Journal of the Society of Korea Industrial and Systems Engineering
Vol, 26, No. 1, pp.47－53, March 2003.

Circular sparse network에서 분할법을 이용한 최단거리 결정

김준홍* · 정영배**

* 수원대학교 산업정보공학과
** 인천대학교 산업공학과

Finding the shortest distance between all pairs of nodes in circular sparse networks by decomposition algorithm

Jun-Hong Kim* · Young-Bae Chung**

* Dept. of Industrial Engineering, University of Suwon
** Dept. of Industrial Engineering, University of Incheon

이 논문은 환(環)을 형성하는 부분네트웍들로 이루어진 sparse network의 특수한 형태에서 최단거리 결정을 위한 효율적인 앨고리즘을 제안한다. 제시된 앨고리즘은 소위 비환(非換) 형태의 sparse network에 대한 최단거리 결정 앨고리즘의 확장이라 할 수 있다. 도우넛 형태를 갖는 sparse network에 대해 최단거리 결정을 위한 접근법으로 하나는 정점제거 방법이고, 다른 하나는 선분제거 방법이다. 여기서 제안된 앨고리즘은 일반적인 n-degree circular sparse network으로 확대될 수 있다.

Keywords : circular sparse network, shortest distance, doughnut shaped network, decomposition algorithm, n-degree circular sparse network

1. Introduction

One of the fundamental problems in network theory is to find shortest distances or paths in a network. The problem of finding a shortest distance often occurs as a sub-problem of other optimization problems. In most applications, a network is very sparse. A sparse network can be regarded as several small networks overlapping each other.

A sparse network is consists of two types shape. The one is composed of a non-circular series of sub-networks like as a linearly-overlaping network, a star-shaped network[Fig.1], or a tree network. The others is composed of a series of sub-networks which form a circle or cycles of sub-networks. For examples, a sparse network can be a doughnut shaped network[Fig.2], a pendulum shaped network, a wheel shaped network, a dumbbell shaped network, a web shaped network and others.

The characteristics of a circular sparse network are as follows :

(1) It can be composed into sub-networks $N_1, N_2, \cdots, N_Q$.
(2) There are arcs connecting all or most pairs of nodes within the sub-networks.
(3) There are arcs connecting the nodes in certain sets of neighboring sub-networks but not others.
(4) The series of sub-networks form a cycle or cycles of sub-networks.

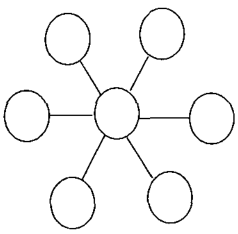
The shortest distances between all pairs of nodes in a circular sparse network can be obtained by an all pair shortest path algorithm. However, when a network is a circular sparse network, all shortest distances can be obtained more efficiently by decomposition algorithms. The decomposition algorithms for finding all shortest distances in a circular sparse network are somewhat different from, and are more complicated than, the decomposition algorithms for finding all shortest distances in a non-circular sparse network.

When a sparse network is a non-circular network, all shortest distances in the network can be obtained more efficiently by decomposition algorithm than by all pair shortest path algorithm. Authors who have introduced different decomposition algorithm for finding all shortest distances in non-circular sparse networks are Blewett and Hu[4], Shier[13], Saltzer et al[12], Rescipno AA[11], Chen CC et al[5], 김준홍[16] etc. and circular sparse networks are Zheng SQ et al[15], Chen DZ et al[6], Atallah MJ et al[12].

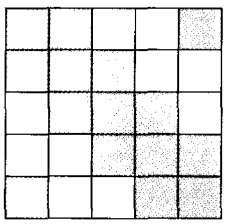
In this study we will present decomposition algorithms for finding all shortest distances in different types of circular sparse networks where the arc distance can be zero, positive or negative in a doughnut shaped network.

2. Decomposition algorithms for finding all shortest distances in doughnut shaped sparse networks

Consider a sparse network consisting of a series of sub-networks, $N_1, N_2, \cdots, N_Q$, such that N_1 is directly connected only to N_Q and N_2, N_2 is directly connected only to N_1 and $N_3, \cdots$, N_3 is directly connected only to N_2, and $N_4, \cdots$, etc. The network of this nature can be shown in Fig.2 by a graph and its arc distances by a matrix. We will call a network of this nature a doughnut shaped sparse network

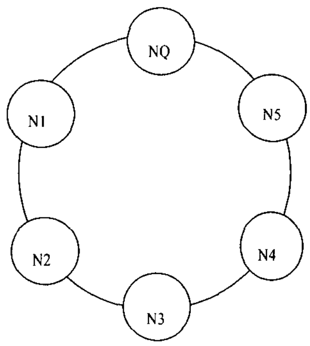


Graph of star-shaped network

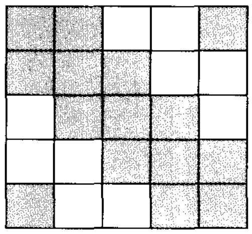


Distance matrix of star-shaped sparse network

<Fig.1> Star-shaped sparse network



Graph of doughnut shaped network



Distance matrix of sparse doughnut shaped network.

<Fig. 2> Doughnut shaped sparse network

All shortest distance in a doughnut shaped sparse network can be obtained by several approaches :

(1) Find all shortest distances by an all pair shortest path algorithms.

(2) Find all shortest distances as follows :

(a) remove from the network an arbitrary sub-network such that the resulting network is a first type of linearly-overlapping sparse network.

(b) find all shortest distances in the new network.

(c) restore the sub-network which was removed in the step(2)(a) and update all shortest distances in the new network. We will call this approach the node elimination approach.

(3) Find all shortest distances as follows :

(a) remove from the network the set of arcs that connect two arbitrary neighboring sub-networks such that the resulting network is a first type of linearly-overlapping sparse network.

(b) find all shortest distances in the new network.

(c) restore the sub-network which was removed in the step(3)(a) and update all shortest distances in the new network. We will call this approach the arc elimination approach.

The first approach is most straight forward but is least efficient. The second and the third approaches are somewhat different yet have a similar efficiency.

In next section, we will develop different algorithms for finding all shortest distances in a doughnut shaped sparse network based on the second and the third approaches.

2.1 A node elimination Approach (Algorithm 1)

Based on a node elimination approach above, we can develop an algorithm for finding all shortest distances in a doughnut shape sparse network as follows :

Step 1. Apply algorithms for finding shortest distance non-circular sparse network to update the sub-matrices $d(N_L, N_K)$, L,K=1, 2,⋯, Q-1.

Step 2. For T=1 and Q-1, let the nodes in N_T be the pivot nodes and update $d(N_L,N_Q)$ and $d(N_Q,N_L)$, L=1, Q-1, Q, i.e., execute :
Divide the nodes in N_T into K sets $N_1, N_2, \cdots, N_K$.

A. For R=1 to K, do the following :
Let $S := M_R$, $\overline{S} := N_Q$ and update $d(S, \overline{S})$, $d(\overline{S},S)$ by;
$d(S, \overline{S}) := \min \{ d(S, \overline{S}), d(S,S) \boxplus d(S, \overline{S}) \}$
$d(\overline{S},S) := \min \{ d(\overline{S},S), d(\overline{S},S) \boxplus d(S,S) \}$,
where the notation $\boxplus$ denotes subroutine for finding all shortest distance by Taubourier[14].
Let $S=\{N_1 \cup N_{Q-1}\}-M_R$, $\overline{S} := N_Q$ and update $d(S, \overline{S})$, $d(\overline{S},S)$, $d(\overline{S}, \overline{S})$ by;
$d(S, \overline{S}) := \min \{ d(S, \overline{S}), d(S,M_R) \oplus d(M_R, \overline{S}) \}$
$d(\overline{S},S) := \min \{ d(\overline{S},S), d(\overline{S},M_R) \oplus d(M_R,S) \}$
$d(\overline{S}, \overline{S}) := \min \{ d(\overline{S}, \overline{S}), d(\overline{S},M_R) \oplus d(M_R, \overline{S}) \}$,
where the notation $\oplus$ denotes subroutine for finding all shortest distance by Hoffman and Winograd[9].

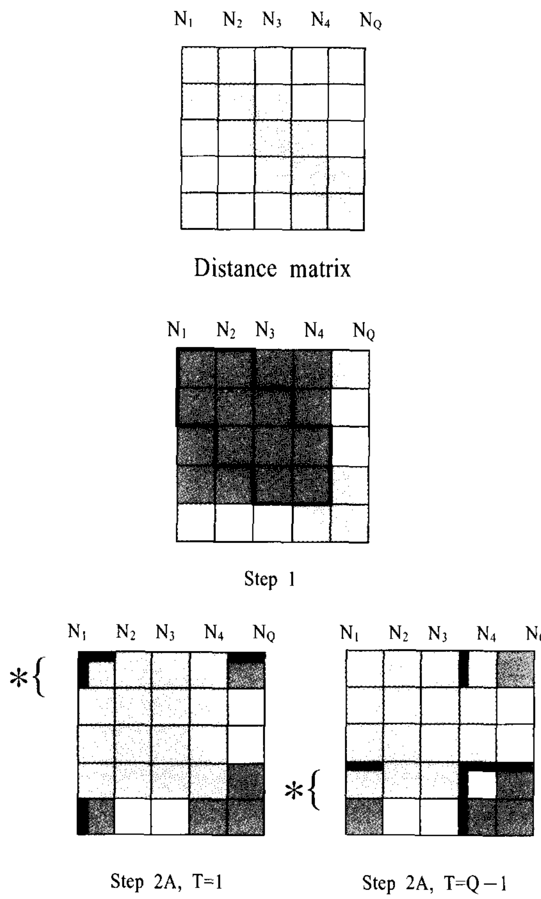
B. $d(N_Q,N_Q) := d^*(N_Q,N_Q)$

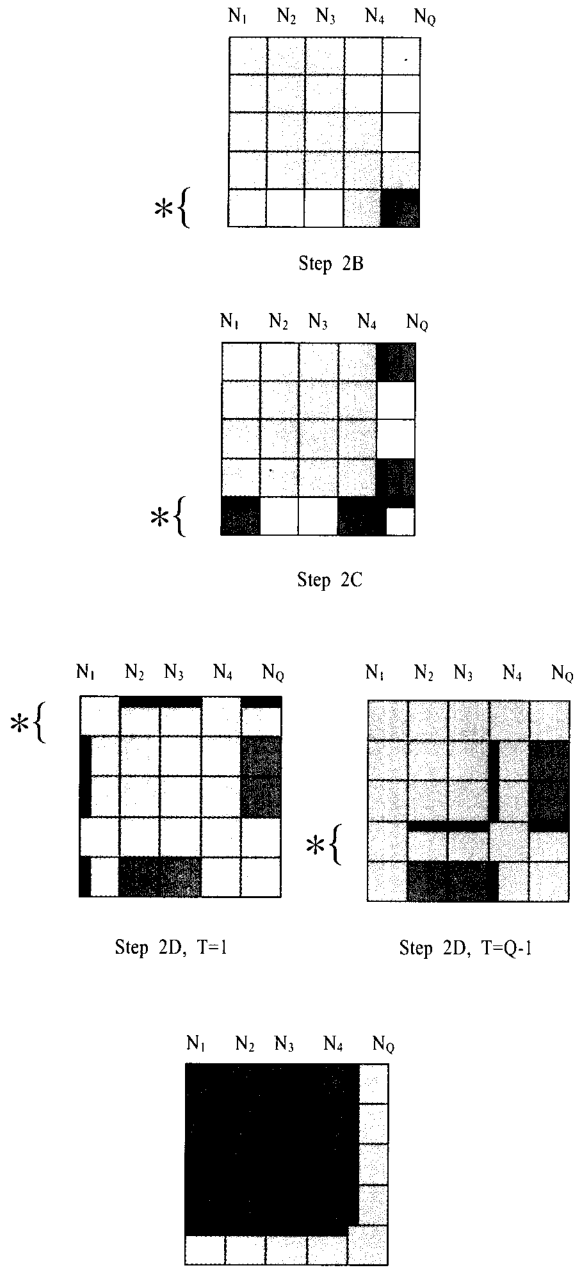
C. Let the nodes in N_Q be the pivot nodes and $d(N_L,N_Q)$ and $d(N_Q,N_L)$, L=1, Q-1, i.e., execute :
Divide the nodes in N_Q into K sets, $M_1, M_2, \cdots, M_K$.
For R=1 to K, do the following :
Let $S := M_R$, $\overline{S} := N_1 \cup N_{Q-1}$ and update $d(S, \overline{S})$, $d(\overline{S},S)$ by;
$d(S, \overline{S}) := \min \{ d(S, \overline{S}), d(S,S) \boxplus d(S, \overline{S}) \}$
$d(\overline{S},S) := \min \{ d(\overline{S},S), d(\overline{S},S) \boxplus d(S,S) \}$.
Let $S=N_Q-M_R$, $\overline{S} := \{N_1 \cup N_{Q-1,}\}$, and update $d(S, \overline{S})$, $d(\overline{S},S)$ by;
$d(S, \overline{S}) := \min \{ d(S, \overline{S}), d(S,M_R) \oplus d(M_R, \overline{S}) \}$
$d(\overline{S},S) := \min \{ d(\overline{S},S), d(\overline{S},M_R) \oplus d(M_R,S) \}$.

D. For T=1 and Q-1, let the nodes in N_T be the pivot nodes and $d(N_L,N_Q)$ and $d(N_Q,N_L)$, L=2, 3,⋯, Q-2, i.e., execute :
Divide the nodes in N_Q into K sets, $M_1, M_2, \cdots, M_K$.
For R=1 to K, let $S := N_Q$, $\overline{S} := N_2 \cup N_3 \cup \cdots \cup N_{Q-2}$ and update $d(S, \overline{S})$, $d(\overline{S},S)$ by;
$d(S, \overline{S}) := \min \{ d(S, \overline{S}), d(S,M_R) \oplus d(M_R, \overline{S}) \}$
$d(\overline{S},S) := \min \{ d(\overline{S},S), d(\overline{S},M_R) \oplus d(M_R,S) \}$.

E. Let the nodes in N_Q be the pivot nodes and $d(N_L,N_K)$, L,K=2, 3,⋯, Q-1, i.e., execute :
Divide the nodes in N_Q into K sets, $M_1, M_2, \cdots, M_K$.
For R=1 to K, let $S := M_R$, $\overline{S} := N_1 \cup N_2 \cup \cdots \cup N_{Q-1}$ and update $d(\overline{S}, \overline{S})$ by;
$d(\overline{S}, \overline{S}) := \min \{ d(\overline{S}, \overline{S}), d(\overline{S},S) \oplus d(S, \overline{S}) \}$.

An illustration of the use of suggested algorithm 1 on a matrix is given in Fig. 3.





* A set of pivot nodes

☐ The empty sub-matrix

☐ The non-empty sub-matrix not currently updated

■ The sub-matrix currently updated

▬▌ A set of shortest distances that are related to the pivot nodes in sub-iteration

<Fig. 3> An illustration of the use of Algorithm 1 for finding all shortest distance in a doughnut shaped sparse network

The number of sub-matrices updated by algorithm are as follows :

$7(Q-1)-6$ key sub-matrices and $(Q-1)^2-3(Q-1)+2$ non-key sub-matrices in step 1, 10 sub-matrices in step 2.A., 1 sub-matrix in step 2.B., 4 sub-matrices in step 2.E.. Therefore a total of $2Q^2+4Q-3$ sub-matrices are updated by the algorithm.

Assuming there are M nodes in each sub-network, then in the best case approximately $0.5M^3-0.5M^{5/2}$ loops, $2M^{5/2}$ addition-subtractions, and $M^3-2M^{5/2}$ comparisons are necessary to update a distance sub-matrix.

In the worst case, approximately $0.5M^3-0.5M^{5/2}$ loops, $4M^{5/2}$ addition-subtractions, and M^3 comparisons are necessary to update a distance sub-matrix. Therefore Algorithm requires in the best case approximately $(2Q^3-4Q-3)(0.5M^3-0.5M^{5/2})$loops, $(2Q^3-4Q-3)2M^{5/2}$ addition-subtractions, and $(2Q^3-4Q-3)$ $(M^3-2M^{5/2})$ comparisons are necessary to update a distance sub-matrix, and in the worst case approximately $(2Q^3-4Q-3)(0.5M^3-0.5M^{5/2}$ loops, $(2Q^3-4Q-3)(4M^{5/2})$ addition-subtractions, and $(2Q^3-4Q-3)(M^3)$ comparisons are necessary to update a distance sub-matrix.

When peripheral memory units are used to store data, algorithm can be executed efficiently using approximately $4M^2$ data storage locations in the CPU.

2.2 An arc elimination approach (Algorithm 2)

Based on an arc elimination approach described previously, we can develop an algorithm for finding all shortest distances in a doughnut shaped sparse
network as follows :

Step 1. Delete from the original distance matrix the distance sub-matrices $d(N_1,N_Q)$ and $d(N_Q,N_1)$.

Step 2. Apply algorithm for finding shortest distance non-circular sparse network to update $d(N_S,N_T)$, $S,T=1,2,\cdots,Q$.

Step 3. A. Update $d(N_I,N_J)$ for $I \in N_1, J \in N_Q$, $I \in N_Q, J \in N_1$, by $d(I,J) = \min\{\overline{D}(I,J), \overline{\overline{D}}(I,J)\}$,

where $\overline{D}(I,J)$ is the current value and $\overline{\overline{D}}(I,J)$ is the original arc distance which was deleted from the distance matrix Step 1.

If all $\overline{\overline{D}}(I,J)$'s are greater than the corresponding $\overline{D}(I,J)$'s, we will terminate the algorithm. In such a case, the $\overline{D}(I,J)$'s for $I,J \in \{N_1 \cup N_2 \cup \cdots \cup N_Q\}$ are the shortest distances of the network.

B. Let the nodes in N_1 be the pivot nodes and update

$d(N_L,N_Q)$ and $d(N_Q,N_L)$, $L=1,2,\cdots, Q$, i.e., execute :
Divide the nodes in N_1, into K sets $N_1, N_2, \cdots, N_K$.
For R=1 to K, do the following :
Let $S := M_R$, $\overline{S} := N_Q$ and update $d(S, \overline{S})$, $d(\overline{S},S)$ by;

$d(S, \overline{S}) := \min \{ d(S, \overline{S}), d(S,S) \boxplus d(S, \overline{S}) \}$

$d(\overline{S},S) := \min \{ d(\overline{S},S), d(\overline{S},S) \boxplus d(S,S) \}$.

Let $S=\{N_1 \cup N_2 \cup \cdots \cup N_{Q-1}\}-M_R$, $\overline{S} := N_Q$ and update $d(S, \overline{S})$, $d(\overline{S},S)$, $d(\overline{S}, \overline{S})$ by;

$d(S, \overline{S}) := \min \{ d(S, \overline{S}), d(S,M_R) \oplus d(M_R, \overline{S}) \}$

$d(\overline{S},S) := \min \{ d(\overline{S},S), d(\overline{S},M_R) \oplus d(M_R,S) \}$

$d(\overline{S}, \overline{S}) := \min \{ d(\overline{S}, \overline{S}), d(\overline{S},M_R) \oplus d(M_R, \overline{S}) \}$.

C. Let the nodes in N_Q be the pivot nodes and update $d(N_S,N_T)$, $S,T=1,2,\cdots, Q$, i.e., execute :
Divide the nodes in N_Q, into K sets $N_1, N_2,\cdots, N_K$.
For R=1 to K, let $S := M_R$, $\overline{S} := \{N_1 \cup N_2 \cup \cdots \cup N_Q\}-M_R$ and $d(S,S) = d^*(S,S)$, then update

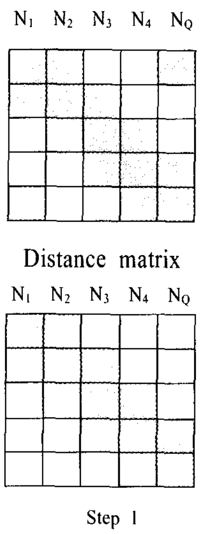
$d(S, \overline{S}) := \min \{ d(S, \overline{S}), d(S,S) \boxplus d(S, \overline{S}) \}$

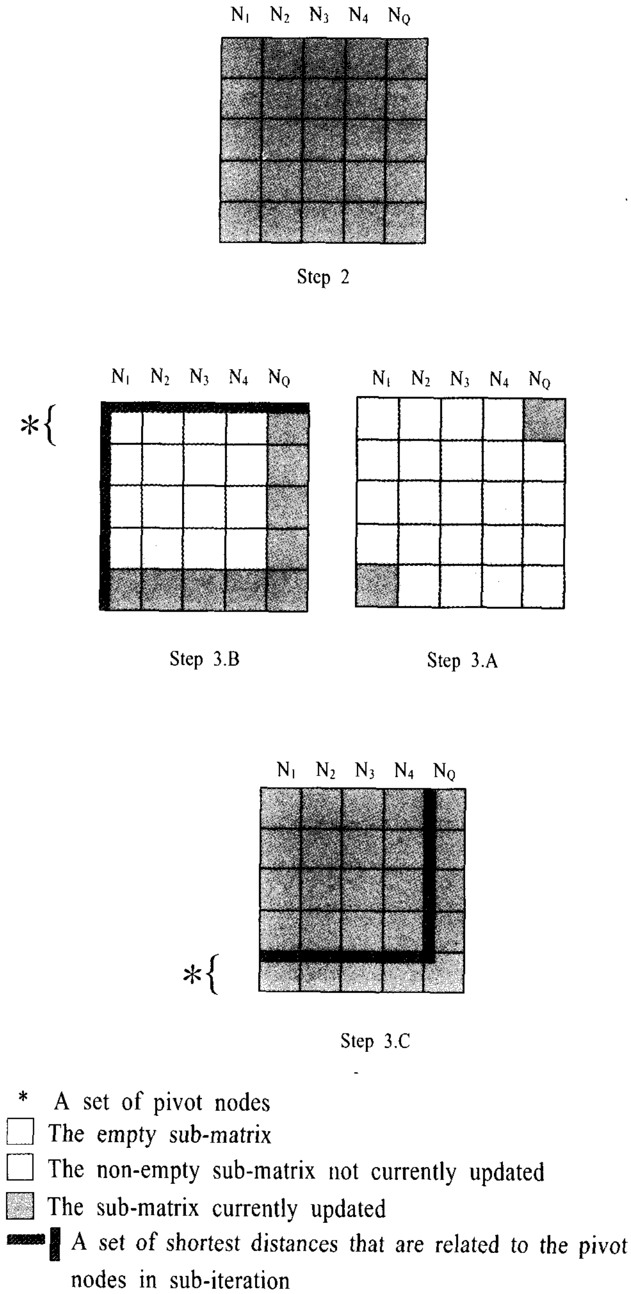
$d(\overline{S},S) := \min \{ d(\overline{S},S), d(\overline{S},S) \boxplus d(S,S) \}$

$d(\overline{S}, \overline{S}) := \min \{ d(\overline{S}, \overline{S}), d(\overline{S},S) \oplus d(S, \overline{S}) \}$.

Algorithm 2 is slightly less efficient than Algorithm 1. This is because sub-matrices $d(N_Q, N_L)$ and $d(N_L, N_Q)$, $L=2, 3,\cdots, Q-2$, are updated three times in Algorithm 2, while they are updated only twice in Algorithm 1.

And sub-matrices $d(N_L,N_K)$, $L,K=Q-1,Q$, are updated four times in Algorithm 2, while they are updated only three times in Algorithm 1. An illustration of the use of Algorithm 2 on a matrix is given in Fig.4.

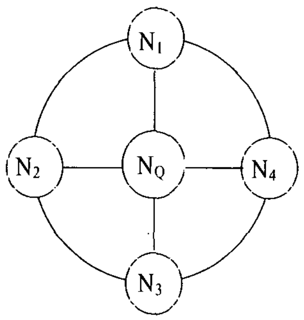




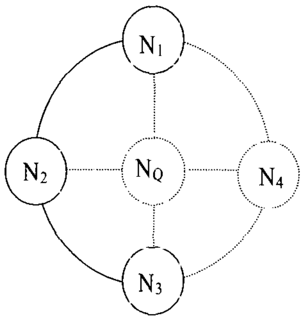
<Fig. 4> An illustration of the use of Algorithm for finding all shortest distance in a doughnut shaped sparse network

3. An algorithmic process of decomposition algorithm for finding all shortest distances in a sparse network.

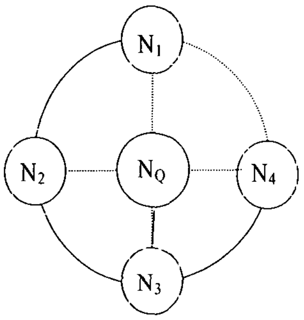
We will present an algorithmic process for a wheel shaped sparse network which has more complicating shape than doughnut shaped network. It consists of a set of sub-networks, N_1, N_2, ⋯, N_Q, where arcs exist between all (or most) pairs of modes in each sub-network, and also between the nodes in sub-networks $N_1 \cup N_2 \cup N_Q$, $N_2 \cup N_3 \cup N_Q$, $N_3 \cup N_4 \cup N_Q$, ⋯, $N_{Q-1} \cup N_1 \cup N_Q$. A sparse network of this type can be shown by the graph in Fig.5.a and its distance matrix in Fig.5.e.



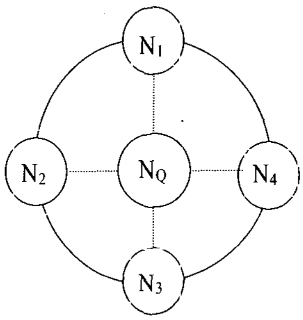
a. A wheel shaped sparsed network



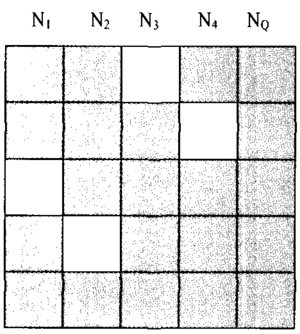
b. Removal of N_4 and N_Q



c. Removal of some arc sets



d. Removal of N_Q and arc set connecting N_1 and N_4



e. Distance matrix of wheel shaped sparse network

<Fig. 5.> Graph of wheel shaped sparse network and its distance matrix

All shortest distances in a wheel shaped sparse network can be obtained in different ways. For example, we can use the following approaches to find all the shortest distances in a wheel shaped sparse network.

(1) Find all shortest distances by an all pair shortest path algorithm.

(2) Find all shortest distance by removing two sub-networks from the original wheel shaped network such that the resulting network becomes a non-circular sparse network, that consists of fewer sub-networks. then restoring the sub-networks one by one as the shortest distance matrix is being updated.

(3) Find all shortest distance by removing sets of arcs connecting two neighboring sub-networks, such that the resulting network becomes a non-circular sparse network then restoring the arcs one by one as the shortest distance matrix is being updated.

(4) Find all shortest distance by using a combination of (2) and (3).

4. Conclusion

The algorithm for doughnut shaped network can be extended to find all shortest distance in sparse networks other than a doughnut shaped sparse network. For example, these ideas can be used to find all shortest distances in different of sparse networks as first degree circular sparse networks which are a wheel shaped sparse network, and second degree sparse network, or n-degree circular sparse network. An n-degree circular network is a circular sparse network which can be simplified into a non-circular sparse network by removing a sub-networks, or the arcs connected to the n sub-networks.

References

[1] Ahuja,R.K.K., Orlin,M,J.B., Tarjan,R.E.T.; "Faster algorithms for the shortest path problem", Technical report No.193, Oper. Res. Center, M.I.T., Cambridge, MA. 1988.

[2] Atallah MJ, Chen DZ Lee DT; "An algorithm for shortest paths on weighted interval and circular arc graphs with applications", Algorithmica, 14(5), pp.429-441, 1995.

[3] Bazaraa,M., Jarvis, J.J.; "Linear programming and Network flows", John Wiley and Sons, New York, 1990.

[4] Blewett,W.J., Hu,T.C.; "Tree decomposition algorithm for large networks", Networks, 7(.4), pp.289-296, 1977.
[5] Chen CC, Chen J; "Nearly optimal one to many routing in star networks, IEEE Transactions on Parallel and distributed systems, 8(12), pp.1196-1202, 1997.
[6] Chen DZ, Lee DT, Sridhar R., Sekharan CN; "Solving the all-pair shortest path query problem on interval and circular arc graphs", Networks, 31(4), pp.249-258. 1998.
[7] Deo,N., Pang,C.; "Shortest path algorithm : Taxonomy and annotation", Networks, 14, pp.275-323, 1984.
[8] Gobow,H.N.; "Scaling algorithms for network problems", Journal of Computer System Science, 31, pp.148-168, 1985.
[9] Hoffman, AJ, Winograd S.; "Finding all shortest distances in directed network", IBM J.Res.Develop., 16(4), pp.412-414, 1972.
[10] Manacher GK, Mankus TA; "A simple linear time algorithm for finding a maximum independent set of circular arcs using intervals alone", Networks, 39(2), pp.68-72, 2002.
[11] Rescigno AA; "Optimally balanced spanning tree of the star network, IEEE Transactions on Computers, 55(1), pp.88-91, 2001.
[12] Saltzer JH, Pogran KT; "A star shaped ring network with high maintainability", Computer networks, 4(5), pp.239-244, 1980.
[13] Shier,D.R.; "A decomposition algorithm for optimality problems in tree-structured networks", Discrete math., 6(3), pp.175-189, 1973.
[14] Tabourier,Y.;"All shortest distance ina graph. An improvement to Danzig's inductive algorithm", Discrete math. 4(1), pp.83-87, 1973
[15] Zheng SQ; Maximum independent sets of circular arc graphs -simplified algorithm and proofs", Networks, 28(1), pp. 15-19, 1996.
[16] 김준홍; "Non-circular sparse network을 위한 분할엘고리즘", 수원대, 11, 1996.