

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部分木生成方法の検討

青木 大

1 はじめに

ロボットの制御プログラムなどのプログラムを、コンピュータを用いて最適化することは重要である。これは、人手による設計では、あらかじめ想定した状況にしか対応できないことや、複数台のロボットを用いて協調行動させるような複雑なプログラムの場合、人手では設計困難であるからである。

我々の研究室では、このようなプログラムを自動生成する手法として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)¹⁾ を提案している。SAP は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手法であり、1 つの木構造に対して部分木の削除、および挿入を繰り返し行うことでプログラムを最適化する手法である。これまでの研究により、ノード数が少なくとも最適解を得ることができる問題においては、SAP は遺伝的プログラミング (Genetic Programming: GP)²⁾ と同等の性能を得ることができ、かつ GP よりコンパクトなプログラムを生成できることが分かっている。

これまで、SAP では次状態を生成する際に挿入する部分木をランダムに生成していた。よって、部分木の生成方法を検討することにより、SAP の探索性能の向上が期待できると思われる。

そこで本研究では、SAP において部分木の生成方法について検討を行い、探索性能の向上を図る。

2 シミュレーテッドアニーリングプログラミング (SAP)

SAP では、現在の木構造に対してランダムにある 1 ノードを選択し、そのノード以下の部分木を削除し、その部分にランダムに生成した部分木を挿入することで次状態を生成する。そして、次状態が改良方向へ生成された場合には無条件でその遷移を認め、改悪方向へ生成された場合でも温度と呼ばれる制御パラメータにより、確率的に改悪方向への遷移を認めるメカニズムを持つ。これにより、SAP で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。

SAP のアルゴリズムを以下に示す。

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。

STEP 2 生成処理

新しい解候補を現在の解候補に対して GP の突然変異と同様の処理を行い生成し、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 (Fig. 1)。

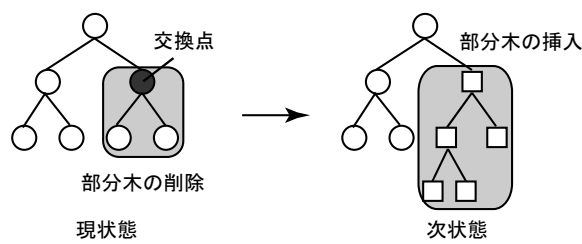


Fig.1 SAP における次状態の生成方法 (出典：自作)

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され、改悪方向へ生成された場合でも確率的に受理される。改悪方向への受理確率は、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 クーリング

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 を小さくするクーリングを行う。クーリング後の温度 T_{k+1} は、式 (2) によって決定する。

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

ここで、 γ は冷却率であり、 T_k は現在の温度である。クーリングを行うことにより、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。そうでなければ STEP2~4 を繰り返す。

3 提案する部分木の生成方法

これまで、次状態を生成する際に挿入する部分木は、GP の突然変異と同じ方法 (最大の深さ 4、非終端記号の選択率 0.5、最大の深さを越えた場合は生成し直す) を用いてランダムに生成していた。よって、部分木の生成方法を検討することで探索効率の向上が期待できると思われる。

本発表では、イントロンを生成しない部分木生成方法

と、有効部分木を確率的に生成する部分木生成方法を提案する。

3.1 イントロンを生成しない生成方法

SAP および GP で生成されるプログラムには、条件分岐の重ね合わせにより、評価時に実行されないノード (イントロンノード) が含まれていることがしばしばある。Fig. 2 に示す木構造の丸で囲んだ部分がイントロンノードとなっている。

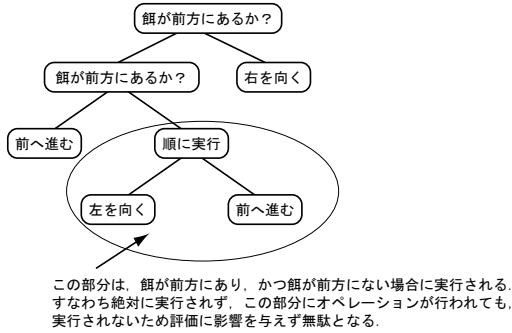


Fig.2 プログラム中のイントロン (出典：自作)

イントロンノードでオペレーションが行われても評価には全く影響を及ぼさない。

GP では現世代ではイントロンとなっているノードが、次世代では交叉によってイントロンノードでなくなる可能性がある。また、イントロンは破壊的な交叉から守る働きがあるとの報告³⁾もあることから、GP ではイントロンは必ずしも不必要ではないとされている。

一方、SAP では突然変異のみを行うため、イントロンノードが GP のように有益な効果をもたらすことはない。よって SAP ではイントロンが含まれると、評価に影響を及ぼさないイントロンノードでオペレーションが行われる可能性が生じ、イントロンは無駄であるといえる。

そこで、SAP においてイントロンの発生を認めない部分木の生成方法を提案する。具体的には、部分木の挿入を行った際に、イントロンが生成されていないかチェックし、イントロンが生成されている場合には、部分木の生成をやり直す。

3.2 有効部分木の確率的生成方法

Santa Fe trail 問題において SAP で生成されたプログラムを解析すると Fig. 3 に示す部分木が多く使われていた。



Fig.3 探索により得られた木に多く含まれた部分木 (出典：自作)

そのため、これらの部分木は Santa Fe trail の探索に有効な部分木であると考えられる。そこで、部分木を挿入する際にこれらの有効と思われる部分木を挿入すれば、探索性能は向上すると思われる。

このことを検証するために、生成処理の際に Fig. 3 に示した有効と思われる部分木を 10% の確率で挿入した実験を行った。その結果を Fig. 4 に示す。横軸は評価計算回数、縦軸は探索の成功率 (30 試行中最適解を得た割合) であり、成功率が高い方が良い結果を意味している。

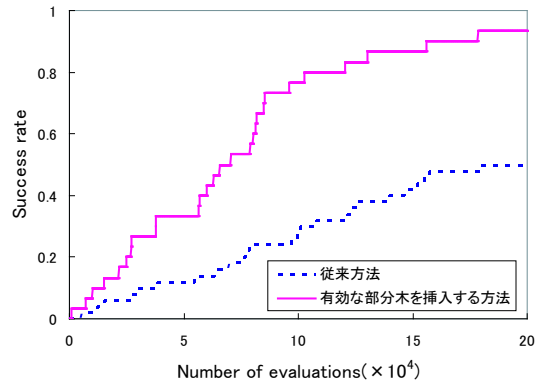


Fig.4 有効と思われる部分木を 10% の確率で挿入した場合の探索性能 (出典：自作)

Fig. 4 より、有効と思われる部分木をある確率で挿入することにより、ランダムに生成した部分木を挿入する場合より探索性能が向上していることが分かる。

しかし多くの場合、事前にどのような部分木が有効であるかを予測することは困難である。

そこで本研究では、有効である可能性が高い部分木を自動抽出し、その部分木を次状態生成時に挿入することを考える。

この提案アルゴリズムのフローチャートを Fig. 5 に示す。

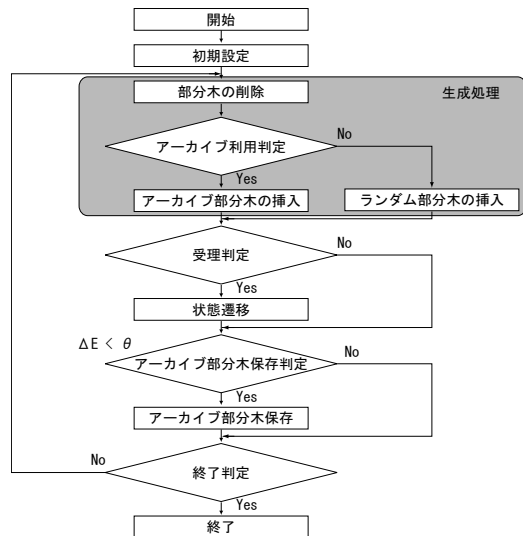


Fig.5 提案アルゴリズムのフローチャート (出典：自作)

具体的な有効部分木の確率的生成方法のアルゴリズムを以下に示す。

1. 生成処理において、部分木を挿入した結果、エネルギー差分 ΔE が閾値 θ より小さくなった場合に、Fig.

6 に示すように、挿入した部分木、および挿入したノードの親ノードを根とした部分木を有効部分木としてアーカイブに加える。

2. アーカイブサイズが上限を超える場合、アーカイブから 2 つの部分木を取り除く。取り除く部分木は、挿入成功率 (挿入受理回数/挿入回数) がもっとも低かったものとする。

なお、探索終盤で評価値が低くなり、 ΔE の値が小さくなった場合にも有効部分木の抽出が行われるよう、閾値 θ は式 (3) に示すように現在の評価値 E を 10 で除算したものを使用している。

$$\theta = \begin{cases} -1 & \text{if } E_{\text{current}} \leq 10 \\ -(E_{\text{current}}/10) & \text{otherwise} \end{cases} \quad (3)$$

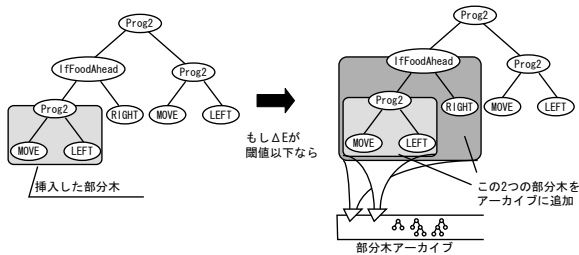


Fig.6 有効部分木の抽出 (出典：自作)

4 数値実験

4.1 実験概要

提案手法での部分木生成の有効性を検討するために、従来の部分木生成方法を用いる SAP と 2 つの提案手法を比較する。比較する項目は、探索の成功率 (30 試行中、最適解を得た割合) とする。対象問題は、次節に示す GP のベンチマーク問題である Santa Fe trail 問題とする。

用いるパラメータは一定温度 4、評価計算回数は 20 万回、試行数は 30 試行とする。有効部分木の確率的生成方法ではアーカイブサイズを 5、生成した有効部分木の挿入確率を 20% とする。

4.2 対象問題

Santa Fe trail 問題とは、一体の人工蟻が、限られたエネルギー内で行動し、Fig. 7 に示すフィールドに配置された餌をできるだけ多く獲得することを目的とする問題である。人工蟻は餌上を通ることにより餌を獲得することができ、エネルギーは終端記号が 1 つ実行されるごとに 1 消費する。人工蟻の初期エネルギーは 400 である。

問題に用いた終端、非終端記号を Table 1 に示す。Table 1 において、非終端記号は小文字で、終端記号は大文字で示している。Santa Fe trail 問題は条件分岐 if_food_ahead の重ね合わせにより、イントロンノードが発生する可能性がある問題である。

評価関数 E_{val} は、餌の総数である 89 から獲得した餌の数 F を引いたものであり、0 を最適値とする最小化問題である。

$$E_{\text{val}} = 89 - F \quad (4)$$

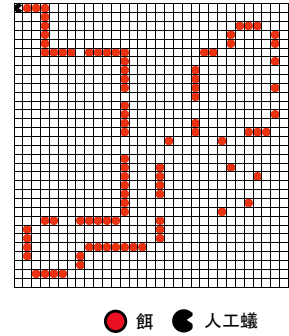


Fig.7 Santa Fe trail 問題のフィールド (出典：自作)

Table1 Santa Fe trail 問題に用いる終端・非終端記号

記号	機能
if_food_ahead	子ノード 2 つもつ。1 マス前方にえさがあれば第 1 子ノードを、なければ第 2 子ノードを実行する。
prog2	子ノード 2 つもつ。第 1 子ノード、第 2 子ノードの順に実行する
prog3	子ノード 3 つもつ。第 1 子ノード、第 2 子ノード、第 3 子ノードの順に実行する。
LEFT	90 度左を向く。
RIGHT	90 度右を向く。
MOVE	1 マス前進する。

4.3 実験結果

4.3.1 イントロンを生成しない生成方法での探索性能

SAP において、イントロンを生成しない生成方法での探索性能を Fig. 8 に示す。Fig. 8 は横軸が評価計算回数、縦軸が探索の成功率であり、成功率が高い方が良い結果を意味している。

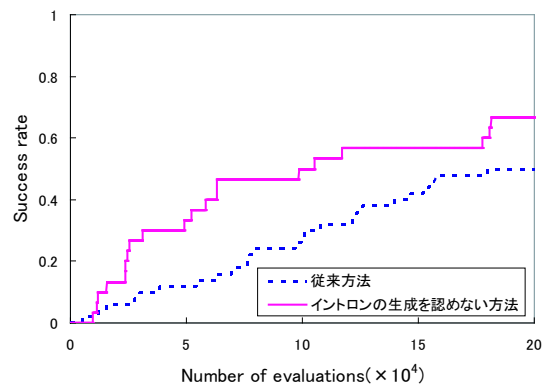


Fig.8 イントロンを生成しない生成方法での探索性能 (出典：自作)

Fig. 8 より、イントロンを生成しない生成方法を用いることで、従来方法よりも良好な探索が行えることがわかる。これは、イントロンを生成しないことで、評価に影響を及ぼさないノードに対しオペレーションを行うことがなくなり、無駄な探索を行うことがなくなったからであると考えられる。

4.3.2 有効部分木の確率的生成方法での探索性能

有効部分木の確率的生成を行う方法での結果を Fig. 9 に示す。

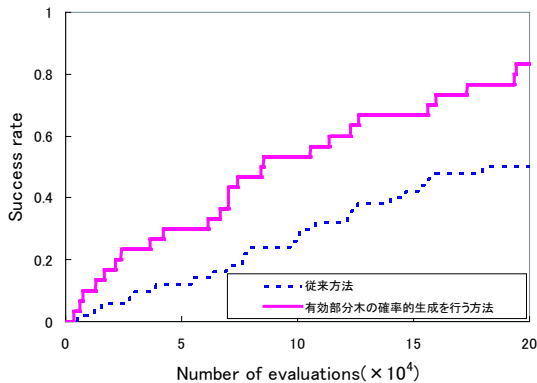


Fig.9 有効部分木の確率的生成方法での探索性能 (出典：自作)

Fig. 9 より、有効部分木の確率的生成方法を用いることで、従来方法より良好な探索が行えていることが分かる。これは、有効な部分木が、ランダムに部分木を生成する方法では生成されにくいのに対し、有効部分木の確率的生成方法では、生成される確率が高くなっているためと考えられる。

4.3.3 2つの手法を組み合わせて用いた場合の探索性能

両手法を組み合わせて用いる方法での結果を Fig. 10 に示す。

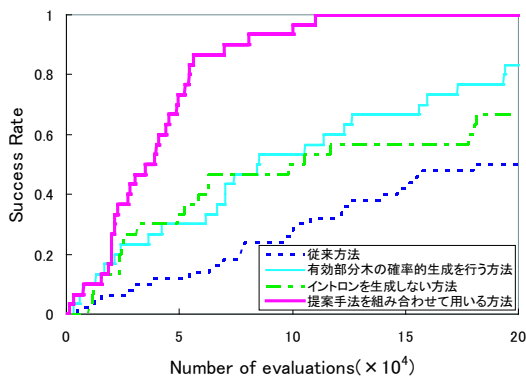


Fig.10 提案手法を組み合わせた場合の探索性能 (出典：自作)

Fig. 10 より、イントロンを生成しない生成方法および有効部分木の確率的生成を行う生成方法を組み合わせて用いることで、一方のみ用いる場合よりさらに良好な結果が得られていることが分かる。

特に、従来方法では評価計算回数が 20 万回で得られる成功率が 0.5 であったが、2つの方法を組み合わせて用いた手法では、従来の方法と比べて 5 分の 1 の約 4 万回で成功率が 0.5 となっている。これにより、SAP における探索の効率化が行えていることが分かる。

5 まとめと今後の課題

SAP において、部分木の生成方法を検討し、探索効率の向上を図った。本報告では、イントロンを生成しない方法と有効部分木を確率的に生成および挿入する方法、および上記 2 つの方法を組み合わせて用いる方法を提案した。

提案手法を Santa Fe trail 問題に適用し実験を行った結果、イントロンを生成しない生成方法、および有効部分木を確率的に生成および挿入する生成方法をそれぞれ用いることで効率的に探索を行うことが確認できた。また、2つの方法を組み合わせて用いる生成方法により、更に効率的な探索が行えることが分かった。

今後、Santa Fe trail 問題とは特徴の異なる問題に対して提案手法を適用し、有効性の検討を行う。また、有効部分木を確率的に生成および挿入する方法では、アーカイブサイズと挿入確率についても、今後検討する予定である。

また、現在は交換点の選択方法については検討が行えていないので、今後検討する必要がある。

参考文献

- 1) 三木 光範, 廣安 知之, 藤田 佳久, シミュレーテッドアニーリングプログラミングによる群知能の発現, 第 67 回情報処理学会全国大会講演論文集 Vol2 pp.299-300, 2005
- 2) Koza, J. R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992
- 3) Angeline, P. J. Two self-adaptive crossover operators for genetic programming. In Angeline, P. J. and Kinnear, Jr., K. E., editors, Advance in Genetic Programming 2, chapter 5, page 89-110. MIT Press, Cambridge, MA, 1996