

# 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<sup>1,a)</sup> 花田 良子<sup>2,b)</sup>

受付日 2014年8月27日, 再受付日 2014年10月17日/2014年12月22日,  
採録日 2015年3月3日

**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, 効率良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.

**キーワード:** 遺伝的プログラミング, ブloat, 部分木サイズ, 頻出部分木

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

KEIKO ONO<sup>1,a)</sup> YOSHIKO HANADA<sup>2,b)</sup>

Received: August 27, 2014, Revised: October 17, 2014/December 22, 2014,  
Accepted: March 3, 2015

**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 the 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.

**Keywords:** Genetic Programming, bloat control, tree size, 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

<sup>1</sup> 龍谷大学理工学部電子情報学科  
Department of Electronics and Informatics, Ryukoku University, Otsu, Shiga 520-2194 Japan

<sup>2</sup> 関西大学システム理工学部  
Faculty of Engineering Science, Kansai University, Suita, Osaka 564-8680, Japan

a) kono@rins.ryukoku.ac.jp

b) hanada@kansai-u.ac.jp

は、交叉法によりブロートを抑制する手法が考えられる。Langdon ら [1], Ryan ら [2], [3] 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 [4], [5], [6], [7]。このなかで最も効果的なブロート抑制法として Size-fair モデル GP があげられる [8]。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ト抑制が可能手法である。

一方、進化計算におけるビルディングブロック仮説 [9], [10], [11] 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 [12]、さら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、性能向上とブロート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ト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 [13]、最右拡張と枝切り技術を用いて効率良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。

また、提案法の評価では、GP の性能評価で一般的に用いられるベンチマーク問題として関数同定問題と、実問題としてスタックフィルタ設計問題を対象とする。スタックフィルタはメジアンフィルタやモルフォロジカルフィルタに代表される順序統計フィルタを一般的に記述したものである [14], [15]。GP は解を木構造で表すことができるため、フィルタ設計と親和性が高く、Miller ら [16] や, Erba ら [17] は GP 用いた組合せ回路のフィルタ設計などの手法を提案している [18], [19]。スタックフィルタの設計はフィルタ窓の要素と AND/OR 論理演算の組合せととらえることができ、AND/OR 論理演算はそれぞれ MIN/MAX 演算に置き換えることができる。これらの要素および演算子の組合せは木構造で表すことができるため、GP を用いた最適化と親和性が高い。また、GP を用いたスタックフィルタの設計問題は処理画像と原画像との MSE を最小にする、要素と演算子の組合せを求める問題と考えることができる。この問題を本論文ではスタック GP 問題と呼ぶ。スタック GP 問題で扱う演算には冪等性があるため、イント

ロンがおきやすい問題であり、効率良いスタック GP 問題の解法が望まれている。

本論文構成は次のとおりである。2章では、本論文で扱う問題の定式化および Size-fair モデル GP の詳細を述べる。3章では、提案 Size-fair レベルと頻出レベルに基づく提案複合ブロート制御戦略について述べる。4章では、提案法の有効性を関数同定問題とスタック GP 問題を用いて示す。テスト画像として、インパルス雑音を含むテキスト画像を用いる。ここでは、まず提案法の実装を説明し、提案法のブロート抑制効果をベースラインとして1点交叉を用いる Original 法と Size-fair モデル GP と比較する。次に、解探索性能に関し、提案法と Size-fair モデル GP と比較する。スタック GP 問題においては、広く用いられている画像ノイズ除去法との比較も行う。さらに、一般に、頻出木の抽出には計算時間がかかるため、計算時間に関し Size-fair モデル GP と提案法を比較する。実験結果より、Size-fair モデル GP に頻出部分木情報を用いた部分解の形成を促進する手法を組み込むことにより、ブロートの抑制および解探索能力が向上することを示し、提案評価基準および提案ブロート抑制戦略の有効性を示す。

## 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ト抑制方法は Size-fair 交叉を適用することである。前章で述べたように、ブロート抑制は種々あるが、Size-fair 交叉は Langdon らにより提案された最もシンプルで効果的なブロート抑制法である。したがって、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。 $P(0)$  は初期世代個体集合を表し、 $p_i(g)$  のノード集合を

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

とする。Size-fair モデル GP の交叉は次のとおりである。

**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 the mother tree  $p_k \leftarrow p_i \in P_k(g)$  at random.

**S4:** Set  $P_k(g) \setminus p_k$ .

**S5:** Set the 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

$\mathcal{N}(p_k(g))$ .

**S7:** Extract the maximum subtree  $s_{mum}$  whose root node is  $n(p_k^m)$ .

**S8:** Calculate the 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、 $| \cdot |$  はノード数を表す。 $\epsilon$  は整数値のパラメータであり、実験では、 $\epsilon = 2$  とした。

### 3. 複合的ブロー抑制戦略

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 [20]. 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 [9], [10]. 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 [11], 確率モデルを用いた様々な手法が提案されている [21], [22]. ここで、部分木のデータベースには、母集団で一定回以上、安定して出現する部分木が保存され、データベースに保存されるリストの大きさは設定されている。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り、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、これらをうま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島モデル GP の移住戦略を提案し [23], 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な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、これらをうま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。

ラベル付き順序木  $t$  は  $(V_t, E_t, LA_t, la_t, v_t^0, \preceq_t)$  の6つの組からなり、 $V_t$  はノード集合、 $E_t$  はリンク集合、 $LA_t$  はラベル集合、 $la_t: V_t \rightarrow LA_t$  はラベリング関数、 $v_t^0$  はルート、 $\preceq_t$  は兄弟関係を表す。ラベル付き順序木  $t$  と  $x$  において、 $V_t$  から  $V_x$  の単射  $\varphi: V_t \rightarrow V_x$  で、親子関係を保存 ( $(v_1, v_2) \in E_t$  iff  $(\varphi(v_1), \varphi(v_2)) \in E_x$ ) し、兄弟関係を保存 ( $v_1 \preceq_t v_2$  iff  $\varphi(v_1) \preceq_x \varphi(v_2)$ ) し、さらにラベルを保存 ( $la_t(v) = la_x(\varphi(v))$ ) するものが存在するとき、 $t$  は  $x$  の部分木という。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  $ft$  が含まれる個体が  $\lambda|E(g)|$  以上である場合、 $ft$  を“頻出部分木”と定義する。ここで、 $\lambda$  は正数である\*2。また、 $\mathcal{F} = \{ft_l(g); l = 1, \dots, L\}$  を  $g$  世代におけるサイズ  $\alpha$  の頻出部分木集合とする。提案法は、部分解の種としてよ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、ノード  $n(p_i^m)$  をルートとする最大部分木  $t_i$  とノード  $n(p_j^m)$  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

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

と定義する\*3。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、新しい評価値  $F(n(p_i^m))$  を次のように定義する。つまり、

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

\*3  $Sim(n(p_i^m), n(p_j^m))$  の詳細な設計に関しては、4.1 節を参照されたい。

\*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 節を参照されたい。



$$\begin{aligned}
 0 &\leq F(n(p_i^m)) \leq 1, \\
 f(n(p_i^m)) &< f(n(p_i^{m'})) \Rightarrow F(n(p_i^m)) < F(n(p_i^{m'})), \\
 g(n(p_i^m), n(p_j^m)) &< g(n(p_i^{m'}), n(p_j^{m'})) \\
 &\Rightarrow F(n(p_i^m)) < F(n(p_i^{m'})),
 \end{aligned}$$

とする. ここで  $i \neq j$  である ( $F(n(p_i^m))$ ) の詳細な設計に関しては 4.1 節を参照されたい. 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ツ戦略を提案する.

- P0:** Extract frequent trees  $T(g)$  among elite individuals.
- P1:** Set  $k \leftarrow 1$ .
- P2:** Set  $P_k(g) \leftarrow P(g)$ .
- P3:** If  $k \leq I$ , then perform Steps S2 to S13, otherwise stop.
- P4:** Set the mother tree  $p_k \leftarrow p_i \in P_k(g)$  at random.
- P5:** Set  $P_k(g) \setminus p_k$ .
- P6:** Set the 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 the 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.

提案法は Size-fair モデル GP をベースとし, 性能を向上を目的としている. GP 法において, 良好な部分木構造を保存する手法としては, イントロンは破壊的な遺伝的操作より解構造を守る役割があるという報告があるように, イントロンと呼ばれる解性能に影響を与えない部分木を増やす手法や, 提案法のように交叉法を改良する手法が考えられる. 本研究では, 我々はブローツを制御する新しい評価尺度を提案し, Size-fair モデル GP に対して頻出木を保存するメカニズムを組み込むことで, 解探索能力の向上とブローツ抑制が可能である, 複合的ブローツ制御戦略を実現する.

#### 4. 評価実験

広く用いられるベンチマーク問題とスタック GP 問題を用い, 提案法の有効性を評価する.

#### 4.1 実装

提案複合的ブローツ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  $\alpha$  の頻出木集合  $T(g)$  の求め方について述べる.

実験では,  $t_i$  と  $t_j$  の類似度は簡単のために boolean 関数を用い,

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

とする. ここで,  $u = 1, \dots, M$ ,

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

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ツ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. したがって, 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さに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  $\beta$  頻出の頻出部分木集合  $T(g)$  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かかるため, 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 [13] 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. 提案法は世代ごと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

提案法はブローツ抑制の促進と性能向上のために, ノードの Size-fair レベルと頻出レベルを部分木のサイズと頻出木に基づいて評価する. そのため, まず初めに, 提案法のノードサイズ (プログラムサイズ) を Size-fair モデル GP および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。提案法は良い部分解ピースを見つけるために頻出木を抽出しそれらの保存を促進するが、一般的に頻出木の抽出には計算量がかかる。次に、提案法と Size-fair モデル GP の解の性能評価と計算時間の評価をそれぞれ 4.7 節および 4.8 節で行う。

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 (Least Mean Absolute algorithm) で最適化した荷重メディアンフィルタ [24] 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ディアフィルタの最適化手法であり、許容精度  $\gamma$  と固定ステップサイズ  $\mu$  の 2 種類のパラメータを必要とする。予備実験より、実験では、 $0.01 \leq \gamma \leq 0.1$  および  $0.1 \leq \mu \leq 1.0$  とした。また、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  $\{(s_j, x_*(s_j)) \in \mathbf{R}^2; j = 1, \dots, 20\}$  からなり、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。 $x_*(s) \in \mathcal{X}$  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  $f(x)$  を次のとおり定義し、 $f(x)$  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 [25] 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を復元するためには、原画像の画素情報の保存とノイズからの画素復元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値の保存とノイズが付加された画素値の推定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

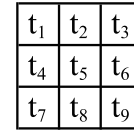


図 1 フィルタ窓における  $t_k$  の配置

Fig. 1 Layout of terminal nodes in a filter window.

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

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

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  $t_\delta$  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  $t_\delta$  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 [26]。ここで  $\xi$  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  $\mathcal{X}$  における GP スタック問題の精度を検証する。図 1 は近傍幅  $N = 1$  で処理画素  $t_r = t_5$  の場合の終端ノード  $t_k$  の位置を示している。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  $x \in \mathcal{X}$  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\*4。 $(i, j)$  は  $t_r \in \mathcal{T}_\delta$  の座標を表す。図 2 は個体  $x$  の木構造の例を示す。

### 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、

\*4 実際には復元対象の劣化画像と特徴が類似した学習画像による推定、あるいは劣化画像から直接推定する目的関数のもとで推定が行われる。ここでは交叉の性能を検証するために単純に原画像を用いた。

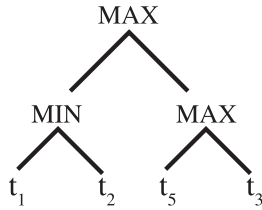


図 2 個体の例

Fig. 2 Example of an individual.

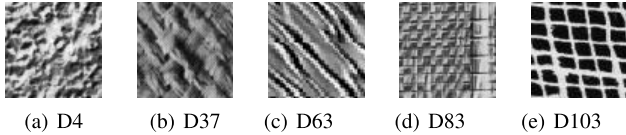


図 3 原画像

Fig. 3 Original images.

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。すべての実験は Intel Xeon X5675 3.07GHz プロセッサ 6 コア, 8GB のメモリ, Linux が搭載された PC を用いた。

また, スタック GP 問題において, 図 3 に実験に用いる原画像を示す [27]。それぞれの画像は 256 階調のグレースケールで  $64 \times 64$  ピクセルである。実験では, 確率  $\phi$  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  $5 \times 5$  の正方窓を用い, 近傍サイズ  $N$  は 2 とする。

#### 4.6 ブロート抑制評価

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。結果はそれぞれの世代  $g$  における 30 試行の平均値を示す。図 4 および図 5 は 30 試行の  $\bar{N}_g$  の平均を, 世代数  $g$  の関数として示している。つまり,

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

である。図 4 は関数同定問題の結果, 図 5 はスタック GP 問題の結果を表す。また図 5 (f) は D4 画像において世代数を 300 まで増やした結果を示す。ここで, 直線は提案法, 一点鎖線は Size-fair モデル GP, 点線は Original 法の結果をそれぞれ示す。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 4 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 4 (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 5 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 問題はブロートが起きやすいこと, 関数同定問題においてもブロートが起きることを示しており, これら

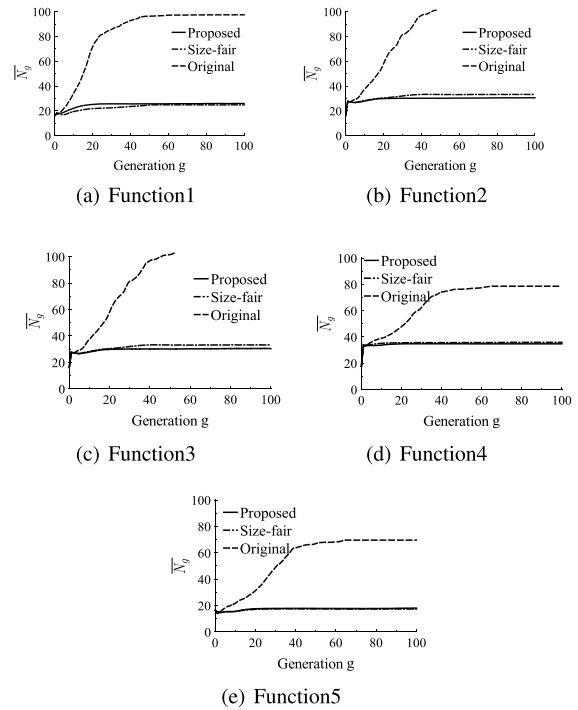


図 4 各世代  $g$  における平均ノードサイズ (関数同定問題)

Fig. 4 History of the average node size in each generation  $g$  (Symbolic Regression problems).

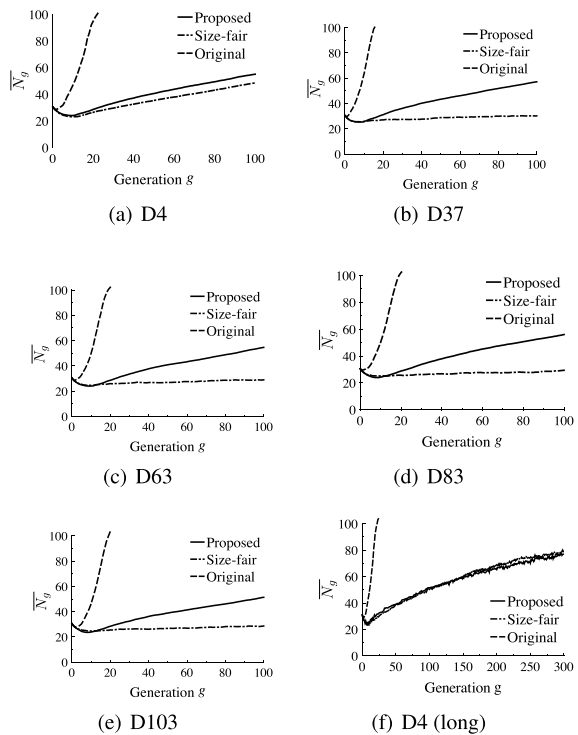


図 5 各世代  $g$  における平均ノードサイズ (スタック GP 問題ノイズ 20%)

Fig. 5 History of the average node size in each generation  $g$  (Stack GP problems with noise 20%).

問題においてブロート抑制を考える必要があることを示している。

さらに, 図 4 と図 5 (a) では提案法と Size-fair モデル

表 1 MSE の結果 (関数同定問題, ノイズ 10%の画像)

Table 1 Results of MSE (Symbolic Regression problems and images with noise 10%).

Function	Symbolic Regression		Image	Noise 10%			
	Proposed	Size-fair		Degraded	Proposed	Size-fair	LMA
Function1	*122.8	128.9	D4	913.7	*252.7	257.6	260.7
Function2	*62.8	68.4	D37	734.0	*126.2	126.4	137.3
Function3	79.3	*72.9	D63	865.7	*317.2	322.3	301.7
Function4	*163.2	175.7	D83	803.1	*311.3	312.1	307.6
Function5	62.5	*60.3	D103	1333.8	455.2	*445.3	462.1

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, プロートが起きておらず, ノード数もせいぜい 2 倍程度であることが分かる. 図 5(f) において世代数を増やした場合においても, 急激なノードの増加現象であるプロートが起きないことが分かった. これらの結果より, 部分木のノードサイズ  $g(n(p_{k+1}^m), n(p_k^{mum}))$  を考えることはプロー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, 急激なプロートを起こさないことが分かった.

#### 4.7 性能評価

4.6 節において, Original 法は提案法および Size-fair モデル GP と比べてノードサイズが非常に大きくなることが分かった. 次に, プロート抑制の観点から性能が良かった提案法および Size-fair モデル GP に対し, 解の精度に関して性能を評価する. また, インパルス雑音除去で有効な荷重メジアンフィルタの代表的な最適化手法である LMA で設計されたフィルタの結果とも比較する.

表 1 と表 2 に MSE の結果を示す. 関数同定問題では提案法と Size-fair モデル GP の MSE の結果を, スタック GP 問題ではそれらの結果に加え, LMA の結果とノイズ除去前の画像と原画像との MSE の結果をそれぞれ示す. また, 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  $\min_{1 \leq i \leq I} f(p_i(g))$  の 30 試行平均値である.

まず初めに, 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%の結果ではつねに LMA は高い性能を示していることが分かる. LMA は許容精度  $\gamma$  と固定ステップサイズ  $\mu$  のパラメータを最適に設定する必要があり, 表 3 はノイズ 20%の D37 画像における LMA の結果を示す. 表 3 において, 最も良い結果と最も悪い結果を太文字で示す. これらの結果より, あらかじめ最も良いパラメータを求めることは難しく, またパラメータにより大きく性能も異なる

表 2 MSE の結果 (ノイズ 20%および 40%の画像)

Table 2 Results of MSE (Images with noise 20% and 40%).

Image	Noise 20%			
	Degraded	Proposed	Size-fair	LMA
D4	1712.3	*487.7	495.5	518.5
D37	1539.8	*243.5	248.9	234.7
D63	1728.1	*576.5	585.4	542.0
D83	1538.0	*540.1	548.3	532.3
D103	2738.8	*953.9	976.3	1001.3
Image	Noise 40%			
	Degraded	Proposed	Size-fair	LMA
D4	3639.0	*1186.8	1190.5	1035.5
D37	3144.6	*555.0	565.9	497.0
D63	5804.2	*1241.9	1271.0	1123.9
D83	3142.9	*1128.1	1142.4	1030.8
D103	5804.2	*2509.9	2517.7	2331.7

るため, スタック GP 問題において最適なフィルタを GP を用いて自動的に求める我々のアプローチはこの問題に適していることが分かる.

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  $f(n(p_i^m))$  を組み込むことの効果を示しており, 提案法は  $g(n(p_{k+1}^m), n(p_k^{mum}))$  だけでなく  $f(n(p_i^m))$  も考慮することで, 4.6 節においてプロート抑制効果もあることが分かっている. これらの結果は提案法の有効性を示している.

#### 4.8 Size-fair モデル GP との計算時間の比較

提案法は  $f(n(p_i^m))$  の評価のために頻出部分木集合  $T(g)$  を抽出する. 一般に, ラベル付き順序木集合からの頻出部分木の抽出は多くの計算量を必要とするタスクであるため, 提案法と Size-fair モデル GP との計算時間と最終世代におけるノードサイズ  $\overline{N}_g$  の比較を行うことで, 提案法の有用性を検証する.

表 4 と表 5 より, 提案法は Size-fair モデル GP に比べて多くの計算時間が必要であることが分かる. しかし, Function4 ではノードサイズが提案法の方が小さく, 提案



表 3 ノイズ 20% の D37 の画像の LMA の結果  
Table 3 Results of LMA in the D37 image with noise 20%.

$\mu$	$\gamma$	$f(x)$	$\mu$	$\gamma$	$f(x)$	$\mu$	$\gamma$	$f(x)$	$\mu$	$\gamma$	$f(x)$	$\mu$	$\gamma$	$f(x)$
0.1	0.01	313.5	0.3	0.01	236.4	0.5	0.01	241.1	0.7	0.01	260.3	0.9	0.01	272.7
0.1	0.02	444.2	0.3	0.02	259.7	0.5	0.02	237.7	0.7	0.02	235.1	0.9	0.02	234.9
0.1	0.03	475.7	0.3	0.03	308.6	0.5	0.03	253.3	0.7	0.03	241.2	0.9	0.03	235.3
0.1	0.04	483	0.3	0.04	384.9	0.5	0.04	274.1	0.7	0.04	254.3	0.9	0.04	235.3
0.1	0.05	497.1	0.3	0.05	446.8	0.5	0.05	312.9	0.7	0.05	268.7	0.9	0.05	236.4
0.1	0.06	502.5	0.3	0.06	473.2	0.5	0.06	347.2	0.7	0.06	304.3	<b>0.9</b>	<b>0.06</b>	<b>234.7</b>
0.1	0.07	502.5	0.3	0.07	498.7	0.5	0.07	412.2	0.7	0.07	340	<b>0.9</b>	<b>0.07</b>	<b>234.7</b>
0.1	0.08	507.1	0.3	0.08	498.7	0.5	0.08	480.8	0.7	0.08	340	<b>0.9</b>	<b>0.08</b>	<b>234.7</b>
0.1	0.09	510.2	0.3	0.09	498.7	0.5	0.09	523.8	0.7	0.09	340	<b>0.9</b>	<b>0.09</b>	<b>234.7</b>
0.1	0.1	510.2	0.3	0.1	519.4	<b>0.5</b>	<b>0.1</b>	<b>570.3</b>	0.7	0.1	340	0.9	0.1	240.8
0.2	0.01	240.9	0.4	0.01	236.8	0.6	0.01	250.4	0.8	0.01	265	1.0	0.01	317.4
0.2	0.02	312.3	0.4	0.02	239.8	0.6	0.02	236.5	0.8	0.02	234.9	1.0	0.02	308.7
0.2	0.03	426.2	0.4	0.03	266.9	0.6	0.03	245.2	0.8	0.03	235.7	1.0	0.03	316.2
0.2	0.04	462.1	0.4	0.04	311.7	0.6	0.04	269.2	0.8	0.04	244	1.0	0.04	301.4
0.2	0.05	483.4	0.4	0.05	367	0.6	0.05	294.3	0.8	0.05	267.2	1.0	0.05	315.9
0.2	0.06	483.4	0.4	0.06	424.8	0.6	0.06	323.1	0.8	0.06	267.2	1.0	0.06	293.8
0.2	0.07	499.5	0.4	0.07	482.3	0.6	0.07	364.3	0.8	0.07	289.3	1.0	0.07	314.4
0.2	0.08	499.5	0.4	0.08	514.7	0.6	0.08	387.7	0.8	0.08	289.3	1.0	0.08	314.4
0.2	0.09	499.5	0.4	0.09	514.7	0.6	0.09	415.2	0.8	0.09	289.3	1.0	0.09	287.3
0.2	0.1	511.0	0.4	0.1	514.7	0.6	0.1	415.2	0.8	0.1	289.3	1.0	0.1	313.9

表 4 Size-fair モデル GP との比較 (関数同定問題, ノイズ 10% の画像)

Table 4 Results of comparison with the Size-fair Model GP (Symbolic Regression problems and images with noise 10%).

	Symbolic Regression				Noise 10%				
	Proposed		Size		Image	Proposed		Size	
Function	Time [sec.]	$\overline{N}_g$	Time [sec.]	$\overline{N}_g$		Time [sec.]	$\overline{N}_g$	Time [sec.]	$\overline{N}_g$
Function1	160.7	26.0	93.8	24.8	D4	379.6	48.5	349.1	44.4
Function2	137.1	21.3	81.1	18.3	D37	416.5	53.2	269.7	29.5
Function3	124.9	30.5	89.3	33.2	D63	408.9	52.8	257.4	27.2
Function4	82.1	34.7	93.4	35.9	D83	365.3	46.8	254.2	26.8
Function5	120.8	18.1	78.3	17.4	D103	392.5	52.1	257.3	28.9

法の方が計算時間が短い。また、ノイズ 10% および 20% の D4 画像はノードサイズが同程度であり、計算時間も提案法と Size-fair モデル GP は同程度である。提案法は Size-fair モデル GP に比べ、アルゴリズム P0 における頻出木の抽出に  $O(2\beta LV)^{*5}$ 、アルゴリズム P10 における頻出部分木とのマッチングに  $O(|T(g)|V)$  の計算量が多くかかる。ここで  $V$  は個体のノード数を示す。しかしながら、提案法は Size-fair モデル GP の計算時間と比較すると 2 倍以下になっている。この理由として、提案法は *FREQT* を計算量の削減のために用いたこと、比較的小さい部分木を対象として頻出部分木を抽出し、かつ、抽出回数を 1 世代に 1 回としたことが考えられる。これらの結果は提案法の有用性を示している。

\*5  $\mathcal{L}$  は *FREQT* において最右拡張で生成される最右リストの長さを示す。

#### 4.9 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、母集団内での頻出木の生成を促進すると考えている。4.6 節においてノードサイズに関して Original 法、Size-fair モデル GP および提案法を比較し、4.7 節において解の性能に関して Size-fair モデル GP と提案法を比較した。ここでは、提案法における頻出木数  $|T(g)|$  の履歴を分析する。図 6 に 30 回試行の平均の頻出部分木数を示し、図 6(a) は関数同定問題の結果を図 6(b) から (f) はスタック GP 問題の結果を示す。ここで、図中の  $N$  はノイズの率を表し、N10 はノイズを 10% 含む画像を意味する。

図 6 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



表 5 Size-fair モデル GP との比較 (ノイズ 20% および 40% の画像)

Table 5 Results of comparison with the Size-fair Model GP (images with noise 20% and 40%).

Image	Noise 20			
	Proposed		Size	
	Time [sec.]	$\bar{N}_g$	Time [sec.]	$\bar{N}_g$
D4	416.3	55.1	374.1	48.5
D37	447.2	57.2	276.2	30.2
D63	422.0	54.7	268.0	29.0
D83	420.1	56.0	264.1	29.4
D103	386.2	51.4	261.8	28.6
Image	Noise 40			
	Proposed		Size	
	Time [sec.]	$\bar{N}_g$	Time [sec.]	$\bar{N}_g$
D4	448.9	59.5	432.9	54.5
D37	486.5	62.4	467.2	59.4
D63	480.8	62.7	426.4	54.0
D83	458.7	60.0	418.6	54.0
D103	435.8	57.4	402.0	53.1

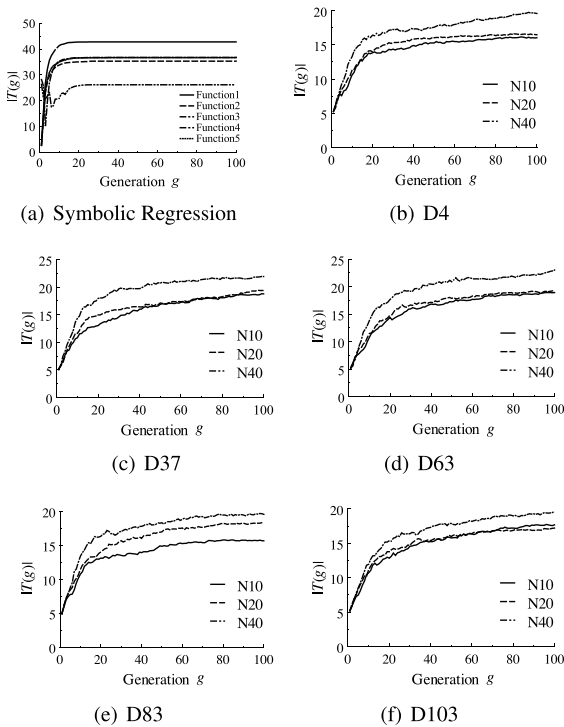


図 6 頻出部分木数の履歴

Fig. 6 History of the number of frequent trees.

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。提案法は  $f(n(p_i^m))$  のみでなく部分木のサイズ  $g(n(p_{k+1}^m), n(p_k^{mum}))$  も考慮することで Size-fair モデル GP より高い性能を示すことを 4.7 節で示した。図 6(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

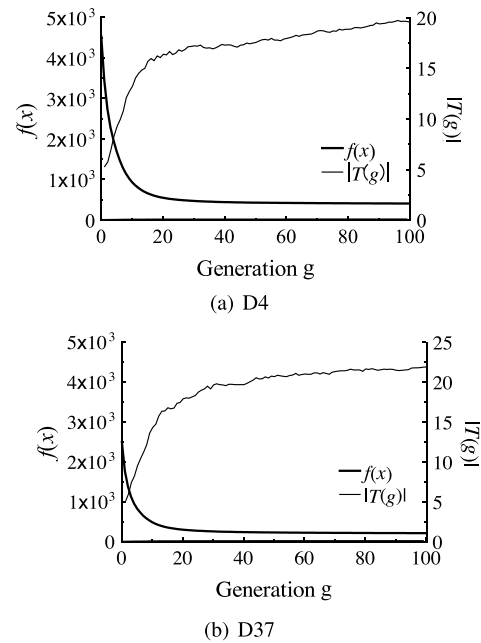


図 7 頻出部分木数と評価値の履歴

Fig. 7 History of the number of frequent trees and  $f(x)$ .

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。

図 7 にノイズ 20% の D4 画像および D37 画像における提案法の頻出木数  $|T(g)|$  の履歴と評価値  $f(x)$  の履歴を示す。第 1 縦軸は  $\min_{1 \leq i \leq I} f(p_i(g))$  の 30 試行平均値であり、第 2 縦軸は頻出木数  $|T(g)|$  を示す。この結果より、探索初盤に  $|T(g)|$  が急激に大きくなるにつれて評価値も良くなっていることが分かり、 $|T(g)|$  と解の評価値には関連があることを示している。一方、探索終盤において評価値は変化しないが頻出木数が少しずつ増加していることが分かる。終盤においてエリート個体の解構造が似てくるのが原因だと考えられる。大域的最適解に収束している場合は問題がないが、多くの解が局所解で停滞する場合、探索に有効でない部分木も頻出部分木として抽出されることが考えられる。探索が停滞しており、かつ解構造がまったく同じ解が複数存在する場合は、探索途中の優良部分木の保存を促進するために、重複解を除去するなどの改良が必要であり、これは今後の課題である。また、本研究では提案法の有効性を検証しやすくするために突然変異率を 0 としたが、局所解への収束を防ぐために突然変異を導入することも考えられる。

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 8, 図 9, 図 10, 図 11, 図 12 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 (a) は処理前画像、(b) は処理後画像を示し、図 (c) の数字は解(木構造)において、 $t_k$  (終端ノード) が出現した数を示

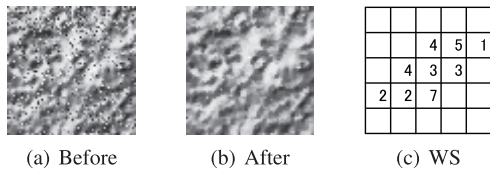


図 8 ノイズ 20% の D4 画像における提案法の結果

Fig. 8 Results of the proposed method in the image D4 with noise 20%.

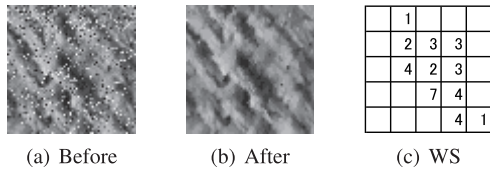


図 9 ノイズ 20% の D37 画像における提案法の結果

Fig. 9 Results of the proposed method in the image D37 with noise 20%.

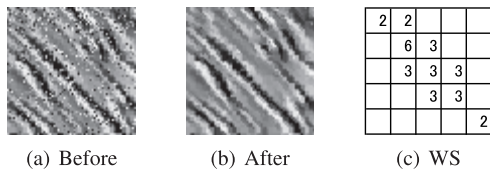


図 10 ノイズ 20% の D63 画像における提案法の結果

Fig. 10 Results of the proposed method in the image D63 with noise 20%.

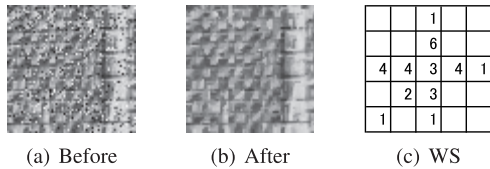


図 11 ノイズ 20% の D83 画像における提案法の結果

Fig. 11 Results of the proposed method in the image D83 with noise 20%.

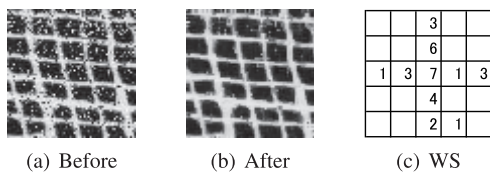


図 12 ノイズ 20% の D103 画像における提案法の結果

Fig. 12 Results of the proposed method in the image D103 with noise 20%.

す。ノイズ 10% と 40% の結果はノイズ 20% の結果と同様であったため、ここではノイズ 20% の結果のみを示す。

図 8 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、図 9 と図 10 より、窓形状は左上から右下への形状となっており、これらもまた原画像の形状と類似していることが分かった。D83 画像と D103 画像は十字形状がある画像であり、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 12 より妥当な窓形状が提案法により得られることが分かった。

### 5. まとめ

GP において効果的なブloat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loat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、また、イントロンにはそれらから解を守る役割があるという報告もある。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loat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loat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た。また、提案法を計算時間、頻出木の増加量の観点から分析し有用性を示した。スタック GP 問題においては、画像形状と提案法により得られた窓形状が一致することを示し、提案法の有効性を示した。

提案法は Size-fair モデル GP と頻出部分木を用いた複合的ブloat抑制手法であり、まず第一歩として最もシンプルな頻出木の保存法を用いた。しかしながら、頻出木によってそれぞれ部分解ピースになるための重要度が異なることは自然に考えられる。これらを考慮することは今後の課題である。

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

### 参考文献

- [1] Langdon, W.B., Soule, T., Poli, R. and Foster, J.A.: The evolution of size and shape, *Advances in Genetic Programming 3*, Spector, L., Langdon, W.B., O'Reilly, U.-M. and Angeline, P.J. (Eds.), ch. 8, pp.163-190, MIT Press, Cambridge, MA, USA (1999).
- [2] Majeed, H. and Ryan, C.: A less destructive, context-aware crossover operator for GP, *Proc. 9th European Conference on Genetic Programming*, ser. Lecture Notes in Computer Science, Collet, P., Tomassini, M., Ebner, M., Gustafson, S. and Ekárt, A. (Eds.), Vol.3905, pp.36-48, Springer, 10-12, April (2006).
- [3] Fitzgerald, J., Azad, R.M.A. and Ryan, C.: Bootstrapping to reduce bloat and improve generalisation in genetic programming, *GECCO (Companion)*, pp.141-142 (2013).
- [4] Ito, T., Iba, H. and Sato, S.: Depth-dependent crossover for genetic programming, *Proc. 1998 IEEE World Congress on Computational Intelligence*, pp.775-780, IEEE Press (1998).
- [5] Crawford-Marks, R. and Spector, L.: Size control via

- size fair genetic operators in the pushgp genetic programming system, *GECCO*, pp.733-739, Morgan Kaufmann (2002).
- [6] Xie, H. and Zhang, A.M.: Depth-control strategies for crossover in tree-based genetic programming, *Soft Computing*, Vol.15, pp.1865-1878 (2011).
- [7] Kim, P.F.M., Becker, Y.L. and O'Reilly, U.-M.: Constrained genetic programming to minimize overfitting in stock selection, *Genetic Programming Theory and Practice VI, Genetic and Evolutionary Computation*, pp.179-195, Springer (2008).
- [8] Langdon, W.B.: Size fair and homologous tree crossovers for tree genetic programming. genetic programming and evolvable machines, *Genetic Programming and Evolvable Machines*, pp.95-119 (2000).
- [9] Poli, R. and McPhee, N.F.: General schema theory for genetic programming with subtree-swapping crossover: Part II, *Evolutionary Computation*, Vol.11, No.2, pp.169-206 (2003).
- [10] McPhee, N.F., Ohs, B. and Hutchison, T.: Semantic building blocks in genetic programming, *Proc. 11th European conference on Genetic programming, EuroGP'08*, pp.134-145 (2008).
- [11] Roberts, S.C., Howard, D. and Koza, J.R.: Evolving modules in genetic programming by subtree encapsulation, *Proc. 4th European Conference on Genetic Programming, EuroGP'01*, pp.160-175 (2001).
- [12] Soule, T. and Foster, J.A.: Removal bias: a new cause of code growth in tree based evolutionary programming, *In 1998 IEEE International Conference on Evolutionary Computation*, pp.781-186, IEEE Press (1998).
- [13] Asai, T., Abe, K., Kawasoe, S., Sakamoto, H. and Arikawa, S.: Efficient substructure discovery from large semi-structured data, *Proc. SIAM International Conference on Data Mining, SDM'02*, pp.158-174 (2002).
- [14] Soille, P.: *Morphological Image Analysis Principles and Applications, 2nd ed.*, Springer (2003).
- [15] Kulkarni, R.K., Lahoti, C.B. and Meher, S.: Impulse denoising using improved progressive switching median filter, *ICWET*, pp.586-590, ACM (2010).
- [16] Miller, J.F., Job, D. and Vassilev, V.K.: Principles in the evolutionary design of digital circuits-part I, *Genetic Programming and Evolvable Machines*, Vol.1, No.1/2, pp.7-35 (2000).
- [17] Erba, M., Rossi, R., Liberali, V. and Tettamanzi, A.: An evolutionary approach to automatic generation of VHDL code for low-power digital filters, *Genetic Programming, Proc. EuroGP'2001*, ser. LNCS, Miller, J.F., Tomassini, M., Lanzi, P.L., Ryan, C., Tettamanzi, A.G.B. and Langdon, W.B. (Eds.), pp.36-50, Vol.2038, Springer-Verlag (2001).
- [18] Garmendia-Doval, A.B., Mohan, C.K. and Prasad, M.K.: Evolving tree representations of stack filters, *Genetic Programming 1998: Proc. 3rd Annual Conference*, Koza, J.R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D.B., Garzon, M.H., Goldberg, D.E., Iba, H. and Riolo, R. (Eds.), pp.103-108 (1998).
- [19] Vasíček, Z., Bidlo, M. and Sekanina, L.: Evolution of efficient real-time non-linear image filters for fpgas, *Soft Comput.*, Vol.17, No.11, pp.2163-2180 (2013).
- [20] Langdon, W.B. and Poli, R.: Fitness causes bloat, *Soft Computing in Engineering Design and Manufacturing*, pp.23-27, Springer-Verlag (1997).
- [21] Shan, Y., McKay, R.I., Essam, D. and Abbass, H.A.: A survey of probabilistic model building genetic programming, *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*, ser. Studies in Computational Intelligence, Pelikan, M., Sastry, K. and Cantu-Paz, E. (Eds.), Vol.33, ch.6, pp.121-160, Springer (2006).
- [22] Yanai, K. and Iba, H.: Estimation of distribution programming based on bayesian network, *Proc. 2003 Congress on Evolutionary Computation, CEC'03*, Vol.3, pp.1618-1625 (2003).
- [23] Ono, K., Hanada, Y., Kumano, M. and Kimura, M.: Island model genetic programming based on frequent trees, *IEEE Congress on Evolutionary Computation*, pp.2988-2995, IEEE (2013).
- [24] Yia, L., Astola, J. and Neuvo, Y.: Adaptive stack filtering with application to image processin, *IEEE Trans. Signal Processing*, Vol.41, No.1, pp.162-184 (1993).
- [25] Tuite, C., O'Neill, M. and Brabazon, A.: Towards a dynamic benchmark for genetic programming, *GECCO (Companion)*, pp.151-152, ACM (2013).
- [26] Abreu, E., Lightstone, M., Mitra, S.K. and Arakawa, K.: A new efficient approach for the removal of impulse noise from highly corrupted images, *IEEE Trans. Image Processing*, Vol.5, No.6, pp.1012-1025 (1996).
- [27] Brodatz, P.: *Textures: a photographic album for artists and designers*, Dover Publications, New York (1966).



小野 景子 (正会員)

2003年同志社大学大学院工学研究科修士課程修了。2007年同志社大学大学院工学研究科博士課程修了。博士(工学), 2009年同志社大学研究開発推進機構省エネルギー照明システム研究センター特定任用研究員(助教)。

2010年龍谷大学理工学部電子情報学科助教。2014年龍谷大学理工学部電子情報学科講師, 現在に至る。並列処理, 最適設計, 進化計算等の研究に従事。IEEE, 電気学会, 進化計算学会各会員。



花田 良子 (正会員)

2007年同志社大学大学院工学研究科博士後期課程修了。博士(工学)。日本学術振興会特別研究員(PD)を経て, 2008年関西大学システム理工学部助教, 現在に至る。進化計算, 最適設計等の研究に従事。IEEE, 電子情報通信学会, 進化計算学会各会員。

2010年龍谷大学理工学部電子情報学科助教。2014年龍谷大学理工学部電子情報学科講師, 現在に至る。並列処理, 最適設計, 進化計算等の研究に従事。IEEE, 電気学会, 進化計算学会各会員。