

遺伝的プログラミングでの多様性維持について

Diversity Maintenance in Genetic Programming

元木 達也
Tatsuya MOTOKI

新潟大学工学部
Faculty of Engineering, Niigata University
motoki@ie.niigata-u.ac.jp

沼口 靖
Yasushi NUMAGUCHI

日立化成工業株式会社
Hitachi Chemical Co., Ltd.

keywords: genetic programming, diversity maintenance, fitness sharing

Summary

This paper is motivated by an experimental result that better performing genetic programming runs tend to have higher phenotypic diversity. To maintain phenotypic diversity, we apply implicit fitness sharing and its variant, called unfitness multiplying. To apply these methods to problems in which individuals have infinite kinds of possible behaviours, we classify possible behaviours into 50 achievement levels, and assign a reward or a penalty to each level. In implicit fitness sharing a reward is shared out among individuals with the same achievement level, and in unfitness multiplying a penalty is multiplied by the number of individuals with the same level and is distributed to related individuals. Five benchmark problems (11-multiplexer, sextic polynomial, four-sine, intertwined spiral, and artificial ant problems) are used to illustrate the effect of the methods. The results show that our methods clearly promote diversity and lead population to a smooth frequency distribution of achievement levels, and that our methods usually perform better than the original implicit fitness sharing on success rate and the best (raw) fitness. We also observe that the unfitness multiplying makes a quite different ranking over individuals than the one by the implicit fitness sharing.

1. は じ め に

遺伝的プログラミング (Genetic Programming: GP) は進化的計算の一種で, Koza の研究以来 1 つの研究分野を構成し, 広範囲の分野で応用されている [Koza 92, Koza 94, Banzhaf 98, 伊庭 05]. 遺伝的アルゴリズム (以下 GA) を始めとする他の進化的探索手法と同様に, GP においても集団内の個体の多様性を保つことは早熟な収束や進化の停滞を防ぐために重要な課題となっている.

集団内の個体の多様性を保つために, GA においては個体間の距離を測り探索空間内で他に類似した個体が集団内にどの程度あるかに応じて個体の適合度に補正を加える, シェアリング (sharing) [Goldberg 87] と呼ばれる手法が提案され一般に用いられているが, GP においては個体が木構造をしていてその構造も不定であるため個体間の距離の定義を迷いなく一意に行える状況にない. そのため GP においては, 集団における個体の多様性を実際にどう測るのかについて色々な考え方がある [Banzhaf 98, Burke 02, Burke 04]. その様な中, Burke ら (2002) は GP における集団の多様性の測度として①初期集団中の節点のうち何個が生き残っているか, ②初期集団中の根節点のうち何個が根節点として生き続けているか,

③素適合度値の分布に基づいて定義されたエントロピー $-\sum_k p_k \log p_k$, ④個体の大雑把な特徴を並べた \langle 終端節点数, 関数節点数, 高さ \rangle の組が何種類集団内にあるか, ⑤互いに構成の異なる木 (個体) が集団内に何個あるか, ⑥素適合度の値が集団内に何種類あるか, の 6 つを挙げ, これら 6 つの多様性測度が世代と伴にどう変わっていくかを人工蟻問題, 関数同定問題 ($x^4 + x^3 + x^2 + x$), 偶 5 パリティ問題の GP 探索で観察した. そして, この観察結果と最終世代の最良適合度の間の相関を調べ, 調べた 3 つの問題に関しては個体の振舞に関する多様性 (③, ⑥) が保たれている方が木の構成に関する多様性 (①, ②, ④, ⑤) が保たれている場合よりも良個体が探索でき易くなっていることを報告した. このことから逆に, 個体の振舞に関して多様性を保つ様を選択を行って GP 探索を行えば, より効果的な探索が出来るのではないかという期待が生まれる. そこで実際にこの方向の GP 探索を実験的に調べた研究として, ここでは論文 [McKay 00] に注目する.

McKay は GP を適用する問題として, (1) 幾つかの部分問題に分解され各個体の適合度が $\sum_{P: \text{部分問題}} (P \text{ における個体の性能評価値})$ と表せ, かつ (2) 各部分問題において個体の振舞結果が有限個の少ない種類に分類される, ようなものを想定した. そして, 特に個体の動作結果が

正しいか否かの 2 つにはっきりと分かれる問題に対して、個体間の距離を明示的に導入することなく (シェアリングと同様に) 個体の性能評価値 (報酬) を同じ動作の個体間で分け合わせて適合度を計算する手法を提案し、それを陰的適合度シェアリング (implicit fitness sharing) と呼んだ。McKay はこのシェアリング手法を 6 マルチプレクサ問題, 11 マルチプレクサ問題, リスト要素のメンバシップ判定問題に適用し、その効果を確認した。しかし、McKay の実験によって個体の振舞に基づいて適合度を補正し多様性を維持する手法が十分に調べられたとは言いがたい。なぜなら、シェアリングの考えに沿わなくても、多様性維持に繋がる適合度補正は可能であるからである。陰的適合度シェアリングに限っても、関数同定問題の様に個体の動作をどれだけ成功に近いかによって 3 つ以上に区分することが自然な場合もあるが、このような区分の場合に性能評価値 (報酬) をどう個体間で分け合わせるのかは示されていない。

そこで、この論文では個体の振舞に基づいて適合度を補正し多様性を維持する手法についての知見を深め、GP の多くの応用で効率的な探索をもたらす手法を整備することを目的とする。そのために、特殊な状況にのみうまく対応し全体としては異常な振舞が多い個体を排除する不適合度積算 (unfitness multiplying) の考え方を提案し、この手法と陰的適合度シェアリング手法を 11 マルチプレクサ問題, 関数同定問題 ($x^6 - 2x^4 + x^2, \sin x + \sin 2x + \sin 3x + \sin 4x$), 絡み螺旋問題, 人工蟻問題に適用し、それらの効果を実験的に調べ比較する。また、特に関数同定問題と絡み螺旋問題においては、個体の振舞を成功と不成功の 2 つに区分する方法だけでなく、振舞の良さの程度に応じて 50 種類に区分して陰的適合度シェアリング, 不適合度積算を行う方法もテストする。

以下、2 章では個体の振舞に関する多様性維持に効果のある適合度補正の方法として、陰的適合度シェアリングを紹介し、不適合度積算の手法、個体の振舞が 3 つ以上の場合の対処法を提案する。続く 3 章では実験に用いるテスト問題と GP 探索の処理手順、実験結果について述べる。4 章では多様性維持の様子、陰的適合度シェアリングと不適合度積算の関係、手法適用の際のパラメータ調節、多様性維持が成功率向上につながらない問題についての考察を行う。最後の 5 章ではまとめを行う。

2. 多様性維持のための適合度設定

本章では、個体の振舞面の多様性維持に効果のある適合度補正の方法として、McKay の陰的適合度シェアリングを紹介し、不適合度積算の手法、個体の振舞が 3 つ以上の場合の対処法を提案する。

(従来手法) 陰的適合度シェアリング [McKay 00]: 個体プログラムを適用する事例 (fitness case) が有限個で、個体 i を事例 c に適用した時の個体の動作を $i(c)$, 動作 $i(c)$

が正しい時は 1 をとり間違っている時は 0 をとる量を $\text{reward}(i(c))$ と表すとき、シェアリングによって個体 i に分配される適合度 (shared fitness) $f_{\text{shr}}(i)$ を

$$f_{\text{shr}}(i) = \sum_{c: \text{事例}} \frac{\text{reward}(i(c))}{\#\{j \in P | j(c) = i(c)\}} \quad (1)$$

と定める。ここで、 P は現集団を表し、 $\#S$ は集合 S 内の要素の個数を表す。 $f_{\text{shr}}(i)$ においては、 $\text{reward}(i(c))$ に掛かる係数は正しい動作が出にくくなるに従って大きくなる。そのため、その様な事例で正しい動作をする個体が適合度評価・選択の際に優遇される。分け合う資源・報酬が明示されている点で、この手法は通常シェアリング [Goldberg 87] と比べて自然界の棲み分けにより近いものとなっている。

(提案手法 1) 動作結果が 3 種類以上の問題に対する拡張: 個体の動作結果が有限の少ない種類しかない場合も、0 と 1 以外の $\text{reward}()$ を使ってやはり式 (1) に基づいて適合度補正を行うことが出来る。これに対して動作結果が無限もしくは多数種類可能な場合は、集団内で各々の個体が全く独自の動作をすることが多くなるので、シェアリングの効果は薄れ式 (1) は $\sum_{c: \text{事例}} \text{reward}(i(c))$ とほぼ等しくなってしまう。そこで、個体の動作結果をどれだけ望ましい動作をするかに応じて比較的少ない種類に大雑把に区分する。個体 i の事例 c に対する達成度レベルを $\text{level}(i(c))$ で表し、達成度レベル $\text{level}(i(c))$ の個体全体で 1 つの報酬 $\text{reward}(\text{level}(i(c)))$ を分け合うことにすれば、式 (1) は次の様に一般化される。

$$f_{\text{shr}*}(i) = \sum_{c: \text{事例}} \frac{\text{reward}(\text{level}(i(c)))}{\#\{j \in P | \text{level}(j(c)) = \text{level}(i(c))\}} \quad (2)$$

実際の達成度レベルの設定に関しては、この論文では適正な動作をレベル 0 とし、誤りの動作をその誤りの程度に応じてレベル 1, 2, 3, ... のいずれかに分類した。

(提案手法 2) 不適合度積算 (unfitness multiplying): 動作に対する報酬を考えて個体の適合度を測る代わりに、動作に対するペナルティを考えて個体の不適合度 (unfitness) を測ることにした場合、類似個体が増えるに従ってペナルティが増す様にすればシェアリングに相当する効果が期待できる。それゆえ、式 (1), (2) の代わりに個体 i の積算不適合度 (multiplied unfitness)

$$u_{\text{mul}*}(i) = \sum_{c: \text{事例}} \text{penalty}(i(c)) \times \#\{j \in P | \text{level}(j(c)) = \text{level}(i(c))\} \quad (3)$$

を用いて選択を行っても、多様性維持の効果が期待できる。また、誤った個体が増えるに従って一律ペナルティが増える様に設定して、不適合度

$$u_{\text{mul}2}(i) = \sum_{c: \text{事例}} \text{penalty}(i(c)) \times \#\{j \in P | j \text{ が } c \text{ を誤って処理}\} \quad (4)$$

を用いて選択を行っても、多様性維持の効果が期待できる。これらの積算不適合度は式の上では単なる式 (2) の変形にすぎないが、これらの式 (3), (4) が式 (1), (2) と全く違った振舞 (個体のランキング) をすることが 4.2 節で明らかになる。

最後に、比較のために多様性を損なう方向の適合度補正も考える。

多様性喪失を促進する適合度設定: 式 (4) とは逆に次の様に誤った個体が増えるにつれて一律ペナルティを軽減させれば、多様性を損なうことになると考えられる。

$$u_{shr2}(i) = \sum_{c:i \text{ が誤る事例}} \frac{\text{penalty}(i(c))}{\#\{j \in P | j \text{ が } c \text{ を誤って処理}\}} \quad (5)$$

3. 実 験

本章では、2 章で紹介・提案した手法の評価実験を行う。まず 3.1 節で実験に用いたテスト問題を説明し、3.2 節でテスト問題に対して適合度補正の式 (1) ~ (5) を具体的に設定する。続く 3.3 節で実験に用いた GP の世代交代モデル、GP のパラメータについて述べ、3.4 節では予備実験として各々の手法で最適なトーナメントサイズを設定する。最後の 3.5 節で実験結果を示す。

3.1 対 象 問 題

性質の異なる問題として文献 [Koza 92, Koza 94] から 11 マルチプレクサ問題、関数同定問題 (2 題)、絡み螺旋問題、人工蟻問題 (Santa Fe trail) の 4 種類を選び、これらを評価実験に用いた。

11 マルチプレクサ問題 [Koza 92]: 11 マルチプレクサはアドレス部 3 ビット $a_0 \sim a_2$ とデータ部 8 ビット $d_0 \sim d_7$ の合計 11 ビットの入力から成るブール関数で、アドレス部の指定したアドレス $\alpha = \sum_i a_i \times 2^i$ を基にそのアドレスのデータ値 d_α を出力する関数 $\text{multiplexer11}(a_0, a_1, a_2, d_0, \dots, d_7) = d_{4a_2+2a_1+a_0}$ である。11 マルチプレクサ問題は、その全ての入力の種類 (2048 組) に対して 11 マルチプレクサ関数と同じ出力をするブール式を探し出す問題である。ここでは、用いることができる関数記号の集合を $F = \{\text{and, or, not, if-then-else}\}$ 、終端子の集合を $T = \{a_0, a_1, a_2, d_0, \dots, d_7\}$ とする。素適合度 (raw fitness) は次の様に定める。

$$f_{\text{raw}}(h) = \begin{cases} 2048 \text{ 組の入力事例の内,} \\ \text{個体 } h \text{ が 11 マルチプレクサ} \\ \text{と同じ出力をもたらす回数} \end{cases} \quad (6)$$

6 次多項式 $x^6 - 2x^4 + x^2$ 同定問題 [Koza 94]: これは、 $x_i = 0.1i - 1, y_i = x_i^6 - 2x_i^4 + x_i^2$ として、21 個の点 $(x_0, y_0), \dots, (x_{20}, y_{20})$ を近似的に通る関数 $y = h(x)$ を x の式の形で探し出す問題である。ここでは、各点の誤差の許容の上限を 0.01 とし、各点で $|y_i - h(x_i)| \leq 0.01$ と

なる関数の式を探すことにする。そして、式を構成する関数記号の集合を $F = \{+, -, \times, \text{pdiv}, \text{ifte}\}$ 、終端子の集合を $T = \{x, 1\}$ とする。pdiv と ifte は各々保護付き除算 (protected division) と条件式で、 $\text{pdiv}(x, y) \stackrel{\text{def}}{=} \text{if } y = 0 \text{ then } 1 \text{ else } x/y$, $\text{ifte}(a, b, c, d) \stackrel{\text{def}}{=} \text{if } a \leq b \text{ then } c \text{ else } d$ という演算を表す。また、式 h の素適合度は次の様に定める。

$$f_{\text{raw}}(h) = \frac{1}{21} \sum_{i=0}^{20} |y_i - h(x_i)| \quad (7)$$

4sin 問題 [Koza 94]: これは、 $x_i = 2\pi i/49 - \pi, y_i = \sin x_i + \sin 2x_i + \sin 3x_i + \sin 4x_i$ として、50 個の点 $(x_0, y_0), \dots, (x_{49}, y_{49})$ を近似的に通る関数 $y = h(x)$ を x の式の形で探し出す問題である。ここでは、式を構成する関数記号の集合を $F = \{+, -, \times, \text{pdiv}\}$ 、終端子の集合を $T = \{x\} \cup \{\text{区間 } [-\pi, \pi] \text{ 内のランダム定数 } 200 \text{ 個}\}$ とする。関数記号の中に sin 関数が入っていないので全ての点で十分な近似となることは難しく、そのためこの問題では各点の誤差の許容の上限を 0.4 とする。式 h の素適合度は次の様に定める。

$$f_{\text{raw}}(h) = \frac{1}{50} \sum_{i=0}^{49} |y_i - h(x_i)| \quad (8)$$

絡み螺旋問題 [Koza 92]: $\theta_i = \pi i/16, r_i = 6.5(104 - i)/104$ として、 $x_i^{(1)} = r_i \sin \theta_i, y_i^{(1)} = r_i \cos \theta_i$ によって決まる 97 個の点 $(x_0^{(1)}, y_0^{(1)}), \dots, (x_{96}^{(1)}, y_{96}^{(1)})$ と $x_i^{(2)} = -x_i^{(1)}, y_i^{(2)} = -y_i^{(1)}$ によって決まる 97 個の点 $(x_0^{(2)}, y_0^{(2)}), \dots, (x_{96}^{(2)}, y_{96}^{(2)})$ がある。絡み螺旋問題は一般には平面上に絡み合って螺旋状に配置されたこれら 2 種類、合計 194 個の点について、その点の座標から点の種類を正しく識別する問題である。GP においては、2 変数 x, y を引数として与えた時の値が 0 以上か 0 未満かで各々の点がどちらの螺旋に属しているかを正しく分類する関数 $h(x, y)$ 、すなわち、1 番目の種類の点 $(x_i^{(1)}, y_i^{(1)})$ に対しては $h(x_i^{(1)}, y_i^{(1)}) \geq 0$ で 2 番目の種類の点 $(x_i^{(2)}, y_i^{(2)})$ に対しては $h(x_i^{(2)}, y_i^{(2)}) < 0$ となる様な関数 $h(x, y)$ を x と y の式の形で得ることが目的となる。ここでは、用いることができる関数記号の集合は $F = \{+, -, \times, \text{pdiv}, \text{ifte}, \sin, \cos\}$ 、終端子の集合は $T = \{x, y\} \cup \{\text{区間 } [-1, 1] \text{ 内のランダム定数 } 200 \text{ 個}\}$ とする。そして、素適合度は次の様に定める。

$$f_{\text{raw}}(h) = \frac{194 \text{ 個の点の内, 個体 } h \text{ が}}{\text{正しく分類する点の個数}} \quad (9)$$

人工蟻問題 (Santa Fe trail) [Koza 92]: 32×32 のマス目が 2 次元トーラス状に繋がり、その内 89 個のマス目に餌が、1 つのマス目に 1 匹の人工蟻が図 1 の様に配置されている。人工蟻問題 (Santa Fe trail) は、この初期状態から始めて 400 ステップ以内に餌を全て獲得する人工蟻の動作プログラムを探索する問題である。ここで、人工蟻は上下左右のいずれかの方向に向いており、左右への 90 度回転、前方への 1 マス前進に各 1 ステップの時間を

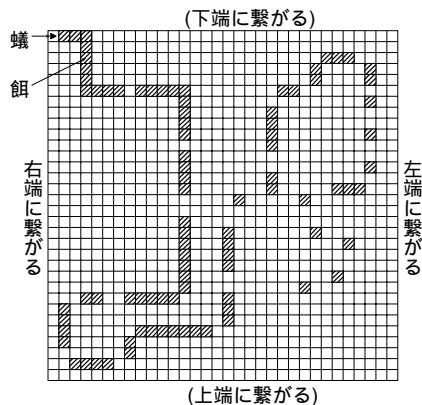


図 1 人工蟻問題の初期配置 (Santa Fe trail)

費すものとする．また，移動先のマス目に餌があれば，人工蟻はそれを必ず獲得する．人工蟻の動作プログラムは関数記号の集合 $F = \{\text{if-food-ahead}(\text{目の前のマスに餌があれば第 1 引数, なければ第 2 引数のプログラムを実行}), \text{prog2}(\text{第 1, 第 2 引数のプログラムを順に実行}), \text{prog3}(\text{第 1, 第 2, 第 3 引数のプログラムを順に実行})\}$ と終端子の集合 $T = \{\text{move-forward, turn-left, turn-right}\}$ から構成され，人工蟻の動作ステップ数が 400 に満たない間は繰り返し実行されるものとする．素適合度は次の様に定める．

$$f_{\text{raw}}(h) = \frac{\text{動作プログラム (個体) } h \text{ を用いた時に人工蟻が獲得する餌の個数}}{\quad} \quad (10)$$

3.2 従来手法，提案手法の適用

この論文でも [McKay 00] と同様に，GP を適用する問題が有限個の部分問題 (事例) に分解され，各個体の適合度が $\sum_{c:\text{事例}} (c \text{ における個体の性能評価値})$ と表されると仮定する．11 マルチプレクサ問題においては 2048 組の入力パターンの各々が，関数同定問題においては x_i 値の各々が事例であり，絡み螺旋問題においては 194 個の点の各々が事例である．また，人工蟻問題では，本来は 1 つの初期配置を 1 つの事例として扱うのが妥当と考えられるが，ここでは Santa Fe trail という餌の配置を全体の問題と捉え，餌の 1 個 1 個を 1 つの事例として扱う．

以下では式 (1) ~ (5) を具体化した 5 つの適合度を考え，それらを用いて選択を行う GP 探索，素適合度 (6) ~ (10) を用いて選択を行う GP 探索を比較する．

個体動作を 2 つに分類した設定: 3.1 節で挙げたどの問題においても，各事例に対する個体の動作を適正と誤りの 2 つに分類することができる．例えば，11 マルチプレクサ問題，絡み螺旋問題においては与えられた入力事例に対して個体が正しい出力結果を出すことを適正な動作と考え，関数同定問題においては与えられた事例 (x 値) に対する個体の値が目的とする値からの許容範囲内にあることが適正な動作と考える．人工蟻問題においては，与えら

れた事例 (1 つの餌) に対して，個体プログラムの制御下で動作する人工蟻が制限時間内にその餌を獲得できることが (その餌だけに関しては) 適正な動作と考える．この分類の下で，適正な動作 $i(c)$ に対して $\text{reward}(i(c)) = 1$ ，誤りの動作 $i(c)$ に対して $\text{reward}(i(c)) = 0$ とすれば，式 (1) は次のようになる．

$$f_{\text{shr2}}(i) = \sum_{c:i \text{ が適正に動く事例}} \frac{1}{\left(\frac{c \text{ に対して適正に動く個体数}}{\quad} \right)} \quad (11)$$

また，適正な動作 $i(c)$ に対して $\text{penalty}(i(c)) = 0$ ，誤りの動作 $i(c)$ に対して $\text{penalty}(i(c)) = 1$ とすれば，不適合度の式 (4), (5) は次のようになる．

$$u_{\text{mul2}}(i) = \sum_{c:i \text{ が誤る事例}} (c \text{ に対して誤る個体数}) \quad (12)$$

$$u_{\text{shr2}}(i) = \sum_{c:i \text{ が誤る事例}} \frac{1}{(c \text{ に対して誤る個体数})} \quad (13)$$

この $u_{\text{shr2}}(i)$ は単に比較のためだけのもので，多様性維持には有害である．

個体動作を 50 に分類した設定: 関数同定問題や絡み螺旋問題の様に，個体の表す式が実数値をとる場合は，個体の振舞も千差万別で，適正の基準をもう少しで満たす個体から全く満たさない個体まで様々である．そこで，個体の振舞の良さの程度に応じて 50 種の達成度レベルに区分する．特に関数同定問題においては，各事例 (x 値) における誤差許容の上限 ϵ (6 次関数問題で 0.01, 4sin 問題で 0.4) と，事例 c における個体 i の値と目標とする値の間の実際の誤差 $\text{error}(i, c)$ を用いて，個体の振舞の達成度 $\text{level}(i(c))$ を次の様に設定する．

$$\text{level}(i(c)) = \begin{cases} 0 & \text{if } |\text{error}(i, c)| < \epsilon \\ 1 & \text{if } |\text{error}(i, c)| \in [2^0\epsilon, 2^1\epsilon) \\ 2 & \text{if } |\text{error}(i, c)| \in [2^1\epsilon, 2^2\epsilon) \\ \dots\dots\dots \\ 49 & \text{if } 2^{48}\epsilon \leq |\text{error}(i, c)| \end{cases} \quad (14)$$

絡み螺旋問題においては，事例 c における個体 i の値 $i(c)$ を用いて，個体の振舞の達成度 $\text{level}(i(c))$ を次の様に設定する．

$$\text{level}(i(c)) = \begin{cases} 0 & \text{if } i \text{ が } c \text{ を正しく分類} \\ 1 & \text{if 誤分類, } |i(c)| < 2^{-2} \\ 2 & \text{if 誤分類, } |i(c)| \in [2^{-2}, 2^{-1}) \\ 3 & \text{if 誤分類, } |i(c)| \in [2^{-1}, 2^0) \\ \dots\dots\dots \\ 49 & \text{if 誤分類, } 2^{45} \leq |i(c)| \end{cases} \quad (15)$$

表 1 トーナメントサイズによる成功率の変動: 丸括弧内は最終の最良適合度の平均・太字はその行での最良の成功率で、これをもたらししたトーナメントサイズ (tsize) を良好な tsize と判断した。太字がない行では、最終の最良適合度の平均も参考にして良好な tsize の判断を行った。

問題	手法	トーナメントサイズ (tsize) の設定						良好な tsize
		tsize=2	tsize=3	tsize=4	tsize=5	tsize=6	tsize=7	
11-Multiplexer 問題 (300 試行)	f_{raw}	1.3%	5.0%	10.7%	13.3%	10.3%	11.3%	5
	f_{shr2}	65.7%	93.3%	97.0%	97.7%	98.3%	97.0%	6
	u_{mul2}	78.0%	99.3%	100.0%	99.7%	99.7%	100.0%	6
人工蟻問題 (100 試行)	f_{raw}	24%	37%	42%	32%	43%	27%	6
	f_{shr2}	12%	20%	29%	36%	35%	22%	5
	u_{mul2}	26%	36%	35%	35%	42%	41%	6
6 次関数同定問題 (300 試行)	f_{raw}	66.0%	73.7%	77.7%	72.3%	73.0%	72.7%	4
	f_{shr2}	78.0%	84.0%	81.0%	81.3%	82.3%	78.0%	3
	f_{shr50}	87.0%	83.7%	88.3%	85.7%	88.3%	85.0%	4
	u_{mul2}	84.7%	88.7%	86.3%	81.7%	82.0%	78.0%	3
	u_{mul50}	87.7% (0.00087)	93.7% (0.00072)	95.0% (0.00074)	94.0% (0.00087)	94.3% (0.00080)	95.0% (0.00083)	4
4sin 問題 (100 試行)	f_{raw}	0% (0.307)	1% (0.275)	0% (0.286)	0% (0.295)	0% (0.294)	1% (0.294)	3
	f_{shr2}	0%	2%	5%	6%	10%	6%	6
	f_{shr50}	0%	4%	13%	13%	19%	23%	7
	u_{mul2}	0%	0%	2%	1%	2%	5%	7
	u_{mul50}	3% (0.266)	5% (0.214)	18% (0.186)	11% (0.183)	16% (0.189)	15% (0.190)	6
絡み螺旋問題 (100 試行)	f_{raw}	0% (168.4)	0% (173.8)	1% (172.6)	1% (171.4)	0% (171.5)	1% (172.8)	4
	f_{shr2}	17%	32%	42%	33%	48%	36%	6
	f_{shr50}	9%	38%	46%	44%	45%	57%	7
	u_{mul2}	23%	50%	67%	61%	63%	53%	4
	u_{mul50}	34%	79%	86%	92%	84%	66%	5

この分類の下で $\text{reward}(i(c))=10^{-\text{level}(i(c))}$ と設定すれば、式 (2) は次のようになる。

$$f_{\text{shr50}}(i) = \sum_{c:\text{事例}} \frac{10^{-\text{level}(i(c))}}{\#\{j \in P | \text{level}(j(c)) = \text{level}(i(c))\}} \quad (16)$$

また、特に関数同定問題において $\text{penalty}(i(c))=|\text{error}(i, c)|$ と設定すれば、式 (3) は次のようになる。

$$u_{\text{mul50}}(i) = \sum_{c:\text{事例}} |\text{error}(i, c)| \times \#\{j \in P | \text{level}(j(c)) = \text{level}(i(c))\} \quad (17)$$

絡み螺旋問題においては、個体 i が事例 c を正しく分類する時に $\text{penalty}(i(c))=0$ 、誤分類の時に $\text{penalty}(i(c))=|i(c)|+1$ と設定すれば、式 (3) は次のようになる。

$$u_{\text{mul50}}(i) = \sum_{c:i \text{ が誤る事例}} (|i(c)|+1) \times \#\{j \in P | \text{level}(j(c)) = \text{level}(i(c))\} \quad (18)$$

3.3 実験に用いた世代交代モデル

ここでは [Koza 92] の世代交代モデルを用いた。すなわち、次の手順で探索を行った。

①初期集団を生成する。

②各個体の素適合度を計算し、目標に合った個体ができている探索を終了する。また、予め設定した最大世代に達していても探索を終了する。

③適合度を基に次世代個体の親を集団サイズ分だけ選択・複製する。

④前ステップ③で選んだ親の内、一部はペアにして部分木交換交叉を施し、一部は突然変異を施し、そして残りの一部は何の操作も施さずに、出来たものを次世代の個体とする。

⑤前ステップ④で出来た個体の集団を現世代の集団とする。(世代交代)

⑥ステップ②～⑤を繰り返す。

特にステップ①の初期集団生成においては、初期個体の節点数を 3～25 と設定し、節点数 k の個体の割合が (節点数 k の可能な木の個数) $\times 0.21^k / (\sum_{i=3}^{25} (\text{節点数 } i \text{ の可能な木の個数}) \times 0.21^i)$ になる様に、かつ、固定された節点数の下では可能な木が全て等確率でできる様に、初期個体を生成した [Iba 96, Alonso 95]。但し、ランダム定数がある場合は、ランダム定数全体の出現確率と他の終端子 1 個の出現確率を同等とした。また、ステップ③で選択に使う適合度は式 (6)～(13), (16)～(18) のいずれかで、選択にはトーナメント選択を用いた。その際に用いるトーナメントサイズは予備実験を行って決めた (3.4

表 2 従来手法, 提案手法の比較: 太字は適用した問題の中での最良値を表す.

問題	集団 サイズ	最終 世代	トーナメント 手法 サイズ tsize	成功率	最終値 (300 試行の平均)		
					最良 素適合度	多様度	平均個体 サイズ
11-Multiplexer 問題 (300 試行)	500	200	f_{raw} (tsize=5)	13.3%	1895	57.6	305.1
			f_{shr2} (tsize=6)	98.3%	2047	134.1	283.7
			u_{mul2} (tsize=6)	99.7%	2048	134.6	282.9
			u_{shr2} (tsize=6)	0.0%	1405	40.7	262.2
人工蟻問題 (300 試行)	1000	500	f_{raw} (tsize=6)	46.3%	80.40	9.87	340.5
			f_{shr2} (tsize=5)	29.7%	82.27	21.07	299.6
			u_{mul2} (tsize=6)	39.0%	83.46	20.40	297.2
			u_{shr2} (tsize=6)	44.3%	72.11	6.88	287.2
6 次関数同定 問題 (300 試行)	500	200	f_{raw} (tsize=4)	77.7%	0.0029	2.32	277.9
			f_{shr2} (tsize=3)	84.0%	0.0031	5.80	233.9
			f_{shr50} (tsize=4)	88.3%	0.0018	6.86	217.7
			u_{mul2} (tsize=3)	88.7%	0.0024	5.02	231.5
			u_{mul50} (tsize=4)	95.0%	0.0007	7.02	197.4
			u_{shr2} (tsize=6)	74.3%	0.0106	1.90	271.0
4sin 問題 (300 試行)	500	500	f_{raw} (tsize=3)	0.3%	0.280	8.90	216.4
			f_{shr2} (tsize=6)	7.7%	0.329	16.31	183.8
			f_{shr50} (tsize=7)	18.7%	0.273	18.05	168.9
			u_{mul2} (tsize=7)	3.3%	0.476	15.47	188.1
			u_{mul50} (tsize=6)	14.7%	0.185	19.81	173.9
			u_{shr2} (tsize=6)	0.0%	0.845	2.92	242.1
絡み螺旋問題 (300 試行)	500	1500	f_{raw} (tsize=4)	0.3%	173.1	11.60	472.1
			f_{shr2} (tsize=6)	44.7%	190.9	23.78	367.4
			f_{shr50} (tsize=7)	55.7%	190.2	56.52	234.9
			u_{mul2} (tsize=4)	64.3%	191.8	26.61	359.3
			u_{mul50} (tsize=5)	87.7%	193.5	32.26	259.2
			u_{shr2} (tsize=6)	0.0%	138.7	6.07	578.2

節, 表 1). 但し, 多様性喪失を促進する手法として比較のために用意した $u_{\text{shr2}}()$ に対してはトーナメントサイズは 6 と固定した. ステップ④においては, 親の 20% は関数節点を交叉点を選んで部分木交換交叉を行い, 親の 50% は自由に交叉点を選んで部分木交換交叉を行い, 親の 15% は自由に節点を選んでその節点のラベルだけを選び直す 1 点突然変異 (point mutation) [Poli 97, Banzhaf 98] を行い, 残りの 15% はそのまま次世代に残した. 特に交叉の際は子個体の木の高さをチェックし, 高さが 18 以上の木が作られたらそれは親の 1 つで置き換えた.

その他では, 集団サイズは人工蟻問題では 1000 個, 残りの問題では 500 個とした. 探索の最大世代数は 11 マルチプレクサ問題と 6 次関数同定問題では 200 世代, 4sin 問題と人工蟻問題では 500 世代とし, 絡み螺旋問題では 1500 世代とした. また, 疑似乱数の生成には Mersenne Twister [松本 98] を用いた.

3.4 予備実験: トーナメントサイズの設定

トーナメントサイズは適合度設定とともに選択圧を決める重要なパラメータで GP の探索能力に大きな影響を及ぼすので, 従来手法と提案手法を公平に比較するためには各々の適合度設定に合ったトーナメントサイズを問

題毎に設定する必要がある. そこで, 各問題, $u_{\text{shr2}}()$ 以外の各手法に対してトーナメントサイズ 2~7 の GP 探索を 100~300 回ずつ行った. その結果を表 1 に示す. この表は問題と手法のペア毎にトーナメントサイズによって成功率, すなわち繰り返した GP 探索のうち指定された条件を全て満たす個体を得られた割合がどう変わるかを示したもので, この結果を基に以後の実験で用いるトーナメントサイズを表の右端の欄の様に定めた.

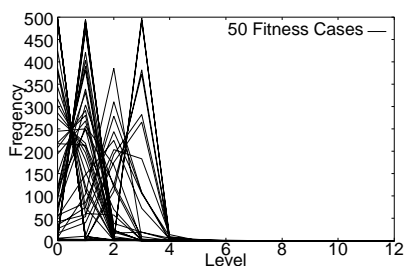
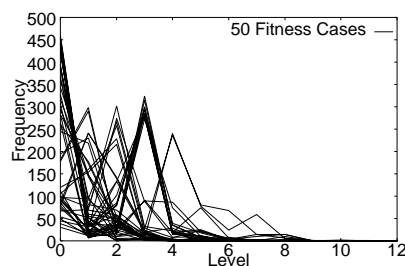
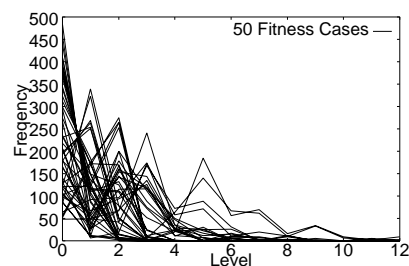
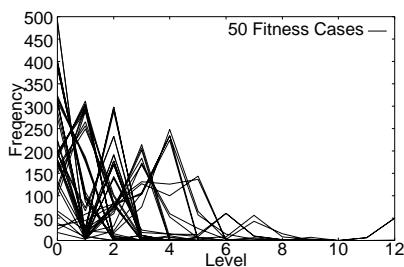
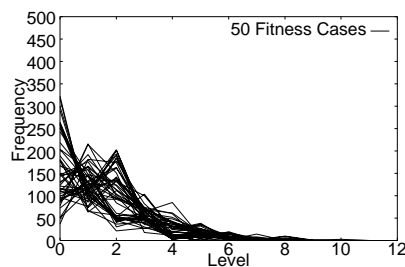
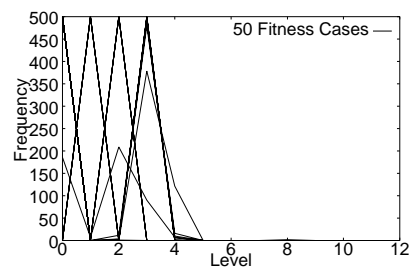
3.5 実験結果

各々の GP 探索は 300 回繰り返した. その際, 式 (11) ~ (13), (16) ~ (18) によって思惑通りの効果が多様性の面で得られるのかどうかを見るために, 個々の GP 探索において定期的に各個体 i, j 間の各事例 c に対する振舞の差

$$d_{ij}(c) = \begin{cases} 1 & \text{if 動作 } i(c), j(c) \text{ の内, 1 つ} \\ & \text{が適正で 1 つが誤り} \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

を調べ, 集団内の全個体間で振舞の差が平均的にどれだけあるかを表す量

$$D = \frac{2}{N(N-1)} \sum_{i,j: \text{異なる個体}} \sum_{c: \text{事例}} d_{ij}(c) \quad (20)$$

図 2 親個体の分布 ($f_{\text{raw}}()$)図 3 親個体の分布 ($f_{\text{shr2}}()$)図 4 親個体の分布 ($f_{\text{shr50}}()$)図 5 親個体の分布 ($u_{\text{mul2}}()$)図 6 親個体の分布 ($u_{\text{mul50}}()$)図 7 親個体の分布 ($u_{\text{shr2}}()$)

を測った．ここで， N は集団サイズを表す．この D をここでは多様度と呼ぶ．

各々の問題，各々の手法に対して GP 探索を繰り返し，最終の成功率，最良の素適合度 (300 試行の平均)，最終世代の多様度 (300 試行の平均)，最終世代の平均個体サイズ (300 試行の平均) を求めた．その結果を表 2 に示す．多様性の面では，どの問題においても，思惑通りの効果が得られている．すなわち，素適合度の代わりに式 (11), (12), (16) ~ (18) に示した $f_{\text{shr2}}()$, $f_{\text{shr50}}()$, $u_{\text{mul2}}()$, $u_{\text{mul50}}()$ を選択に用いることによって (個体の振舞に関する) 多様性が維持され，式 (13) に示した $u_{\text{shr2}}()$ を用いることによって多様性喪失が進むことが，この表から読み取れる．

最良の素適合度に関しては，一部のケース (4sin 問題での $u_{\text{mul2}}()$ ，絡み螺旋問題での $f_{\text{shr50}}()$ の 2 つ) を除いて，素適合度 $f_{\text{raw}}()$ や従来手法 $f_{\text{shr2}}()$ を用いるよりも提案手法である $f_{\text{shr50}}()$ や $u_{\text{mul2}}()$, $u_{\text{mul50}}()$ を用いた方が良い結果になっている．その中でも特に $u_{\text{mul50}}()$ が抜きん出た性能を発揮している．

成功率に関しては，人工蟻問題で多様性維持が全く改善に結び付いていない．実際，多様性維持を進める $f_{\text{shr2}}()$ と $u_{\text{mul2}}()$ がともに多様性喪失を進める $u_{\text{shr2}}()$ よりも悪い結果をもたらしている．この原因については 4.4 節で議論する．人工蟻問題以外では，やはり一部のケース (4sin 問題での $u_{\text{mul2}}()$ のみ) を除いて，素適合度 $f_{\text{raw}}()$ や従来手法 $f_{\text{shr2}}()$ を用いるよりも提案手法である $f_{\text{shr50}}()$ や $u_{\text{mul2}}()$, $u_{\text{mul50}}()$ を用いた方が良い成功率を納めている．表 2 の結果から，提案手法が多くの問題で従来手法を上回る成功率を納めることが期待できる．

最後に，テストした全てのケースで，個体動作を 3 つ以上に分類することによって GP の探索性能が上がっていることにも注目すべきである．表 2 を見る限り，どの場合も， $f_{\text{shr50}}()$ は $f_{\text{shr2}}()$ よりも，そして $u_{\text{mul50}}()$ は $u_{\text{mul2}}()$ よ

りも高い成功率，良好な最良適合度をもたらしている．

4. 考 察

4.1 多様性維持の様子

詳しく多様性維持の様子を見るために，関数同定問題と絡み螺旋問題の各々で 1 つの GP 探索の初期の幾つかの世代において，事例毎に集団内の個体は何個どの達成度レベルに属するか度数分布，および，選択によって選ばれた親個体は何個どの達成度レベルに属するか度数分布を調べた．この内 4sin 問題に関しては，事例が 50 個あるので親個体の度数分布のグラフは各々の世代で 50 本できる．20 世代目におけるこれら 50 本をまとめて 1 つに描いたものを図 2 ~ 図 7 に示す．これらの図 2 ~ 図 7 はそれぞれ $f_{\text{raw}}()$, $f_{\text{shr2}}()$, $f_{\text{shr50}}()$, $u_{\text{mul2}}()$, $u_{\text{mul50}}()$, $u_{\text{shr2}}()$ を選択に用いた場合の分布を表す．式 (8) の素適合度や式 (13) の $u_{\text{shr2}}()$ を選択に用いた場合 (図 2, 図 7) は level=1 ~ 3 の所に高さが 500 (近く) の山ができており，これはこれらの度数分布の事例では (ほとんど) 全ての個体が同程度の達成度レベルで誤ることを物語っている．特に図 7 では，山の高さが 500 未満の度数分布のグラフがほんの数本程度しか見えないので，振舞面での個体の種類はほとんど無くなっていると判断できる．これに対し， $f_{\text{shr2}}()$ や $u_{\text{mul2}}()$ を用いると，どの事例においても 1 以上 (誤り) の単一の達成度レベルに個体が集中することが避けられている．そして， $f_{\text{shr50}}()$ や $u_{\text{mul50}}()$ を用いると，この傾向が強まり，度数分布の多くの折れ線が緩やかな右下がりになる傾向も出てくる．これは，局所解の手前の解候補を集団内に十分に保持している 1 つの証拠と見ることができる．

表 3 Spearman 相関係数 (11 マルチプレクサ)

探索手法	ランキングの組	0 世代目	100 世代目
raw	raw-shr2	0.8880	0.7322
	raw-mul2	-0.0248	0.1397
	shr2-mul2	0.0186	0.3322
shr2	raw-shr2	0.8904	0.6886
	raw-mul2	-0.0530	-0.1858
	shr2-mul2	-0.0132	0.0464
mul2	raw-shr2	0.8902	0.7955
	raw-mul2	-0.0129	-0.3738
	shr2-mul2	0.0289	-0.2981

表 4 Spearman 相関係数 (人工蟻)

探索手法	ランキングの組	0 世代目	100 世代目
raw	raw-shr2	0.9886	0.6510
	raw-mul2	0.4509	0.1690
	shr2-mul2	0.4635	0.4529
shr2	raw-shr2	0.9887	0.7050
	raw-mul2	0.4442	-0.1615
	shr2-mul2	0.4564	-0.0085
mul2	raw-shr2	0.9886	0.6492
	raw-mul2	0.3962	-0.1138
	shr2-mul2	0.4098	0.0635

表 5 Spearman 相関係数 (6 次関数)

探索手法	ランキングの組	0 世代目	100 世代目
raw	raw-shr2	0.2170	-0.0110
	raw-shr50	0.0204	0.1173
	raw-mul2	0.1317	0.7309
	raw-mul50	0.2848	0.7340
	shr2-shr50	0.8247	0.7380
	shr2-mul2	0.4077	0.0684
	shr2-mul50	0.2032	0.0104
	shr50-mul2	0.0824	0.1679
	shr50-mul50	0.0216	0.1220
	mul2-mul50	0.1458	0.6935
shr50	raw-shr2	0.2317	-0.1037
	raw-shr50	0.0376	-0.0601
	raw-mul2	0.1396	0.3470
	raw-mul50	0.3506	0.3307
	shr2-shr50	0.8274	0.8932
	shr2-mul2	0.3809	-0.0826
	shr2-mul50	0.2065	-0.1993
	shr50-mul2	0.0503	-0.0456
	shr50-mul50	0.0247	-0.1625
	mul2-mul50	0.1533	0.1667
mul50	raw-shr2	0.2190	0.0145
	raw-shr50	0.0163	0.0228
	raw-mul2	0.1229	0.3491
	raw-mul50	0.3476	0.4098
	shr2-shr50	0.8248	0.9489
	shr2-mul2	0.3475	0.1143
	shr2-mul50	0.2040	0.0157
	shr50-mul2	0.0082	0.1168
	shr50-mul50	0.0246	0.0327
	mul2-mul50	0.1325	0.2389

表 6 Spearman 相関係数 (4sin)

探索手法	ランキングの組	0 世代目	100 世代目
raw	raw-shr2	0.0691	-0.0630
	raw-shr50	-0.0072	-0.0273
	raw-mul2	0.1752	0.2116
	raw-mul50	0.2880	0.1935
	shr2-shr50	0.9843	0.9101
	shr2-mul2	0.0963	0.0172
	shr2-mul50	0.0713	-0.0435
	shr50-mul2	0.0156	0.0511
	shr50-mul50	-0.0051	-0.0170
	mul2-mul50	0.1557	0.1169
shr50	raw-shr2	0.1082	-0.0289
	raw-shr50	0.0349	-0.0244
	raw-mul2	0.1659	0.1950
	raw-mul50	0.2831	0.1026
	shr2-shr50	0.9843	0.9673
	shr2-mul2	0.1172	-0.0449
	shr2-mul50	0.1128	-0.0135
	shr50-mul2	0.0381	-0.0380
	shr50-mul50	0.0400	-0.0086
	mul2-mul50	0.1598	0.0297
mul50	raw-shr2	0.0884	-0.0198
	raw-shr50	0.0119	-0.0214
	raw-mul2	0.1934	0.1608
	raw-mul50	0.2704	0.1139
	shr2-shr50	0.9844	0.9821
	shr2-mul2	0.1098	-0.0111
	shr2-mul50	0.1003	-0.0322
	shr50-mul2	0.0315	-0.0112
	shr50-mul50	0.0258	-0.0313
	mul2-mul50	0.1542	0.1661

表 7 Spearman 相関係数 (絡み螺旋)

探索手法	ランキングの組	0 世代目	500 世代目
raw	raw-shr2	0.4458	0.4910
	raw-shr50	0.1839	-0.2588
	raw-mul2	-0.0111	-0.1597
	raw-mul50	-0.0456	-0.4445
	shr2-shr50	0.7300	0.3528
	shr2-mul2	0.0370	0.1118
	shr2-mul50	0.0362	-0.2159
	shr50-mul2	0.0153	0.2589
	shr50-mul50	0.0368	0.0683
	mul2-mul50	0.0536	0.2484
shr50	raw-shr2	0.4540	0.9501
	raw-shr50	0.1912	0.5596
	raw-mul2	-0.0212	-0.0879
	raw-mul50	-0.0273	-0.1303
	shr2-shr50	0.7299	0.6720
	shr2-mul2	0.0413	-0.0450
	shr2-mul50	0.0702	-0.0949
	shr50-mul2	0.0183	-0.0122
	shr50-mul50	0.0848	-0.0555
	mul2-mul50	0.0723	0.6216
mul50	raw-shr2	0.4535	0.9588
	raw-shr50	0.1894	0.2614
	raw-mul2	-0.0359	-0.1367
	raw-mul50	-0.0464	-0.0645
	shr2-shr50	0.7312	0.3642
	shr2-mul2	0.0150	-0.0810
	shr2-mul50	0.0087	-0.0255
	shr50-mul2	0.0101	0.0216
	shr50-mul50	-0.0018	0.0289
	mul2-mul50	0.0591	0.3082

4.2 素適合度, 分配適合度, 積算不適合度の間の相関

トーナメント選択においては個体のランキングのみを利用して選択が行われるので, 適合度を設定する手法間の相関/違いを明らかにするためには, それらの手法のモタラすランキングを比較すればよい.

そこで, まず最初に GP 探索の際に定期的に各種ランキング間の Spearman 相関係数を測定した. 2 つのランキング A, B の間の Spearman 相関係数は一般に次の様に計算される.

$$1 - \frac{6 \sum_{i \in P} (\text{rank}_A(i) - \text{rank}_B(i))^2}{N^3 - N}$$

ここで, P はランキング対象物の集合, N はランキング対象物の個数を表し, $\text{rank}_A(i)$ はランキング A における対象物 (今の場合は個体) i の順位を表す. Spearman

相関係数は 2 つのランキング間の類似度を -1 から 1 の間の値で表す指標で, 同じランキング間では 1 , 逆のランキング間では -1 という値になる. 各々の問題における Spearman 相関係数の測定結果 (の一部) を表 3 ~ 表 7 に示す. 様々な集団に対するランキングの違いを調べるために, 問題毎に素適合度を用いた GP 探索, $f_{\text{shr}*}()$ 系統の適合度を用いた GP 探索, $u_{\text{mul}*}()$ 系統の適合度を用いた GP 探索を各々 100 ~ 300 回繰り返し, 世代毎に Spearman 相関係数の平均値を計算しこれらの表に載せている. 表の中では 0.5 以上の相関係数値は太字で表している. これらの表から, $f_{\text{shr2}}()$ によるランキングと $f_{\text{shr50}}()$ によるランキングの間の非常に強い正の相関が問題や GP 探索の種類によらずに認められる. 11 マル

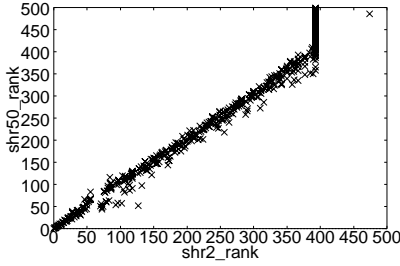


図 8 点 $(\text{rank}_{\text{shr}2}(i), \text{rank}_{\text{shr}50}(i))$ の集合 (4sin)

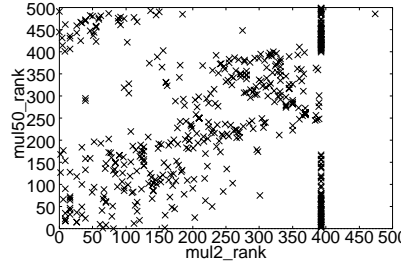


図 10 点 $(\text{rank}_{\text{mul}2}(i), \text{rank}_{\text{mul}50}(i))$ の集合 (4sin)

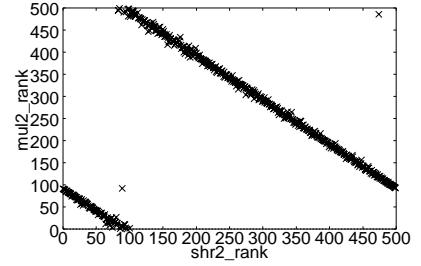


図 12 点 $(\text{rank}_{\text{shr}2}(i), \text{rank}_{\text{mul}2}(i))$ の集合 (11 マルチプレクサ)

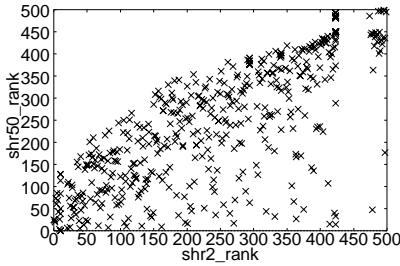


図 9 点 $(\text{rank}_{\text{shr}2}(i), \text{rank}_{\text{shr}50}(i))$ の集合 (絡み螺旋)

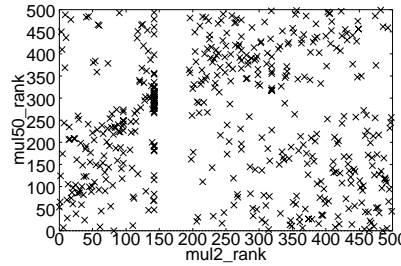


図 11 点 $(\text{rank}_{\text{mul}2}(i), \text{rank}_{\text{mul}50}(i))$ の集合 (絡み螺旋)

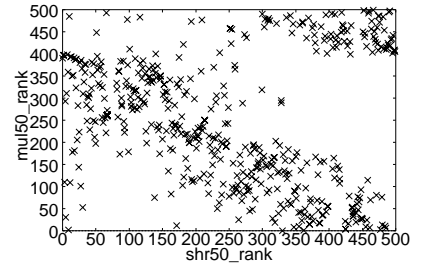


図 13 点 $(\text{rank}_{\text{shr}50}(i), \text{rank}_{\text{mul}50}(i))$ の集合 (4sin)

チプレクサ問題, 人工蟻問題においては, $f_{\text{raw}}()$ によるランキングと $f_{\text{shr}2}()$ によるランキングの間の非常に強い正の相関が GP 探索の種類によらずに認められる. これら以外には, GP 探索の種類によらない目立った相関は表 3 ~ 表 7 の数値からはあまり認められない. 例えば, $u_{\text{mul}2}()$ によるランキングと $u_{\text{mul}50}()$ によるランキング間ではほとんどの場合弱い正の相関しか認められない. また, $f_{\text{shr}2}()$ と $u_{\text{mul}2}()$ のランキング間も, $f_{\text{shr}50}()$ と $u_{\text{mul}50}()$ のランキング間も Spearman 相関係数の値は一部の例外を除いて $-0.2 \sim 0.2$ の間を変動しているだけである.

Spearman 相関係数はランキング間の類似度を 1 つの数値で表したものであるが, ランキング間に特徴的な関係がある場合も 0 に近い値になることがある. そこで次に, 今考えている適合度設定による全てのランキングのペア $\langle A, B \rangle$ に対して, 実際の GP 探索の際に定期的に点集合 $\{(\text{rank}_A(i), \text{rank}_B(i)) | i \text{ は現集団内の個体}\}$ を平面上にプロットする作業を行った. それによって得られた図の一部を図 8 ~ 図 13 に示す. これらはいずれも初期集団に関して描かれた図である. 図 8 と図 9 は $f_{\text{shr}2}()$ と $f_{\text{shr}50}()$ のランキングペアに対して描かれたもので, 各々 4sin 問題と絡み螺旋問題の GP 探索で得られたものである. これらの図は $f_{\text{shr}2}()$ と $f_{\text{shr}50}()$ のランキング間の典型的な関係を表しており, ほとんどの場合これらのランキング間で Spearman 相関係数の値が 0.5 以上になる状況をよく表している. 次の図 10 と図 11 は $u_{\text{mul}2}()$ と $u_{\text{mul}50}()$ のランキングペアに対して描かれたもので, 各々 4sin 問題と絡み螺旋問題の GP 探索で得られたものである. $u_{\text{mul}2}()$ と $u_{\text{mul}50}()$ のランキングの間に弱い正の Spearman 相関しか認められないのもこれらの図から納得できる. 最後に, 図 12, 図 13 は各々 $f_{\text{shr}2}()$ と $u_{\text{mul}2}()$

のランキングペア (11 マルチプレクサ問題), $f_{\text{shr}50}()$ と $u_{\text{mul}50}()$ のランキングペア (4sin 問題) に対して描かれたもので, $f_{\text{shr}*}()$ の系統と $u_{\text{mul}*}()$ の系統の典型的な関係をよく表している. これらの図では 2 本程度の右下がりの直線に沿う形で点が散らばっている. この様に, $f_{\text{shr}*}()$ の系統と $u_{\text{mul}*}()$ の系統は全く異なるランキング結果をもたらす. 比較的特徴のある関係ではあるが, Spearman 相関係数は 0 に近い値にしかならない. 実際, 図 12 の場合の Spearman 相関係数は $1 - \frac{6}{500^3} (\sum_{i=0}^{99} (99-2i)^2 + \sum_{i=100}^{499} (599-2i)^2) \approx -0.04$ と見積もることができる.

次に, $f_{\text{shr}*}()$ の系統と $u_{\text{mul}*}()$ の系統の間のどういう違いが図 12 や図 13 の結果をもたらしたのかを考える. 例えば関数同定問題の場合, 式 (14) から $\text{level}(i(c)) \approx \log_2 |\text{error}(i, c)| - \log_2 \epsilon$ であるので, 式 (16) を

$$f_{\text{shr}50}(i) \approx \sum_{c: \text{事例}} \frac{|\text{error}(i, c)|^{-\log_2 10}}{\#\{j \in P | \text{level}(j(c)) = \text{level}(i(c))\}} \quad (21)$$

と近似的に見ることができる. それゆえ, この式 (21) と $u_{\text{mul}50}()$ の式 (17) を見比べ 1 つの事例 c についてだけを見れば, c に関する分配適合度 (shared fitness; 大きい方が良い) に基づいた個体のランキングと, c に関する積算不適合度 (multiplied unfitnes; 小さい方が良い) に基づいた個体のランキングはほぼ一致したものとなる. しかし, 全ての事例について総和をとり全体を眺めてみると, $f_{\text{shr}50}(i)$ においては小さな誤差 $|\text{error}(i, c)|$ をもたらす事例の項が他の項より適合度全体に対して大きな影響を及ぼし, 一方の $u_{\text{mul}50}(i)$ においては大きな誤差をもたらす事例の項が他の項より不適合度全体に対して大きな影響を及ぼす. 従って, 幾つかの事例で他に類を見ない目覚しい働きをするが多くの事例で異常な振舞をする個

体は $f_{shr50}()$ を選択に用いた場合は増殖し, $u_{mul50}(i)$ を選択に用いた場合は死滅する. この様な個体は図 13 では左上の点に相当する. 一方, 他に類を見ない独創的な動きをすることもなく異常な振舞もしない個体は, $f_{shr50}()$ では死滅し, $u_{mul50}(i)$ では生き長らえる. この様な個体は図 13 では右下の点に相当する.

積算不適合度 $u_{mul50}()$ を用いて選択すると, 特殊な幾つかの事例にのみうまく対応し全体としては異常な振舞が多い個体を排除することになるので, 関数同定問題では非常に良好な最良 (素) 適合度が得られるものと考えられる. 一般に, $u_{mul*}()$ の系統を用いると, 個々の事例において適正な動作を得るのがそれほど困難でない問題においては無駄な探索が抑制され全体としてバランスの良い個体ができ易くなると考えられる.

4.3 個体動作分類の細かさ等の影響

個体動作が無限もしくは多数種類可能な問題に関して, 3.2 節では適合度シェアリング, 不適合度積算の 1 つの実装方法を示したが, ここではこれに代る実装の可能性を探る.

まず, 3.2 節で関数同定問題, 絡み螺旋問題の個体の動作を 50 種類に分類した理由について説明する. これらの問題では, 個体の動作結果は非負実数値 (例えば関数同定の場合は $|\text{error}(i, c)|/\epsilon$ で評価される. 保護付き除算 pdiv 等の影響でこれらの非負実数値は非常に大きくなることもある. 従って, 全ての非負実数値を有限種類に分類する必要がある. その際, 問題やその時点の個体分布によらない分類が望ましい. また, 分類に要する計算量もできるだけ抑えなければならない. これらの点を考慮に入れて, 個体の動作結果の評価を表す非負実数値のコンピュータ内部のデータ形式 (例えば IEEE-754 規格) を考え, その指数部を利用して分類することが妥当と判断した. ただ, 指数部が 8 ビット, 指数部の表せる非負整数が 0 ~ 128 の 129 種類, 指数部が負の時が正常動作を表すと考えて非負実数全体を 130 種類に分類しても, 指数部の値が 50 ~ 128 となる個体はほとんど出現しないと思われる. 一方では, 指数部が負, 0, 1, 2, ..., $k-3$, $k-2$ 以上 という風に, 非負整数を k 個のグループに分類するとした場合, k の値が小さいと「指数部が $k-2$ 以上」のグループに属する個体が多くなり, 悪個体を過度に冷遇することにもなる. そこで「指数部が $k-2$ 以上」のグループがほぼ空になる様に十分大きな分類数として $k=50$ としたのが, 式 (14) のレベル値の定義である. 実際には, 図 2 ~ 図 7 等から判断してレベル値が 15 以上の個体はほとんど出現しない様なので, 個体の分類を定めている式 (14), (15) を 17 ~ 49 分類に手直しして実験を行っても 3.5 節と同等の結果が得られるものと予想される.

次に, 適合度シェアリング, 不適合度積算の実装に関して, 他の可能性を 3 つ示す.

(他の実装例 1) 3.2 節では 2 種類に分類することの対極として (シェアリングが機能する範囲内で) できるだけ細かく分類することを考えた. しかし, 分類の細かさ / 粗さは探索性能を左右する重要なパラメータと考えられるので, 式 (14), (15) の誤りのレベルを数個 (β 個) ずつ束ねて, レベル $1 \sim \beta$ の動作, レベル $\beta+1 \sim 2\beta$ の動作, レベル $2\beta+1 \sim 3\beta$ の動作, ... を各々同じと見なして, 適合度シェアリングや不適合度積算の手法を適用することも十分に考えられる. すなわち,

$$l(i, c, \beta) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if level}(i(c)) = 0 \\ \lfloor (\text{level}(i(c)) - 1) / \beta \rfloor \times \beta + 1 & \\ \text{otherwise} & \end{cases} \quad (22)$$

として, 式 (16), (17), (18) の代わりに各々次の式を用いるという考え方もある. ($\beta=1$ の時が元の式 (16) ~ (18) に相当する.)

$$f_{shr50/\beta}(i) = \sum_{c:\text{事例}} \frac{10^{-l(i, c, \beta)}}{\#\{j \in P | l(j, c, \beta) = l(i, c, \beta)\}} \quad (23)$$

$$u_{mul50/\beta}(i) = \sum_{c:\text{事例}} |\text{error}(i, c)| \times \#\{j \in P | l(j, c, \beta) = l(i, c, \beta)\} \quad (24)$$

$$u_{mul50/\beta}(i) = \sum_{c:i \text{ が誤る事例}} (|i(c)| + 1) \times \#\{j \in P | l(j, c, \beta) = l(i, c, \beta)\} \quad (25)$$

(他の実装例 2) 適合度シェアリングの式 (16) を導出する際, 式 (2) の $\text{reward}(i(c))$ を $10^{-\text{level}(i(c))}$ と設定したが, 他に $\text{reward}(i(c)) = 2^{-\text{level}(i(c))}$ や $\text{reward}(i(c)) = 5^{-\text{level}(i(c))}$ という設定も十分考えられる. 従って, 式 (23) は更に次の様に一般化できる.

$$f_{shr50(\gamma)/\beta}(i) = \sum_{c:\text{事例}} \frac{\gamma^{-l(i, c, \beta)}}{\#\{j \in P | l(j, c, \beta) = l(i, c, \beta)\}} \quad (26)$$

(他の実装例 3) 絡み螺旋問題に関する式 (18) を導出する際, 式 (3) の $\text{penalty}(i(c))$ を $|i(c)| + 1$ (誤分類時) と設定したが, 他に $\text{penalty}(i(c)) = |i(c)|$ (誤分類時) や $\text{penalty}(i(c)) = |i(c)| + 2$ (誤分類時) という設定も十分考えられる. 従って, 式 (25) は更に次の様に一般化できる.

$$u_{mul50(\alpha)/\beta}(i) = \sum_{c:i \text{ が誤る事例}} (|i(c)| + \alpha) \times \#\{j \in P | l(j, c, \beta) = l(i, c, \beta)\} \quad (27)$$

表 8 各種パラメータによる成功率の変動：太字はその行の中での最良値を表す．
下段括弧内は最終世代における個体サイズの平均を表す．

問題	トーナメント 手法 サイズ tsize	分割の細かさを表すパラメータ β				
		$\beta = 1$	$\beta = 2$	$\beta = 4$	$\beta = 6$	$\beta = 8$
6 次関数同定 問題 (300 試行)	$f_{shr50(2)}/\beta()$ (tsize=4)	78.3%	82.3%	76.7%	68.3%	76%
	$f_{shr50(5)}/\beta()$ (tsize=4)	90%	84.3%	84%	78.7%	80.7%
	$f_{shr50(10)}/\beta()$ (tsize=4)	88.3%	87%	83.7%	82%	82%
	$u_{mul50(0)}/\beta()$ (tsize=4)	95%	89.3%	81.3%	79%	78.7%
	$u_{mul50(0.01)}/\beta()$ (tsize=4)	88.3%	86.3%	88.7%	79.3%	81.7%
	$u_{mul50(0.1)}/\beta()$ (tsize=4)	57.7%	71%	85%	73.7%	79.7%
4sin 問題 (300 試行)	$f_{shr50(2)}/\beta()$ (tsize=7)	3.7%	12.3%	14.0%	9.7%	12.0%
	$f_{shr50(5)}/\beta()$ (tsize=7)	26.7%	16.7%	8.0%	11.0%	10.0%
	$f_{shr50(10)}/\beta()$ (tsize=7)	18.7%	8.3%	9.3%	12.7%	8.0%
	$u_{mul50(0)}/\beta()$ (tsize=6)	14.7%	9.3%	4.0%	6.0%	2.3%
	$u_{mul50(0.01)}/\beta()$ (tsize=6)	12.3%	10.7%	5.3%	7.7%	3.0%
	$u_{mul50(0.1)}/\beta()$ (tsize=6)	11.7%	8.7%	4.3%	5.3%	5.7%
絡み螺旋問題 (100 試行)	$f_{shr50(2)}/\beta()$ (tsize=7)	0% (7.4)	0% (11.1)	0% (15.9)	0% (14.7)	0% (12.5)
	$f_{shr50(5)}/\beta()$ (tsize=7)	13%	18%	22%	22%	33%
	$f_{shr50(10)}/\beta()$ (tsize=7)	57%	60%	45%	55%	51%
	$u_{mul50(0)}/\beta()$ (tsize=5)	0.0% (470.8)	0.0% (462.1)	0.0% (510.0)	1.0% (478.5)	0.0% (442.8)
	$u_{mul50(0.5)}/\beta()$ (tsize=5)	77%	62%	72%	65%	65%
	$u_{mul50(1)}/\beta()$ (tsize=5)	92%	85%	80%	70%	69%
	$u_{mul50(2)}/\beta()$ (tsize=5)	86%	84%	75%	61%	64%
	$u_{mul50(4)}/\beta()$ (tsize=5)	87%	80%	69%	58%	61%

関数同定問題に関しても同様のパラメータを導入することが可能で、式 (24) は更に次の様に一般化できる．

$$\begin{aligned}
 & u_{mul50(\alpha)}/\beta(i) \\
 &= \sum_{c: \text{事例}} (|\text{error}(i, c)| + \alpha) \times \#\{j \in P | \\
 & \quad l(j, c, \beta) = l(i, c, \beta)\} \quad (28)
 \end{aligned}$$

以上の (他の実装例 1) ~ (他の実装例 3) で導入したパラメータ α, β, γ の探索能力への影響を調べるために、6 次関数同定, 4sin の問題については、 $\gamma \in \{2, 5, 10\}, \beta \in \{1, 2, 4, 6, 8\}$ として式 (26) の $f_{shr50(\gamma)}/\beta()$ を選択に用いた GP 探索と、 $\alpha \in \{0, 0.01, 0.1\}, \beta \in \{1, 2, 4, 6, 8\}$ として式 (28) の $u_{mul50(\alpha)}/\beta()$ を選択に用いた GP 探索を、全ての α, β, γ の組合せに対して 300 回ずつ繰り返した．また、絡み螺旋問題については、やはり $\gamma \in \{2, 5, 10\}, \beta \in \{1, 2, 4, 6, 8\}$ として式 (26) の $f_{shr50(\gamma)}/\beta()$ を選択に用いた GP 探索と、 $\alpha \in \{0, 0.5, 1, 2, 4\}, \beta \in \{1, 2, 4, 6, 8\}$ として式 (27) の $u_{mul50(\alpha)}/\beta()$ を選択に用いた GP 探索を、全ての α, β, γ の組合せに対して 100 回ずつ繰り返した．これらの結果 (成功率) を表 8 に示す．この表によれば、一部の場合を除いて粗く分類するより細かく分類して適合度シェアリングや不適合度積算の手法を適用する方が高い成功率に繋がっている．適合度シェアリング $f_{shr50(\gamma)}/\beta()$ におけるパラメータ γ に関しては、6 次関数同定問題では $\gamma = 5, 10$ が良好、4sin 問題では $\gamma = 5$ が良好、絡み螺旋問題では $\gamma = 10$ が良好、 $\gamma = 2$ が劣悪な結果になっている．特に絡み螺旋問題で $\gamma = 2$ の場合は、初期世代以

降個体が徐々に小さくなる．100 世代目以降で平均個体サイズが 7 ~ 16 程度という状態が続き、適合度の改善が全然起こらない．(最終の多様度は 90 ~ 93 と異常に大きく、最終の最良適合度は 102 ~ 106 程度のままである．)
 γ 値としては 5 ~ 10 が無難な様である．最後に、積算不適合度 $u_{mul50(\alpha)}/\beta()$ のパラメータ α に関しては、6 次関数同定, 4sin の問題では $\alpha = 0$ が最良で α の増加と共に徐々に劣悪な結果となり、絡み螺旋問題では $\alpha = 1$ が良好、 $\alpha = 0$ が劣悪な結果になっている．特に絡み螺旋問題で $\alpha = 0$ の場合は極端なプロート現象が起きており、適合度の改善は僅かである．(最終の最良適合度が 129 ~ 136 程度になっている．) これは、 $\alpha = 0$ の場合は、個体の表す式の値が小さくなるだけで $u_{mul50(\alpha)}/\beta()$ の値を小さくできるため、単に表す式の値が小さくなるだけの改造が個体に繰り返し加えられるためだと考えられる．

4.4 多様性維持が成功率向上をもたらさない問題

3.5 節/表 2 の実験結果によると、人工蟻問題においては、 $f_{shr2}()$ や $u_{mul2}()$ を選択に用いると個体の多様性や最良素適合度は $f_{raw}()$ を用いる場合よりも良くなるが、成功率は $f_{raw}()$ や多様性喪失を進める $u_{shr2}()$ を用いる場合よりも低い結果になっている．この原因としてまず第一に考えられるのは、適合度ランドスケープの複雑な形状である．文献 [Langdon 98] によれば、人工蟻問題においては探索空間内の候補解のほとんどが低い素適合度

しかもたないので、山登り探索では局所解に陥る可能性が高い。また解を構成するスキーマを考えても、スキーマに属する候補解の素適合度の変動幅は大きく、それらの標準偏差は平均値と同程度以上の大きさをもつことが多いので、有効なスキーマが積み重なって解が構成されるという解生成過程は受け入れにくい。その一方で、探索空間全体の中で解の割合は $2 \times 10^{-6} \sim 10^{-5}$ と予想外に大きく、そのためランダム探索でも時間をかければ解を見つけ出せる状況にある。それゆえ、 $f_{shr2}()$ や $u_{mul2}()$ を使って個体の多様性を維持することの利点がほとんどない。多様性維持のために色々な候補解を保持することは、かえって探索効率を落とすことにもなりかねない。

以上の人工蟻問題の状況から判断して、一般に、山登りや交叉等の技法で解が見つけないが探索空間の中に解が多く散らばっている問題では、多様性維持は成功率向上にあまり役立たないと考えられる。

5. ま と め

本論文では、GP 探索において個体の振舞上の多様性を維持するための手法として、特殊な状況にのみうまく対応し全体としては異常な振舞が多い個体を排除する不適合度積算の考え方を提案し、この手法と [McKay 00] の提案した陰的適合度シェアリング手法の効果を 11 マルチプレクサ問題、関数同定問題 ($x^6 - 2x^4 + x^2, \sin x + \sin 2x + \sin 3x + \sin 4x$)、絡み螺旋問題、人工蟻問題でテストした。また、関数同定問題、絡み螺旋問題においては、個体を振舞の良さの程度に応じて 50 種類に分類して陰的適合度シェアリングや不適合度積算を行う方法もテストした。その結果、多くの問題で不適合度積算の手法が McKay の陰的適合度シェアリングを上回る成功率をもたらすことが観察された。陰的適合度シェアリングや不適合度積算を用いる際に多種類に個体動作を分類することの有用性が確認された。また、不適合度積算が陰的適合度シェアリングとは全く異なる個体ランキングをもたらすことが明らかになった。

ここで提案した手法はもちろん絶対的なものではない [Wolpert 97]。また、4・3 節で見た様に不適合度積算に関しては実際にはパラメータ調節が難しい場合もある。それでも、提案手法は多くの問題で良好な探索性能を発揮するものと思われる。

謝 辞

3・4 節、4・2 節、4・3 節は査読者の方々からの照会をきっかけに構成しました。有益な意見を出して下さった査読者の方々に深く感謝します。

◇ 参 考 文 献 ◇

[Alonso 95] Alonso, L. and Schott, R.: *Random Generation of Trees*, Kluwer Academic Publishers (1995)

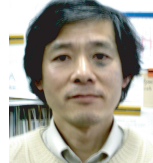
- [Banzhaf 98] Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D.: *Genetic Programming - An Introduction*, Morgan Kaufmann Publishers, Inc. (1998), (邦訳: 遺伝的プログラミング, 伊庭斉志, 新田徹 訳, 科学技術出版, (2000))
- [Burke 02] Burke, E., Gustafson, S., and Kendall, G.: A Survey and Analysis of Diversity Measures in Genetic Programming, in *Proc. of the Genetic and Evolutionary Computation Conference*, pp. 716-723 (2002)
- [Burke 04] Burke, E., Gustafson, S., and Kendall, G.: Diversity in Genetic Programming: An Analysis of Measures and Correlation with Fitness, *IEEE Trans. on Evolutionary Computation*, Vol. 8, No. 1, pp. 47-62 (2004)
- [Goldberg 87] Goldberg, D. E. and Richardson, J.: Genetic Algorithms with Sharing for Multimodal Function Optimization, in *Genetic Algorithms and their Applications: Proc. of the Second International Conference on Genetic Algorithms*, pp. 41-49 (1987)
- [Iba 96] Iba, H.: Random Tree Generation for Genetic Programming, in *Proc. of the Sixth International Conference on Parallel Problem Solving from Nature*, pp. 144-153 (1996)
- [伊庭 05] 伊庭 斉志: 知の科学 進化論的計算手法, オーム社 (2005)
- [Koza 92] Koza, J. R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press (1992)
- [Koza 94] Koza, J. R.: *Genetic Programming II: Automatic Discovery of Reusable Programs*, The MIT Press (1994)
- [Langdon 98] Langdon, W. B. and Poli, R.: Why Ants are Hard, in *Genetic Programming 1998: Proc. of the Third Annual Conference*, pp. 193-201 (1998)
- [松本 98] 松本 眞: コイン投げで一儲けする方法 — 疑似乱数研究の現状 —, 情報処理, Vol. 39, No. 11, pp. 1166-1170 (1998)
- [McKay 00] McKay, R. I. B.: Fitness Sharing in Genetic Programming, in *Proc. of the Genetic and Evolutionary Computation Conference*, pp. 435-442 (2000)
- [Poli 97] Poli, R. and Langdon, W. B.: A New Schemata Theory for Genetic Programming with One-point Crossover and Point Mutation, in *Genetic Programming 1997: Proc. of the Second Annual Conference*, pp. 278-285 (1997)
- [Wolpert 97] Wolpert, D. H. and Macready, W. G.: No Free Lunch Theorems for Optimization, *IEEE Trans. on Evolutionary Computation*, Vol. 1, pp. 67-82 (1997)

〔担当委員: 小野田 崇〕

2005 年 6 月 27 日 受理

著 者 紹 介

元木 達也 (正会員)



1977 年電気通信大学電気通信学部電子計算機学科卒業。1979 年同大学大学院電気通信学研究科電子計算機学専攻修士課程修了。同年茨城大学工学部助手。1987 年 8 月新潟大学教養部講師。1992 年 2 月同部助教授。1994 年より同大学工学部に所属。現在に至る。進化的計算の研究に従事。

沼口 靖



2003 年新潟大学工学部情報工学科卒業。2005 年同大学大学院自然科学研究科情報・計算機工学専攻修士課程修了。現在、日立化成工業株式会社に勤務。