A Heuristic Approach to the Capacity Assignment Problem in a Centralized Network

Jae-Moon Koh
Department of Industrial Engineering
(Received October 30, 1982)

〈**Abstract**〉

This paper deals with the problem of designing the minimum-cost centralized network with multi-point linkage where options are available as to discrete link capacities. A heuristic approach (link exchange technique) is adopted and the result is compared to earlier works. With the approach in this paper, the value of the objective function is improved with more allowable computation time.

중앙집중식 네트워크에서 회선용량의 결정에 관한 발견적 방법

고 재 문
산 업 공 학 과
(1982. 10. 30. 접수)

〈요 약〉

본 논문은 중앙 집중식 네트워크의 설계에 있어서 회선의 용량이 여러 개인 경우에 대하여 연구하였다. 발견적 기법의 일종인 회선교환방식이 선택되어 이전의 연구결과와 비교되었다. 그 결과 계산기간은 증가되었으나 목적함수의 값이 개선되고 증가된 계산시간도 충분히 허용할 수 있는 정도였다.

I. Introduction

In recent years, interests in networks such as transportation network, traffic network, nation-wide road network, oil-pipe network, and computer communication network, have increased. But the network problem is in itself complex and time-consuming, and thus much efforts have been made to solve the associated problems.

Two basic types of network are possible; the centralized network and the decentralized network. In the decentralized network, there are multiple processing centers which are usually controlled by different operating systems. In the centralized network, the network has only a single processing site, and essentially all flows are between remote nodes and the center. But in the decentralized network, if we notice one processing site, we know that partially it is a centralized network. Therefore it can be said that a study on a centralized network is an essential part of designing a network. This paper deals with the problem assigning a capacity to each link to minimize the total link cost in a centralized network.

Network problems can be formulated as mathematical programming in most cases, so it is possible to obtain exact optimal solutions theoretically. But it is much time-consuming and usually it is difficult to reach an exact optimal solution within a reasonable time limit. Thus it is general to use a heuristic technique. In this paper, a heuristic method—link exchange technique—is adopted to obtain a desirable solution rather than an exact optimal solution.

II. Modeling

1. General problem and earlier works

A general problem in this paper is as follows:
Given: (a) the number and locations of nodes and the centers,
(b) the amount of flow from each node to the center,
and (c) link costs,
Minimize: the total link cost
With respect to: the link capacities
Under constraints: link capacity constraints.

The above problem can be formulated as follows.

$$\text{Minimize} \quad Z=\sum_{i=1}^{n}\sum_{j=1}^{n}D_{ij}(C_{ij}) \qquad (1)$$

subject to

$$\sum_{p=1}^{n}\sum_{q=1}^{n}f_{ij}^{pq}\leq C_{ij} \text{ for all } (i,j) \qquad (2)$$

$$\text{(A)} \quad \sum_{j=1}^{n}f_{ij}^{pq}-\sum_{l=1}^{n}f_{li}^{pq}=\begin{cases} v_{pq} & \text{for } i=p \\ 0 & \text{for } i\neq p,q \\ -v_{pq} & \text{for } i=q \end{cases} \qquad (3)$$

$$r_{pq}(t)\leq v_{pq} \text{ for all } (p,q) \text{ and } t \qquad (4)$$

$$f_{ij}^{pq}\geq 0 \text{ for all } (p,q) \text{ and } (i,j) \qquad (5)$$

where n: the number of nodes and the centers
C_{ij}: the capacity of link (i,j)
$D_{ij}(C_{ij})$: the link cost of link (i,j)
f_{ij}^{pq}: the amount of flow on link (i,j) with the source p and the destination q
$r_{pq}(t)$: the amount of flow from the source p to the destination q at time t.

Here, (1) is the total cost which is the objective function of the problem. If the capacity of a link is zero in a solution, this implies that the link is not installed. Thus this objective function contains a link installation problem as well as a link capacity assignment problem. (2) represent the capacity constraints on link (i,j) and (3) represent the conservation law of flow at node i. The first term of the left-hand side of (3) is the amount of flow with the source p and destination q leaving the node i, and the second term, entering the node i, and the difference of the two is the amount of net flow leaving the node i. (4) show that the network must satisfy flow demands at all times.

Observing the model (A), we know that it is a mathematical program with a general non-linear objective function and linear constraints.

Applying one of the various non-linear techniques, we can solve the problem theoretically, but this approach lacks flexibility and requires much computing time and thus it may be difficult to obtain an optimal solution practically. Therefore it is necessary to develop solution procedures which are efficient in computing time although they do not obtain the exact optimal solution.

As for the model (A), it is very complex and difficult to solve the problem as it is. Many authors tried to solve the problem with some assumptions added to the given conditions. T. C. Hu[1] treated the case that the cost function is linear and time is discretized and developed two ways to solve the problem which are called primal and dual approach.

B. Jr. Yaged[5] developed a heuristic method using a computer for the case that the cost function is concave and the network is centralized and Y. S. Lee[4] studied the case that the capacity of link is not limited and $r_{pq}(t)$ are time-independent. A. Kershenbaum and W. Chou[2] developed a modified Kruskal's minimum spanning tree algorithem for the case that all the links in a centralized network have the same capacity and $r_{pq}(t)$ are time-independent. J.M. Koh[3] developed a heuristic algorithm for the case of capacity option. In the paper, he used the basic structure of Esau-Williams algorithm and derived a polynomial bound of computation.

But the efficiency of this algorithm depends greatly upon the initialization of the network and thus it is necessary to develop a more efficient method.

2. Establishment of the problem

This paper deals with the problem considered in the thesis of J.M. Koh[3] and develops a more efficient algorithm in the value of objective function with slightly more computation time. Assumptions added to the model (A) are as follows.

(1) The network is centralized and of tree-type.

(2) The amount of flow from each node to be destined for the center is deterministic.

(3) A finite number of link capacities are available.

With the above assumptions and given the amount of flow from each terminal and the link costs for various capacities, the problem is to design the minimum-cost network that is able to process all the flow demands. It can be formulated as follows.

$$\text{Minimize} \quad Z=\sum_{i=2}^{n}\sum_{j=1}^{n}D_{ij}(C_{ij}) \qquad (6)$$

(B) subject to

$$\sum_{j=1}^{n}f_{ij}-\sum_{l=2}^{n}f_{li}=r_i \quad i=2,3,\cdots,n \qquad (7)$$

$$0\leq f_{ij}\leq C_{ij} \quad j=2,3,\cdots,n \qquad (8)$$

Here, (6) is the summation of link costs with associated capacities and the index 1 represents the center and there is no flow demand at node 1. (7) imply the conservation law of flow and (8) are self explanatory. This model (B) can be also formulated as a mixed integer program[3]. But both model (B) and the related mixed integer program require much computation time and thus it is necessary to develop a method efficient in computation time although it does not obtain the exact optimal solution.

Ⅲ. Algorithm

The approach in this paper basically applies the link-exchange technique and in each link-exchange, the flow change with the associated capacity change on the "main path" is checked. Then if cost-saving occurs, it exchanges the related links.

1. Specification of the network

Before we discuss solution procedures, it is convenient to define some terms specifying a network. The network to be studied in this paper is specified by the following factors; the location of nodes, the connectivity and direction of each link, and the flow on each directed link.

These factors and the related terms are represented as follows.

(1) The location of a node is written as an index i,

$i=1,2,\cdots,n$ in which 1 refers to the center for convenience.

(2) Since the network is assumed to be of tree-type and all flows are destined for the center, the number of arcs emanating from each node is one and thus the "next node" of node i is defined to be the node incident to the arc emanating from the node i, denoting it as NEXT(i).

(3) Given a node i, FLOW(i) is defined to be the amount of flow on the link (i, NEXT(i)).

Similary CAP(i) is defined to be the capacity of the link (i, NEXT(i))

(4) Given a node X connected to the center, a segment X, SEG(X), is defined to be a set of nodes directly or indirectly to the node X before reaching the center.

(5) ISEG(i) is the label of a segment containing node i.

(6) A main path, MAINP(i), is defined to be the path from node i to the center, i.e., the set of $i_0, i_1, i_2, \cdots, i_{m-1}, i_m$ such that $i_0=i$, $i_m=1$, and $i_l=\text{NEXT}(i_{l-1})$, $l=1,2,\cdots,m$.

Given a node i, the main path, MAINP(i), is uniquely determined since the network is of tree-type, and on the main paths the flow or capacity change is checked during the solution procedures.

2. Calcultion of trade-off functions

The main part of this paper is the calculation of trade-off functions. Consider a network in Fig. 1. Circles represent segments and a variable in each circle is the label of the segment.

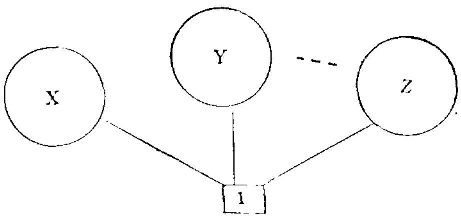


Fig. 1. A Network as segment representation.

Our algorithm selects a pair of links yielding the maximum cost saving and deletes one link and adds the other link, which is the basic structure of link-exchange technique. Related with cost-savings, a trade-off function $T(X:Z)$ is defined by

$$T(X:Z) \triangleq T(X, X1^*, X2^* : Z, Z1^*) = \underset{\substack{X1\in \text{SEG}(X)\\ X2\in \text{MAINP}(X1)\\ Z1\in \text{SEG}(Z)}}{\text{Minimum}}\{TC2(X, X1, X2 : Z, Z1) - TC1(X, X1, X2 : Z, Z1)\}$$

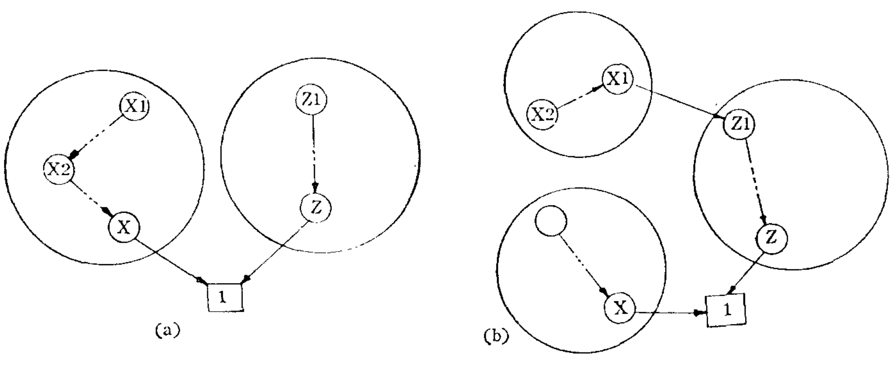
for each pair of segments X and Z. $TC1(X, X1, X2 : Z, Z1)$ is the cost of segments X and Z before link-exchange (Fig. 2-a) and $TC2(X, X1, X2 : Z, Z1)$ is the cost after deleting the link ($X2$, NEXT($X2$)) and adding the link ($X1, Z1$) where $X1\in\text{SEG}(X)$, $Z1\in\text{SEC}(Z)$, and $X2\in\text{MAINP}(X1)$ (Fig. 2-b). Thus $-T(X:Z)$ is the maximum cost-saving from deleting one link in segment X and connecting the related part of segment X to segment Z.

Let

$$T^* \triangleq T(X^*, X1^{**}, X2^{**} : Z^*, Z1^{**}) = \underset{X,Z}{\text{Minimum}}\ T(X:Z).$$

If $T^* \geqslant 0$, then there is no cost-saving and thus the current network is the most satisfactory. If $T^*<0$, this means that we can save cost the most beneficially by deleting ($X2^{**}$, NEXT($X2^{**}$)) and adding ($X1^{**}, Z1^{**}$). Therefore the important question in our algorithm is how to calculate trade-off functions effectively.

Consider a network shown in Fig. 3. This figure is the detailed form of Fig. 2-a (real line)



(a) before and (b) after link-exchange

Fig. 2. A part of network

and Fig. 2-b (dottde line). In the network, $X1 \in \mathrm{SEG}(X)$, $X2 \in \mathrm{MAINP}(X1)$, and $Z1 \in \mathrm{SEG}(Z)$. Moreover $U_0 = \mathrm{NEXT}(X2)$, $U_1 = \mathrm{NEXT}(U_0), \cdots, U_{l+1} = \mathrm{NEXT}(U_l), \cdots, X = U_L = \mathrm{NEXT}(U_{L-1})$, $V_0 = Z1$, $V_1 = \mathrm{NEXT}(V_0), \cdots, V_{m+1} = \mathrm{NEXT}(V_m), \cdots, Z = V_M = \mathrm{NEXT}(V_{M-1})$, and $W_0 = X1$, $W_1 = \mathrm{NEXT}(W_0), \cdots, W_{k+1} = \mathrm{NEXT}(W_k), \cdots, X2 = W_K = \mathrm{NEXT}(W_{K-1})$. As obvious in the figure, it is enough to consider main paths MAINP($X1$) and MAINP($Z1$) for calculating the trade-off function $T(X\,;Z)$.

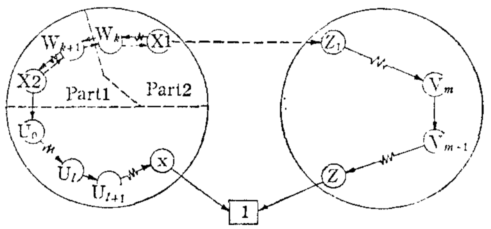


Fig. 3. A detailed form of figure 2-a (real line) and figure 2-b (dotted line).

Before deleting the link (X, NEXT($X2$)), the direction of flow on MAINP(U_0) is $U_l \longrightarrow U_{l+1}$, the flow value on the link (U_l, U_{l+1}) is FLOW(U_l), the direction of flow on MAINP($X1$) − MAINP(U_0) is $W_k \longrightarrow W_{k+1}$, and the flow value on the link (W_k, W_{k+1}) is FLOW(W_k). On the other hand, if we delete the link ($X2$, NEXT($X2$)) and add the link ($X1$, $Z1$), then the direction of flow on MAINP(U_0) and MAINP($Z1$) does not change and the direction of flow on MAINP($X1$) − MAINP(U_0) is reversed. And the flow value on the link (U_l, U_{l+1}) changes from FLOW(U_l) to FLOW(U_l) − FLOW($X2$) and the flow value on the link (V_m, V_{m+1}) from FLOW(V_m) to FLOW(V_m) + FLOW($X2$). But the flow value on the link (W_{k+1}, W_k) after link-exchange is shown to be FLOW($X2$) − FLOW(W_k). In Fig. 3, FLOW($X2$) is the sum of flow demands in part 1 and part 2, and FLOW(W_k) is the sum of flow demands in part 2. When deleting ($X2$, NEXT($X2$)) and adding ($X1, Z1$), the flow demands in part 1 are directed toward the node W_k. But the sum of flow demands in part 1 is FLOW($X2$) − FLOW(W_k) and thus the flow value on the link (W_{k+1}, W_k) after link-exchange is FLOW($X2$) − FLOW(W_k). Any other part than MAINP($X1$), MAINP($Z1$), and the link ($X1, Z1$) does not change from link-exchange. Therefore the cost change, $T(X, X1, X2\,; Z, Z1)$, when deleting ($X2$, NEXT($X2$)) and adding ($X1, Z1$), is calculated as follows.

$$
\begin{aligned}
T(X, X1, X2\,; Z, Z1) &= TC2(X, X1, X2\,; Z, Z1) - TC1(X, X1, X2\,; Z, Z1) \\
&= \sum_{l=0}^{L} \{D(U_l, U_{l+1}, \mathrm{FLOW}(U_l) - \mathrm{FLOW}(X2)) \\
&\quad - D(U_l, U_{l+1}, \mathrm{FLOW}(U_l))\} \\
&\quad + \sum_{m=0}^{M} \{D(V_m, V_{m+1}, \mathrm{FLOW}(V_m) + \mathrm{FLOW}(X2)) \\
&\quad - D(V_m, V_{m+1}, \mathrm{FLOW}(V_m))\} \\
&\quad + \sum_{k=0}^{K} \{D(W_{k+1}, W_k, \mathrm{FLOW}(X2) - \mathrm{FLOW}(W_k)) \\
&\quad - D(W_k, W_{k+1}, \mathrm{FLOW}(W_k))\} \\
&\quad + D(X1, Z1, \mathrm{FLOW}(X2)) \\
&\quad - D(X2, U_0, \mathrm{FLOW}(X2))
\end{aligned}
$$

where $D(i, j, f)$ is the link cost of (i, j) with the flow value f and $U_{L+1} = V_{M+1} = 1$.

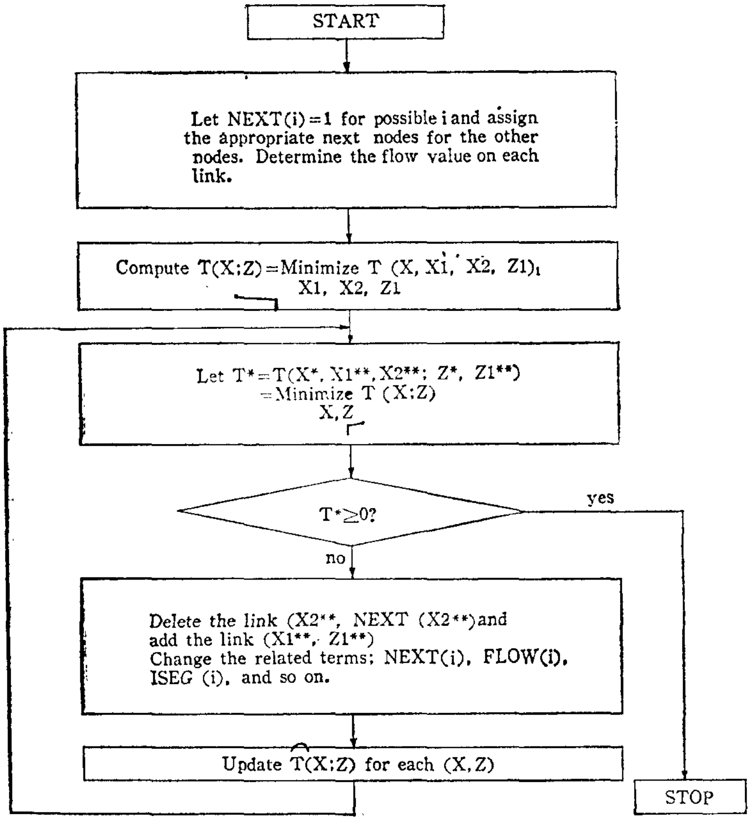
The first term of the righthand side in the above equation is the cost change on MAINP(U_0), the second term, on MAINP($Z1$), and the third term, on the MAINP($X1$) − MAINP(U_0). And the fourth and the fifth terms are link costs of the link ($X1, Z1$) and the link ($X2, U_0$) respectively. Investigating the above equation, we find that given $X2$, it can be expressed as a recursive form, which is a powerful charateristic with respect to computation time.

3. Solution procedure

Using the calculation method developed in the previous section, our algorithm can be described as follows. The flow chart is shown in Fig. 4.

Step 0 (Initialization)

(1) Select nodes allowed to be connected



F.g.4. Flow chart of capacity assignment algorithm

directly to the center and define each of them as one segment. In this case, the label of each segment is the node number itself and the next node and the flow value of such node are the center and flow demand at the node respectively.

(2) If any, connect the remaining nodes to the allowable nodes, and define t e next nodes and the segment labels of them, and change the related flow values.

(3) For each pair of segments, calculate the trade-off function $T(X;Z)$.

Step 1 (Selection of the most desirable pair of segments)

(1) Select $T^*=T(X^*;Z^*)$

$$=\underset{X,Z}{\text{Minimum}}\ T(X;Z)$$

$$=T(X^*, X1^{**}, X2^{**};Z^*, Z1^{**}).$$

(2) If $T^*\geqslant 0$, terminate the algorithm and the curent network is the most satisfactory. Otherwise proceed to step 2.

Step 2 (Link-exchange)

(1) Delete the link($X2^{**}$, NEXT($X2^{**}$)) and add the link ($X1^{**}, Z1^{**}$), and adjust the elements of SEG(X^*) and SEG(Z^*).

(2) Change the flow values on MAINP (NEXT($X2^{**}$)) and MAINP($Z1^{**}$).

(3) Change the direction of flow on MAINP ($X1^{**}$)−MAINP(NEXT($X2^{**}$)) and the related flow values.

Step 3(Calculation of trade-off functions)

Update $T(X;Z)$ for each pair of segments as follows.

(1) If $X=X^*$ or Z^*, then recalculate $T(X;$

Z) for each segment Z.

(2) If $Z=X^*$ or Z^*, then recaculate $T(X ; Z)$ for each segment X.

(3) Otherwise new $T(X ; Z)$=old $T(X ; Z)$. Go to step 1.

Ⅳ. Applications

1. Examples

For comparison, consider a problem in reference 3, since this paper is the expansion of the approach in the reference 3. Its original problem is 'a telephone layout problem where the cost function is continous and concave'. But our algorithm can be applied to the problem and we can compare the results of the approaches. Flow demand at each node, unit cost with each capacity, and distance between each pair of nodes are given in table 1, table 2, and table 3. For each pair of nodes (i,j) with a certain capacity C, link cost is calculated as follows.

$D(i,j,C)$=(unit cost of C)×(distance between i and j)×0.01.

With these data, the computational result is obtained using a digital computer(PRIME

Table 1. Flow demand at each node

Node	Demand	Node	Demand	Node	Demand
2	450	17	540	32	280
3	420	18	420	33	420
4	300	19	410	34	250
5	150	20	520	35	195
6	82	21	305	36	457
7	120	22	170	37	315
8	270	23	191	38	305
9	420	24	220	39	340
10	370	25	210	40	210
11	210	26	240	41	150
12	470	27	175	42	120
13	170	28	520	43	175
14	150	29	570	44	55
15	370	30	350	45	59
16	280	31	520	46	175

Table 2. Capcity and unit cost

Index	Capacity	Cost/distance	Index	Capacity	Cost/distance
1	25	97	11	1800	1481
2	50	124	12	2100	1688
3	100	175	13	2400	1888
4	200	258	14	3000	2403
5	300	339	15	3600	2862
6	400	419	16	4800	3770
7	600	578	17	6000	4697
8	900	825	18	7200	5614
9	1200	1044	19	9000	8791
10	1500	1359			

Table 3. Distance between each pair of nodes

Node-nod	Dist.	Node-node	Dist.	Node-node	Dist.
1— 2	170	15—16	289	28—29	323
1— 3	170	15—17	204	28—36	306
1—17	425	16—17	204	29—30	340
1—42	731	16—42	374	29—31	340
2— 9	306	17—18	289	31—32	340
3— 4	323	17—20	289	31—36	714
4— 5	255	18—19	221	32—33	306
4—10	544	18—21	340	32—34	629
5— 6	170	18—41	306	33—34	459
5—11	680	19—20	170	33—35	510
6— 7	153	19—21	255	34—35	170
7— 8	323	19—24	289	36—37	238
8—15	370	21—22	204	36—38	374
9—10	357	21—24	476	37—39	272
9—12	255	21—41	374	38—39	493
9—14	289	22—23	459	39—40	493
10—12	357	22—27	323	41—44	459
10—14	150	23—24	238	41—46	680
12—13	374	23—25	374	42—46	340
12—20	289	25—26	374	43—44	714
12—25	680	25—28	391	44—45	986
12—30	476	26—27	204	45—46	527
13—14	289	26—38	510		
13—29	663	27—43	646		

* Other pairs of nodes are not allowed to be connected.
* Each pair of nodes is bi-directional.

750) and is shown in table 4. The computation time by the digital computer is 9 seconds and the total link cost is 16980.25. For comparison, the total link cost is 17359.74 (table 5) and the computation time is 5 seconds by the approach in the reference 3. Therefore our algorithm take longer time to obtain the result and the value of the objective function can be improved. But the computation time (9 seconds) may be affirmatively allowed for the case of 46 nodes. And the algorithm in reference 3 depends greatly upon the initial network, so that if unfortunately we initialize the network far from the optimal network, the algorithm may be inefficient in the value of the objective function. But our algorithm can improve efficiently the 'bad' initial network within an allowable computation time. Moreover in the case of the continuous cost function, our algorithm improves the network more efficiently. Table 6 is the result of our algorithm and table 7 shows the result of the algorithm in reference 3 for the case of the continuous cost function, which is defined by

$$D(i,j,f)=d_{ij}\times(1067.0+7.646\times f)\times 0.0001$$

where d_{ij} is the distance between nodes i and j and f is the flow value on the link(i,j). In the continuous case, the value of the objective function is improved by 4.25%.

2. Applicable fields

The model and algorithm in this paper can

Table 4. Result(by the algorithm in this paper)

I	NEXT	FLOW	CAP(I)	COST
2	1	6308.0	7200.0	954.4
3	1	1162.0	1200.0	177.5
4	3	742.0	900.0	266.5
5	4	442.0	600.0	147.4
6	5	82.0	100.0	29.7
7	8	120.0	200.0	83.3
8	15	390.0	400.0	155.0
9	2	5858.0	6000.0	1437.3
10	9	370.0	400.0	149.6
11	5	210.0	300.0	230.5
12	9	4918.0	6000.0	1197.7
13	12	170.0	200.0	96.5
14	9	150.0	200.0	74.6
15	17	760.0	900.0	168.3
16	42	280.0	300.0	126.8
17	1	4790.0	4800.0	1602.2
18	17	2970.0	3000.0	694.5
19	18	630.0	900.0	182.3
20	17	520.0	600.0	167.0
21	18	1620.0	2100.0	573.9
22	21	1615.0	1800.0	302.1
23	25	191.0	200.0	96.5
24	19	220.0	300.0	98.0
25	12	1693.0	1800.0	1007.1
26	27	1095.0	1200.0	213.0
27	22	1445.0	1500.0	439.0
28	25	1292.0	1500.0	531.4
29	30	2235.0	2400.0	641.9
30	12	2585.0	3000.0	1143.8
31	29	1665.0	1800.0	503.5
32	31	1145.0	1200.0	355.0
33	32	615.0	900.0	252.4
34	32	250.0	300.0	213.2
35	33	195.0	200.0	131.6
36	28	772.0	900.0	252.4
37	36	315.0	400.0	99.7
38	26	855.0	900.0	420.7
39	38	550.0	600.0	285.0
40	39	210.0	300.0	109.5
41	46	205.0	300.0	230.5
42	1	839.0	900.0	603.1
43	27	175.0	200.0	166.7
44	41	55.0	100.0	80.3
45	46	59.0	100.0	92.2
46	42	439.0	600.0	196.5

* TOTAL LINK COST=16980.25

Table 5. Result(by the approach in reference 3)

I	NEXT	FLOW	CAR(I)	COST
2	1	7033.0	7200.0	954.4
3	1	1532.0	1800.0	251.8
4	3	1112.0	1200.0	337.2
5	4	442.0	600.0	147.4
6	5	82.0	100.0	29.7
7	8	120.0	200.0	83.3
8	15	390.0	400.0	155.0
9	2	6583.0	7200.0	1717.9
10	4	370.0	400.0	227.9
11	5	210.0	300.0	230.5
12	9	6013.0	7200.0	1431.6
13	12	170.0	200.0	96.5
14	9	150.0	200.0	74.6
15	17	760.0	900.0	168.3
16	42	280.0	300.0	126.8
17	1	3520.0	3600.0	1216.3
18	17	1700.0	1800.0	428.0
19	18	630.0	900.0	182.3
20	17	520.0	600.0	167.0
21	18	650.0	900.0	280.5
22	21	345.0	400.0	85.5
23	25	191.0	200.0	96.5
24	19	220.0	300.0	98.0
25	12	2788.0	3000.0	1634.0
26	25	1095.0	1200.0	390.5
27	22	175.0	200.0	83.3
28	25	1292.0	1500.0	531.4
29	30	2235.0	2400.0	641.9
30	12	2585.0	3000.0	1143.8
31	29	1665.0	1800.0	503.5
32	31	1145.0	1200.0	355.0
33	32	615.0	900.0	252.4
34	32	250.0	300.0	213.2
35	33	195.0	200.0	131.6
36	28	772.0	900.0	252.4
37	36	315.0	400.0	99.7
38	26	855.0	900.0	420.7
39	38	550.0	600.0	285.0
40	39	210.0	300.0	109.5
41	46	380.0	400.0	284.9
42	1	1014.0	1200.0	763.2
43	44	175.0	200.0	184.2
44	41	230.0	300.0	155.6
45	46	59.0	100.0	92.2
46	42	614.0	900.0	280.5

* TOTAL LINK COST=17359.74

Table 6. Result (for the case of continuous cost function by the algorithm in this paper)

I	NEXT	FLOW	COST
2	1	6667.0	885.2
3	1	1282.0	185.3
4	3	862.0	248.3
5	4	562.0	137.5
6	5	202.0	44.9
7	6	120.0	30.8
8	15	270.0	117.0
9	2	6217.0	1488.1
10	9	370.0	140.2
11	5	210.0	183.8
12	9	5277.0	1056.8
13	12	170.0	89.6
14	9	150.0	64.8
15	17	640.0	122.2
16	17	280.0	66.1
17	1	4796.0	1605.1
18	17	2816.0	654.0
19	18	821.0	163.0
20	17	520.0	146.6
21	18	1195.0	348.0
22	21	890.0	161.2
23	24	191.0	60.9
24	19	411.0	122.5
25	12	2052.0	1141.5
26	27	545.0	107.4
27	22	720.0	213.2
28	25	1842.0	593.6
29	30	2235.0	618.3
30	12	2585.0	993.0
31	29	1665.0	470.1
32	31	1145.0	335.0
33	32	615.0	177.5
34	32	250.0	189.2
35	33	195.0	132.0
36	28	1322.0	342.9
37	36	865.0	183.5
38	26	305.0	174.9
39	37	550.0	144.2
40	39	210.0	87.3
41	18	380.0	122.5
42	1	354.0	278.0
43	44	175.0	173.9
44	41	230.0	131.1
45	46	59.0	81.6
46	42	234.0	98.1

* TOTAL LINK COST=14910.70

Table 7. Result (for the case of continuous cost function by the approach in reference 3)

I	NEXT	FLOW	COST
2	1	7033.0	932.8
3	1	1532.0	217.8
4	3	1112.0	310.1
5	4	442.0	114.2
6	5	82.0	29.3
7	8	120.0	65.1
8	15	390.0	150.9
9	2	6583.0	1573.8
10	4	370.0	213.6
11	5	210.0	183.8
12	9	6013.0	1200.3
13	12	170.0	89.6
14	9	150.0	64.8
15	17	760.0	140.9
16	17	1014.0	180.5
17	1	4534.0	1520.0
18	17	1700.0	407.4
19	18	630.0	130.7
20	17	520.0	146.6
21	18	650.0	206.3
22	21	345.0	76.2
23	25	191.0	95.6
24	19	220.0	80.3
25	12	2788.0	1524.2
26	25	1095.0	354.2
27	22	175.0	78.7
28	25	1292.0	429.1
29	30	2235.0	618.3
30	12	2585.0	993.0
31	29	1665.0	470.1
32	31	1145.0	335.0
33	32	615.0	177.5
34	32	250.0	189.2
35	33	195.0	132.0
36	28	772.0	214.2
37	36	315.0	83.4
38	26	855.0	389.4
39	38	550.0	261.4
40	39	210.0	87.3
41	46	380.0	272.2
42	16	734.0	250.9
43	44	175.0	173.9
44	41	230.0	131.1
45	46	59.0	81.6
46	42	614.0	196.9

* TOTAL LINK COST=15573.98

be applied to a computer communication network design problem, a telephone layout problem, minimum-cost transportation route problem, capacity assignment problem of water pipe or oil pipe, and so on. Generally our algorithm is applicable to the problem in which suppliers are various and dispersed geographically, a demander is unique, and suppliers are flow generators.

Ⅴ. Conclusion

The approach in this paper is the extension of the reference 3, so that the basic structure is similar to the one in the reference 3. The algorithm in the reference 3 obtains results very fast, but once the initial network is constructed, it lacks flexibility and the value of the objective function is not sufficient. The algorithm developed in this paper takes longer time to obtain results, but it allows flexibility and improves the value of the objective function. Nevertheless whether the algorithm is an optimal technique or not is yet to be proved and if not, it is necessary to develop an effecient optimal method. Also the case of the decentralized network is yet to be studied.

References

1. Hu, T.C. "Synthesis of a communication network", Integer Programming and Network Flows, pp.197–213. Addison-Wesley, 1970.
2. Kershenbaum, A. and Chou, W., "Unified algorithm for designing multidrop teleprocessing networks", IEEE Trans. Commun., vol. com. 22, no.11, pp.1762–1770, 1974.
3. Koh, J.M., "A heuristic approach to the design of a multipoint linkage teleprocessing network under capacity option", Thesis for degree of master of science in the department of I.E. in K.A.I.S., 1981.
4. Lee, Y.S., "Optimal telephone cable layout in exchange area", Thesis for the degree of master of science in the department in K.A.I.S., 1978.
5. Yaged, B.Jr., "Minimum cost routing for static network models", Networks, vol.1, pp.139–172, 1971.