

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、シミュレーテッドアニーリング (SA) を木構造が扱えるように拡張したシミュレーテッドアニーリングプログラミング (SAP) を提案する。SAP は交叉を用いず、突然変異のみでプログラムを進化させる手法である。Santa Fe trail 問題、および Symbolic Regression 問題に一定温度の温度スケジュールを用いる SAP を適用した結果、構文的イントロンが発生する可能性がある Santa Fe trail 問題では、SAP は標準的な GP より良好な探索をすることができた。また、一定温度の温度スケジュールを用いる SAP はブloatが生じなかった。

Automatic Programming with Simulated Annealing

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

In this paper, we propose Simulated Annealing Programming (SAP), an extension of Simulated Annealing (SA) that allows SA to handle tree structures. SAP is a mutation-based program search method. SAP with fixed temperature schedules applied to the Santa Fe trail and Symbolic Regression problems. SAP obtained better solutions than standard Genetic Programming (GP) in Santa Fe trail problem, which have the possibility of syntactic introns. In addition, bloating does not occur in SAP with fixed temperature schedules.

1. はじめに

ロボットを制御するプログラムなどは、コンピュータを用いて自動設計する場合がある。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設計できることや、複数台のロボットが協調行動するような複雑なプログラムを容易に設計することができるからである。

コンピュータを用いてこのようなプログラムを自動設計する (本研究では「自動プログラミング」と呼ぶ) 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) 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る。これまでに GP は、ロボットの行動ルールや関数、電子回路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や電子回路の設計、株価予測、音楽の自動作曲など様々な分野に応用されている²⁾。

このように自動プログラミングには、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、問題に適した探索手法が用いられている。例えば、立体骨組の最適化³⁾ 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⁴⁾ などにはシミュレーテッドアニーリング (Simulated Annealing: SA)⁵⁾ が用いられている。このことから、自動プログラミング分野においても、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 と同じ代表的なヒューリスティック手法の一つである SA を用いて自動プログラミングを行い、標準的な GP と比較することでその可能性について検討する。なお、本研究では、SA を用いた自動プログラミング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。

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 (SAP) は、シミュレーテッドアニーリング (SA) を

[†] 同志社大学大学院工学研究科

Graduate Student, Doshisha University

^{††} 同志社大学工学部

Faculty of Engineering, Doshisha University

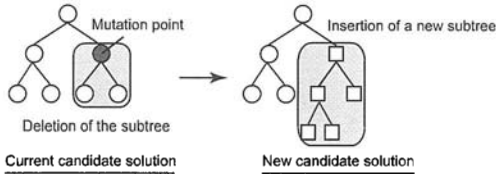


図 1 SAP における新しい解候補の生成方法
Fig. 1 Generation method of a new candidate solution in SAP

木構造が扱えるように拡張した手法であり、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。なお、温度スケジュールは、一定温度の温度スケジュール⁶⁾を用いる。これは、予備実験により、冷却するとプロトタイプが生じたのに対し、一定温度の場合ではプロトタイプが生じなかったためである。

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。

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 を基に、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⁵⁾を用いる。

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

P_{AC} は受理確率である。

STEP 4 終了判定

STEP2、および 3 を定められた回数行えば、探索を終了する。

3. SAP の有効性の検討 (GP との比較)

標準的な GP と比較を行い、自動プログラミングにおける SAP の有効性を検討する。

3.1 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

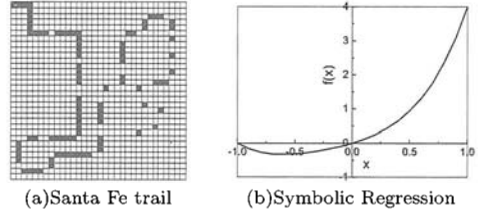


図 2 テスト問題
Fig. 2 test problems

ある Santa Fe trail 問題¹⁾、および Symbolic Regression 問題¹⁾とする。

Santa Fe trail 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。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。

3.1.1 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, PROGN2, PROGN3}、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。PROGN N 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。

すべてのエサを獲得すれば、探索が成功したとする。

3.1.2 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⁾。本実験では、式(2)に示す関数 f_{obj} を同定する。その関数 f_{obj} の概形を図 2(b) に示す。

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

非終端記号は {+, ×, −, %, sin, cos, exp, rlog}、終端記号は {x} である。

-1 から 1 の間を 0.1 刻みにした 21 個の入力に対する出力誤差の絶対値の総和が 0.01 以下ならば、探索が成功したとする。

3.2 実験に用いるパラメータ

実験に用いる SAP のパラメータを表 1 に、GP のパラメータを表 2 に示す。ここで、 T_{fixed} は一定温度、

* 探索が進むにつれてプログラムサイズが増大する現象

表 1 SAP のパラメータ
Table 1 Parameter of SAP

	Santa Fe trail	Symbolic Regression
T_{fixed}	4	0.5

表 2 GP のパラメータ
Table 2 Parameter of GP

	Santa Fe trail	Symbolic Regression
M	2000	2000
P_c	0.9	0.9
P_m	0.1	0.1
K	5	2
E	1	1
D_{max}	17	17

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),2)}である. 評価計算回数は Santa Fe trail 問題では 40 万回, Symbolic Regression 問題では 10 万回である. 試行数は 50 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, 問題で扱う非終端記号の引数の数がすべて 2 の場合, 平均して深さ約 1.4 の部分木が生成される.

3.3 SAP の解探索性能

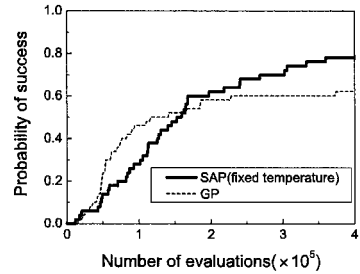
実験により得られた探索成功率 (50 試行において最適解を得た割合) を図 3 に示す.

図 3 より, 構文的イントロンが発生する Santa Fe trail 問題では, 探索終了時における SAP の探索成功率は 0.78 であるのに対して, GP は 0.62 である. 従って, この問題では, SAP は GP よりも良好な探索ができていることが分かる. 一方, 構文的イントロンが発生しない Symbolic Regression 問題では, SAP の探索成功率は 0.44 であるのに対して, GP は 1.00 である. 従って, この問題では GP の方が良好な探索ができていることが分かる.

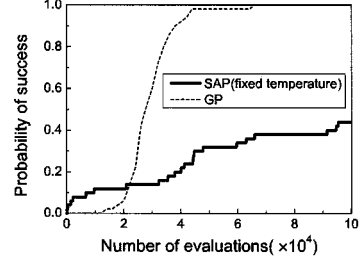
3.4 SAP のプログラムサイズの変化

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 の母集団内の全個体の平均プログラムサイズの履歴と, SAP の 100 アンニリング期間中の平均プログラムサイズの履歴を図 4 に示す.

* ルートノードの深さを 1 としている



(a) Santa Fe trail



(b) Symbolic Regression

図 3 探索成功率

Fig. 3 Probability of success

図 4 より, 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.

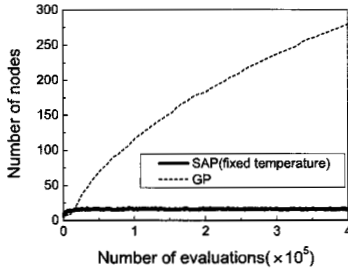
4. 考 察

4.1 解探索性能

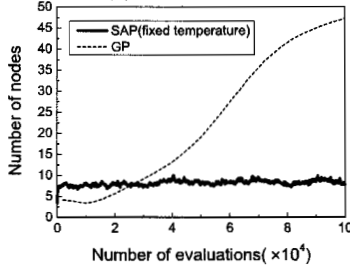
構文的イントロンが発生する Santa Fe trail 問題では, SAP は標準的な GP よりも良好な探索をすることができた. ここでは, 構文的イントロンに着目して, その理由について考察する.

構文的イントロンでのオペレーション(部分木の交換)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 5 に示す. 図 5 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 5 より, 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 ではある一定の割合に収束していることが分かり, GP と比べその割合は約 8 分の 1 と小さい.



(a) Santa Fe trail



(b) Symbolic Regression

図4 プログラムサイズの変化
Fig. 4 Impact of bloat

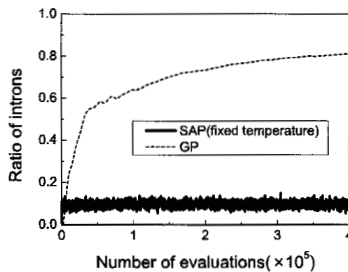


図5 プログラムに占める構文的イントロンの割合 (Santa Fe trail)

Fig. 5 Ratio of syntactic introns in program

よって、SAPは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。従って、構文的イントロンが発生する問題においては、SAPはGPよりも効率的に探索ができ、GPよりも良好な探索をすることができたと考えられる。

4.2 プログラムサイズ

適切な一定温度の温度スケジュールを用いるSAPはプロットが生じなかった。ここでは、生成処理のオペレーション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、オペレーションを行う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。また、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。

よって、適切な一定温度の温度スケジュールを用いるSAPでは、そのような改悪となる解候補にもある一定の確率で遷移するため、プロットが生じなかったと考えられる。

5. 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、構文的イントロンが発生する Santa Fe trail 問題では、SAPは標準的なGPより良好な探索ができた。また、一定温度の温度スケジュールを用いるSAPは、プロットが生じなかった。

よって、自動プログラミングにおいても対象問題に応じて探索手法を適切に使い分けることにより、効率的に探索が行える可能性があると言え、本論文により探索手法の選択肢を広げることができたと考えられる。

参考文献

- 1) Koza, J. R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press (1992).
- 2) 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)).
- 3) 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).
- 4) 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野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).
- 5) 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).
- 6) T.CONNOLLY, D.: An improved annealing scheme for the gap, *European Journal of Operational Research*, Vol. 46, pp. 93-100 (1990).