Journal of the Operations Research
Society of Japan
Vol.20, No.3, September, 1977

URBAN ROAD NETWORK MODELING PROBLEM —— FORMULATION AND ALGORITHMS ——

YUICHIRO ANZAI
Keio University

(Received March 10, 1977; Revised August 31, 1977)

Abstract Planners of urban traffic or transportation systems treat with their own network models of urban road networks constructed based on their objectives. Usually, those models seem to be determined empirically, and it may not be clear if the planners' objectives are reflected on them or not. This paper presents (i) a formulation of a network modeling problem for urban road networks as a combinatorial optimization problem, (ii) its solution algorithm based on implicit enumeration, (iii) its extensions for some restricted conditions, (iv) some suboptimal techniques for large-scale networks, and (v) some practical examples from vehicle traffic network planning of Tokyo and Nagoya city areas. By the presented algorithm, the planners can obtain network models of appropriate size which are "optimal" in the sense that they reflect best the planners' objectives. From the results of examples, it is verified that the presented algorithm, its extensions and suboptimal techniques provide an effective procedure for network modeling problems, which is applicable for practical use.

1. Introduction

Planners or designers of urban traffic or transportation planning usually deal with their own network models of urban road networks. For any project, some working model of appropriate size would be constructed based on the planner's objective. However, as the original road network is usually quite large and complex, it is not likely that definite models reflecting both of the planner's objective and the characteristics of the given original network can be easily determined in a reasonable way. Also, there exist similar

cases as above, which may frequently occur in some other urban planning projects, e.g., energy or water supply systems. Thus, it is desirable for the planners of urban systems that some practical algorithms to find reasonable models of urban road networks are available.

The purpose of this paper is to establish an algorithm for constructing models of urban road networks and verify its adequacy.

The outline of development of the algorithm is as follows. A given original urban road network is considered as a finite directed graph. It is supposed that, from the objective of modeling, some 'degree of importance' related to the objective is attached to each road, which is interpreted as a weight of each edge in the corresponding graph. It may be computed from social states of each road such as regional and traffic characteristics. An optimal model is selected as a strongly connected subgraph of the original graph including some restricted number of edges such that the sum of the weights of edges contained in the subgraph is the largest. Strong connectedness is supposed to be the most fundamental condition satisfied by the graphs corresponding to models of urban road networks. Other conditions may be added later to the models satisfying the above basic restrictions.

Thus, the problem can be formulated as a combinatorial selection of an optimal subset of edges from the original graph under some conditions. An algorithm based on implicit enumeration is developed for solving those problems with some extensions. Several heuristic algorithms for finding suboptimal models of large-scale networks are also given. For all the algorithms are provided practical examples including modeling of vehicle traffic networks of Tokyo and Nagoya city areas.

Some aggregation techniques for networks have been known so far [5]. However, they usually provide *aggregated* models, and do not generate physically meaningful ones. Our models must be *parts* of the original networks for practical use. So far as the author knows, the works on such modeling technique as above have been scarcely known.

2. Formulation

First, it is supposed that an urban road network is given and it is identified with a finite directed graph $G=(V,A)$ where V is a finite set of nodes and A is a finite set of edges, i.e., $A\subseteq\{(v_i,v_j)\,|\,v_i,v_j\in V\}$. A node and an edge of G represent an intersection and a (directed) link of the road network,

respectively. (A *directed* graph is necessary because of possibility of the existence of an one-way link and the situation that only one direction of a link is important for the planner's objective).

Also it is supposed that no isolated node exists in G for simplicity, and the elements of A are numbered as $1,2,\cdots,n$. Furthermore, it is assumed that c_i, the weight of each $i\varepsilon A$, has already been determined as a nonnegative real number. For instance, let $s^i=(s_{i1},\cdots,s_{ik})$ be the nonnegative 'social state vector' of $i\varepsilon A$ representing social states of i, e.g., s_{i1}=traffic volume/day of i, s_{i2}=population of the area around i, s_{i3}=number of manufacturers located around i, etc.. Let a be a k-dimensional nonnegative weighting coefficient vector determined based on the planner's objective. Then, for any $i\varepsilon A$, the weight of i can be defined as $c_i=as^{i^T}$, where the superscript T denotes transposition. The weights of edges determined in this way may represent degrees of importance of links for the planner.

The size of the model should be given by the planner. Hence, r, the least upper bound of the number of edges included in the model, is supposed to be given in advance. Then, our problem may be formulated as finding a subgraph of G which includes no more than r edges and the sum of whose edge weights is maximum under some conditions representing the model characteristics considered below. Without a suitable condition, the problem becomes trivial since the edges having r largest weights must be selected.

As a fundamental condition, assume that the obtained subgraph must be *strongly connected*, which is defined as follows:

Definition. Let $G=(V,A)$ be a finite directed graph where V and A are the sets of nodes and edges, respectively. G is said to be *strongly connected* if and only if, for any $v_i,v_j\varepsilon V$ such that $v_i\neq v_j$, $(v_i,v_j)\varepsilon A$ or there exist $v_{k_1},\cdots,v_{k_m}\varepsilon V$ $(m\geq 1)$ satisfying $(v_i,v_{k_1}),(v_{k_1},v_{k_2}),\cdots,(v_{k_{m-1}},v_{k_m}),(v_{k_m},v_j)\varepsilon A$, that is, v_j is *attainable* from v_i.

The above condition that models must satisfy is intuitive, however, it clearly represents one of general properties that models of urban road networks must possess.

Now our basic modeling problem can be formulated as follows:

$$
\text{(P)}\qquad
\begin{aligned}
&\text{maximize} \quad \sum_{i=1}^{n} c_i y_i \\
&\quad \text{subject to} \quad \sum_{i=1}^{n} y_i \leq r \\
&\qquad\qquad\qquad\quad y_i = 0 \text{ or } 1,\ i=1,\cdots,n \\
&\qquad\qquad\qquad\quad \text{subgraph of } G \text{ defined by edges in} \\
&\qquad\qquad\qquad\quad \{i \mid i \in A,\ y_i = 1\} \text{ is strongly connected.}
\end{aligned}
$$

Let $(y_1^*,\cdots,y_n^*)$ be an optimal solution of (P). Then, the subgraph constructed with $\{i \mid i \in A,\ y_i^*=1\}$ corresponds to an optimal model pursued by the planner.

3. Algorithm

A combinatorial algorithm is necessary for solving (P). Our algorithm is based on implicit enumeration. Basic idea of implicit enumeration can be found in [1, 3, 4] but ours is developed specifically for solving (P).

Schematically, strong connectedness of subgraphs having exactly r edges is decided first with some specified order of the sum of the edge weights. If some subgraph is strongly connected or if the sum is not larger than that of the best feasible subgraph obtained earlier, turn to subgraphs with r-1 edges and repeat. Some machinaries make the algorithm be an effective one.

Renumber the edges of G so that $c_1 \geq c_2 \geq \cdots \geq c_n$ holds. Denote by $(i,j,k,\ell,\cdots)$ the subgraph of G consisting of edges $i,j,k,\ell,\cdots \in A$. Then, the detailed stepwise procedure is shown below:

Step 1. (Initialization) Set $p=r$, $z^0=-\infty$, and $(i_1,\cdots,i_p)=(1,\cdots,r)$.

Step 2. (Fathoming) If $z=\sum_{j=1}^{p} c_{i_j} \leq z^0$, (i) if $(i_1,\cdots,i_p)=(1,\cdots,p)$, let $p=1$ and go to Step 6, and (ii) if $(i_1,\cdots,i_p)\neq(1,\cdots,p)$, go to Step 5 since all $(i_1',\cdots,i_p')$ such that $i_k' \geq i_k$ for all $k=1,\cdots,p$ have been implicitly searched. If $z>z^0$, go to Step 3.

Step 3. (Feasibility check) Decide whether $(i_1,\cdots,i_p)$ is strongly connected by the *algorithm for strong connectedness test* described below. If it is strongly connected, let $z^0=z$ and go to Step 5 since all $(i_1',\cdots,i_p')$ such that $i_k' \geq i_k$ for all $k=1,\cdots,p$ have been implicitly searched. Otherwise, go to Step 4.

Step 4. (Generation) Construct the succeeding subgraph by the *subgraph generation ordering rule*:

Suppose that the preceeding subgraph is $(i'_1,\cdots i'_p)$. If $i'_p<n$, then $(i'_1,\cdots i'_p+1)$ is generated. Else if $i'_{p-1}<n-1$, then $(i'_1,\cdots,i'_{p-1}+1,i'_{p-1}+2)$ is generated. Else if $i'_{p-2}<n-2$, then $(i'_1,\cdots,i'_{p-2}+1,i'_{p-2}+2,i'_{p-2}+3)$ is generated. Else if..., else if $i'_{p-m}<n-m$, then $(i'_1,\cdots,i'_{p-m}+1,\cdots,i'_{p-m}+m+1)$ is generated. Else if..,elseif $i'_2<n-p+2$, then $(i'_1,i'_2+1,\cdots,i'_2+p-1)$ is generated. Else if $i'_1<n-p+1$, then $(i'_1+1,i'_1+2,\cdots,i'_1+p)$ is generated. Else no succeeding subgraph exists.†

Let the generated succeeding subgraph be $(i_1,\cdots,i_p)$ if it exists, and go to Step 2. If it does not exist, all the subgraphs having p edges have been searched, and go to Step 6.

Step 5. (Backtracking) Find the maximal $j\varepsilon\{1,\cdots,p-1\}$ such that there exists a subgraph $(i_1,i_2,\cdots,i_{j+1},*,*,\cdots,*)$ which has not yet been searched, where * implies any edge such that the subgraph can be generated by the subgraph generation ordering rule in Step 4. If such j exists, denote it as j^*, let $(i_1,\cdots,i_p)=(i_1,\cdots,i_{j^*-1},i_{j^*}+1,i_{j^*}+2,\cdots,i_{j^*}+(p-j^*+1))$ and go to Step 2. If it does not exist, all the subgraphs having p edges have been searched, and go to Step 6.

Step 6 (Termination) If $p=1$, stop. The problem is infeasible if $z^o=-\infty$. If $z^o>-\infty$, a subgraph providing z^o is optimal. If $p>1$, let $p=p-1$, $(i_1,\cdots,i_p)=(1,\cdots,p)$ and go to Step 2.

<Algorithm for strong connectedness test>

Let $G'=(V',A')$ be the subgraph considered in Step 3, where V' and A' are the sets of nodes and edges of G', respectively. For any $v_j\varepsilon V'$, let $S_{v_j}=\{v_k|v_k\varepsilon V',(v_j,v_k)\varepsilon A'\}$ and $T_{v_j}=\{v_k|v_k\varepsilon V',\ (v_k,v_j)\varepsilon A'\}$.

Step a. If there exists $v_j\varepsilon V'$ such that $S_{v_j}=\phi$ or $T_{v_j}=\phi$, G' is not strongly connected. Otherwise, let $V^o=V'$ and go to Step b.

Step b. If $V^o=\phi$, G is strongly connected. If $V^o\neq\phi$, select any $v_i\varepsilon V^o$ and let $P=\{v_i\}$, $Q=\phi$ and $V^o=V^o-\{v_i\}$.

Step c. Let $Q=P\cup Q$. If $Q=V'$, go to Step b. Otherwise, go to Step d.

Step d. Select any $v_j\varepsilon P$, and let $P=(P-\{v_j\})\cup(S_{v_j}-(S_{v_j}\cap Q))$. If $P=\phi$, G is not strongly connected. If $P\neq\phi$, go to Step c.

---

†For example, suppose that $n=9$, $p=4$ and $(i'_1,\cdots,i'_4)=(2489)$. Then $i'_p=n(=9)$, $i'_{p-1}=n-1(=8)$ and $i'_{p-2}(=4)<n-2(=7)$. Hence, by the rule, the succeeding subgraph is (2567).

It is easily seen that, when a subgraph G' having p-q edges appears on the way of the strong connectedness test, any subgraph obtained by adding any q edges of G to G' is not strongly connected if the number of nodes v_j satisfying $S_{v_j}=\phi$ (or similarly $T_{v_j}=\phi$) is not less than $q+1$. Furthermore, it can be readily seen that, for a graph G', only the case $q+1\leqslant p-q$, i.e., $q\leqslant(p-1)/2$ is necessary to be considered.

The above algorithm for strong connectedness test is quite efficient as it is purely combinatorial. Since it plays a decisive role in the main procedure, its efficiency is critical for our algorithm.

<Illustrative example>

A small example is illustrated in Figs. 1 and 2 to show the mechanism of the above procedure.

Consider a directed graph G having 9 weighted edges shown in Fig. 1-a. The modeling problem (P) corresponding to it is formulated as follows, where the planner is supposed to obtain an optimal model with no more than 4 edges:

$$\text{maximize } 5y_1+2y_2+3y_3+8y_4+3y_5+5y_6+4y_7+3y_8+6y_9$$

$$\text{subject to } \sum_{i=1}^{9} y_i \leqslant 4$$

$y_i=0$ or 1, $i=1,\cdots,9$

subgraph of G defined by $\{i \mid i\varepsilon A,\ y_i=1\}$ is strongly connected.

First, the edges of G are renumbered with decreasing order of weights as shown in Fig.1-b. Let $z(i_1,\cdots,i_p)=\sum_{j=1}^{p} c_{i_j}$ where c_{i_j} is the weight of the edge i_j. Then, the algorithm proceeds as follows:

Iteration 1. Let $p=4$ and $z^o=-\infty$. Then, $z(1,2,3,4)> z^o$. $(1,2,3,x)$ can not be strongly connected for any $x\varepsilon\{4,\cdots,9\}$. (See Fig.2-a.)

Iteration 2. $z(1,2,4,5)> z^o$. $(1,2,4,x)$ can not be strongly connected for any $x\varepsilon\{5,\cdots,9\}$. (See Fig.2-b.)

Iteration 3. $z(1,2,5,6)> z^o$. $(1,2,5,6)$ is not strongly connected. (See Fig.2-c.)

Iterations 4 - 9. $(1,2,5,7)$, $(1,2,5,8)$, $(1,2,5,9)$, $(1,2,6,x)$ $(x\varepsilon\{7,8,9\})$, $(1,3,7,x)$ $(x\varepsilon\{8,9\})$, and $(1,3,8,9)$ are not strongly connected. z^o is unchanged.

Iteration 10. $z(1,4,5,6)> z^o$. $(1,4,5,6)$ is strongly connected. (see Fig. 2-d.) Let $z^o=z(1,4,5,6)=20$.

Iteration 11. $z(2,3,4,5)=20\leqslant z^o$.

Iteration 12. Let $p=3$. $z(1,2,3)=19< z^o$. Hence, the procedure stops. The obtained optimal subgraph is $(1,4,5,6)$, which is shown in Fig.2-e.

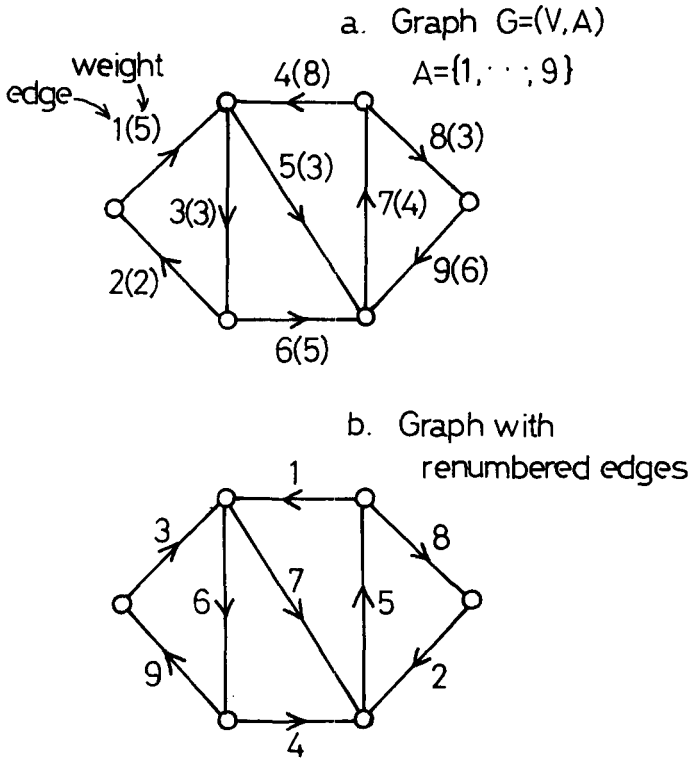


Fig. 1 Graphs for illustrative example --
a. original directed graph G with weighted edges, and, b. graph with renumbered edges.

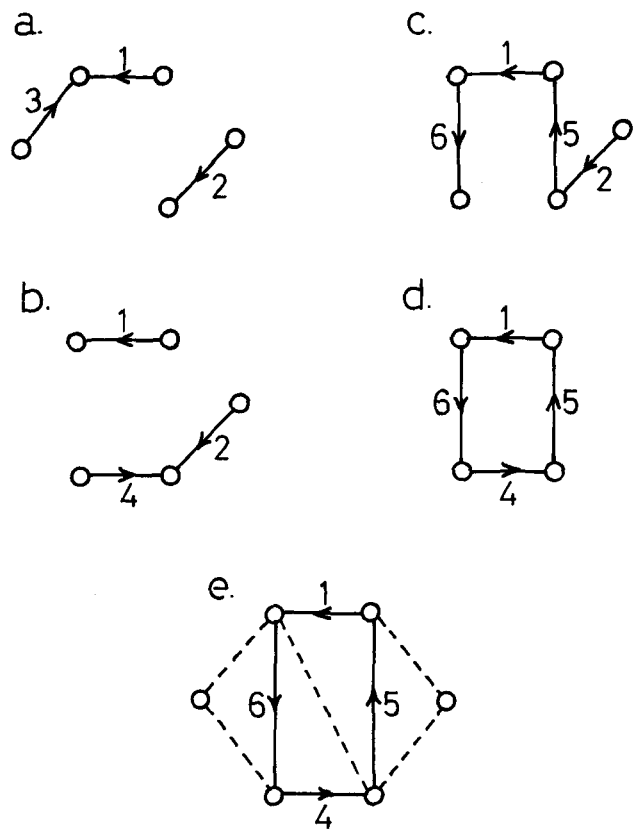


Fig. 2 Subgraphs appearing in iterations for illustrative example

The above algorithm terminates in a finite number of iterations, and provides an optimal solution of (P) if it exists and shows infeasibility of (P) if it does not exist.

Thus, the algorithm can be used for finding an *exact* optimal model of an original network in the sense of (P). It might need substantial computation time for large-scale or ill-conditioned networks. However, from the author's experience, it seems that usually there exist sufficiently good suboptimal solutions having exactly r edges if spatially correlated networks such as urban road networks are treated. Hence, in those cases, the procedure can be made terminate as soon as it falls into Step 6, which makes our procedure quite practical.

4. Computational Examples

First, the relation between r and computation time necessary for finding an optimal subgraph having r edges is shown in Fig. 3 for an original network with 38 edges whose weights are randomly generated.† It shows that an exact optimal solution of (P) can be detected in an admissible time for small-scale networks. Note that the figure in Fig.3 is not symmetric. It is because a strongly connected subgraph may be obtained more easily if $r/n>1/2$ where n is the number of edges included in the original network.

Next, an example from the urban vehicle traffic network modeling of the south-eastern part of Tokyo is given. The directed graph with 38 edges corresponding to the original network is shown in Fig.4. Edge weights are defined from social state vectors as explained in Section 2. That is, the vehicle traffic state vector for the link i is defined as $s^i=(s_{i1},s_{i2})$, where s_{i1}= passenger vehicle traffic volume/day of i and s_{i2}=freight vehicle traffic volume/day of i. Three weighting coefficient vectors, $a^1=(1,0)$ and $a^2=(0,1)$

†The programs were coded in FORTRAN, and run on FACOM 230-45S for all the examples presented in this paper. Computation time was formally recorded only for the example in Fig.3, for the other examples were served for practical purposes and their computation times were sufficiently small to consider seriously at that time. Roughly, they are the same as or less than the time for the example in Fig.3 for the networks of similar sizes, though they depend on the input data to some extent.

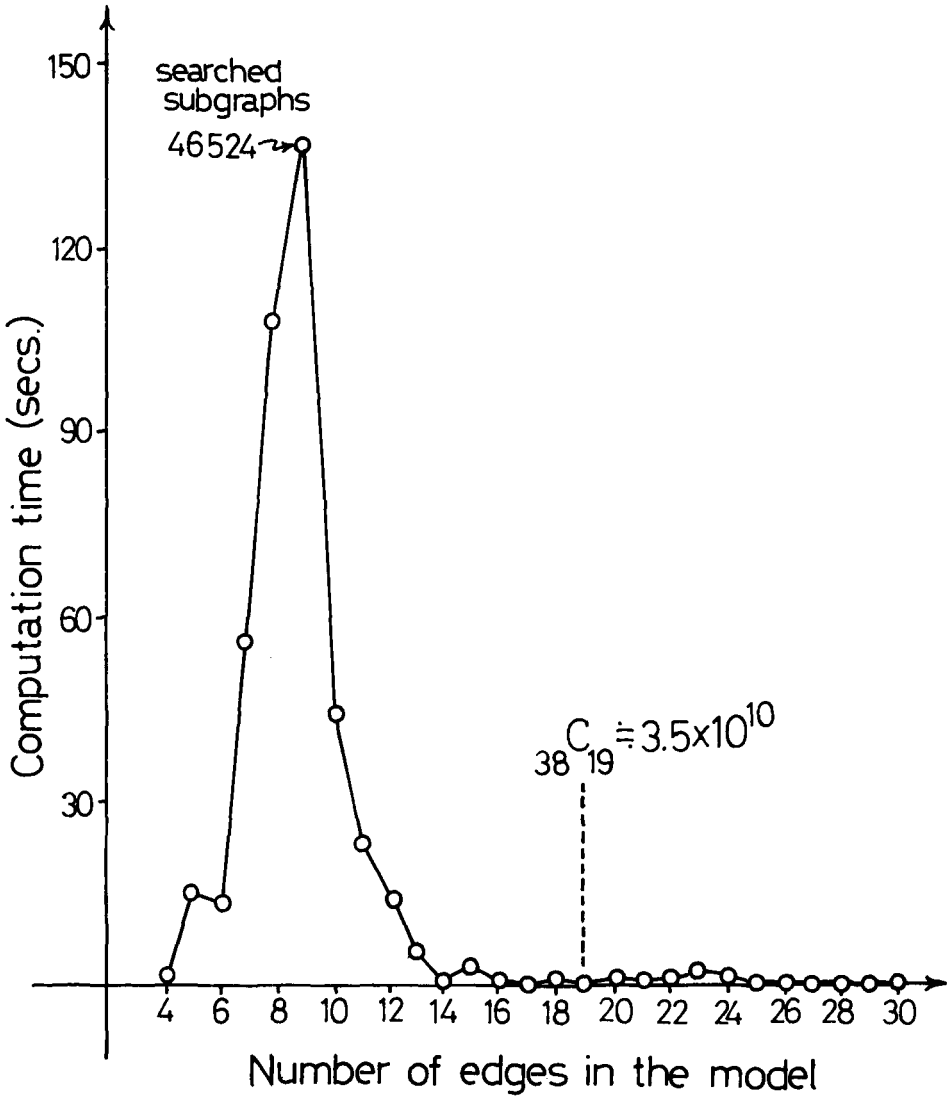
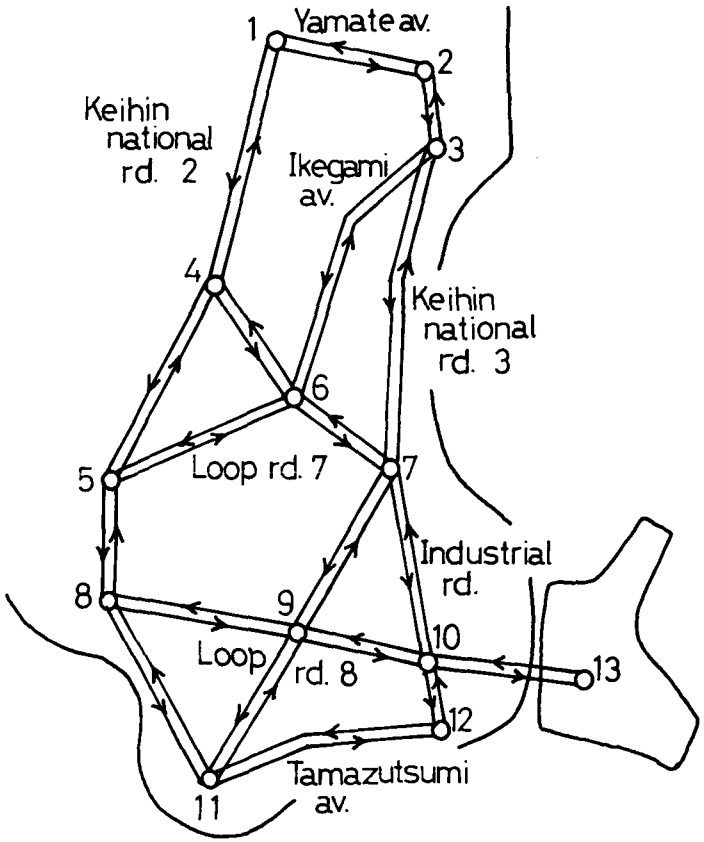


Fig. 3 Relation between number of edges in the model and computation time (FACOM 230-45S) necessary for obtaining it for a graph including 38 edges with randomly generated weights.



Intersections

1. Osaki-hirokoji
2. Kitashinagawa 2-chome
3. Aomono-yokocho
4. Matsubara Bridge
5. Ikegami 3-chome
6. Kasuga Bridge
7. Omori-east
8. Yaguchi 2-chome
9. Kamata 4-chome
10. Otorii
11. Rokugo Bank
12. Haneda
13. Haneda Airport

Fig. 4 Original graph with 38 edges corresponding to vehicle traffic network of the south-eastern part of Tokyo.

and $a^3=(1,1)$, are used to define the weights, $c_i=a^k s^{i^T}$, $k=1,2,3$, and, corresponding to them, three problems of the type (P) are defined.

Optimal models with 28 links obtained by the presented algorithm are shown in Figs.5-a, 5-b, and 5-c. Note that three different patterns have been generated. This fact implies that it may be necessary to change network models properly when the objectives of urban vehicle traffic network planning are altered.

5. Some Extensions with Examples

Here, it is shown that some restrictions which the planner may want to make satisfy can be easily attached to the presented algorithm.

5.1. Modeling of undirected networks

In Fig.5-a, only one direction of some of the principal streets was adopted as a link of the model. For such cases that that situation is not natural, the algorithm can be extended straightforward to undirected networks. Construction of macro-models is a typical case. For undirected networks, connectedness is used in place of strong connectedness in the formulation and algorithm. (The definition of connectedness is similar to that of strong connectedness, and omitted here.) Since an undirected graph is connected if, from any fixed node, every node is attainable, the procedure becomes even simpler than that for directed graphs.

An example from the urban vehicle traffic network of the south-eastern part of Tokyo is shown in Fig.6. Total(=passenger+freight) vehicle traffic volume/day is taken as the edge weight. The original undirected graph contains 39 edges, while the obtained subgraph has 25 edges.

5.2. Modeling with unremovable edges

The case that the planner wants to include some specified roads in the model may frequently occur. It can be solved by the presented algorithm.

Let $\{y_{i_1},\cdots,y_{i_{n-r'}}\}$ be the variables of (P) corresponding to the edges other than unremovable ones, where r' is the number of unremovable edges. Then, consider the problem:

$$\text{(P')} \quad \begin{array}{l} \text{maximize} \ \sum_{j=1}^{n-r'} c_{i_j} y_{i_j} \\ \text{subject to} \ \sum_{j=1}^{n-r'} y_{i_j} \leqslant r-r' \\ \text{subgraph defined by unremovable edges} \\ \text{and edges in } \{i_j \mid 1 \leqslant j \leqslant n-r',\ y_{i_j}=1\} \text{ is} \\ \text{strongly connected.} \end{array}$$

(P') can be solved by the presented algorithm if its strong connectedness test is modified so that unremovable edges should always be contained in subgraphs considered.

An example from the total vehicle traffic network modeling of the southeastern part of Tokyo (the original network is the same as that in Fig.4) is shown in Fig.7-a, where 20 edges are taken in the model and two edges corresponding to Tamazutsumi av. are chosen as unremovable edges. If no unremovable edge existed, the model would have been as shown in Fig.7-b.

5.3 Modeling with unremovable nodes

Also the case that the planner wants to include some specified places or intersections in the model may frequently occur. This case can readily be imputed to a problem of modeling with unremovable edges by adding a self loop to each unremovable node, that is, by modifying A, the set of edges, as $A \cup \{(v_i, v_i) \mid v_i$ is an unremovable node$\}$, and regarding those self loops as unremovable edges.

An example from the same situation as that of 5.2 is shown in Fig.7-c, where a node corresponding to Otorii intersection is considered as an unremovable node. Note that the result is different from the graph in Fig.7-b in which no unremovable node exists.

5.1, 5.2 and 5.3 described above can be used together. Also some other similar restrictions as 5.2 and 5.3 can be attached to the presented algorithm, though they are omitted here.

6. Suboptimal Algorithms for Large-scale Networks

Though exact optimal models can be obtained by the presented algorithm, its effectiveness depends on the structures of original networks. If the edges of an original network are strongly correlated as the usual urban road networks are, the algorithm may work effectively. However, if networks with

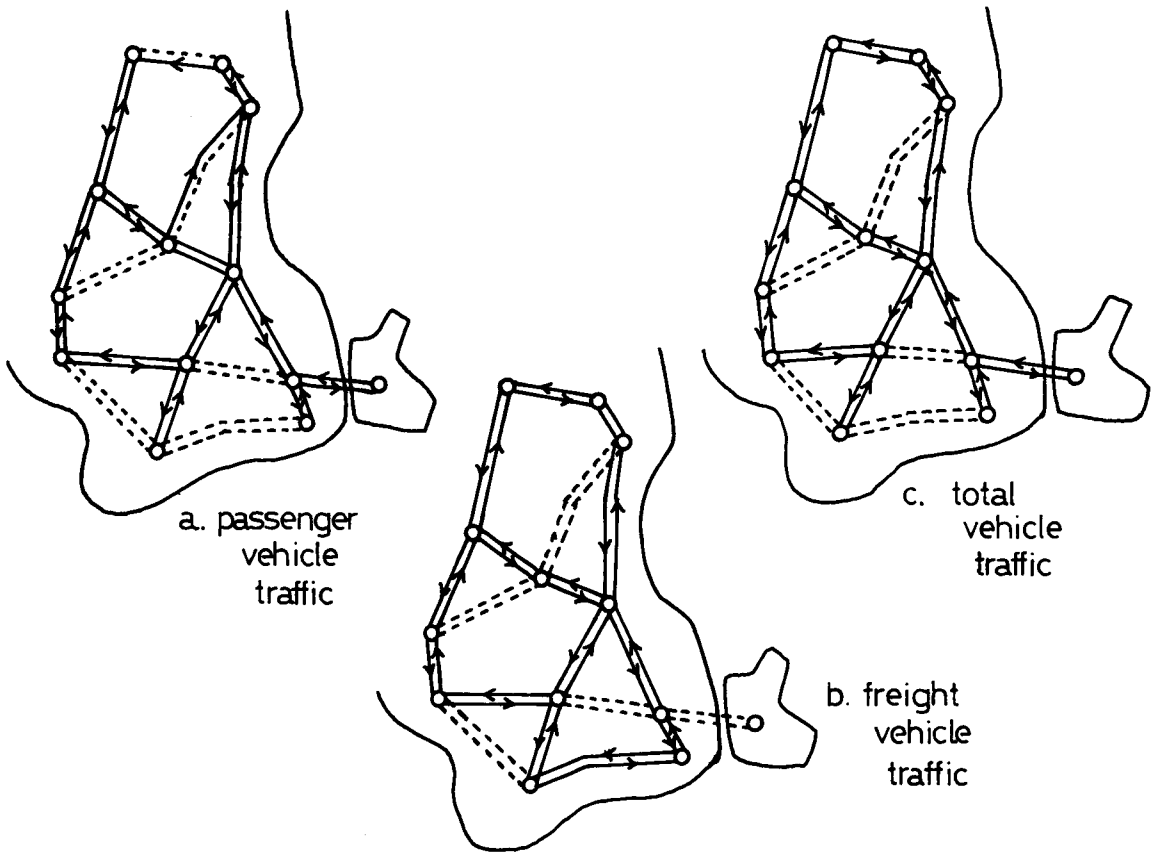
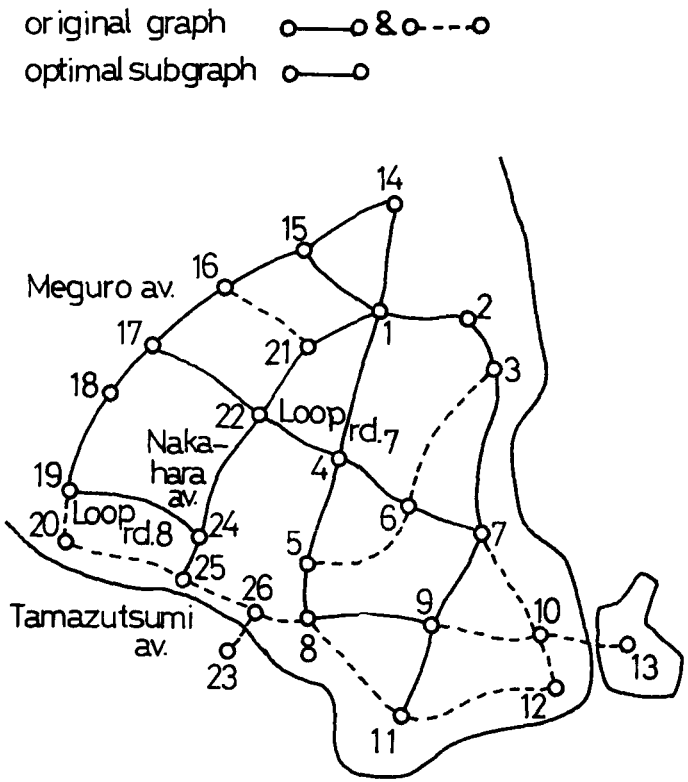


Fig 5. Optimal subgraphs all with 28 edges extracted from original graph with 38 edges (Fig. 4) for urban vehicle traffic network modeling of the south-eastern part of Tokyo.



Intersections

14. Seishoko-mae
15. Otori Shrine
16. Meguro Post Office
17. Kakinoki-zaka
18. Nakane
19. Todoroki-fudo-mae
20. Todoroki 2-chome
21. Hiratsuka Bridge
22. Senzoku-south
23. Gas Bridge
24. Chofu-east Police Station
25. Maruko Bridge
26. Shimo-maruko

Fig. 6 Original graph with 39 edges and optimal subgraph with 25 edges corresponding to undirected network of the south-eastern part of Tokyo. (Numbering of intersections is continued from Fig. 4.)

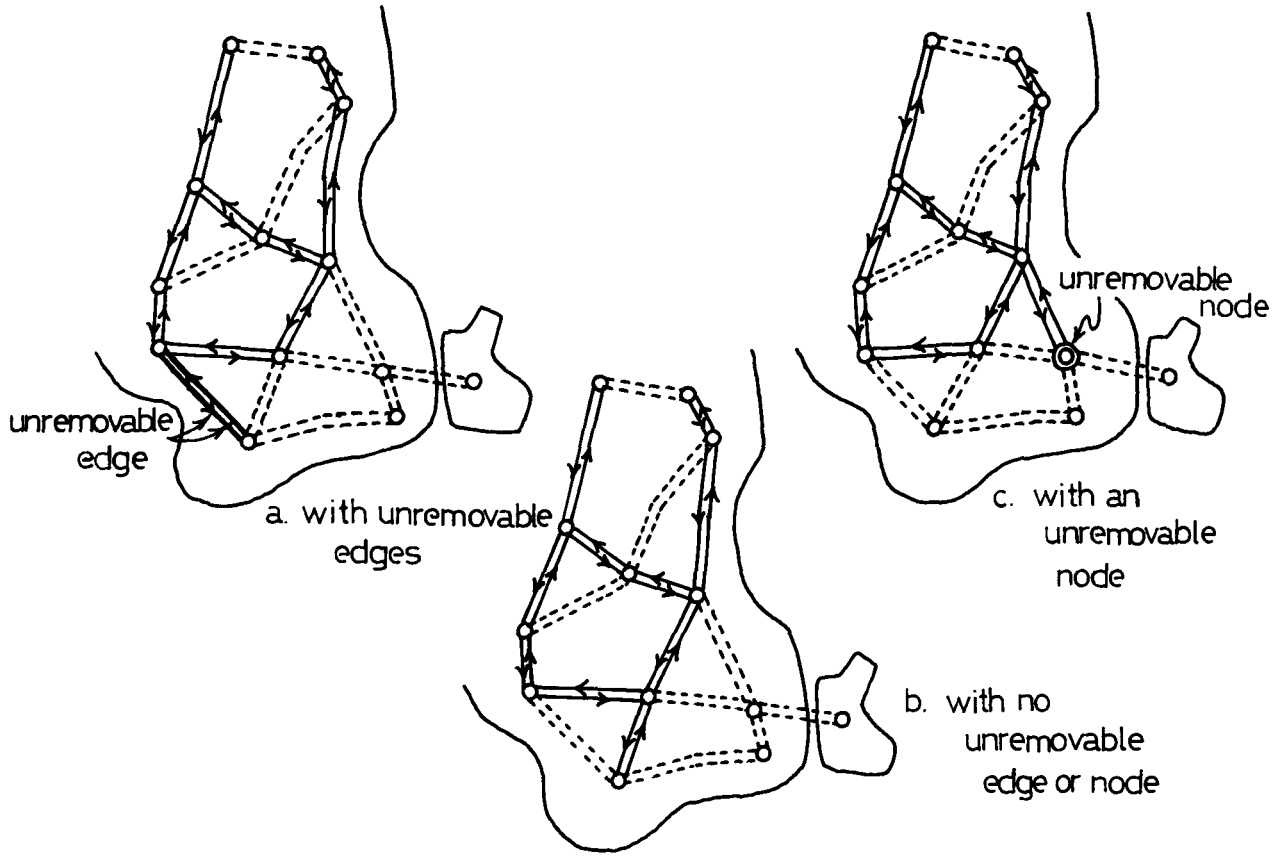


Fig. 7 Optimal subgraphs all with 20 edges extracted from original graph with 38 edges (Fig. 4) for total vehicle traffic network modeling of the south-eastern part of Tokyo.

the large number of edges are treated, exact optimal solutions might not be obtained in a practically admissible time. Some approximation techniques are necessary for those cases. In this section, two such methods are provided with examples.

6.1. Reduction and extension methods

Given an original graph with $\hat{n}$ edges, suppose that the planner wants to construct a model with $\hat{r}$ edges. In the following, n and r of (P) and (P') are considered as parameters, and (n,r)-parameter families of (P) and (P') are treated. Let $r_1,\cdots,r_k$ be intergers satisfying $\hat{r}<r_1<\cdots<r_k<\hat{n}$, and suppose that (P) can be solved for all $(n,r)\varepsilon\{(\hat{n},r_k),(r_k,r_{k-1}),\cdots,(r_2,r_1),(r_1,\hat{r})\}$ even if it can not provide a solution of (P) for $(n,r)=(\hat{n},\hat{r})$ in a practically admissible time. $r_1,\cdots,r_k$ can be found empirically based on the experiences as in Figs.3 - 7. Then, a suboptimal solution for $(\hat{n},\hat{r})$ can be found by solving the sequence of problems (P) for $(\hat{n},r_k),(r_k,r_{k-1}),\cdots,(r_1,\hat{r})$. Such technique is called *Reduction method.*

Analogously, $r_1,\cdots,r_k$ may also be set to satisfy $0<r_1<\cdots<r_k<\hat{r}$. In this case, the solution is obtained by solving the sequence of (P') for $(\hat{n},r_1),(\hat{n},r_2),\cdots,(\hat{n},r_k),(\hat{n},\hat{r})$, where the edges obtained by solving (P') for $(\hat{n},r_j)$ are supposed to be unremovable when solving (P') for $(\hat{n},r_{j+1})$ for $j=2,\cdots,k$ $(\hat{r}=r_{k+1})$. The problem for $(\hat{n},r_1)$ is the same as (P) for $(\hat{n},r_1)$. This method is called *Extension method.*

Inferring from the data in Fig.3, the algorithm may work faster for the cases that r is large in (P) or (P'). Hence, Reduction method seems more effective than Extension method. However, both methods are suboptimal, and they may give different objective values. Thus, if both work well, the solution with the better objective value should be adopted. In Fig. 8 is shown an example from the total vehicle traffic network modeling for the south-eastern part of Tokyo, where Extension method is used. (An example for Reduction method is shown in the next section.) There, solutions for (P) with $(\hat{n},\hat{r})=(78,34)$ and $(78,42)$ are illustrated. In the procedures, $r_1,\cdots,r_k$ are taken to be $r_j=10+6(j-1)$ for $j=1,\cdots,5$, and $r_6=34$. For the problem with $(\hat{n},\hat{r})=(78,34)$, five (P')'s are solved, while six for the problem with $(\hat{n},\hat{r})=(78,42)$. The objective values for the obtained solutions are 659,396 and 813,975, respectively.

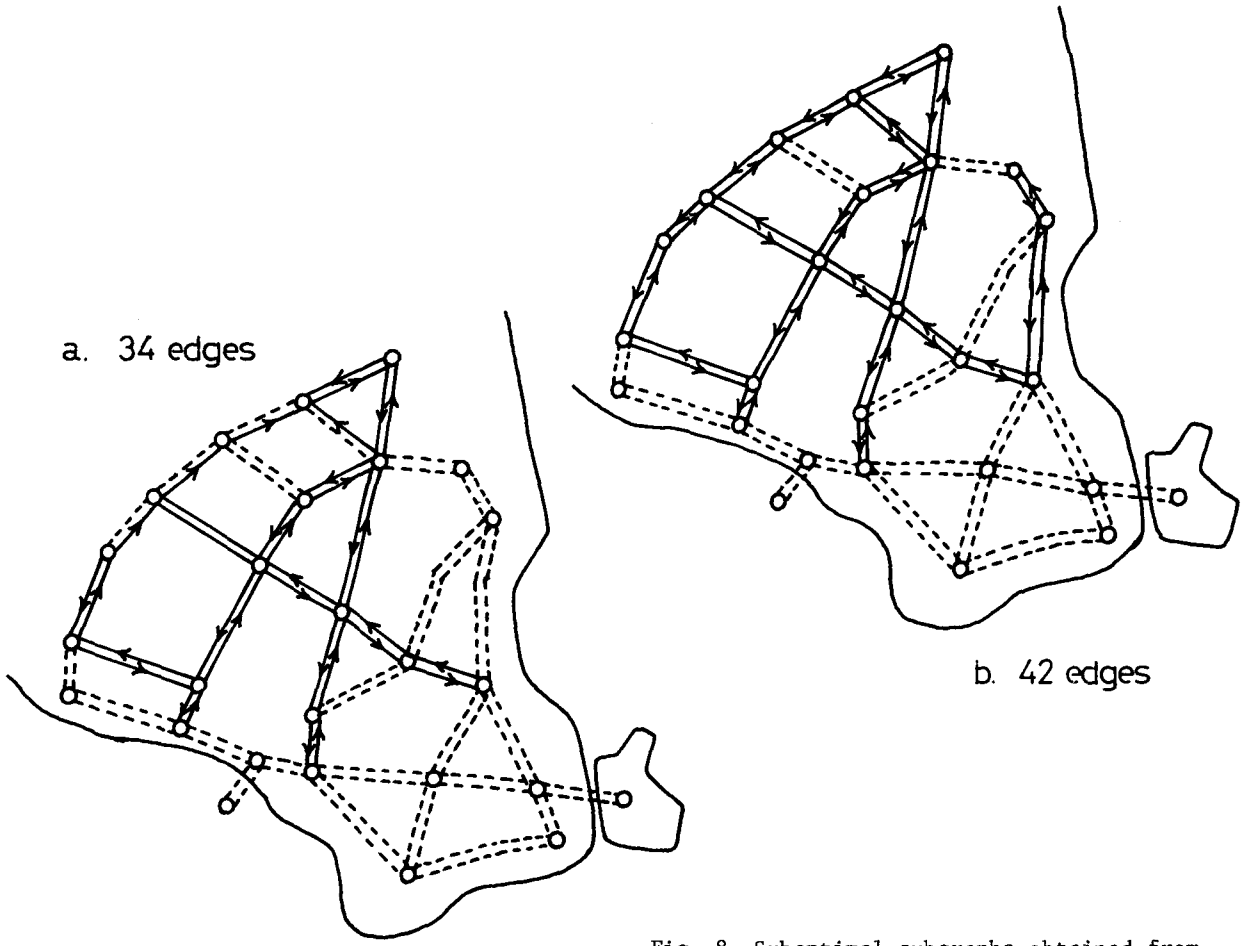


Fig. 8 Suboptimal subgraphs obtained from original graph with 78 edges (whose road structure is the same as that in Fig. 6 except that two directed edges correspond to one undirected edge) by Extension method for total vehicle traffic network modeling of the south-eastern part of Tokyo.

6.2. Decomposition method

Suppose that a large-scale network is decomposed into some subnetworks, an optimal model for each subnetwork is detected, and they are combined to construct a global suboptimal model. Such method is called *Decomposition method*.

Suppose that an original graph G with n edges is decomposed into m subgraphs by some procedure. Denote by $[k,j]$ the jth edge of the kth subgraph, i.e., $A=\{[k,j]\,|\,j=1,\cdots,h_k,\ k=1,\cdots,m\}$, where h_k is the number of edges included in the kth subgraph. Let c_j^k be the weight of $[k,j]$. Then, (P) can be rewritten as follows:

$$\begin{aligned}
&\text{maximize} \quad \sum_{k=1}^{m}\sum_{j=1}^{h_k} c_j^k y_j^k \\
&\qquad \text{subject to} \quad \sum_{k=1}^{m}\sum_{j=1}^{h_k} y_j^k \leqslant r \\
&\qquad\qquad y_j^k=0 \text{ or } 1,\quad j=1,\cdots,h_k,\ k=1,\cdots,m \\
&\qquad\qquad \text{subgraph of } G \text{ defined by } \{[k,j]\,|\,[k,j]\in A, \\
&\qquad\qquad y_j^k=1,\ j=1,\cdots,h_k,\ k=1,\cdots,m\} \text{ is strongly} \\
&\qquad\qquad \text{connected.}
\end{aligned}$$

Note that $y_j^k=1$ if $[k,j]$ is adopted in the model, and $\sum_{k=1}^{m} h_k=n$.

If an optimal solution for the over-all graph G also provides an optimal solution for each subgraph k $(k=1,\cdots,m)$, the above problem is equivalent to the following decomposition problem:

master problem

$$\begin{aligned}
&\underset{r_k}{\text{maximize}} \quad \sum_{k=1}^{m} f_k(r_k) \\
&\qquad \text{subject to} \quad \sum_{k=1}^{m} r_k \leqslant r \\
&\qquad\qquad r_k \geqslant 0,\ \text{integer},\ k=1,\cdots,m \\
&\qquad\qquad \text{subgraph of } G \text{ defined by } \{[k,j]\,|\,[k,j]\in A, \\
&\qquad\qquad y_j^k(r_k)=1,\ j=1,\cdots,h_k,\ k=1,\cdots,m\} \text{ is strongly} \\
&\qquad\qquad \text{connected,}
\end{aligned} \tag{DP}$$

$$\begin{array}{lll} \textit{subproblem}_k \textit{ for } r_k \ (k=1,\cdots,m) & & \\ \text{maximize} & \sum_{j=1}^{h_k} c_j^k y_j^k & \\ y_j^k & \text{subject to} & \sum_{j=1}^{h_k} y_j^k \leq r_k \\ & & y_j^k=0 \text{ or } 1,\ j=1,\cdots,h_k \\ & & \text{subgraph of } G \text{ defined by } \{[k,j] \mid [k,j] \in A, \\ & & y_j^k=1,\ j=1,\cdots,h_k\} \text{ is strongly connected,} \end{array}$$

where r is the least upper bound of the number of edges in the model, $(y_1^k(r_k),\cdots,y_{h_k}^k(r_k))$ is an optimal solution of the subproblem k in which the least upper bound of the number of edges in the subgraph is r_k, and $f_k(r_k)$ is its corresponding objective value.

The master problem of (DP) can be solved in the following way: if the strong connectedness condition is ignored, an optimal, 2nd optimal, 3rd optimal,$\cdots$, solutions can be found in order by using dynamic programming [2]. The ith optimal solution satisfying the strong connectedness condition for the smallest i is an optimal solution of the master problem.

Each subproblem has the same form as (P) and can be solved by the presented algorithm. The number of subproblems to be solved might be large, however, the whole computation time can be made admissible if the decomposition is adequate.

Generally, an optimal solution of the over-all graph may not provide optimal solutions of subgraphs. Thus, (DP) may not provide an exact optimal solution of the original problem.

Two main methods exist for decomposition of graphs: (a) edge decomposition and (b) node decomposition as shown in Fig.9-a. In both methods, it can be seen that strong connectedness of the over-all graph and that of subgraphs are usually far from each other. But, by introducing the procedure derived from the following property, such difficulty may be fairly eliminated:

Property. Let G be a strongly connected graph. Then, for a subgraph G' obtained by edge or node decomposition of G, the graph defined by identifying as one node (i) $\{v_j \mid (v_i,v_j)$ is a cut edge, $v_i \in V'\}$ if edge decomposition is adopted, or (ii) $\{v_j \mid v_j$ is a cut node$\}$ if node decomposition is adopted, is strongly connected, where V' is the set of nodes of G'.

The proof of Property is straightforward. By using connectedness in place of strong connectedness, it also holds for undirected graphs.

Fig.9-b shows subgraphs obtained by edge and node decompositions shown in Fig.9-a.

If edge decomposition is adopted, some edges are made belong to several subgraphs as shown in Fig.9-b. Hence, in this case, $\sum_{k=1}^{m} h_k = n$ does not hold. However, by making each of those edges include in any one subgraph to which it belongs and treating as a dummy edge in the other subgraphs to which it belongs, Decomposition method works by using (DP). Concretely, it is counted for only one j $(1 \leqslant j \leqslant m)$ when $r_1, \cdots, r_m$ and $f_1(r_1), \cdots, f_m(r_m)$ are considered in the master problem.

It is easy to see that Decomposition method is only suboptimal even if the above procedure is introduced. Furthermore, it may consume admissible but fairly large computation time. However, as it may generate a good solution, it can not be abandoned.

An example under the same situation as in 6.1 is solved by using edge decomposition. The original network with n=78 is the same as shown in Fig.8. The decomposition procedure is illustrated in Fig.10, and the results for r=34 and r=42 are shown in Fig.11. The objective values obtained are 659,745 and 813,975, respectively. Comparing with the results obtained by Extension method in 6.1, Decomposition method has generated a better model for the problem with r=34. While, both methods have provided the same model for the problem with r=42.

7. Vehicle Traffic Network Planning of Nagoya City Area

In this section, the results for network modeling of Nagoya city area by using Reduction method are shown for illustrating the applicability of the presented algorithm to much larger-scale networks.

The purpose of modeling is to construct a basic network model of the road network of Nagoya city area for planning of a vehicle traffic control system of Nagoya city area. The road network is regarded as a directed graph with the edge weights representing total vehicle traffic volume/day of the directed links.

First, the original graph with 69 nodes and 204 edges is considered, which represents the southern part of Nagoya city area. A model with 100 edges has been obtained as shown in Fig.12, where the sequence of 5 problems (r_j=118+20(j-1) for j=1,$\cdots$,4 using the notation described in the previous section) was solved. In four of those problems, the subgraph searched first

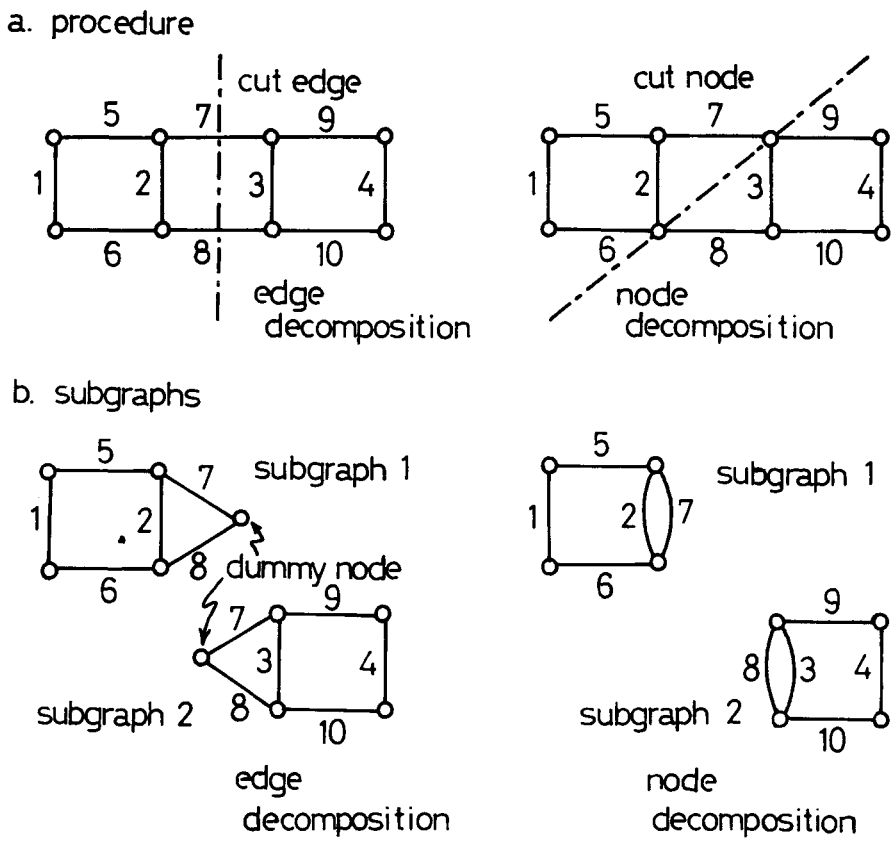


Fig. 9 Procedures and obtained subgraphs for edge and node decompositions of graphs.

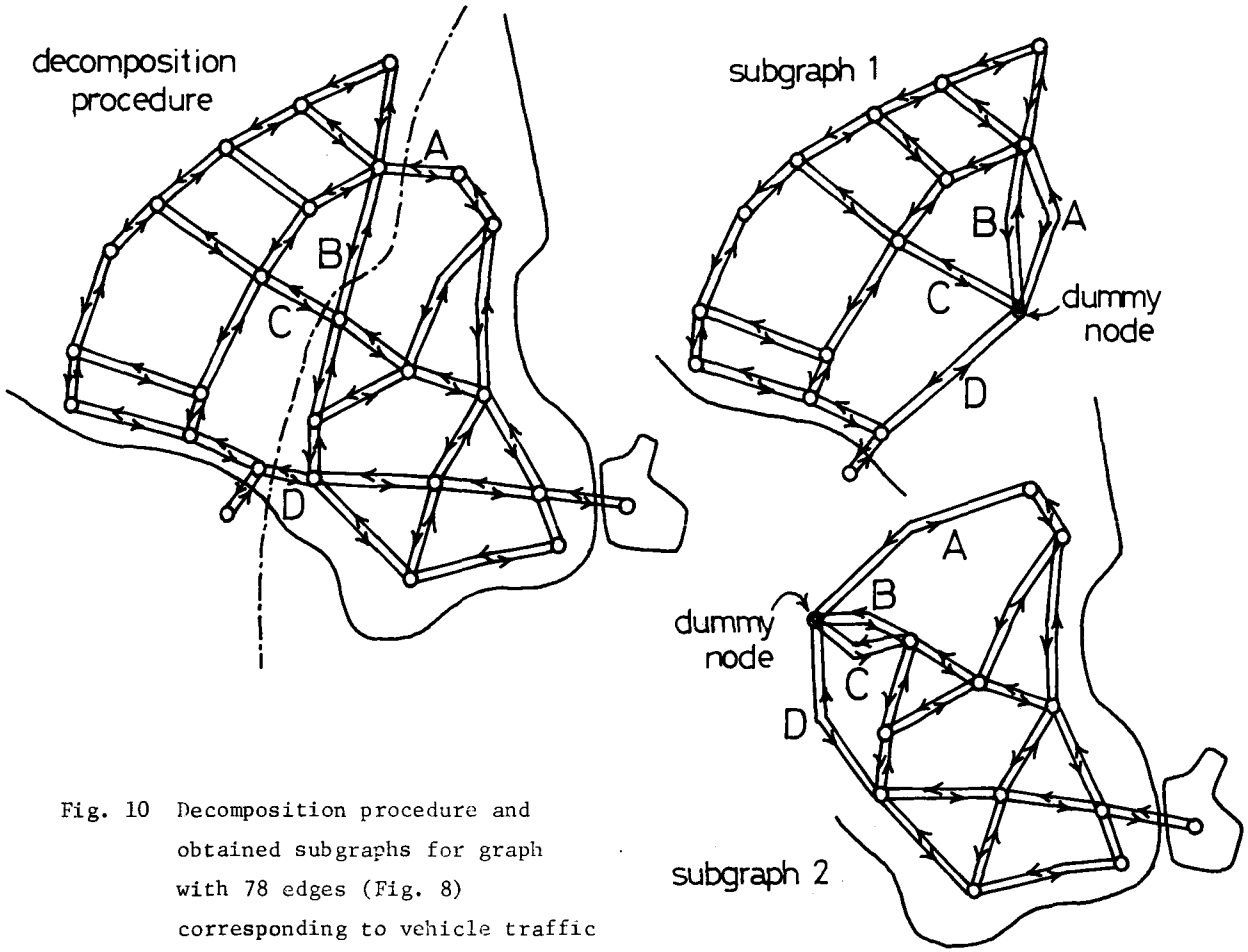


Fig. 10 Decomposition procedure and obtained subgraphs for graph with 78 edges (Fig. 8) corresponding to vehicle traffic network of the south-eastern part of Tokyo.

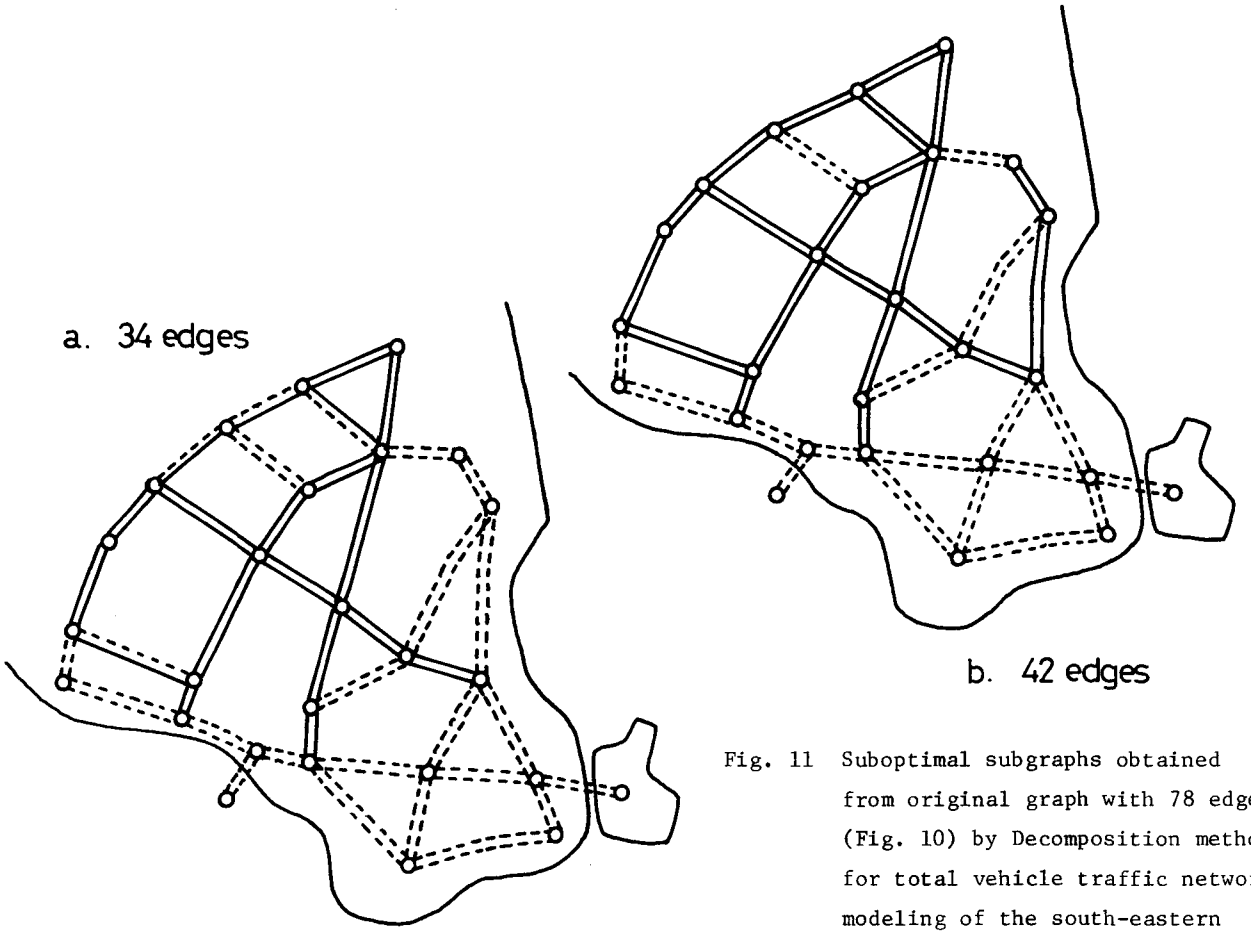


Fig. 11 Suboptimal subgraphs obtained from original graph with 78 edges (Fig. 10) by Decomposition method for total vehicle traffic network modeling of the south-eastern part of Tokyo.

was optimal. The remained problem searched 551 subgraphs.

Next, a model having 350 edges for Nagoya city area inside Nagoya Loop Roads is searched. The original graph has 206 nodes and 670 edges. The result is shown in Fig.13, where the sequence of 16 problems was solved (r_j= 370+20(j-1) for j=1,$\cdots$,15). In each of those problems, the subgraph searched first was optimal. It implies that the subgraphs of fairly large sizes defined by 'important' edges of physically correlated networks may usually be themselves strongly connected. It may save computation time considerably.

The suboptimal subgraphs obtained from the above two problems show that some of the principal roads of Nagoya play main roles in construction of the models and the planner can find what roads or avenues are basically important for his objective. Especially, the suboptimal model for the second problem shows that it contains roads more in the northern part of Nagoya than in its southern part.

The original graph of the first problem was selected so as to include the southern part of that of the second problem. It can be seen in Figs. 12 and 13 that the model for the former is slightly different from the southern part of that for the latter. Especially, some of the roads belonging to Nagoya Loop Roads contained in the former model do not appear in the latter model. It implies that more refined models may be constructed by applying the presented methods to some large or small parts of a city and analyzing the obtained models.

In that sense, models obtained by the presented algorithm may be only basic or first-order for the planner, and he would have to modify them so that they could be handled more conveniently. However, it can be seen from the above results that the presented algorithm is applicable as a fundamental algorithm to practical network modeling problems.

8. Conclusion

The modeling problem of urban road networks was formulated as a combinatorial optimization problem, and its solution algorithm based on implicit enumeration was presented. Some extensions for modeling with restrictions and suboptimal methods for large-scale networks were also given.

Computational examples from urban vehicle traffic networks of Tokyo and Nagoya city areas were shown so that the effectiveness and applicability of the algorithm were verified. A planner of an urban road network can

obtain a network model reflecting his objective for practical use by the presented algorithm, and more refined models can be expected if he constructs some models for large or small parts of the original network,analyzes them and integrates the results so as to get one global model.

Acknowledgements

The author is greatly indebted to Profs. Y.Hayashi, H.Kawashima, Mr. A. Fujino and Mr. Y.Takami of Keio University for their helpful supports on this research. Also he appreciates Mr. M.Matsui and Mr. T.Enomoto of Toyota Motor Sales Co. Ltd. for their supports on computer experiments.

References

1. Balas, E.: An Additive Algorithm for Solving Linear Programs with Zero-One Variables. *Operations Research*, Vol. 13 (1965), pp. 517-546.
2. Bellman, R. and Kalaba, R.: On k-th Best Policies. *J. Soc. Indust. Appl. Math.*, Vol. 8 (1960), pp. 582-588.
3. Garfinkel, R. and Nemhauser, G.: *Integer Programming*, John Wiley and Sons, New York, 1972.
4. Glover, F.: A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem, *Operations Research*, Vol. 13 (1965), pp. 879-919.
5. *On Aggregation in Data Generation* Report T-73-2, Operations Research Society of Japan, 1973.(in Japanese)

Yuichiro ANZAI: Department of Administration Engineering, Faculty of Engineering, Keio University
832 Hiyoshi-cho, Kohoku-ku,
Yokohama, 223, Japan

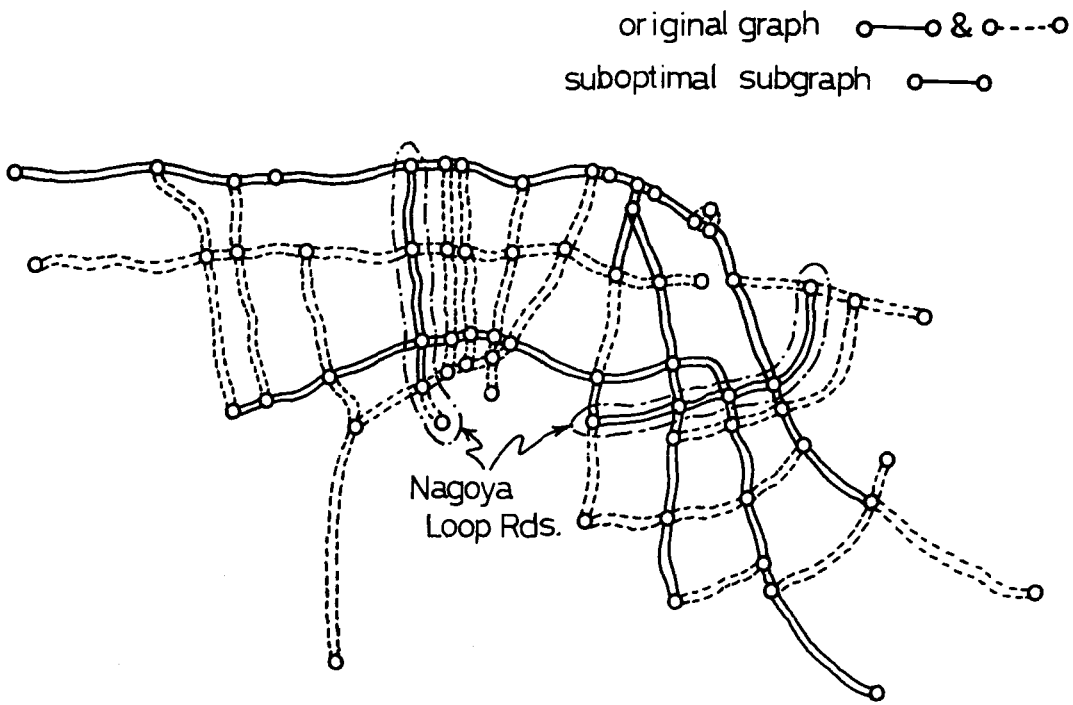


Fig. 12 Original graph with 204 edges and suboptimal subgraph with 100 edges obtained by Reduction method for total vehicle traffic network modeling of the southern part of Nagoya. (Arrowheads for directed edges are omitted in the figure.)

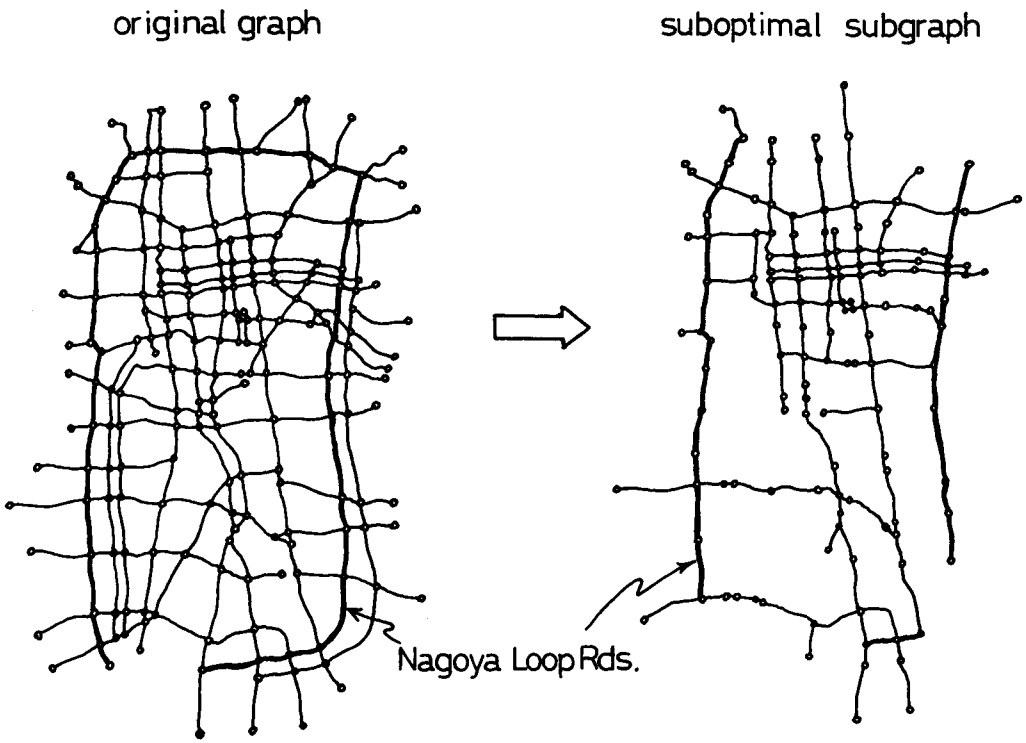


Fig. 13 Original graph with 670 edges and suboptimal subgraph with 350 edges obtained by Reduction method for total vehicle traffic network modeling of Nagoya city area inside Nagoya Loop Roads. (Each pair of two directed edges connecting two nodes is illustrated as one line segment joining two nodes in the figure for simplicity.)