Network Programming To Derive Turning Movements from Link Flows

PETER T. MARTIN AND MARGARET C. BELL

It is generally accepted that there is a need to develop traffic models to manage and control congestion in real time. This control needs to be integrated with route guidance systems in order to achieve rerouting. A basic requirement of such models is to automatically establish turning movements from link flows. Conventional models such as entropy maximizing and information minimizing have been developed for use off line for transportation planning and are not suitable for application on line. A novel approach is proposed that uses linear programming to forecast traffic congestion in an urban network and define junction turning flows. The algorithms, originally developed for optimizing flows of water and electricity, use detector flow measurements, weighted links, and constrained upper and lower flow bounds. The principles underlying this approach are explained. The development, calibration, validation, and implementation of the model in a real network in the city of Leicester, England, are described. The results show that turning movements can be predicted with sufficient accuracy to justify further work, in particular to carry out a demonstration of the application of the algorithm on street.

Current coordinated traffic signal systems, both fixed time and demand responsive, are designed to control networks operating at 90 and 95 percent saturation. Some demand-responsive systems have facilities to gate, meter, and favor traffic to ease congestion. Currently, congestion management relies heavily on the engineer's judgment of the traffic situation viewed through closed-circuit television systems and can only be achieved with operator intervention. Often small adjustments to timings help to alleviate congestion locally. However, comprehensive management and control of congestion need strategic control because traffic has to be redistributed along those routes with sufficient spare capacity, otherwise the overload would simply be transferred to another part of the network. Such a control system would be based on a philosophy that optimizes space in time. It would need to monitor traffic movements to forecast the onset of congestion and select appropriate control strategies. The system would monitor performance on line, build up a knowledge base, and eventually learn from experience, in other words, an expert system.

Traffic detection is advanced and reliable (*1,2*), and the current research effort is investigating methods for data-base management (*3*) of large volumes of information on line. Therefore, the infrastructure to support the implementation of the expert system techniques will shortly be available.

P. T. Martin, Department of Civil Engineering, University of Nottingham, University Park, Nottingham NG7 2RD, United Kingdom. Current affiliation: ARDFA, California Polytechnic State University, San Luis Obispo, Calif. 93407. M. C. Bell, Department of Civil Engineering, University of Nottingham, University Park, Nottingham NG7 2RD, United Kingdom.

Traffic routes vary for many different reasons: longer-term shifts follow changes in land use and implementation of traffic management schemes, and in the short term changes occur in response to the build-up of recurrent congestion and incidents. Also, drivers' choice of route and demand for travel vary daily and seasonally. A fourth generation of signal control will therefore have to continuously monitor the shifts in traffic routes that take place and the corresponding transient changes in traffic conditions. The linear programming method proposed here, once calibrated, could be used to predict these changes in traffic patterns as they happen, thus allowing remedial congestion control strategies to be defined on line.

On a large scale, traffic routes are described by origin and destination matrices. On a small scale, traffic routes are defined by the flow along links and the turning movement at the junctions. The corresponding signal control strategy, defined off line in a fixed-time system and on line in a demand-responsive system, matches or governs the demand for travel on each link. The vehicle detectors supply a measure of link flows and these, with the constraints of the signal control timings, enable the turning movements to be inferred. The linear programming technique is therefore system driven, which is quite different from conventional transportation modeling, which is behavioral.

Conventional models that derive origin-destinations from link flows have been developed for transportation planning rather than for signal control. Since they are applied off line, computer run time and storage are not critical. Maximum entropy methods (*4*) select the most likely origin and destination matrices to be consistent with the given set of link flows with the least bias. Information minimization models (*5*) are founded on the principle that for a junction, a set of turning movements exists that is most probable. However, a solution fails to produce converged solutions when the data are noisy, which is usually the case for traffic. Therefore, dynamic methods (*6*) have been developed to interpret the rhythmic nature of the traffic data, but success has been limited.

Predicted flows from these large-scale transportation models rarely agree with those measured on the street, but are adequate for planning. These models are both slow and exhaustive on computer time and in their present form are unlikely ever to have application for traffic monitoring and control on line.

Previous research (*7*) had demonstrated that linear programming had potential. The particular advantage of the method was in the speed with which solutions to fairly large and complex problems could be achieved. This early study was based on traffic data generated by Monte Carlo simulation modeling of flows on the network. The main conclusion of this study was that before substantial advances could be made in solving the problem of deriving turning movements from detector flows, there was a need to research the algorithms applied to traffic data from a real network. This could only be done by having direct access to signal timing and detector flow data from a network and carrying out surveys to enable simultaneous measurement of junction turning movements. It was for this reason that funding from the Science and Engineering Research Council (SERC) was secured to set up surveys to provide unique data sets. The third annual survey was made in May 1991.

THE STUDY AREA

Leicester, in the East Midlands of England, has a population of 289,000 and is a city with a series of radials and ring roads. In 1988 the fixed-time signal control system was replaced by a demand-responsive traffic signal control system, Split Cycle Offset Optimization Technique, or SCOOT (*8*). A subarea of SCOOT, Region R, to the south of the city was chosen for this research.

Figure 1 shows the street network of six signal-controlled junctions. This particular SCOOT subarea was chosen because it has a spatial geometry that offers alternative routes to traffic both into and out of the city center. During the morning peak, congestion builds up along London Road and drivers are known to seek alternative routes along Regent Road. During the evening peak, traffic can leave the city either along London Road or Regent Road. Another reason for choosing Region R was the knowledge that substantial route changes were expected following traffic management alterations. A new link to the ring road was built in 1989, and the city center was pedestrianized on October 14, 1990. These network changes are substantial and likely to have created noticeable shifts in route patterns in Region R.

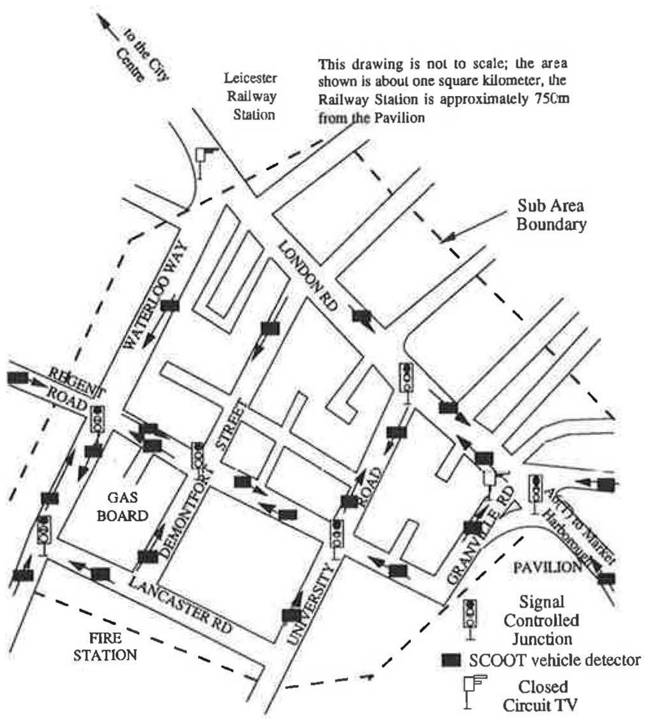

FIGURE 1 Street layout, SCOOT Region R, central Leicester.

The Nottingham University Transport Research Group (NUTRG) has a computer work station with a dedicated communication link with the Leicestershire County Council Traffic Management Computer. This link allows traffic flows, congestion indicators, and signal timing data to be retrieved at 5-min intervals continuously from detectors throughout Leicestershire. Detector flow data were monitored via SCOOT for 21 of the 170 links in the study area. Seventy on-street observers provided a comprehensive set of turning movements measured simultaneously with the gathering of the link traffic flows from the detectors. Three surveys, on Wednesday May 10, 1989, Wednesday May 9, 1990, and Wednesday May 8, 1991, between 1500 and 1800 hours provided comprehensive data. The linear programming method was used to infer the 54 unknown turning movements at 7 signal-controlled junctions from the 21 measured flows from SCOOT.

THE ALGORITHMS

A linear program model has three components. There must be a system, a problem, and a solution. The system is the network represented by a series of nodes connected by arcs. The geography of the town or city defines the structure of the network. Each arc accommodates a traffic demand or flow. The term "arc" is the network representation of a "link," which is used to describe the road. The signal timings quantify the control environment, that is, the capacities of both junctions and links. The problem is to predict a comprehensive set of turning flows from a sparse set of link flows measured by detectors. The solution is the feasible set of turning movements predicted with acceptable accuracy that satisfies the constraints.

When the system is constrained too severely, no feasible solution can be found. When the constraints are too lax, feasibility is too readily accomplished and the prediction is poor. The challenge of this research was to identify the constraints for the linear programming algorthms in such a way that all relevant information available from the control system is used so that the particular feasible solution, the correct one, is identified. The resulting solution then represents the turning flows reliably. It is important that the constraints conform to some network or traffic characteristics that are quantifiable. This enables the algorithm to be both repeatable and transferable. In this way the model can be applied to any control network for varying traffic demands at different times of day.

The algorithm relies on the Simplex method (*9*), which defines and solves a series of inequalities. If a set of solutions to the series of equations exists, such solutions are said to be feasible. The one solution that results from the minimization of a cost or a weighted function of attributes is known as the optimum solution. An algebraic procedure moves from one basic feasible solution to a better adjacent feasible solution by choosing a basic entering and leaving variable to solve a system of linear equations using Gaussian elimination. An iterative procedure is applied until the solution cannot be further improved, which is deemed the optimal solution, and the computation of the algorithm stops. The program NETFLOW (*10*), written in FORTRAN, is the Simplex method in matrix form, which serves to streamline the original method considerably and makes it computationally faster with potential on-line application. At each iteration, sparse matrices hold the minimum of information in a more compact form, and therefore the procedure requires less computer storage. As a result much larger network problems can be solved. The modeling process is summarized in Figure 2.

NETFLO ALGORITHM

The network is modeled as a series of nodes connected by arcs. Arcs represent homogeneous stretches of road between junctions. Nodes represent the intersection of arcs, which are unidirectional. The NETFLO model applies a heuristic procedure to obtain the initial basic feasible solution. The main idea of this procedure is to quickly find the low-weight paths through the network that will accommodate the demand to and from each node in the network. Mathematically, the linear programming problem reduces to simply minimizing an objective function f that takes the form

$$f = \mathbf{w}\,\overline{\mathbf{q}} \tag{1}$$

such that

$$\mathbf{A}\,\overline{\mathbf{q}} = \overline{\mathbf{r}} \tag{2}$$

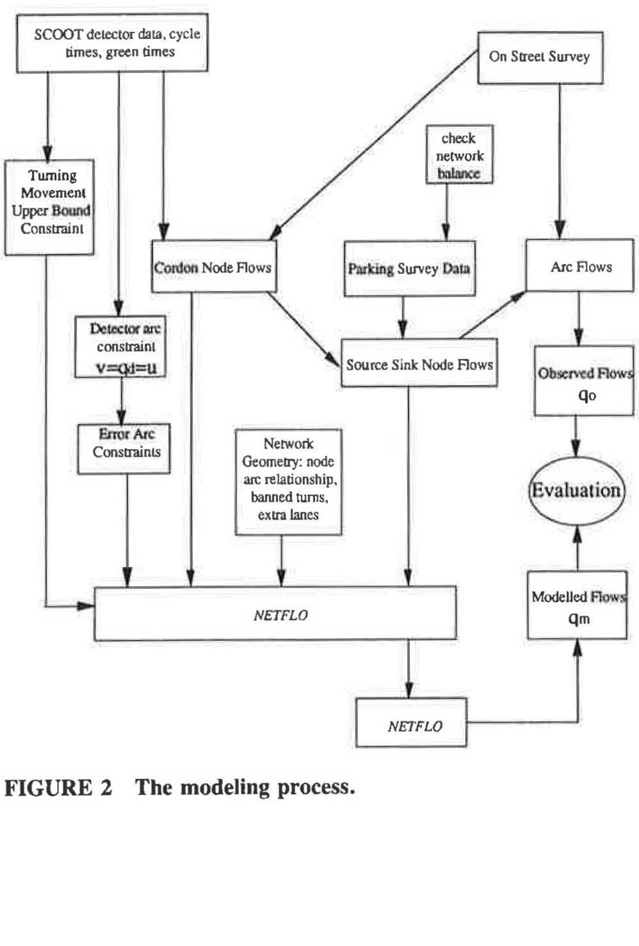

FIGURE 2 The modeling process.

and

$$\mathbf{0} \le \overline{\mathbf{v}} \le \overline{\mathbf{q}} \le \overline{\mathbf{u}} \tag{3}$$

where

- $\mathbf{A}$ = $i \times j$ node-arc incidence matrix, which defines the network structure;
- $\mathbf{w}$ = $1 \times j$ vector of unit weights associated with each arc;
- $\overline{\mathbf{r}}$ = $i \times 1$ vector of flows in and out of nodes, known as the requirement vector;
- $\overline{\mathbf{v}}$ = $j \times 1$ vector of arc lower bounds;
- $\overline{\mathbf{u}}$ = $j \times 1$ vector of arc upper bounds or arc capacities; and
- $\overline{\mathbf{q}}$ = $j \times 1$ vector of arc flow, the decision variable that is the unknown quantity.

The node-arc incidence matrix $\mathbf{A}$ is defined so that element A_{ij} of the array takes a value of $+1$ if Arc j is directed away from Node i and a value of -1 if Arc j is directed toward Node i. Should Arc j and Node i not meet, the value 0 is applied. If for Node i, $r > 0$, Node i is a supply node where traffic flows into the network; that is, the supply is equal to r. If for Node i, $r < 0$, Node i is a demand node and traffic flows out of the network. The value of entry or exit flow is r. Internal nodes or turning points at a junction have $r = 0$; these are known as transshipment points. Introducing the vectors $\mathbf{q}$, $\mathbf{r}$, and $\boldsymbol{\alpha}$ in order to transform the lower bound $\overline{\mathbf{v}}$, now let

$$\overline{\mathbf{q}} = \mathbf{q} + \overline{\mathbf{v}} \tag{4}$$

and

$$\mathbf{r} = \overline{\mathbf{r}} - \mathbf{A}\,\overline{\mathbf{v}} \tag{5}$$

$$\boldsymbol{\alpha} = \mathbf{w}\,\overline{\mathbf{v}} \tag{6}$$

$$\mathbf{u} = \overline{\mathbf{u}} - \overline{\mathbf{v}} \tag{7}$$

as the lower bounds are reduced to zero. The objective function then reduces to

$$f_{\min} = \mathbf{wq} + \boldsymbol{\alpha} \tag{8}$$

such that

$$\mathbf{Aq} = \mathbf{r} \tag{9}$$

and

$$\mathbf{0} \le \mathbf{q} \le \mathbf{u} \tag{10}$$

The NETFLO program applies a heuristic procedure by quickly finding paths through the network to satisfy node continuity. Spanning trees are supplemented by artificial arcs. A spanning tree is a subsidiary network that contains all the nodes but a number of arcs are reduced so that loops are eliminated. Part of a spanning tree is formed so as to satisfy the induced supply and induced demand. For each supply node, s, an artificial node, y, is set up, connected by an artificial arc (s,y) of weight ∞, capacity ∞, and flow t_{sy} satisfying node continuity. For each

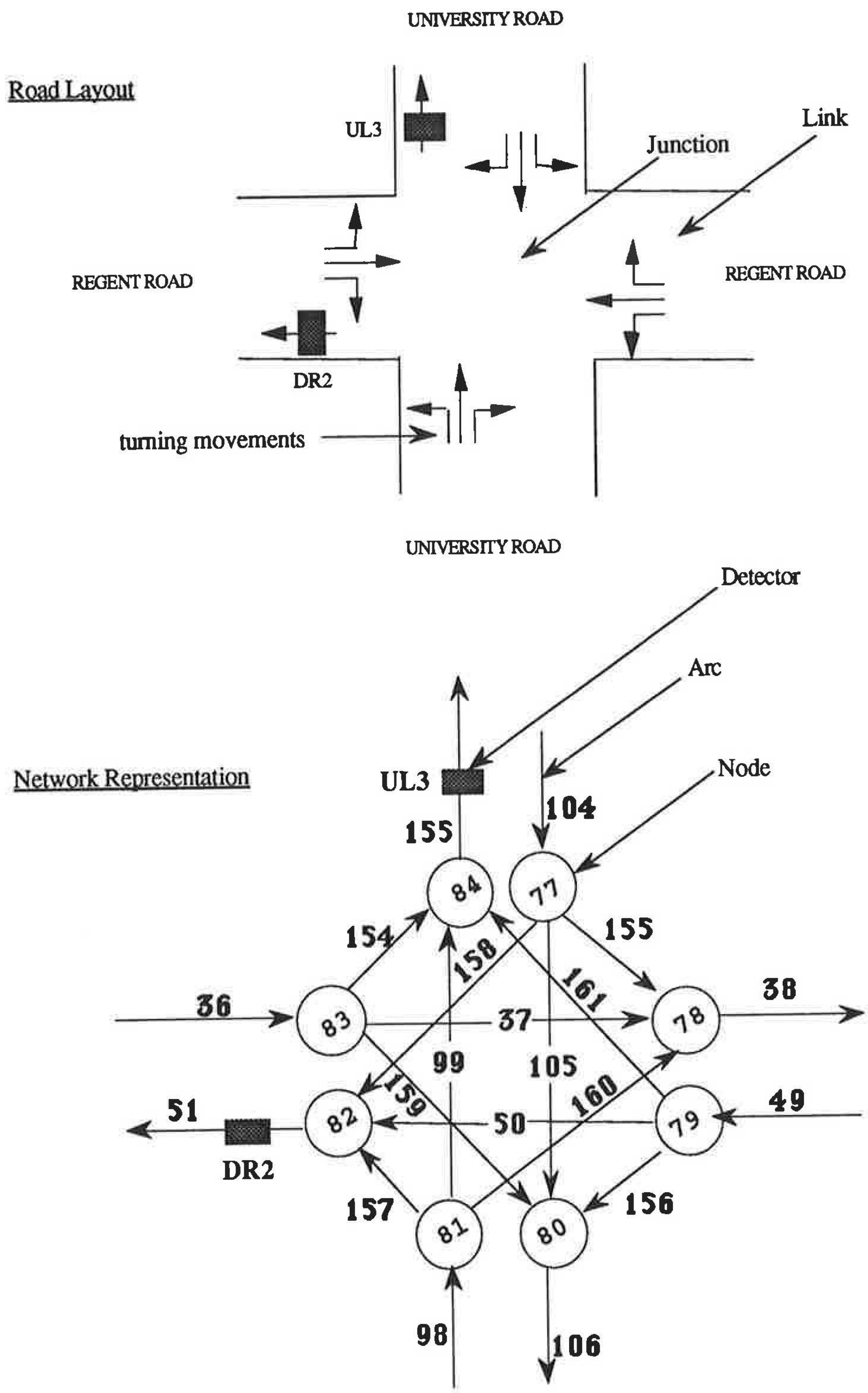

FIGURE 3 Node arc representation of a junction.

demand node (d), where flow leaves the network, an artificial node p is set up connected by an artificial arc (p,d) of weight ∞, capacity ∞, and flow t_{pd} satisfying node continuity.

Since the artificial arcs are given such high weights, the optimum solution is dominated by a set of flows whereby all artificial arcs are assigned zero flow. The approach is similar to the "Big-M" method (*11*), whereby the objective function is supplemented with an additional term M, which denotes a very large positive number and thus carries an overwhelming penalty.

The objective function now becomes

$$f_{\min} = \mathbf{wq} + \boldsymbol{\alpha} + M\mathbf{t} \quad (11)$$

where $\mathbf{t}$ is an $[m \times 1]$ vector of artificial arc flows, where m is the number of artificial arcs generated by the particular network. The spanning trees are completed with the addition of more artificial arcs, and basis exchanges are performed to achieve optimality.

NETWORK DEFINITION

A formal node-arc classification has been devised. The node-arc representation of a simple two-way junction at the intersection of Regent and University roads is shown in Figure 3. The notation is shown in Figure 4. Each traffic movement at the road intersection is expanded so that each turn is represented by its own unique arc. The arcs are represented by straight lines and the nodes by circles, triangles, or squares. Left-turn movements are uncrossed. All other turning movements, including straight-on maneuvers, are shown as crossed lines. Banned turning movements are accommodated by the absence of an arc in the network. Channelized lanes are modeled as separate arcs. The network representation of the street network of Region R is shown in Figure 5. Figures 2 and 5 have direct equivalence.

Five types of arc are defined:

1. An internal arc transmits flows from any node type to any other node type,
2. An external arc represents entry and exit flows; it is not afforded a label;
3. The external detector arc sits on the cordon perimeter and models the site of a SCOOT detector that supplies flow to a cordon node,
4. An internal detector arc models the site of a SCOOT detector within the network, and
5. A sorsink arc, like an external arc, is virtual, signifying the inflow or outflow of traffic from a source or sink within the cordon.

An external arc is a virtual arc that signifies the entry or exit flow into or out of the cordon. The model does not assign flow along either external arcs or external detector arcs because one end of the arc is free.

Three types of real node are defined:

1. A cordon node marks the point of entry to or exit from the network. Each is associated with an external arc and is connected to any number of simple arcs or detector arcs, or both.
2. Sorsink nodes mark the point of injection or extraction within the cordon. They always have one virtual sorsink arc, one entering arc (simple or detector), and one exit arc (simple or internal detector).
3. Transshipment nodes are not connected to external or sorsink arcs. They have any number of inflow arcs (simple or internal detector) and any number of outflow arcs (simple or internal detector).

CONSTRAINT REGIME

Node continuity is the principle of conservation of traffic flows at a node; it is analogous to Kirchoff's law for electric flow. For the linear program to respect node continuity and obey the law, net inflow must match net outflow. Cordon arcs model the external flows that enter and leave the network. The net imbalance of external flows is matched by a set of sorsink flows. Sorsink arcs model flows generated by internal sources and flows absorbed by internal sinks. The imbalance is distributed among the sources and sinks in proportion to the capacity of on-street parking and car parks. (Later enhancement of the model will include data showing level of

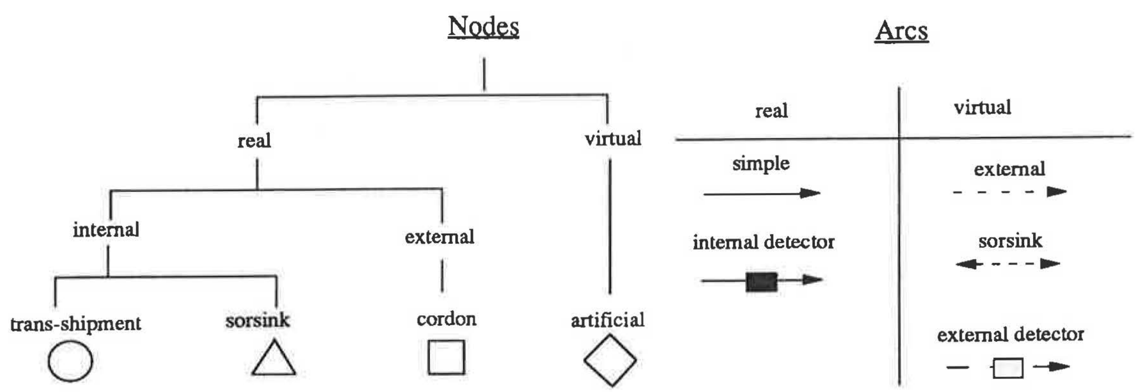

FIGURE 4 Network notation.

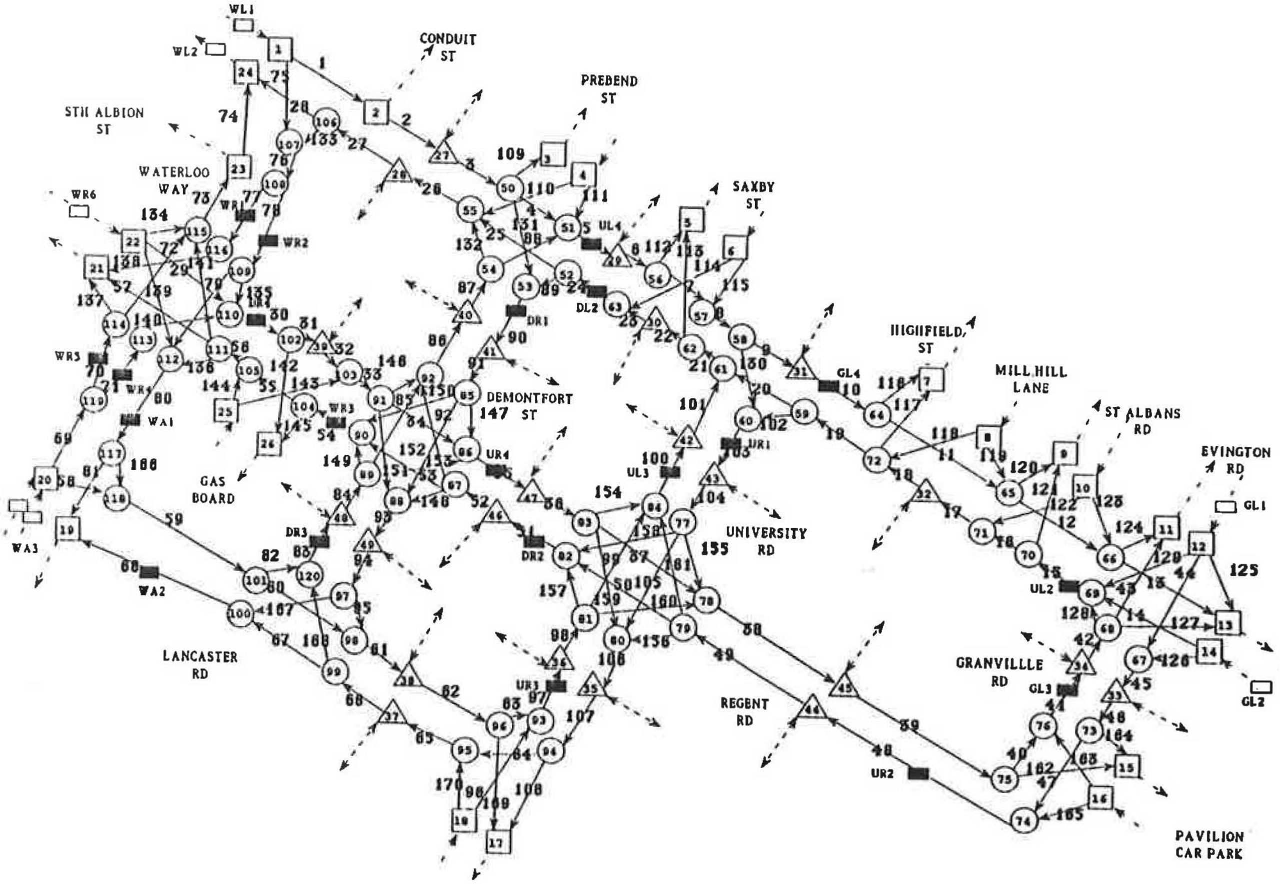

FIGURE 5 The network.

car park usage from automatic vehicle detection systems at the entrances and exits of carparks.)

Once the external and internal flows have been loaded onto the network, measured link flows from detectors are introduced. A degree of flexibility must be introduced to accommodate the dynamics or variability of the system and permit a feasible solution. A known flow, such as that measured on line from a detector, may be forced into the solution by specifying

$$u = v = q_d \qquad (12)$$

where q_d is the detector flow.

However, fixing the upper and lower bounds in this way and tightly constraining the algorithm by forcing the detector flow along specified arcs leads to infeasibility. On the other hand, excessive relaxation of the upper and lower bounds merely serves to dilute the quality of the detector information. Therefore, a means of providing "room to breath" is needed. A series of constraint regimes was tested in order to simulate the effect of including detector data flows. Bell (7) proposed a method of introducing flexibility to the system by introducing an additional two error arcs $e+$ and $e-$ alongside the arc representing the measured flow, as shown in Figure 6.

This flexibility enabled the linear program to arrive at a feasible solution. In defining a solution for the network, the linear program assigns flows to arcs to minimize the sum of

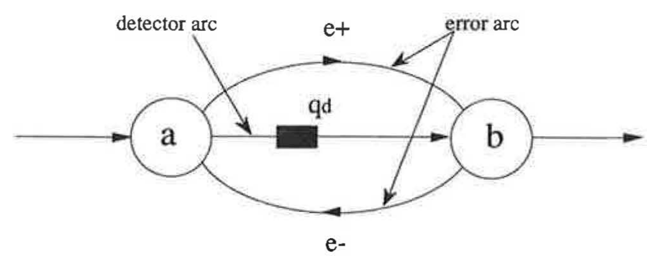

FIGURE 6 Error arc configuration.

the flows on the error arcs over the entire network. A solution is found such that

$$\sum(e+ \; + \; e-) \text{ is a minimum for all links in the network}$$

A series of tests investigated the sensitivity of the network to different error tolerances. These produced an optimum error arc configuration to produce a best and still feasible solution. The application of a range of weights to the error arcs failed to improve the reliability of reproducing the detector arc flows. A combination of varying upper and lower arc constraints along the detector arc itself with adjustment in the upper bounds of the error arc showed that an appropriate solution would be obtained with a maximum tolerance of 2.5 percent of the measured flow.

The SCOOT model supplies flow levels and signal timings every 5 min. The green splits and saturation flows provide capacities for each signal phase as follows:

$$u = \frac{gsx}{\tau} \tag{13}$$

where

g = green time for the link defined by the fixed-time plan,
s = saturation flow at the stop line,
x = degree of saturation, and
τ = cycle time, assumed fixed for the network over the 5-min period.

The effect of using measured flows on different arcs, equivalent to moving the locations of the detectors in the network, was simulated. Detectors were "moved" closer to junctions and straight-on maneuvers were simulated. The Demontfort Street–Regent Road junction was subject to complete definition by constraining each turning movement with the known flow. The existing SCOOT detector locations proved to give the best estimate.

Sensitivity tests applying lower-bound constraints demonstrated no useful purpose. On the contrary, they tended to reduce the reliability of flow prediction.

RESULTS

The simulated flows compared with the flows actually observed on site are shown in Figure 7. The cluster points on the x-axis, close to the origin, represent low flow turning movements, which have been assigned a zero by the algorithm. Straight-on maneuvers are reliably predicted. Half of the right-turn maneuvers are successfully predicted, but the model failed to predict the majority of left-turn movements. Overall, the model performs well, achieving a correlation coefficient of 92 percent.

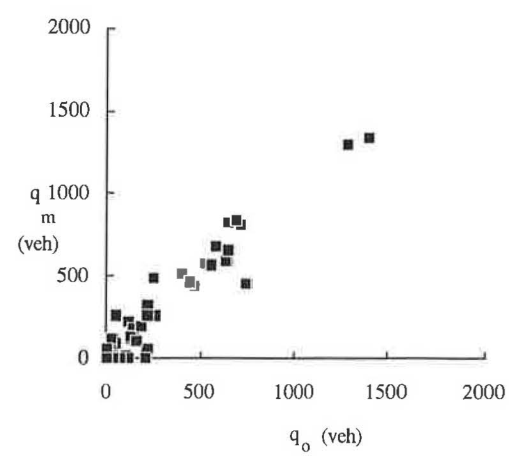

FIGURE 7 Turning movements: observed q_0 and modeled q_m.

CONCLUSIONS AND FURTHER WORK

The NETFLOW algorithm applied to traffic networks has been shown to be successful, yielding a correlation coefficient of 92 percent for predicted and actual traffic flows. However, it is the low flows, those of less importance in solving congestion problems, that are not reliably inferred.

The method proposed can deal with car parks as sinks and sources within the network. This element of the model is fairly complex and the modeling of car park storage is crude. More work is needed. Other limitations of the application of the model include the following:

1. The basic calculations are performed as integer arithmetic, which is especially important for the arc weights and for the calculation of sink and source flows from parking areas;
2. The minimum-flow parameter v cannot force the algorithm to assign a minimum flow to specified arcs; and
3. There is scope for an algorithm to be rendered dynamic by processing smoothed 5-min flow data in favor of the 1-hr aggregated flows.

The next step in this research program is to demonstrate a practical on-line application with traffic flows measured both into and out of the network. In the SCOOT environment, therefore, extra detectors would need to be installed. Further development of the algorithms will enable hypothetical incidents to be modeled by imposing heavy penalties on affected links. The flow predictions would provide a library of congestion control plans that could be implemented in conjunction with route guidance systems to improve traffic flow immediately after an incident. The algorithm could contribute to an expert system of traffic control.

The method stands apart from others in two ways. First, its route logic is node oriented. It moves through the network node by node, seeking to satisfy continuity by balancing inflows with outflows. The flows on consecutive links are effectively defined independently. Second, it has been structured to draw on detector data that are at the heart of a dynamically responsive unified traffic control model.

Transportation models that predict flows seek optimality by minimizing an objective function that usually has a behavioral basis. An objective function can take the form of the sum of a series of costed links. Costs reflect some form of material travel characteristic such as journey time, distance, or other perceived journey cost. An equilibrium model, for example, would suggest that drivers reroute when links approach capacity, minimizing journey length or travel time. Usually, there is a notion of cost implicit in the objective function minimized for optimality.

The approach described here has no behavioral basis because its purpose is to infer traffic movements from a limited set of detected flows. A weight function is introduced that serves as a controlling mechanism rather than a cost function. Some weights deter, whereas others encourage and no physical cost is minimized. The method does not seek to imitate drivers' route choice. In principle, it seeks to estimate the state of the traffic system (i.e., to derive turning movement flows and unknown flows) from geometric and traffic data. For this reason, the objective function plays a subordinate role. It is a mechanism for identifying the one solution from the many possible solutions and has no tangible meaning. Since each flow prediction is associated with a set of constraints, the aim is to find a constraint regime whereby the minimized objective function is associated with a particular solution and unknown flows are reliably inferred.

Notwithstanding the subsidiary role of the objective function, however, there remains scope for its improvement. Furthermore, the link weight functions are linear. Since flow-speed curves are nonlinear, some degree of gradually increasing weights would offer more realistic traffic modeling.

ACKNOWLEDGMENTS

The authors acknowledge the Science and Engineering Research Council and the Department of Civil Engineering at the University of Nottingham for funding the research and the Leicestershire County Council for their close cooperation and support.

REFERENCES

1. M. McDonald, N. B. Hounsell, N. Sittampalam, and F. N. McLeod. *Traffic Incidents and Route Guidance in a SCOOT Network*. Science and Engineering Research Council, United Kingdom, Oct. 1987.
2. P. Davies, D. R. Salter, and M. Bettison. Loop Sensors for Vehicle Classification. *Traffic Engineering and Control*, Vol. 23, Feb. 1982, pp. 55–59.
3. M. C. Bell and P. T. Martin. The Use of Traffic Detector Data for Traffic Control Strategies. In *Institution of Electrical Engineers International Conference on Road Traffic Control*. Conference Publication 320. IEE, London, May 1990.
4. H. J. Van Zuylen and L. G. Willumsen. The Most Likely Trip Matrix Estimated from Traffic Counts. *Transport Research B*, Vol. 14B, 1980, pp. 281–293.
5. H. J. Van Zuylen. The Estimation of Turning Flows on a Junction. *Traffic Engineering and Control*, Vol. 20, Nov. 1979, pp. 539–541.
6. M. G. H. Bell, D. Inaudi, J. Lange, and M. Maher. Techniques for the Dynamic Estimation of O-D Matrices in Traffic Networks. In *Proceedings of the DRIVE Conference: Advanced Telematics in Road Transport*. 4–6 Feb. 1991, Brussels, pp. 1040–1056.
7. M. C. Bell. *The Use of Automatic Control Algorithms To Define Urban Traffic Routes*. Internal Report. Durham University, 1987.
8. P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton. *SCOOT A Traffic Responsive Method of Coordinating Signals*. TRRL Report LR 1014. Transport and Road Research Laboratory, Crowthorne, Berkshire, England, 1981.
9. G. B. Danzig, A. Orden, and P. Wolfe. The Generalised Simplex Method for Minimising a Linear Form under Linear Inequality Restraints. *Pacific Journal of Mathematics*, Vol. 5, 1955, pp. 183–195.
10. J. L. Kennington and R. V. Helgason. *Algorithms for Network Programming*. John Wiley & Sons, New York, 1980.
11. F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, 5th ed., 1990, 81 pp.

Publication of this paper sponsored by Committee on Traffic Flow Theory and Characteristics.