Chapter 5

Transportation Network Development and Algorithm

Key words: network development, network properties, shortest path algorithms.

5.1 Transportation Networks

In order to predict how the demand for mobility will be manifested in space and time, it is necessary to represent the transportation infrastructure in some formal, simple but sufficiently detailed way. The approach adopted almost universally is to represent the infrastructure by a set of links and a set of nodes. The relationship between the links and the nodes, referred to as the network topology, can be specified by a link-node incidence matrix. This is one of the few areas of transportation modeling where there is a near unanimity of approach. While attempts to develop continuum models are periodically made, these have yet to replace network-based models, principally because of their inability to represent the properties of a transportation infrastructure in sufficient detail. This chapter sets out the principles of the network-based representation of transportation systems.

5.1.1 Network Terminology

A transport network may be formally represented as a set of links and a set of nodes. A link connects two nodes and a node connects two or more links. Links may be either directed, in which case they specify the direction of movement, or undirected. Two links are said to be parallel if they connect the same pair of nodes in the same direction. A loop is a link with the same node at either end. Some of the algorithms described later require the exclusion of parallel links and loops. All links referred to in later chapters are directed.

Links may have various characteristics. In the context of transportation network analysis, the following are some of the characteristics of interest:

- link length (in metres or perhaps in average number of vehicles);
- link cost (sometimes travel time but more generally a linear combination of time and distance); and
- Link capacity (maximum flow).

A link may be regarded as a conduit for flow whose units of measurement will depend on the application (for example, vehicles per hour or passengers per hour). Flow consists of one or
more commodities. While the commodities may refer to different kinds of goods and services, in this context they often refer to other aspects of a unit of flow, like its origin or its destination (sometimes referred to as the source and sink respectively).

If there is only one origin and one destination, and no other relevant classification, flow is said to be single commodity. In addition to commodities, user classes are sometimes identified to distinguish between units of flow with different forms of travel behavior.

A movement in a transportation network corresponds to a flow with a distinct origin and destination. Origins and destinations may correspond to specific buildings, like a house or an office, or to zones, depending on the level of aggregation. From the perspective of a transportation network, an origin or destination is represented by a kind of node, referred to as a centroid. Each centroid is connected to one or more internal nodes by a kind of link referred to as a centroid connector (or just a connector). While links tend to correspond to identifiable pieces of transport infrastructure (like a section of road or railway), centroid connectors are artifacts, especially when the respective centroid corresponds to a zone with, in reality, multiple entrances and exits.

Figure 5.1 provides an example of a transportation network with one origin centroid, two destination centroids, five links, four internal nodes, three connectors and 5 paths.

![Figure 5.1: A Sample Network](image)

Other useful terms with mostly intuitive interpretations (Figure 5.2) are: a path, which is a sequence of distinct nodes connected in one direction by links; a cycle, which is a path
connected to itself at the ends; a tree, which is a network where every node is visited once and only once; and a cutset, which is a minimal collection of links whose removal from the network would cut the network in two with no links between the two resulting sub-networks.

![Path, Cycle, Tree Diagrams]

**Figure 5.2: Example of a path, tree, and cycle**

### 5.1.2 Transportation Network Types

Some network topologies are commonly encountered in transportation systems. One such is the linear network, characterizing perhaps an expressway, an arterial road or a railway line. There may be many origins and destinations but no choice of path. Another is the grid network, representing perhaps an urban area consisting of blocks as is common in the USA and Japan. There may be many origins and destinations as well as many alternative routes.

Junctions may be represented in different ways, depending on the context. Take the junction portrayed in Figure 5.3 as an example. Where little detail is required, a junction may be represented as shown in Figure 5.4(a) by a single node and each approach by two links (one in and one out). However, in this case vehicles approaching from W and making a right turn constitute a separate stream and may well in practice be signalized differently. When queuing behavior is of interest, as is the case in traffic signal control, it will be necessary to represent the streams by separate links, as shown in Figure 5.4(b). Since parallel links can cause
problems for some algorithms, specifically shortest path algorithms, it may be necessary to introduce an additional node. At some junctions particular turning movements may be excluded. Each turning movement and stream may be represented explicitly by introducing extra nodes, as shown in Figure 5.4(c).

Figure 5.3: Example Junction

Figure 5.4: Junction Representation
Trips can be made by either an individual mode, like a car or a bicycle, or a community mode, like a bus or a tram. Trips can also be made by different kinds of infrastructure, like road, rail, canal or air. Mode and infrastructure choice can be combined in the same transportation network. Each mode and kind of infrastructure has its own characteristics, which have to be captured in sufficient detail. Interchange penalties, (the costs of changing modes), as well as the walk to and from stops or stations, have to be represented appropriately.

Clearly as the level of detail required increases so does the complexity of the representation in terms of the numbers of links and nodes. The amount of network information needed to represent the road system of an urban area in sufficient detail for analysis can be enormous. In general, network design is the art of compromise; sufficient detail is required to capture the main behavioral relationships at a cost in terms of network data collection that is commensurate with the value of the exercise.

### 5.1.3 Flow and Cost Variables

The following notation is adopted for flow variables:

- $v_k = $ the flow on link $k$;
- $h_{p_{ij}} = $ the flow on path $p$ between origin $i$ and destination $j$; and
- $t_{ij} = $ the flow between origin $i$ and destination $j$.

Vector notation will be used to represent sets of the above variables. Vectors are columns of variables unless transposed.

For example,

$$v^T = [v_1, v_2, \ldots, v_K]$$

Where superscript $T$ indicates that the vector has been transposed to produce a row of variables and where $K$ is the number of links. In the case of path flows

$$h^T = [h_{111}, h_{211}, \ldots, h_{p_{11}}, h_{121}, h_{221}, \ldots, h_{p_{21}}, \ldots, h_{p_{ij}}]$$
Where \( P \) is the number of paths between each origin and destination (in general, this need not be equal for each origin-destination pair), \( i \) is the number of origins and \( j \) is the number of destinations.

When the number of origins and/or destinations exceeds one, the flow on link \( k \) is multi-commodity, in the sense described earlier. The origin-to-destination flows, referred to collectively as the trip table, may be conveniently represented in vector form as follows:

\[
t^T = [t_{11}, t_{12}, t_{13}, \ldots, t_{ij}, \ldots, t_{ij}]
\]

Where \( i \) is the number of origins and \( j \) is the number of destinations.

The following notation is adopted for cost variables,

\[
C_k = \text{the unit cost of travel on link } k;
\]
\[
g_{pij} = \text{the unit cost of travel on path } p \text{ between origin } i \text{ and destination } j; \text{ and}
\]
\[
Z_{ij} = \text{the unit cost of a trip from origin } i \text{ to destination } j.
\]

As with path flows, the origin and destination subscripts are usually omitted from path cost variables and the paths numbered distinctly and consecutively. In vector and matrix notation:

\[
c^T = [c_1, c_2, \ldots, c_K]
\]
\[
g^T = [g_1, g_2, \ldots, g_p]
\]
\[
Z^T = [Z_{11}, Z_{12}, Z_{13}, \ldots, Z_{ij}, \ldots, Z_{ij}]
\]

### 5.1.4 Incidence Matrices

An incidence matrix is a table of binary or ternary variables stating the presence or absence of a relationship between network elements and other variables. The incidence matrix specifies the network topology. A useful matrix for network processing is the node-link incidence matrix with elements:

\[
ed_{nk} = \begin{cases} 
1 & \text{if node } n \text{ lies at the exit of link } k \\
-1 & \text{if node } n \text{ lies at the entrance of link } k \\
0 & \text{otherwise}
\end{cases}
\]

The node link matrix has the form:
Two incidence matrices are the link-path incidence matrix and the origin-destination-path incidence matrix.

The **link-path incidence** matrix has elements:

\[ a_{kpij} = 1 \text{ if path } p \text{ from origin } i \text{ to destination } j \text{ uses link } k, \text{ and } 0 \text{ otherwise.} \]

Or else \( a_{kp} = 1 \text{ if path } p \text{ uses link } k, \text{ and } 0 \text{ otherwise.} \)

The link path incidence matrix has the form:

\[
A = \begin{bmatrix}
    a_{11} & a_{12} & \ldots & a_{1P} \\
    a_{21} & a_{22} & \ldots & a_{2P} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{K1} & a_{K2} & \ldots & a_{KP}
\end{bmatrix}
\]

The origin-destination-path matrix has the elements:

\[ b_{ijp} = 1 \text{ if path } p \text{ connects origin } i \text{ with destination } j, \text{ and } 0 \text{ otherwise.} \]

And has the following form

\[
B = \begin{bmatrix}
    b_{111} & b_{112} & \ldots & b_{11P} \\
    b_{121} & b_{122} & \ldots & b_{12P} \\
    \vdots & \vdots & \ddots & \vdots \\
    b_{IJ1} & b_{IJ2} & \ldots & b_{IPP}
\end{bmatrix}
\]

**5.1.5 Conservation Relationship**

It is assumed that flow enters and leaves the network only by the centroids. At all internal nodes flow is conserved. This means that the flow into each internal node must equal the flow out of it, or in the case of multicommodity flows, the flow of each commodity into each internal node must equal the flow of the same commodity out. For each internal node \( n \) therefore there is a relationship of the form:

\[ e_n^T v = 0 \]
Where $e^T_n$ is row $n$ of the node-link incidence matrix defined earlier. This implies that for each node, one of the associated flows is linearly dependent on the other flows.

Conservation relationships have a number of other forms. One form frequently encountered is $v = Ah$

This asserts that the flow on any link is equal to the sum of the flows on all of the paths using that link.

Another conservation relationship encountered is the following:

$t = Bh$

This asserts that the flow between any origin and destination is equal to the sum of the flows on all the paths connecting that origin to that destination.

A further useful conservation relationship is:

$v = Pt$

Where $P$ is a matrix of link choice proportions (or, looking at the relationship stochastically, link choice probabilities).

### 5.2 Transport Network Representation

A network may be specified as a directed graph consisting of finite set of elements called nodes, pairs of which are joined by one or more arcs usually referred to as links.

#### 5.2.1 Links:

The transport network consists of four basic types of links, which are:

**Centroid connector:** Schematic representation of the local network connecting the zonal trips to the network. The location and number of centroid connectors can have a significant impact on how traffic is assigned to the network. Centroid connectors should represent, as closely as possible, the local streets within the zone and reasonable access points to collectors/arterials in the system. Figure 5.5 shows a sample centroid connector in a network.
Road links: Road links are classified on the basis of their capacity and speed-flow relationships. For an example in typical road network there will be different links of Freeways, arterial roads, etc. each having different capacities and speed limits. Figure 5.6 shows typical road links in a network.

Public Transport Links: Public transport links includes bus route, railway lines etc. within which each could be classified based on their capacity. Figure 5.7 represents a bus route with links and nodes.
5.2.2 Nodes:

The network consists of four basic types of nodes which are listed below:

**Centroid:** Points within the zone at which all the trips are assumed to start and finish. Figure 5.9 show the centroids in a typical network.

**Intersections:** Intersections are located wherever the links intersect. It represents the point at which two road links intersect. Figure 5.10 shows typical intersections in a road network.

**Dummy Links:** Attributes that characterize the network are assigned to the links. Examples of link- characteristics are length, speed, travel time, capacity, etc. No characteristics are assigned to the nodes of the model network. Specific characteristics of intersections, such as long waiting times for some exits or the prohibition to use certain turns, can be modeled by adding extra (dummy) links. Figure 5.8 shows typical network with dummy links.
Public transport terminals: These features include the facilities like bus stop, rail station etc. Figure 5.7 shows typical bus stop representation in network.
5.2.3 Link Properties

Links possess spatial and locational properties, such as length, direction and capacity, while nodes possess no dimensions. The level of detail provided about the attributes of links depends on the general resolution of the network and on the type of model used. At the very minimum the data for each link should include:

- Link length
• Link travel speed
• The capacity of the link, usually in passenger car equivalent units per hour (PCU).
• Type of road (e.g.: express road, trunk road, local street)
• Road width or number of lanes or both
• An indication of presence of bus lanes or prohibitions of use by other vehicles.
• One way or two way traffic movement.
• Type of junction

Figure 5.11 shows the methodology used for typical road network mapping.

![Road Network mapping Procedure Diagram](image-url)
5.3 Transport Network development

The basic data used in travel demand forecasting are origin–destination matrices and network data. The origin–destination data represent the demand side of quantity. The network data describe the supply side that accommodates the demand. Among the criticisms identified by researchers in making those data sets for transportation planning and demand forecasting, labor-intensiveness is considered as one of the most serious problems, especially in preparing transit network data.

During the preparation of the transit network and related data set needed, transportation planners have to prepare maps to describe study areas and actual transportation networks. The network generation requires extensive data collection and integration efforts. Furthermore, the generated networks are frequently modified to reflect changes, such as (1) changes in study area boundaries, (2) changes in zone delineation due to the land use change, (3) modified networks (link shape and node location change) for testing alternative network scenarios, and (4) link attribute changes like speed limit or capacity. In such cases, data requirements are extensive. Two typical problems in transit network preparation are described first, along with some possible problem solving activities.

5.3.1 Typical problem I

A directed graph, \(G = \{N, A\}\), consists of a set \(N\) of nodes and a set \(A\) of arcs whose elements are ordered pairs of distinct nodes. A network is a directed graph whose nodes and/or arcs have associated numerical values, typically costs, capacities, and/or supplies and demands. In everyday life, transportation networks are perhaps the most common and the most readily identifiable classes of networks among physical networks. Focusing on the transit network can identify typical problems encountered in the preparation of the transit network data.

Unlike the highway network, stops in a transit network are not spatially coincident with intersections (represented by nodes in GIS) of highway network and are sometimes located in the middle of links (Figure 5.12). Therefore, it is impossible to use the vector topology database provided by the highway GIS network database, and some steps are required to transform the highway database to the data type needed for transit network data generation and modeling.
5.3.2 Typical problem II

Besides node and link data, a complete transit network requires additional input for transit demand modeling. This is called line data. It is composed of an ordered set of stops on top of link data. The most prevalent problem encountered during the network preparation is the lack of consistency between link and line data. Figure 5.13 illustrates such a situation where link 102–111 exists in the line data set but is absent from the link data set.

Figure 5.12: Highway network vs. transit network.

Figure 5.13: Lack of Consistency.
5.3.3 Two Types of Stop Data

Transit stops can be represented either by a point or by a node. Both are primitives of geographic elements representing reality. Nodes are assigned to intersections where two or more lines meet, whereas points can be positioned along the line. If a stop position is fixed and connected to arcs, as in the case of a subway, it is preferable to represent it with a node connecting to incoming and outgoing arcs (Figure 5.14). But bus stops, which might be moved frequently, are better represented by points, and dynamic segmentation and other GIS capabilities should be used to derive link and line data sets.

**Figure 5.14: Transit stop data types.**

5.3.4 Node-Based vs. Point-Based Transit Networks

A work done by Choi and Jang in 1997 proposed node-based transit network development procedures in which all stops are assumed to be located at physical highway intersections and are represented as nodes. But in actual transit networks, stops are normally represented as points along arcs rather than nodes. The node-based transit network development procedure is rather straightforward to implement. The data preparation relationship in node-based transit network development procedures is shown in Figure 5.15. In figure 5.15 the terminology used is consistent with the use of ESRI’s ARC/INFO GIS application. The RAT, AAT, NAT, SEC, and AML acronyms stand for route attribute table, arc attribute table, node attribute
Table, section table, and arc macro language, respectively. On the other hand, the point-based procedure requires more complex steps to derive link and line data. Table 5.1 summarizes the major differences between the two procedures.

![Figure 5.15: Node-based transit network development](image)

**Table 5.1: Node-based vs. point-based transit network development**

<table>
<thead>
<tr>
<th>Aspects</th>
<th>Node-based</th>
<th>Point-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop is represented by</td>
<td>Node</td>
<td>Stop</td>
</tr>
<tr>
<td>Line data are</td>
<td>Series of nodes</td>
<td>Ordered set of stops</td>
</tr>
<tr>
<td>Link length</td>
<td>Arc length (embedded in AAT)</td>
<td>Need to be calculated</td>
</tr>
<tr>
<td>Application</td>
<td>Not practicable</td>
<td>Practicable</td>
</tr>
</tbody>
</table>

5.3.5 Data preparation

Before proceeding with the description of the main algorithm for generating link and line data, processes involved in stop positioning, stop–arc relationship building, and link distance calculation are introduced in this section.

5.3.5.1 Positioning Stops

When the stop is positioned during stop data preparation, care must be taken to maintain the relationship between arcs and stops. As is the case in Figure 5.16, the stop position should be...
inserted more closely to the link to which it is eventually related. For example, stop 2 must be positioned closer to link A than to link B to represent that stop 2 is located along the link A. This kind of relationship is needed to build the route system and to generate transit network.

![Diagram of stop positioning along arcs](image)

**Figure 5.16: Positioning stops along arcs.**

### 5.3.5.2 Building the relationship between stops and arcs

The relationship between stops and arcs does not exist at first (after the stop data preparation). However, the relationship can be constructed easily by GIS functions and Figure 5.17 show this procedure. As shown in the figure, stops represented as points have no relational information about arcs nearby. However, by computing the shortest distance between a node and an arc, we can identify the relationship, that is, which stops are laying on which arc.

### 5.3.5.3 Calculation of Stop-to-Stop Distance

Link data information in a transit network consists of a pair of stops, link distance, and other related attribute data of the link. In a node-based transit network development procedure, the distances between stops are embedded in the topological information of the arc-node relational database.
Figure 5.17: Relationship between stops and arcs

But in a point-based procedure, there is no direct link distance (stop-to-stop distance along the route) information available. Therefore, a separate procedure/algorithmmust be developed to calculate transit link distances.

As shown in Figure 5.18, if either stop is located in the middle of an arc, the algorithm has to calculate the partial length of the arc where the stop is located. Four steps are required to achieve this. First, find the perpendicular points from the stops to the nearest arc segments using GIS’ nearest arc finding capability from a node representing stop using spatial analysis. Second, calculate the distance from the beginning stop of the link to the first node encountered along the route (using the dynamic segmentation). Third, calculate the distance from the ending stop of the link to the first node encountered back through the route using the arc–node and section tables, as in the second step. Fourth, sum the results from second and third steps and all intervening arc lengths as illustrated in figure 5.19. If the beginning stop is located exactly at the node position, the second step is not needed. The same is true of the third step.
5.3.5.4 Algorithm for Calculating Transit Link Distances

Figure 5.20 shows the schematic flow of the procedure algorithm, whereas Figure 5.21 depicts the flow chart of the proposed algorithm. The upper part of the algorithm in Figure 5.20 depicts the stop positioning on top of the existing GIS layer, building a route system with stops positioned using dynamic segmentation and nearest arc finding process using GIS spatial analysis. The lower part of Figure 5.20 describes the link and line data derivation using the prescribed data set in the upper part of the algorithm.
Figure 5.20. Overall Flow of Algorithm for Transit Link Distance Calculation.

The algorithm (figure 5.21) converts route data (generated by the GIS dynamic segmentation model) to point-based links and lines using the relationship between stops and arcs described earlier. The algorithm converts one route at a time and examines all sections in that route. At the section examining stage, it searches for all stops that have the same arc number. If any stops are found, they are checked to see if the route passes through them. Once the stop is found to be on the route, it is added to the temporary stop buffer with its cumulative length from the beginning point of the route. After examining all sections, stops stored in the temporary buffer are sorted by cumulative length, and link and line data are extracted. After a route system is processed in this fashion, the algorithm goes to the next route and so on until all the routes are processed.
### 5.4 Shortest Path Algorithm

A crucial step in many network flow programming methods is the finding of shortest (or, more generally, least cost) paths between any pair of nodes or centroids. Two shortest path algorithms are described in this section, each with a conceptually distinctive approach. Both the algorithms presented here operate on a connections matrix and a back-node matrix. Let $N$ be the number of nodes and $Z$ the number of centroids. The connections matrix is a matrix...
with \( N + Z \) rows and \( N + Z \) columns whose elements are \( c_{mn} \). It is initialized as follows:

Where there is a link or centroid connector from node or centroid \( m \) to node or centroid \( n \), \( c_{mn} \) is assigned its cost; where no such link or centroid connector exists, \( c_{mn} \) is assigned a large value.

During processing, connections are made and, if new or better than the preceding connection, entered in the connections matrix. The algorithms differ with respect to the way in which the connections are sequenced. After processing, the connections matrix gives the minimum cost of travelling from any node or centroid to any other node or centroid. When \( c_{mn} \) retains its preassigned large value, node or centroid \( n \) is not reachable from node or centroid \( m \).

The back-node matrix has the same size as the connections matrix. After processing, element \( q_{mn} \) of this matrix gives the penultimate node on the path from node or centroid \( m \) to node or centroid \( n \). The back-node matrix enables the optimal path to be traced. Suppose that \( q_{mn} = k \), then the optimal path from \( m \) to \( n \) has as its penultimate node \( k \), and the node preceding \( k \) is given by \( q_{mk} \) etc...

A method that is simple to program is that due to Floyd and Warshall developed in 1962. The Floyd- Warshall algorithm can be stated as follows, using a BASIC-like language.

**Floyd—Warshall shortest path algorithm**

**Step 1** (initialisation of the connections and backnode matrices)

\[
\begin{align*}
q_{mn} &\leftarrow m \\
&\text{if a link joins } m \text{ to } n \text{ then } c_{mn} \leftarrow \text{link cost} \\
&\text{otherwise } c_{mn} \leftarrow \infty
\end{align*}
\]

**Step 2** (obtain least cost paths)

\[
\begin{align*}
&\text{for all nodes } k \\
&\text{for all nodes or centroids } m \text{ not equal to } k \\
&\text{for all nodes or centroids } n \text{ not equal to } k \text{ or } m \\
&\text{if } c_{mk} + c_{kn} < c_{mn} \text{ then } c_{mn} \leftarrow c_{mk} + c_{kn} \text{ and } q_{mn} \leftarrow q_{kn}
\end{align*}
\]

As the iterations progress, paths are formed, and when parallel paths emerge, the lower of the two costs is retained. The algorithm works because all paths and sub-paths are generated.

To illustrate the algorithm, consider figure 5.2. Suppose \( A \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow B \) is the optimal path from \( A \) to \( B \). At some stage during the first outer loop (when \( k = 1 \)), the
connection A →2 is made. This is followed in the second outer loop (when k = 2) by the optimal connection A →4 via nodes 1 and 2, and in the third outer loop (when k = 3) by the connection 4→B. Finally, in the fourth outer loop (when k = 4) the optimal connection A →B is established. If the nodes were numbered differently, a different series of connections would be made but the end result would be the optimal connection A →B.

**Figure 5.22: Sample Network**

An alternative, more efficient approach to finding least cost paths is to build a least cost path tree, as is done by the algorithm due to Dijkstra developed in 1959. Let V be the set of nodes visited so far. Dijkstra’s algorithm may be stated as follows:

**Dijkstra’s shortest path algorithm**

**Step 1** (initialisation of the connections and backnode matrices)
for all nodes or centroids m and n
\[ q_{mn} \leftarrow m \]
if a link joins m to n then \[ c_{mn} \leftarrow \text{link cost} \]
otherwise \[ c_{mn} \leftarrow 0 \]

**Step 2** (obtain least cost paths)
for all centroids m
\[ k \leftarrow m \]
set \( V \) is empty
set \( E \) is empty
repeat
add \( k \) to set \( V \)
for all \( n \) connected to \( k \)
put \( n \) in set \( E \)
if \( c_{mk} + c_{kn} < c_{mn} \)
then \[ c_{mn} \leftarrow c_{mk} + c_{kn} \] and \[ q_{mn} \leftarrow q_{kn} \]
select \( k \) in \( E \) and not in \( V \) so that \( c_{mk} \) is minimum
until \( V \) contains all nodes
Set V contains the set of visited nodes; note that every node is visited only once. Set E contains the nodes reached so far by branching, not all of which are eligible for selection. Each iteration of the innermost loop adds a branch to the tree and checks whether any node has thereby been reached with less cost. If so, the minimum cost and back-node matrices are updated.

The operation of the algorithm in relation to Figure 5.22 is illustrated in Figure 5.23 for hypothetical link costs. When the least cost path tree is two links deep, node 2 is selected for branching as it is the node reached so far (namely in set E) that is not in set V and is closest to A (the route of the tree). Node 2 is added to set V. From node 2, nodes 3 and 4 are reached and therefore added to set E. The node next selected for branching is node 4, as it is in set E, has least cumulative cost and is not in set V. Node 4 is added to the set V, which ensures that the path A→1→4 is not developed further. This path would clearly be sub-optimal, as node 4 is reached at less cost via nodes 1 and 2.

The optimality of Dijkstra's algorithm is guaranteed by the selection mechanism for k, the node for branching. Each node k is selected only once. When node k is selected, it must lie on an optimal path; if not, k must be reachable with less cost, in which case it would have been selected earlier.

---

**Figure 5.23: Operation of Dijkstra’s Algorithm**
Dijkstra's algorithm, unlike Floyd's algorithm, works only with positive costs, although in the context of transportation network analysis this is not usually a limitation.

Consider again the network in Figure 5.22 and assume that the connections matrix before processing by one of the algorithms is as given in Table 5.2. The entry 99999 indicates that there is no link from the given node/centroid to the given node/centroid. The corresponding back-node matrix before processing is as given in Table 5.3.

**Table 5.2: Connection Matrix before Processing**

<table>
<thead>
<tr>
<th>From</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99999</td>
<td>10</td>
<td>99999</td>
<td>30</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
</tr>
<tr>
<td>2</td>
<td>99999</td>
<td>99999</td>
<td>30</td>
<td>10</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
</tr>
<tr>
<td>3</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>15</td>
<td>99999</td>
</tr>
<tr>
<td>4</td>
<td>99999</td>
<td>99999</td>
<td>10</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>15</td>
</tr>
<tr>
<td>A</td>
<td>15</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
</tr>
<tr>
<td>B</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
</tr>
<tr>
<td>C</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
</tr>
</tbody>
</table>

**Table 5.3: Back Node Matrix before Processing**

<table>
<thead>
<tr>
<th>From</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>A</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>
After processing, the connections matrix is as shown in Table 5.4. It gives the least cost between any pair of zones. An entry 99999 indicates that the given destination node/centroid is not reachable from the given origin node/centroid. The corresponding back node matrix after processing, given in Table 5.5, allows the least cost paths to be traced. For example, the back-node to B on the least cost path from A to B is 3, the back-node to 3 is 4, the back node to 4 is 2, and finally the back node to 2 is 1.

**Table 5.4: Connection Matrix after Processing**

<table>
<thead>
<tr>
<th>From</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99999</td>
<td>10</td>
<td>30</td>
<td>20</td>
<td>99999</td>
<td>45</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>99999</td>
<td>99999</td>
<td>20</td>
<td>10</td>
<td>99999</td>
<td>35</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>15</td>
<td>99999</td>
</tr>
<tr>
<td>4</td>
<td>99999</td>
<td>99999</td>
<td>10</td>
<td>99999</td>
<td>99999</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td>A</td>
<td>15</td>
<td>25</td>
<td>45</td>
<td>35</td>
<td>99999</td>
<td>60</td>
<td>50</td>
</tr>
<tr>
<td>B</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
</tr>
<tr>
<td>C</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
<td>99999</td>
</tr>
</tbody>
</table>

**Table 5.5: Back Node Matrix after Processing**

<table>
<thead>
<tr>
<th>From</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>A</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

### 5.5 Traffic Assignment

The process of allocating given set of trip interchanges to the specialized transportation system is usually referred to as traffic assignment. The fundamental aim of the traffic
assignment process is to reproduce on the transportation system, the pattern of vehicular movements which would be observed when the travel demand represented by the trip matrix, or matrices, to be assigned is satisfied.

A digitized transport network in GIS is very helpful in analyzing and displaying the results of traffic assignment in transportation planning. This gives an estimate of the directional volume of persons or vehicles on each link of the network in a given time period. Comparing this volume with the capacity of the transport network forms a very important and basic analysis for identifying the transportation projects and for assessing the implications of any policy design. Figure 5.24 shows typical traffic assignment using GIS.

![Figure 5.24: Traffic Assignment](image)

5.6 Transit Network and Paths

Figure 5.25 shows typical transit network in GIS. Transit networks are similar to other highway networks in that they are graphs comprised of nodes and links that indicate allowable paths of travel, along with information on the cost of traversing each link (e.g.,
The link in the transit network represent actual transit segments (segments between two consecutive transit stops), and segments from the underlying geographic line layer that provide connectivity to the network in the form of zone-to-station access links, walking links, driving links, and transfer links. Transit Networks are used for:

- Finding the best route from place to place
- Calculating the attributes of the best paths (i.e., creating transit skims)
- Performing transit assignment of passengers to routes.
- Performing Origin-Destination matrix estimation (ODME) to estimate or update a transit OD matrix from base ridership counts.

Software like TransCAD includes the most realistic and flexible transit path finding and skimming methods. The key methods provide the multiple paths that travelers will use and give the analyst fine control over access, egress, and transfer properties. Shortest paths are routes over transportation network that have the lowest generalized cost, where the cost can be any combination of factors such as distance, time, or actual monetary cost of travel.

### 5.6.1 Transit Assignment

Transit assignment models are used to estimate the number of passengers that utilize segments in a transit network as a function of transit level of service. These models take as input a matrix of passenger flows between origins and destinations and a transit network, and produce link level and aggregate ridership statistics. Software TransCAD includes an array of sophisticated transit network assignment procedures. Figure 5.26 shows typical transit assignment procedure.

These procedures include methods that are sensitive to fares and park and ride access, as well as equilibrium assignments, which take account of the capacity of transit service and the effect of ridership on crowding, comfort, and, optionally through dwell time effects, on the travel time on the route. These methods distribute the flow between a particular origin and destination to multiple paths, based on their relative attractiveness.

The transit assignment procedures produce a table of ridership at every stop along each route in the transit network. Optional outputs include critical link analysis, boarding and alighting
counts, stop-to-stop flows, route-to-route transfers, and aggregate ridership counts. We can also save the paths utilized and visualize them directly.

Figure 5.25. Transit Networks and Paths
Figure 5.26. Transit Assignment

Exercises
1. What are transportation networks?
2. List the terminologies used in transport network representation.
3. Define and explain the variables used in transport network representation.
4. What are incident matrices?
5. Explain links and their types.
6. Explain network mapping procedure.
7. Compare different shortest path algorithms.

Assignment
1. Use any GIS Software for developing networks and perform shortest path analysis.