

人工知能

第7回講義課題 課題番号 20

提出締切: 2014.01.09

提出日: 2014.01.09

工学部電子情報工学科

03-123006 岩成達哉

1 概要

本レポートでは、アントコロニー最適化 (ACO) を用いて、N クイーン問題を解くプログラムを作成し、各種パラメータに依る探索の時間やステップの変化を調べた。

2 ACO アルゴリズム

2.1 概要

ACO アルゴリズムは、アリがホルモンを分泌することで、他のアリに道を示すという習性になぞらえたアルゴリズムである。アリは道の中で最も良い物に強くホルモンを残し、他のアリはそのホルモンを元に通路を知ることができる。また、ときにはそのホルモンに従わないアリが現れることで、別の道を模索することができる。

この習性を探索アルゴリズムとして用いることで、良い探索経路を多くたどるようになり、かつ、ときには新しい探索経路を調べることで、他に良い解がないか探することができる。

2.2 手順

N クイーン問題を解く ACO として、参考文献 [1] がある。この手法では、以下の手順で探索が進む。

1. フェロモンの初期化を行う。フェロモンは $N^2[\text{移動元}] \times N^2[\text{移動先}]$ 個だけ用意する。
2. N^2 個のマスに対して、アリを複数匹ランダムに設置する。このとき、同じマスに複数匹置いても良い。
3. アリを一匹ずつフェロモンを元に通路のマスに移動させる。このとき、今までいたことのあるマスには置けない。 i 番目のマスから j 番目のマスに移動する確率は以下ようになる。ただし、 S は今までいたことのないマスの集合、 $\tau_{i,j}$ は i 番目のマスから j 番目のマスへのフェロモン、 $\zeta_{i,j}$ は i 番目のマスから j 番目のマスへ行くヒューリスティックな評価値である。

$$p_{i,j} = \frac{[\tau_{i,j}]^\alpha \cdot [\zeta_{i,j}]^\beta}{\sum_{k \in S} [\tau_{i,k}]^\alpha \cdot [\zeta_{i,k}]^\beta} \quad (1)$$

ヒューリスティックな評価値は、N クイーン問題では例えば、「次にそのマスにアリを移動させることによってクイーンがいくつ死ぬか」などを与えることができる。

4. 3 を $N - 1$ 回繰り返す。
5. 一匹のアリが移動したマスを盤面に投影したものが一つのパターンとなる。全てのアリの経路に対して、解が得られているかを調べる。解でなかった場合は、そのアリが通過した経路すべてのフェロモンを更新する。N クイーンでは例えば、アリが i 番目のマスから j 番目のマスへ移動していた場合、殺したクイーンの数 L 、ある定数を Q とすると、

$$\text{pheromone}[i][j] \leftarrow \text{pheromone}[i][j] + \frac{Q}{L} \quad (2)$$

と更新できる。

6. 全てのフェロモンを蒸発させる。ある定数 ρ を使って、

$$\text{pheromone}[i][j] \leftarrow \text{pheromone}[i][j] \times \rho \quad (3)$$

とあらわせる。これは、5 によってフェロモンが増え続けるのを防ぐ役割がある。

3 プログラムの実行方法

プログラムはC言語で作成した。ソースコードのコンパイルは、Makefileによって行うことができる。具体的には、コマンドプロンプトを用いてソースコードのあるフォルダに移動してリスト1のようにコマンドを実行すれば良い。

リスト 1: make

```
1 $ make
```

実行は、リスト2のように行う。Nには正の整数を与える。これが、Nクイーン問題の問題の大きさとなる。

リスト 2: 実行

```
1 $ ./aco N
```

実行されると、Nクイーン問題の一つの解を探索し、結果を示す。解がない場合は無限ループとなるため、収束しない場合を含めて1000回のステップで終了するようにしている。

4 実験方法

実験は、学科PC(Ubuntu 12.04 [Intel Core i7 @2.40GHz, 8コア, メモリ 16GB])によって行った。

5 結論

本課題では、ACOアルゴリズムを用いて、Nクイーン問題の解を求めるプログラムを作成し、このアルゴリズムが確かに正しい解を求めること、パラメータによってその探索速度が異なることが確かめられた。

参考文献

[1] S. Khan et al. "Solution of n-Queen problem using ACO". In proc. of 13th IEEE International Multi-topic Conference (INMIC 2009), 2009., <http://www.iba.t.u-tokyo.ac.jp/~iba/AI/khan.pdf>

A ソースリスト

ソースのリストを以下に示した。

- Makefile
makeをするためのファイル
- util.c / util.h
汎用的に用いる関数群。メモリの確保などを含んでいる。
- aco.c / aco.h
Nクイーン問題を解く、ACOアルゴリズムを実装したもの。
- main.c
ACOアルゴリズムを呼び出すメイン関数。問題の大きさを入力するなどの機能を持つ ..