迷路の木の探索とワーシャル・フロイド法

Shimura Masato
JCD02773@nifty.ne.jp

2008 年 8 月 8 日

目次

概要

迷路の木の探索問題をワーシャルフロイド法を用いて最短距離と経路を求める問題としてシンプルにとらえる。

1 データと構造

西川「J による迷路の木の探索ー木構造を基本から考えてみる」(JAPLA 研究会/July2008) の迷路の脱出問題が紹介されている。

迷路の人工頭脳に相当する複雑なアルゴリズムを最短距離を求めるワーシャルフロイド法を用いて最短距離と経路を求める問題としてシンプルにとらえて、解を得る。

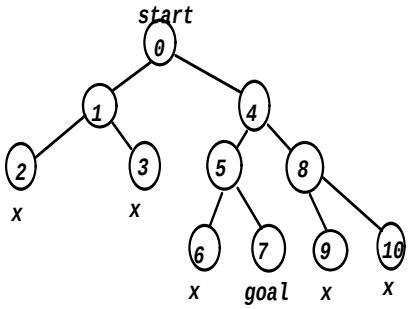


図 1　tree of maze

西川の迷路を *Tree* にしてアルファベットを数字にするとこのようになる。*X* は行き止まり。

ノードをマトリクスに展開し、データのないところを大きな数字（99999) などに置き換える。見やすくなるように無限大_を用いた。

距離は必要ないので、全て 1 ステップとする。往復も全て 1 ステップであるのでマトリクスは対角行列から上の半分を用いればよい。ノードの構成を次のようにする。

```
   DAT0
0  1 1
0  4 1
1  2 1
1  3 1
4  5 1
4  8 1
5  6 1
5  7 1
8  9 1
8 10 1
Y0=:around_nr   make_mat_half DAT0
```

```
 _ 0 1 2 3 4 5 6 7 8 9 10
 0 _ 1 _ _ 1 _ _ _ _ _  _
 1 _ _ 1 1 _ _ _ _ _ _  _
 2 _ _ _ _ _ _ _ _ _ _  _
 3 _ _ _ _ _ _ _ _ _ _  _
 4 _ _ _ _ _ 1 _ _ 1 _  _
 5 _ _ _ _ _ _ 1 1 _ _  _
 6 _ _ _ _ _ _ _ _ _ _  _
 7 _ _ _ _ _ _ _ _ _ _  _
 8 _ _ _ _ _ _ _ _ _ 1  1
 9 _ _ _ _ _ _ _ _ _ _  _
10 _ _ _ _ _ _ _ _ _ _  _
```

2　ワーシャルフロイド法

2.1　マトリクスに展開する

2.2　ワーシャル · フロイド法

Warshall · Floyd 法 は全最短距離を求める効率的な方法として知られる。

計算エンジンは次の部分。(0 行,0 列），(1 行,1 列）,· · · , (m 行, 列 m）と順に取っていき、プラスの内積を作成する。最初に原マトリクスとこれを比較し短い方の距離を採用する。順次繰り返すと一巡したところで最短距離のテーブルができあがる。(*wf_non_loop* はマトリクスのサイズ分繰り返している。)

2.2.1 数式

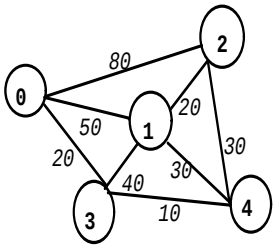
手順と数式は伊理　古林による

手順 0　$d_{ij}^{(0)} = d_{ij}, (i, j = 0, 1, \cdot, m), l = 1$
手順 1　$d_{ij}^{(l)} = min(d_{ij}^{(l-1)}, d_{il}^{(l-1)} + d_{lj}^{(l-1)}, (i, j = 0, 1, \cdot, m)$
手順 2　$d_{ii}^{(l)}$ の中に負のものがあれば終了する。
手順 3　$l = m$ ならば終了する。$l < m$ ならば手順 2 から繰り返す。

```
for_ctr. i.  # Y0  do.
Y0=. ((ctr {"1 Y0)+/ ctr{ Y0) <. Y0
end.
```

2.2.2 Working Example

例題（伊理 P48)




```
   IRI
0 1 50 50
0 2 80 80
0 3 20 20
1 2 20 20
1 3 40 40
1 4 30 30
2 4 30 30
3 4 10 10
```

図 2　iri example

A

```
] a =. make_mat IRI
```

$$\begin{bmatrix} 0 & 50 & 80 & 20 & \_ \\ 50 & 0 & 20 & 40 & 30 \\ 80 & 20 & 0 & \_ & 30 \\ 20 & 40 & \_ & 0 & 10 \\ \_ & 30 & 30 & 10 & 0 \end{bmatrix}$$

最初の原型

B

```
(0{a) +/ 0{"1 a
```

$$\begin{bmatrix} 0 & 50 & 80 & 20\_ & \\ 50 & 100 & 130 & 70 & \_ \\ 80 & 130 & 160 & 100 & \_ \\ 20 & 70 & 100 & 40 & \_ \\ \_ & \_ & \_ & \_ & \_ \end{bmatrix}$$

0 行と 0 列の＋の内積

C

```
((0{a) +/ 0{"1 a)<. a
```

$$\begin{bmatrix} 0 & 50 & 80 & 20 & \_ \\ 50 & 0 & 20 & 40 & 30 \\ 80 & 20 & 0 & \mathbf{100} & 30 \\ 20 & 40 & \mathbf{100} & 0 & 10 \\ \_ & 30 & 30 & 10 & 0 \end{bmatrix}$$

A と B を比較して小さい方に置き換える。太字が置き換え

$$\begin{bmatrix} 0 & 50 & 80 & 20 & \_ \\ 50 & 0 & 20 & 40 & 30 \\ 80 & 20 & 0 & 100 & 30 \\ 20 & 40 & 100 & 0 & 10 \\ \_ & 30 & 30 & 10 & 0 \end{bmatrix}$$

$A_1 = C_0$

```
((1{a) +/ 1{"1 a)
```

$$\begin{bmatrix} 100 & 50 & 70 & 90 & 80 \\ 50 & 0 & 20 & 40 & 30 \\ 70 & 20 & 40 & 60 & 50 \\ 90 & 40 & 60 & 80 & 70 \\ 80 & 30 & 50 & 70 & 60 \end{bmatrix}$$

A_1 の 1 行 1 列の内積

$$\begin{bmatrix} 0 & 50 & \mathbf{70} & 20 & \mathbf{80} \\ 50 & 0 & 20 & 40 & 30 \\ \mathbf{70} & 20 & 0 & \mathbf{60} & 30 \\ 20 & 40 & \mathbf{60} & 0 & 10 \\ \mathbf{80} & 30 & 30 & 10 & 0 \end{bmatrix}$$

A_1, B_1 の小さい方を採択

3 経路

経路は最初の行の右マトリクスを作成し置き換えたときにカウンターの番号と置き換えると、最終経路が記入されている。カウンターの番号はノードの番号と同一である。（同一距離の場合は最後のカウンターが優先する。）

wf_trace は経路の結果を全て記録する。*wf* は最終結果のみ

```
   wf_trace  a
+-+---------------+---------+
|0| 0 50 80 20  _  |0 0 0 0 0|
| |50  0 20 40 30  |1 1 1 1 1|
| |80 20  0  _ 30  |2 2 2 2 2|
| |20 40  _  0 10  |3 3 3 3 3|
| | _ 30 30 10  0  |4 4 4 4 4|
+-+---------------+---------+
|1| 0 50  80  20  _|0 0 0 0 0|
```

```
| |50  0  20  40 30|1 1 1 1 1|
| |80 20   0 100 30|2 2 2 0 2|
| |20 40 100   0 10|3 3 0 3 3|
| | _ 30  30  10  0|4 4 4 4 4|
+-+----------------+---------+
|2| 0 50 70 20 80  |0 0 1 0 1|
| |50  0 20 40 30  |1 1 1 1 1|
| |70 20  0 60 30  |1 2 2 1 2|
| |20 40 60  0 10  |3 3 1 3 3|
| |80 30 30 10  0  |1 4 4 4 4|
+-+----------------+---------+
|3| 0 50 70 20 80  |0 0 1 0 1|
| |50  0 20 40 30  |1 1 1 1 1|
| |70 20  0 60 30  |1 2 2 1 2|
| |20 40 60  0 10  |3 3 1 3 3|
| |80 30 30 10  0  |1 4 4 4 4|
+-+----------------+---------+
|4| 0 50 70 20 30  |0 0 1 0 3|
| |50  0 20 40 30  |1 1 1 1 1|
| |70 20  0 60 30  |1 2 2 1 2|
| |20 40 60  0 10  |3 3 1 3 3|
| |30 30 30 10  0  |3 4 4 4 4|
+-+----------------+---------+
|5| 0 50 60 20 30  |0 0 4 0 3|
| |50  0 20 40 30  |1 1 1 1 1|
| |60 20  0 40 30  |4 2 2 4 2|
| |20 40 40  0 10  |3 3 4 3 3|
| |30 30 30 10  0  |3 4 4 4 4|
+-+----------------+---------+
```

置き換えの経過は次のサマリーで確認できる。

```
   wf_trace_summary  a
+---------+---------+---------+---------+---------+
|0 0 0 0 0|1 1 1 1 1|2 2 2 2 2|3 3 3 3 3|4 4 4 4 4|
|0 0 1 0 1|         |1 0 0 0 0|0 0 0 0 0|1 0 0 0 0|
|0 0 4 0 3|         |4 0 0 1 0|0 0 1 0 0|3 0 0 0 0|
```

```
|         |         |0 0 0 4 0|0 0 4 0 0|         |
+---------+---------+---------+---------+---------+
```

4　西川の迷路の解

4.1　ワーシャル · フロイド法による最短ステップ

Start は 0 *Goal* は 7 と指定されているので、0→7 は 3 ステップが最短である。

```
around_nr wf_non make_mat_half DAT0

 _ 0 1 2 3 4 5 6 7 8 9 10
 0 _ 1 2 2 1 2 3 3 2 3  3
 1 _ _ 1 1 _ _ _ _ _ _  _
 2 _ _ _ _ _ _ _ _ _ _  _
 3 _ _ _ _ _ _ _ _ _ _  _
 4 _ _ _ _ _ 1 2 2 1 2  2
 5 _ _ _ _ _ _ 1 1 _ _  _
 6 _ _ _ _ _ _ _ _ _ _  _
 7 _ _ _ _ _ _ _ _ _ _  _
 8 _ _ _ _ _ _ _ _ _ 1  1
 9 _ _ _ _ _ _ _ _ _ _  _
10 _ _ _ _ _ _ _ _ _ _  _
```

4.2　経路

0→7 の経路は 5 を示している。

```
wf make_mat_half DAT0
+--------------------+--------------------------------+
|_ 1 2 2 1 2 3 3 2 3 3| 0  0  1  1  0  4  5  5  4  8  8|
|_ _ 1 1 _ _ _ _ _ _ _| 1  1  1  1  1  1  1  1  1  1  1|
|_ _ _ _ _ _ _ _ _ _ _| 2  2  2  2  2  2  2  2  2  2  2|
|_ _ _ _ _ _ _ _ _ _ _| 3  3  3  3  3  3  3  3  3  3  3|
|_ _ _ _ _ 1 2 2 1 2 2| 4  4  4  4  4  4  5  5  4  8  8|
|_ _ _ _ _ _ 1 1 _ _ _| 5  5  5  5  5  5  5  5  5  5  5|
|_ _ _ _ _ _ _ _ _ _ _| 6  6  6  6  6  6  6  6  6  6  6|
|_ _ _ _ _ _ _ _ _ _ _| 7  7  7  7  7  7  7  7  7  7  7|
```

```
|_ _ _ _ _ _ _ _ _ 1 1| 8  8  8  8  8  8  8  8  8  8  8|
|_ _ _ _ _ _ _ _ _ _ _| 9  9  9  9  9  9  9  9  9  9  9|
|_ _ _ _ _ _ _ _ _ _ _|10 10 10 10 10 10 10 10 10 10 10|
+--------------------+--------------------------------+
```

5 Reference

伊理正夫 古林隆「ネットワーク理論」日科技連　1976

西川利男　「J による迷路の木の探索ー木構造を基本から考えてみる」(JAPLA 研究会/July2008)

6 Script

6.1 データの入力

,:は層連結/Raminate

DAT0=: 0 1 1,0 4 1,1 2 1,1 3 1,4 5 1,:4 8 1

DAT0=: DAT0,5 6 1,5 7 1,8 9 1,:8 10 1

6.1.1 Script

```
wf_non=: 3 : 0
NB. Calc once//calc engine
NB. Warshall Floyd method
NB. calc with non trace
NB. Usage: wf_non y.
Y0=. y
for_ctr. i.  # Y0  do.
Y0=. ((ctr {"1 Y0)+/ ctr{ Y0) <. Y0
end.
)

around_nr=: 3 : ' (_,i.# y),.  (i. # y),y'
```