□ 論 文 □

交通網 設計 問題의 解決을 위한 交通網 集成化 節次

A Network Aggregation Procedure for Solving Network Design Problem

吳 世 昌

(國土開發硏究院 交通硏究室)

목 차

국문요약

본 연구는 교통망 집성화(aggregation)를 수행하기 위한 간단한 개념을 제시하고, 그 개념을 교통망 설계 문제(Network Design Problem)의 개략적이나 상당히 타당성 있는 해답을 얻기 위함에 적용함으로써 교통망 집성화 정도(degree)의 효율성 및 유용성을 검증함에 그 목적이 있다.

교통망 집성화 및 O-D 교통량 측정 절차가 먼저 제시된 후, 교통망 설계 문제에 적용되었다. 본래의 교통망과 집성화된 교통망상에서의 교통망 설계 문제의 실행 결과가 언급되고, 또한 정확도 및 계산 시간 측면에서의 비교가 고찰되었다.

교통망 설계 문제에 대한 개략적이나 비교적 정확한 해답을 고도로 집성화된 교통망으로 부터 추정할 수 있음이 본 논문의 결과로서 검증되었다.

1. Introduction

The network design problem is to select a set of link improvements or link additions to optimize a selected objective function when those actions are implemented. Minimizing the total vehicle hours travelled on the network is traditionally used as an objective function with flows and travel times calculated based on user equilibrium. When the network design problem is formulated into mathematical form, it results in a large scale binary integer problem.

In practice the network design problem deals with large networks, the solution of which require a significant computation efforts. Many researchers have tried to develop efficient algorithms which are exact or approximate to solve the network design problem. Some of the exact and heuristic approaches to deal with the network design problem was reviewed by Magnanti and Wong [1]. Friesz [2] provided a succinct review of major developments in the network design problem and suggested research opportunities to advance this field. Poorzahedy [3] suggested efficient algorithms for solving network design problems.

The purpose of this research is to implement the simple idea for aggregating the network to the smaller size and to test the efficiency of network aggregation for improving solutions to the network design problem. This research also examines the degree of network aggregation to obtain approximate, but relatively accurate solutions for the network design problem. It is obvious that if the detailed network which has all nodes and links is used in solving the network design problem, there will be no error in project selection. However this often requires significant computational efforts and can be very expensive and sometimes impractical. Thus the network aggregation arises to solve the network design problem rapidly and approximately.

Network aggregation which has been done previously is a kind of multi−step aggregation that extracts / abstracts one or several links at each step. But the aggregation in this paper is an one−step aggregation that extracts all possible links at one time. Therefore the contribution of this research is the deployment of a simple aggregation idea that extracts all unnecessary links at once to solve the network design problem rapidly and effectively.

The remainder of this paper is organized as follows. Section 2 describes the network aggregation procedure to obtain the aggregate subnetworks to be used for solving the network design problem and the origin−destination matrix estimation procedure for the aggregate subnetworks. In section 3, the application of network aggregation to the network design problem is described. Section 4 presents the results of the network design solutions on the original and the aggregate networks and compares these results. Finally, Section 5 concludes this research.

2. Network Aggregation and O/D Matrix Estimation Procedure

Network Aggregation

Network aggregation is needed to reduce the large computational burden involved in the analysis of detailed network in many problem contexts. Especially in the context of network design problem, we could save a lot of computational burden by solving the smaller binary integer problem through network aggregation. The computational burden comes from the large number of decision variables and constraints in real-world networks. The aggregate subnetwork should be small enough to be manipulated efficiently as well as effectively, and should keep the required characteristics of the original network.

In a broad sense, there are two types of aggregation approach; network extraction and network abstraction. The network extraction removes insignificant links and nodes from an original network based on a pre-specified criterion. The network abstraction combines detailed links into aggregate links to reduce the network size. Chan et al. [4] proposed the hierarchical search algorithm to solve the network design problem using the network abstraction approach and compared network abstraction approach with network extraction approach. Haghani [5] and Haghani and Daskin [6] developed a network extraction algorithm. The method of this extraction algorithm is to discard insignificant links one by one and to update the O/D trip matrix such that the flow level in the remaining links of the aggregate network remains unchanged. A link in a network is defined as insignificant if the corresponding user equilibrium flow is below prespecified percentage of the maximum equilibrium link flow in the network. Haghani and Daskin [7] tested an extraction algorithm to examine the aggregation effects on the network design problem. The extraction algorithm in Ref.[7] is almost same as one in Ref.[6] except that users are allowed to provide a list of links to be extracted or to be preserved in Ref.[7].

In this paper, the aggregate network is selected so that the closest links to the candidate project links are kept in the aggregate networks as much as possible. The links which will be maintained in the aggregate network are determined purely by visual observation and insight. Therefore, it is possible to change the size and topology of the original network in any desired pattern and design subnetworks of interest. In this network aggregation, all of the unnecessary links are extracted at one time rather than extracting just one link at each iteration which is the case in Ref.[5], [6], and [7].

O/D Matrix Estimation Procedure

Once the aggregate network is determined, the estimation of O/D matrix for the aggregate network is needed as a next step to compensate for the capacity reduction in the network due to the link extraction. In the real world it is extremely expensive and time-consuming to obtain an accurate origin-destination matrix. To reduce the time and effort to acquire a real O/D matrix, researchers such as Robillard [8] and Nguyen [9] proposed methods to determine the O/D matrix from observed link volumes. Turnquist [10] developed an iterative algorithm to estimate a trip table from observed link traffic volumes which can be relatively easily obtainable. In this article, the Turnquist algorithm to determine the O/D matrix based on the observed link flows is therefore applied. An iterative solution algorithm requires the following basic steps [10]:

1. Specify an initial trip table T^1 and a volume-delay(impedance) function for each link.
2. Find the u(skim trees) by using observed link impedances.
3. Assign T^1 to the unloaded network by using free-flow impedances to obtain a set of link volumes f^1. Denote this current solution as a vector(f^1, T^1).
4. Let i = 1.
5. Determine link impedances at the current volume f^i and again build minimum impedance trees. Denote the resulting values u^i.
6. Given T^i, u and ui, find a correction trip table V that is closer to a solution.
7. Assign V^i to the trees built in step 5 to obtain correction link volumes S^i.
8. Find a weight ri such that $0 \le r^i \le 1$ and the solution $[(f^{i+1}, T^{i+1}) = r^i(s^i,V^i) + (1 - r^i) * (f^i,T^i)]$ minimizes the objective function.
9. Check the convergence criterion. If it is met, stop; otherwise, set i = i + 1 and go to step 5.

The origin-destination trip table for each aggregate network which is decided by the implementation of Turnquist algorithm will be used as an input to the network design problem.

3. Application of Network Aggregation to the Network Design Problem

The network of the city of Sioux Falls, South Dakota, is selected as a detailed network which consists of 24 nodes and 76(38 two-way) links. The link cost functions are of the form of $C_i(x_i) = a_i + b_i(x_i)^4$ where x_i is the flow over link i and a_i and b_i are link-specific constants. The detailed network and the candidate project links are shown in Figure 1. In Figure 1,projects 1 through 4 are on existing links and projects 5 through 7 are added as new links. The original link cost functions are shown in Table 1, and the improved or new link cost functions together with improvement or construction costs for the seven candidate projects are provided in Table 2.

Haghani and Daskin hypothesized that the degree of link extraction near the candidate project links directly influences the accuracy of the outcome of the network design problem in the aggregate subnetworks. Referring to their assumption, the closest links to the candidate project links in the aggregate network was kept and four aggregate subnetworks were selected by the degree of adjacency to the candidate project links. Four different aggregate networks which extracts 22, 34, 44, 52 links and consequently remains 54, 42, 32, 24 links are shown in Figure 2,3,4, and 5, respectively.

The aggregate O/D trip matrix for each of these aggregate networks was obtained by implementing the algorithm proposed by Turnquist [10]. These O/D trip matrices were used as the input to the network design problem. The program was run for all four aggregate subnetworks. In the program, user equilibrium link volumes and surveyed O/D trip table were used as observed link volumes and an initial trip table, respectively. It is true that the result is sensitive to the choice of initial trip table. The better the initial solution is, the better the final solution will be. However, the procedure is able to move toward a reasonable solution even if the initial trip table is far from a solution [10]. In each iteration Newton's method was implemented to calculate the corresponding weight, or step size. Under the condition that surveyed O/D trip table is used as an initial trip table, 100, 20, 10 iterations and 3%, 5%, 10% converge ratios were tested as a stopping criterion. Converge ratio is defined as the maximum of absolute value of (updated flow − previous flow) / updated flow 100. Among six stopping criteria, 10 iterations criterion was chosen because it gave the reasonably good result within a short time period. It is noted that this stopping criterion could vary with different transportation network structure.

Three network design problems with 5, 6, and 7 candidate project links were solved on the original and each of the 4 aggregate networks. A wide range of sensitivity analyses were also implemented with different

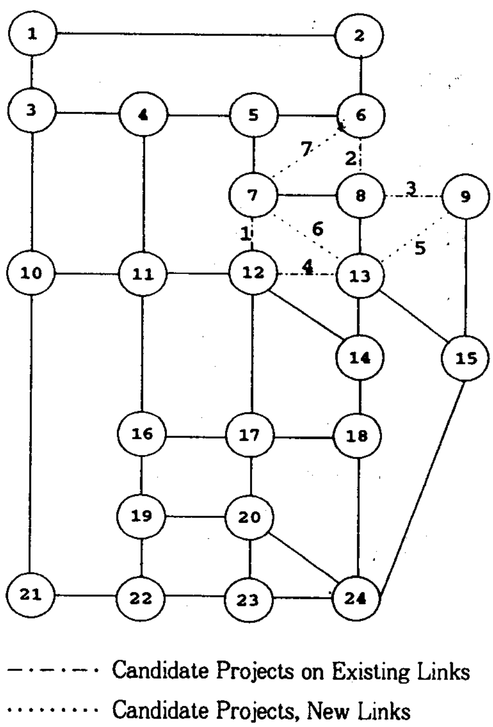

Figure 1. The Original Network and Candidate Project Links(38 Two-Way Links excluding New Links)

Table1. Parameters of Original Link Cost Functions

Link	a_i	$b_i(\times10^{-4})$	Link	a_i	$b_i(\times10^{-4})$
1, 2	0.0596	0.00023	12, 13	0.0270	0.03240
1, 3	0.0434	0.00017	12, 14	0.0804	0.19296
2, 6	0.0517	0.12408	12, 17	0.0587	0.00265
3, 4	0.0431	0.00069	13, 14	0.0167	0.04008
3, 10	0.0414	0.00016	13, 15	0.0269	0.00025
4, 5	0.0216	0.00035	14, 18	0.0231	0.05544
4, 11	0.0646	0.15504	15, 24	0.0446	0.00017
5, 6	0.0417	0.10008	16, 17	0.0452	0.10848
5, 7	0.0503	0.00755	16, 19	0.0425	0.10200
6, 8	0.0130	0.01562	17, 18	0.0350	0.00104
7, 8	0.0961	0.23064	17, 20	0.0350	0.00525
7, 12	0.0160	0.00037	18, 24	0.0399	0.09576
8, 9	0.0150	0.00355	19, 20	0.0400	0.09600
8, 13	0.0482	0.11568	19, 22	0.0188	0.04512
9, 15	0.0218	0.00008	20, 23	0.0167	0.04008
10, 11	0.0646	0.15504	20, 24	0.0471	0.11304
10, 21	0.0298	0.00011	21, 22	0.0220	0.02678
11, 12	0.0500	0.00750	22, 23	0.0329	0.07896
11, 16	0.0442	0.10608	23, 24	0.0572	0.13728

Note : Link is denoted by two nodes.
a_i in hours, b_i in hours/(1000 veh. per day)^4

Table 2. Candidate Projects Characteristics

link	a_i	$b_i(\times10^{-4})$	cost($\times10^3$)
1	0.0160	0.00037	625
2	0.0130	0.01562	650
3	0.0150	0.00355	850
4	0.0270	0.03240	1000
5	0.0300	0.00321	1200
6	0.0220	0.02678	1500
7	0.0150	0.00411	1650

Note : a_i in hours, b_i in hours/(1000 veh. per day)^4, cost in dollars budget levels for all 5 networks and for all 3 design problems. The network design problem was solved by Poorzahedy's Algorithm I [3]. This algorithm is a kind of branch-and-backtrack implementation with strong lower bounds obtained by a few arithmetic operations. It uses Frank-Wolfe Algorithm to get a solution of the conditional problem to the original problem. Readers interested in the notational representation of Poorzahedy's Algorithm I are encouraged to consult Ref.[3].

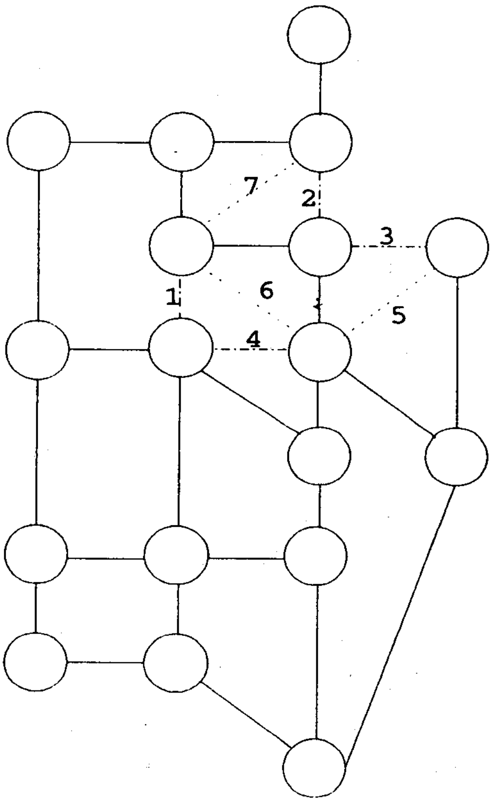

Figure 2. First Aggregate Network(11 Two-Way Links Extracted)

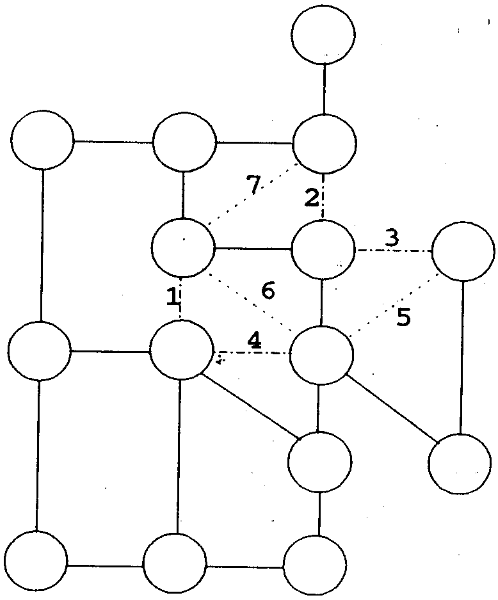

Figure 3. Second Aggregate Network(17 Two-Way Links Extracted)

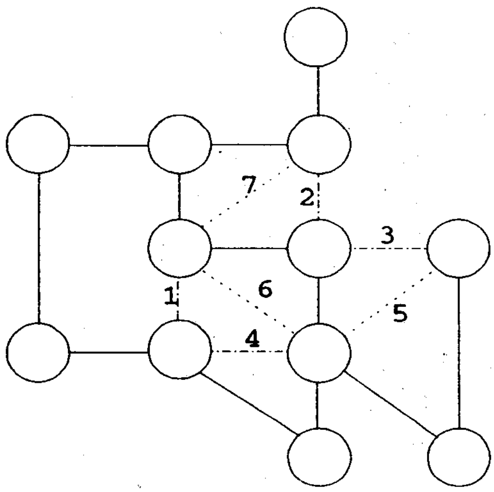

Figure 4. Third Aggregate Network(22 Two-Way Links Extracted)

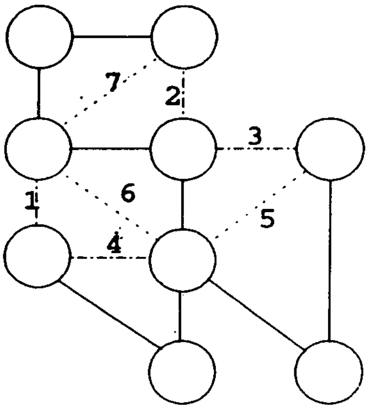

Figure 5. Fourth Aggregate Network(26 Two -Way Links Extracted)

The descriptive procedure of Poorzahedy's Algorithm I is as follows:

Step 0. Order the projects according to the increase in their construction costs.

Step 1. (Initialization)
Find the initial optimal flow pattern of the conditional problem of the original problem with null selection of projects.

Step 2. (Branching)
Produce all alternative solutions which are feasible at the current budget level. In this step, the algorithm resolves the infeasibility problem of the solution by branching.

Step 3. (Decision Making)
Choose the best solution among the alternative solutions produced in Step 2 by implementing Frank-Wolfe Algorithm.

Step 4. (Sensitivity Analysis)
Find solutions, if exist, such that the least amount of additional budget is required to get an extra benefit by accepting an extra project.

4. Results

The results of network design problems for the original and aggregate networks are summarized in Table 3. As shown in Table 3, there were 22 unique budget levels in case of 5 candidate projects, 44 unique budget levels in the 6 project case, and 80 unique budget levels in the 7 project case. (A unique budget level means the budget range for which a group of project combinations is budget feasible. Some project combinations would be budget infeasible with smaller budget. Additional project combinations would be budget feasible with larger budget.) In the 5 project case, the use of the aggregate network led to the choice of a combination of projects that was different from the one chosen on the original network in at most 8 of the 22 budget levels (36.4 %). In the 6 project case, it happened in at most 28 of the 44 budget levels (63.6 %). For the case of 7 candidate projects, it occurred in at most 46 of the 80 budget levels (57.5 %). Using total hours in Poorzahedy Algorithm I as a measure of effectiveness, maximum percentage errors in aggregate networks are 4.92, 2.90, and 1.63 with 7,6, and 5 candidate projects, respectively. These values are computed as follows. At a given budget level, let $\underline{X}_0$ and

$\underline{X}_a$ be the optimal solutions to the network design problem for the original and the aggregate network,respectively, where $\underline{X} = x_i$ and x_i = (1,0) if project i (is, is not) selected in the optimal set. Let $F(\underline{X}_0)$ and $F(\underline{X}_a)$ represent the objective function values of original and aggregate network, respectively. The percentage error is calculated as $[F(\underline{X}_a) - F(\underline{X}_0)] / F(\underline{X}_0) * 100$. Moreover, using the traditional measure of effectiveness which is the total vehicle-hours of travel and the by-product of this research, maximum percentage errors in aggregate networks are 9.68, 3.72, and 3.92 with 7,6, and 5 candidate projects, respectively.

The variance between the network design problem results on the original and aggregate networks comes from several sources such as the aggregation nature of the original network, the comparatively narrow domain of equilibrium flows in the original network, the large number of candidate project links compared to the total number of links especially in the aggregate networks, and the extraction of too many links adjacent to the project links [6].

Table 3. Summary of Network Design Problem Solution

No. of candidate projects	7				6				5			
No. of links extracted	22	34	44	52	22	34	44	52	22	34	44	52
No. of unique budget levels	80	80	80	80	44	44	44	44	22	22	22	22
Percentage of budget levels with different links selected for improvement	42.5	48.8	45.0	57.5	63.6	43.2	54.5	56.8	22.7	13.6	36.4	36.4
Maximum percentage error in vehicle-hrs	8.25	9.68	8.25	8.25	3.72	2.53	2.53	2.53	3.92	2.52	3.92	3.92
Maximum percentage error in hours	3.99	4.92	3.99	3.99	2.90	1.71	1.71	1.71	1.63	1.15	1.57	1.57

Even if there are errors in project selection, the errors of objective function in the network design problem are not substantial. Accordingly, this research suggests that the relatively good results for the network design problem can be obtained by using the aggregate network which keeps most links near candidate projects as much as possible by visual observation.

As the number of projects becomes larger, the computation time to solve a network design problem grows exponentially. To investigate the economy of solving the network design problem on the aggregate networks, computation times are recorded both for the network aggregation procedure and for the network design problem solution. Table 4 (a) shows the run times only for the network design problems on the original and four aggregate networks. The run times for the network aggregation procedure to get four aggregate networks are 40, 29, 19, and 14 seconds, respectively. Table 4(b) represents the run times for network design problem on the original and four aggregate networks plus times to get the corresponding aggregate network.

As indicated in Table 4, the run times on the aggregate networks are smaller than those on the original network for the network design problem. Even if the time required to get the aggregate network is included, the total run times on the aggregate networks are smaller than those on the original network except for one case which is with five candidate projects on the first aggregate network. Thus it is more practical and economical to solve the network design problem on the highly aggregated subnetworks than on the original network in transportation planning. Also, the more the number of candidate projects is, the greater the run time savings are.

5. Conclusions

This research examined the effects of network aggregation on the solution of network design problem. First, link volume survey for the original network was implemented. Second, four aggregate networks were defined according to the level of adjacency to the candidate project links by the visual observation and insight. Third, Turnquist algorithm was executed to get the O/D trip table for the input to the network design problem. Finally, network design problems for the original and four aggregate networks were solved by Poorzahedy Algorithm I and the results were compared.

It turned out that the solution of network design problem on the aggregate network was within the reasonable range of accuracy comparing to that on the original network. Run times for both the network aggregation procedure and the solution of network design problem were recorded to check the efficiency and practicality of solving the network design problems on aggregate networks. It was shown that run time for the solution of network design problem on the aggregate network plus run time for the network aggregation procedure is shorter than that on the original network. Thus the network aggregation procedure is proved to be an efficient and adequate tool in the field of network design problem.

Table 4. Run Times

Networks	Original	1st Agg.	2nd Agg.	3rd Agg.	4th Agg.
(a) Network Design Run Times Excluding Aggregation					
7 Candidate Projects	448	372	223	153	92
6 Candidate Projects	266	201	132	80	48
5 Candidate Projects	122	89	63	36	22
(b) Network Design Run Times Including Aggregation					
7 Candidate Projects	448	412	252	172	106
6 Candidate Projects	266	241	161	99	62
5 Candidate Projects	122	129	92	55	36

All times are recorded in seconds on a 80486/50 PC.

Acknowledgements

The author would like to acknowledge Prof. Haghani in University of Maryland at College Park, whose idea and help contributed to this research.

References

[1] Magnanti, T.L. and R.T. Wong, 1984, "Network Design and Transportation Planning: Models and Algorithms," Transportation Science, Vol. 18, No. 1.

[2] Friesz, T.L., 1985, "Transportation Network Equilibrium, Design and Aggregation: Key Developments and Research Opportunities," Transportation Research, Vol. 19A, No. 5/6.

[3] Poorzahedy, H., 1980, "Efficient Algorithms for Solving the Network Design Problem," Ph.D. Dissertation, Northwestern University.

[4] Chan, Y., T.T. Shen, and N.M. Mahaba, 1988, "The Transportation Network Design Problem: Application of a Hierarchical Search Algorithm," Working Paper, Air Force Institute of Technology.

[5] Haghani, A.E., 1982, "A Network Extraction Approach to the Network Aggregation Problem," M.S. Thesis, Northwestern University.

[6] Haghani, A.E. and M.S. Daskin, 1984, "A Network Design Application of Extraction Algorithm for Network Aggregation," Transportation Research Record 944.

[7] Haghani, A.E. and M.S. Daskin, 1986, "Aggregation Effects on the Network Design Problem," Journal of Advanced Transportation, Vol. 20, No. 3.

[8] Robillard, P., 1975, "Estimating the O-D Matrix from Observed Link Volumes," Transportation Research, Vol. 9, pp. 123-128.

[9] Nguyen, S., 1977, "Estimating an O-D Matrix from Network Data: A Network Equilibrium Approach," Univ. of Montreal, Rept. 60.

[10] Turnquist, M. and Y. Gur, 1979, "Estimation of Trip Tables from Observed Link Volumes," Transportation Research Record 730.