Simulated Annealing Programming Using Effective Subtrees

Yuichiro UEDA* , Mitsunori MIKI** and Tomoyuki HIROYASU***

(Received October 20, 2008)

Simulated Annealing Programming (SAP), an automatic programming method, is an extension method of Simulated Annealing (SA) that allows SA to handle tree structures. In this method, the point to exchange is chosen randomly, and the subtree to insert is also generated randomly. In this paper, we propose the method that finds out effective subtrees in search and that uses them to generate subtree for inserting. The proposal method can perform search more efficiently than standard SAP in Santa Fe trail problem and Symbolic Regression problem.

Key words : automatic programming, program search, genetic programming, simulated annealing, effective subtrees

キーワード : 自動プログラミング，プログラム探索，遺伝的プログラミング，シミュレーテッドアニーリング，有効部分木

探索に有効な部分木を活用したシミュレーテッドアニーリングプログラミング

上田祐一郎・三木光範・廣安知之

1. はじめに

ロボットの行動を制御するプログラムなどを，計算機を用いて自動設計する自動プログラミングの研究が注目されている．これは，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や，複数のロボットが協調動作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ためである．このような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）[1] や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）[2] が提案されている．

GP は代表的な自動プログラミング手法である．GP の研究は数多くされており，ADF[3] や頻出部分木発見手法 [4]，ノードの使用頻度に基づく交叉 [5] などによって，探索の効率化を実現している．しかし GP では，探索が進むにつれてプログラムのサイズが劇的に増大する「ブロート」が生じる．ブロートは探索の停滞や実行時間の増大につながり，GP の最大の問題点とされている [6]．

* Graduate Student, Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6921, Fax:+81-774-65-6716, E-mail:yueda@mikilab.doshisha.ac.jp
** Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6930, Fax:+81-774-65-6716, E-mail:mmiki@mail.doshisha.ac.jp
*** Faculty of Life and Medical Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6932, Fax:+81-774-65-6019, E-mail:tomo@is.doshisha.ac.jp

一方 SAP は，ブロートの発生原因である交叉を用いない手法であり，著者らによって提案された．SAP ではブロートが生じることなく GP と同等の性能が得られる [2]．しかし，従来の SAP における次状態の生成方法に関しては検討がされておらず，ランダムに生成した部分木をランダムに選択した交換点に挿入する方法がとられている．このため，探索が効率的に進まない可能性がある．

そこで本研究では，探索に有効な部分木（以降，有効部分木と呼ぶ）を探索中に見つけ出す方法を考え，これを生成する部分木に対して活用することで，SAP の探索性能の効率化を図る．

2. シミュレーテッドアニーリングプログラミング（SAP）

SAP は，SA を木構造が扱えるように拡張した手法である．以下に SAP のアルゴリズムを示す．

STEP 1 初期解の生成

初期解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，それを評価する．具体的には Fig. 1 に示したように，現在の解に対してランダムに突然変異点（交換点）を選択し，その点を根とする部分木を削除する．その後，ランダムに部分木を生成し，削除した部分に挿入する．

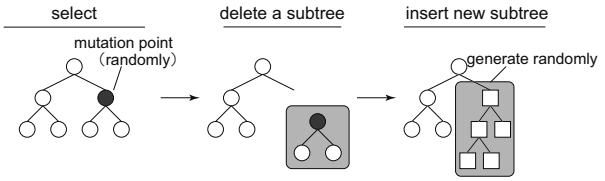


Fig. 1. Generation Method in SAP.

STEP 3 受理判定，状態遷移

現在の解の評価値 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 T を基に，新しい解候補を受理するかの判定（受理判定）を行う．受理判定には式 (1) に示す Metropolis 基準 [7] を用いる．

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

STEP 4 クーリング

STEP2，および 3 を一定期間繰り返した後，温度 T を小さくする．クーリング後の温度 T_{k+1} は，式 (2) によって決定する．

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

ここで，γ は冷却率であり，T_k は現在の温度である．

STEP 5 終了判定

STEP2〜4 を定めた回数行えば，探索を終了する．

3. 対象問題

本研究では，GP の代表的なベンチマーク問題である Santa Fe trail 問題 [1] および Symbolic Regression 問題 [1] を対象とした．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．

3.1 Santa Fe trail 問題

Santa Fe trail 問題とは，1 匹の人工蟻が Fig. 2(a) に示す 32 × 32 のマス目上に配置された餌を，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]．なお，すべての餌を獲得すれば探索が成功したとする．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 1 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，$\cdots$，第 n 引数の順に実行する．本問題では，IF_FOOD_AHEAD の連鎖により，構文的イントロンが発生する．また評価関数 E_{val} は，餌の総数から獲得した餌の数 F を引いたものであり，0 を最適解とする最小化問題である．

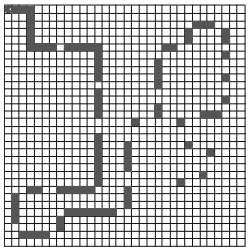
3.2 Symbolic Regression 問題

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]．本研究では，式 (3) に示す関数 f_{obj} を同定する．その関

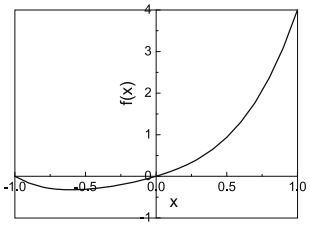
数 f_{obj} の概形を Fig. 2(b) に示す.

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

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 0.001 以下ならば, 探索が成功したとする最小化問題である. 本問題では構文的イントロンは発生しない.



(a)Santa Fe trail



(b)Symbolic Regression

Fig. 2. Test Problems.

4. 有効部分木の活用

本研究では, SAP の探索性能を向上させる方法として, 有効部分木に着目した. 以下ではまず有効部分木を自動抽出する方法について記し, 続いてこの有効部分木を活用した提案手法について記す.

4.1 有効部分木の発見

SAP では次状態生成の際, ランダムに挿入木を生成し, ランダムに突然変異点（交換点）を選択する. しかし, 有効部分木を活用できれば, 効率的な探索が期待できる. しかし, 人があらかじめどのような部分木が有効であるかを判断することは困難である.

GP では, 頻出部分木発見手法 [4] やノードの使用頻度に基づく交叉 [5] などが提案されているが, 前者は GP の多点探索を利用した手法であるため SAP には適さず, 後者は条件分岐が非終端記号に含まれる問題には適用できるがすべてのノードを一様に使用する問題には適用できない. そこで本研究では, SAP における有効部分木の自動抽出法として, 挿入木に着目した方法を提案する.

これは次状態生成の際, 評価値が改良された場合は挿入木が探索に有効である可能性が高く, そうでない場合は有効でない可能性が高い, という考えに基づく. 本研究では, 以下の 3 種類の部分木を有効部分木の定義として考えた.

【type A】評価値が改良された際に挿入した部分木.
【type B】評価値が改良された際に挿入した部分木の親ノードを根とする部分木.
【type C】上記 2 種類の部分木.

これは, 挿入木がそのまま評価に大きな影響を与えている可能性, 挿入木を含むより大きな部分木が評価に大きな影響を与えている可能性, これら両方の部分木が評価に大きな影響を与えている可能性, を考慮したためである.

4.2 有効部分木の活用

有効部分木の活用方法として, 確率的に挿入木に活用する手法を提案する. これにより, 生成処理をランダムに行っていた従来手法よりも効率的な探索が期待できる. この提案手法のフローチャートを Fig. 3 に示し, 以下に提案手法の具体的なアルゴリズムを記す.

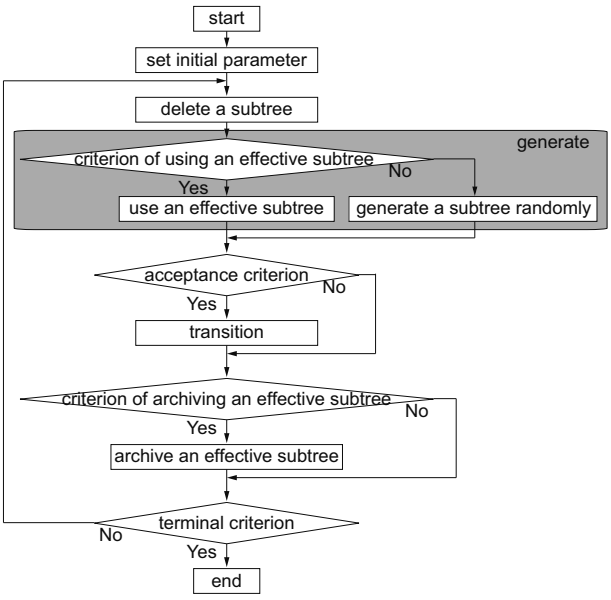


Fig. 3. Algorithm of Proposal Method.

1 有効部分木の抽出

生成処理において部分木を挿入した結果，評価値の差分 ΔE が 0 より小さくなった場合，有効部分木をアーカイブに加える．

2 有効部分木の更新

アーカイブサイズが上限を超える場合，上限を超える数だけアーカイブから部分木を削除する．削除する部分木は，アーカイブ中から挿入した結果評価値が改良した割合が低い部分木から順に取り除く．これにより，より有効な部分木がアーカイブに残りやすくなる．

3 有効部分木の挿入

挿入する部分木は，基本的には従来と同様ランダムに生成し，確率的に有効部分木を挿入木とする．

5. 数値実験

5.1 実験概要

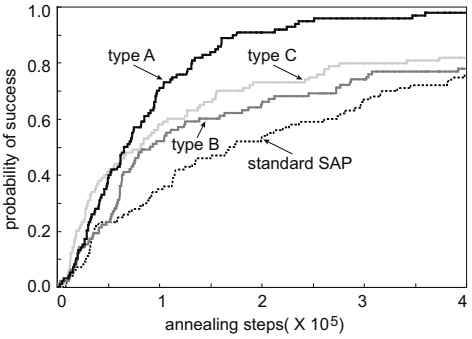
提案手法における有効部分木を検討するため，提案手法と従来手法の比較を行う．試行数は 100 回である．対象問題は 3 章で記した問題であり，比較項目は 100 試行中最適解を得た回数（成功率）とする．用いたパラメータについて，探索回数は Santa Fe trail 問題では 40 万回，Symbolic Regression 問題では 20 万回とし，温度は Santa Fe trail 問題では 4，Symbolic Regression 問題では 0.5 で固定した [2]．なお，提案手法において，アーカイブサイズを 10，生成した有効部分木の挿入確率を 20%とした．

5.2 実験結果

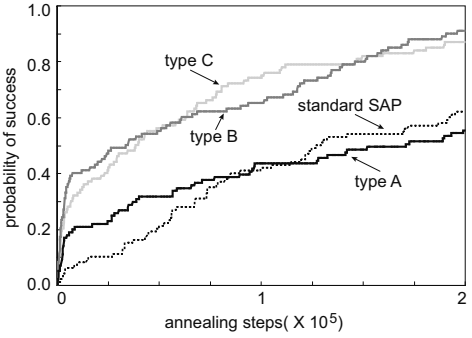
実験結果として，100 試行中の成功率の履歴を Fig. 4 に示す．なお，Fig. 4 中の (a) は Santa Fe trail 問題，(b) は Symbolic Regression 問題における結果である．また Fig. 4 は縦軸に成功率，横軸に探索回数を示す．したがってグラフは上部ほど良い結果を意味する．

Fig. 4 より，両対象問題に対して提案手法は従来手法よりも高い成功率を得ている．したがって，提案手法が従来手法より効率的な探索を行うことができると言える．

しかし，有効部分木の定義によってその性能は異なり，Santa Fe trail 問題では挿入木のみを有効部分木とした手法がもっとも性能がよく，Symbolic Regression 問題では挿入木の親ノードを根とする部分木を有効部分木とした手法が最も性能がよかった．これにより，有効部分木の定義は対象問題に依存すると言える．



(a)Santa Fe trail



(b)Symbolic Regression

Fig. 4. Probability of Success.

6. 考察

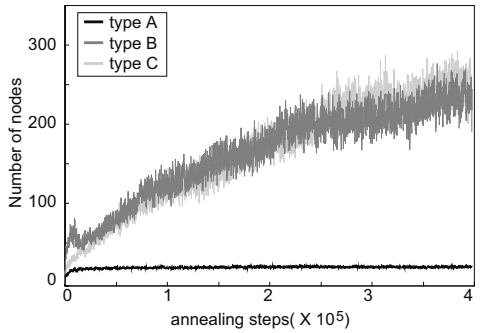
ここで，各有効部分木の定義におけるノード数の履歴を Fig. 5 に示す．なお，Fig. 5 中の (a) は Santa Fe trail 問題，(b) は Symbolic Regression 問題における結果である．また Fig. 5 は縦軸にノード数，横軸に探索回数を示す．

Fig. 5 より，まず Santa Fe trail 問題では挿入木のみを有効部分木とした場合のノード数はある一定の値に収束しているのに対し，それ以外の定義ではノード数が徐々に増大している．これは，Santa Fe trail 問題は構文的イントロンが発生する問題であるため，有効部分木を挿入木を含むより大きな部分木と定義することで，構文的イントロンを含む有効部分木を生成する可能性が高くなるためだと考えられる．このため，ノード数が増大し続けない挿入木のみを有効部分木と定義した手法が Santa Fe trail 問題において有効な結

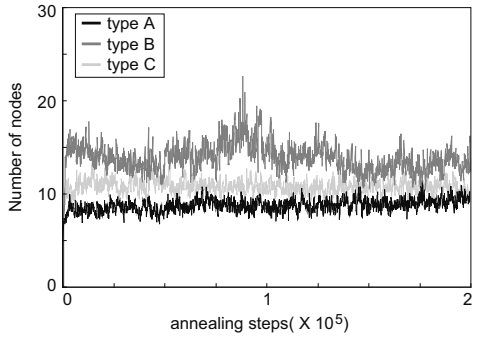
果を得たと考えられる．

次にSymbolic Regression問題では，各手法ともノード数はある一定の値に収束し増大し続けていない．これは，構文的イントロンが発生しない問題であるためと考えられる．そしてこの場合，挿入木の親ノードを根とする部分木を有効部分木と定義した手法が最も有効な結果を得たことから，挿入木を含むより大きな部分木が探索に有効に働くと考えられる．

以上より，SAPでは挿入木を含むより大きな部分木が探索に大きな影響を与えると言える．そして，構文的イントロンが発生する問題ではこの部分木に構文的イントロンを含む可能性も増える．従って，構文的イントロンが発生しない問題に対しては挿入木の親ノードを根とする部分木を，構文的イントロンが発生する問題では挿入木を，それぞれ有効部分木と定義することが有効であると言える．



(a)Santa Fe trail



(b)Symbolic Regression

Fig. 5. History of Program Size.

7. まとめ

本研究では，SAPの次状態生成方法の改良として，有効部分木を自動抽出し活用する手法を提案した．数値実験の結果，提案手法は従来手法よりも高い探索性能を得た．また，構文的イントロンが発生するSanta Fe trail問題では挿入木を有効部分木と定義したものが，構文的イントロンが発生しないSymbolic Regression問題では挿入木の親ノードを根とする部分木と定義したものが，それぞれ最も効率的な探索をすることができた．よって，対象問題の特徴に応じて有効部分木の定義を使い分けることで，提案手法がより効率的な探索を行えることを示すことができた．

参 考 文 献

1) J. R Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.

2) 藤山佳久, 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”, 「情報処理学会論文誌」, Vol.48, pp. 88–102, 2007.

3) J. R Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, 1994.

4) T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Sakamoto, S. Akikawa, “Efficient Substructure Discovery from Large Semistructured Data”, *Proc. of the 2nd SIAM Intl. Conf. on Data Mining*, pp. 158–174, 2002.

5) D.Katagiri, S. Yamada, “Speedup of Evolutionary Behavior Learning with Crossover Depending on the Usage Frequency of a Node”, *IEEE International Conference on Systems, Man, and Cybernetics*, 1999, No. 5, pp. 601–606, 1999.

6) 伊庭斉志, 「遺伝的プログラミング入門」, 東京大学出版会, 2001.

7) N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, “Equation of State Calculation by Fast Computing Machines”, *Journ. of Chemical Physics*, Vol. 21, pp. 1087–1092, 1953.