

# 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ステップ数や解の求まる確率の変化を調べた。

## 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回道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  $N^2$  個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、殺しあったクイーンの数  $L$ 、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  $\rho$  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。この時の  $\rho$  を持続率、 $1 - \rho$  を蒸発率と呼ぶこととする。

### 3 作成したプログラム

#### 3.1 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 200 回のステップで終了するようにしている．

#### 3.2 工夫した点

今回作成したプログラムでは，ヒューリスティックによって与える値を前述のように「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」によって決めた．具体的には，以下の式によって，式 (1) の  $\zeta_{i,j}$  を求めた．

$$\zeta_{i,j} = \frac{1}{j \text{ 番目に置いた時に死ぬクイーンの数} + 1} \quad (4)$$

+1 を行っているのは，死ぬクイーンの数 が 0 のときに発散することを防ぐためである．

### 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．以下では， $N$  を問題の大きさ， $\alpha$  を式 (1) によるフェロモンの重み， $\beta$  を式 (1) によるヒューリスティックの重み， $\rho$  を式 (3) での蒸発率として用いる．

また，フェロモンの初期値とアリの数は 1000 とし，ステップ数が上限 (200 回) を超えた場合は，局所的最適解に陥ったとして試行を取りやめ，カウントしないことにした．さらに，フェロモンの更新に用いる式 (2) の  $Q$  は 1.0 としている．

#### 4.1 $\rho$ を変化させたときのステップ数

持続率  $\rho$  を変化させ，探索ステップ数の変化をみた．実験方法は以下のとおりである．

1.  $N = 5$   $\alpha = 1.0$   $\beta = 1.0$   $\rho = 0.1$  として ACO アルゴリズムによって探索を行った．
2. 上記を 100 回繰り返した．
3. ステップ数の平均を求めた．
4.  $\rho$  を 0.9 まで 0.1 刻みで変化させて同様の操作を行い，ステップ数の平均を求めた．
5. 以上の操作を  $N$  を 9 まで 1 ずつ変化させて同様の操作を行った．

## 4.2 $\alpha$ を変化させたときのステップ数

フェロモンの重み  $\alpha$  を変化させ、探索のステップ数の変化をみた。実験方法は以下のとおりである。

1.  $N = 5$   $\alpha = 1.0$   $\beta = 1.0$   $\rho = 0.1$  として ACO アルゴリズムによって探索を行った。
2. 上記を 100 回繰り返した。
3. ステップ数の平均を求めた。
4.  $\alpha$  を 2.0 まで 0.25 刻みで変化させて同様の操作を行い、ステップ数の平均を求めた。
5. 以上の操作を  $N$  を 9 まで 1 ずつ変化させて同様の操作を行った。

## 4.3 $\beta$ を変化させたときのステップ数

ヒューリスティックの重み  $\beta$  を変化させ、探索のステップ数の変化をみた。実験方法は以下のとおりである。

1.  $N = 5$   $\alpha = 1.0$   $\beta = 1.0$   $\rho = 0.1$  として ACO アルゴリズムによって探索を行った。
2. 上記を 100 回繰り返した。
3. ステップ数の平均を求めた。
4.  $\beta$  を 2.0 まで 0.25 刻みで変化させて同様の操作を行い、ステップ数の平均を求めた。
5. 以上の操作を  $N$  を 9 まで 1 ずつ変化させて同様の操作を行った。

## 4.4 解の収束率

持続率、フェロモンの重み、ヒューリスティックの重みを変化させ、それぞれでステップ数の上限を超えた回数をカウントし、解が求まらなかった回数がどう変化するかを調べた。実験方法は以下のとおりである。

1. 前述の持続率、フェロモンの重み、ヒューリスティックの重みを変えてステップ数をカウントする実験を行った。
2. ステップ数の上限を超えた回数をそれぞれカウントした。

# 5 結果

## 5.1 $\rho$ を変化させたときのステップ数

持続率  $\rho$  を変化させてステップ数の変化をみた結果を図 1 に示した。この結果から、解を求めるまでのステップ数が小さい 5, 6, 7Queen では持続率の影響は大きくないが、ステップ数が大きくなる 8, 9Queen では、持続率が大きくなるほど、ステップ数が大きくなっていることがわかる。

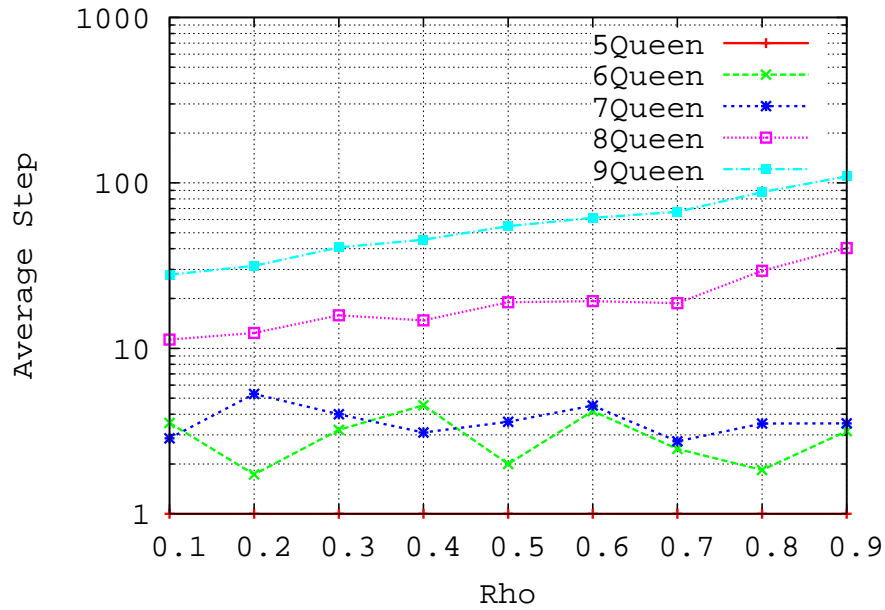


図 1: 持続率  $\rho$  を変化させた結果

## 5.2 $\alpha$ を変化させたときのステップ数

フェロモンの重み  $\alpha$  を変化させてステップ数の変化をみた結果を図 2 に示した．この結果から，解を求めるまでのステップ数が小さい 5, 6, 7Queen ではフェロモンの重みの影響は大きくないが，ステップ数が大きくなる 8, 9Queen では，フェロモンの重みが大きくなるほど，ステップ数が小さくなっていることがわかる．

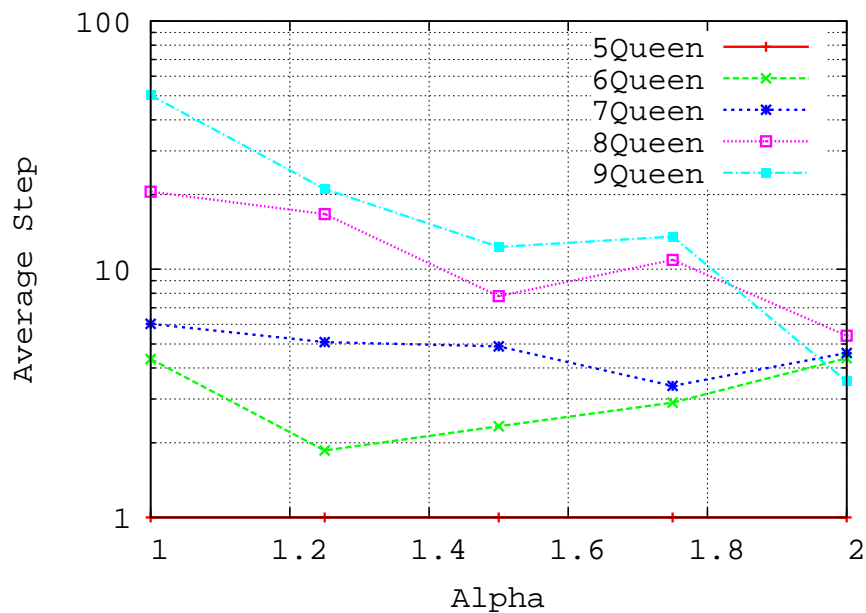


図 2: フェロモンの重み  $\alpha$  を変化させた結果

## 5.3 $\beta$ を変化させたときのステップ数

ヒューリスティックの重み  $\beta$  を変化させてステップ数の変化をみた結果を図 3 に示した．この結果から，ヒューリスティックの重みを大きくすると解が求まるまでのステップ数が小さくなっていることがわかる．

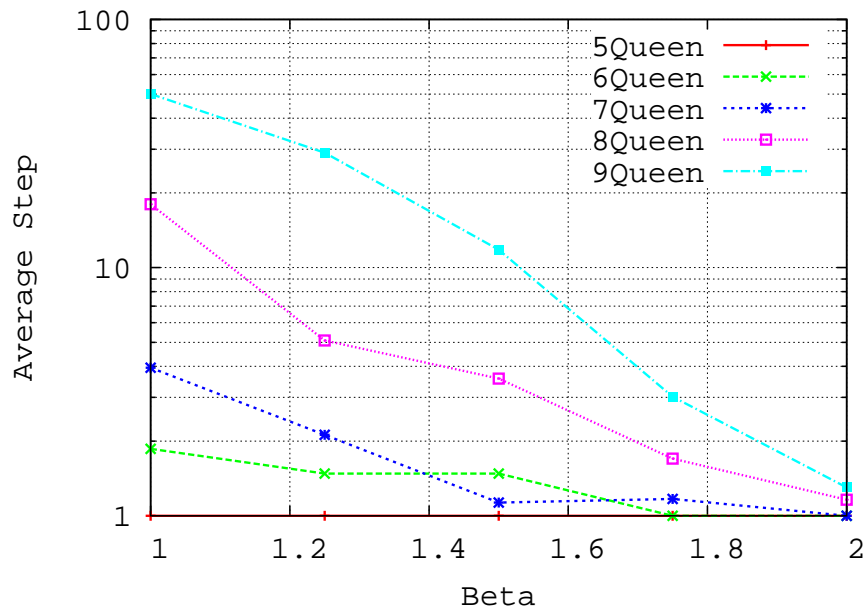


図 3: ヒューリスティックの重み  $\beta$  を変化させた結果

#### 5.4 解の収束率

持続率  $\rho$ , フェロモンの重み  $\alpha$ , ヒューリスティックの重み  $\beta$  を変化させて解が求まらない回数を見た結果を図 4, 5, 6 に示した。この結果から,  $\rho$  と  $\beta$  を大きくすると解が求まる確率が大きくなり,  $\alpha$  を大きくすると解が求まる確率が小さくなっていることがわかる。

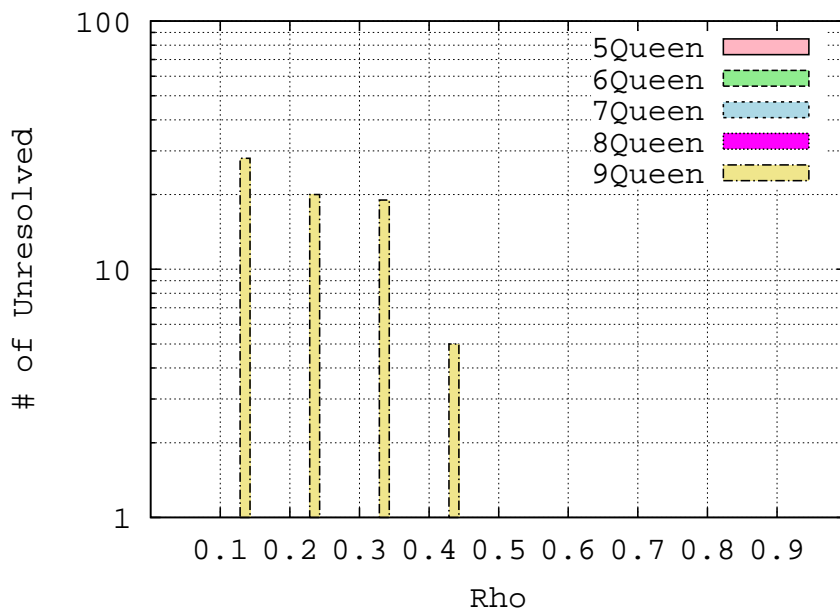


図 4: 解が求まらなかった回数 ( $\rho$ )

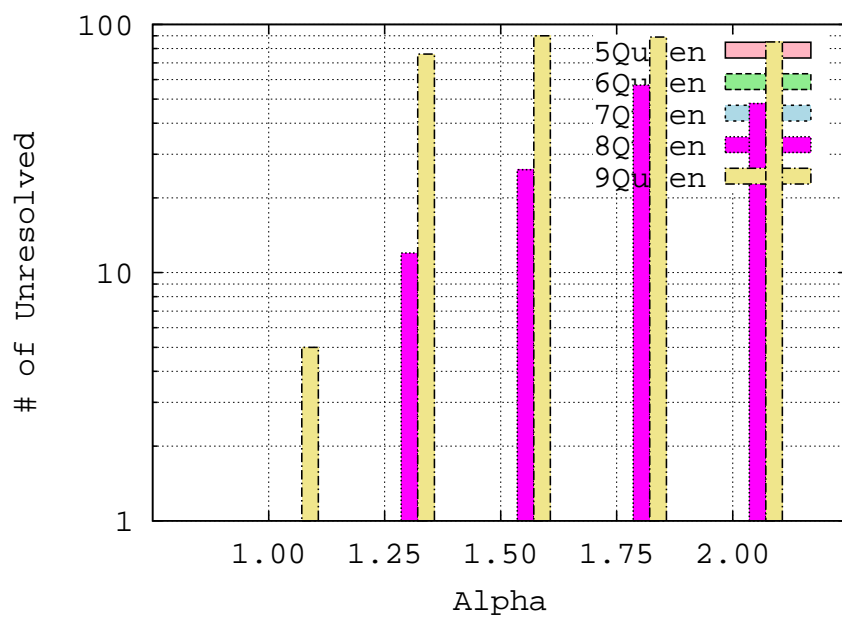


図 5: 解が求まらなかった回数 ( $\alpha$ )

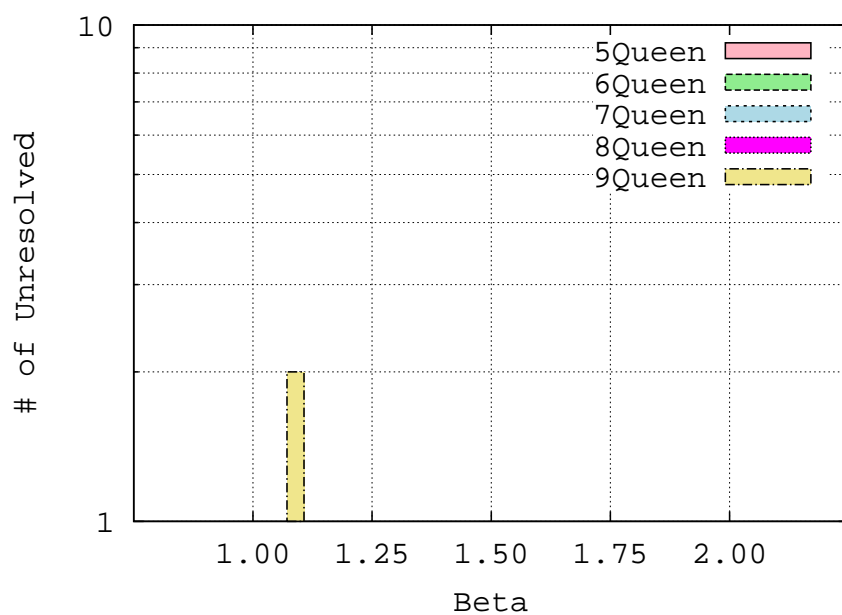


図 6: 解が求まらなかった回数 ( $\beta$ )

## 6 考察

### 6.1 $\rho$ を変化させたときのステップ数

持続率  $\rho$  を変化させたときのステップ数の変化が、持続率が大きくなるほどステップ数が大きくなっていることがわかった。したがって、蒸発率を大きくすることでステップ数が減ることがわかった。これは、持続率を上げるほど、全体のフェロモンの値が大きくなり、フェロモンの更新の際に加えた値の影響がなかなか大きくなりえないためであると考えられる。

これは、前回の課題の焼きなまし法で、温度をゆっくりと下げていく様子に類似している。温度を下げるのがゆっくりであるほど、局所的に良い解に陥ることは少なくなるため、最適解が求まらないということは少なくなるが、試行回数が増えてしまうと推察できる。

また、 $N$  が小さいときに、ステップ数には大きな影響を与えなかったことは、たまたま解けてしまった場合あるいは時間がかかってしまった場合があり、誤差が大きくなったためと考えられる。

## 6.2 $\alpha$ を変化させたときのステップ数

フェロモンの重み  $\alpha$  を変化させたときのステップ数の変化は、フェロモンの重みが大きくなるほど小さくなっていくことがわかった。これは、収束する場合に限って、アリのフェロモンを大きくすることで少ないステップで解が求まっただけであり、間違っただけの局所的な最適解に陥ると収束までに時間がかかってしまうと推察される。

## 6.3 $\beta$ を変化させたときのステップ数

ヒューリスティックの重み  $\beta$  を大きくすると、ステップ数が大きく減少することがわかった。これは、アリの移動の際に、ヒューリスティックの値が大きく影響することになるからである。例えば、Queen が死ぬ数が 0 のときと 1 のときでは、ヒューリスティックの値はそれぞれ  $1/1 = 1$ ,  $1/2 = 0.5$  となる。 $\beta$  の値を大きくすると、前者は 1 のままであるが、後者は 0 に近づいていく。これを移動する確率にかけることになるため、ヒューリスティックの重みが大きの場合、Queen が死ぬ数が 0 のセルに移動する確率はそのまま、死ぬ数が多いセルほど移動する確率が 0 に近づく。

つまり、バックトラックのような処理をアリの数だけ行うことになる。NQueen のような問題では、巡回セールスマン問題などとは異なり、コスト (NQueen ではクイーンが死ぬ数、巡回セールスマン問題では道の長さ) が大きくなる経路は解とはならないため通るべきではない。よって、バックトラックのようになる  $\beta$  を大きくする、手法がよりステップ数を減らせたといえる。

本課題では、問題のサイズが大きくなると時間が大きくかかるために、複数回の実験を行うことはできなかったが、 $\beta$  を更に大きくして、実験を数回行った結果、以下のような結果を得た。

- $N = 10, \rho = 0.300000, \alpha = 1.000000, \beta = 10.000000$  の場合  
100 回試行して平均 1.0 ステップ
- $N = 20, \rho = 0.300000, \alpha = 1.000000, \beta = 10.000000$  の場合  
10 回試行して平均 1.1 ステップ
- $N = 30, \rho = 0.300000, \alpha = 1.000000, \beta = 10.000000$  の場合  
5 回試行して平均 2.2 ステップ
- $N = 40, \rho = 0.300000, \alpha = 1.000000, \beta = 10.000000$  の場合  
1 回試行して 14 ステップ

このように、 $\beta$  が大きければ非常に少ないステップで解にたどり着くことが確かめられた。

## 6.4 解の収束率

持続率  $\rho$ 、フェロモンの重み  $\alpha$ 、ヒューリスティックの重み  $\beta$  を変化させて解が求まらない回数を見た結果、 $\rho$  と  $\beta$  を大きくすると解が求まる確率が大きくなり、 $\alpha$  を大きくすると解が求まる確率が小さくなっていくことがわかった。

$\rho$  の場合は、6.1 節でも述べたように、 $\rho$  を大きくすることで焼きなまし法でゆっくりと温度を下げていくことと同様の操作となり、ステップ数は増えてしまうが大局的最適解の求まる確率は大きくなるからであると考察できる。

$\alpha$  の場合は、6.2 節でも述べたように、 $\alpha$  を大きくすると、ステップ数の上限までで収束する場合は、たまたま良い解へ向かっていったためであり、実際は大局的最適解に到達できる確率が小さくなっていくことがわかる。

$\beta$  の場合は、6.3 節で述べたように、バックトラック法に近い手法を用いるため、最適な解を優先して探すため、大局的最適解にたどり着く確率が非常に大きいと結論できる。

## 7 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ステップが異なることが確かめられた。また、N クイーン問題に対しては、ACO アルゴリズムよりも、バックトラック法のほうが有効であることが示された。



## 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 ..