103. E.B. Lieberman. Simulation of Corridor Traffic--the SCOT Model. HRB, Highway Research Record 409, 1972, pp. 34-45.
104. E.B. Lieberman and D.A. Wicks. Traffic Flow Simulation Study--The SCOT Model. GASL, Washington, DC, Tech. Rept. 769, Feb. 1972.
105. D. Wicks and J. Cohen. SCOT Model, Calibration and Validation, Data Acquisition and Reduction. KLD Associates, 1974.
106. Final Design Report--Diversion of Intercity Traffic at a Single Point. Sperry Systems Management, Sept. 1974.
107. Description of Sperry Traffic Analysis and Research (STAR) Simulation Program. Sperry Systems Management, Oct. 1974.
108. SCOT Model, User's Manual and Program Documentation--Working Document. FHWA, May 1975.
109. A. Muzyka. Bus Priority Strategies and Traffic Simulation. TRB, Special Rept. 153, 1975, pp. 39-49.
110. L. McCabe, R. Ricci, and P. Coucannon. Simulation Studies for an Urban Traffic Corridor. Transportation Systems Center, Cambridge, MA, June 1975.
111. C.R. Berger and L. Shaw. Division Control of Freeway Traffic. Presented at 6th IFAC Congress, Cambridge/Boston, MA, Aug. 1975.
112. C.R. Berger and others. Single Point Diversion of Freeway Traffic. Presented at 55th Annual Meeting, TRB, Jan. 1976.
113. SCOT Model User's Manual and Program Documentation. KLD Associates, Inc., 1976.
114. P. Zove and others. Integrated Motorist Information System Feasibility and Design Study--Phase I, Feasibility Study; Vol. 1, Final Report; Volume 2, Appendices; Volume 3, Executive Summary. Sperry Systems Management, April 1977.
115. C. Berger and L. Shaw. Discrete Event Simulation of Freeway Traffic. Sperry Systems Management, Internal Rept. (prepared after 1976).
116. Optimizing Traffic Flow in an Intercity Corridor: Volume I, Program Discussion and User Guide; Volume II, Sub-Routine Description; Volume III, SCOT Model Modification and Traffic Control; and Volume IV, Derivation of Traffic Logic. Sperry Systems Management, Jan. 1978.
117. R.A. Reiss and others. Corridor Control of Vehicular Traffic. Sperry Systems Management, Internal Rept., 1978.
118. P. Zove and C. Berger. Integrated Motorist Information System Feasibility and Design Study--Phase II: Generalized Methodology for IMIS Feasibility Studies; Volume I, IMIS Feasibility Study Handbook; Volume II, Feasibility Handbook Application. Sperry Systems Management, May 1978.
119. Traffic Logic Application Summary. Sperry Systems Management, Sept. 1979.
120. M.C. Davila and E.B. Lieberman. A Hybrid Macroscopic-Microscopic Traffic Simulation Model. Presented at 59th Annual Meeting, TRB, Jan. 1980.
121. E.B. Lieberman. Traffic Simulation: Past, Present, and Potential. Proc., International Symposium on Traffic Control Systems, Vol. 1, Resource Papers, Univ. of California, Aug. 1979, pp. 242-261.
122. M. Feldman and A.D. May. FREQ3CP Student Workbook and Instructor's Guide. FHWA, June 1976.
123. A.D. May and R. Cooper. FREQ4CP Student Workbook and Instructor's Guide. California Department of Transportation, Sacramento, March 1977.
124. V. Siu, P. Jovanis, and A.D. May. Freeway Priority Treatment Strategies Workshop Student Workbook. April 1980.

Enhanced NETSIM Program

E. Lieberman

The NETSIM traffic simulation model (1) has been applied extensively over the past seven years to a wide variety of problem areas by a large number of public and private agencies. The experience gained with NETSIM has prompted many suggestions for improving and extending the program with the view to further enhancing its value as an engineering and research tool.

In an informal survey conducted by KLD Associates a few years ago, the following suggestions were made:

1. The input preparation effort should be eased,
2. The cost of computing should be reduced,
3. Many additional features should be introduced, and
4. The output capabilities should be extended.

Interestingly, the last two suggestions conflict with the first two. Whenever additional features are introduced, some added input requirements are usually implied. Furthermore, any additional feature leads to the development of additional software that, in turn, occupies computer memory and consumes computer resources. Similarly, enhanced output capabilities imply the need to compute and to store additional data; writing output is also costly in computer time. Such conflicting user requests impose a burden on the designer to be responsive in the most cost-effective manner.

This paper describes the enhancements incorporated into the new version of NETSIM. This version constitutes the result of the first development stage of the Integrated Traffic Simulation Software System known as TRAF.

The techniques that have been applied to produce a cost-effective, enhanced version of NETSIM will also be described.

NETSIM ENHANCEMENTS

Specific NETSIM enhancements are described briefly here.

Blockers and Parkers

Blockers are defined as illegal parkers who occupy a

portion of a lane dedicated to moving traffic. Such violators may be either short-term (less than 1 min) or long-term, and generally represent pickup or delivery (PUD) activities. Blockers exact a toll on the traffic stream in the form of increased travel time, reduced capacity, or both.

Unlike the previous versions of NETSIM, this enhancement explicitly models the interaction between moving vehicles and blockers. In addition, the impedance experienced by vehicles while attempting to evade such blockers through the mechanism of lane changing is also modeled. The concept of a "preferred lane" was introduced, which provided a basis for vehicles to return to a blocked lane downstream of a blocker.

Parkers are also treated as impeders, but are restricted to parking zones of specified location and length along the curb. The duration and location of parkers and of blockers are assigned by the program by using data specified by the user.

Look-Ahead Feature

When the first vehicle in a lane approaches an intersection to execute a through movement, it now responds in a car-following mode to its lead vehicle, which is on the receiving link. In the previous version, the subject vehicle did not "see" its leader if the leader was on the receiving link. Consequently, it was possible for the subject vehicle to "collide" with its leader if the latter was at the tail of a long queue.

Overflowing Turn Pockets

During periods of heavy demand, turn bays (or "pockets") of inadequate length could overflow, thus blocking the adjoining through lane. In previous versions, this overflow condition was not represented. The current version explicitly models this condition.

Bus Stops and Pockets

When a bus stop is created in the parking lane (by prohibiting parking there) or is created by a bay cut into the curb, it is called "protected". That is, a bus in dwell at a protected station will not block vehicles in a moving traffic stream.

Often, such protected nearside stations, when empty, are used by right-turning vehicles as a turn pocket. Ignoring such usage can have a pronounced effect on the validity of the simulation results. The new version of NETSIM incorporates logic that represents this behavior.

Dual Turns

When traffic on the "leg" of a T-intersection can execute both right and left turns, it is necessary to assign this turning traffic to appropriate lanes. If this "leg" approach has more than two lanes, it is necessary to assign traffic to the center lane in an appropriate, consistent manner. Additional logic was introduced to improve this feature relative to prior versions of the model.

Lane Alignment

Occasionally, the number of lanes on a link will differ from that on the downstream receiving link. Even when the number of lanes is the same, it is possible for one link to be offset relative to its receiving link. To account for such cases, a new feature was introduced to allow the user to specify the lane alignment between subject link and its receiving link.

Improve Efficiency

In previous versions of NETSIM, vehicles on entry links are treated the same way as vehicles on internal links. While there is no functional problem with this approach, it is more costly in computer resources--storage and time--than is necessary. Since no statistics are gathered on vehicles occupying entry links, it is permissible to limit the number of vehicles actually stored to those at the stop line. This approach is now incorporated into NETSIM.

Extended Range of Program

The new version permits up to 12 signal intervals to be specified instead of the previous maximum of nine intervals. Also, turn pockets may have two lanes instead of the previous limitation of one. (Each link may have seven lanes including those within turn pockets, compared with five lanes previously.)

A new feature has been introduced that permits the user to specify up to 16 vehicle types assigned to four categories. Each vehicle type is defined in terms of its length and acceleration, speed and discharge headway properties, and the categories to which it is assigned. This information allows the categorization of vehicle types as indicated in the following table:

Vehicle Type	Category: Private Automobile	Car-pool	Bus	Truck	Total (%)
1	60	40	0	0	100
2	10	30	10	50	100
3	0	0	100	0	100

In the above table, for example, 10 percent of the vehicles of type 2 appear on the analysis network as private automobiles, 30 percent as carpools, 10 percent as buses, and 50 percent as trucks. The percentage of the vehicle fleet that is composed of type 2 vehicles--not shown in the table--is specified by the user.

The vehicle category concept permits the user to specify different treatments for each category, e.g., special lanes. Other treatments can be added in the future as the need arises. A new treatment is the addition of carpool lanes and lanes for carpool vehicles and buses.

Extended Input-Output Capabilities

With the onset of the metrication program, it became advisable to provide the user with the capability of specifying input data in either customary or metric units, and of obtaining output data expressed in either or both units. This feature is now available.

Outputs are also provided in person-specific units, based on user-specified occupancy for each vehicle type. This feature permits the user to examine the people throughput and travel time for different high-occupancy-vehicle strategies.

The operational performance of traffic on a roadway segment (i.e., link) is a function of the turning movements at the intersection. It is well known that different turning movements are serviced at different rates. Consequently, vehicles performing one maneuver may experience significantly greater delay than those performing another maneuver. In fact, it is entirely feasible for the vehicles executing a left turn, for example, to experience severe congested conditions, while those moving straight through the intersection experience little delay.

The prior versions of NETSIM did not provide the

data necessary to obtain this level of detail and insight into the operational conditions. The new version does output measures of effectiveness (MOE) for each link that is stratified by movement if so requested.

Another new output feature permits the user to aggregate specified contiguous links for the purpose of obtaining statistics for that group of links. This is very useful for those who wish to examine how small sections within a larger network are operating.

A limitation on the placement of long loop detectors was removed.

Other features were also introduced. The number of lanes on a link was increased from five to seven and additional flexibility provided for the fuel consumption and vehicle emission feature. Internally, the treatment of queued vehicles was greatly improved, providing smooth vehicle trajectories regardless of queue length.

MODIFICATIONS TO EASE USER ACCESS AND MINIMIZE COST

Input Format

The input stream is designed as a collection of card types; each card type contains data that are functionally coherent. For example, data describing geometric characters are organized on separate cards from data describing traffic flow characteristics. Furthermore, data required for optional features (e.g., bus traffic, detectors, and vehicle types) are assigned to special card types that may be omitted if the feature is not used. As before, all input data items are specified as integers. With a few exceptions, all field widths are set to four columns, or digit positions, to promote a uniform format.

The input-processing software contains a wide range of diagnostic tests that are far greater in number than those of prior versions. It is our view that this extensive investment in software development is amply justified by subsequent savings in user resources. These tests are applied to the entire input data stream regardless of the number of user input errors detected. Each such error produces a diagnostic message that provides sufficient information to identify the source and cause of the error.

In addition, warning messages alert the user to examine input data that the logic determines to be suspect in some sense but that may be perfectly valid. For example, unusual network topologies are flagged by the software to prompt the user to confirm the validity of the relevant inputs. Of course, any fatal error detected by the software will terminate execution prior to any simulation processing.

Throughout the input stream, default values are provided by the software whenever feasible. While this feature relieves the user of significant effort in data preparation, the user should be cautioned to confirm that these default data items will not compromise the integrity of the study.

In our view, the best long-term solutions to minimizing user effort, in addition to the features noted above, are a file management system and automated data entry.

A file management system would consist of software that would enable the user to manipulate existing, stored input data bases so as to conduct a series of studies, with a minimum expenditure of time and effort. Ideally, this would be accomplished on-line by using a CRT terminal; an off-line system would also be cost-effective but to a lesser extent. It would work as follows:

1. The user will input the data stream for the base case. The file management software will store this data stream.
2. The program will perform its diagnostic tests and identify any and all errors.
3. The user will then correct these errors by modifying the input data stream appropriately, employing the file management software.
4. Steps 2 and 3 will be repeated until a satisfactory data stream is acceptable to NETSIM. The file management software will then store this correct data stream, properly identified for subsequent retrieval.
5. For all subsequent runs, the user will retrieve a data stream, implement necessary changes, and continue with step 2.
6. The user could purge any data stream at any time to reduce storage costs, subject to satisfying security measures designed to protect the stored files.

Automated data entry, such as the system installed by the Michigan Department of Transportation, will greatly ease the task of input data coding. This software could be integrated with the existing NETSIM software that performs card-specific diagnostic testing so that coding errors could be detected immediately, before the data are stored.

By integrating the automated data entry, diagnostic testing, and file management software into a separate preprocessing system, distinct from the main body of the NETSIM model, the appeal of NETSIM as an engineering tool will be greatly enhanced. The cost of program implementation will be greatly reduced, the input coding activity can be assigned to subprofessional personnel, and the elapsed (turnaround) time between data entry and receipt of the simulation results will narrow.

Minimize Computer Resources

The user is charged for a wide variety of computer resources, the most prominent being storage and time. The TRAF system has been designed to provide the user community with versions of the NETSIM (and other) models that are tailored to user needs:

1. The features required by the user will be provided; all others, not used, will not be included in the program. This capability will limit the computer memory required.
2. The size of the internal data base will be limited to that which is required by the user. Different versions of NETSIM will be available so that users can select the version that is most suitable for the size of the network to be studied.

Other modifications were designed to minimize input-output activity. Specifically, calculation of energy consumption and of vehicle emission for different vehicle fleet compositions may be accomplished with only a single execution of the simulation program. Also, the calculation of these environmental measures is accomplished without the need for spooling trajectory data to and from disk storage, as was done previously.

Reliability and Flexibility

The software maintenance function is an ongoing activity that must be responsive to the needs of the user community. This activity must provide for (a) reliable software (corrective maintenance) and (b) need for new or modified capabilities (constructive maintenance).

The new version of NETSIM has been designed and

programmed by using the latest techniques of structured software development. As a result, the investment required to perform these needed maintenance activities should be substantially lower than that which would be required if traditional techniques had been applied.

Every effort has been made to develop software modules (i.e., routines) that are functionally independent, are of limited size, and are logically cohesive. Such software has proven to be more reliable and more amenable to change than that of the previous version whose development predated the evolution of structural techniques.

For example, the new enhancements enumerated earlier in this paper were introduced by adding new routines, with moderate changes introduced to existing routines. The additional input card types that were required were essentially plugged in to the existing input-processing software with virtually no disturbance to the existing code.

To expedite corrective maintenance, special diagnostic software was developed. This software permits the maintenance analyst to selectively and efficiently examine the software performance in the process of locating the cause of any adverse symptoms, to identify any software defect.

It must be emphasized, however, that any software system requires a continuing program of maintenance support in order to enhance the reliability of the software and to provide the responsiveness needed by the user community. It is an established fact that the use of a software product is directly related to the confidence of the user community in the performance and utility of the product and in the availability of continuing support.

Availability of New NETSIM Program

The NETSIM program is a component submodel of the current version of the TRAF software system. This current version, named TRAF I.5, combines NETSIM with TRAFLO in an integrated format. That is, one can implement NETSIM concurrently with any of the submodels in TRAFLO (2), on a single analysis network. Details are provided in the TRAF I.5 Users Guide (3).

Of course, NETSIM may be executed, as in the past, as a stand-alone program. The availability of TRAFLO within the overall structure of the TRAF software system does not burden NETSIM users in any way. Each submodel in the TRAF system resides in a separate overlay that is stored on disk; a submodel that is not used is not retrieved from disk and does not consume computer time nor storage in central memory.

As noted earlier, the input format is designed so that no additional inputs are necessary for any submodel that is not used. In summary, the additional capability provided the user community through the integrated simulation model concept implies no penalties whatever if only one submodel, such as NETSIM, is used.

REFERENCES

1. Traffic Network Analysis with NETSIM--A User Guide. FHWA, Implementation Package FHWA-IP-80-3, Jan. 1980.
2. E. Lieberman. Macroscopic Simulation for Urban Traffic Management--The TRAFLO Model: Vol. 1, Executive Summary. FHWA, June 1980.
3. E. Lieberman and others. TRAF I.5 Users Guide. FHWA, June 1981.

Models for Design and Evaluation of Traffic Signal Timings

Kenneth G. Courage and Charles E. Wallace

Optimization and evaluation models are valuable aids to the design of traffic signal systems. While many traffic engineers still use manual techniques for this purpose, others are finding that computer models offer substantial improvements both in the final product and in the productivity of the staff creating that product.

The product, of course, is signal timing. Its parameters are the duration and sequence of the signal phases at a given intersection and their relationship to similar parameters at neighboring intersections. The results of a good product are fewer stops, less delay and fuel consumption, and reduced accidents--all of which lead to lower operating costs for the motorist. Although the quality of the product is ultimately determined on the street, several traffic signal optimization and evaluation models have proved their ability to assist the traffic engineer in developing cost-effective operational improvements.

MODEL CLASSIFICATION

Most traffic signal models in practical use are macroscopic and deterministic. They deal with the traffic stream as a whole and not with individual vehicles. They make little or no use of probabilities or statistical distributions. Most do not use sophisticated analytical techniques; they rely instead on search techniques, simple analytical equations, or graphical approaches. On the other hand, some excellent applications of operations research techniques are also apparent (e.g., hill climbing, linear programming, etc.). The best way to classify the models to be discussed in this paper is by the following four areas of application:

1. Single intersections,
2. Arterial routes,
3. Two-dimensional networks, and
4. Diamond interchanges.

Each of these areas has unique problems and objectives, and each, therefore, has generated its own models.

The specific models discussed in this paper are shown in Table 1 and are classified by application area. This table identifies five computer programs that are frequently used for design and evaluation and summarizes the most important functions of these programs. This list is not exhaustive; other programs are also available, or under development. (See also papers by Gibson and May in this report.)