの問題を用いて、過度の特殊化、及びこう配の喪失、それぞれに対する解決手法計7個を比較評価した。本章では、比較評価の結果を踏まえながら、より一般的な解釈を試みる。

一次元ナンバーズゲームは、戦略間の優劣が線形順序である典型的な問題であり、絶対適応度または適応度が等しいパラサイト個体同士は、どの相手に対しても全く同じ評価を与えるのが特徴である。この問題に対し、唯一の遺伝的操作である突然変異を一方の集団に対してのみ優位に働かせ、その割合をパラメータ B_p で調整することにより、こう配の喪失の生じやすさに対する各手法の性能を調べた。実験結果は、過度の特

殊化に対する解決手法がこう配の喪失に対しても有効であることを支持するものであった。ただし、こう配の喪失が非常に発生しやすい状況では、G-FS、P-FS、RS はそれを防げなかった。これに対して、B-FS と ES は防ぐことができ、問題の難易度に比較的左右されずに良い解を発見し得ることが示された。一方、こう配の喪失に対する従来の解決手法である RPV は、確かにこう配の喪失を抑えることには成功している。しかし、パラサイト集団の探索空間上の良解が発見しやすければ (B_p 大) 良い解を得ているが、良解が発見しにくい場合 (B_p 小) には、競合型共進化アルゴリズムのみを適用した場合をも下回ってしまうことが、得られたホスト個体の絶対適応度により示された (図 3)。これは、ホスト集団が進化し続けるにはパラサイト集団も伴って進化し続けることが必要であるが、良解を発見しやすい問題では、RPV が優良個体を淘汰してもパラサイト集団の進化は安定して進むのに対して、良解を発見しにくい問題では、温存が必要な優良個体を残せずパラサイトの進化が停滞するためであると考えられる (図 4)。ES についても RPV と同様な傾向が見られ、これは適応度関数が類似していることから上記と同様の説明が可能である。

一方、ソーティングネットワーク設計問題は、一次元ナンバースゲームと違って、同じ適応度でも相手次第で異なる評価を与え得る典型的な問題である。評価実験では、初期世代でこう配の喪失が既に発生しており、適応度のこう配を生み出すためには、簡単なテストデータをいったん増やすことが必要となる。過度の特殊化の解決手法のうち、RS は簡単なテストデータを増やせなかったためこう配を生み出すことができなかったが、G-FS はより簡単なテストデータへ緩やかに進化させることができ、こう配の喪失を抑えていた。一方、B-FS と ES は、こう配の喪失の解決手法である TR や RPV と同様に、早い世代のうちにこう配の喪失を完全に防ぐことができ、テストデータを難しいものへ進化させることができていた。いったん、簡単なテストデータへ進化させた後、ホスト集団とともに軍拡競争をさせるために、テストデータを難しいものへ進化させる必要がある。実験結果より、難度をどれだけ上昇させられたかによって、得られたソーティングネットワークの正答率が異なることが示されており、特に B-FS は、今回比較した手法の中で最も良いソーティングネットワークを得た。この問題において軍拡競争をするために必要な難しいテストデータは発見し

づらいため、上述の一次元ナンバースゲームの場合と同様の理由から、ES や RPV は良解を発見できなかったと考えられる。また、この問題のように、個体間の優劣に線形順序をもたない問題の場合、良解を探索するには、適応度上の多様性を生み出すことで適応度のこう配を与えるだけでなく、どのような相手に対してどのような評価を与えているかを考慮することがポイントとなる。そのため、適応度についてしか考慮しない RPV や ES に比べ、評価結果の多様性を維持できる B-FS がより良い解を得たと推測できる。

ニムゲーム [30] や三次元マルバツゲーム [30]、オセロ [31] などのように先後手が決まっているゲームの戦略の競合型共進化アルゴリズムによる探索や捕食-被食者問題は、一方の集団が他方に対して優位になりやすく、かつ優良な解が発見しづらい問題である。これらの問題に対して資源共有法など評価結果の多様性を生み出す手法を適用することで、良解が得られたことが報告されており、このことから、これらの問題に対しても B-FS が有効に働くのではないかと考えられる。

手法の評価方法として、実際の問題に適用してどれだけ解決できるのかを評価する方法もあるが、1. で述べたように競合型共進化アルゴリズムは解の絶対的評価が困難な問題を対象とできる点に特徴があり、また、比較評価も複雑化する。そこで、本研究は、手法の比較検討を目的として、注目しているこう配の喪失のみが生じ、なおかつ解の絶対的評価が可能なベンチマーク問題に適用し、評価する方法を採用し、基本的な性質を探っている段階である。

5. む す び

過度の特殊化に対する解決手法のこう配の喪失に対する有効性、及び我々の提案手法の有効性を検証するために、一次元ナンバースゲーム、及び SN 設計問題による計算機実験により比較評価を行った。

一次元ナンバースゲームによる評価実験では、こう配の喪失の生じやすさに対する各手法の性能を調べた。実験結果は、過度の特殊化に対する解決手法がこう配の喪失に対しても有効であることを支持するものと考えられる。特に、B-FS と ES はこう配の喪失が非常に発生しやすい状況でも、それを防ぐことができ、そして、ホスト集団の絶対適応度の結果から、得られた解の質においてもこう配の喪失の解決手法に近い性能であった。

より複雑なタスクにおける性能評価のために、SN 設計問題による評価実験を行った。この実験では、初期世代でこう配の喪失が既に発生しており、そこから適応度差を生み出すためには、簡単なテストデータを増やすことが必要となる。過度の特殊化の解決手法のうち、RS は簡単なテストデータを増やせず、G-FS は緩やかにテストデータをより簡単なものへ進化させ、ソーティングネットワークの正答率を上昇させた。一方、B-FS と ES は、こう配の喪失の解決手法である TR や RPV と同様に、こう配の喪失を完全に防ぐことができた。そして、これらはこう配の喪失の解決手法よりも高い正答率をもつソーティングネットワークを発見し、特に B-FS は、今回比較した手法の中で最も良いソーティングネットワークを得た。

先行研究 [11] において、B-FS と ES は過度の特殊化を解決し得ることが報告されている。更に、本研究の結果を踏まえることで、これら 2 手法は両問題点を解決し得ることが示唆される。

謝辞 本研究の一部は、文部科学省 21 世紀 COE 「計算科学フロンティア」の援助による。

文 献

- [1] M.A. Potter and K.A. De Jong, "Cooperative co-evolution: An architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol.8, no.1, pp.1-29, 2000.
- [2] W.D. Hillis, "Co-evolving parasites improve simulated evolution as an optimization procedure," in *Artificial Life II*, pp.313-324, Addison-Wesley 1992.
- [3] H. Juillé and J.B. Pollack, "Coevolving the ideal trainer: Application to the discovery of cellular automata rules," *Proc. Third Annual Genetic Programming Conference*, pp.519-527, 1998.
- [4] N. Williams and M. Mitchell, "Investigating the success of spatial coevolution," *Proc. Genetic and Evolutionary Computation Conference*, pp.523-530, 2005.
- [5] R.A. Watson and J.B. Pollack, "Coevolutionary dynamics in a minimal substrate," *Proc. Genetic and Evolutionary Computation Conference*, pp.702-709, 2001.
- [6] H. Juillé and J.B. Pollack, "Co-evolving intertwined spirals," *Proc. Fifth Annual Conference on Evolutionary Programming*, pp.461-468, 1996.
- [7] J. Paredis, "Coevolving cellular automata: Be aware of the red queen!" *Proc. 7th International Conference on Genetic Algorithms*, pp.393-400, 1997.
- [8] R.E. Smith and B. Gray, "Co-adaptive genetic algorithms: An example in othello strategy," *Proc. Florida Artificial Intelligence Research Symposium*, pp.259-264, 1994.
- [9] H. Juillé and J.B. Pollack, "Coevolutionary learning: A case study," *Proc. Fifteenth International Conference on Machine Learning*, pp.251-259, 1998.
- [10] S.G. Ficici and J.B. Pollack, "Pareto optimality in coevolutionary learning," *Advances in Artificial Life: 6th European Conference (ECAL 2001)*, pp.316-325, 2001.
- [11] 佐藤竜也, 有田隆也, "競合型共進化アルゴリズムにおける過度の特殊化問題に対する適応度共有法の適用," *情処学論*, vol.49, no.1, pp.476-486, 2008.
- [12] J. Cartlidge and S. Bullock, "Combating coevolutionary disengagement by reducing parasite virulence," *Evolutionary Computation*, vol.12, no.2, pp.193-222, 2004.
- [13] S.G. Ficici, "Monotonic solution concepts in coevolution," *Proc. Genetic and Evolutionary Computation Conference*, pp.499-506, 2005.
- [14] J. Reisinger, E. Bahceci, I. Karpov, and R. Miikkulainen, "Coevolving strategies for general game playing," *Proc. 3rd Annual IEEE Symposium on Computational Intelligence and Game*, pp.320-327, 2007.
- [15] A. Bucci, *Emergent geometric organization and informative dimensions in coevolutionary algorithms*, PhD Thesis, Brandeis University, 2007.
- [16] B. Olsson, "Evaluation of a simple host-parasite genetic algorithm," *Proc. Evolutionary Programming VII*, pp.53-62, 1998.
- [17] D. Cliff and G.F. Miller, "Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations," *European Conference on Artificial Life*, pp.200-218, 1995.
- [18] 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.
- [19] A. Bucci and J.B. Pollack, "Order-theoretic analysis of coevolution problems: Coevolutionary statics," *Proc. Genetic and Evolutionary Computation Conference*, pp.229-235, 2002.
- [20] J.C. Bongard and H. Lipson, "'managed challenge' alleviates disengagement in co-evolutionary system identification," *Proc. Genetic and Evolutionary Computation Conference*, pp.531-538, 2005.
- [21] C.D. Rosin, *Coevolutionary search among adversaries*, PhD Thesis, University of California, San Diego, CA, 1997.
- [22] S.G. Ficici and J.B. Pollack, "Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable state," *Proc. Sixth International Conference on Artificial Life*, pp.238-247, 1998.
- [23] D.E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," *Proc. Second International Conference on Genetic Algorithms*, pp.41-49, 1987.
- [24] S.G. Ficici and J.B. Pollack, "A game-theoretic mem-

- ory mechanism for coevolution,” Proc. Genetic and Evolutionary Computation Conference, pp.286–297, 2003.
- [25] E.D. De Jong and A. Bucci, “DECA: Dimension extracting coevolutionary algorithm,” Proc. Genetic and Evolutionary Computation Conference, pp.313–320, 2006.
- [26] E.D. De Jong, “A monotonic archive for pareto-coevolution,” Evolutionary Computation, vol.15, no.1, pp.61–93, 2007.
- [27] H. Juillé, “Incremental co-evolution of organisms: A new approach for optimization and discovery of strategies,” Third European Conference on Advances in Artificial Life, pp.246–260, 1995.
- [28] S. Ronald, “More distance functions for order based encodings,” Proc. IEEE Conference on Evolutionary Computation, pp.558–563, 1998.
- [29] D.E. Knuth, The art of computer programming: Volume 3 — Sorting and searching, Addison Wesley, 1973.
- [30] C.D. Rosin and R.K. Belew, “New methods for competitive coevolution,” Evolutionary Computation, vol.5, no.1, pp.1–29, 1997.
- [31] R.E. Smith, S. Forrest, and A.S. Perelson, “Searching for diverse, cooperative populations with genetic algorithms,” Evolutionary Computation, vol.1, no.2, pp.127–149, 1993.

(平成20年8月22日受付, 21年3月13日再受付)



佐藤 竜也

2004 群馬大・工・情報工学卒。2006 名古屋大学大学院情報科学研究科博士課程（前期課程）了。2009 同大学院博士課程（後期課程）単位取得満期退学。在学中，進化的計算手法の研究に従事。現在，楽天（株）に勤務。



有田 隆也（正員）

1983 東大・工・計数工学卒。1988 同大学院工学系研究科博士課程了。工博。名古屋工業大学講師，カリフォルニア大学ロサンゼルス校客員研究員を経て，現在，名古屋大学大学院情報科学研究科教授。人工生命や情報科学の研究に従事。複雑適応系，進化ダイナミックス，言語の進化などに興味をもつ。著書に“人工生命”（医学出版，2002年），“心はプログラムできるか”（ソフトバンククリエイティブ，2007年）など。