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
d. 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

Enhanced NETSIM Program

E. Lieberman
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 be-
tween moving vehicles and blockers. In addition,
the impedance experienced by vehicles while attempt-
ing 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 loca-
tion 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 leading vehi-

cle, which is on the receiving link. In the pre-
vious version, the subject vehicle did not "see" its
leader if the leader was on the receiving link.
Consequently, it was possible for the subject vehi-
cle 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 repre-
sented. 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 occupyp-
ing 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:

<table>
<thead>
<tr>
<th>Vehicle Type</th>
<th>Category</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carpool</td>
<td>Automobile</td>
<td>100</td>
</tr>
<tr>
<td>Bus</td>
<td>Truck</td>
<td>100</td>
</tr>
<tr>
<td>Private Car</td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>

In the above table, for example, 10 percent of
the vehicles of type 2 appear on the analysis net-
work 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 speci-
fied 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 carpools 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 perform-
ing one maneuver may experience significantly
greater delay than those performing another maneu-
ver. 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.

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 (turn-around) 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
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—each 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.)