Program for Assigning Traffic to a Highway Network

GLENN E. BROKKE, Division of Highway Planning, U. S. Bureau of Public Roads

●FOR MANY YEARS highway engineers have been faced with the problem of estimating the volume of traffic that would use a proposed new facility. In addition, they would like to know how well this facility will continue to serve traffic in the future.

A realistic solution to this problem is not simple. All zones in an entire area have the potential of affecting any highway facility. The degree of this effect is governed by the location and design standards of the highway system as it now exists or as it may be subsequently improved. Thus, individual highway improvements can cause a realinement of traffic throughout the area.

Much research work has been done to provide a method of solving these problems. Although the future will probably bring added refinements, and even major changes in concepts, present knowledge is sufficient to warrant establishing at least an interim procedure for assigning present and future traffic.

A characteristic of traffic problems is the necessity of handling a mass of information in the relatively simple but coherent manner. To process these data in a reasonable time, it has usually been necessary to use a series of short cuts, approximations, or judgment estimates, each exacting its toll in deviation from accuracy. It now seems possible to employ electronic computers to provide a consistent solution subject only to the limitation of knowledge of the behavior of traffic.

The use of computers does involve the time consuming operation of preparing a program. Although it is desirable to allow flexibility in the program, this is not easily accomplished. Thus, to use an existing program it is necessary to have access to the machine for which the program is written, to furnish input data in precisely the correct form, to be willing to accept the logic that is incorporated in the program, and to accept the results in predetermined format.

The U. S. Bureau of Public Roads has written, tested, and used an IBM 705 program to predict the future distribution of trips. The input required is the number of existing zone-to-zone trips and the growth factor for each zone. The program logic follows the Fratar formula. The output is the estimated number of future zone-to-zone trips. Either three different modes of travel or one mode to three different future periods can be processed simultaneously. For a city the size of Washington, D. C., with 500 zones, about one-half hour of IBM 705 time is required for each iteration. Two or three iterations are usually sufficient, thereby making the cost of obtaining future zone-to-zone trips somewhat less than $600 for a city the size of Washington.

With the present and future zone-to-zone trips available, the next problem is to estimate the loading of these trips on a highway network. To do this, some method of predicting the distribution of traffic between routes is required.

Three types of diversion curves are in current use—the time-ratio curve as attached to the Bureau's "Guide for Forecasting Traffic on the Interstate System," dated October 15, 1956; the distance-ratio and speed-ratio curves as used in Detroit; and the time-and-distance differential curve as used in California.

The Bureau's time-ratio curve relates the percentage of trips using a freeway facility based on the ratio of the travel time via the freeway to the travel time via the best alternate route. The percentage of trips using the freeway varies as an S-shaped curve from 100 percent at a time ratio of 0.5 or less to 0 percent at a time ratio of 1.5 or more. If the travel time via the freeway is equal to the travel time via the alternate route (time ratio = 1.0), approximately 42 percent of the trips are assigned to the freeway.

The speed-ratio curves developed for the Detroit Area Transportation Study consist of a family of curves where the percentage of freeway use is related to speed ratio and distance ratio. These curves are also S-shaped for normal conditions, and with a speed ratio of 1.0 and a distance ratio of 1.0 assign approximately 45 percent of the trips to the freeway. Because these curves represent a three-dimensional surface with an undefined mathematical relationship, they are difficult to use in a computer application.

The California time-and-distance curve consists of a family of hyperbolas where equal time and distance on the freeway as compared to the best alternate route will assign 50 percent of the traffic to the freeway. These curves can be expressed by

$$P = 50 + 50\,(d + \tfrac{1}{2}t)\,\left[(d - \tfrac{1}{2}t)^2 + 4.5\right]^{-\frac{1}{2}} \qquad (1)$$

in which

P = percentage using the freeway;
d = distance saved, in miles, via the freeway; and
t = time saved, in minutes, via the freeway.

The development of an assignment procedure has two major difficulties, as follows:

1. Measurement of the minimum travel time between each pair of zones over the arterial network and then over the entire highway network, including the contemplated freeways.
2. Accumulation of the assigned volumes on the various segments of the highway system.

The Washington, D.C., Regional Highway Planning Committee, having recently completed an origin-destination study and having predicted the zone-to-zone movements to 1980, desired to assign this traffic to a highway network using the Bureau's time-ratio diversion curve. The labor involved in accomplishing this task for an area the size of Washington was clearly beyond the range of practicability unless an electronic computer could be used for a major portion of the work. The Bureau of Public Roads offered technical assistance.

It was established, almost immediately, that no program was available which would handle the complexities of Washington in a reasonable manner. Hence, it became necessary to develop a Washington assignment program.

At about this time it became known that the staff of the Chicago Area Transportation Study had discovered a method of determining the minimum path through a highway network. It was also known that the staff of the Detroit Area Transportation Study had carried this minimum path principle along somewhat different lines toward an assignment procedure. Both organizations were visited and each was entirely cooperative and responsive in outlining its procedures and ideas on the problem.

Because all diversion curves being considered are based on the relationship between the travel time (and distance) on the most favorable freeway route and the travel time (and distance) on the most favorable alternate route, the initial problem is to determine which of the freeway routes and which of the alternate routes are truly the most favorable.

The difficulty of this problem can be appreciated most easily if a rectangular street network is considered. To arrive at a point four blocks east and four blocks south of an origin, there are more than 40 different routes or paths that appear approximately equal to the eye as far as travel time (or distance) is concerned. However, by accurately adding the time (or distance) values on each of the segments involved for each route, the route with the least over-all travel time can be selected. This selected route is the minimum travel time route, or minimum path.

It is true, and probably apparent, that the longer the trip, the more alternate routes there are available between two points. For travel across an entire city, there may be literally thousands of alternate paths or routes and the initial problem of determining which path is the minimum appears rather difficult.

Fortunately, the staff of the Chicago Area Transportation Study were able to use a procedure developed by Moore (1). The same basic method is used in this program and essentially consists of accumulating the minimum time and path from a central point to an ever-increasing circle of points surrounding this central point.

The value of the use of the minimum path principle can hardly be overestimated in the solution of the assignment problem. The distance and travel time on each segment of the highway network are determined and fed into the computer. These initial time and distance measurements are required for any of the methods being used and are neither easier nor more difficult to obtain for the minimum path procedure. Once the travel times are in the computer, however, very substantial advantages begin to accrue.

The most obvious advantage is the saving in man-hours. On the optimistic basis that the best route of travel between a pair of zones can be located and measured in three minutes by manual methods, approximately two man-years of labor would be required to find the travel time via the arterial network and via the freeway network for the 40,000 zone-to-zone volumes occurring in the Washington area. The computer can absorb all of this manual work at the rate of about two computer-hours being equivalent to one man-year of manual computation.

A second advantage is the increase in accuracy and consistency. A manual determination of the best route has two sources of frequent errors. The routing selected as the minimum path may actually be longer than some other path. Secondly, an error may be made in adding the time intervals that make up the selected path. The minimum path program, however, tests all possible paths, selects the minimum and adds the time intervals unerringly.

A third advantage, somewhat more obscure, is the ability of the computer to take additional factors into account. For example, the computer can be rather easily instructed to test a routing and insert a turn penalty whenever a right- or left-hand turn occurs. To add this or a similar complication into a manual procedure would be entirely impractical.

The Washington traffic assignment program is in reality a library of programs that can be called upon in any desired order through the use of a master control program. To use this library of programs certain conventions must be observed, and to understand these conventions a few definitions are required, as follows:

Node—A node is any specific point on the highway system that is needed for identification purposes. Primarily, nodes are used to designate zone centroids and highway intersections. They are identified by number.

Link—The portion of the highway system between two nodes is a link. To avoid needless complication only the "through" or more important highways are identified by actual location. Links are identified by the two node numbers which terminate the link.

Route—A group of connecting links between a pair of zones is the route of travel between these zones. If a particular route has a shorter travel time than any other route, it is called the minimum time path. If distance instead of time were the criterion, a minimum distance path could be similarly described.

Tree—All minimum path routes from one particular node to all other nodes in the system constitute the tree for that particular node. In practice, trees need to be built only for the zone centroid nodes.

PROGRAM CONVENTIONS AND LIMITATIONS

The following program conventions and limitations are observed:

1. No more than four links may meet at any node. To accommodate five or more links which would otherwise intersect at a single node, it is necessary to separate the one node into two (or more) nodes with zero time and distance between them.

2. No node may be numbered more than 4,000.

3. The node numbers must be arranged in sequence in four separate groups in the following order:

Group A—Zone centroids starting with number 1.
Group B—Four-way arterial nodes.
Group C—Two- or 3-way arterial nodes.
Group D—Freeway nodes.

4. From each 4-way node there must be at least one link to a numerically larger node number.

5. The zone-to-zone trip cards must be in major sort by the first zone and in minor sort by the second zone before being placed on tape.

6. To be able to insert a turn penalty for right and left turns, it is necessary to designate each link as positive or negative. Movements between links of the same sign involve no turn. To accommodate diagonal or curving streets a flag position is also available to change signs as needed.

The Washington traffic assignment program library consists of the following individual programs:

Program No.	Title
0	Master control.
1	Build trees.
2	Load arterial network (all or nothing).
3	Load entire network (time-ratio curve).
4	Sum vehicle-miles and vehicle-hours.
5	Convert link data from decimal to binary.
6	Make freeway corrections to link data.
7	Convert trip volumes from decimal to binary.
8	Correct trip data.
9	Prepare time-ratio diversion table.

These programs are on tape and can be called in any desired order through the master control program.

INPUT DATA REQUIREMENTS

Zone-to-Zone Trips

Each of the zone-to-zone trip volumes must be represented by a trip card identifying the two zones (or stations) involved and the number of trips between them. Zero volumes need not be represented by a card except that each zone other than the last one must be represented by at least one card (zero volume if necessary) when arranged in major sort by the first identifying zone number. If not already accomplished in a previous stage, the zones and stations must be renumbered to form an unbroken sequence starting with number 1. When placed on tape the zone-to-zone cards must be in major sort by the first identifying zone number and in minor sort by the second identifying zone number.

Highway Link Data

The highway network must be described in a manner that the computer can understand. This is done by listing each link of the highway network on a coding sheet. The listing consists of the two identifying nodes, together with the sign, a flag if needed, the travel time in minutes and hundredths, and the distance in miles and hundredths. These data are then key punched with one link to a card. The cards are then duplicated, reversing the two identifying nodes, so that in the final deck each link is actually listed on two cards. The data on the cards are then transferred to magnetic tape.

PROGRAM OPERATION

Program 0—Master Control

Program 0 sets the program for the specific area in which traffic is being assigned and permits the choice of any of the programs included in the program library. As input it requires the number of zone centroids, the number of 4-way intersections, the number of 2- or 3-way intersections, the number of freeway intersections, and the amount of turn penalty in minutes and hundredths.

In addition it has been necessary to scale the time and distance values into units which will economically use the memory availability of the computer. Therefore, the maximum travel time on any link plus the turn penalty is equated to 63. The time values on all other links are converted to 63d's of this maximum travel-time link. For the same reason the maximum distance link is likewise equated to 63 and all distances converted to 63d's of this maximum distance link. It should be noted that the maximum travel-time link and the maximum distance link need not be the same link.

The maximum travel-time link and the maximum distance link are also necessary inputs to the master control program.

All of the other programs are then set with the specific characteristics entered through this master control program.

Program 5—Convert Link Data from Decimal to Binary

In program 5 the computer edits the link data for impossible codes, scales the time and distance values to appropriate units, converts all data from decimal into binary, and pecks the information to fit exactly into a block of computer memory. The output is a binary coded tape containing this large block of information.

Program 6—Make Freeway Corrections to Link Data

The most difficult problem in determining the freeway route was to arrive at some method which would compute a minimum freeway path even though it was longer than an arterial street path. This was necessary because some diversion to a freeway exists even if the time ratio is more than 1.0.

To retain the advantage of the minimum path method and still obtain a freeway time longer than arterial time, it was decided to temporarily halve the time on the freeway links. Once the tree has been established, the time values are corrected.

By this program the previous arterial links are modified as needed by the addition of the freeway nodes, the freeway links are inserted in the system with their time values cut in half, the information is converted to binary and packed into a block of memory. The memory is then written out on tape in binary code.

Program 7—Convert Trip Volumes from Decimal to Binary

The tape containing the zone-to-zone trip volumes is edited in Program 7. The numbers are converted from decimal into binary and all of the trips from the first listed (or origin) zone to all other zones are packed into one record block, which is written out on tape. There must be a trip record block for each zone except for the last (highest numbered) zone.

Program 8—Correct Trip Data

If in the process of editing or through subsequent checking it is found that some of the zone-to-zone trips are in error, the values may be corrected by Program 8 without rerunning the entire program.

Program 9—Prepare Time-Ratio Diversion Table

Program 9 builds the diversion curve table for converting time ratio to percent diversion. At present the traffic diversion curve attached to a BPR circular memorandum to division engineers (dated October 15, 1956) is incorporated in the program.

Program 1—Build Trees

Program 1 determines the minimum path from each zone to all other nodes in the highway network. If only the arterial links are used, the program builds arterial trees. If the freeway links are also included, the program builds freeway trees. Thus, the previously prepared link data are the input for this program.

The program then instructs the computer to set aside a block of memory for the tree, with each memory word of the block corresponding to an actual node on the highway system. The memory words in the block are in the same sequence as the node numbers. Thus, the position of the memory word identifies the node number.

In addition each word in the tree memory block will contain two major items of information, as follows:

1. The preceding node through which the route has passed in building the tree (back node).
2. The total elapsed time from the tree centroid to the node represented by this memory word.

Each memory word is initially set to the largest possible value. The computer then starts building the tree from zone centroid 1 in the following manner:

1. Because the tree is being built from node 1, the first step is to set the back node and the elapsed time in memory word No. 1 to zero. At the same time this node is listed in an elapsed time sequence table in the zero time slot.
2. The computer then takes the miminum entry in the elapsed time table, erases this entry, and from the link memory block, finds all links that emanate from this node, which can be called node A.
3. At the end of each of these links there is a second node, which can be called node B. The link time from node A to node B, plus a turn penalty if required, is added to the total elapsed time at node A to give the total elapsed time from the tree centroid to node B. The machine compares the computed elapsed time at node B with the previously established elapsed time stored in the word represented by node B in the tree memory. If the new time is less than the stored time, it replaces the stored time and node A replaces the previously stored back node. At the same time this node B is stored in the elapsed time sequence table in the appropriate time slot. If, however, the new time is equal to or greater than the previously stored elapsed time, the route is not a minimum path and the computer discards this value.
4. When all of the links emanating from node A have been completed in this manner, the computer again selects the minimum entry in the elapsed time sequence table and repeats the process.
5. When all values in the elapsed time sequence table have been used, the tree from zone centroid 1 has been completed and the tree memory is written out on tape.
6. The computer then proceeds to zone 2 and builds the tree from this zone in exactly the same manner.
7. When trees have been built from all zone centroids, the arterial tree routine has been completed. For the Washington, D.C., area, about 450 trees will be needed.

Freeway Tree Routine.—The program has been written so that any freeway pattern may be superimposed on the arterial street network without destroying the arterial trees already completed.

The freeway trees are built in exactly the same manner as the arterial trees except that the freeway links as well as the arterial links are included in the input data.

Program 3—Load Entire Network (Time-Ratio Curve)

The previously completed freeway and arterial tree tapes and the zone-to-zone trip tape become input for Program 3. In addition, the relationship between time ratio and percent diversion is placed in the computer memory. The program then performs the following operations:

1. The arterial tree for node 1 (also zone centroid 1) is read into a block of memory.
2. The freeway tree for node 1 is read into a separate block of memory.
3. The zone-to-zone trips from zone 1 are read into a third block of memory.
4. The trips between the first pair of zones initiates the following action:

(a) The destination zone of the trip becomes the first entry in the arterial route.

(b) From the arterial tree, the back node of the destination zones becomes the second entry in the arterial route.

(c) The back node of the second entry becomes the third entry for the route, and so on, until the route reaches zone centroid 1.

(d) The freeway route is established in same manner with the corrections to travel time on the freeway links being made.

(e) The arterial route is compared with the freeway route:

(1) If the routes are identical, all of the trips are accumulated on the arterial routing in a block of memory where each word represents a corresponding highway link.

(2) If the routes are different, the two points of choice are determined.

(f) The travel time via the freeway and via the arterial system between points of choice is computed and converted to time ratio and then to percent diversion.

(g) The freeway traffic is accumulated in memory via the freeway route and the arterial traffic is accumulated in memory via the arterial route.

(h) At all 4-way intersections two of the turning movements are recorded separately in a turn table so that in the final analysis all turning movements are available.

5. The remaining trips from zone 1 are handled in the same manner, after which the trees and trips from zone 2 replace those of zone 1 and the process is repeated.

6. This process is continued until all zone-to-zone trips have been processed, at which time the accumulated volumes are written on tape in decimal form. The decimal tape is printed on peripheral equipment. The printed output is the traffic load on all segments of the entire network, including all turning movements.

Program 2—Load Arterial Network (All or Nothing)

The library also includes a program for loading all trips on the shortest route. This is accomplished by reading only the arterial trees into memory and loading all trips on the routing established by these trees.

Program 4—Sum Vehicle-Miles and Vehicle-Hours

The vehicle-miles and vehicle-hours on the freeway network on the arterial system, and on the local system, are then computed and printed.

COMPUTER RUNNING TIME

The entire program library has been completed and a major portion tested on the Virginia portion of the Washington metropolitan area. This area consists of 102 zones, 543 nodes, and 3,488 zone-to-zone movements. The time required to run through the various programs in the library was as follows:

Program	Units	Computer time (sec)	Rate
Convert links to binary	543 nodes	25	¾ min per 1,000 nodes
Convert volumes to binary	3,488 z-z cards	60	3 min per 10,000 z-z cards
Build trees	102 zones 543 nodes	280	8½ min per 100 zones per 1,000 nodes
Load network (time ratio)	3,488 z-z cards	275	13 min per 10,000 z-z cards
Load network (all or nothing)	3,488 z-z cards	97	4⅔ min per 10,000 z-z cards $\times\left(\frac{\text{No. of nodes}}{500}\right)^{1/4}$
Compute vehicle-miles and vehicle-hours[1]	543 nodes	35	1 min per 1,000 nodes

[1] Includes conversion of link volumes and turning movements from binary back to decimal, written out on tape in an appropriate format.

Using these rates it is possible to estimate the computer time required for one complete assignment for any area. For example, it is expected that the Washington, D. C., metropolitan area will consist of about 450 zones, 3,100 nodes, and 40,000 zone-to-zone trip cards. The estimated computer time for this area would be as follows:

Program	Units	Rate	Total Time
Convert arterial links to binary	3,100 nodes	¾ min per 1,000 nodes	3 min
Convert freeway links to binary	3,100 nodes	¾ min per 1,000 nodes	3 min
Convert volumes to binary	40,000 cards	3 min per 10,000 cards	12 min
Build arterial trees	450 zones 3,100 nodes	8½ min /100/1,000	120 min
Build freeway trees	450 zones 3,100 nodes	8½ min /100/1,000	120 min
Load network (time ratio)	40,000 cards	13 min per 10,000 $\times\left(\frac{\text{No. of nodes}}{500}\right)^{1/4}$	90 min
Compute veh-mi and veh-hr	3,000 nodes	1 min per 1,000 nodes	3 min
Total			5 hr 51 min
Set up and control time			10 min
Contingency time	(10%)		34 min
Grand total			6 hr 35 min

An IBM 704 computer with 32,000 words of memory will cost from $350 to $400 per hour. Thus, the machine cost for one assignment for the Washington area will be about $2,200 to $2,600.

ACCURACY OF ASSIGNMENT PROGRAM

The accuracy of the assignment program rests basically with the accuracy of the assumption that traffic divides between routes in accordance with the time-ratio diversion curve. In any particular city, the accuracy of this assumption can be checked by assigning present trips to the existing highway system and checking against current traffic counts. If better criteria are subsequently developed which improve the distribution of traffic among routes, they will be incorporated into the program.

The program as written has considerable flexibility. Changes in the extent or location of the proposed freeways can be tested by merely altering the freeway network and rerunning the program. If any of the proposed highway segments are loaded beyond capacity, the travel times on these sections can be adjusted and the program rerun until there is a balance between capacity and travel time on each segment.

If directional zone-to-zone trips are available, the program can give directional assignments. Thus, traffic on one-way streets or ramps can be directly computed.

It is likely that the future use of the program will develop additional subroutines which will be useful in designing highways. By way of illustration, consider the case of a ramp connection immediately before an interchange between freeways. If the predominant flow of traffic from the ramp turns left at the freeway interchange, it may be advantageous to bring the ramp into the freeway from the left to avoid the confusion of weaving this traffic across the freeway. By suitable instructions, the computer can develop these or similar data.

The program is written for an IBM 704 computer with 32,000 words of memory. If there are less than 900 nodes in the highway system, an 8,000-word memory will be sufficient. In addition to memory capacity, it is essential that the computer have extremely fast access time to all memory positions. This consideration, at least for the present, precludes the use of computers which rely on a magnetic drum memory.

ACKNOWLEDGMENTS

In addition to J. Douglas Carroll, Jr., of the Chicago Area Transportation Study, and A. J. Mayer of the Detroit Area Transportation Study, the preparation of this program owes much to the efforts of W. F. Boardman of the Washington Area Transportation Study, W. L. Mertz of the Division of Development, U. S. Bureau of Public Roads, and, of course, to the General Electric Computer Laboratory of Tempe, Arizona, who reduced the ideas to the machine language the computer understands.

REFERENCE

1. Moore, E. F., "The Shortest Path Through a Maze." Presented at Internat. Symposium on Theory of Switching, Harvard Univ. (1957).