

シミュレーテッドアニーリングプログラミングの温度並列化

小畑 拓也

1 はじめに

近年、ロボットの行動を制御する行動規則など、人の手では容易に組むことのできないプログラムを、計算機を用いて自動生成させる、自動プログラミングと呼ばれる方法が注目されている。本研究室では、自動プログラミングの手法の 1 つである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⁾ を研究している。SAP とは、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させた手法であり、温度と呼ばれる重要なパラメータが存在する。しかし、温度スケジュールの決定は容易でない。SA では、温度並列化によってこの問題を解決した有効な手法として、温度並列 SA (Temperature Parallel Simulated Annealing:TPSA)²⁾ が提案されている。温度並列化とは、複数のプロセッサがそれぞれ異なる一定の温度パラメータで探索を行い、ある周期ごとに解交換を行うアルゴリズムのことである。このアルゴリズムを SAP に適用した温度並列シミュレーテッドアニーリングプログラミング (Temperature Parallel Simulated Annealing Programming:TPSAP) の有効性を検討する。本手法では、TPSA と同様の温度並列化により、温度スケジュールの自動化だけでなく、局所解に陥りにくい探索が期待できる。本研究では、Santa Fe Trail 問題において比較実験を行い、TPSAP の有効性を検討した。

2 シミュレーテッドアニーリングプログラミング (SAP)

2.1 SAP のアルゴリズム

SAP のアルゴリズムは SA と同様であり、以下に詳細を示す。

1. 生成処理

生成処理の様子を Fig. 1 に示す。Fig. 1 のように、現在の解候補に対して、ランダムに挿入点を選択し、挿入点を根とする部分木を削除する。その後、削除した部分にランダムに生成した部分木を挿入し、新しく生成した次解候補の評価値を得る。

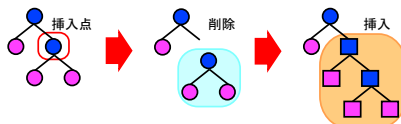


Fig.1 生成処理 (出典:自作)

2. 受理判定, 状態遷移

新しく生成された解候補を受理するか判定を行う。受理判定には、温度パラメータ T 、および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 ΔE を用いた Metropolis 基準 (式 1) を用いる。

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

P は受理確率であり、解が改良方向に生成された場合は必ず受理し、改悪方向に生成された場合でも P により確率的に受理する。温度は、高温であるほど改悪を受理しやすく、低温であるほど改悪を受理しにくくなる。

3. クーリング

1 と 2 を繰り返し行うことをアニーリングと呼び、ある一定期間アニーリングを行った後に、式 2 で示すように温度を下げる。徐々に温度を下げていくことで、大域的探索から局所的探索に移行していくため、良好な解探索が期待できる。

$$T_{next} = \gamma T_{current} \quad (0.8 \leq \gamma < 1) \quad (2)$$

2.2 SAP の重要なパラメータ

木構造が進化していく中で、温度は進化を制御する重要なパラメータである。そのため、温度スケジュールを決めることは容易ではない。主な温度スケジュールには、クーリングや一定温度が挙げられる。クーリングを行う場合には、探索に伴って大域的探索から局所的探索まで行えるため、良好な解探索が期待できるが、そのためには適切な最高温度や最低温度、さらにクーリング周期や冷却率などのパラメータを設定する必要がある。一方で、一定温度を行う場合は探索に良好な温度領域を発見した場合には、その温度を用いた一定温度で探索を行うことで良好な解探索ができるが、有効な温度領域の発見のためには十分な予備実験が必要であり、膨大な計算コストがかかる。

3 温度並列 SA (TPSA)

3.1 TPSA のアルゴリズム

TPSA は、温度並列化によって SA の温度スケジュールを自動化した手法である。具体的には、複数のプロセッサに異なる温度を与え、各プロセッサは一定温度で探索を行う。そして、一定の間隔で隣接する温度のプロセッサ間で解交換を行う。解自身が温度を決定するため、Fig. 2 に示すように温度スケジュールが自動化される。また、解交換の決定は、式 3 で得られる確率により確率的に決定する。式 3 において、温度の差分 ΔT 、および評価値の差分 ΔE は、 $\Delta T = T' - T$ 、 $\Delta E = E' - E$ である。

$$P_{\Delta T, \Delta E} = \begin{cases} 1 & \text{if } \Delta T \cdot \Delta E \leq 0 \\ \exp(-\frac{\Delta T \cdot \Delta E}{T \cdot T'}) & \text{otherwise} \end{cases} \quad (3)$$

$P_{\Delta T, \Delta E}$ は、SAP の受理判定における受理確率 P と同様、解交換時に高温側のプロセッサにおける解が低温

側よりも良好な解を持っている場合、解交換は無条件で行われる。また、そうでない場合は温度とエネルギーから計算される確率によって解が交換される。これによって、局所解に陥ることなく真の最適解を得ることが期待できる。

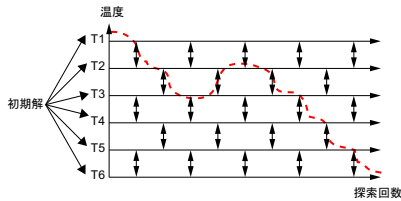


Fig.2 TPSA の温度スケジュール (参考文献²⁾より参照)

3.2 SAP への適用

前節で記した TPSA のアルゴリズムを SAP に適用する。これを温度並列 SAP (TPSAP) と呼ぶ。したがって、SAP においても課題であった温度スケジュールの決定を自動化することができる。また、これにより局所解に陥ることなく最適解を得ることも期待でき、SAP の性能向上につながると考えられる。

4 対象問題

本研究では、Santa Fe Trail 問題を対象問題とする。Santa Fe Trail 問題は、人工蟻が限られたエネルギー内で Fig. 3 に示すフィールドに設置された餌をより多く獲得させることを目的としている。なお、初期エネルギーは 400 である。

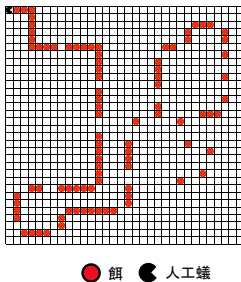


Fig.3 Santa Fe Trail 問題のフィールド (出典:自作)

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、終端記号は大文字、非終端記号は小文字で示している。

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

5 数値実験

TPSAP の有効性を検討するため、比較実験を行った。なお、比較手法はクーリングを用いた SAP とした。

5.1 パラメータの設定

本実験で用いたパラメータを Table 2 に示す。なお、TPSAP の最高温度は最大改悪を 50% 以上の確率で受理する温度とし、最低温度は解交換周期内で最小改悪を 1 回は受理する温度とした。

Table2 各手法のパラメータ

パラメータ	クーリング SAP	TPSAP
最高温度	129	129
最低温度	0.28	0.28
クーリング周期	10,000	-
解交換周期	-	40
総探索回数	80,000	10,000 × 8

5.2 性能比較と評価

Fig. 4 に、各手法を 30 試行行った時の最良値における評価値の昇順に並べたものを示す。なお、Fig. 4 は、横軸に試行回数、縦軸に評価値をそれぞれ示している。ただし、TPSAP における最良値とは、最低温度の探索において得た最も良好な解とする。

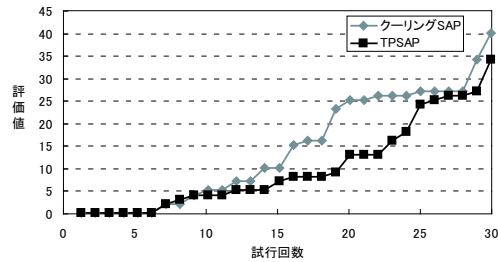


Fig.4 各手法の性能比較 (出典:自作)

Fig. 4 より、TPSAP もクーリング SAP も最適解を得ている。しかし、30 試行において、TPSAP の方がより良好な解を得ることができており、TPSAP の方が高精度である結果を得た。

5.3 考察

前節で示したとおり、TPSAP の方がクーリング SAP よりも高精度であった。TPSAP では、各温度におけるプロセッサ間で解交換を行うため、良好な探索を行うことができたと考えられる。そこで、TPSAP で得た最良解の遷移履歴をとり、Fig. 5 に示す。

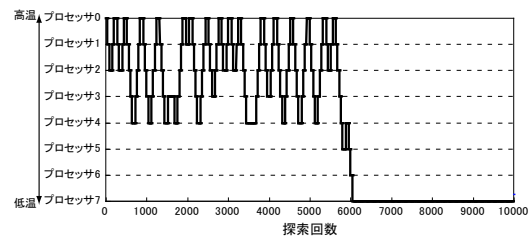


Fig.5 最適解の遷移履歴 (出典:自作)

Fig. 5 より、探索序盤では高温のプロセッサ間で遷移を行っている。また、探索中盤で低温のプロセッサへと遷移していき、終盤では最も低温のプロセッサで探索を行っている。そして、低温のプロセッサから高温のプロセッサへの遷移も多く見られる。これは、局所解を抜け出すためと考えられる。このことから、TPSAP では適応的な温度スケジュールによって良好な探索が行えたと言える。したがって、TPSAP の有効性を示すことができた。今後は異なる温度での比較を行ってみたい。

参考文献

- 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を用いた自動プログラミング, 情報処理学会研究報告 Vol.19 pp.89-92, 2007
- 小西 健三, 瀧 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, 4月 1995