

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

Discovery of Swarm Intelligence with Simulated Annealing Programming

藤田 佳久

Yoshihisa FUJITA

Abstract: In this paper, I propose Simulated Annealing Programming(SAP) which is the new algorithm of discovering Swarm Intelligence. SAP is algorithm in solving a program discovery problem, extending Simulated Annealing(SA) can solve tree structure problems. So, I experimented the comparison of SAP and Genetic Programming(GP). Consequently, SAP was found that it prevents bloat which is the demerit of GP.

1 はじめに

近年、我々の社会では1台のPCやロボットの能力を超える大きな仕事が求められている。例えば、環境の変化に適応的に対応するシステムやロボットが要求されている。このような背景から、PCやロボットが複数で協調を行い、最適な動作を行う群知能研究が注目を浴びている。群知能とは、単純な知能を持つエージェントが集団となることで創発してくる知能のことである。創発とは、個々のエージェントが環境に適応して進化することで、全体としてエージェントの協調行動が自発的に生じることである。これまで協調行動を創発させるための戦略は人によって考え出されてきた。しかし、より複雑で動的な問題を解決しようとするとき、人による協調行動の実装は非常に困難になる。また、あらかじめ想定した状況にしか対応できない。

そこで、これらの問題を解決するために、エージェントの行動規則を自動生成する遺伝的プログラミング(Genetic Programming:GP)が提案されている。GPとは、最適化手法の1つである遺伝的アルゴリズム(Genetic Algorithm:GA)を構造的な表現が扱えるように拡張した進化論的計算手法である。GPでは遺伝的操作(選択, 交叉, 突然変異)を繰り返し行うことで目的とする行動規則を生成する。

本研究では、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)を提案する。SAPとは、金属の焼きなましを模倣した最適化手法であるシミュレーテッドアニーリング(Simulated Annealing:SA)を構造的な表現が扱えるように拡張した計算手法である。SAPでは突然変異, 受理判定を繰り返し行うことで目的とする行動規則を生成する。またマルチエージェント問題を対象問題として、GPとの比較実験を行った。

2 SAP

SAPのアルゴリズムをFig. 1に示す。SAPのアルゴリズムはSAと同様である。

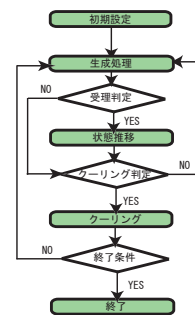


Fig. 1 SAPのアルゴリズム

SAPの生成処理をFig. 2に示す。

1. 突然変異点を選択
2. 突然変異点からの部分木を削除
3. 突然変異木を生成し、これを突然変異点に挿入

これらの操作によりSAPは次状態の生成を行う。

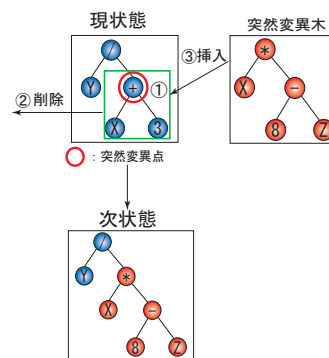


Fig. 2 SAPの生成処理

3 対象問題

対象問題は、マルチエージェント問題である Fig. 3 に示すエサ集め問題とした。問題設定を以下に示す。

3.1 問題設定

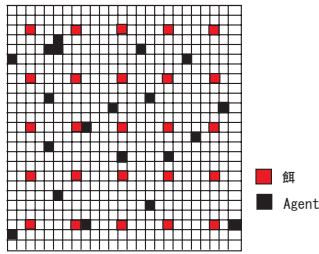


Fig. 3 エサ集め問題のフィールド

- フィールド：25 × 25 の格子状
- エサ
 - エサ数：25 個
 - 同じ場所にエサは重ねて置けない
 - 初期位置は Fig. 3 に示す通り規則的に配置
- エージェント
 - エージェント数：20
 - 同じ場所にエージェントは重ねられない
 - 初期位置と向きはランダムに配置
 - 8 方向に移動可能
 - 全体の状況把握は不可
 - エサを持ち運んでいる場合はエサがある地点には移動できない（エサが重ねられないルールに反するため）
- 行動規則：Homogeneous（全てのエージェントが同じ行動規制）
- 終了条件：非終端記号，終端記号の評価回数が 3000 回の時

3.2 評価関数

エサ集め問題の評価関数を以下に示す。評価はエサの密集度で行う。評価関数は 0 を最適解とする最小化問題である。

$$EvaluationValue = 146 - \sum_{i=1}^{25} (\text{周囲にあるエサ数})(1)$$

3.3 非終端記号

用いた非終端記号は以下の通りである。() の中の数字は引数の数である。

- IF_FOOD_HERE(2)
 - 現在位置にエサがあれば第 1 引数を，なければ第 2 引数を実行

- IF_CARRYING_FOOD(2)
 - エサを保持していれば第 1 引数を，してなければ第 2 引数を実行
- IF_FOOD_ADJACENT(2)
 - 周囲にエサがあればその方向を向き第 1 引数を，なければ第 2 引数を実行
- PROG2(2)
 - 第 1 引数を実行して，第 2 引数を実行（順に実行）

3.4 終端記号

用いた終端記号は以下の通りである。

- RIGHT
 - 45 度右を向く
- LEFT
 - 45 度左を向く
- MOVE
 - 前に一步前進
- PICK_UP
 - エサを拾う
- DROP_FOOD
 - エサを置く

4 数値実験

対象問題における SAP のパラメータ検討を行った。検討パラメータは、最高温度、最低温度である。また、ロバスト性の向上についての検討と群知能の発現についての検討を行った。本実験で用いたパラメータを Table 1 に示す。また、エージェントの初期位置は試行ごとに変えた。

Table 1 SAP パラメータ

総アニーリング回数	25500
クーリング回数	51
最大の木の深さ	20
試行回数	30

4.1 最高温度パラメータの検討

4.1.1 実験概要

三木らの報告²⁾によれば、SA では必要以上に高温での探索は無駄であり、最高温度は重要温度領域より少し高い値であれば十分であると報告されている。このことより、一定温度による探索を行い、重要温度領域の検討を行った。そして、重要温度領域から最適な最高温度を導く。

4.1.2 実験結果

Fig. 4 に解探索能力の比較結果を示す。Fig. 4 は横軸に一定温度、縦軸に評価値を示している。

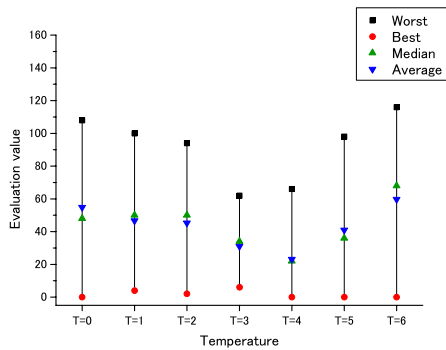


Fig. 4 一定温度での解探索能力の比較

Fig. 4 より、温度 3~5 付近が重要温度領域と考えられる。よって最高温度は 5 が適切だと考えられる。

4.2 最低温度パラメータの検討

4.2.1 実験概要

最高温度パラメータを 5.0 に固定して、最低温度パラメータの検討を行った。

4.2.2 実験結果

Fig. 5 に解探索能力の比較結果を示す。Fig. 5 は横軸に最低温度、縦軸に評価値を示している。

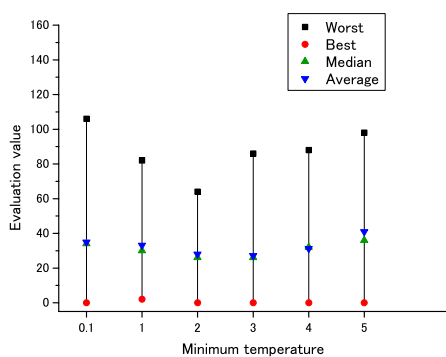


Fig. 5 最低温度での解探索能力の比較

Fig. 5 より、最低温度が解探索に与える影響は少ないことが分かる。つまり、エサ集め問題においては最高温度（重要温度領域）が重要であると考えられる。よって今後の温度パラメータは重要温度（一定温度）4 で探索を行っていく。

4.3 ロバスト性の検討

4.3.1 実験概要

様々な環境において行動できる行動規則の生成を行うため、ロバスト性の検証を行った。比較対象は、エー

ジェントの初期位置を試行ごとに变化させた場合（手法 1）とクーリングステップごとに变化させた場合（手法 2）である。ロバスト性の評価方法は、獲得された最良木（行動規則）をランダムに生成したエージェントのフィールド（50 フィールド）に対して適用した時の評価値を用いた。

4.3.2 実験結果

その結果を Fig. 6 に示す。Fig. 6 では横軸に手法、縦軸に評価値を示している。手法 1 は試行ごとにエージェントの初期位置を变化させた手法、手法 2 はクーリングステップごとにエージェントの初期位置を变化させた手法である。

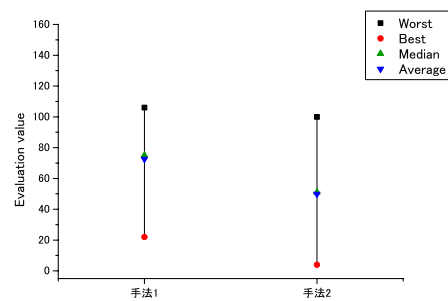


Fig. 6 ロバスト性の比較

Fig. 6 より、解探索途中でエージェントの初期位置を变化させることによって、ロバスト性が向上することが分かった。

4.4 群知能の発現について

4.4.1 実験概要

SAP が群知能を発現するものであるか検証を行う。実験方法は、SAP を用いて得られた行動規則が環境の変化に対応しながら目的の行動が発現するかを確かめる。

4.4.2 実験結果

Fig. 7 に SAP を用いて生成した行動規則でエサ集め問題を実行した時の行動履歴を示す。Fig. 7 は横軸に非終端・終端記号の評価回数、縦軸に評価値を示している。

Fig. 7 より、SAP で得られた行動規則が周りの環境の変化に対応しながら目的の行動を達成できていることが分かる。上記より、SAP を用いることで群知能を発現できることが確かめられた。

Fig. 8 に実際に SAP で獲得されたエサ集めの行動規則を示す。

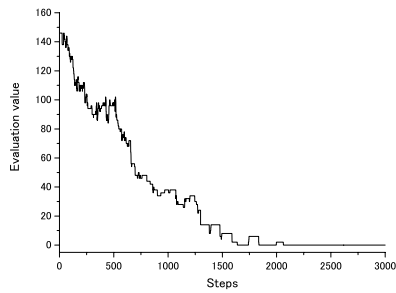


Fig. 7 エサ集めの行動履歴

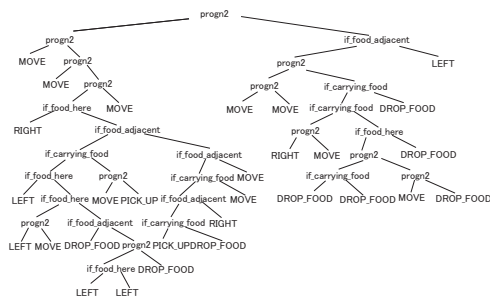


Fig. 8 SAP で獲得されたエサ集めの行動規則

5 GP との比較

5.1 実験概要

SAP と同じくエージェントの行動規則を自動生成する手法である GP と SAP の比較を行った。比較項目は、解探索性能、解探索速度、ロバスト性とした。GP のパラメータは最適とされる Table 2 のパラメータを用いた。またエージェントの初期位置は世代ごとに変えた。

Table 2 GP パラメータ

世代数	50
個体数	500
選択手法	トーナメント選択
トーナメントサイズ	10
交叉率	0.9
突然変異率	0.1
最大の木の深さ	20
試行数	30

5.2 実験結果

5.2.1 解探索性能

Fig. 9 に解探索性能の比較結果を示す。Fig. 9 は横軸に手法、縦軸に評価値を示している。

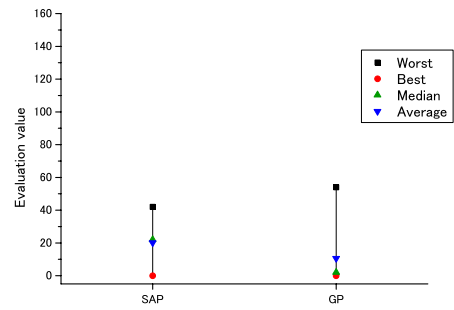


Fig. 9 GP との解探索性能の比較

Fig. 9 より、SAP、GP とともに 30 回試行を行えば最適解が得られることが分かる。しかし、解探索性能の効率の点から見れば、GP の方が効率の良い解探索を行っていることが分かる。

5.2.2 解探索速度

Fig. 10 に 30 回試行の中央値の解探索履歴を示す。Fig. 10 は横軸に評価計算回数、縦軸に評価値を示している。

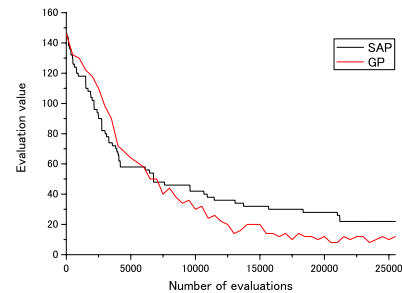


Fig. 10 解探索履歴

Fig. 10 より、GP の方が解探索速度が速いことが分かる。また、得られた評価値も GP の方が良好だと分かった。しかし、SAP においてアニーリング回数を増やすことによって、GP とほぼ同等の解探索性能が出せる可能性があることも Fig. 10 は示している。この点については今後検討を行っていく。

5.2.3 ロバスト性

ロバスト性の評価方法は、獲得された最良木（行動規則）をランダムに生成したエージェントのフィールド（50 フィールド）に対して適用した時の評価値を用いた。その結果を Fig. 11 に示す。Fig. 11 では横軸に手法、縦軸に評価値を示している。

Fig. 11 より、GP で生成された行動規則の方が SAP よりロバスト性が高い行動規則を生成していることが分かった。

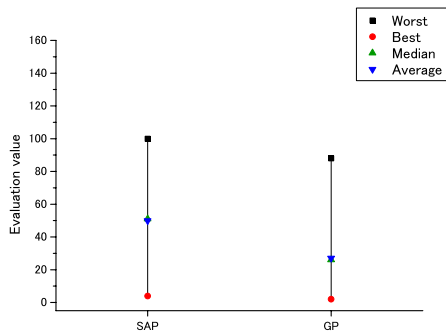


Fig. 11 GP とのロバスト性の比較

5.3 考察

GP, SAPなどを適用した結果, 得られる木としてはなるべく浅い方が良い. なぜなら, 木の深さがあまりにも深くなりすぎると, 解を獲得する前に計算資源を使い切ってしまうためである. この問題をプロート現象といい, GPにおける問題点として知られている.

Fig. 12に評価計算回数ごとに得られた木の平均の深さの推移を示す. ただしFig. 12は, 最大の木の深さを20に制限をして探索を行った結果である. Fig. 12より, GPは解探索中盤にはすでにプロート現象を起こしてしまっているが, SAPではプロート現象を回避できていることが分かる. プロート現象は, 問題の大規模化・複雑化に伴い増大する傾向があるので, 大規模・複雑問題ほどSAPは, この点でGPより優れた結果を示すことができると期待できる.

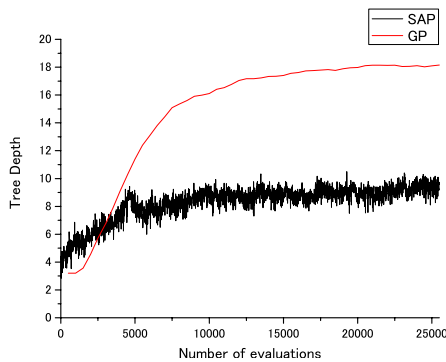


Fig. 12 木の深さの推移

6 まとめ

GPによる群知能の発現については多くの研究が行われているが, SAを用いたプログラミング(ここではSAPと呼ぶ)は提案されているものの, その有効性をあまり検証されていない. そのため今回の発表では, SAPとGPを比較し, SAPがどの程度群知能の発現に有効であるか検証を行った. その結果, 解探索性能, 解探索

速度, ロバスト性に関してはGPの方が勝っていたが, SAPを用いることでGPの問題点であるプロート現象の回避が期待できることが分かった. 今後, この点に関してさらに検証を行う予定である.

参考文献

- 1) Una-May O'Reilly, Franz Oppacher, Program Search with a Hierarchical Variable Length Representation: Genetic Programming, Simulated Annealing and Hill Climbling, Lect Notes Comput Sci Vol.866, Page397-406(1994)
- 2) 三木 光範, 廣安 知之, 實田 健, 適応的最高温度を持つシミュレーテッドアニーリング, 情報処理学会論文誌 Vol.44, No.11, Page2787-2795(2003)
- 3) 三田 英之, 石塚 満, 伊庭 斉志, 遺伝的プログラミングを用いた脱出問題の解法
- 4) Wolfgang Banzhaf, Peter Nordi, Robert E. Keller, Frank D. Francone, 遺伝的プログラミング, 科学技術出版 (2000)
- 5) 伊庭 斉志, 遺伝的プログラミング入門, 東京大学出版会 (2001)
- 6) 伊庭 斉志, 遺伝的プログラミング, 東京電機大学出版局 (2001)