

# Network on Network

M.Shimura

JCD02773@nifty.ne.jp

2008年5月28日

## 目次

|   |                  |   |
|---|------------------|---|
| 1 | ワーシャル・フロイト法      | 1 |
| 2 | 高速道路の最短距離のアルゴリズム | 3 |

## 1 ワーシャル・フロイト法

Warshall-Floyd 法は列と行の外積（和）を求め、行列の比較を行い、 $\min$  を採っていく方法である。組み合わせによるフロイド法よりもアルゴリズムが簡潔であり、堅牢である。

### 1.0.1 Matrix sample

伊理ー古林に紹介されているマトリクス

|    |    |    |    |    |
|----|----|----|----|----|
| 0  | 50 | 80 | 20 | -  |
| 50 | 0  | 20 | 40 | 30 |
| 80 | 20 | 0  | -  | 30 |
| 20 | 40 | -  | 0  | 10 |
| -  | 30 | 30 | 10 | 0  |

### 1.0.2 Algolism

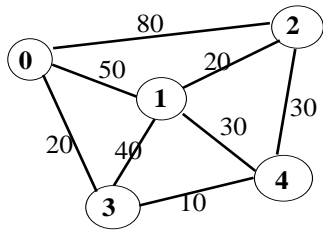
0 列と 0 行の加算の外積（和）をもとめる。

当初の行列と比較し  $\min$  があれば書き換える。（更新）無限大から 1 0 0 に 2 カ所更新される。

$$d_{ij}^1 = \min(d_{ij}^0, d_{i1}^0 + d_{1j}^0)$$

(0{"1 wfd) +/- 0{wfd NB. 0 行と 0 列との加算の外積

|   |    |    |    |   |
|---|----|----|----|---|
| 0 | 50 | 80 | 20 | - |
|---|----|----|----|---|



|    |     |            |            |   |
|----|-----|------------|------------|---|
| 50 | 100 | 130        | 70         | - |
| 80 | 130 | 160        | <b>100</b> | - |
| 20 | 70  | <b>100</b> | 40         | - |
| -  | -   | -          | -          | - |

|    |    |     |     |    |
|----|----|-----|-----|----|
| 0  | 50 | 80  | 20  | -  |
| 50 | 0  | 20  | 40  | 30 |
| 80 | 20 | 0   | 100 | 30 |
| 20 | 40 | 100 | 0   | 10 |
| -  | 30 | 30  | 10  | 0  |

1 列と 1 行の加算の外積 (和) をもとめる。

先の更新された行列と比較し  $min$  があれば書き換える。(更新) される。

$$d_{ij}^2 = \min(d_{ij}^1, d_{i1}^1 + d_{1j}^1)$$

|           |    |           |           |           |
|-----------|----|-----------|-----------|-----------|
| 100       | 50 | <b>70</b> | 90        | <b>80</b> |
| 50        | 0  | 20        | 40        | 30        |
| <b>70</b> | 20 | 40        | <b>60</b> | 50        |
| 90        | 40 | <b>60</b> | 80        | 70        |
| <b>80</b> | 30 | 50        | 70        | 60        |

|    |    |    |    |    |
|----|----|----|----|----|
| 0  | 50 | 70 | 20 | 80 |
| 50 | 0  | 20 | 40 | 30 |
| 70 | 20 | 0  | 60 | 30 |
| 20 | 40 | 60 | 0  | 10 |
| 80 | 30 | 30 | 10 | 0  |

以後同様に  $n$  列  $n$  行の外積 (和) を計算し、 $min$  を求めて、更新を最終行列まで繰り返す。

最終結果と軌跡 (TRACE)

wf wfd

|    |    |    |    |    |
|----|----|----|----|----|
| 0  | 50 | 60 | 20 | 30 |
| 50 | 0  | 20 | 40 | 30 |
| 60 | 20 | 0  | 40 | 30 |
| 20 | 40 | 40 | 0  | 10 |
| 30 | 30 | 30 | 10 | 0  |

TRACE

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 4 | 0 | 3 |
| 1 | 1 | 1 | 1 | 1 |
| 4 | 2 | 2 | 4 | 2 |

3 3 4 3 3

3 4 4 4 4

## 2 高速道路の最短距離のアルゴリズム

東名高速道路 東京-御殿場間のノードと距離

NODE

```
0 1 7.6 7.6 NB. 東京-東名川崎
1 2 5.7 5.7
2 3 6.4 6.4
3 4 15.3 15.3
4 5 15.1 15.1
5 6 7.8 7.8
6 7 25.8 25.8 NB. 大井松田-御殿場
```

計算の基礎となるマトリクス(両側)

```
make_mat_1 NODE
0 7.6 - - - - -
7.6 0 5.7 - - - - -
- 5.7 0 6.4 - - - - -
- - 6.4 0 15.3 - - - - -
- - - 15.3 0 15.1 - - - - -
- - - - 15.1 0 7.8 - - - - -
- - - - - 7.8 0 25.8 - - - - -
- - - - - - 25.8 0
```

距離テーブル

```
wf_non make_mat_1 NODE
NB.
東京 川崎 青葉 横浜 厚木 秦野 大井 御殿場
0 7.6 13.3 19.7 35 50.1 57.9 83.7
7.6 0 5.7 12.1 27.4 42.5 50.3 76.1
13.3 5.7 0 6.4 21.7 36.8 44.6 70.4
19.7 12.1 6.4 0 15.3 30.4 38.2 64
```

35 27.4 21.7 15.3 0 15.1 22.9 48.7  
50.1 42.5 36.8 30.4 15.1 0 7.8 33.6  
57.9 50.3 44.6 38.2 22.9 7.8 0 25.8  
83.7 76.1 70.4 64 48.7 33.6 25.8 0