Implementing Combined Model of Origin-Destination and Route Choice in EMME/2 System

PAUL METAXATOS, DAVID BOYCE, MICHAEL FLORIAN, AND ISABELLE CONSTANTIN

The issue of "feedback" in the traditional four-step urban travel forecasting procedure (UTFP) has reemerged recently under the pressure of the Clean Air Act Amendments of 1990 and the Intermodal Surface Transportation Efficiency Act of 1991. FHWA now requires that metropolitan planning organizations implement feedback in the UTFP. The combined origin-destination and route choice (OD-UE) model solves simultaneously the trip distribution and the user equilibrium traffic assignment models and hence provides for feedback. Computer codes for the computation and calibration of combined models are available from various researchers. However, they lack detailed documentation and, moreover, require computer programming expertise to adapt to professional practice. In view of these drawbacks, this paper documents the coding of a macro that implements the OD-UE model in EMME/2. The scope of this effort was twofold: first, to respond to certain modeling requirements arising from modern urban transportation planning practice; second, to motivate transportation professionals to use more sound planning methods. The quality of the results obtained using data from the city of Winnipeg, Manitoba, Canada, supports the use of the macro in planning applications.

The issue of "feedback" in the traditional four-step urban travel forecasting procedure (UTFP) has reemerged recently with the impetus of the Clean Air Act Amendments of 1990 and the Intermodal Surface Transportation Efficiency Act of 1991. FHWA now requires that metropolitan planning organizations implement feedback in the UTFP. A sound and mostly appealing alternative toward the solution to this problem is a model that combines the trip distribution, mode split, and assignment steps of the UTFP (*1*). This type of model is not new; its adoption, however, in transportation planning practice is slow. Transportation professionals seem to experience difficulty in understanding the solution procedure. In addition, research codes do not provide relief because they lack detailed documentation and require computer programming expertise to be adapted to professional practice. Moreover, software developers have ignored the issue simply because there has not been sufficient demand, at least until recently.

A number of algorithms that solve the combined origin-destination (O-D), mode choice, and user equilibrium traffic assignment model exist and their properties are well documented (*2,3*). Among those algorithms the Frank-Wolfe linear approximation algorithm and its variant Evans' partial linearization algorithm have been applied to large-scale urban networks. The algorithm implemented here is the one proposed by Evans (*4*); its advantages, compared with those of the Frank-Wolfe algorithm, especially for large-scale applications, have been reported elsewhere (*1,2,5–8*). In a recent study by Boyce et al. (*1*) the Evans algorithm for the combined distribution, mode split, and traffic assignment model was compared against various heuristics used in practice and found to provide superior results, as defined by its more rapid convergence to the true equilibrium solution.

Briefly speaking, four main reasons are presented as favoring the Evans algorithm. First, it is not heuristic; it is, however, a mathematical structure with well-understood properties. Second, the speed with which Evans fills the cells of an O-D matrix (all destinations are loaded from every origin at each iteration) is much superior to Frank-Wolfe (only two destinations per origin per iteration). Third, Evans' partial linearization approximation (as in all approximations of that kind) provides superior feasible directions (subproblem solutions closer to the optimum) compared with Frank-Wolfe linear approximation method. Fourth, the Evans algorithm provides an exact solution of the trip distribution model at each iteration given the current O-D travel costs, whereas the Frank-Wolfe algorithm converges only to the solution of the trip distribution model with equilibrium travel costs. The last becomes an issue in large-scale applications where a solution algorithm never reaches exact convergence because of the high computational costs involved.

In this paper the solution of a combined model of trip distribution and user-equilibrium traffic assignment (OD-UE) is discussed. In a subsequent paper the inclusion of mode choice will be discussed. The implementation of the Evans algorithm is realized by making use of the EMME/2 macro language capability to use various modules for mathematical and network operations both sequentially and iteratively.

The scope of this effort is twofold: first, to respond to the modeling demand arising from the modern urban transportation planning practice; second, to motivate transportation professionals to use more sound planning methods.

The paper is organized as follows. The implementation of the Evans algorithm for the OD-UE model in EMME/2 is documented in the next section. Immediately after, comparisons between the combined model and the sequential procedure are made. Finally, suggestions for future enhancements are made in the last section.

EVANS ALGORITHM FOR COMBINED OD-UE MODEL AND IMPLEMENTATION IN EMME/2

The combined OD-UE model formulated as an equivalent optimization problem requires one to minimize functions of the link travel costs (network term) and the costs of the O-D flows (demand term) subject to conservation of flow constraints, marginal constraints, nonnegativity constraints and definitional constraints. The

P. Metaxatos and D. Boyce, Urban Transportation Center, University of Illinois at Chicago, 1033 West Van Buren Street, Suite 700 South, Chicago, Ill. 60607. M. Florian and I. Constantin, INRO Consultants, Inc., 5160 Decarie Boulevard, Suite 610, Montreal H3X 2H9, Canada.

(Evans) partial linear approximation method linearizes only the first term (network term) of the objective function. The method finds, given a current solution (v, g), a descent direction $(z - v, w - g)$ by solving a doubly constrained trip distribution model (whereas the Frank-Wolfe algorithm solves a transportation problem of linear programming). The algorithm involves two solutions at each iteration: the main problem solution, and the subproblem solution for determining the direction of descent for an improved main problem solution. The Evans algorithm is described by the following steps:

- Step 0: *Initialization.* Choose an initial solution for link flows, $v_l^0 = 0$, and demand, $g_{ij}^0 = 1$. Set the counter, $k := 0$.
- Step 1: *Update link cost.* s^k: $= s(v^{k-1})$, $k := k + 1$; and compute minimum cost routes c_{ij}^k, on the basis of updated link costs, for every O-D pair (i,j).
- Step 2: *Find the descent direction.*
  - –For demand term: Solve a doubly constrained gravity model as a function of the shortest route costs, w_{ij}^k: $w_{ij}^k = A_i^k O_i B_j^k D_j \exp(-\beta c_{ij}^k)$, applying the two-dimensional balancing method;
  - –For network term: z^k: Perform an all-or-nothing assignment of demand w_{ij}^k to the shortest routes computed with the updated link costs s^k.
- Step 3: *Compute the optimal step size.* Conduct a line search to find what linear combination of demand and link flows minimizes the objective function; that is, find λ^k, $0 \le \lambda^k \le 1$ that minimizes $f(\lambda) = f[v_l^{k-1} + \lambda(z_l^k - v_l^{k-1}); g_{ij}^{k-1} + \lambda(w_{ij}^k - g_{ij}^{k-1})]$.
- Step 4: *Update link flows and demand.* Update the link flows and the demand solution with the best linear combination of solutions from the current and previous iterations, that is, $v_l^k := v_l^{k-1} + \lambda(z_l^k - v_l^{k-1})$ for every link, and $g_{ij}^k := g_{ij}^{k-1} + \lambda(w_{ij}^k - g_{ij}^{k-1})$ for each pair (i, j).
- Step 5: *Convergence check.* If an appropriate convergence criterion is satisfied then stop; otherwise go to Step 1.

Preliminary Considerations

The first task is to build in the same directory as the EMME/2 system a file of the link cost functions of the network where, instead of the usual link flows, the initial solution v_l^0 is read. To be more specific, consider a typical link cost function used in EMME/2 automobile assignment. It is the usual Bureau of Public Roads function

$$s_l(v_l) = s_0\left[1 + \alpha_1\left(\frac{v_l}{k_l}\right)^{\alpha_2}\right] \quad (1)$$

where

α_1, α_2 = parameters calibrated from a previous study (typically the values are 0.15 and 4, respectively);
s_0 = free-flow travel time stored in EMME/2 link attribute length;
v_l = autoflow on link l stored in EMME/2 link attribute *volau*;
$s_l(v_l)$ = travel time on link l, an increasing function of autoflow on same link v_l; and
k_l = capacity on link l determined by assumed level of service and stored in EMME/2 link attribute lanes.

It is worth noting that the labels of the EMME/2 link attributes used to store the different arguments of the BPR function need not be taken literally. For example, the label length does not mean the link length in this application. The same is true for the field labeled lanes.

By default, when EMME/2 computes an automobile assignment it reads the link cost functions with flows different from 0 (from some previously performed assignment). To compute an initial solution for the link flows in the initialization step of the algorithm, however, these flows need to be replaced with zero link flows. This can be done by using a text editor to replace the attribute *volau* by *ul1* (which has been previously initialized to 0 in the volume-delay functions stored in the function file. Then by saving the edited file as a separate function file (here saved as *d411.ul1*), it can be read as needed. Thus, the link flows are first initialized to 0, whereas at subsequent iterations *ul1* always stores the current flows.

Step 0: Initialization

The iteration number, controlled by register **x,** is set to 0. Then an initial solution for the demand matrix (g_{ij}^0) and the link flow vector (v_l^0) is computed. The production and attraction vectors (O_i), (D_j), respectively, from an observed matrix are then computed (the observed automobile demand for Winnipeg in 1976 is used, in matrix **mf1**). A zero demand to be used later in the computation of the step size is also computed. Finally, the tolerance level of the secant root-finding method (explained later in Step 3) is saved in a scalar. These operations are summarized in Table 1.

Although the order of the modules employed does not matter from a modeling perspective, it is more efficient to do as many computations as possible in one module before starting to employ the next one. Because there will be many matrixes and scalars involved in the computations, it is a good idea to plan in advance where to store different results. It has been convenient to use the ability of EMME/2 to store full matrixes, O-D vectors, and scalars as **mf"name," mo"name," md"name," ms"name,"** respectively, where **name** is the name of the operand.

Step 1: Link Costs Update and Computation of Minimum Cost Routes

The iteration number x is increased by 1, the link costs vector s^k is updated as $s^k := s(v^{k-1})$, and the matrix of minimum cost routes (c_{ij}^k) is computed for every O-D pair (i, j). To update the link costs the link flows are initialized v^0 to those produced by assigning the demand (g_{ij}^0); otherwise, zero link flows would be used in the update of the link costs. The minimum cost routes (c_{ij}^k) result from an all-or-nothing assignment of the demand ms1 $=(g_{ij}^0) = 1$ and are saved in matrix **mf"cijk."** These operations are summarized in Table 2.

Step 2: Computation of Descent Direction

The descent direction for the demand term (w_{ij}) is computed by balancing the matrix $\{\exp(-\beta c_{ij}^k)\}$ to the marginal constraints **mo8,**

TABLE 1 Implementation of Step 0

Module	Purpose	Saved in
3.21	initialize demand to one	ms"gij0"
2.41	initialize link flows to zero	ul1
3.21	compute productions from mf1	mo"produc"
3.21	compute attractions from mf1	md"attrac"
3.21	compute zero demand	ms"zero"
3.21	compute tolerance 10^{-3}	ms"larnacc"

TABLE 2 Implementation of Step 1

Module	Purpose	Saved in
4.11	read link costs based on zero link flows	d411.ul1
5.11	all-or-nothing assignment of ms"gij0"	
5.21	perform the assignment	
	save the shortest routes	mf"cijk"
2.41	link flows from volau	ul1

md8 computed in Step 0. This computation of the doubly constrained gravity model is done by applying the two-dimensional balancing method. To compute the direction of descent for the network term, an all-or-nothing assignment of the demand (w_{ij}) is performed.

It is important to note that the dispersion parameter β is held constant during the solution of the model. To obtain a reasonable value for it, β was set equal to the inverse of the observed mean travel time in the network. For the Winnipeg network (in the demonstration data bank), β was set equal to 0.06.

Finally and only in the first iteration, the demand was initialized (g_{ij}^0) to (w_{ij}^0) (otherwise, in each macro iteration, **ms1** = (g_{ij}^0) = 1 to compute the minimum cost routes) and the main problem link flows v_l to subproblem link flows z_l would be used. These operations are summarized in Table 3.

Step 3: Computation of Optimal Step Size

In each iteration of the algorithm the optimal step size λ^* is obtained by performing a one-dimensional search of the objective function along the feasible direction $\{(z_l - v_l^k), (w_{ij} - g_{ij}^k)\}$. This is done by solving the following problem:

$$\min_{\lambda} f(\lambda) = f_l(\lambda) + f_{ij}(\lambda) = \sum_{l \in L} \int_{v_l}^{v_l + \lambda(z_l - v_l)} s_l(x)dx + \frac{1}{\beta} \sum_i \sum_j [g_{ij}^k + \lambda(w_{ij} - g_{ij}^k)] \ln[g_{ij}^k + \lambda(w_{ij} - g_{ij}^k)] \tag{2}$$

An efficient method of solving Equation 2 is to find the value of λ, which equates the gradient $f'(\lambda)$ to 0, where

$$f'(\lambda) = f_l'(\lambda) + f_{ij}'(\lambda) = \sum_{l \in L} s_l[v_l + \lambda(z_l - v_l)](z_l - v_l) + \frac{1}{\beta} \sum_i \sum_j \ln[g_{ij}^k + \lambda(w_{ij} - g_{ij}^k)](w_{ij} - g_{ij}^k) \tag{3}$$

To find the 0 of the gradient function, a variation of the secant root-finding method is used. The secant method involves approximating the tangent by a secant through the two most recent iterates and using the 0 of this line as the next iterate. In this particular implementation, one end of the current bracketing interval remains fixed.

An efficient way to compute the gradient of the network term suggested by Heinz Spiess is now presented. The idea consists of using the EMME/2 equilibrium algorithm to compute the new link costs instead of the network calculator. The implementation is described next.

The two λ-values that bracket the search interval are initialized between 0 and 1 and saved in scalars **ms"lam1," ms"lam2"** (in this implementation, **ms61, ms62,** respectively). Note that **ms"lam2"** will also hold the current upper bound of the search interval. Using Module 2.41 the current main problem and subproblem link flows (saved in *ul*1 and *volau,* respectively) are copied to link attributes *ul*3, *ul*2, respectively, of a dummy scenario, say 3000 (created before the macro execution in this case). Their linear combination, *ul*3 + *%msy%* × (*ul*2 − *ul*3) for every λ is computed using Module 2.41 and saved in *ul*1 in the dummy scenario. The need to save them in *ul*1 comes from the definition of the link cost functions in Module 4.11 where the link flows are saved in *ul*1.

To compute the costs of those combined flows, an all-or-nothing assignment is performed in the dummy scenario with zero demand. The link costs based on the link flows in *ul*1 are saved by default in the link attribute *timau.* When the line search begins (in each iteration of the macro) register **y**, which controls which of the two λ-values is read, is set to $y = 61$, whereas register **z,** which keeps track of the respective gradient values for the network term, is set to $z = 71$.

The gradient of the network term in the dummy scenario is finally computed after the evaluation (in Module 2.41) of the sum of the expression 0.06 * *timau* * (*ul*2 − *ul*3). The sum is saved in scalar *ms%z%*. The multiplication by $\beta = 0.06$ is equivalent to multiplying the demand term of the gradient by $1/\beta$. In this manner, scalar **ms71** contains the sum of the gradient values for the network term with respect to the first λ (in **ms"lam1"**), and scalar **ms72** contains the sum of the gradient values for the network term with respect to the second λ (in **ms"lam2"**).

Back in the working scenario, what remains to be computed is the gradient for the demand term. Remembering that the main problem demand is saved in **mf"gijk"** and the subproblem demand in **mf"wij,"** the last task involves the computation of the expression

$$\sum_i \sum_j \ln(mf"gijk" + \%msy\% * put[mf"wij" - mf"gijk")] * get(1) \tag{4}$$

where the special functions **get** (.) and **put**(.) are used to save some computation time (*9,* pp. 3–67). The summation over all origins and destinations in the matrix calculations of Module 3.21 gives the gradient for the demand term that is saved in scalar **ms"gradem."** It is worth noting that care is taken to avoid evaluating the last expression for zero values (because the logarithm of 0 is not defined). This can be accomplished in module 3.21 by providing a constrained matrix and a constrained interval. In this case using the default-

TABLE 3 Implementation of Step 2

Module	Purpose	Saved in
3.21	compute a function of shortest route costs as exponential	mf"ecijk"
3.22	balance mf"ecijk" to mo"produc" and md"attrac"	mf"wij"
5.11	all-or-nothing assignment of mf"wij"	
5.21	perform the assignment	
3.21	iteration 1: $g_{ij}^k = w_{ij}$	mf"gijk"
2.41	iteration 1: $v_l^k = z_l$.	ul1

constrained interval (0, 0, exclude), only nonzero values are retained. The same is done whenever the logarithm of the demand matrix is involved in the computations.

So far, for each λ the gradients for both the network and the demand terms have been computed. The total gradient is then saved in scalar ms%z% as %msz% + **ms"gradem."** For example, for the first λ in **ms"lam1"** the total gradient is saved in **ms71** containing the sum of the contents of scalar **ms71** (the gradient for the network term) and the contents of scalar ms"gradem" (the gradient of the demand term). This procedure is repeated once more for the second λ-value. Note that if the gradient is found to be positive the optimal step size is set to 0 and the Evans algorithm terminates because the current solution is optimal; however, in real problems such a result never occurs because the optimal solution is never reached.

The next task is to compute the slope of the secant line. In particular, using Module 3.21 the slope ϕ of the line $\{[\lambda_{k-1}, \nabla(\lambda_{k-1})], [\lambda_k, \nabla(\lambda_k)]\}$ is computed and saved in scalar **ms"phil12"** as

$$ms\text{"phil2"} = (ms72 - ms71)/(ms62 - ms61) \quad (5)$$

The optimal λ, saved in scalar **ms"xlopt"** (in ms70 here), is next computed from the formula

$$ms\text{"xlopt"} = (0 - ms71)/ms\text{"phil2"} \quad (6)$$

Finally, the convergence of the secant loop is monitored as follows. First, the following expression is evaluated:

$$1 \times \left\{ \frac{abs(ms\text{"xlopt"} - ms\text{"xlamn2"})}{abs(ms\text{"xlopt"})} - ms\text{"lamacc"} \right\} \leq 10^{-6} \quad (7)$$

Equation 7 is a boolean expression with values of 1 for true and 0 for false. If the result of this evaluation is 1, then the secant loop has converged. A value of **ms"lamacc"** $= 10^{-3}$ is used as a stopping criterion. The optimal step size taken from the last secant iteration is then used in Step 4 to update the current solution of the combined model. If the result of the above expression is not 1, the secant loop is repeated. Table 4 summarizes these operations.

Step 4: Current Solution Update

The main problem demand solution (from the previous iteration), currently in matrix **mf"gijk,"** is saved in matrix **mf"gijk − 1."** The current solution for demand is then a weighted average (optimal weight) of the previous (main) problem solution and the current (subproblem) solution as follows:

$$mf\text{"gijk"} = mf\text{"gijk} - 1\text{"} + ms\text{"xlopt"} \times (mf\text{"wij"} - mf\text{"gijk} - 1\text{"}) \quad (8)$$

The same is done for the link flows. The main problem link flows solution (from the previous iteration), currently in link attribute *ul1,* is saved in link attribute *ul2.* The current solution for the link flows is then a weighted average (optimal weight) of the previous (main) problem solution and the current (subproblem) solution, as follows:

$$ul1 = ul1 + \%ms\text{"xlopt"}\% * (volau - ul1) \quad (9)$$

Table 5 summarizes the above operations. Note that in Module 2.41 the matrix operations cannot be performed. Therefore, the optimal λ has to be represented not as the scalar **ms"xlopt"** but rather as the contents of the scalar **ms"xlopt."**

Step 5: Criteria for Convergence

To monitor the convergence rate of the algorithm with respect to the solution for demand, the maximum over all terms (origins and destinations) of the absolute deviations between the current solution and the solution from the previous iteration is considered and saved in scalar ms"gdif"; that is,

$$\max_{i \in I, j \in J} \|g_{ij}^k - g_{ij}^{k-1}\| = \max_{i \in I, j \in J} \|mf\text{"}gijk - mf\text{"}gijk - 1\text{"}\| \quad (10)$$

The convergence of the link flows is monitored, similarly, by computing the maximum over all links of the absolute deviations between the current solution and the solution from the previous iteration and saved in scalar **ms"vdif"**; that is,

$$\max_{l \in L} \|v_l^k - v_l^{k-1}\| = \max_{l \in L} \|ul1 - ul2\| \quad (11)$$

Another convergence criterion that is strongly recommended is the current value of the GAP function. At each iteration of the Evans algorithm, the subproblem solution provides a lower bound for the objective function value. That is, the current GAP is the distance from the current value of the objective function to the lower bound. The current value of the GAP function for the combined (OD-UE) model at iteration *k,* which is simply the value of 3 corresponding to λ = 0, is

$$GAP^k = LB^k - f^k(v, g) = \sum_{l \in L} s_l(v_l^k)(z_l - v_l^k) + \frac{1}{\beta} \sum_i \sum_j \ln(g_{ij}^k)(w_{ij} - g_{ij}^k) \quad (12)$$

where $f^k(v, g)$ is the current value of the objective function. The GAP function converges to 0, although not monotonically.

An alternative convergence criterion that may also serve as a condition for the termination of the algorithm is to test at each iteration a "modified" relative gap for the network flows and the demand because

TABLE 4 Implementation of Step 3

Module.	Purpose	Saved in
3.21	$\lambda_1 = 0$	ms"lam1"
3.21	current upper bound of search interval: λ_2	ms"lam2"
2.41	gradient of network term for λ_1, λ_2	ms71,ms72
3.21	gradient of demand term (each λ)	ms"gradem"
3.21	total gradient for each λ	ms71,ms72
3.21	the slope of the secant line	ms"phi12"
3.21	optimal λ	ms"xlopt"

TABLE 5 Implementation of Step 4

Module	Purpose	Saved in
3.21	save demand from previous iteration	mf"gijk-1"
3.21	update demand solution	mf"gijk"
2.41	save link flows from previous iteration	ul2
2.41	update link flows solution	ul1

both of them converge to their equilibrium values. For the network flows the "modified" relative gap RG_l^k at iteration k is defined as

$$RG_l^k = \frac{\sum_{l \in L} s_l(v_l^k)\,(z_l - v_l^k)}{\sum_{l \in L} s_l(v_l^k) v_l^k} \tag{13}$$

while, for the demand terms, the "modified" relative gap RG_{ij}^k at iteration k is defined as

$$RG_{ij}^k = \frac{\sum_i \sum_j \ln(g_{ij}^k)(w_{ij} - g_{ij}^k)}{\sum_i \sum_j \ln(g_{ij}^k) g_{ij}^k} \tag{14}$$

Eventually as $\sum_{l \in L} s_l(v_l^k) z_l \rightarrow \sum_{l \in L} s_l(v_l^k) v_l^k$ and $w_{ij} \rightarrow g_{ij}^k$, and all the demand is on shortest routes, both these measures go to 0. However, they are not decreasing monotonically (which is also true for the relative gap in the fixed demand user equilibrium traffic assignment). Table 6 summarizes these operations.

COMPARISON BETWEEN THE COMBINED MODEL AND THE SEQUENTIAL PROCEDURE

The results presented below were obtained by solving the combined model for the city of Winnipeg, Manitoba, Canada. The network consists of 154-zone centroids, 903 regular nodes, and 2,535 automobile links. The computations were performed in a SUN SPARC-2 workstation with 64MB of memory; the macro needs about 46 sec (real time) or almost 19 sec (central processing unit time) time per iteration. The observed (1976) automobile demand in **mf1** was increased by 50 percent because the network was not very congested.

As presently implemented, the macro iterates until some prespecified convergence criterion for the link flows is satisfied. It is straightforward to apply any of the stopping criteria suggested earlier. The various performance measures from the application of the macro to the Winnipeg network are indicated in Figure 1. The rates of convergence of the optimal step size, demand, "modified" relative GAP for demand RG_{ij}, link flows, and the GAP function are satisfactory. Although it is not expected that more than 10 or 20 iterations are required in practice, the results from additional iterations provide information about the convergence of the Evans algorithm.

Further evidence of the quality of the results can be seen in a link scattergram in Figure 2. In the absence of observed link flow data the macro for 10 iterations has been solved and the obtained link flows in *ul3* (horizontal axis) have been saved. Then a trip distribution model was estimated on the basis of free-flow travel times and balanced to the production and attraction totals of the observed automobile demand matrix in **mf1** increased by 50 percent. The estimated trips were then assigned to the network for 10 iterations and the link flows obtained were saved in *ul1* (vertical axis). The plot shows that the link flows from the combined model are lower (better converged) than from a trip table based on free-flow travel times. If the two methods were equivalent, the points would lie on the line shown in the figure. Because the points lie above the line, the link flows from the four-step procedure are higher, which results from the longer trips on the basis of free-flow travel times.

In addition to the plot, a number of statistics for both variables in the combined model (automobile link flows and automobile O-D flows) have been computed. The purpose here is to compare two pairs of variables: first, the trip table estimated from free-flow travel times with the trip table from the combined model (after 10 iterations); and second, the link flows after the assignment of the estimated trip table for 10 iterations with those obtained from the solution of the combined model. The root mean square error (*RMSE*) and the χ^2 statistics reported in Table 7 are based on the following formulas:

$$RMSE = \left\{ \frac{\sum_{i=1}^{m} (M_i - T_i)^2}{m} \right\}^{0.5} \tag{15}$$

$$\chi^2 = \sum_{i=1}^{m} \left\{ \frac{(M_i - T_i)^2}{T_i} \right\} \tag{16}$$

where

T = solution from combined model,
M = solution from sequential procedure, and
m = number of data elements with positive values.

Zero values in the solutions were removed because these values are a property of the model formulation or the data, rather than the solution method.

FURTHER CONSIDERATIONS

Finally, a number of possible improvements and extensions of the initial formulation and implementation of the OD-UE model are considered. In this implementation of the OD-UE model, car drivers seek to minimize their travel time. Obviously, travel time is just one component of the travel cost. Other components may include monetary costs incurred by owning (for example, depreciation costs) and operating a car (insurance costs, fuel costs, parking costs, tolls,

TABLE 6 Implementation of Step 5

Module	Purpose	Saved in
3.21	maximum absolute difference of mf"gijk" and mf"gijk-1"	ms"gdif"
2.41	maximum absolute difference of ul1 and ul2	ms"vdif"
3.21	"modified" relative GAP for demand	ms"rgdem"
3.21	current GAP	ms"gap"

TABLE 7 Comparison Between Combined Model and Sequential Procedure

Variable	Positive Flows	RMSE	Desired	χ^2	Desired
Auto Link Flows	2,366	119.41	0	84054.31	0
Auto O-D Flows	18,630	2.35	0	4104.19	0

etc.). Consideration of these additional costs requires changes in the model formulation and, of course, data availability.

The model formulation considered here assumes one person per car. However, car occupancy data by origin zone are immediately available in the demonstration data bank of EMME/2 and could have been used in the macro. In addition, the model formulation can be modified to accommodate the occupancy factor endogenously. This idea may be applied when the model is used to assess air quality impacts from relevant policy interventions.

The combined model can be enhanced to include other modes of travel. This involves reformulating the combined model to account for mode choice and is being pursued in the application of the macro in a sketch planning network for Chicago [Boyce et al. (*10*)]. Finally, it is hoped that transit operations can be integrated into the macro and that the impacts of changing parameters such as waiting time, loading time, and headway can be studied.

Although all the above extensions and improvements are possible and interesting to study, it is not known how they will affect the per-

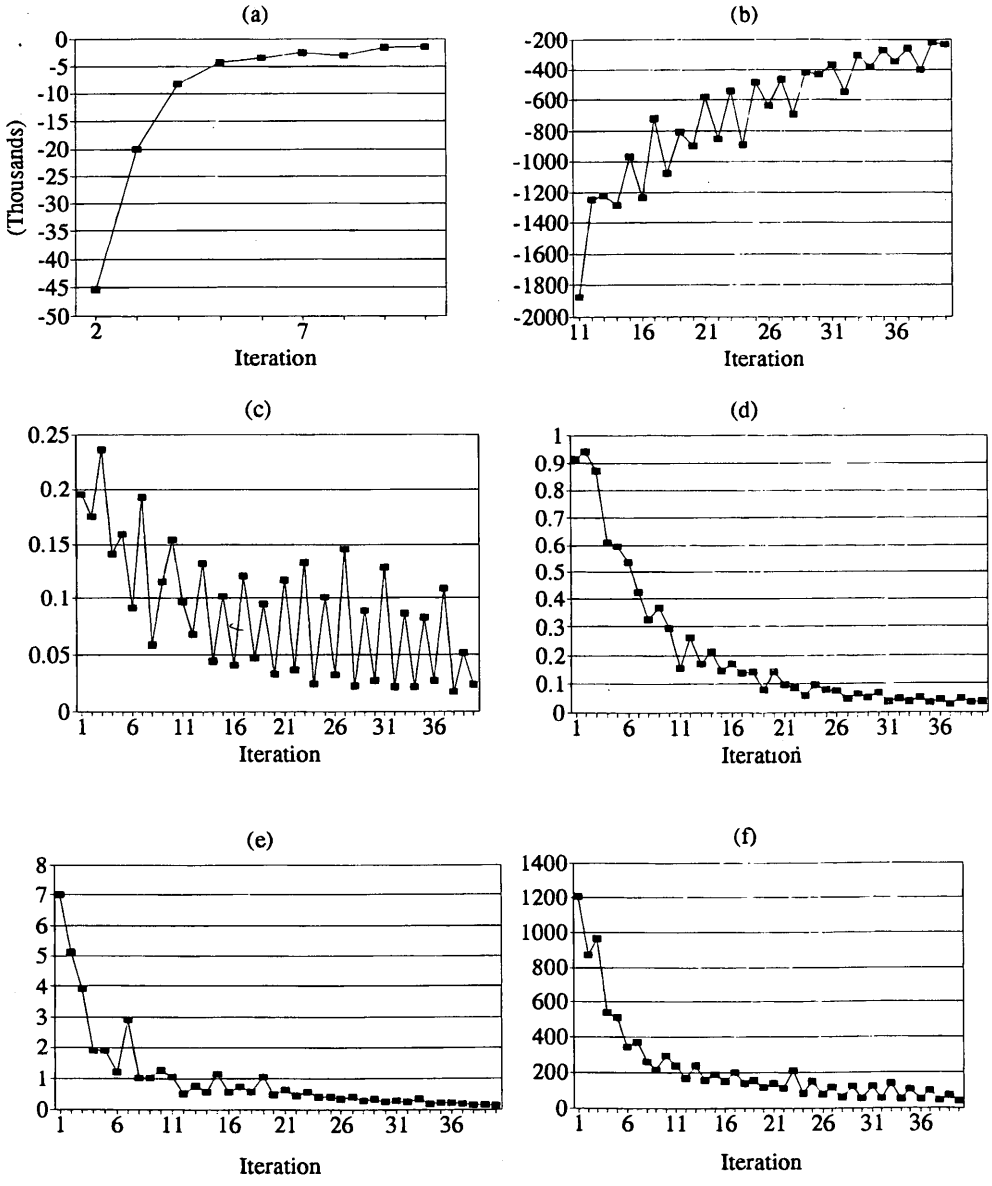

FIGURE 1 Monitoring the convergence of the Evans algorithm: (*a*) GAP function (Iterations 2–10); (*b*) GAP function (Iterations 11–40); (*c*) optimal step size; (*d*) modified relative GAP for demand; (*e*) maximum absolute deviations for demand; and (*f*) maximum absolute deviations for link flows.

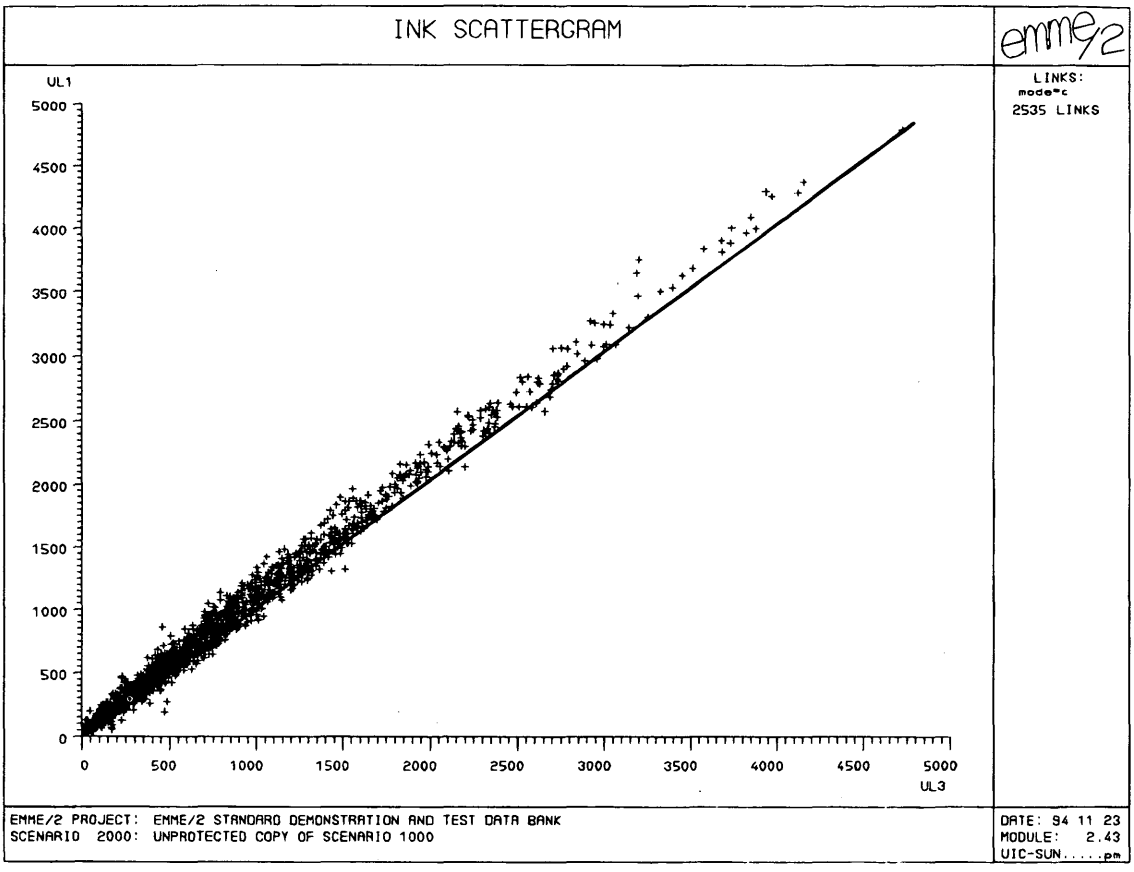

FIGURE 2 Link flows solution of sequential procedure versus link flows solution of combined model.

formance of the macro with respect to the computer requirements. If they can add to the detail in representing travel behavior without overburdening the computational effort, then such a modified macro can be seen as a powerful planning tool. Meanwhile, the implementation of a combined model in EMME/2 can meet some of the modeling requirements arising from modern urban transportation planning practice and motivate transportation professionals to use more sound planning methods. The quality of the results obtained to date seems to encourage the use of the macro in planning studies.

ACKNOWLEDGMENTS

The research presented in this paper began with a visit by the first author to the office of INRO Consultants in Montreal, with the support of the University of Illinois at Chicago. The authors also are indebted to three anonymous referees, whose comments greatly improved the quality of the paper.

REFERENCES

1. Boyce, D. E., Y.-F. Zhang, and M. R. Lupa. Introducing "Feedback" into Four-Step Travel Forecasting Procedure Versus the Equilibrium Solution of Combined Model. In *Transportation Research Record 1443,* TRB, National Research Council, Washington, D.C., 1994, pp. 65–74.
2. Florian, M. Nonlinear Cost Network Models in Transportation Analysis. *Mathematical Programming Study,* Vol. 26, 1986, pp. 167–196.
3. Patriksson, M. *The Traffic Assignment Problem: Models and Methods.* VSP BV, Utrecht, The Netherlands, 1994.
4. Evans, S. P. Derivation and Analysis of Some Models for Combining Trip Distribution and Assignment. *Transportation Research,* Vol. 10, 1976, pp. 37–57.
5. Frank, C. *A Study of Alternative Approaches to Combined Trip Distribution-Assignment Modelling.* Ph.D. thesis. Department of Regional Science. University of Pennsylvania, Philadelphia, 1978.
6. LeBlanc, L. J., and K. Farhangian. Efficient Algorithms for Solving Elastic Demand Traffic Assignment Problems and Mode Split Assignment Problems. *Transportation Science,* Vol. 15, 1981, pp. 306–312.
7. Boyce, D. E., L. J. LeBlanc, and K. S. Chon. Network Equilibrium Models of Urban Location and Travel Choices: A Retrospective Survey. *Journal of Regional Science,* Vol. 28, 1988, pp. 159–183.
8. Chu, Y.-L. Combined Trip Distribution and Assignment Model Incorporating Captive Travel Behavior. In *Transportation Research Record 1285,* TRB, National Research Council, Washington, D.C., 1990, pp. 70–77.
9. *EMME/2 User's Manual: Software Release 7.* INRO Consultants, Inc., Montreal, Quebec, Canada, 1994.
10. Boyce, D. E, M. Tatineni, and Y.-F. Zhang. *Scenario Analyses for the Chicago Region with a Sketch Planning Model of Origin-Destination Mode and Route Choice.* Final Report to Illinois Department of Transportation. Urban Transportation Center, University of Illinois, Chicago, 1992.

Publication of this paper sponsored by Committee on Passenger Travel Demand Forecasting.