

シミュレーテッドアニーリングプログラミングにおける探索に有効な部分木

Effective Subtrees in Simulated Annealing Programming

○非 上田 祐一郎 (同志社大院) 正 三木 光範 (同志社大理工)
正 廣安 知之 (同志社大生命医科)

Yuichiro UEDA, Graduate School of Engineering, Doshisha University
Mitsunori MIKI, Doshisha University, Tatara Miyakodani 1-3, Kyo-Tanabe, Kyoto
Tomoyuki HIROYASU, Doshisha University, tomo@is.doshisha.ac.jp

Key word: Automatic Programming, Optimization, Simulated Annealing, Subtrees

1 はじめに

ロボットの行動規則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。主な手法として、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り、探索の効率化を実現する数多くの研究がされている。しかしGPでは、探索が進むにつれてプログラムサイズが劇的に増大する「プロート」の問題がある。

一方SAPは、プ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ではプロートが生じることなくGPと同等の探索性能が得られる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。

本研究では、SAPにおいて、探索に有効に働く部分木を確認し、この部分木を生成処理に活用することで、探索を効率化を図る。

2 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、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。

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³⁾を用いる。

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

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。

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 Santa Fe trail 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⁾を対象とした。Santa Fe trail問題とは、1匹の人工蟻が 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{IfFoodAhead, Prog2, Prog3}, 終端記号は{left, right, move}である。IfFoodAhead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。Prog n は引数を n 個持ち、第1引数, 第2引数, ..., 第 n 引数の順に実行する。本問題では、IfFoodAhead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。

4 有効部分木の確認

最適解に共通して出現する部分木が存在すれば、その部分木は最適解を探索する上で重要となる部分木と言える。そこで、Santa Fe trail問題にSAPを適用して得た最適解100個に対して、部分木の出現頻度解析を行った。

ここで、Table. 1に出現頻度が上位10個の部分木を示す。Table. 1より、多くの最適解に含まれている部分木が存在することがわかった。したがって、最適解を得るためには必要となる重要な部分木が存在すると考えられる。この部分木を有効部分木と呼ぶ。

Table 1 Lanking of subtrees

Appearance ratio	Subtrees
69/100	Prog2 - move - move
52/100	IfFoodAhead - move - left
35/100	IfFoodAhead - Prog2 - move - move - right
23/100	IfFoodAhead - Prog2 - move - move - left
22/100	Prog3 - move - move - right
20/100	Prog3 - right - left
17/100	Prog2 - move - right
17/100	IfFoodAhead - move - right
16/100	Prog3 - move - move - left
15/100	Prog2 - right - left

5 有効部分木の活用

5.1 活用方法

前節で述べた有効部分木を探索中で活用すれば、探索性能は向上すると考えられる。そこで、本研究では出現頻度が高かった部分木4種類を終端記号に含めることで、探索性能が向上するか一般的なSAPと比較を行うことで検討した。なお、この部分木の使用方法として、カプセル化する場合とカプセル化しない場合の2種類を考えた。

この結果として、各手法を100試行行った際の成功率の履歴をFig. 1に示す。Fig. 1より、終端記号に有効部分木を含めると探索性能は向上し、特にその部分木をカプセル化することが有効であることがわかった。

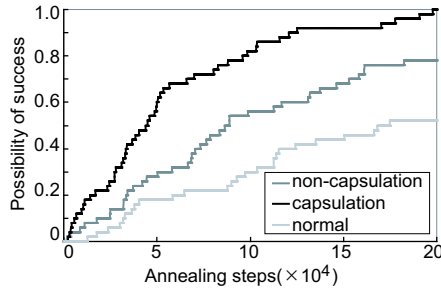
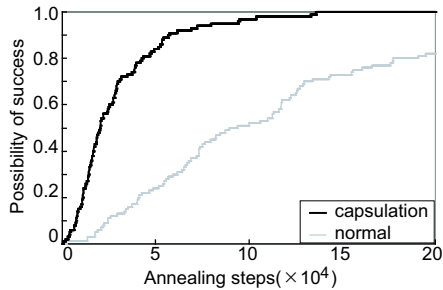


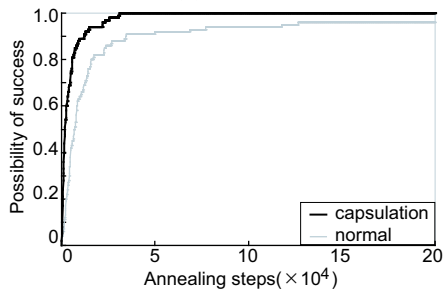
Fig. 1 Effectiveness of capsulation

5.2 SAPにおける有効部分木

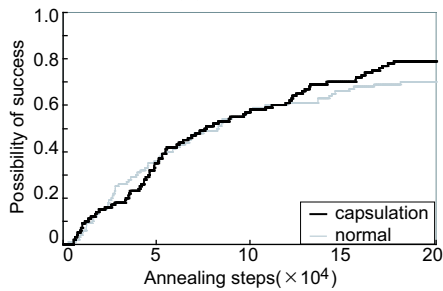
ここでは、Santa Fe trail問題の餌の配置を変えた問題3種類(Field A, B, C)を対象にした。それらの問題に対して、前節と同じ部分木(Santa Fe trail問題における有効部分木)をカプセル化したSAPと一般的なSAPを適用した結果として、100試行の成功率の履歴をFig. 2に示す。なお、実験の詳細については、紙面の都合上割愛する。



(a)Field A



(b)Field B



(c)Field C

Fig. 2 History of success ratio

Fig. 2より、Santa Fe trail問題における有効部分木をカプセル化したSAPは一般的なSAPより、Field A, Bでは性能が向上し、Field Cでは同等の性能であった。しかし、どのFieldにおいても性能が悪化することはなかった。

5.3 考察

前節の通り、どのFieldにおいても性能が悪化することはなかった。しかし、Fig. 2(c)では性能が向上しなかった。これは、Santa Fe trail問題では有効に働いた部分木がField Cでは有効に働かなかったためと考えられる。そこで、こ

のField Cに一般的なSAPを適用して得た最適解100個に対して、再度部分木の出現頻度解析を行った。Table. 2に出現頻度が上位10個の部分木を示す。Table. 2より、Santa Fe trail問題における有効部分木と異なるものも含まれていることがわかる。すなわち、同じ餌集め問題においても、有効部分木は問題に依存するものとしらないものが存在すると言える。

Table 2 Lanking of subtrees

Appearance ratio	Subtrees
83/100	Prog2 - move - move
52/100	Prog2 - move - left
49/100	IfFoodAhead - Prog2 - move - left - left
44/100	Prog2 - move - right
43/100	IfFoodAhead - Prog2 - move - right - right
41/100	IfFoodAhead - left - right
38/100	IfFoodAhead - Prog2 - move - move - right
37/100	IfFoodAhead - right - left
35/100	IfFoodAhead - Prog2 - move - move - left
12/100	IfFoodAhead - move - left

また、Table. 2に示すField Cにおける有効部分木4種類をカプセル化して再度数値実験を行った。この結果として100試行の成功率の履歴をFig. 3に示す。Fig. 3より、有効部分木をカプセル化することによって探索性能が向上することがわかる。

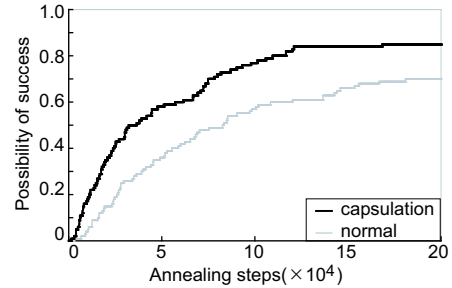


Fig. 3 Effectiveness of effective subtrees

以上よりSAPでは、同じ餌集め問題でも有効部分木は問題に依存するものとしらないものがあり、各問題に有効な部分木をカプセル化することで、探索性能を向上することができると言える。

6 まとめ

本研究では、SAPにおいて有効部分木が存在することを確認した。また、有効部分木は同じ餌集め問題を対象とした場合でも、問題に依存しない部分木と依存する部分木が存在することを確認した。そしてSAPでは、これらの部分木をカプセル化し終端記号として用いることで、探索効率を向上することができることを示した。今後これを実現するためには、以下の方法が考えられる。すなわち、問題に依存しない有効部分木については数種類の問題を解くことによってライブラリ化する。問題に依存する有効部分木については、探索過程で自動的に発見する。そして、これらを利用することでSAPの探索効率向上が図れると考えられる。

参考文献

- 1) Jhon R. Koza. Genetic programming : On the programming of computers by means of natural selection. *MIT Press*, 1992.
- 2) 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. *情報処理学会論文誌*, Vol. 48, pp. 88-102, 2007.
- 3) N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculation by fast computing machines. *Journ of Chemical Physics*, Vol. 21, pp. 1087-1092, 1953.