

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

藤田 佳久[†] 三木 光範^{††}
橋本 雅文^{††} 廣安 知之^{††}

本論文では、木構造により記述されたプログラムの生成を対象としたシミュレーテッドアニーリング(SA)について検討する。SAでは遺伝的プログラミング(GP)で用いられている突然変異を繰り返して行うことでプログラムを生成する。Santa Fe trail問題、Wall-following問題、および2種類のSymbolic Regression問題に一定温度の温度スケジュールを用いるSAを適用した結果、プロットが生じなかった。また、構文的イントロンが発生する可能性があるSanta Fe trail問題、およびWall-following問題では、SAは良好な探索性能を示すことが分かった。

Automatic Programming with Simulated Annealing

YOSHIHISA FUJITA,[†] MITSUNORI MIKI,^{††} MASAFUMI HASHIMOTO^{††}
and TOMOYUKI HIROYASU^{††}

In this paper, we examine Simulated Annealing (SA) to generate a program described by tree structure. SA is a mutation-based method. SA with fixed temperature schedules applied to the Santa Fe trail, Wall-following, and two Symbolic Regression problems. Bloating does not occur in SA with fixed temperature schedules. In addition, SA obtained good solutions in Santa Fe trail and Wall-following problems, which have the possibility of syntactic introns.

1. はじめに

ロボットを制御するプログラムや関数などのコンピュータプログラムは、コンピュータを用いて自動生成する場合がある。これは、コンピュータを用いることにより、あらかじめ人が想定できない状況にも対応できるプログラムを生成できることや、複数台のロボットが協調行動するような大規模・複雑なプログラムを容易に生成することができるからである。

このようなプログラムをメタヒューリスティック手法を用いて自動生成する代表的な手法として、遺伝的プログラミング(Genetic Programming: GP)¹⁾がある。GPは遺伝的アルゴリズム(Genetic Algorithm: GA)²⁾の遺伝子型を、木構造などの構造的表現が扱えるよう拡張した手法であり、LISPのS式のような木構造で記述できるプログラムを自動生成することができる。GP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て行うことで目的に適したプログラムを生成する。これまでにGPは、ロボットの行動

ルールや電子回路、関数、楽譜、画像フィルタなどを木構造で表現することによって、ロボットの制御プログラム^{3),4)}や電子回路の設計⁵⁾、株価予測⁶⁾、音楽の自動作曲⁷⁾、画像処理⁸⁾など様々な分野に応用されている。

このようにメタヒューリスティック手法を用いたプログラムの自動生成には、ほとんどの場合、GPが用いられている。しかし、多くの最適化問題では、より効率的に最適化を行うために、各問題に適した探索手法が用いられている。たとえば、立体骨組の最適化⁹⁾やマイクロクラスタの最低エネルギー構造の予測¹⁰⁾などにはGAが用いられており、利得等化フィルタの最適化¹¹⁾や空港ターミナル領域での軌跡最適化¹²⁾などにはシミュレーテッドアニーリング(Simulated Annealing: SA)¹³⁾が用いられている。このことから、プログラムの自動生成においても、問題に応じて探索手法を使い分けることで、より効率的に最適化を行える可能性があると考えられる。

そこで本研究では、GPの基となるGAと同じ代表的なメタヒューリスティック手法の1つであるSAを用いて木構造で表現できるプログラムの自動生成を行い、標準的なGPと比較することでSAの可能性について検討する。なお、本研究では、SAを用いたプログラ

[†] 同志社大学大学院工学研究科

Graduate School of Engineering, Doshisha University

^{††} 同志社大学工学部

Faculty of Engineering, Doshisha University

ムの自動生成方法を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) と呼ぶ。

SA の最大の特徴は、改良方向への遷移を認めるだけでなく、改悪方向への遷移も温度と呼ばれる制御パラメータにより確率的に認めることである。これにより、SA は局所解を持つ問題でも最適解を得ることが期待できる。

これまでに SA を用いたプログラムの自動生成の研究は、O'Reilly ら^{14),15)} と Langdon¹⁶⁾ によって行われている。O'Reilly らの研究^{14),15)} では新しい解候補の生成方法に HVL-Mutate を用いることで SA は GP と同等の性能を得ることができると報告されており、Langdon の研究¹⁶⁾ では the length neutral mutation と呼ばれる新しい解候補の生成方法を用いることで SA は GP の問題点であるプロートが生じないと報告されている。このようにプログラムの自動生成に SA を用いることは有効であると考えられる。

しかし、これらの研究では、SA の探索に多大な影響を及ぼすパラメータである温度スケジュールについてはまったく検討されていない。また、O'Reilly ら^{14),15)} や Langdon¹⁶⁾ の研究の後、SA を用いたプログラムの自動生成の研究はされておらず、プログラムの自動生成における SA の有効性はまだ十分に明らかになっていないと考える。

本研究では、SAP において温度スケジュールを検討し、標準的な GP と比較を行うことで、木構造で記述できるプログラムの自動生成における SA の有効性を明らかにする。

2. シミュレーテッドアニーリングプログラミング (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) とは、金属の焼き鈍しを模倣したメタヒューリスティック手法であるシミュレーテッドアニーリング (SA) を用いて、木構造により記述されたプログラムを自動生成することを本研究ではいう。SAP は GP における突然変異をベースに探索を行う。

以下に SAP のアルゴリズムを示す。

STEP 1 初期解候補の生成

初期解候補をランダムに生成し、その評価を行う。

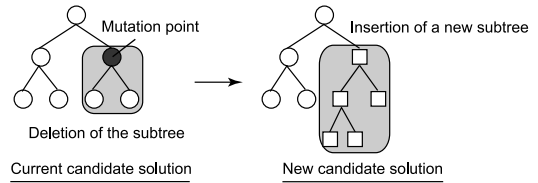


図 1 SAP における新しい解候補の生成方法

Fig. 1 Generation method of a new candidate solution in SAP.

STEP 2 生成処理

現在の解候補に対して GP の突然変異と同様の操作を行うことで新しい解候補を生成し、その解候補を評価する。具体的な生成方法は、現在の解候補に対してランダムに突然変異点を選択し、その点を根とする部分木を削除し、ランダムに生成した部分木を挿入する (図 1)。

STEP 3 受理判定, 状態遷移

現在の解候補の評価値 E と新しい解候補の評価値 E' との差分 $\Delta E (= E' - E)$ 、および温度パラメータ T により、新しい解候補に遷移するか否かの判定 (受理判定) を行う。受理判定には式 (1) に示す Metropolis 基準¹³⁾ 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} は受理確率であり、新しい解候補が改良方向へ生成された場合は無条件で受理し、改悪方向へ生成された場合でも確率的に受理する。改悪方向への受理確率 P_{AC} は、改悪幅 ΔE が小さいほど高く、また、温度パラメータ T が大きいほど高いという特徴を持つ。

STEP 4 冷却

STEP2, および 3 を所定の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。冷却後の温度 T_{k+1} は、式 (2) によって決定する¹⁷⁾。

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

ここで、 γ は冷却率であり、 T_k は現在の温度である。冷却を行うことにより、改悪方向への遷移確率が低くなる。

STEP 5 終了判定

STEP2~4 を定められた回数行えば、探索を終了する。

3. 温度パラメータが探索に及ぼす影響

3.1 温度パラメータの役割

SA の大きな特徴は、新しい解候補が改悪方向へ生

木構造の変化をできる限り小さくする突然変異
挿入木のサイズを削除木のサイズとほぼ同等にする突然変異
探索が進むにつれてプログラムサイズが増大することであり、探索の停滞や評価時間の増大をもたらす。

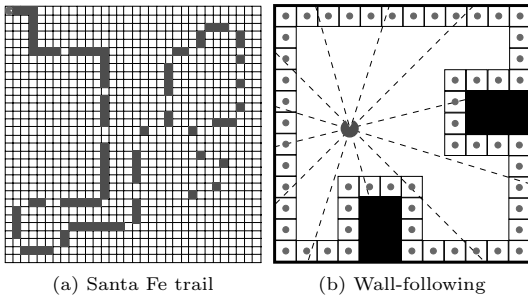


図 2 テスト問題のフィールド
Fig. 2 Fields of test problems.

成された場合でも、その解候補への遷移を確率的に認めることである。この確率は式 (1) の Metropolis 基準により決定され、その式から分かるように、改悪方向への遷移確率は温度パラメータ T によって制御することができる。この温度パラメータ T が大きい (高温) 場合、改悪方向への遷移を多く受理し、大域的な探索が行われる。一方、温度パラメータ T が小さい (低温) 場合、改悪方向への遷移をほとんど認めず、局所的な探索が行われる。

3.2 実験に用いるテスト問題

テスト問題は、GP の代表的なベンチマーク問題である Santa Fe trail 問題¹⁾、Wall-following 問題¹⁾、および複雑さが異なる 2 つの Symbolic Regression 問題 (Simple Symbolic Regression 問題、Complex Symbolic Regression 問題)¹⁾ とする。

Santa Fe trail 問題、および Wall-following 問題は構文的イントロンが発生する問題、すなわち、解を評価する際に実行されないノードが含まれる可能性のある問題である。また、Koza の文献 1) では、Santa Fe trail 問題は 18 ノードで、Wall-following 問題は 145 ノードで解かれていることから、Wall-following 問題は Santa Fe trail 問題より大規模な問題である。

一方、Symbolic Regression 問題は構文的イントロンが発生しない問題、すなわち、すべてのノードが解の評価に影響を及ぼす問題である。

以下に、これらの問題について説明する。

3.2.1 Santa Fe trail

Santa Fe trail 問題とは、人工蟻 1 匹が図 2 (a) に示す 32×32 のマス目に配置された餌を、限られたエネルギー内でできるだけ多く獲得するプログラムを生成する問題である¹⁾。

非終端記号は {IF-FOOD-AHEAD, PROGNN2, PROGNN3}、終端記号は {LEFT, RIGHT, MOVE} である。IF-FOOD-AHEAD は引数を 2 つ持ち、人工蟻の一マス前方に餌があれば第 1 引数を、なければ第

2 引数を実行する。PROGNN は引数を N 個持ち、第 1 引数、第 2 引数、 \dots 、第 N 引数の順に実行する。この問題では、IF-FOOD-AHEAD の連鎖により、構文的イントロンが発生する。

評価関数 E_{val} 式 (3) は、餌の総数である F_{max} ($= 89$) から人工蟻が獲得した餌の数 F を引いたものであり、すべてのエサを獲得すれば ($E_{val}=0$)、探索が成功したとする最小化問題である。

$$E_{val} = F_{max} - F \quad (3)$$

3.2.2 Wall-following

Wall-following 問題¹⁾ とは、ロボットが不規則な壁に囲まれた部屋で、壁に沿って移動するプログラムを生成する問題である¹⁾。部屋の環境は図 2 (b) に示す Koza の文献 1) の環境と同様である。

非終端記号は {IFLTE, PROGNN2}、終端記号は {S00, S01, S02, S03, \dots , S11, MSD, EDG, SS, MF, MB, TR, TL} である。IFLTE は引数を 4 つ持ち、第 1 引数の値が第 2 引数の値以下ならば第 3 引数を実行し、そうでなければ第 4 引数を実行する。S00 ~ S11 は各センサと壁との距離であり、SS はその中の最小値である。MSD、および EDG は安全距離を表しており、それぞれ 2.0、2.3 の定数である。MF は前進、MB は後退、TR は右旋回、TL は左旋回である。この問題では IFLTE と終端記号の組合せにより、構文的イントロンが発生する。

評価関数 E_{val} 式 (4) は、壁際に設置されたタイルの枚数である N_{max} ($= 56$) からロボットが通過した壁際のタイルの枚数 N を引いたものであり、すべてのタイルを通過すれば ($E_{val}=0$)、探索が成功したとする最小化問題である。

$$E_{val} = N_{max} - N \quad (4)$$

3.2.3 Simple Symbolic Regression

Symbolic Regression 問題とは、 n 組の入出力データから未知の関数 f_{obj} を同定する問題である¹⁾。

Simple Symbolic Regression 問題は、その中でも式 (5) に示す単純な関数 f_{obj} を同定する問題である¹⁾。その関数 f_{obj} の概形を図 3 (a) に示す。

$$f_{obj}(x) = x^4 + x^3 + x^2 + x \quad (5)$$

非終端記号は {+, \times , -, %, sin, cos, exp, rlog}、終端記号は { x } である。

評価関数 E_{val} 式 (6) は、 -1 から 1 の間を 0.1 刻みにした 21 個の入力 x_i に対する出力誤差の絶対値の総和であり、その総和が 0.01 以下 ($E_{val} \leq 0.01$) の場合、探索が成功したとする最小化問題である。prog は生成されたプログラムである。

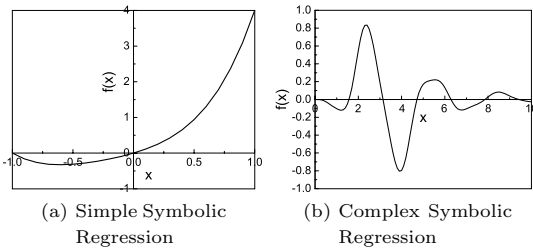


図 3 同定する関数の概形
Fig. 3 Target functions.

$$E_{val} = \sum_{i=0}^{20} |prog(x_i) - f_{obj}(x_i)| \quad (6)$$

3.2.4 Complex Symbolic Regression

Complex Symbolic Regression 問題は、Symbolic Regression 問題の中でも式 (7) に示す複雑な関数 f_{obj} を同定する問題である¹⁸⁾。その関数 f_{obj} の概形を図 3 (b) に示す。

$$f_{obj}(x) = x^3 \cos(x) \sin(x) e^{-x} (\sin^2(x) \cos(x) - 1) \quad (7)$$

非終端記号は $\{+, \times, -, \%, \sin, \cos, \exp, rlog\}$ 、終端記号は $\{x, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ である。

評価関数 E_{val} 式 (8) は、0 から 10 の間を 0.1 刻みにした 101 個の入力 x_i に対する出力誤差の絶対値の総和であり、その総和が 2 以下の場合 ($E_{val} \leq 2$)、探索が成功したとする最小化問題である。

$$E_{val} = \sum_{i=0}^{100} |prog(x_i) - f_{obj}(x_i)| \quad (8)$$

3.3 温度パラメータが探索に及ぼす影響に関する実験概要

SAP の温度パラメータが探索に及ぼす影響について検討するために、一定温度の温度スケジュール^{19),20)}を用いて数値実験を行う。

Santa Fe trail 問題、および Wall-following 問題では温度 T を 2^{-6} から 2^6 までの間を等比的に分割した 13 温度に対して、Simple Symbolic Regression 問題、および Complex Symbolic Regression 問題では 2^{-12} から 2^6 までの間を等比的に分割した 19 温度に対して、それぞれ SAP による探索を 50 試行行う。評価計算回数は Santa Fe trail 問題、Wall-following 問題、および Complex Symbolic Regression 問題は 40 万回であり、Simple Symbolic Regression 問題は 10 万回である。

新しい解候補を生成する際に挿入する部分木の生成

方法は、GP の突然変異と同じ方法 (最大の深さを 4、非終端記号の選択率を 0.5) を用いる。この方法では、Santa Fe trail 問題では平均して深さ 1.5 の部分木が、Wall-following 問題では平均して深さ 1.3 の部分木が、Simple Symbolic Regression 問題、および Complex Symbolic Regression 問題では平均して深さ 1.7 の部分木が生成される。

3.4 温度パラメータが解に及ぼす影響

SAP の温度パラメータが解に及ぼす影響を図 4 に示す。図 4 の横軸は温度、左軸は 50 試行の最良解の平均評価値、右軸は 50 試行の最良解の平均プログラムサイズ (ノード数) であり、評価値は 0 に近いほど良い値を示している。なお、探索中の最良解については、評価値が同等の場合はプログラムサイズがより小さい方の解を最良解としている。

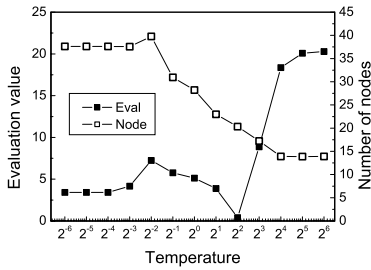
図 4 より、温度パラメータは解の性能だけでなく、プログラムサイズにも影響を及ぼしていることが分かる。解の性能は、一定温度での探索により良好な解が得られる温度領域 (Santa Fe trail 問題では $T = 2^2$ 付近、Wall-following 問題では $T = 2^1$ 付近、Simple Symbolic Regression 問題では $T = 2^{-1}$ 付近、Complex Symbolic Regression 問題では $T = 2^{-2}$ 付近) が存在していることが分かる (以下、このような温度領域を重要温度領域^{19),20)} と呼ぶ)。また、プログラムサイズは、温度が低いほど大きいことが分かる。

3.5 温度パラメータがプログラムサイズの変化に及ぼす影響

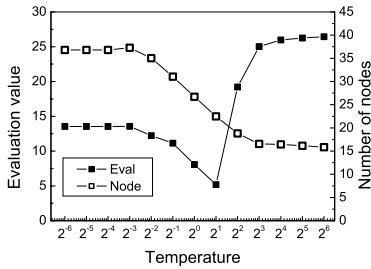
プロートとは、探索が進むにつれてプログラムサイズが増大する現象のことである²¹⁾。プロートの発生は、探索の停滞や探索時間、およびメモリ消費量の増大をもたらす²¹⁾。これはプログラムサイズが大きくなると、探索空間が広がることやプログラムの評価時間が長くなるからである。そのため、プロートは GP の最大の問題点とされており、新しい自動プログラミング手法を開発する場合、プロートについての議論を行う必要があると考える。

そこで、SAP の温度パラメータがプログラムサイズの変化に及ぼす影響について検討する。3.3 節の実験に対して、100 アニーリング期間中の平均プログラムサイズを算出し、それをプロットしたものを図 5 に示す。図 5 の横軸は評価計算回数、縦軸はプログラムサイズである。図 5 より、温度が低いほどプログラムサイズが大きく、広い探索空間で探索が行われていることが分かる。特に、Wall-following 問題、

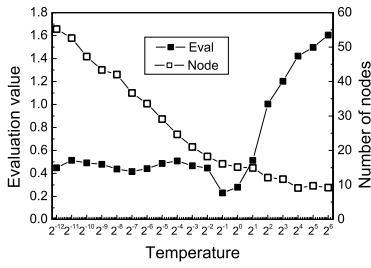
ルートノードの深さを 1 としている。



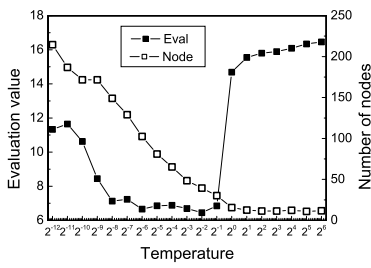
(a) Santa Fe trail



(b) Wall-following



(c) Simple Symbolic Regression

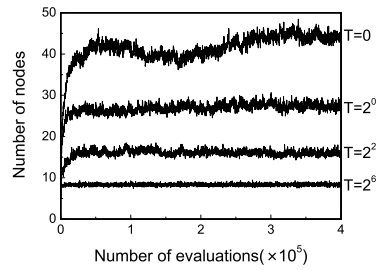


(d) Complex Symbolic Regression

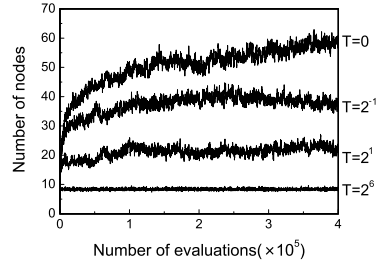
図 4 温度が解に及ぼす影響

Fig. 4 Effect of the temperature on solution.

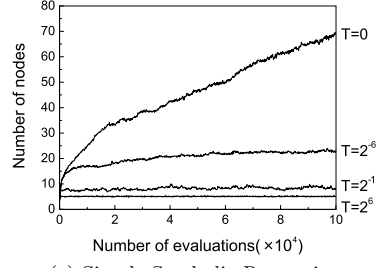
Simple Symbolic Regression 問題, および Complex Symbolic Regression 問題では, 改悪方向への遷移を認めない $T = 0$ の場合, 探索が進むにつれてプログラムサイズが大きくなっており, プロートが生じていることが分かる. 一方, 重要温度領域の温度 (Santa Fe trail 問題では $T = 2^2$ 付近, Wall-following 問題では $T = 2^1$ 付近, Simple Symbolic Regression 問題では $T = 2^{-1}$ 付近, Complex Symbolic Regression 問題では $T = 2^{-2}$ 付近) では, プロートが生じていないことが分かる.



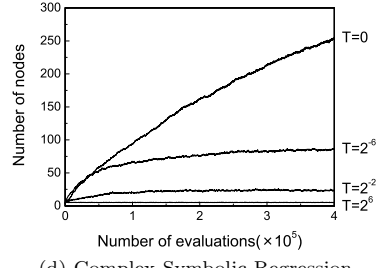
(a) Santa Fe trail



(b) Wall-following



(c) Simple Symbolic Regression



(d) Complex Symbolic Regression

図 5 温度がプログラムサイズの変化に及ぼす影響
Fig. 5 Effect of the temperature on the bloat.

3.6 温度パラメータに関する考察

すべての問題において, 温度が低いほどプログラムサイズが大きく, 広い探索空間で探索が行われた. ここでは, 生成処理のオペレーション (部分木の交換) が行われるノードの深さに着目して, このことについて考察する.

SAP では, 図 6 に示すように, オペレーションが行われるノードがルートノードに近い場合, 新しく生成される解候補のプログラムサイズは比較的小さくなる (図 6 は「木全体の深さに対するオペレーションが

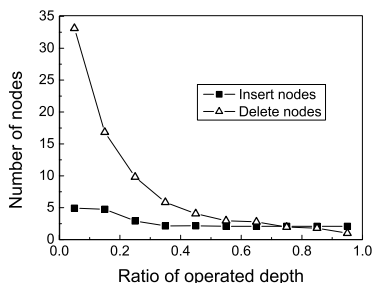


図 6 オペレーション時に挿入するノード数と削除するノード数 (Santa Fe trail)

Fig. 6 Number of insert and delete nodes during operation.

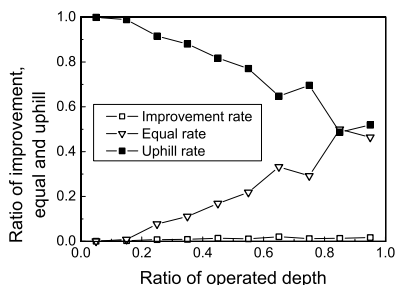


図 7 新しい解候補が改良される割合、改悪される割合、同等の割合 (Santa Fe trail)

Fig. 7 Improvement rate, uphill rate, and equal rate.

行われたノードの深さの比率(オペレーションが行われたノードの深さ/木全体の深さ)とそのときに「挿入されたノード数、および削除されたノード数」の関係を示しており、横軸は0に近いほどルートノード付近でオペレーションが行われたこと表している)。また、図7に示すように、オペレーションが行われるノードがルートノードに近い場合、これまでに評価をした多くのノードが削除される代わりに、評価をしていないランダムに生成したノードが挿入されるため、評価が悪くなる可能性が高い(図7は「木全体の深さに対するオペレーションが行われたノードの深さの比率」とそのときに生成された解候補が「改良された割合、改悪された割合、同等の割合」の関係を示している)。

よって、温度が高いほどルートノード付近でオペレーションが行われた解候補のような悪い解候補でさえ受理することが多くなり、高温ではプログラムサイズが小さくなったと考えられる。一方、温度が低いほどそのような解候補を受理しにくいために、低温ではプログラムサイズが大きくなったと考えられる。

また、「適応度の悪化」と「プログラムサイズの減少」の関係について検討する。その結果を図8に示す。図8の横軸は生成処理が行われたときの適応度の

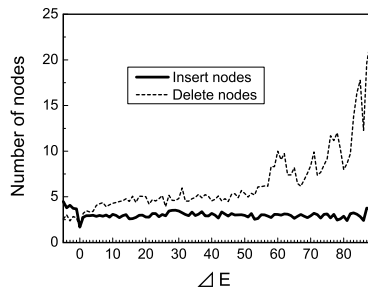


図 8 適応度の悪化とプログラムサイズの増減の関係 (Santa Fe trail)

Fig. 8 Relations of deterioration of fitness and program size.

差分であり、右へいくほど適応度の悪化を表している。図8より、評価が悪くなる解候補が生成されると、プログラムサイズは平均すると小さくなっていることが分かる。また、適応度が悪化するにつれてプログラムサイズの減少量は大きくなっていることが分かる。以上より、評価が悪くなる解候補に多く遷移することでプログラムサイズが小さくなると考えられる。

4. 温度スケジュールが探索に及ぼす影響

4.1 温度スケジュールとは

一般的にSAでは、温度スケジュールに冷却を用いる。冷却を用いる温度スケジュールでは、温度を高温から徐々に低温へ下げていくことで、大域的な探索から局所的な探索へと徐々にシフトしていく。その初期温度、すなわち最高温度はすべての局所解を抜け出すような十分高い温度に設定し、探索終了時の温度、すなわち最低温度は解が収束するような低い温度に設定する^{22),23)}。これらは、最高温度が低すぎると局所解からの脱出が困難となることや、最低温度が高すぎると局所的探索が十分に行われず最適解に到達することが困難になるからである。

一方、一定温度での探索によって良好な解が得られる温度領域が存在し、その温度スケジュールで得られる解が一般的な冷却を用いる場合より良好であることも報告されている^{19),20)}。

4.2 温度スケジュールが探索に及ぼす影響に関する実験概要

SAPの温度スケジュールが探索に及ぼす影響について検討するために、一定温度の温度スケジュールを用いるSAP(以下、一定温度SAP)と冷却を用いるSAP(以下、冷却SAP)の比較を行う。

一定温度SAPで用いるパラメータを表1に、冷却SAPで用いるパラメータを表2に示す。ここで T_{fixed} は一定温度、 T_{max} は最高温度、 T_{min} は最低

表 1 一定温度 SAP のパラメータ
Table 1 Parameter of SAP with fixed temperature.

	Santa Fe trail	Wall-following	Simple Symbolic Regression	Complex Symbolic Regression
T_{fixed}	4	2	0.5	0.25

表 2 冷却 SAP のパラメータ
Table 2 Parameter of SAP with cooling.

	Santa Fe trail	Wall-following	Simple Symbolic Regression	Complex Symbolic Regression
T_{max}	35	16	4	7
T_{min}	0.25	0.25	2^{-12}	2^{-12}
N	40	40	10	40

温度, N は冷却回数である. T_{fixed} は 3.3 節の実験より求めた最良値であり, T_{max} , および T_{min} はそれぞれ, 最大の改善を 50% で受理する温度, 最小の改善を冷却期間中に 1 回は受理するような温度である²²⁾. 評価計算回数, 試行数, および新しい解候補を生成する際に挿入する部分木の生成方法は 3.3 節の実験と同様である.

4.3 温度スケジュールが解探索性能に及ぼす影響
解探索性能の結果を図 9 に示す. 図 9 の横軸は評価計算回数, 縦軸は 50 試行の最良解の平均評価値であり, 評価値は 0 に近いほど良好な値を示している.

図 9 より, 探索終了時における最良解の平均評価値を比較すると, 一定温度 SAP と冷却 SAP はほぼ同等であることが分かる.

4.4 温度スケジュールがプログラムサイズの変化に及ぼす影響

SAP の温度スケジュールがプログラムサイズの変化 (プロットの有無) に及ぼす影響を図 10 に示す. 図 10 の横軸は評価計算回数, 縦軸は 100 アニリング期間中の平均プログラムサイズを算出し, それをプロットしたものである.

図 10 より, 冷却 SAP は, 探索中盤以降ではプログラムサイズが増大しており, プロットが生じていることが分かる. 一方, 一定温度 SAP は, ある一定のプログラムサイズに収束しており, プロットが生じていないことが分かる.

4.5 温度スケジュールに関する考察

4.5.1 探索の特徴

最良解の平均評価値で比較を行うと, 一定温度 SAP と冷却 SAP はほぼ同等であった. ここでは, 一定温度 SAP と冷却 SAP の探索の特徴について探索履歴, および得られた解の評価値を基に考察する.

Santa Fe trail 問題において, 一定温度 SAP, および冷却 SAP で探索したときのある 1 試行の評価値の履歴を図 11 に示す. 図 11 より, 一定温度 SAP では改善をつねに一定の割合で認めながら探索を行っている

ことが分かる. 一方, 冷却 SAP では探索序盤は改善を多く認め, 探索終盤はほとんど改善を認めずに探索が行われており, 大域的な探索から局所的な探索へと徐々にシフトしていることが分かる.

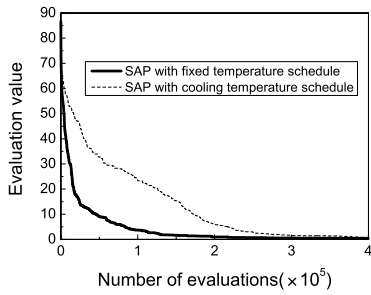
次に, Santa Fe trail 問題に対して, それぞれの温度スケジュールにおいて各試行で得られた最良解の評価値を良い順にソートしたものを図 12 に示す. 図 12 より, 最適解 (評価値 E_{val} が 0 の解) を得た試行数は, 一定温度 SAP では 39 試行であるのに対し, 冷却 SAP では 45 試行であることから, 最適解を得た試行数で比較を行うと冷却 SAP の方が良好な結果となる. これは, 冷却 SAP では探索終盤で温度が低温になり, 一定温度 SAP に比べ局所的探索が十分に行われるためだと考えられる. ただし, 図 12 より, 冷却 SAP では評価が大きく悪い解を得ている試行があるが, 一定温度 SAP では安定して良好な解を得ていることが分かる. これは, 重要温度での一定温度 SAP では局所解を比較的容易に抜け出すことができるのに対し, 冷却 SAP では図 13 に示すように重要温度領域での探索において良好な解を得ることができなかった場合に, 局所解から抜け出しにくいためだと考えられる.

よって, 冷却 SAP は局所的探索も十分に行うことができるため最適解を得る割合は多くなるが, 試行によって良好な解を得ることができないリスクもあることが分かる. 一方, 一定温度 SAP はすべての探索期間において局所解を比較的容易に抜け出すことができるが, 局所的探索が十分行われなことがある.

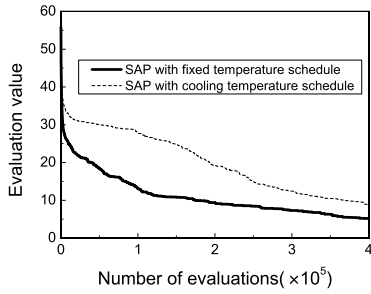
4.5.2 プログラムサイズの変化

すべての問題において, 冷却 SAP では探索中盤以降, プログラムサイズが増大しておりプロットが生じたのに対し, 一定温度 SAP ではプロットが生じなかった.

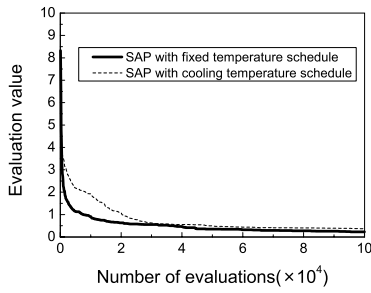
これは, 3.4 節の実験結果より, 温度が低いほどプログラムサイズが大きくなることから, 探索が進むにつれて温度が低くなる冷却 SAP ではプロットが生じた



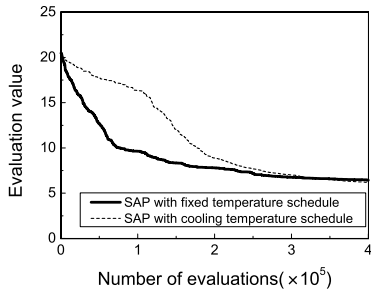
(a) Santa Fe trail



(b) Wall-following



(c) Simple Symbolic Regression

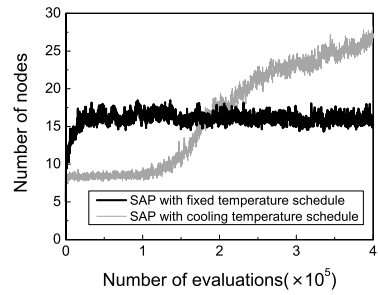


(d) Complex Symbolic Regression

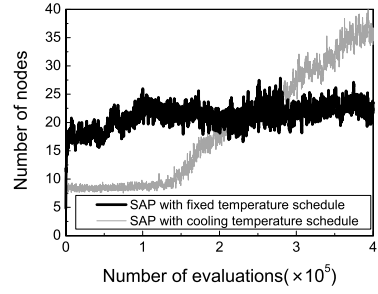
図 9 温度スケジュールが解探索性能に及ぼす影響

Fig. 9 Effect of the temperature schedule on search performance.

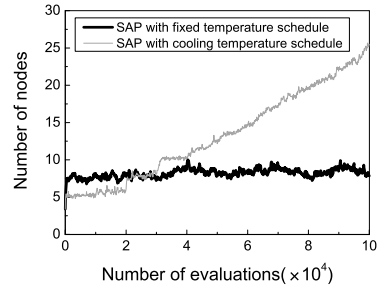
と考えられる。一方、重要温度領域での一定温度 SAP では、 ΔE が同じであれば一定の割合で改悪を認めながら探索するため、ルートノード付近でオペレーションが行われたプログラムサイズが小さくなる解候補にもつねにある一定の確率で遷移することで、プロートが生じなかったと考えられる。



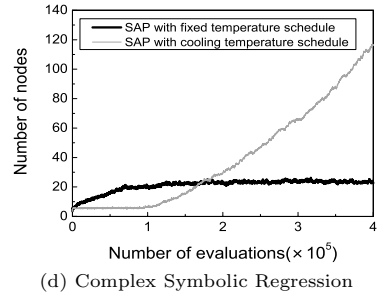
(a) Santa Fe trail



(b) Wall-following



(c) Simple Symbolic Regression



(d) Complex Symbolic Regression

図 10 温度スケジュールがプログラムサイズの変化に及ぼす影響

Fig. 10 Effect of the temperature schedule on the bloat.

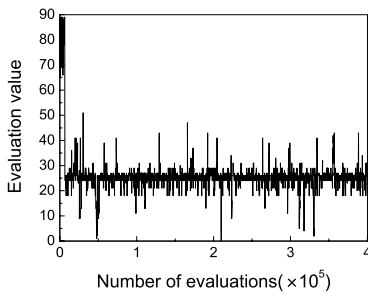
5. SAP の有効性の検討 (GP との比較)

5.1 SAP の有効性に関する実験概要

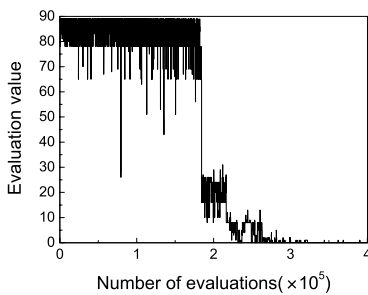
標準的な GP、およびプロートが生じないとされる GP との比較を行い、SAP の有効性を検討する。SAP の温度スケジュールは、4 章の実験でプロートが生じなかった一定温度の温度スケジュールを用いる。プロートが生じないとされる GP には、Poli らの一樣交叉や

表 3 GP のパラメータ
Table 3 Parameter of GP.

	Santa Fe trail	Wall-following	Simple Symbolic Regression	Complex Symbolic Regression
M	2000	2000	2000	2000
P_c	0.9	0.9	0.9	0.9
P_m	0.1	0.1	0.1	0.1
K	5	3	2	7
E	1	1	1	1
D_{max}	17	17	17	17



(a) SAP_{fixed}



(b) SAP_{cooling}

図 11 ある 1 試行の一定温度 SAP と冷却 SAP の探索履歴
Fig. 11 Search histories of SAP with fixed temperature and with cooling.

一点交叉²⁴⁾, Langdon の相同性交叉やサイズ平等交叉²⁵⁾, Crawford-Marks らの平等交叉²⁶⁾ など様々な手法が提案されているが, ここでは数多くの研究で用いられている「適合度計算にプログラムサイズを反映される GP」を用いる. 以後, 適合度計算にプログラムサイズを反映される GP を GP-penalty と呼ぶ.

実験に用いる SAP のパラメータを表 1 に, それぞれの GP のパラメータを表 3 に示す. ここで, M は個体数, P_c は交叉率, P_m は突然変異率, K はトーナメントサイズ, E はエリート個体数, D_{max} は最大の深さである. T_{fixed} , および K は予備実験より求めた値であり, M, P_c, P_m , および D_{max} は一般的に用いられている値^{1),21)} である. GP-penalty では,

一定温度の最良値は問題ごとに異なるので, 図 4 のような予備実験は避けられない

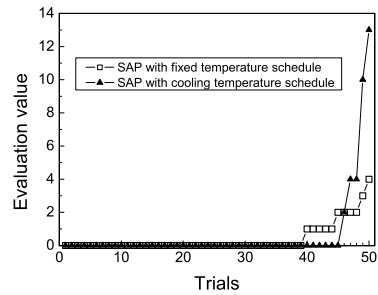


図 12 最良解の評価値 (Santa Fe trail)
Fig. 12 Evaluation value of best solutions.

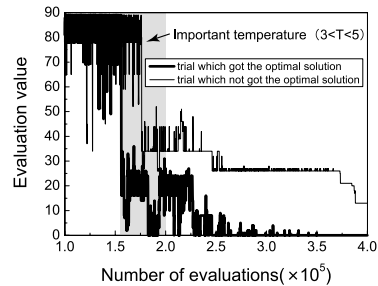


図 13 最適解を得た試行と得られなかった試行の探索履歴 (Santa Fe trail)
Fig. 13 Search histories of SAP with cooling having obtained the optimal solution and not having obtained it.

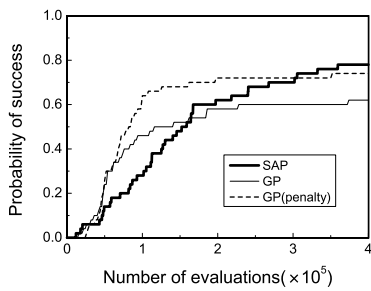
すべての問題において, 評価関数 E_{val} にプログラムサイズ $\times 0.001$ のペナルティを与える.

評価計算回数, 試行数, および新しい解候補を生成する際に挿入する部分木の生成方法は 3.3 節の実験と同様である.

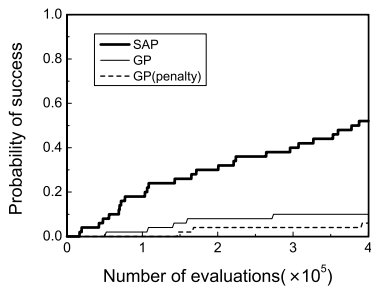
5.2 SAP の解探索性能

実験により得られた探索成功率 (50 試行において最適解を得た割合) を図 14 に示す. Santa Fe trail 問題, および Wall-following 問題の最適解は, 評価値 E_{val} が 0 のプログラムである. Simple Symbolic Regression 問題の最適解は評価値 E_{val} が 0.01 以下のプログラム, Complex Symbolic Regression 問題の最適解は評価値 E_{val} が 2 以下のプログラムである.

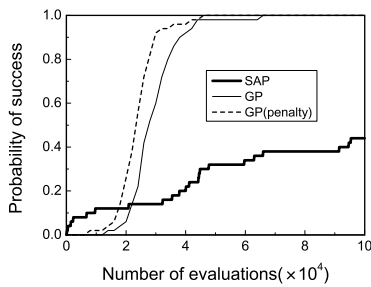
図 14 より, 構文的イントロンが発生する Santa Fe



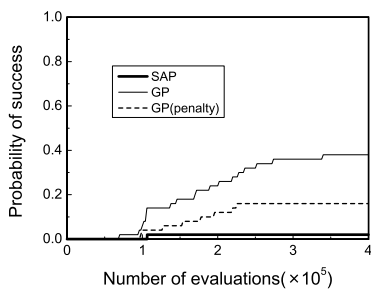
(a) Santa Fe trail



(b) Wall-following



(c) Simple Symbolic Regression

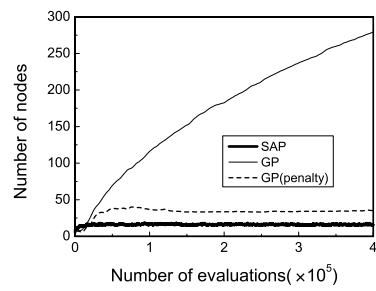


(d) Complex Symbolic Regression

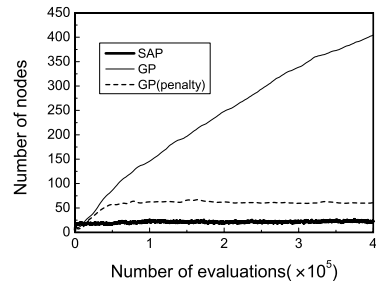
図 14 探索成功率

Fig. 14 Probability of success.

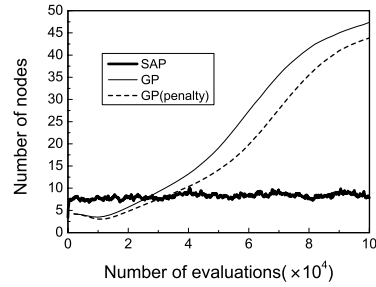
trail 問題, および Wall-following 問題では, 探索終了時における SAP の探索成功率はそれぞれ 0.78, 0.52 であるのに対して, 標準的な GP は 0.62, 0.10, GP-penalty は 0.74, 0.06 である. したがって, これらの問題では, SAP は良好な探索ができているといえる. 一方, 構文的イントロンが発生しない Simple Symbolic Regression 問題, および Complex Symbolic Regression 問題では, SAP の探索成功率はそれぞれ 0.44,



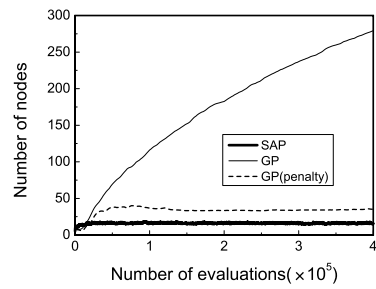
(a) Santa Fe trail



(b) Wall-following



(c) Simple Symbolic Regression



(d) Complex Symbolic Regression

図 15 プログラムサイズの変化

Fig. 15 Impact of bloat.

0.02 であるのに対して, 標準的な GP は 1.00, 0.38, GP-penalty は 1.00, 0.16 である. したがって, これらの問題では GP の方が良好な探索ができているといえる.

5.3 SAP のプログラムサイズの変化

SAP, および GP の探索におけるプログラムサイズの変化 (プロートの有無) について検討するために, 探索中のプログラムサイズを比較する. GP の母集団

表 4 最良解の平均プログラムサイズ

Table 4 Average program size of the best solution.

	Santa Fe trail	Wall-following	Simple Symbolic Regression	Complex Symbolic Regression
SAP_{fixed}	20.3	22.5	16.1	39.7
GP	48.3	97.4	13.1	167.8
$GP_{penalty}$	28.9	36.2	13.0	70.0

内の全個体の平均プログラムサイズの履歴と、SAP の 100 アニーリング期間中の平均プログラムサイズの履歴を図 15 に示す。

図 15 より、すべての問題において、標準的な GP では探索が進むにつれてプログラムサイズが増大しており、プロートが生じていることが分かる。一方、SAP は GP-penalty と同様に、ある一定のプログラムサイズに収束しており、プロートが生じていないことが分かる。

5.3.1 解のプログラムサイズの比較

プロートが生じると、最終的に得られるプログラムはサイズが大きくなり、冗長な表現が多く含まれると考えられる。そこで、SAP、および GP の探索で最終的に得られたプログラムのサイズの平均を表 4 に示す。ただし、評価値が同等の場合は、プログラムサイズが小さい方の解を最良解としている。

表 4 より、GP においてすべての試行で最適解を得た Simple Symbolic Regression 問題を除き、SAP は GP-penalty と同様に、標準的な GP よりコンパクトなプログラムを生成していることが分かる。特に、SAP が良好な探索をした Santa Fe trail 問題、および Wall-following 問題においては、SAP が生成したプログラムは標準的な GP が生成したプログラムの半分以下のサイズである。ただし、Simple Symbolic Regression 問題においては、GP の方がプログラムサイズが小さい。これは、この問題では評価計算回数 7 万回の時点で GP のすべての試行において最適値を示す解が得られており、残りの約 3 万回の評価計算の間で、より小さなプログラムサイズの解が探索されたためであると考えられる。

5.3.2 評価計算時間の比較

プロートが生じると、探索時間が増大するとされている²¹⁾。これは、プログラムサイズの大きいプログラムほど、出力を得るまでに多くの時間を要するからである。たとえば、Symbolic Regression 問題のような関数木では、プログラムサイズが大きいほど多くの演算処理が必要となり、出力を得るまでに多くの時間を要する。

そこで、SAP が良好な探索をした Santa Fe trail 問題において、1 試行 (40 万回の評価計算) に要する

表 5 実験環境

Table 5 Execution environment.

Processor	Intel(R) Pentium(R)4 CPU 2.80[GHz]
Memory	512[MB]
Compiler	java 1.5.0

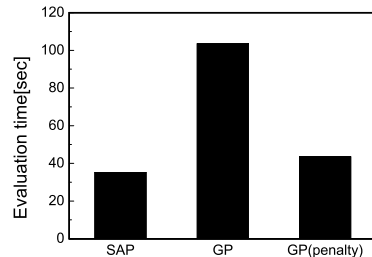


図 16 平均評価計算時間 (Santa Fe trail)
Fig. 16 Average evaluation time.

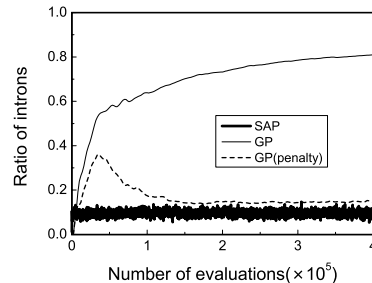


図 17 プログラムに占める構文的イントロンの割合 (Santa Fe trail)
Fig. 17 Ratio of syntactic introns in program.

評価計算時間を実際に測定した。ここでの評価計算時間とは、プログラムの評価計算時間のことであり、新しい解候補の生成や受理判定、交叉などの処理時間は含んでいない。実験に用いた計算機の環境は表 5 に示すとおりである。結果を図 16 に示す。図 16 より、SAP では GP-penalty と同様に、標準的な GP の約 3 分の 1 の時間しかかかっていないことが分かる。

5.4 SAP の探索に関する考察

構文的イントロンが発生する Santa Fe trail 問題、および Wall-following 問題では、SAP は良好な探索をすることができた。ここでは、構文的イントロンに着目して、その理由について考察する。

構文的イントロンでのオペレーションは、評価にまったく影響を及ぼさない。そのため、プログラムに占め

る構文的イントロンの割合が高くなるほど、そのようなオペレーションが行われる回数が増え、探索が停滞する可能性が高くなる。

そこで、Santa Fe trail 問題において、SAP、および GP の探索中のプログラムに占める構文的イントロンの割合を検討した。その結果を図 17 に示す。図 17 の横軸は評価計算回数であり、縦軸はプログラムに占める構文的イントロンの割合である。図 17 より、標準的な GP では探索が進むにつれてプログラムに占める構文的イントロンの割合が高くなっていることが分かる。一方、SAP では GP-penalty と同様に、ある一定の割合に収束していることが分かり、標準的な GP と比べその割合は約 8 分の 1 と小さい。

よって、SAP は標準的な GP と比べプログラムに占める構文的イントロンの割合が少なく、探索に無駄な構文的イントロンでのオペレーションも少なかったと考えられる。したがって、構文的イントロンが発生する Santa Fe trail 問題、および Wall-following 問題では、SAP は良好な探索をすることができたと考えられる。

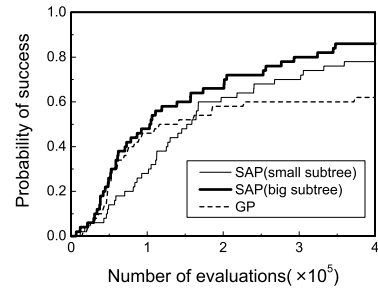
以上より、構文的イントロンが発生し、かつ Santa Fe trail 問題、および Wall-following 問題と同等の複雑さの問題においては、少なくとも SAP は GP と同等の解探索性能であることが考えられる。

6. 部分木のサイズが探索に及ぼす影響

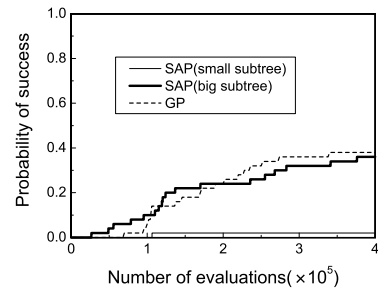
6.1 部分木のサイズが探索に及ぼす影響に関する実験概要

5 章までの実験では、新しい解候補を生成する際に挿入する部分木の生成方法を「最大の深さを 4、非終端記号の選択率を 0.5」としていた。ここでは、その部分木の生成方法を「深さ 1~4 の木を均等に生成する」とし、大きなサイズの部分木を挿入しても適切な一定温度の温度スケジュールを用いる SAP はプロットが生じず、かつ良好な解を得ることができるのかを検討する。なお、5 章までの実験で用いていた前者の生成方法は、問題で扱う非終端記号の引数の数がすべて 2 なら平均して深さ約 1.4 の小さな部分木を生成するため、ここでは小部分木法 (*small subtree*) と呼ぶ。一方、後者の方法は非終端記号の引数の数に関係なく平均して深さ 2.5 の大きな部分木を生成するため、ここでは大部分木法 (*big subtree*) と呼ぶ。

テスト問題は、大きなプログラムサイズを必ずしも必要としない Santa Fe trail 問題、および大きなプログラムサイズが本質的に必要な Complex Symbolic Regression 問題とする。用いたパラメータは 5.1 節



(a) Santa Fe trail



(b) Complex Symbolic Regression

図 18 部分木のサイズが及ぼす探索性能の影響

Fig. 18 Effect of the subtree size on success rate.

の実験と同様である。

6.2 部分木のサイズが解探索性能に及ぼす影響

小部分木法を用いた SAP、大部分木法を用いた SAP、および標準的な GP で探索を行ったときの探索成功率を図 18 に示す。

図 18 より、大部分木法を用いた SAP は、小部分木法を用いた SAP と比べ、良好な探索をしていることが分かる。特に、Complex Symbolic Regression 問題においては、小部分木法を用いた場合は SAP は GP に比べ大幅に劣っていたが、大部分木法を用いることで GP とほぼ同等の性能を得ていることが分かる。

以上より、SAP では部分木の生成方法を検討することで、さらなる探索性能の向上が期待できると考えられる。

6.3 部分木のサイズがプログラムサイズの変化に及ぼす影響

図 19 に部分木の生成方法がプログラムサイズの変化に及ぼす影響を示す。図 19 の横軸は評価計算回数であり、縦軸は GP では母集団内の全個体の平均プログラムサイズの履歴、SAP では 100 アニーリング期間中の平均プログラムサイズを算出し、それをプロットしたものである。

図 19 より、部分木のサイズを大きくすることで、SAP はプログラムサイズが大きくなり、広い探索空間で探索が行われていることが分かる。しかし、部分木のサイズを大きくしても、適切な一定温度の温度ス

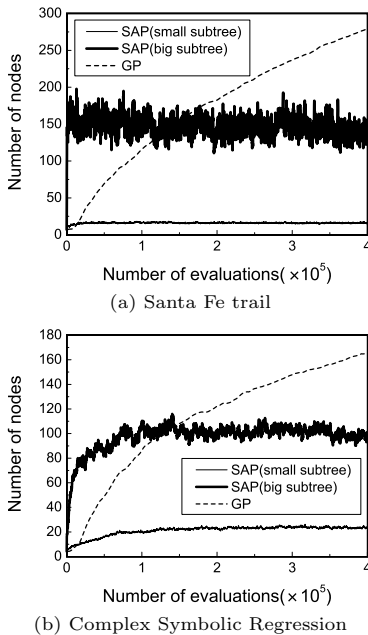


図 19 部分木のサイズが及ぼすプログラムサイズの変化の影響
Fig. 19 Effect of the subtree size on bloat.

表 6 部分木のサイズが及ぼす最良解の平均プログラムサイズ
Table 6 Effect of the subtree size on program size.

	Santa Fe trail	Symbolic Regression
$SAP_{small\ subtree}$	20.3	39.4
$SAP_{big\ subtree}$	29.9	134.1
GP	48.3	167.8

ケジュールを用いる SAP は、プロートが生じていないことが分かる。

また、得られた最良解の平均プログラムサイズを表 6 に示す。表 6 より、プログラムサイズが大きくなりやすいと考えられる SAP においても、挿入する部分木のサイズを大きくすることで、大きなサイズのプログラムを得ることができることが分かる。

6.4 部分木のサイズに関する考察

適切な一定温度の温度スケジュールを用いる SAP は、サイズの大きい部分木を挿入しても、プロートが生じないことが分かった。これは、適切な一定温度の温度スケジュールを用いることで、プログラムサイズが小さくなるような、ルートノード付近でオペレーションが行われた解候補にも、つねにある一定の確率で遷移するためだと考えられる。

以上より、どのような部分木を挿入しても、適切な一定温度の温度スケジュールを用いることで、SAP はプロートが生じないと考えられる。

7. ま と め

本論文では、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 (SAP) の有効性を検討した。

数値実験の結果、適切な一定温度の温度スケジュールを用いることで、SAP はプロートが発生しないことが分かった。これは、適切な一定温度を用いることで、挿入される小さな部分木が大きな部分木と入れ替わるような解候補にも、つねにある一定の割合で遷移するからである。また、構文的イントロンが発生する Santa Fe trail 問題、および Wall-following 問題においては、SAP は標準的な GP よりも良好な探索をすることができた。これは、プログラムサイズが小さくなるような解候補にもつねにある一定の割合で遷移するメカニズムにより、プログラムに占める構文的イントロンの割合が低くなり、探索性能が向上したためである。また、新しい解候補を生成する際に挿入する部分木の生成方法を検討することにより、さらなる SAP の探索性能の向上が期待できることが分かった。

以上より、今後とも SAP の研究をする価値があると考えられ、メタヒューリスティック手法を用いた自動プログラミング分野における探索手法の選択肢を広げることができたと考えられる。

今後は、適切な一定温度の自動温度チューニングの検討や、オペレーション方法の検討 (挿入する部分木の生成方法、突然変異点の選択方法など)、GP とのハイブリッド探索手法の開発などを検討し、SAP のさらなる可能性、およびメタヒューリスティック手法を用いた自動プログラミング分野における探索効率性について探究していく予定である。

参 考 文 献

- 1) Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press (1992).
- 2) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional (1989).
- 3) Nordin, P. and Banzhaf, W.: *Genetic Programming Controlling a Miniature Robot, Working Notes for the AAAI Symposium on Genetic Programming*, Siegel, E.V. and Koza, J.R. (Eds.), pp.61-67, MIT, Cambridge, MA, USA, AAAI (1995).
- 4) 片上大輔, 山田誠二: ノード使用頻度に依存した交叉による進化ロボティクス的高速化, 人工知能学会論文誌, Vol.16, No.4, pp.392-399 (2001).
- 5) Koza, J.R., Bennett III, F.H., Andre, D. and

- Keane, M.A.: Automated WYWIWYG Design of Both the Topology and Component Values of Electrical Circuits Using Genetic Programming, *Genetic Programming 1996: Proc. 1st Annual Conference*, Koza, J.R., Goldberg, D.E., Fogel, D.B. and Riolo, R.L. (Eds.), pp.123-131, Stanford University, CA, USA, MIT Press (1996).
- 6) Iba, H. and Nikolaev, N.: Genetic Programming Polynomial Models of Financial Data Series, *Proc.2000 Congress on Evolutionary Computation CEC00*, La Jolla Marriott Hotel La Jolla, California, USA, pp.1459-1466, IEEE Press (2000).
- 7) Johanson, B. and Poli, R.: GP-Music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters, *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.181-186, University of Wisconsin, Madison, Wisconsin, USA, Morgan Kaufmann (1998).
- 8) Daida, J.M., Bersano-Begey, T.F., Ross, S.J. and Vesecky, J.F.: Computer-Assisted Design of Image Classification Algorithms: Dynamic and Static Fitness Evaluations in a Scaffolded Genetic Programming Environment, *Genetic Programming 1996: Proc. 1st Annual Conference*, Koza, J.R., Goldberg, D.E., Fogel, D.B. and Riolo, R.L. (Eds.), pp.279-284, Stanford University, CA, USA, MIT Press (1996).
- 9) Jussi, J. and Juhani, K.: Heuristic Methods in Space Frame Optimization, *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Vol.2, pp.1214-1228 (2005).
- 10) Zeiri, Y.: Prediction of the lowest energy structure of clusters using a genetic algorithm, *Physical Review E*, Vol.51, No.4, pp. R2769-R2772 (1995).
- 11) 三木光範, 廣安知之, 市川親司, 真武信和: 最適化法の進展と多分岐統合問題への拡張実最適化問題における進化的アプローチの有効性—利得等化フィルタの最適設計, *日本航空宇宙学会誌*, Vol.52, No.608, pp.225-231 (2004).
- 12) Min, X. and M, A.E.: Terminal Area Trajectory Optimization using Simulated Annealing, *44th AIAA Aerospace Sciences Meeting*, Vol.23, pp.17651-17661 (2006).
- 13) Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equation of State Calculation by Fast Computing Machines, *Journ. of Chemical Physics*, Vol.21, pp.1087-1092 (1953).
- 14) O'Reilly, U.-M. and Oppacher, F.: Program Search with a Hierarchical Variable Length Representation: Genetic Programming, Simulated Annealing and Hill Climbing, *Parallel Problem Solving from Nature — PPSN III*, Davidor, Y., Schwefel, H.-P. and Manner, R. (Eds.), No.866, pp.397-406, Jerusalem, Springer-Verlag (1994).
- 15) O'Reilly, U.-M. and Oppacher, F.: Hybridized Crossover-Based Search Techniques for Program Discovery, *Proc. 1995 World Conference on Evolutionary Computation*, Vol.2, pp.573-578, Perth, Australia, IEEE Press (1995).
- 16) Langdon, W.B.: Fitness Causes Bloat: Simulated Annealing, Hill Climbing and Populations, Technical Report CSRP-97-22, University of Birmingham, School of Computer Science (1997).
- 17) Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P.: Optimization by Simulated Annealing, *Science*, 13 May 1983, Vol.220, No.4598, pp.671-680 (1983).
- 18) Salustowicz, R.P. and Schmidhuber, J.: Probabilistic Incremental Program Evolution: Stochastic Search Through Program Space, *Machine Learning: ECML-97*, van Someren, M. and Widmer, G. (Eds.), Vol.1224, pp.213-220, Springer-Verlag (1997).
- 19) Connolly, T.D.: An improved annealing scheme for the gap, *European Journal of Operational Research*, Vol.46, pp.93-100 (1990).
- 20) Fielding, M.: Simulated Annealing With An Optimal Fixed Temperature, *j-SIAM-J-OPT*, Vol.11, No.2, pp.289-307 (2000).
- 21) Banzhaf, W., Nordin, P., Keller, R.E. and Francone, F.D.: *Genetic Programming — An Introduction; On the Automatic Evolution of Computer Programs and its Applications*, Morgan Kaufmann, dpunkt.verlag (1998). 伊庭斉志, 新田 徹 (訳): 遺伝的プログラミング, 科学技術出版 (2000).
- 22) 小西健三, 滝 和男, 木村宏一: 温度並列シミュレーテッドアニーリング法とその評価, *情報処理学会論文誌*, Vol.36, No.4, pp.797-807 (1995).
- 23) 喜多 一: シミュレーテッドアニーリング, *日本ファジィ学会誌*, Vol.9, No.6, pp.875-880 (1997).
- 24) Poli, R. and Langdon, W.B.: On the Search Properties of Different Crossover Operators in Genetic Programming, *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.293-301, University of Wisconsin, Madison, Wisconsin, USA, Morgan Kaufmann (1998).

- 25) Langdon, W.B.: Size Fair and Homologous Tree Genetic Programming Crossovers, *Proc. Genetic and Evolutionary Computation Conference*, Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M. and Smith, R.E. (Eds.), Vol.2, pp.1092-1097, Orlando, Florida, USA, Morgan Kaufmann (1999).
- 26) Crawford-Marks, R. and Spector, L.: Size Control Via Size Fair Genetic Operators In The PushGP Genetic Programming System, *GECCO 2002: Proc. Genetic and Evolutionary Computation Conference*, Langdon, W.B., Cantú-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M.A., Schultz, A.C., Miller, J.F., Burke, E. and Jonoska, N. (Eds.), pp.733-739, New York, Morgan Kaufmann Publishers (2002).

(平成 19 年 1 月 26 日受付)

(平成 19 年 4 月 6 日再受付)

(平成 19 年 4 月 17 日採録)



藤田 佳久 (学生会員)

1983 年生 . 2005 年同志社大学工学部知識工学科卒業 . 2007 年同志社大学大学院工学研究科知識工学専攻博士前期課程修了 . 2007 年株式会社日立製作所に入社 . 自動プログラ

ミングやロボット制御の研究に従事 .



三木 光範 (正会員)

1950 年生 . 1978 年大阪市立大学大学院工学研究科博士課程修了 , 工学博士 . 大阪市立工業研究所研究員 , 金沢工業大学助教授を経て 1987 年大阪府立大学工学部航空宇宙工学科助教授 , 1994 年同志社大学工学部教授 . 進化的計算手法とその並列化 , および知的なシステム的设计に関する研究に従事 . 著書は『工学問題を解決する適応化・知能化・最適化』(技法堂出版) 等多数 . IEEE , 米国航空宇宙学会 , 人工知能学会 , 日本機械学会 , 計算工学会 , 日本航空宇宙学会等各会員 . 通産省産業技術審議会委員等歴任 . 超並列計算研究会代表 .



橋本 雅文

1981 年大阪府立大学大学院航空工学専攻修士課程修了 . 同年大阪府立大学工学部航空工学科助手 , 1989 年広島大学工学部機械系助教授を経て , 2004 年同志社大学工学部情報システムデザイン学科教授となり , 現在に至る . 工学博士 . ロボティクスやメカトロ情報システムにおけるセンシング , コントロール , 安全システムの研究に従事 . IEEE , 日本ロボット学会 , 計測自動制御学会等の会員 .



廣安 知之 (正会員)

1997 年早稲田大学大学院理工学研究科後期博士課程修了 . 早稲田大学理工学部助手を経て , 1998 年同志社大学工学部助手 . 現在 , 同大学工学部インテリジェント情報工学科准教授 . 進化的計算 , 最適設計 , 並列処理 , 設計工学 , 医療画像工学等の研究に従事 . IEEE , 電子情報通信学会 , 計測自動制御学会 , 日本機械学会 , 超並列計算研究会 , 日本計算工学会各会員 .