Network Design Application of an Extraction Algorithm for Network Aggregation

ALI E. HAGHANI AND MARK S. DASKIN

The performance of a network extraction algorithm is described, and the algorithm is tested by using the network design problem. A network is chosen as the original network and is aggregated at different levels. The results of the optimal decision making under a common set of alternative actions are then compared against the original network design problem, and to allow a broader range of policy options to be tested in a fixed amount of computer time than would be allowed by using the original disaggregated network.

Network aggregation is the art and science of condensing a given network into another one that (a) is small enough to be managed efficiently and effectively, and (b) preserves some desired characteristics or satisfies certain objectives or both (1). The usefulness of network aggregation schemes is particularly evident in instances when similar problems are to be solved on a network, or sensitivity analyses of various types are to be performed. Dealing with the detailed network in solving such problems entails high costs in terms of computer storage and time.

There are two main approaches to the network aggregation problem: network element (link or node) extraction and network element abstraction. Extraction of network elements means deletion of the elements of the network that are identified as being insignificant based on a prespecified criterion. Abstraction of the elements collapses the insignificant ones into pseudo or dummy elements. Network element extraction has the disadvantage of causing network disconnection because of the removal of links. As a result, the remaining links of the network will be overloaded if the origin-destination (O-D) trip matrix is not adjusted appropriately. Network element abstraction is more difficult to perform. It is hard to transform the original network into an aggregate one, and moreover, it is even harder to translate the actions taken on the aggregate network into actions on the detailed network because of drastic changes in the topology of the network that occur during the aggregation process.

The primary objective in developing a network aggregation scheme should be to find an aggregation process that, when applied to a detailed network, results in an aggregate network that retains the physical appearance of the original one as much as possible. Thus when solving a decision-making problem, such as the network design problem on the aggregate network, the results should be easily transferable to the original one. With this in mind, and because the abstraction process changes the topology of the network and cannot effectively serve the process, it is proposed that an aggregation algorithm, which focuses primarily on link extraction, be used. Node extraction is a process that follows link extraction; when all links incident to a node are extracted, the node will be extracted. The algorithm is presented in the next section.

NETWORK EXTRACTION ALGORITHM

Let \( N(V,A) \) be a network, where \( V \) is the set of vertices or nodes and \( A \) is the set of arcs or links. Let \( T \) be the set of destinations and \( S \) be the set of origins, \( S \cap T = V \). Let \( x_i^j \) be the flow over link \( i \) destined to \( t \), \( x_i^j \) be the flow over link \( i \) originated from \( s \), and \( x_i^s \) be the flow over link \( i \) that originates from \( s \) and is destined to \( t \), \( s \in S \) and \( t \in T \). Also, let \( x_i^j \) denote the flow over link \( i \),

\[
\begin{align*}
    x_i^j &= \sum_{s \in S} x_i^s = \sum_{t \in T} x_i^j, \quad i \in A, s \in S, t \in T
\end{align*}
\]

Moreover, let \( D \) be the O-D trip matrix. Finally, let \( C_i(x_i) \) represent the average cost of travel on link \( i \) at flow \( x_i \) that is continuous, differentiable, Riemann integrable, convex, and strictly increasing. It is assumed that the distribution of flow over a transportation network is based on Wardrop's first principle (2)—user equilibrium (3). There are some links in the network that, after the distribution of the flow has taken place, will not carry a significant amount of traffic. These links are the ones that will be focused on in the aggregation process.
to be described. The criterion used for the identification of insignificant links is defined as follows.

A link in a network is insignificant if the corresponding equilibrium flow is below a percent of the maximum equilibrium link flow in the network. The level of network aggregation changes, depending on the value of \( \alpha \). As \( \alpha \) increases, the network becomes more aggregated and vice versa.

The reason for choosing the level of flow in the links as a criterion for identifying the insignificant links is that many transportation problems deal with the equilibrium flow levels in the network. It has already been proved (1) that the equilibrium level in the significant or nonextracted links remains unchanged when the aggregation scheme is applied. By preserving the level of equilibrium flow in the nonextracted links, the aggregation scheme should produce an aggregate network that is more representative of the detailed network than is an aggregation process that failed to preserve these flow levels. This should be particularly important in solving problems in which the objective function is based on the level of equilibrium flow in the links. The network design problem is one such problem.

Thus the network extraction algorithms as it has been coded, is presented. A more rigorous presentation is included elsewhere (1). The inputs to the algorithm are the specifications of the original network \( N(V,A) \), the average link cost functions \( C_i(x_i) \), \( i \in A \), and the O-D trip matrix \( B \). Either the maximum number of links to be extracted or the maximum \( \alpha \) percentile denoting the cutoff point between the insignificant and significant link flows should also be given. Through this process certain prespecified links in the aggregate network can be maintained; also, specific links can be extracted. Furthermore, the algorithm extracts the links one by one and provides the results after each iteration. As a result, several different aggregate networks are obtained. The principle is to extract insignificant links and to update the trip matrix such that the flow level in the remaining links of the aggregate network remains unchanged. The algorithm is as follows.

**Step 1:** Specify \( \alpha \) or the maximum number of links that may be extracted \( M \). Solve the equilibrium flow problem. Let \( x_i^0 \), \( i \in A \) be the equilibrium flow on link \( i \).

**Step 2:** Identify the unextracted link \( k \) with the minimum flow. Let \( t(k) \) and \( h(k) \) denote the tail and head nodes of link \( k \). Compute

\[
\alpha_k = x_k^0/\text{Max} (x_i^0)
\]

If \( \alpha_k > \alpha \) as specified in step 1, or if the number of extracted links is greater than the maximum number of links that may be extracted (specified in step 1), stop. Otherwise disaggregate the flow on link \( k \) by specifying the origin and destination of all flow on the link, which is done by solving the equilibrium flow problem on the most aggregate network generated. (An outline of how the O-D specific link flow \( d_k^0 \) may be obtained from the solution procedure to the equilibrium flow problem, and the problems associated with the nonuniqueness of this quantity, are discussed elsewhere (1).) Go to step 3.

**Step 3:** Discard link \( k \). Declare \( t(k) \) a destination (if it is not already such a node) and \( h(k) \) an origin (if it is not already such a node).

**Step 4:** Update the trip matrix as follows: (a) type I entry, where \( t(k) \) is a destination, i.e.,

\[
d_k(t) = d_k^0(t) + x_k^0
\]

where \( d_k^0(t) \) is the original O-D matrix element and is taken to be zero if the \( t(k) \) is a new destination, and \( d_k^0(t) \) is the updated O-D matrix element; (b) type II entry, where \( h(k) \) is an origin, i.e.,

\[
d_k(h) = d_k^0(h) + x_k^0
\]

where \( d_k^0(h) \) is the original O-D matrix element and is taken to be zero if \( h(k) \) is a new origin, and \( d_k^0(h) \) is the updated O-D matrix element; and (c) type III entries, where all remaining entries of the O-D trip matrix \( d_k^0 \) are substituted by \( d_k = d_k^0 - x_k^0 \) (subtracting from \( d_k^0 \) the part of the demand from \( s \) to \( t \) that is now destined to a new destination and that will reoriginate from a new origin).

Certain properties of the algorithm are worth mentioning. First, as previously noted, the algorithm preserves the level of equilibrium flow in the links of the network that are not extracted. Second, in cases in which all of the nodes of the network are not both origins and destinations, the algorithm will increase the number of origins and destinations in the aggregate network. This in turn might have adverse effects on the computation time of the network design problem by increasing the number of origins and the associated time for computation of the minimum paths in the network. This situation has not been examined in this paper. However, this increase in computation time should be offset through other means.

Third, the result of the extraction process may be a set of disconnected subnetworks. If this occurs, the analysis of the aggregate network, now a set of subnetworks, will be much easier to undertake. In fact, in cases in which link extraction will increase the number of origins and destinations (and thereby increase the computation time for the network design problem), specification of the links to be extracted can force the aggregate network to be a set of disconnected subnetworks. In this way the computational savings obtainable by having disconnected subnetworks may be used to offset the increased time that results from additional origins and destinations.

Finally, in the network design problem it is shown that for a given budget level, the total cost to the users of the network, as measured on the detailed network, is overestimated by the solution to the network design problem that uses the aggregate network (1).

In the next section this algorithm is applied to an original network, and the network design problem is solved on the original detailed network and on a series of aggregate networks.

**Applications of Network Extraction Algorithm to Network Design Problem**

**Problem Description**

The network design problem is that of finding a set of feasible actions or projects from among a collection of such actions that, when implemented, optimize the objective function(s) being considered. The feasibility of a set of actions is determined by resource, physical, and environmental constraints (4). Traditionally, the objective function in the network design problem has been formulated as the minimization of the total number of vehicle hours of travel on the network, with flows and travel times...
Transportation Research Record 944

computed based on user equilibrium. This is represented as

$$\text{Minimize } \sum_{i \in A} x_i^* C_i(x_i^*)$$

(5)

Subject to budget constraint on the cost of implemented projects p

where $x_i^*$ is the user equilibrium flow on link i, and P is a set of projects (p) under consideration for implementation. In solving the network design problem, a modified objective function, which was suggested by Poorzahedy (4) in his algorithm I, has been used. This form is as follows:

$$\text{Minimize } \sum_{i \in A} x_i^* C_i(x_i^*)$$

(6)

Subject to budget constraint on the cost of implemented projects p

where $x_i^*$ is the user equilibrium flow on link i.

The modified form of the problem was selected because of the availability of a computer code to solve this problem. Also, solving this form of the problem has been found to be more efficient than solving the traditional formulation and generally results in similar actions being taken on the network (5).

Thus the results of a set of experiments designed to test the effectiveness of the proposed NA algorithm in solving the modified network design problem can be presented. For the detailed network, the Sioux Falls, South Dakota, network is used in the experiments because it is a well-documented network and has been used by other researchers (4,6,7) in analyzing network design problems.

The detailed network, which consists of 24 nodes and 76 links (or 38 link pairs, allowing two-way traffic movements), is shown in Figure 1 (4). The link travel costs $[C_i(x_i^*)]$ are given by functions of the form

$$C_i(x_i) = a_i + b_i (x_i)^4$$

(7)

The constants $a_i$ and $b_i$ for each of the existing links in the network, as well as for the six candidate links, are given in Table 1 (4). Also provided in the table is the cost of implementing each of the candidate links. The first five projects represent improvements on existing links, whereas the sixth project is an entirely new link. Two different sets of experiments were considered. In the first, only the five improvement projects are used; in the second, all six candidate projects are used. The O-D matrix for this network is given in Poorzahedy (4).

Five aggregate networks are developed, which result from the extraction of 6, 12, 18, 24, and 30 links. The aggregate networks are shown in Figures

Figure 1. Original network (4).

Legend:

1. Node i
2. Project Number
3. Link Pair $j, j+1$
4. Candidate Project on Existing Links
5. Candidate Project, New Link
The five aggregate networks shown in Figures 2-6 and the detailed network shown in Figure 1, along with their corresponding O-D matrices, constitute the basis for the experiments. On each of these six networks, two network design problems were solved: one with the first five candidate projects, and the second with all six projects. The initial budget was set at $2,000,000 in all cases, and a complete sensitivity analysis (with respect to increases in the budget) was performed for all six networks and both design problems.

Results of the Five-Project Experiment

The results of the five-project experiment are summarized in Table 2. More detailed results are given in Haghani (1). The data in the table report (a) the percentage error in the total number of vehicle hours on the aggregate networks as compared with the detailed network, and (b) the number of projects that are selected differently when the network design problem is solved on the detailed and aggregate networks. Note that there are 18 unique budget levels that must be considered in performing the sensitivity analyses, beginning with a budget of $2,000,000 and ending with a budget of $4,325,000, which allows the implementation of all five projects. Of the 18 budget levels, 5 result in different solutions for the network design problem on the original and aggregate networks in the worst case.

The data in Table 2 indicate that with six links deleted from the original network, the solutions to the network design problem on the original network and the aggregate network are identical for all budget levels. For higher levels of aggregation, discrepancies occur between the solution using the aggregate network and that found using the original network. Also note that most of the errors occur when the ratio of the budget level to the total cost

**Table 1.** Link parameters of test network III (4).

<table>
<thead>
<tr>
<th>Link</th>
<th>a(x10^-2)</th>
<th>b(x10^-4)</th>
<th>Link</th>
<th>a(x10^-2)</th>
<th>b(x10^-4)</th>
<th>Link</th>
<th>a(x10^-2)</th>
<th>b(x10^-4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>5.96</td>
<td>0.00023</td>
<td>3,4</td>
<td>4.34</td>
<td>0.00017</td>
<td>5,6</td>
<td>5.17</td>
<td>0.12408</td>
</tr>
<tr>
<td>3,4</td>
<td>35,36</td>
<td>2.98</td>
<td>5,6</td>
<td>37,38</td>
<td>4.52</td>
<td>7,8</td>
<td>4.31</td>
<td>0.00069</td>
</tr>
<tr>
<td>5,6</td>
<td>57,69</td>
<td>2.17</td>
<td>7,8</td>
<td>41,42</td>
<td>3.50</td>
<td>9,10</td>
<td>4.14</td>
<td>0.00016</td>
</tr>
<tr>
<td>7,8</td>
<td>71,72</td>
<td>3.72</td>
<td>9,10</td>
<td>43,44</td>
<td>1.67</td>
<td>11,12</td>
<td>2.16</td>
<td>0.00035</td>
</tr>
<tr>
<td>9,10</td>
<td>67,68</td>
<td>1.60</td>
<td>11,12</td>
<td>45,46</td>
<td>2.69</td>
<td>13,14</td>
<td>6.46</td>
<td>0.15504</td>
</tr>
<tr>
<td>13,14</td>
<td>47,48</td>
<td>2.31</td>
<td>15,16</td>
<td>49,50</td>
<td>4.46</td>
<td>17,18</td>
<td>5.03</td>
<td>0.00755</td>
</tr>
<tr>
<td>15,16</td>
<td>0.00017</td>
<td>cost $625.x10^3</td>
<td>17,18</td>
<td>51,52</td>
<td>3.99</td>
<td>19,20</td>
<td>2.18</td>
<td>0.00008</td>
</tr>
<tr>
<td>17,18</td>
<td>0.05544</td>
<td>cost $650.x10^3</td>
<td>19,20</td>
<td>53,54</td>
<td>5.72</td>
<td>21,22</td>
<td>9.61</td>
<td>0.23064</td>
</tr>
<tr>
<td>19,20</td>
<td>67,68</td>
<td>1.60</td>
<td>21,22</td>
<td>55,56</td>
<td>4.71</td>
<td>23,24</td>
<td>4.82</td>
<td>0.11568</td>
</tr>
<tr>
<td>21,22</td>
<td>0.00017</td>
<td>cost $850.x10^3</td>
<td>23,24</td>
<td>57,58</td>
<td>1.67</td>
<td>25,26</td>
<td>5.00</td>
<td>0.00750</td>
</tr>
<tr>
<td>23,24</td>
<td>0.04008</td>
<td>cost $1000.x10^3</td>
<td>25,26</td>
<td>59,60</td>
<td>3.29</td>
<td>27,28</td>
<td>5.87</td>
<td>0.00265</td>
</tr>
<tr>
<td>25,26</td>
<td>0.07896</td>
<td>cost $1200.x10^3</td>
<td>27,28</td>
<td>61,62</td>
<td>4.00</td>
<td>29,30</td>
<td>8.04</td>
<td>0.19296</td>
</tr>
<tr>
<td>27,28</td>
<td>0.09600</td>
<td>cost $1500.x10^3</td>
<td>29,30</td>
<td>63,64</td>
<td>4.25</td>
<td>31,32</td>
<td>6.46</td>
<td>0.15504</td>
</tr>
<tr>
<td>29,30</td>
<td>0.04512</td>
<td>cost $1500.x10^3</td>
<td>31,32</td>
<td>65,66</td>
<td>1.88</td>
<td>33,34</td>
<td>4.42</td>
<td>0.10608</td>
</tr>
<tr>
<td>33,34</td>
<td>0.00124</td>
<td>cost $1500.x10^3</td>
<td>33,34</td>
<td>67,68</td>
<td>2.75</td>
<td>35,36</td>
<td>35,36</td>
<td>2.98</td>
</tr>
<tr>
<td>35,36</td>
<td>69,70</td>
<td>2.17</td>
<td>35,36</td>
<td>71,72</td>
<td>3.72</td>
<td>37,38</td>
<td>71,72</td>
<td>3.72</td>
</tr>
<tr>
<td>37,38</td>
<td>71,72</td>
<td>3.72</td>
<td>37,38</td>
<td>73,74</td>
<td>2.50</td>
<td>39,40</td>
<td>73,74</td>
<td>2.50</td>
</tr>
<tr>
<td>39,40</td>
<td>75,76</td>
<td>4.50</td>
<td>39,40</td>
<td>75,76</td>
<td>4.50</td>
<td>41,42</td>
<td>75,76</td>
<td>4.50</td>
</tr>
<tr>
<td>41,42</td>
<td>77,78</td>
<td>-</td>
<td>41,42</td>
<td>77,78</td>
<td>-</td>
<td>New Data</td>
<td>43,44</td>
<td>77,78</td>
</tr>
</tbody>
</table>

Note: a parameters are given in hours, and b parameters are given in hours + (1,000 vehicles per day) +

2-6; the resulting O-D matrices are given in Haghani (1).

Results for the Aggregation Model

The results of the five-project experiment are summarized in Table 2. More detailed results are given in Haghani (1). The data in the table report (a) the percentage error in the total number of vehicle hours on the aggregate networks as compared with the detailed network, and (b) the number of projects that are selected differently when the network design problem is solved on the detailed and aggregate networks. Note that there are 18 unique budget levels that must be considered in performing the sensitivity analyses, beginning with a budget of $2,000,000 and ending with a budget of $4,325,000, which allows the implementation of all five projects. Of the 18 budget levels, 5 result in different solutions for the network design problem on the original and aggregate networks in the worst case.

The data in Table 2 indicate that with six links deleted from the original network, the solutions to the network design problem on the original network and the aggregate network are identical for all budget levels. For higher levels of aggregation, discrepancies occur between the solution using the aggregate network and that found using the original network. Also note that most of the errors occur when the ratio of the budget level to the total cost...
Figure 2. Aggregate network with 6 extracted links.

Figure 3. Aggregate network with 12 extracted links.

Note: Legend is given in Figure 1.
Figure 4. Aggregate network with 18 extracted links.

Figure 5. Aggregate network with 24 extracted links.

Note: Legend is given in Figure 1.
Figure 6. Aggregate network with 30 extracted links.

Table 2. Percentage of vehicle hour errors and number of misselected projects for five-project case.

<table>
<thead>
<tr>
<th>Budget Levels ($000s)</th>
<th>No. of Budget Levels</th>
<th>Vehicle Hour Error (%)</th>
<th>No. of Misselected Projects</th>
<th>Vehicle Hour Error (%)</th>
<th>No. of Misselected Projects</th>
<th>Vehicle Hour Error (%)</th>
<th>No. of Misselected Projects</th>
<th>Vehicle Hour Error (%)</th>
<th>No. of Misselected Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>B = 2.000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3.11</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3.11</td>
<td>1</td>
</tr>
<tr>
<td>B = 2.050</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6.73</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6.73</td>
<td>1</td>
</tr>
<tr>
<td>2.125 &lt; B &lt; 2.475</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3.82</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3.82</td>
<td>1</td>
</tr>
<tr>
<td>2.475 &lt; B &lt; 2.675</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>7.95</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>7.95</td>
<td>1</td>
</tr>
<tr>
<td>B = 2.675</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.700 &lt; B &lt; 3.325</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3.325 &lt; B &lt; 4.325</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B = 4.325</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

of all candidate links is low. In all cases in which the solution on the aggregate network differed from the solution on the detailed network, the number of misselected links was only one. By using vehicle hours as the measure of effectiveness, the maximum percentage error is 7.95 percent. The identity of the errors, the equality of their severity, and the similarity of their frequency across the various levels of aggregation suggest that the size of the network may be reduced significantly without increasing the magnitude of the errors. This phenomenon is also apparent in the case of six projects.

(Note that the maximum percentage error of 7.95 percent is computed as follows. At a given budget level, let $\mathbf{y}_k$ and $\mathbf{y}_0$ be the optimal solutions to the network design problem for the aggregate and original networks, respectively, where $\mathbf{y} = (y_i)$ and $\mathbf{y}_1 = (0,1)$ if project $i$ (is not, is) chosen to be in the optimal set. Also, let $V(\mathbf{y})$ represent the decrease in the total number of vehicle hours in the original network that results from implementing project set $\mathbf{y}$. The percentage error is defined as $\left| V(\mathbf{y}_0) - V(\mathbf{y}_1) \right| / V(\mathbf{y}_0) \cdot 100$.)

Note also that the total travel time on the network is overestimated by the solution to the design problem on the aggregate networks as compared with the total time found when using the detailed network. This is also shown in the six-project experiment and, as noted at the end of the section on Network Extraction Algorithm, may be shown to be a general property of the aggregation process.
Results of the Six-Project Experiment

The results of the six-project experiment are given in Table 3. Again, the data in the table report two sets of statistics: (a) the percentage error in the total number of vehicle hours on the aggregate networks as compared with the number on the detailed networks, and (b) the number of projects that are selected differently when the network design problem is solved on the detailed and aggregate networks. [More detailed results are given in Haghani (1).]

There are now 40 unique budget levels that must be considered in performing the budget sensitivity analysis. The 40 budget levels, 9 result in different solutions for the network design problem on the detailed network and the aggregate networks in the worst case.

Again, when only six links are extracted from the network, the results on the detailed network and on the aggregate network are identical. As the level of network aggregation is increased, the number of budget levels at which discrepancies occur between the solutions on the two networks increases: when 12 or 18 links are extracted, errors occur at 5 of the 40 budget levels, or 12.5 percent of the time; with 24 links extracted, errors occur 20 percent of the time (in 8 cases); and with 30 links extracted, errors occur 22.5 percent of the time (in 9 cases). Again, the maximum number of misselected projects is 1, and the maximum difference in vehicle hours is 7.95 percent.

The relatively sharp increase in the number of budget levels at which errors occur, as the level of aggregation is increased from 18 extracted links to 24, suggests that a trade-off must be made between the expected accuracy of the results and the level of detail preserved in the network. The desired point on this trade-off curve depends on the use to which the analysis will be put; this is a decision best left to the analyst in each case.

Finally, there appear to be patterns to the misselected links. In both sets of experiments the aggregate network solution replaced project 2 or project 6 with project 4 in the six-project experiment, the aggregate solution also replaced project 6 with project 4. These replacements appear to be related to the degree of network aggregation in the neighborhood of the candidate projects. Also, because the errors occur at the same budget levels in all three cases, there appears to be a relationship between the errors and the specified budget levels.

A more complete exploration of the relationship between these errors and (a) the degree of network aggregation in the neighborhood of the candidate projects and (b) the specific budget levels would be an interesting area for future research. The nature of the errors found in this analysis are discussed in greater detail later in the paper.

Computation Times

One of the major reasons for implementing a network aggregation process is, as already noted, the savings in computer time that should result from solving problems on smaller, less-detailed networks. The central processing unit (CPU) times required for the solution of the two network design problems, with all the budget sensitivity analyses on each of the six networks, are given in Table 4. All CPU times are for the Univac 1100, unless otherwise noted.

The network design problem for the five-project case may be solved 6 times faster on the most aggregate network considered in this analysis than on the original detailed network. In the six-project case, the ratio of CPU time on the detailed network to that on the most aggregate network is 4.4. However, these savings must be offset by the CPU time required to perform the aggregation process before the network design problem is solved. In some cases this may be of the same order of magnitude as the saving that results from using the aggregate network to solve the network design problem.

Nevertheless, several points are worth mentioning. First, the aggregation algorithm provides the analyst with one aggregate network for each link extracted from the detailed network. Thus the aggregate network that best suits the analysis purposes can be selected. Second, it is not likely that only a single network design problem will be solved; rather, a series of problems will be solved, each of which might be solved on a different aggregate network. For example, the analyst might elect to screen a large number of candidate projects by using a highly aggregated network. The most promising projects, along with others that might be of interest for nontechnical reasons, would be retained in more detailed analyses conducted on more detailed networks.

Third, with a given aggregate network, a large number of sensitivity analyses can be performed. As already indicated, it is likely that budget sen-
Activity analyses will be performed. In addition, the sensitivity of the solution to additional constraints that require selected projects to be included in (or excluded from) the optimal solution may need to be analyzed. In all of these cases the network aggregation algorithm needs to be solved only once. Thus, the CPU time for the network aggregation algorithm is best viewed as a fixed cost that may be distributed over a large number of analyses.

Finally, note that the CPU time involved in solving the network design problem decreases significantly as the result of extracting six links from the network in both the five- and six-project experiments. The CPU times for the cases of 6, 12, and 18 extracted links are comparable. A slight decrease in CPU time is experienced as a result of extracting 24 links, and a significant decrease is found when 30 links are extracted. This result, combined with the results outlined in the section Results for the Aggregation Model, which describes the accuracy of the results at various levels of aggregation, clearly suggests that there is an important trade-off to be made between decreased computation costs (and greater network aggregation) on one hand and improved solution accuracy on the other hand.

In the sample problems previously discussed, it appears that desirable aggregation levels would correspond to either the extraction of 6 links (resulting in a moderate decrease in computer time and a high level of accuracy) or the extraction of 30 links (resulting in a large decrease in computer time at the expense of decreased solution accuracy). Intermediate levels of aggregation appear to result in relatively large solution errors without large compensations in terms of solution times. An interesting area of future research would be to determine whether or not the network aggregation algorithm results in such identifiable choices between aggregation and solution accuracy in other network design problems, and more generally, in other network problems.

**Sources of Discrepancy Between Aggregate Network and Detailed Network Results**

The results presented in the previous sections on the application of the proposed network aggregation algorithm to the network design problem are generally promising. In no case is the difference in the improvement in vehicle hours between the solutions on the detailed and the aggregate networks greater than 7.95 percent. Also, the two solutions differ by at most one candidate link in all cases. Nevertheless, there are several differences that warrant further explanation. As indicated in the following paragraphs, the test case selected is likely to exaggerate the extent of the differences that are likely to result in a more realistic planning context.

Two characteristics of the test problem will tend to result in an overestimation of the errors that result from using the network aggregation scheme. First, the Sioux Falls network being used is already a highly aggregate representation of the actual road network. This is evident when the range in equilibrium flows on the individual network under the do-nothing option is examined; i.e., the maximum link flow is less than 4 times the minimum link flow. The average link flow is 12,989 vehicles and the maximum flow is 24,901 vehicles. The flow in the 30th link extracted from the network is 9,839 vehicles, or almost 40 percent of the maximum link flow. A real network is likely to exhibit a much greater range in equilibrium flows. If the extracted links truly carry an insignificant level of flow compared with the flow on the maximum flow link, the solutions to design problems on the aggregate networks are likely to be much better than they were in the test problem in which the flow levels on the extracted links were actually quite large and significant.

Second, the number of candidate links in the design problem was large relative to the total number of links in the detailed network. There are 6 two-way candidate links on a network with only 38 links. In the aggregate networks the situation is even more dramatic. When 30 (one-way) links are extracted, 26 percent of the links are being considered as candidate links. Thus the changes under consideration for the network are quite radical when compared with more realistic situations in which only 1 or 2 percent of the links are likely to be considered candidate links. Again, if the ratio of the number of candidate links to the number of links in the detailed network is small, the solution to the design problem on an aggregate network is more likely to replicate the solution on the detailed network than was found in the test problem, in which almost 16 percent of the links in the detailed network were candidate links.

In summary, the test network chosen for study is already a highly aggregate network that exhibits a relatively small range in equilibrium link flows. Also, the number of candidate links is already large relative to the total number of links in the network. It is expected that, if a more realistic detailed network is used, the solution to the network design problem using an aggregate network will more closely approximate the solution using the detailed network than was found in the small test network.

Finally, note that the aggregation process extracts links sequentially, thereby propagating computational errors and accumulating them in the final aggregate network. The resulting O-D trip matrix carries these errors to the decision-making model--in this case the network design model. Had a simultaneous extraction process been developed, this source of error would have been eliminated. To date, however, a simultaneous extraction process that circumvents the multiple counting danger has not been implemented.

**Conclusions and Recommendations for Future Work**

A network extraction algorithm for the network aggregation problem has been presented. The algorithm is based on the extraction of those links in a detailed network whose equilibrium link flows are less...
than a user-specified fraction of the maximum equilibrium link flow. The algorithm is sufficiently flexible to allow the analyst to force certain links out of the detailed network or to retain particular links in the resulting aggregate network. Links are sequentially extracted from the network and, after each extraction, a modified O-D matrix is derived. The revision in the O-D trip matrix preserves the level of equilibrium flow in the nonextracted links. By extracting links sequentially, the algorithm provides the analyst with multiple aggregate networks— one after each link extraction.

The network aggregation algorithm was tested by examining the performance of a network design algorithm (4) on both a detailed network and five aggregate networks derived from the detailed network. The results are quite encouraging. The maximum percentage error in the improvement in vehicle hours of travel between a solution using the detailed network and a solution using an aggregate network was 7.95 percent. In most cases the same projects were selected for implementation when using both the detailed and the aggregate networks; when the solutions differed, at most one link was misselected when using the aggregate network. As suggested in the previous section, it is anticipated that even better results will occur when the algorithm is applied to networks that are larger and more realistic than is the 76-link, 24-node test network presented here.

Several promising areas for future research are suggested by this study. First, links are extracted from the network in increasing order of the ratio of the equilibrium flow in the link to be extracted to the maximum equilibrium link flow. Other criteria should also be investigated. For example, in certain contexts it may be desirable to extract links based on the ratio of the equilibrium flow in the link to the capacity of the link. Alternatively, hybrid criteria might be developed. For example, in the traditional formulation of the network design problem, the objective function is

\[ \text{Minimize } Z = \sum_{\text{all links}} x_i^L C_i(x_i^L) \]

which may be rewritten as

\[ \text{Minimize } Z = \sum_{\text{links in aggregate network}} x_i^L C_i(x_i^L) + \sum_{\text{deleted links}} x_i^L C_i(x_i^L) \]

In solving a network design problem on an aggregate network, it is hoped that changes in the network caused by the actions taken will not significantly affect the second term of the objective function and that it may, therefore, be treated as a constant and omitted from the calculations. This suggests that the rate of change in the objective function from a change in the flow on link \( i \) can be computed, and that links for which changes in the flow will only marginally change the objective function can be deleted. Specifically, the rate of change in the objective function because of a change in the flow of link (which is denoted \( W_i \)) is

\[ \frac{\partial Z}{\partial x_i} = W_i = C_i(x_i^L) + x_i^L C_i'(x_i^L) \]

where \( C_i(x_i) \) is the derivative of \( C_i(x) \) evaluated at \( x = x_i^L \). A hybrid strategy would be to compute \( W_i \) for all links and to delete those links for which \( W_i \) is less than a \( \max\{W_i\} \).

Second, the O-D trip matrix that is derived after each link is extracted is not unique because it is based on the O-D specific flows in the extracted link, which are not unique. The effect of other O-D matrices on the aggregate networks and on the uses to which those aggregate networks are put is worthy of additional research.

Third, to avoid multiple counting problems, a sequential link extraction procedure has been implemented. Research should be devoted to the development of a simultaneous link extraction procedure. Such a procedure would be faster than the sequential procedure that has been used and would be less prone to accumulating and propagating round-off errors from one aggregate network to the next.

Fourth, based on the network design experiments, it was suspected that the quality of the network design solution that uses an aggregate network is related to the degree of network aggregation in the neighborhood of the candidate links and to the ratio of the available budget to the budget required to implement all candidate links. Additional research should explore these relationships.

Finally, the algorithm should be tested on networks that have a limited number of origins and destinations to determine when the increase in the size of the O-D matrix that results from the extraction algorithm increases the computation time more than the time is reduced because of the deletion of links. Recall that this did not occur in the network used in the set of experiments because all nodes were origins and destinations. If this does occur, it might limit the usefulness of the proposed approach to cases in which the increase in the size of the O-D matrix can be predicted to be small.

ACKNOWLEDGMENT

We would like to acknowledge Houssain Poorzahedy, whose ideas and assistance contributed significantly to this research. We would also like to acknowledge Bedayat Aashtiani, who helped secure a computer account for the component of the research conducted in Iran, as well as the computer staff of the Isfahan University of Technology in Iran for their technical assistance. Funds for this research were partly provided by a National Science Foundation grant; this support is also greatly appreciated.

REFERENCES


Publication of this paper sponsored by Committee on Passenger Travel Demand Forecasting.