

# 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 Santa Fe Trail 問題のシミュレータ

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といったプログラムや電子回路の設計はコンピュータによって自動設計されるようになってきた。これは、人間があらかじめ想定できないような状況にも対応できるプログラムや、人の手では組むことが容易でないプログラムを自動で生成することができるためである。このプログラムをコンピュータにより自動生成する方法を自動プログラミングと呼び、プログラムは多くの場合、木構造で表現する。自動プログラミングの手法の 1 つとして、我々の研究室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) <sup>1)</sup> の研究を行っている。本発表では、この SAP で生成できる木構造とそのプログラムを実行した際の動作を可視化する。これにより、木構造と動作の関係が容易に把握でき、プログラム探索に有効な部分木の生成方法を考案する際に有用であると考えられる。

## 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、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自動プログラミング手法である。

このため、SAP のアルゴリズムは、以下の通りである。

### 1. 初期解の生成

初期解である木構造をランダムに生成し、その評価値  $E$  を求める。

### 2. 生成処理

現在の木構造に対し、ランダムに挿入点を選択し、その点をルートとする部分木を削除する。その点に新たにランダムに生成した部分木を挿入することで新しい木構造を生成し、評価値  $E'$  を求める。

### 3. 受理判定、状態遷移

現在の木構造の評価値  $E$  と新しい木構造の評価値  $E'$  との差分  $\Delta E (= E' - E)$ 、および温度パラメータ  $T$  を基に、新しい木構造に遷移するかしないかの判定を行う。 $\Delta E \leq 0$  の場合は必ず遷移し、すなわち新しい木構造が改良方向へ生成され、逆に改悪方向へ生成された場合は式 (1) に示すメトロポリス基準に従って確率的に遷移する。

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

### 4. 冷却

部分木の挿入、受理判定と状態遷移を所定の回数繰り返した後に、温度パラメータ  $T$  を小さくする。

### 5. 終了判定

部分木挿入、受理判定と状態遷移、冷却を定められた回数行うことで、探索を終了する。その際に、探索中に得られた解の中で最もエネルギー値  $E$  の小さい木構造を解とする。

## 3 対象問題

本発表では、Santa Fe Trail 問題に着目して木構造のプログラムを生成する。Santa Fe Trail 問題は、自動プログラミングの代表的なベンチマーク問題の 1 つである。

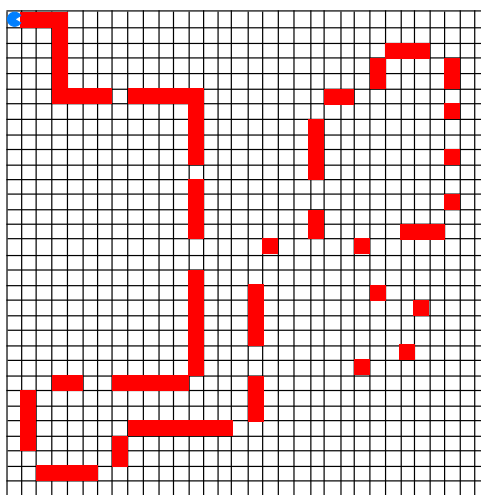


Fig.1 Santa Fe Trail (出典：参考文献<sup>2)</sup>より参照)

Santa Fe Trail 問題とは、Fig. 1 に示したように、1 体の人工蟻が Santa Fe Trail と呼ばれる格子座標上に配置される餌を、限られたエネルギー内に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、終端記号を 1 つ実行するごとにエネルギーを 1 消費する。人工蟻の初期エネルギーは 400 である。この問題に用いる非終端記号は {if-food-ahead, prog2, prog3}、終端記号は {LEFT, RIGHT, MOVE} である。これらの記号とその機能を Table 1 に示す。

Table1 利用する終端記号と非終端記号

記号	機能
if-food-ahead	引数 2 つ。1 マス前方に餌があれば第 1 引数, なければ第 2 引数を実行する。
prog2	引数 2 つ。第 1 引数, 第 2 引数の順に実行する。
prog3	引数 3 つ。第 1 引数, 第 2 引数, 第 3 引数の順に実行する。
LEFT	90 度左を向く。
RIGHT	90 度右を向く。
MOVE	1 マス前進する。

#### 4 シミュレーションの可視化

本システムは, ある探索において, 探索回数を指定すると, その探索回数以降で得られた木構造とその木構造によるシミュレーションを順に表示するものである。

生成された木構造と, その木構造における実際の動作を可視化することで, それらの関係を視覚的にとらえることができる。これにより, 人工蟻の無駄な動作に関わる木構造を把握することができるため, 探索の効率化などにつなげることが期待できる。

本システムの初期画面を Fig. 2 に示す。

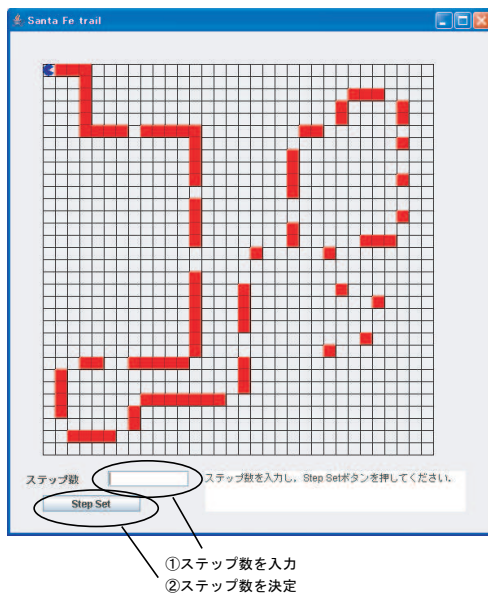


Fig.2 初期画面 (出典: 自作)

Fig. 2 において, 表示させたい木構造のステップ数をテキストボックスに入力し「Step Set」ボタンを押すことで, 指定したステップの木構造が表示される。この画面を Fig. 3 に示す。

表示されている木構造は「Execute」ボタンを押すと, その木構造によるシミュレーションが実行される。そして, このシミュレーションと木構造の表示を数十ステップ分繰り返す。このシステムにより, 各ステップにおけ

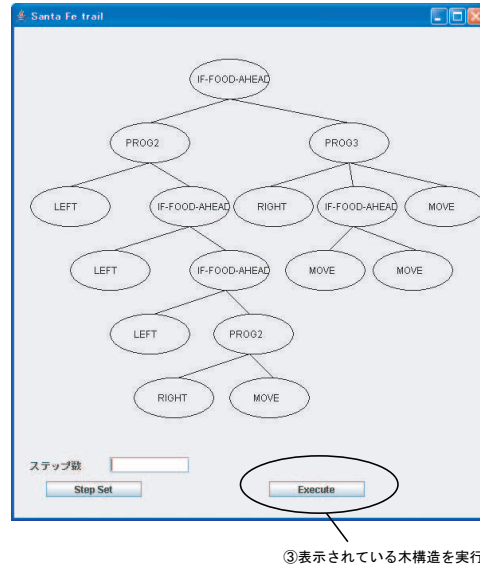


Fig.3 生成される木構造の例 (出典: 自作)

る木構造とその動作の関係をとらえることができる。また, 数十ステップ分繰り返すことにより, 木構造のどの部分が変化したため, 動作がどのように変化したかを視覚的にとらえることも可能となる。

#### 5 まとめ

木構造のプログラムを見ただけでは, その木構造を実行した際の動作にどのような影響を及ぼすか把握することが容易でない。そして, 木構造とその木構造を実行した際のシミュレーション表示するシステムを作成した。これにより, 同じ評価値であっても構造の異なる木構造が存在することや, 木構造によって動作が異なることがわかる。また, 本システムでは, 指定した探索回数以降の構造及びシミュレータの動作をステップごとに表示させる。そのため, 進化の過程においてどの部分木が有効であるか把握しづらかったが, 本システムにより部分木の及ぼす影響が把握しやすくなる。今後の課題として, 有効な部分木の生成方法の検討を行う。

#### 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, 情報処理学会論文誌, Vol. 67, No.2, pp.299-300, 2005
- 2) John R. Koza. *Genetic Programming : On the Programming of Computers by Means of Natural Selection*. MIT Press,1992.