

A-038

シミュレーテッドアニーリングプログラミングによるロボット行動の自動獲得 — 遺伝的プログラミングとの性能比較 —

Automatic Acquisition of Robot Action by Simulated Annealing Programming - Performance Comparison with Genetic Programming -

藤田 佳久[†] 三木 光範[‡] 橋本 雅文[‡] 廣安 知之[‡]

Yoshihisa FUJITA Mitsunori MIKI Masafumi HASHIMOTO Tomoyuki HIROYASU

1. はじめに

ロボットの行動を自動獲得する研究として、遺伝的プログラミング (Genetic Programming: GP)[1] を用いた研究が数多く行われている。GP は、交叉によって解プログラムの一部に相当する小さな部品が組み合わされることにより良好なプログラムを獲得することができる。しかし、GP を適用する際には探索が進むにつれてプログラムのサイズが爆発的に増加するプロトが問題になる。プロトにより探索能力の低下を招いたり、冗長なプログラムが生成されることが知られている。このようなプロトが生じる原因として、交叉や選択であるという報告がされている [2]。

そこで筆者らは、上記の問題を解決するためにシミュレーテッドアニーリング (Simulated Annealing: SA) を用いてロボットの行動を自動獲得す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) を提案した [3]。本報告では、人工蟻の探索問題 (Santa Fe Trail) とマルチエージェントによるエサ集め問題を対象問題として SAP と GP の性能比較を行う。

2. シミュレーテッドアニーリングプログラミング

シミュレーテッドアニーリングプログラミング (SAP) は、金属の焼きなましを模倣した SA を木構造が扱えるように拡張したアルゴリズムである [3]。SAP は、1つの木構造に対して次々と状態を変化させ、推移していくことにより最適な木構造に状態を近づけていく。推移は、次の解候補が改良方向へ生成された場合には無条件でその推移を認め、改善方向へ生成された場合にも温度とよばれる制御パラメータにより確率的にその推移を認めるメカニズムを持つ。これにより、局所解をもつ問題の場合でも最適な木構造を生成することが期待できる。なお、本研究での SAP の温度スケジュールは、一定温度のみの温度スケジュール [4] を用いた。その理由としては、通常の温度スケジュールより良好な探索を示したからである。提案手法である SAP のアルゴリズムについて説明する。

STEP 1 初期設定

ランダムに1つの木構造を生成し、その評価値を計算する。ただし、木構造の生成方法は GROW とし、初期ノードは非終端記号とする。

STEP 2 生成処理

ランダムにノードを1つ選択し、選択されたノードを根とする部分木を削除する。そして、削除された場所に新たな部分木をランダムに生成する (図 1)。これを次状態として次状態における評価値を計算する。SAP の生成処理は GP の突然変異と同様の処理を行うことになる。

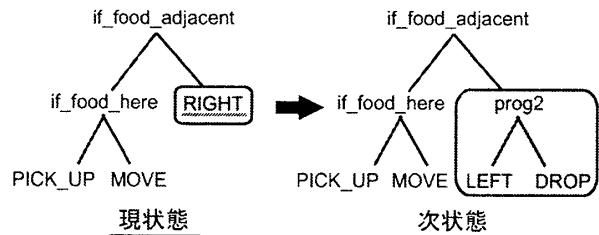


図 1: SAP の生成処理

STEP 3 受理判定, 状態推移

次状態の評価値と現状状態の評価値との差分 ΔE および温度 T を基に、次状態への推移を受理するか否かを判定する。受理判定には Metropolis 基準 (式 1) を用いる。受理の場合は、次状態を推移させる。

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

STEP 4 終了判定

指定した評価計算回数に到達すれば終了。そうでなければ STEP2 へ戻る。

3. 対象問題

提案手法である SAP と GP の性能比較を行うために、以下の2つの対象問題による評価を行った。

(1) 人工蟻の探索問題 (Santa Fe Trail)

人工蟻の探索問題 [5] は GP のテスト問題としてよく知られたものである。この問題は人工蟻 (Ant) が Santa Fe Trail と呼ばれる格子座標上に散乱している餌を限られたエネルギー内で全て獲得するプログラムの生成を目的としている。

Ant の初期エネルギーは 400 とした。また、評価値は餌の総数である 89 から餌を獲得した数を引いたものとした。

(2) マルチエージェントによるエサ集め問題

マルチエージェントによるエサ集め問題は伊庭のエサ集め問題 [5] に変更を加えたものである。この問題は複数のエージェントによって限られた時間内にエサを集めるプログラムの生成を目的としている。設定した環境は以下の通りである。

- フィールド: 25 × 25 の格子状

[†]同志社大学大学院 工学研究科

[‡]同志社大学 工学部

- エサ
 - エサ数は 25 個とする
 - 同じ場所にエサは重ねて置けない
 - 初期位置は図 2 に示す通り規則的に配置
- エージェント
 - エージェント数は 20 とする
 - 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する
 - 8 方向に移動可能とする
 - 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
- 行動規則: Homogeneous(全エージェント均一)
- 終了条件: 終端・非終端記号の実行回数が 3000 回の時とする

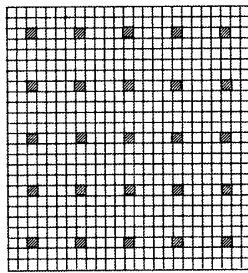


図 2: エサ集め問題の初期配置

エサ集め問題に用いた終端・非終端記号を表 1 に示す。表 1 において、非終端記号は小文字で、終端記号は大文字で表し、非終端記号の引数はすべて 2 である。

表 1: エサ集め問題に用いた終端・非終端記号

記号	機能
if_food_here	現在の位置にエサがあるかどうか
if_carrying_food	エサを持っているかどうか
if_food_adjacent	周囲にエサがある場合その方向を向く
prog2	第 1 引数, 第 2 引数の順に実行
RIGHT	45 度右を向く
LEFT	45 度左を向く
MOVE	一步前進する
PICK_UP	エサを拾う
DROP_FOOD	エサを置く

評価はエサの密集度 C で行う。エサの密集度 C は式 (2) とする。ここで、 N はエサの周囲にあるエサの数、 n はエサ数である。

$$C = \sum_{i=1}^n (N) \quad (2)$$

評価関数は式 (3) とし、0 を最適解とする最小化問題とする。ここで、 C_{\max} はエサの密集度の最大値であり、本発表ではエサ数を 25 にして実験を行ったため C_{\max} は 146 となる。

$$EvaluationValue = C_{\max} - C \quad (3)$$

4. 数値実験

対象問題において SAP と GP の比較実験を行った。比較項目は、学習性能とノード数とした。人工蟻の探索問題の実験結果を図 3 に示す。結果は 50 回試行の平均値である。マルチエージェントによるエサ集め問題の実験結果を図 4 示す。結果は 30 回試行の平均値である。用いたパラメータは予備実験より求めた最適なパラメータ

を用いた。人工蟻の探索問題では、SAP は一定温度 4 を用い、GP は個体数 1000、トーナメントサイズ 2、交叉率 0.9、突然変異率 0.1、最大の深さ 7 を用いた。マルチエージェントによるエサ集め問題では、SAP は一定温度 4.5 を用い、GP は個体数 2000、トーナメントサイズ 2、交叉率 0.9、突然変異率 0.1、最大の深さ 9 を用いた。

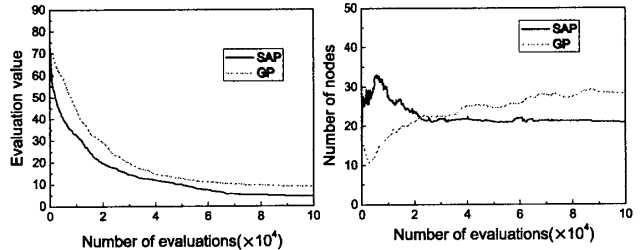


図 3: 人工蟻の探索問題 (左:評価値, 右:ノード数)

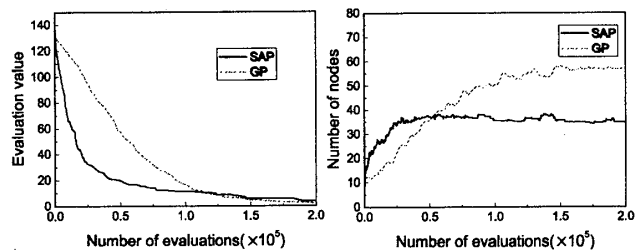


図 4: エサ集め問題 (左:評価値, 右:ノード数)

図 3 より、提案手法である SAP と GP を比較すると SAP の方が性能が良く、かつノード数の少ないプログラムを生成していることがわかる。また、SAP は GP の問題点であるプロットが生じないことがわかる。図 4 より、マルチエージェント環境においても SAP は GP と比較を行うと同等の性能を得ていて、かつノード数の少ないプログラムを生成していることがわかる。また、SAP で生成されるプログラムのノード数は問題に応じて適応的に決まることがわかる。

5. まとめ

本報告では、シミュレーテッドアニーリングプログラミング (SAP) を人工蟻の探索問題とマルチエージェントによるエサ集め問題に適用し、GP と性能比較を行った。その結果、SAP は GP と同等以上の性能を得ることができ、かつノード数の少ないプログラムを生成することがわかった。以上より、SAP はロボット行動の自動獲得に有効な手法であるといえる。

参考文献

- [1] J.Koza. Genetic Programming, On the Programming of Computers by means of Natural Selection. MIT Press, 1992.
- [2] W.B.Langdon, R.Poli. Fitness Causes Bloat. Second On-line World Conference on Soft Computing in Engineering Design and Manufacturing, pp. 13-22, 1997.
- [3] 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. 情報処理学会全国大会講演論文集, Vol.67th, No.2, pp.299-300, 2005.
- [4] Mark Fielding. Simulated Annealing with an optimal fixed temperature. SIAM J.OPTIM, Vol.11, No.2, 2000.
- [5] 伊庭 育志. 遺伝的プログラミング. 東京電機大学出版局, 1996.