

IV-7 マトリックス法による最短路の計算

— シンメトリックネットワークにおける方法 —

大阪市立大学工学部 正員 毛利正光

〃 〃 〃 〇西村 昂

あるリンクについて一端から他端に至る長さとその逆方向の長さが等しいときそのリンクはシンメトリックであるという。またシンメトリックなリンクのみよりなるネットワークをシンメトリックなネットワークという。N個のノードからなるシンメトリックネットワークを考える。ノード*i*とノード*j*が直接連絡されている場合はそのリンクの長さを d_{ij} で表わす。 $d_{ij} = d_{ji}$ であるからこれを d_{ij}^1 ($i < j$) のみで代表させる。 d_{ij}^1 については i から j に行く場合と j から i に行く場合についての2種の先行ノードが定義される。一般に $i < j$ のとき、 i から j に至る場合の先行ノード β を右側に、 j から i に至る場合の先行ノード α を左側に添字として示せば、 m 個以下のリンクよりなる i, j 間の最短路の長さは $\alpha d_{ij}^m \beta$ で表わされる。シンメトリックネットワークであるから次の関係が成立する。

$$\alpha d_{ij}^m \beta = \beta d_{ji}^m \alpha \quad (1)$$

与えられたネットワークを d_{ij}^1 ($i, j = 1, 2, \dots, N$) を要素とするマトリックスで表示して D^1 とする。 D^1 は三角のマトリックスとなり $\frac{1}{2}N^2$ のオーダーのメモリーを必要とする。 D^1 においては $i < j$ のとき d_{ij}^1 の先行ノードは i であり、 d_{ji}^1 の先行ノードは j であるから i と j を結ぶリンクは $j d_{ij}^1 i$ で表わされる。マトリックスの改良には次の式が利用される。

$$d_{ij}^2 = \min_k (d_{ik}^1 + d_{kj}^1) \quad k = 1, 2, \dots, N \quad (2)$$

$$d_{ij}^{m+1} = \min_k (d_{ik}^m + d_{kj}^m) \quad k = 1, 2, \dots, N \quad (3)$$

$$d_{ij}^{2m} = \min_k (d_{ik}^{2m} + d_{kj}^{2m}) \quad k = 1, 2, \dots, N \quad (4)$$

$$d_{ij}^{m+m} = \min_k (d_{ik}^{2m} + d_{kj}^{2m}) \quad k = 1, 2, \dots, N \quad (5)$$

(4)式は改良の速度が早いのでこの式による場合を先行ノードを含めた演算式で示せば次のようになる。

(i) $k < i < j$ の場合

$$\beta d_{ij}^{2m} \alpha = \min_k (\alpha d_{ki}^{2m} \beta + r d_{kj}^{2m} \delta) \quad k = 1, 2, \dots, N$$

(ii) $i < k < j$ の場合

$$\alpha d_{ij}^{2m} \delta = \min_k (\alpha d_{ik}^{2m} \beta + r d_{kj}^{2m} \delta) \quad k = 1, 2, \dots, N \quad (6)$$

(iii) $i < j < k$ の場合

$$\alpha d_{ij}^{2m} r = \min_k (\alpha d_{ik}^{2m} \beta + r d_{kj}^{2m} \delta) \quad k = 1, 2, \dots, N$$

この行列演算は次の条件のうちの1つが成立すれば終了し、 D^m は解行列となる。

(i) $D^{2m} = D^m$ (6)式による場合

(ii) $m \geq N - 1$

条件(i)は行列がそれ以上改良されなくなったことを示し、条件(ii)は任意のノード間に $N -$

1以上のリンクが入り得ないことを示す。解行列において各ノードの組合せに対して、先行ノードをたどりながら起終点の連絡が可能となる。

【計算例】 図-1のシンメトリックネットワークについて考えて見る。各リンクに示した数字はそのリンクの走行時間、ノードに示した()の数字はノードでの待時間Eを示す。

ノードの待時間を考えない場合は各演算式に、長が起終点ノードと一致しない場合に待時間 d_{RR} を加えればよい。このネットワークをマトリックスで示すと D^1 のように表わされる。 D^1 を(6)式により改良すれば D^2 、 D^3 、 D^4 が順次求められる。 $D^4 = D^3$ となったので D^3 は解行列である。この解行列は(1)式の関係を利用して下半分をうめれば正方行列に作りかえることができる。しかし計算機内部のみの処理には正方行列に作る必要はない。正方行列に直した場合は各行がそれぞれ最短経路樹を形成する。そこで各要素の2種の先行ノードは路線樹形成の他に発生交通、集中交通の分布先をノード毎に知るためにも利用することができる。

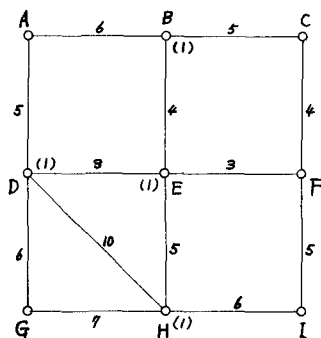


図 - 1

D^1

0 \ D	A	B	C	D	E	F	G	H	I
A	0	B^6_A	x	D^5_A	x	x	x	x	x
B		1	C^5_B	x	E^4_B	x	x	x	x
C			0	x	x	F^4_C	x	x	x
D				1	E^3_D	x	G^6_D	H^{10}_D	x
E					1	F^3_E	x	H^5_E	x
F						0	x	x	I^5_F
G							0	H^7_G	x
H								1	I^6_H
I									0

D^2

0 \ D	A	B	C	D	E	F	G	H	I
A	0	B^6_A	B^{12}_B	D^5_A	D^9_D	x	D^{12}_D	I^6_D	x
B		1	C^5_B	E^8_B	E^4_B	E^8_E	x	E^{10}_E	x
C			0	x	F^7_C	F^4_C	x	x	F^9_F
D				1	E^3_D	E^7_E	G^6_D	E^9_E	H^{17}_H
E					1	F^3_E	D^{10}_D	H^5_E	F^8_F
F						0	x	E^9_E	I^5_F
G							0	H^7_G	H^{14}_H
H								1	I^6_H
I									0

D^4

0 \ D	A	B	C	D	E	F	G	H	I
A	0	B^6_A	B^{12}_B	D^5_A	D^9_D	D^{13}_D	D^{12}_D	D^{15}_D	I^{18}_D
B		1	C^5_B	E^8_B	E^4_B	E^8_E	E^{15}_E	E^{10}_E	I^{13}_E
C			0	F^{11}_C	F^7_C	F^4_C	F^{18}_F	F^{13}_F	F^9_F
D				1	E^3_D	E^7_E	G^6_D	E^9_E	E^{12}_E
E					1	F^3_E	D^{10}_D	H^5_E	F^8_F
F						0	E^{14}_D	E^9_E	I^5_F
G							0	H^7_G	H^{14}_H
H								1	I^6_H
I									0

D^3

0 \ D	A	B	C	D	E	F	G	H	I
A	0	B^6_A	B^{12}_B	D^5_A	D^9_D	D^{13}_D	D^{12}_D	D^{15}_D	I^{18}_D
B		1	C^5_B	E^8_B	E^4_B	E^8_E	E^{15}_E	E^{10}_E	I^{13}_E
C			0	F^{11}_C	F^7_C	F^4_C	F^{18}_F	F^{13}_F	F^9_F
D				1	E^3_D	E^7_E	G^6_D	E^9_E	E^{12}_E
E					1	F^3_E	D^{10}_D	H^5_E	F^8_F
F						0	E^{14}_D	E^9_E	I^5_F
G							0	H^7_G	H^{14}_H
H								1	I^6_H
I									0