

# 画像生成フィルタの構築における GP と SAP の比較

藤田 宗佑<sup>1</sup> 廣安 知之<sup>2</sup> 三木 光範<sup>3</sup> 渡辺 章人<sup>1</sup>  
<sup>1</sup>同志社大学 <sup>2</sup>同志社大学生命医科学部 <sup>3</sup>同志社大学理工学部

## Comparison between GP and SAP in construction of Image Processing Filters

Sosuke FUJITA<sup>1</sup> Tomoyuki HIROYASU<sup>2</sup> Mitsunori MIKI<sup>3</sup> Akihito WATANABE<sup>1</sup>

<sup>1</sup>Doshisha University <sup>2</sup>Department of Life and Medical Sciences, Doshisha University  
<sup>3</sup>Department of Science and Engineering, Doshisha University

**Abstract:** Genetic Programming (GP) is one of evolutionary computation algorithms that can design tree topology operations. Simulated Annealing Programming (SAP) is also an emergent algorithm that can create tree topology operations. These two algorithms, GP and SAP, are applied to construct Image Processing Filters. These filters are useful for finding the information of cancers from medical images. Image processing filters can be expressed as tree topology operations. In this paper, the search characteristics of GP and SAP were compared. As a result, SAP produced the result whose tree length is short. On the other hand, GP obtained good solutions whose filters can distinguish the image precisely.

### 1. はじめに

画像処理の分野において、原画像より特定部位を抽出する技術は重要な技術である。また、画像処理は様々なフィルタを組み合わせることで目的とする画像処理を行う。しかし、複雑な画像処理を行うにつれて、処理の組み合わせの結果を予想することや、最適な画像処理を行うための画像生成フィルタの特定は困難となる。そこで長尾ら [1] は、進化的計算法を用いることで、画像処理を組み合わせ最適化問題として捉え画像処理を行う、進化的画像処理という手法を考案した。長尾らは、この進化的画像処理に遺伝的アルゴリズム (Genetic Algorithm: GA) や、遺伝的プログラミング (Genetic Programming: GP) などの進化的計算法を適用することにより、画像生成フィルタの構築を行っている。

本研究では、この進化的画像処理に対し、個体に構造的な表現を用いることが可能となる GP および、シミュレーテッドアニーリングプログラミング (Simulated Annealing Programming: SAP) を適用する。また、画像生成フィルタの構築における GP と SAP の性能比較を行う。

### 2. 遺伝的プログラミング (Genetic Programming: GP) とシミュレーテッドアニーリングプログラミング (Simulated Annealing Programming: SAP)

本研究では、自動プログラミングの手法として、遺伝的プログラミング (Genetic Programming: GP) とシミュレーテッドアニーリングプログラミング (Simulated Annealing Programming: SAP) を用いる。以下、それぞれの進化的計算法について述べる。

#### 2.1 遺伝的プログラミング (Genetic Programming: GP)

遺伝的プログラミング (Genetic Programming: GP) [2] は、1992 年に John Koza らにより提案された進化的計算法である。また GP は、生物の進化を模倣した遺伝的アルゴリズム (Genetic Algorithm: GA) を木構造が扱えるように拡張した自動プログラミング手法である。GP では、選択、交叉、突然変異といった遺伝的オペレータを繰り返し行うことで、問題に適した木構造を生成する。以下に GP のアルゴリズムを示す。

##### Step 1 初期個体群の生成

複数の個体をランダムに生成し、初期個体群 (母集団) とする。また生成時、各個体の評価を行う。

##### Step 2 選択

各個体の評価値を判断基準にして、次世代に残す個体を選択する。本稿では、母集団の中からランダムに選択した一定数の個体の中から、評価値が最も高い個体を選択して次世代に残す処理を行うトーナメント選択を用いる。

##### Step 3 交叉

交叉対象となる個体から、ランダムに選択した交叉点をルートとする部分木同士を入れ替える (Fig. 1)。

##### Step 4 突然変異

突然変異の対象となる個体から、ランダムに選択した突然変異点以下の木を、ランダムに作成した突然変異木に置き換える (Fig. 2)。

##### Step 5 終了条件

終了条件に達するまで、Step2~4 の遺伝的オペレータを繰り返す。本稿では、指定した探索世代数に達すれば終了とする。

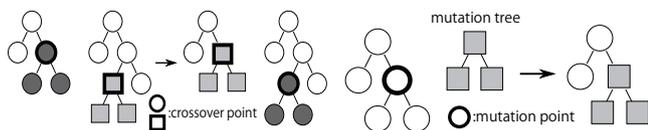


Fig. 1 Crossover

Fig. 2 Mutation

## 2.2 シミュレーテッドアニーリングプログラミング (Simulated Annealing Programming: SAP)

シミュレーテッドアニーリングプログラミング (Simulated Annealing Programming: SAP) [3] は、金属の焼き鈍しを模倣したシミュレーテッドアニーリング (Simulated Annealing: SA) を、木構造が扱えるように拡張した自動プログラミング手法である。SAP では、生成処理、受理判定、状態遷移、冷却を繰り返し行うことで、問題に適した木構造を生成する。以下に SAP のアルゴリズムを示す。

### Step 1 初期個体の生成

初期個体をランダムに生成し、その個体の評価を行う。

### Step 2 生成処理

現在の個体に対し、GP の突然変異 (Fig. 2) と同様の処理を行うことにより、新しい個体を生成し、その個体の評価を行う。

### Step 3 受理判定、状態推移

現在の個体の評価値  $E$  と生成処理により生成した新しい個体の評価値  $E'$  との差分  $\Delta E (= E' - E)$ 、および温度パラメータ  $T$  により、状態を推移するか否かの受理判定を行う。受理判定には、式 (1) に示す Metropolis 基準を用いる。

$$P_{AC} = \begin{cases} 1 & \text{if } \Delta E \leq 0 \\ \exp(-\frac{\Delta E}{T}) & \text{otherwise} \end{cases} \quad (1)$$

ここで  $P_{AC}$  は受理確率である。Metropolis 基準では、新しい個体が改良方向 ( $\Delta E \leq 0$ ) の場合は必ず受理し、それ以外の場合は、温度  $T$  と、評価値の差分  $\Delta E$  によって、受理する確率が変化する。

### Step 4 冷却

Step 2, 3 の処理を一定回数繰り返したならば、温度パラメータ  $T$  を小さくする冷却を行う。ここでは式 (2) に示すような指数型アニーリングを用いる。

$$T_{k+1} = \gamma T_k \quad (0.8 \leq \gamma \leq 1) \quad (2)$$

ここで、 $\gamma$  は冷却率であり、 $T_k$  は現在の温度、 $T_{k+1}$  は冷却後の温度を表す。

### Step 5 終了条件

終了条件に達するまで、Step 2 ~ 4 の処理を繰り返す。本稿では、指定した探索世代数に達すれば終了とする。

## 3. 自動プログラミング手法による画像生成フィルタの構築

本章では、自動プログラミング手法による画像生成フィルタの構築について述べる。

## 3.1 概要

先行研究として、長尾らが提案している進化的画像処理がある。この進化的画像処理とは、実現したい未知の画像処理を、既知の単純な画像フィルタ (基本フィルタ) の組み合わせとして表現し、進化的計算法を用いて最適な基本フィルタの組み合わせを求めるものである。このように、画像処理の問題を組み合わせ最適化問題と捉えることで、複雑な処理を行う画像処理でも、画像処理を意識することなく、最適な画像生成フィルタを構築することが可能となる。

本研究では、各個体を木構造状フィルタで表す。木構造状フィルタについては、3.2 節で述べる。また予め、学習用画像セット (入力画像、目標画像、重み画像) を用意し、自動プログラミングを行う際に学習させる。評価は適合度関数を用いて、出力画像と目標画像を比較することで行う。適合度関数については、3.3 節で述べる。

## 3.2 木構造状フィルタ

GP や SAP のような自動プログラミング手法を用いた画像生成フィルタの構築には、Fig. 3 に示すような、木構造状フィルタを用いて出力画像の生成を行う。

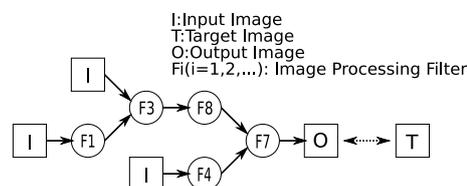


Fig. 3 Principle of automatic Image Processing with tree filter

ここでは、画像処理の対象になる画像を原画像、処理結果として示した画像を目標画像とし、目標画像は出力画像との比較に用いることで、画像生成フィルタの学習を行う。このとき、目標画像、重み画像は原画像から手作業で作成したものを利用する。出力画像は、まず原画像を、構築した画像生成フィルタの終端ノードに入力し、木構造の各ノードに格納された画像処理を終端ノードから順番に実行することで作成する。また、木構造状フィルタを構成する基本フィルタを、Table 1, Table 2 に示す。

Table 1 Basic filters with an input

Filter ID	Effect
f101	Mean
f102	Maximum value of eighborhood
f103	Minimum value of eighborhood
f104...f106	Edge emphasis(sobel, laplacian, dark edge)
f107	Contraction
f108	Expansion
f109	Inversion
f110...f112	Binarization(threshold:128, 64, 192)
f113, f114	Addition(value:25, -25)
f115	Maximization(threshold:128)
f116	Minimization(threshold:128)
f117, f118	Multiplication(value:1.5, 0.5)

Table 2 Basic filters with two input

Filter ID	Effect
f201	Logical Sum
f202	Logical Product
f203	Algebraic Sum
f204	Algebraic Product
f205	Bounded Sum
f206	Bounded Product

### 3.3 適合度関数

各個体の適合度は、画像生成フィルタからの出力画像  $O(x, y)$  と目標画像  $T(x, y)$  の差分によって求める。また適合度関数は、式 (3) に示すように、1.0 を最適解とする最大値問題とする。

$$fitness = \frac{1}{K} \sum_{k=1}^K \left\{ 1 - \frac{\sum_{x=1}^{W_x} \sum_{y=1}^{W_y} w_i(x, y) |O_i(x, y) - T_i(x, y)|}{\sum_{x=1}^{W_x} \sum_{y=1}^{W_y} w_i(x, y) \cdot V_{max}} \right\} \quad (3)$$

ここで、 $K$  は学習用画像セット数、 $w(i, j)$  は重み画像の値、 $V_{max}$  は最大階調値を示す。

本研究では、適合度関数に重み画像を適用することを考える。重み画像とは”画像中の画素の重要度を表す 0.0 ~ 1.0 までの重み値を各画素の値にする画像”である。この重み画像を用いることで、目標画像中において抽出すべき領域の重みを設定することが可能となる。また、複数の学習用画像セットを用いることで、画像処理手法の最適化において汎用性を持たせ、未知の画像に対しても同様の画像処理が行えるようにする。

## 4. 数値実験

遺伝的プログラミング (Genetic Programming: GP) およびシミュレーテッドアニーリングプログラミング (Simulated Annealing Programming: SAP) を進化的画像処理に適用し、画像生成フィルタの構築を行う。

### 4.1 評価事項

本研究では、以下の事項について、GP と SAP の比較を行う。

- 学習性能

学習性能とは、目的を達するためのプログラムを生成する能力である。良い自動プログラミング手法であるということは、学習性能が良好であるということである。

- プログラムサイズ

プログラムサイズが小さいほど、一般的に良い自動プログラミング手法である。

- ロバスト性

ロバスト性とは、ある学習環境で得られたプログラムが、未知の環境でも同等の性能を得られるという性質である。従って、一般的に、生成されるプログラムはロバスト性が高い方が好ましい。

### 4.2 GP と SAP の比較検討

画像生成フィルタの構築において、GP と SAP を適用し、比較実験を行う。GP および、SAP のパラメータを Table 3, Table 4 に示す。なお、SAP においては木構造の深さの制限を設けていない。また本研究では、学習用画像セットに Fig. 4 に示すような学習用画像セットを 1 セット使用する。

Table 3 Parameter of GP

Parameter	Value
Generations	40
Populations	100
Way of Selection	Tournament
Crossover Rate	1.0
Mutation Rate	0.01
Max Depth	25

Table 4 Parameter of SAP

Parameter	Value
Number of Evaluation	4000
Max Temperature	5.0
Min Temperature	0.01

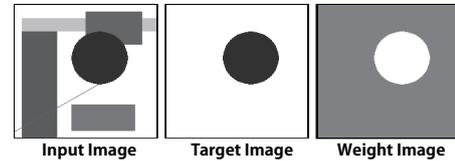


Fig. 4 Image set for learning

#### 4.2.1 学習性能

それぞれの自動プログラミング手法で得られた評価値の結果を Fig. 5 に示す。Fig. 5 は、横軸が評価計算回数、縦軸が評価値を示す。なお、結果は 30 試行の中央値である。

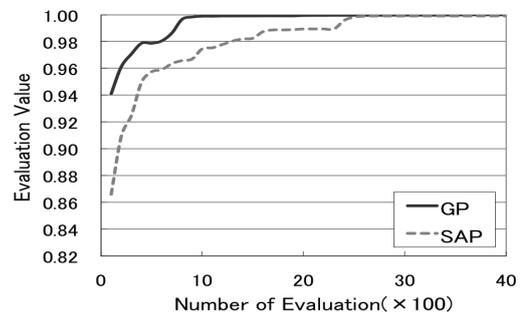


Fig. 5 Comparison between GP and SAP in Solution Research

Fig. 5 より、学習性能ではほぼ同等の結果を示した。SAP では途中、局所解に陥るが、最終的には局所解を抜け出し、GP とほぼ同等の解を生成していることが分かる。これは、SAP が SA の局所探索に優れた性質を引き継いでいるためであると考えられる。一方 GP は大域探索に優れた性質を持っているが、Fig. 5 では SAP よりも探索回数の早い段階で評価値が高くなっている。これは、SAP

では1個体で探索を進めるのに対し、GPでは多数の母集団の中から最適な解候補を求めているためであると考えられる。

#### 4.2.2 プログラムサイズ

それぞれの自動プログラミング手法で得られる木構造の深さとノード数について比較を行う。木構造の深さの結果を Fig. 6 に、ノード数の結果を Fig. 7 に示す。Fig. 6 は、横軸が評価計算回数、縦軸が木構造の深さを示し、Fig. 7 は、横軸が評価計算回数、縦軸が木構造のノード数を示す。なお、結果はそれぞれ30試行の中央値である。

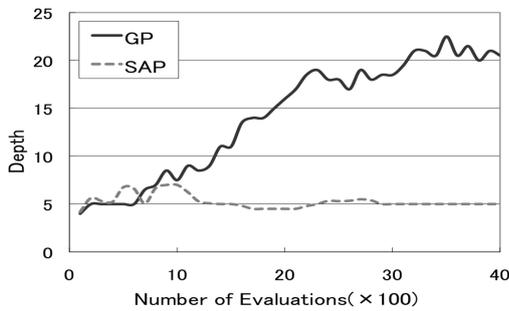


Fig. 6 Comparison between GP and SAP in Depth

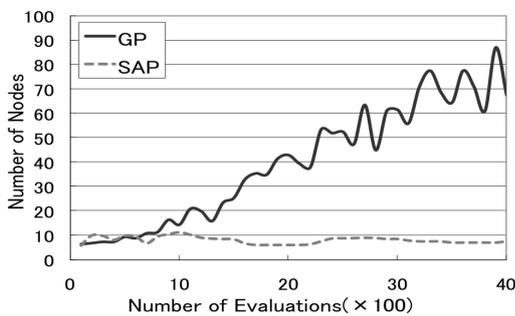


Fig. 7 Comparison between GP and SAP in Number of Nodes

Fig. 6 より、GPでは探索が進むにつれて、木構造の深さが増加し、探索の終盤では、設定した最大の深さ付近まで増加していることが分かる。一方SAPは、最大の深さを設定していないにも関わらず、木構造の深さは5付近に収束している。また、Fig. 7 より、GPでは探索が進むにつれて、木構造の深さと同様に、ノード数も増加している。一方SAPは、探索が進むにつれてもノード数は増加せず、最終的に10付近に収束している。以上より、GPでは過度のプロートが生じるのに対し、SAPでは過度のプロートは生じず、プログラムサイズも小さいものを生成することが可能であると分かる。

#### 4.2.3 ロバスト性の検討

GPおよび、SAPで構築した画像生成フィルタのロバスト性について検討する。検討方法は、Fig. 4の学習用画像セットを用いて構築した30個の画像生成フィルタを、それぞれFig. 8に示したような4つの未知の画像にも適用し、GP、SAPそれぞれ120回評価を行う。なお、目標画像、重み画像は、Fig. 8の原画像よりそれぞれ手作業で作成する。GP、SAPのロバスト性の比較の結果をFig.

9に示す。Fig. 9は、構築した画像生成フィルタをFig. 8に示した4つの画像に適用して得られた評価値の平均値、中央値、最大値、最小値を表したものである。

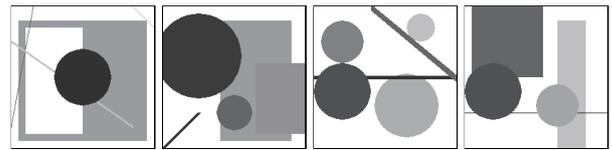


Fig. 8 Test images

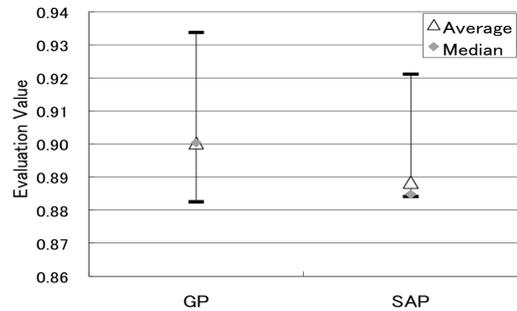


Fig. 9 Comparison between GP and SAP in Robustness

Fig. 9 より、GPでは評価値の最小値はSAPよりも低かったが、平均値、中央値ともGPの評価値の方が高くなった。そのため、GPの方がSAPよりも汎用性が高いことが分かった。これは、GPではノード数がSAPと比べて遥かに多いため、局所探索の実験の際に機能していなかった基本フィルタが機能して、SAPよりも良い結果が得られたと考えられる。

## 5. まとめ

本研究では、進化的画像処理を用いて、画像生成フィルタ構築におけるGPとSAPの比較を行った。実験結果から、GPはSAPと比べて同等以上の学習性能が得られ、また、より汎用性が高い画像生成フィルタが構築できた。しかし、GPではプログラムサイズが増加するといった問題があるため、今後の課題として、GPの過度なプロートの抑制について検討する必要があると考えられる。

## 参考文献

- [1] 長尾智晴, 進化的画像処理, 79/114, 昭晃堂, 2002
- [2] J.Koza, Genetic programming, on the programming of computers by means of natural selection, MIT Press, 1992
- [3] 藤田 佳久, 三木 光範, 橋本 雅文, 廣安 知之, シミュレーテッドアニーリングを用いた自動プログラミング, 情報処理学会研究報告. MPS, 数理モデル化と問題解決研究報告 IPSJ SIG Notes, No.19, 89/92, 2007