

# 遺伝的アルゴリズムによる小さなネットワーク検索

SHIMURA Masato  
JCD02773@nifty.ne.jp

2009年5月28日

## 目次

1	ネットワーク検索	1
2	遺伝的アルゴリズム	3
3	自然淘汰	6
4	ベストアンサー	9
5	Reference	11

## 概要

遺伝的アルゴリズム (GA) を小規模なネットワーク検索に適用して GA の特色と効力を検証する。

## はじめに

最近、新幹線や航空機の流体デザインや新薬開発で、決定論的世界ではなく、数多い試行の中から幾つかの有力な候補を絞り込む場合に遺伝的アルゴリズムなどかって人工頭脳と呼ばれた手法を活用する事例を見受けることが多くなった。

ここでは小さなネットワーク検索に適応した遺伝的アルゴリズムを作成してその特性を見てみよう。

## 1 ネットワーク検索

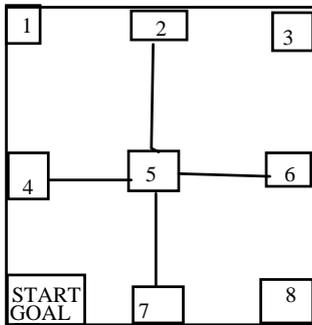
ネットワークノード間の距離計算にはワーシャルフロイド法を用いる。ワーシャルフロイド法の詳細は *Shimura* を参照

小さなネットワークの例 塵収集輸送計画（出典 脇田）

1-8 塵集積場

START/GOAL 処理場

処理場を起終点として1台の収集車が1から8までの集積場を最短距離で回る方法。



- 満杯になれば一旦戻って空にする。
- 満杯を考慮しない。

後者の組み合わせを小さな SCRIPT で実験してみても相当数ある。ここでは後者の場合に限定する。

```
$ ~. 8 rep 20000
```

```
15843 8
```

入力(node0 node1 距離 距離)

距離は1としている。実距離や所要時間を用いることができる。

```
DAT
N1N2D1D2
0 4 1 1
0 7 1 1
1 2 1 1
1 4 1 1
2 3 1 1
2 5 1 1
3 6 1 1
4 5 1 1
5 6 1 1
5 7 1 1
6 8 1 1
7 8 1 1
```

マトリクスへの展開（往復型）

```
(_,i.9),.(i.9),make_mat DAT
_ 0 1 2 3 4 5 6 7 8
0 0 _ _ _ 1 _ _ 1 _
1 _ 0 1 _ 1 _ _ _ _
2 _ 1 0 1 _ 1 _ _ _
3 _ _ 1 0 _ _ 1 _ _
4 1 1 _ _ 0 1 _ _ _
5 _ _ 1 _ 1 0 1 1 _
6 _ _ _ 1 _ 1 0 _ 1
7 1 _ _ _ _ 1 _ 0 1
8 _ _ _ _ _ _ 1 1 0
```

各ノード間の距離、最適評価に用いる。

```
wf make_mat DAT,1
```

距離	最短経路 (直前)
0 2 3 4 1 2 3 1 2	0 4 4 4 0 4 5 0 7
2 0 1 2 1 2 3 3 4	4 1 1 2 1 2 3 4 6
3 1 0 1 2 1 2 2 3	4 2 2 2 1 2 3 5 6
4 2 1 0 3 2 1 3 2	4 2 3 3 2 2 3 5 6
1 1 2 3 0 1 2 2 3	4 4 1 2 4 4 5 0 6
2 2 1 2 1 0 1 1 2	4 2 5 2 5 5 5 5 6
3 3 2 1 2 1 0 2 1	5 3 3 6 5 6 6 5 6
1 3 2 3 2 1 2 0 1	7 4 5 5 0 7 5 7 7
2 4 3 2 3 2 1 1 0	7 6 6 6 6 6 8 8 8

## 2 遺伝的アルゴリズム

遺伝的アルゴリズム (Genetic Algorithm GA) は生物の進化をモデルとした手法で次のような遺伝的操作を用いて問題を解こうというものである。遺伝的アルゴリズムの各手法は *Suganuma* が詳しい。

- 交叉
- 突然変異
- 自然淘汰

遺伝的アルゴリズムは様々な方法が提案されている。本稿は脇谷による簡潔な方法を用いた。ネットワーク検索であるのでデータは全てバイナリーではなく順序数である。

### 2.1 初期設定

#### 2.1.1 グローバル変数の初期値

変数が多いのでグローバル変数を最初に設定する。指定と変更は自由で、フォームに載せれば簡単に扱うことができる。

NB. -----global definition

NODE=: 8 NB. ノード数 designated by data

GENERATION=: 30 NB. 演算する世代の数 assignment freely

EXCHANGE\_RATE=: 40 NB. 交叉率 exchange 40%

MUTATION\_RATE=: 5 NB. 突然変異率 5% assignment freely

### 2.1.2 初期集団の作成

経路の順番を乱数で生成する。Start/Goal は0で、最初と最後に付加するが固定値なので乱数には含めない。

```
NODE=.8 NB. global definition anytime change avilable
```

```
mkRandGeneric 6 NB. 5 random initial data
+-----+
|8 6 7 2 4 1 5 3|
+-----+
|5 4 7 3 6 8 2 1|
+-----+
|3 7 2 1 5 4 6 8|
+-----+
|5 7 8 3 6 1 2 4|
+-----+
|4 2 3 6 5 8 1 7|
+-----+
|7 8 1 3 4 5 6 2|
+-----+
```

```
(Start)0-->8 6 7 2 4 1 5 3-->0(Goal)
```

```
mkRandGeneric=:3 : 0
ANS=.<'
for. i. y do. ANS=. ANS,< NODE ? NODE end.
TMP=.: L:0 ,. }.ANS NB. add 1 to Node
)
```

## 2.2 交叉

- 交叉率によりカップリングする。

```
CAND_EX
+-----+-----+
|8 6 7 2 4 1 5 3|5 4 7 3 6 8 2 1|
+-----+-----+
```

- 乱数により交差部分を決める。

```
pick_exchange_sub CAND_EX
3 4 5
```

3 4 5 番を交叉させる。単純に交叉させると重複が生じる。

```
TMP
+-----+-----+-----+
|0 1 7|2 4 1|8 6 7 3 6 8 5 3|
+-----+-----+-----+
|1 6 7|3 6 8|5 4 7 2 4 1 2 1|
+-----+-----+-----+
IO      EX      GM0
doubled id/ out/simple exchange
```

- 重複した箇所を抜き去った箇所の順に復元する。

交叉は親子の常であり、進歩も退化も起こる。

TMP=. (EX) (I0)} GM0 NB.amend を用いる  
交叉の作用は相当複雑であるが、スクリプトは簡潔である。

```
,2{.exchange_sub TMP
+-----+-----+
|2 4 7 3 6 8 5 1|5 3 7 2 4 1 6 8|
+-----+-----+
```

- 交叉を終えた親の扱い

自然淘汰との関係で2とおりのモデルができる。

1. 交叉を終えた親は子と世代更代する。(更代モデル)
2. 交叉を終えたも残る。(残存モデル)

更代モデルの場合は総数が増えないので自然淘汰で2重コピーで優れたものを増殖させなければならない。

残存モデルの場合は親が残り子が消滅することもある。

2モデルを作成したが顕著な優劣は見受けられないようだ。

```
(100 0) genetic DAT NB. 更代モデル
(100 1) genetic DAT NB. 残存モデル
```

## 2.3 突然変異

- 初期設定の突然変異率に従い突然変異する個体を選び出す。
- 突然変異は個体内の入れ替えである。入れ替える箇所は確率により決定する。
- A(離れた箇所の入れ替え)と、B(連続箇所の反転)の2タイプのいずれかを確率により選択した。

```
CAND_MU,.mutation CAND_MU
+-----+-----+
|5 7 8 3 6 1 2 4|6 7 1 3 5 8 2 4| NB. type A
+-----+-----+
* * * * --> * * * *
```

```
CAND_MU,.mutation CAND_MU
+-----+-----+
|5 7 8 3 6 1 2 4|5 3 8 7 6 1 2 4| NB. type B
+-----+-----+
* * * --> * * *
```

## 3 自然淘汰

### 3.0.1 評価方法と計算

(オリジナルデータで) ルーレット選択を用いて自然選択(適応種が多く生き残る。)を試みる。適応度の良い方にウエイトがかかっているがエリート選択ではない。

- 距離で評価

*ev\_sub* が距離による評価エンジンである。

試しにオリジナル経路でのワーシャルフロイド法で求めた距離を算出する。最短距離は数字が小さい方が優れている。

```
EXAMPLE ev_sub DAT
+-----+
|8 6 7 2 4 1 5 3 18|
+-----+
|5 4 7 3 6 8 2 1 16|
+-----+
|7 2 1 5 4 6 8 0 12| NB.partial best
+-----+
```

```
|5 7 8 3 6 1 2 4 14|
+-----+
|4 2 3 6 5 8 1 7 16|
+-----+
|7 8 1 3 4 5 6 2 18|
+-----+
```

巡回コースを1ステップずつ距離テーブルと照合して全距離を求める

```
ev_sub=:4 : 0
NB. GMAT ev_sub DAT
NB. done mutation and exchange
ROUTE=. 0, L:0 x,(L:0) 0 NB. add node 0 (start goal) to both side
TABLE=. anal_sub0 y
sizeTable=. ; $ TABLE
TMP=. 2{. L:0 TABLE NB. net table
MAX=. .2 + >./; >./ (L:0) TMP NB. add 2
ETABLE=. MAX #. (L:0)/:~(L:0) TMP
ETABLEX=. ~. /:~(;ETABLE),,;{: L:0 TABLE NB. sort and remove pair
GX0=. ;(L:1) MAX #. (L:0)/:~(L:0) 2<\ L:0 ROUTE
IND=. ({."1 ETABLEX) e. L:0 GX0
EVAL=. {: L:0 +/ L:0 EMAT=: IND # L:0 ETABLEX
x, L:0 EVAL NB. minus 1 (where ? added)
)
```

- 上の平均、標準偏差、最短と最長を求める。

```
ev_calc EXAMPLE ev_sub DAT
+-----+-----+-----+
|16.6667|2.22222|14 18|
+-----+-----+-----+
```

### 3.0.2 ルーレット選択

自然淘汰で生き残りを選ぶ方法にはルーレット選択が良く用いられている。

ルーレット選択とは適応度によってウェイトを用いたルーレットを適宜作成し、乱数によりランダムに選択した値 (ex. 0-99) の落ちたゾーンを選択する方法である。

- A 40 0 - 39
- B 20 40 - 59
- C 20 60 - 79
- D 10 80 - 89
- E 10 90 - 99

簡易ルーレットの作成

ポイント（経路距離 短い程良い）と該当個数。

得点と個数を逆転させ優れたポイント（12）の配分個数を多くしている。

得点個数 配分得点

- 12 3 22
- 14 6 20
- 16 27 18
- 18 32 16
- 20 23 14
- 22 9 12

この後配分個数と実個数との照合や残存総数の微差調整には相当の手数が必要である。

```
EXAMPLE ,. ( EXAMPLE select0 DAT) evaluation DAT
+-----+-----+
|8 6 7 2 4 1 5 3|5 7 8 3 6 1 2 4 14|
+-----+-----+
|5 4 7 3 6 8 2 1|5 7 8 3 6 1 2 4 14|NB. duplicate
+-----+-----+
|3 7 2 1 5 4 6 8|5 4 7 3 6 8 2 1 16|
+-----+-----+
|5 7 8 3 6 1 2 4|4 2 3 6 5 8 1 7 16|
+-----+-----+
|4 2 3 6 5 8 1 7|4 2 3 6 5 8 1 7 16|NB. duplicate
+-----+-----+
|7 8 1 3 4 5 6 2|8 6 7 2 4 1 5 3 18|
+-----+-----+
original          sorted and value
```

同じオリジナルデータでルーレット選択を用いて自然選択（適応種が多く生き残る。）を試みる。適応度の良い方にウエイトがかかっているがエリート選択ではない。この例では2個体（左の行の3,5番目）が子孫を残せなかった。

## 4 ベストアンサー

### 4.1 遺伝的アルゴリズムを用いないで最適解の出る確率は

今までのスクリプトを転用してストレートに初期設定に最適解が含まれているかを求める短いスクリプトを作成した。

```
select_best=: 4 : 0
NB. x is number of GA//y is DAT
(\:;{: L:0 TMP0){TMP0=(mkRandGeneric x) ev_sub y
)
```

```
5000 select_best DAT
```

```
|1 2 3 8 6 5 4 7 12|
+-----+
|4 1 2 3 6 5 7 8 10|
+-----+
|7 5 6 8 3 2 1 4 10|
+-----+
|7 8 6 3 1 2 5 4 10|
+-----+
|4 1 2 3 5 6 8 7 10|
+-----+
```

初期値 5000 組内での最適解は数個程度で (0.1%)、出ないこともある。遺伝的アルゴリズムが最適解を求める方法として有効であることの左証となる。

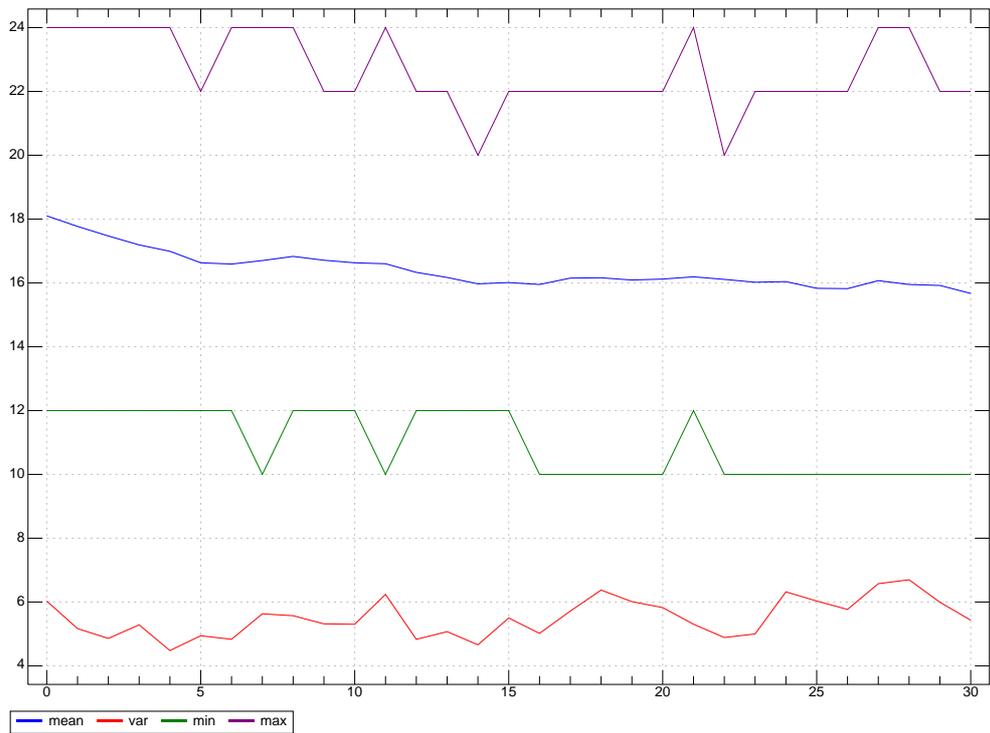
### 4.2 遺伝的アルゴリズムの解

200 個のサンプルを 30 世代の遺伝操作を施して最適解に辿り着く過程をグラフに示した。10 世代以内に最適解 (10) が得られ、比較的良好に保持されている。(x 軸は世代数である)

```
(200 1) genetic DAT
(200 0) plot_genetic DAT
pd 'eps /temp/genetic1.eps'
```

折角であるから GA の最適解を求めるスクリプトを作成し重複を除いて取り上げてみよう。

```
(100 0) pick_best DAT
```



```

8 6 3 2 1 4 5 7 10
7 8 3 6 5 2 1 4 10
4 1 2 3 6 8 7 5 10
4 1 2 3 6 5 8 7 10
4 1 2 3 6 8 5 7 10
4 1 2 5 6 3 8 7 10

```

(各行の最後の値の10は最適解である。)

```
pick_best_sub=: 3 : '<(tmp e. <./ tmp =. ; {: L:0 y) # y'
```

```
pick_best=: 4 : 0
```

```
NB. Usage: 100 pick_best DAT
```

```
~. ;("1) > pick_best_sub ; }."1 x genertcB y
```

```
)
```

```
*1
```

---

\*1 geneticB はベストアンサー用のスクリプト

## 5 Reference

脇田英治 「数値解析のはなし」 技報道出版 1999

[Suganuma sist.ac.jp/~suganuma/kougi/SE/.../GA.htm](http://Suganuma.sist.ac.jp/~suganuma/kougi/SE/.../GA.htm)

Shimura 迷路の木の探索とワーシャルフロイド法 JAPLA 2008.09

J 言語 <http://www.jsoftware.com> から DL

JAPLA [http://homepage3.nifty.com/asagaya\\_avenue](http://homepage3.nifty.com/asagaya_avenue)

APL/workshop