

GPにおける頻出構造に着目した複合的ブloat抑制戦略

小野 景子^{1,a)} 花田 良子^{2,b)}

概要: 遺伝的プログラミング (GP) を実問題に適用する際、多くの場合、解構造が急激に大きくなるブloatの抑制を考える必要がある。我々は本論文で、複合的ブloat抑制戦略に関して論じる。ブloat抑制のための効果的な手法に“Size-fair モデル GP”があるが、この手法は交叉対象となる木の大きさのみに着目するため、重要な木構造が壊れる可能性がある。そこで、我々は、解の探索に有効に働く部分木の発見を促進するために、優良解群に頻出する比較的小さな部分木に着目する。また、それらを保存する手法を Size-fair モデル GP 手法に組み込むことで、効率よい解探索を同時に実現するブloat抑制手法を提案する。提案法の性能評価には、広く使われているベンチマーク問題および画像の雑音除去フィルタの最適化問題を用い、ブloat抑制性能および解探索性能の観点より提案法の有効性を検証する。

Assembling Bloat Control Strategies in Genetic Programming Based on Frequent Tree Structures

KEIKO ONO^{1,a)} YOSHIKO HANADA^{2,b)}

Abstract: In Genetic programming (GP), it is important to control program size for avoiding consume of computational power. We address the problem of controlling bloat in this paper. One of the most effective bloat control models is the *Size-fair Model GP*. In the *Size-fair Model GP*, the size of a new individual is limited to as almost same as that of parents. This method can limit bloat significantly; however, some researchers mentioned that introns occur as a defense mechanism against destructive genetic operations. In order to enhance the size-fair model GP, we propose a novel bloat control method based on tree size and frequent trees, where the frequent trees mean the relatively-small sub-trees appearing frequently among populations. Using some well-known benchmark problems and noise reduction for images, we demonstrate that the proposed method can achieve bloat control, and performance improvement can be achieved through bloat control based on tree size and frequent trees.

1. はじめに

遺伝的プログラミング (GP) において、プログラムサイズを制御することは計算量の観点から重要な課題である。GP では、進化の過程で急激にプログラムサイズが大きくなる現象を“ブloat”と呼び、評価値に影響を与えない、つまり評価値に中立である部分構造を“イントロン”と呼

ぶ。GP の解構造は木で表現できるため、最もナイーブなブloatの抑制法は木のノードサイズや深さを制限することである。この方法は単純であるため様々な実問題に適用しやすいが、任意の問題に対する適切な木のノードサイズや木の深さを求めることは難しい。他のアプローチとしては、交叉法によりブloatを抑制する手法が考えられる。Langdon ら [1], Ryan ら [2] などは深さやノードサイズに基づく交叉法やローカルサーチに基づく方法などを提案している。このなかで最も効果的なブloat抑制法として Size-fair モデル GP が挙げられる [3]。Size-fair モデル GP は子個体のノードサイズに制限を設け、親個体とほぼ同じ

¹ 龍谷大学 理工学部 電子情報学科
Department of Electronics and Informatics, Ryukoku University

² 関西大学 システム理工学部
Faculty of Engineering Science, Kansai University

^{a)} kono@rins.ryukoku.ac.jp

^{b)} hanada@kansai-u.ac.jp

大きさの個体が生成されるようにノードサイズを制御し、大きな性能低下なしにブロー抑制が可能な手法である。

一方、進化計算におけるビルディングブロック仮説 [4], [5] では、解はビルディングブロックと呼ばれる部分解が組み合わさって形成されると考えられている。我々は進化終盤に出現する母集団内に頻出する部分木構造、つまり“頻出部分木”は部分解に関係すると考えている。また、イントロンは遺伝的操作で起きる部分解の破壊を抑制する役割があるという報告があり [6]、更なる性能向上のために、Size-fair モデル GP においてノードサイズを制限するだけでなく、頻出木の保存を組み込むことは、性能向上とブロー抑制の両観点から有効であるといえる。本論文では、“Size-fair レベル”と“頻出レベル”という評価基準を提案する。また、これらの提案評価基準に基づく複合的ブロー抑制戦略を提案する。一般的に、母集団におけるラベル付き順序木集合からの頻出木の抽出は多くの計算量がかかるタスクであるため、頻出部分木を列挙しそれらの保存を促進することは計算量の観点から実用的でなかった。Asai らは“FREQT”と呼ばれる頻出部分木抽出法を提案し [7]、最右拡張と枝切り技術を用いて効率よく頻出木を抽出することを可能にした。我々は実験において、FREQT を用いて頻出部分木を抽出および頻出レベルの評価することで、実運用に耐える手法の提案を行う。

2. 問題の定式化および Size-fair モデル GP

遺伝的プログラミング (GP) の枠組みにおいて、関数 $f(x)$ の最小化問題を考える。ここで、 x はラベル付き順序木を表す。各ノードにおけるノード以下の部分木サイズが与えられた場合、最も効果的なブロー抑制方法は Size-fair 交叉を適用することである。従って、我々はこの手法に注目し、この手法の改良を目指す。また、本論文ではこの手法を“Size-fair モデル GP”と呼ぶ。まず、Size-fair モデル GP の定式化を行う。整数 $i (i = 1, \dots, I)$ に対して、 $p_i(g)$ は g 世代 ($g = 1, \dots, G$) の i 番目の個体を表し、 $P(g)$ は g 世代の個体集合を表す。ここで、 $p_i(g) \in P(g)$ である。 $P(0)$ は初期世代を表し、 $p_i(g)$ のノード集合を

$$N(p_i(g)) = \{n(p_i^m); m = 1, \dots, M\},$$

とする。Size-fair モデル GP の交叉は次の通りである。

- S0:** Set $k \leftarrow 1$.
- S1:** Set $P_k(g) \leftarrow P(g)$.
- S2:** If $k \leq I$, then perform Steps S2 to S13, otherwise stop.
- S3:** Set a mother tree $p_k \leftarrow p_i \in P_k(g)$ at random.
- S4:** Set $P_k(g) \setminus p_k$.
- S5:** Set a father tree $p_{k+1} \leftarrow p_i \in P_k(g)$ at random.
- S6:** Choose a crossover node $n(p_k^m)$ at random from $N(p_k(g))$.
- S7:** Extract the maximum subtree s_{mum} whose root node is $n(p_k^m)$.
- S8:** Calculate a size of s_{mum} .

S9: Select the almost same size subtree s_{dad} as s_{mum} in the father tree p_{k+1} *1.

S10: Set a crossover node $n(p_{k+1}^m)$ whose root node is s_{dad} .

S11: Generate a new individual $p_k(g+1)$ by crossover between p_k and p_{k+1} , where crossover points are $n(p_k^m)$ and $n(p_{k+1}^m)$.

S12: Set $k \leftarrow k + 1$.

S13: Return to Step S1.

ここで、ほぼ同じサイズの部分木 s_{dad} と s_{mum} は

$$|s_{mum}| - \epsilon \leq |s_{dad}| \leq |s_{mum}| + \epsilon,$$

を満たす。ここで、 $||$ はノード数を表す。 ϵ は整数値のパラメータであり、実験では、 $\epsilon = 2$ とした。

3. 提案複合ブロー抑制戦略

目的関数 $f(x)$ の最小化問題において、Size-fair モデル GP の性能向上を考える。Size-fair モデル GP を用いた場合、急激なブローを防ぐことはできるが、一方で、イントロンは破壊的な遺伝操作から解構造を守る役割があるとも考えられている [8]。そのため、Size-fair モデル GP において解の探索に有効な解構造を保護するメカニズムを組み込むことも必要であると考えられる。ビルディングブロック仮説において、解は部分解の集まりで形成されると考えられている。この考えに基づき、部分木のデータベースを作成し部分解を特定する手法や [9] 確率モデルを用いた様々な手法が提案されている [10], [11]。これらのモデルは部分木のサイズなどを指定する必要がある、部分解を抽出するために最適な設定を求めることは難しい。

我々は部分解の種としてよい部分解片 (以後、部分解ピースと呼ぶ) が存在し、これらを上手く組み合わせることで部分解を形成できると考えている。これまでに我々はこの仮説に基づき鳥モデル GP の移住戦略を提案し [12]、広く用いられるベンチマーク問題を用いて、その有効性を示した。また、エリート選択により部分解を成さない個体は除かれるため、評価値が低いエリート個体によく現れる比較的小さい部分木である“頻出部分木”は良い部分解ピースの候補であり、これらを上手く組み合わせることで部分解を直接求めるよりも容易に部分解を発見できると考えている。この考えに基づき、Size-fair モデル GP を改良し、部分木のサイズのみでなく部分解ピースの種であると考えている頻出部分木に基づく複合的ブロー抑制戦略を提案する。

エリート個体集合において、ラベル付き順序木である部分木 f_l が含まれる個体が $\lambda |E(g)|$ 以上である場合、 f_l を“頻出部分木”と定義する。ここで、 λ は正数である*2。また、 $\mathcal{F} = \{f_l(g); l = 1, \dots, L\}$ を g 世代におけるサイズ α の頻出部分木集合とする。提案法は、部分解の種として良い部分解ピースを探すこと、同時にブローを避けるため、交叉

*1 ほぼ同じ大きさの部分木を選択する手法は種々考えられる。ここでは最も基本的なものを用いた。詳細は 4.1 を参照されたい。

*2 λ はパラメータであり、実験では $\lambda = 0.5$ とした。

後の個体（木）のサイズが母親木とほとんど同じになることを目標とする。

$S(p_i(g))$ を個体 $p_i(g)$ の部分木集合、 $S(n(p_i^m))$ をノード $n(p_i^m)$ より木の階層が深くかつルートノードが $n(p_i^m)$ である部分木集合とする。 $S(n(p_i^m))$ は $S(p_i(g))$ の部分集合であり、

$$S(n(p_i^m)) \subset S(p_i(g)),$$

である。次に、 $n(p_i^m)$ をルートとする最大部分木 t_i と $n(p_j^m)$ をルートとする最大部分木 t_j との類似度 $g(n(p_i^m), n(p_j^m))$ を

$$g(n(p_i^m), n(p_j^m)) = \text{Sim}(n(p_i^m), n(p_j^m)), \quad (i \neq j), \quad (1)$$

と定義する。また、ノード $n(p_i^m)$ の頻出レベルを評価するために、頻出レベル

$$f(n(p_i^m)) = |\mathcal{F}(g) \cap S(n(p_i^m))|,$$

を定義する。つまり、ノード $n(p_i^m)$ 以下の部分木がエリート個体から抽出される頻出部分木を多く含んでいる場合は頻出レベルは高くなる。ブロードを抑制しつつ頻出部分木を考慮した交叉点を求めるため、 g 評価 $g(n(p_i^m), n(p_j^m))$ と f 評価 $f(n(p_i^m))$ を組み合わせ、この提案評価尺度 $F(n(p_i^m))$ に基づき、次に示す複合的ブロード戦略を提案する。

- P0:** Extract frequent trees $T(g)$ among elite individuals.
- P1:** Set $k \leftarrow 1$.
- P2:** Set $P_k(g) \leftarrow P(g)$.
- P3:** If $k \leq l$, then perform Steps S2 to S13, otherwise stop.
- P4:** Set a mother tree $p_k \leftarrow p_i \in P_k(g)$ at random.
- P5:** Set $P_k(g) \setminus p_k$.
- P6:** Set a father tree $p_{k+1} \leftarrow p_i \in P_k(g)$ at random.
- P7:** Choose a crossover node $n(p_k^{mum})$ at random from $\mathcal{N}(p_k(g))$.
- P8:** Extract the maximum subtree s_{mum} whose root node is $n(p_k^{mum})$.
- P9:** Calculate a size of s_{mum} .
- P10:** Choose a crossover node $n(p_{k+1}^m)$ of a father tree with probability proportional to evaluation measure $F(n(p_{k+1}^m))$.
- P11:** Generate a new individual $p_k(g+1)$ by crossover between p_k and p_{k+1} , where crossover points are $n(p_k^m)$ and $n(p_{k+1}^m)$.
- P12:** Set $k \leftarrow k+1$.
- P13:** Return to Step P2.

4. 評価実験

4.1 実装

提案複合的ブロード抑制戦略の実装方法を説明する。まず初めに、ステップ P10 における t_i と t_j の類似度と $F(n(p_i^m))$ の実装法について述べる。次に、サイズ α の頻出木集合 $T(g)$ の求め方について述べる。

t_i と t_j の類似度は簡単のため boolean 関数を用い、

$$\text{Sim}(n(p_{k+1}^m), n(p_k^{mum})) = \begin{cases} 1 & (-\epsilon \leq |s_{dad} - |s_{mum}|| \leq \epsilon), \\ 0 & (\text{otherwise}), \end{cases} \quad (2)$$

とする。ここで、

$$s_{dad} = \operatorname{argmax}_{s \in S(n(p_{k+1}^m))} S(s),$$

であり、 $S(s) = |s|$ である。提案法で用いる提案評価法は先に述べた類似度と f 評価をもとに交叉点を決定することでブロード抑制と性能向上を図る。つまり、評価尺度 $F(n(p_{k+1}^m))$ を

$$F(n(p_{k+1}^m)) = \frac{g(n(p_{k+1}^m), n(p_k^{mum})) \cdot f(n(p_{k+1}^m))}{\sum_{m=1}^M g(n(p_{k+1}^m), n(p_k^{mum})) \cdot f(n(p_{k+1}^m))},$$

と定める。その際、部分木候補は式 (2) により抽出され、小さすぎる部分木候補や大きすぎる部分木候補は取り除かれる。従って、提案評価尺度 $F(n(p_{k+1}^m))$ を用いることで、母親木の交叉点以下のノードサイズ $|s_{mum}|$ と父親木からの部分木のサイズがほぼ同じ大きくなることで母親個体とほぼ同じ大きさの子個体が生成されるだけでなく、 f 評価により多くの部分解の候補となる部分解ピースが多く含む子個体が生成されやすくなる。

次に、 g 世代のエリート個体 $E(g)$ から β 頻出の頻出木部分木集合 $T(g)$ の抽出法を述べる。一般にラベル付き順序木集合からの部分木の抽出は計算量が多くなるため、GP 手法において部分木の利用に関して制限を設ける必要があった。実験では、 $T(g)$ を *FREQT*[7] を用いて抽出する。この手法の計算量は抽出する頻出部分木のサイズに関してほぼ線形であり、効率が良いアルゴリズムである。提案法は世代毎に比較的小さい頻出部分木の抽出を行い、*FREQT* を用いるため、Size-fair モデル GP に比べて計算量が大きく増えないようにした。

4.2 比較法

提案法のノードサイズを Size-fair モデル GP と一点交叉法と比較する。本論文では一点交叉法を“Original 法”と呼ぶ。また、スタック GP 問題の性能比較において、一般的な手法として LMA で最適化した荷重メジアンフィルタ [13] との解の性能比較を行う。LMA は最も基本的な学習画像を用いた荷重メジアンフィルタの最適化手法であり、許容精度 γ と固定ステップサイズ μ の 2 種類のパラメータを必要とする。予備実験より、 $0.01 \leq \gamma \leq 0.1$ および $0.1 \leq \mu \leq 1.0$ とした。学習画像として原画像を用いる。

4.3 関数同定問題

性能評価実験において提案法の有効性を関数同定問題を用いて検証する。ここで、関数同定問題について簡単に説明する。非終端ノードが $\{+, -, \times, /, \sin, \cos\}$ 、終端ノードが $\{s, 0.05, 0.10, 0.15, 0.20, \dots, 0.95, 1.00\}$ であるラベル付き順序木で構成される関数空間 \mathcal{X} における関数を同定する。ここで、 s は変数であり、学習データは 20 点の $\{(s_j, x_s(s_j)) \in \mathbf{R}^2; j = 1, \dots, 20\}$ から成り、関数 1 から 3 および 5 は $s_j = 0.2(j-1) - 1$ 、関数 4 は $s_j = 0.2(j-1)$ とする。 $x_s(s) \in \mathcal{X}$ を同定する関数とした場合、任意の $x(s) \in \mathcal{X}$

に関して評価関数 $f(x)$ を次の通り定義し、 $f(x)$ の最小化問題を考える。

$$f(x) = 50 \sum_{j=1}^{20} |x(s_j) - x_*(s_j)|.$$

$x_*(s)$ として、GP のための関数同定問題のベンチマーク [14] より、次の 5 つの関数を用いる。

Function 1: $x^8 + 3 \cos(x) + x^5 + x,$

Function 2: $x^5 + x^4 + x^3 + \frac{x^2}{4} + x,$

Function 3: $\cos(1) \times \sin(x^3) - \ln(1) + \exp(x+1) + x^4,$

Function 4: $\frac{x}{x+1} - x^3 + \pi,$

Function 5: $x^6 + x^5 + x^4 + x^3 + x^2 + x.$

4.4 スタック GP 問題

性能評価実験において提案法の有効性をスタック GP 問題を用いて検証する。ここで、スタック GP 問題について簡単に説明する。インパルスノイズが重畳した画像の復元するためには、原画像の画素情報の保存とノイズの推定を同時に実現するフィルタを設計する必要がある。そのため、最適なスタックフィルタを求めるための目的関数はノイズがない画素の保存とノイズが付加された画素の復元の両方を考えた設計にする必要がある。原画像の画素集合とノイズが付加された画像の画素集合をそれぞれ

$$\mathcal{T}_o = \{t_o; o = 1, \dots, T^2\},$$

$$\mathcal{T}_\delta = \{t_\delta; \delta = 1, \dots, T^2\},$$

とする。 T^2 は画素数を表す。また、 $t_o(i, j)$ と $t_\delta(i, j)$ はそれぞれ画素 t_o および t_δ の画素値とし、 $1 \leq i \leq T, 1 \leq j \leq T$, (i, j) は座標とする。このとき、処理画素 $t_r \in \mathcal{T}_\delta$ の N 近傍を次のように定義する。

$$\mathcal{T}_\delta^N(t_r) = \{t_\delta \in \mathcal{T}_\delta \mid d(t_\delta, t_r) < N\},$$

ここで、 d は t_δ と処理画素 $t_r \in \mathcal{T}_\delta$ との距離を示す。そのため、 N 近傍の画素数は $(2N+1)^2$ となる。実験では、近傍として画素 t_r を中心とする正方窓および、256 段階のグレースケール画像を用いた。インパルスノイズを持つ画像は

$$t_\delta(i, j) = \begin{cases} \xi & (\text{prob. } \phi), \\ t_o(i, j) & (\text{otherwise}), \end{cases}$$

のように作成した [15]。ここで ξ は非負の一様分布を持つ整数乱数値である。

本実験では、非終端ノードが $\{MIN, MAX\}$ 終端ノードが $\{t_1, \dots, t_k, \dots, t_{(2N+1)^2}\}$ であるラベル付き順序木で構成される関数空間 X における GP スタック問題の精度を検証する。近傍幅が $N = 1$ の場合、フィルタ窓の要素数は 9 となる。任意のラベル付き順序木 $x \in X$ に関して、目的関数 $f(x)$ を

$$f(x) = \frac{1}{|T^2|} \sum_{i,j} |O(i, j) - t_o(i, j)|^2,$$

と定義し、 $f(x)$ の最小化問題を考える。ここで、 $O(i, j)$ は

GP により求めたフィルタ x を (i, j) 画素に適用した出力値であり、

$$O(i, j) = x(\mathcal{T}_\delta^N(t_r)),$$

である。 (i, j) は $t_r \in \mathcal{T}_\delta$ の座標を表す。

4.5 実験設定

実験で用いるパラメータは、交叉率 0.9, 突然変異率 0, 個体数 $I = 50$, 最大ノード数 $M = 100$, エリート個体数 20 とした。また、初期個体のノードサイズは 25 から 35 とし、ランダムに生成した。また、スタック GP 問題において、図 1 に実験に用いる原画像を示す。これらは Brodatz のテクスチャ画像 [16] から特徴的な部分を切り取ったものである。それぞれの画像は 256 階調のグレースケールで 64×64 ピクセルである。実験では、確率 ϕ を 10%, 20% および 40% として異なる雑音レベルのテスト画像を生成した。また、フィルタ窓は 5×5 の正方窓を用い、 N は 2 とする。

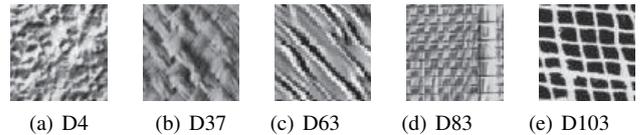


図 1 原画像

Fig. 1 Original images.

4.6 プロット抑制評価

まず初めに、ノードサイズに関して提案法と Size-fair モデル GP と Original 法の性能を比較する。図 2 および図 3 は 30 試行の \bar{N}_g の平均

$$\bar{N}_g = \frac{1}{I} \sum_{i=1}^I |\mathcal{N}(p_i(g))|,$$

を示す。図 2 は関数同定問題の結果、図 3 はスタック GP 問題の結果を表す。スタック GP 問題に関しては他のノイズ値においても同様の結果を示したため、紙面の都合上、ノイズ 20% の結果のみを示す。図 2 より、Original 法はノード数が大きくなっていることが分かる。特に、図 2(c) ではノード数の上限を超えていることが分かる。また、図 3 より、Original 法はノード数が急激に増加していることが分かる。これらの結果はスタック GP 問題はプロットが起きやすいこと、関数同定問題においてもプロットが起きることを示しており、これら問題においてプロット抑制を考える必要があることを示している。

さらに、図 2 と図 3(a) では提案法と Size-fair モデルのノードサイズはほぼ同程度だが、その他の問題においては提案法のノードサイズは Size-fair モデル GP より少し大きくなっていることが分かる。提案法は頻出レベル $f(n(p_i^m))$ と部分木のサイズの両方を考慮するが、プロットが起きておらず、ノード数もせいぜい 2 倍程度であることが分かる。これらの結果より、部分木のノードサイズを考えることはプロット抑制に非常に効果的であり、提案法で用いる頻出レベル $f(n(p_i^m))$ を導入した場合においても、急激なプロットを起こさないことが分かった。

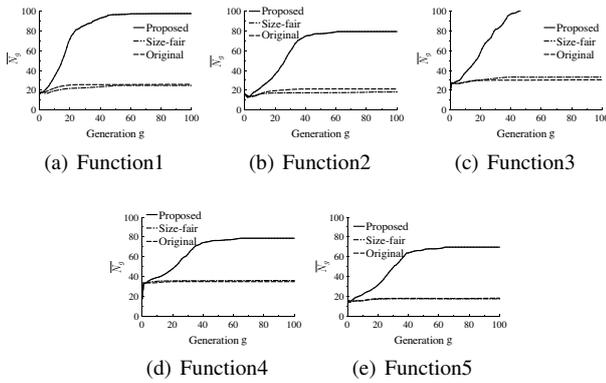


図2 平均ノードサイズ(関数同定問題).

Fig. 2 Average node size (Symbolic Regression problems).

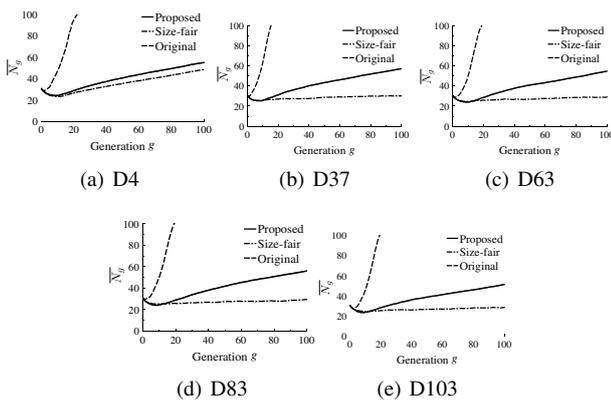


図3 平均ノードサイズ(スタック GP 問題 ノイズ 20%).

Fig. 3 Average node size(Stack GP problems with noise 20%).

4.7 性能評価

表2から表4にMSEの結果を示す. それぞれの問題において最も解の性能が高かった結果を太文字で, 提案法とSize-fairモデルGPを比べて性能が高い方を記号*で表し, それらの値は $\min_{1 \leq l \leq L} f(p_l(g))$ の30試行平均値である. スタックGP問題において, ノイズ10%と20%ではLMAにより設計された荷重メジアンフィルタはその他の手法と比べて性能が高い場合, 低い場合があるが, ノイズ40%の結果では常に高い性能を示している. しかしながら, LMAは許容精度 γ と固定ステップサイズ μ を最適に設定する必要があり, 性能に大きく影響を与える. 次に, 提案法とSize-fairモデルGPの結果を比較する. 関数同定問題では5つの問題中3つで, スタックGP問題にいておいてはノイズ10%のD103画像以外の問題では提案法がSize-fairモデルGPより性能が高いことが分かる. これらの結果は $f(n(p_i^m))$ を組み込む, 提案法の有効性を示している.

4.8 分析

本論文で, 我々は頻出レベル $f(n(p_i^m))$ が大きいとき, 母集団 $P(g)$ は頻出木を多く含み, 良い部分解ピースを保持していると仮説を立てた. つまり, 提案評価基準 $f(n(p_i^m))$ は破壊的な遺伝操作から優良な部分解を守り, 母集団内で

表1 MSEの結果(関数同定問題).

Table 1 Results of MSE (Symbolic Regression problems).

Function	Proposed	Size
Function1	*122.8	128.9
Function2	*62.8	68.4
Function3	79.3	*72.9
Function4	*163.2	175.7
Function5	62.5	*60.3

表2 MSEの結果(ノイズ10%の画像).

Table 2 Results of MSE (Images with noise 10%).

Image	Degraded	Proposed	Size	LMA
D4	4100.8	*252.7	257.6	260.7
D37	1907.8	*126.2	126.4	137.3
D63	4363.3	*317.2	322.3	301.7
D83	3121.9	*311.3	312.1	307.6
D103	8174.7	455.2	*445.3	462.1

表3 MSEの結果(ノイズ20%の画像).

Table 3 Results of MSE (Images with noise 20%).

Image	Degraded	Proposed	Size	LMA
D4	4543.9	*487.7	495.5	518.5
D37	2494.7	*243.5	248.9	234.7
D63	4797.0	*576.5	585.4	542.0
D83	3549.3	*540.1	548.3	532.3
D103	8707.2	*953.9	976.3	1001.3

表4 MSEの結果(ノイズ40%の画像).

Table 4 Results of MSE (Images with 40%).

Image	Degraded	Proposed	Size	LMA
D4	5687.7	*1186.8	1190.5	1035.5
D37	3722.2	*555.0	565.9	497.0
D63	5808.7	*1241.9	1271.0	1123.9
D83	4519.2	*1128.1	1142.4	1030.8
D103	9984.3	*2509.9	2517.7	2331.7

の頻出木の生成を促進すると考えている. ここでは, 提案法における30回試行の平均の頻出木数 $|T(g)|$ の履歴を分析する. 図4より世代数 g が増加するにつれ母集団の頻出部分木数 $|T(g)|$ が増加することが分かる. これらの結果は提案法において頻出木の保存が促進され, 頻出部分木が増加することを示しており, これらより, 新しい評価基準 $f(n(p_i^m))$ を組み込むことで良い部分解ピースの探索が促進されると考えられる. 図4(b)から(f)のスタックGP問題の結果では, ノイズレベルが高い場合, ノイズレベルが低い場合に比べて頻出木の数も増加していることが分かる. これらの結果はノイズレベルが高い難しい問題において, 提案法がより多くの頻出木が保存されるように働き, 部分解ピースの探索を促進しているためだと考えられる.

最後に, スタックGP問題において得られた窓形状について考察を行う. 図5から図6に処理後の画像と得られた窓形状を示す. ここで, 30回試行の最も評価値が良かった解の窓形状を示している. 図(c)の数字は解(木構造)に

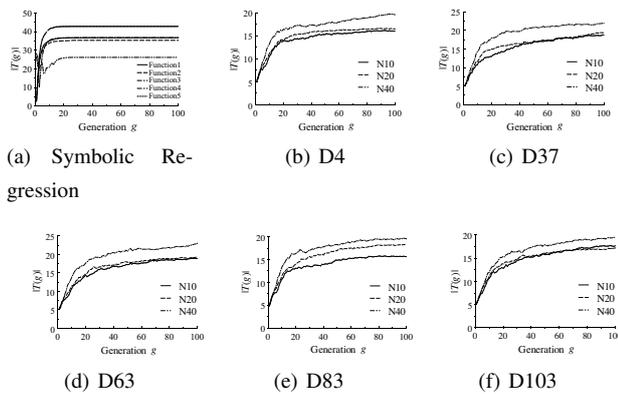


図4 頻出部分木数の履歴.

Fig. 4 History of the number of frequent trees.

において、 t_k (終端ノード) が出現した数を示す。ここではノイズ 20% の D4 と D103 画像の結果のみを示す。図 5 から窓形状は左下から右上への形状となっており、D4 画像の形状と類似していることが分かった。また D103 画像のノイズ 20% の問題は MSE が最も悪い難しい問題であるが、図 6 より十字形状が現れており、提案法により妥当な窓形状が得られることが分かった。

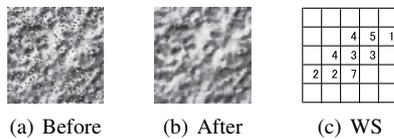


図5 Results of the proposed method in the image D4 with noise 20%.

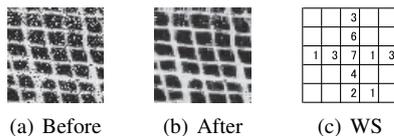


図6 Results of the proposed method in the image D103 with noise 20%.

5. まとめ

GP において効果的なブロー抑制戦略を考えることは重要な課題である。Size-fair モデル GP は高いブロー抑制能力を持つ手法であり、適用のしやすさから優れた手法であるが、破壊的な遺伝操作より解構造を守るメカニズムが組み込まれておらず、そのため、交叉において部分木の大きさのみを考えることは解の性能向上の観点から十分でなく、本論文では我々は Size-fair モデル GP の改良を試みた。本論文で、ビルディングブロック仮説に基づき、頻出木を用いた新しい評価基準“頻出レベル”を提案し、ノードのサイズと頻出レベルを組み込んだ複合的ブロー抑制戦略を提案した。広く知られているベンチマーク問題と、スタック GP 問題を用いて、提案法はブロー抑制効果が高いこと、Size-fair を比べ高い性能を実現したことを確認し、提案法の有効性を示した。

謝辞 本研究は科研費 (26730133,26330290) の助成を受けた。

参考文献

- [1] W. B. Langdon, T. Soule, R. Poli, and J. A. Foster, “The evolution of size and shape,” in *Advances in Genetic Programming 3*, L. Spector, W. B. Langdon, U.-M. O’Reilly, and P. J. Angeline, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. 8, pp. 163–190.
- [2] H. Majeed and C. Ryan, “A less destructive, context-aware crossover operator for GP,” in *Proceedings of the 9th European Conference on Genetic Programming*, ser. Lecture Notes in Computer Science, P. Collet, M. Tomassini, M. Ebner, S. Gustafson, and A. Ekárt, Eds., vol. 3905. Springer, 10–12 April 2006, pp. 36–48.
- [3] W. B. Langdon, “Size fair and homologous tree crossovers for tree genetic programming. genetic programming and evolvable machines,” in *Genetic Programming and Evolvable Machines*, 2000, pp. 95–119.
- [4] R. Poli and N. F. McPhee, “General schema theory for genetic programming with subtree-swapping crossover: Part II,” *Evolutionary Computation*, vol. 11, no. 2, pp. 169–206, May 2003.
- [5] N. F. McPhee, B. Ohs, and T. Hutchison, “Semantic building blocks in genetic programming,” in *Proceedings of the 11th European conference on Genetic programming EuroGP’08*, 2008, pp. 134–145.
- [6] T. Soule and J. A. Foster, “Removal bias: a new cause of code growth in tree based evolutionary programming,” in *In 1998 IEEE International Conference on Evolutionary Computation*. IEEE Press, 1998, pp. 781–186.
- [7] T. Asai, K. Abe, S. Kawasoe, H. Sakamoto, and S. Arikawa, “Efficient substructure discovery from large semi-structured data,” in *Proceedings of SIAM International Conference on Data Mining SDM’02*, 2002, pp. 158–174.
- [8] W. B. Langdon and R. Poli, “Fitness causes bloat,” in *Soft Computing in Engineering Design and Manufacturing*. Springer-Verlag, 1997, pp. 23–27.
- [9] S. C. Roberts, D. Howard, and J. R. Koza, “Evolving modules in genetic programming by subtree encapsulation,” in *Proceedings of the 4th European conference on Genetic programming EuroGP’01*, 2001, pp. 160–175.
- [10] Y. Shan, R. I. McKay, D. Essam, and H. A. Abbass, “A survey of probabilistic model building genetic programming,” in *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, ser. Studies in Computational Intelligence, M. Pelikan, K. Sastry, and E. Cantu-Paz, Eds. Springer, 2006, vol. 33, ch. 6, pp. 121–160.
- [11] K. Yanai and H. Iba, “Estimation of distribution programming based on bayesian network,” in *Proceedings of the 2003 Congress on Evolutionary Computation CEC’03*, vol. 3, dec. 2003, pp. 1618–1625.
- [12] K. Ono, Y. Hanada, M. Kumano, and M. Kimura, “Island model genetic programming based on frequent trees,” in *IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 2988–2995.
- [13] L. Yia, J. Astola, and Y. Neuvo, “Adaptive stack filtering with application to image processing,” *IEEE Transactions on Signal Processing*, vol. 41, no. 1, pp. 162–184, 1993.
- [14] C. Tuite, M. O’Neill, and A. Brabazon, “Towards a dynamic benchmark for genetic programming,” in *GECCO (Companion)*. ACM, 2013, pp. 151–152.
- [15] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, “A new efficient approach for the removal of impulse noise from highly corrupted images,” *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 1012–1025, 1996.
- [16] P. Brodatz, *Textures : a photographic album for artists and designers*. New York: Dover Publications, 1966.