# Balancing Link Counts at Nodes Using a Variety of Criteria: An Application in Local Area Traffic Assignment

REFAT BARBOUR AND JON D. FRICKER

**A study of the impact of a major change in a campus street network began with the collection of link flows before the change. Despite the care with which the link counts were made, conservation of flow at each node was not satisfied. Therefore, the flows through the nodes had to be "balanced." This paper discusses the variety of techniques developed to balance the network. The techniques fell into two categories: algorithms and mathematical programming formulations. A comparison was made between these procedures and the maximum-likelihood method advocated in the literature. It became evident that the node-balancing solution depends on the criteria chosen to evaluate the solution, which in turn can offer guidance as to the specific method to choose or develop.**

Recently, the street network in the northeast portion of the Purdue University campus underwent a major change. The main entrance to the university was permanently closed to permit construction of a new academic building. Before this change took place, link flows in this portion of campus and on the urban streets immediately adjacent to it (a study area hereafter referred to as "Campus NE") were observed and recorded during the afternoon peak hour. The intent was to provide the basis for a forecast of the link flows after the network change and thereby identify potential traffic bottlenecks. Therefore, all street facilities used by vehicles in the area of interest were represented by links in the network abstraction of Campus NE. Because of this level of detail, and because the study area was less than 1 mi$^2$ in size, we use the phrase "local area traffic assignment" to distinguish our activity from that of the traditional city- or regionwide travel demand modeling process. In fact, our work could be considered a type of site impact analysis, although our initial emphasis was on route choice behavior, with signal timing confined to a subsequent phase of the project.

## NODE-BALANCING ALGORITHMS

Despite the data collectors' best efforts to be accurate in recording the link flows and turning movements they observed, when the information was put into the link-node model of the network, it was clear that the recorded flows at most of the intersections violated the conservation of flow requirement, which is that the sum of flows in equals the sum of flows out. In other words, these intersection nodes were "unbalanced" with respect to their recorded flows.

Since the origin–destination (O-D) table estimation and traffic assignment models available to us required balanced link counts, we had to improve the traffic count data to restore conservation of flow at all nodes (*1, 2*). We found the node-balancing (NB) method presented previously (*1*) to be the principal alternative to a manual trial-and-error adjustment of link flows. This method assumes that observed flows are Poisson distributed and then employs a maximum-likelihood method (MLM) to find the most likely set of link flows from the many possible solutions. Although we accepted the idea behind the MLM, we believed that if we were going to write any computer code, we would be more comfortable trying to apply some familiar network algorithms to adjust the unbalanced link flows than trying to convert the ideas described elsewhere (*1*) into FORTRAN. We were also curious about the impact of various objectives or solution criteria on the solution itself. In the next section we report on the evolution of the NB algorithms we developed. In later sections, we present a set of optimization procedures and some comparative evaluations.

### Method NB1: Automated Trial and Error

Method NB1 automates a form of the trial-and-error method that we might have used manually. It was encoded to provide a basis of comparison against what we anticipated would be more sophisticated methods. In the steps below, $V(\text{in})$ and $V(\text{out})$ are the inflow and outflow rates at an unbalanced node $u$, and $I(u)$ is the amount of the imbalance at a node $u$, $I(u) = V(\text{in}) - V(\text{out})$. At any iteration $k > 1$, node $u$ is considered to be approximately balanced if the absolute value of $I(u)$ is either (a) less than or equal to 1 or (b) within 1 percent of $0.5 * [V(\text{in}) + V(\text{out})]$.

#### *Method NB1*

Step 0. $k = 0$.

Step 1. $k = k + 1$. Identify all nodes $j$ in the network that are not origin or destination centroids but have unbalanced flows and place them in the set of unbalanced nodes ($U$) in

R. Barbour, Strand Associates, Inc., 910 West Wingra Drive, Madison, Wis. 53715. J. D. Fricker, School of Civil Engineering, Purdue University, West Lafayette, Ind. 47907.

order of their original node numbers. If all nodes are balanced, go to Step 4.

Step 2. If set $U$ is empty, go to Step 1. Otherwise remove the first node in set $U$ and call it the "$u$-node."

Step 3. (a) If $I(u) > 0$, decrease each inflow by $p(i,u) * 0.5 * I(u)$, where $p(i,u)$ is the proportion of all inflows that enter node $u$ via link $(i,u)$. For example, if $I(14) = +36$, and node 14 receives 25 percent of its inflow from link (4,14), then $V(4,14)$ will be reduced by $0.25 * 0.5 * 36 = 4.5$ vehicles. Likewise, each outflow will be increased by $p(u,i) * 0.5 * I(u)$, where $p(u,i)$ is the proportion of outflows that depart node $u$ via link $(u,i)$. (b) If $I(u) < 0$, add $-p(i,u) * 0.5 * I(u)$ to each inflow link $(i,u)$ and subtract $-p(u,i) * 0.5 * I(u)$ from each outflow link $(u,i)$. [Note: the minus sign before "$p$" is necessary because $I(u) < 0$.] (c) If any of these flow adjustments would cause a link flow to become negative, leave that link's flow unchanged and redefine $p(u,i)$ or $p(i,u)$ among the remaining links involved. (d) Go to Step 2.

Step 4. (a) Identify those noncentroid nodes $u$ that are approximately (but not exactly) balanced. (b) If $I(u) > 0$, find the centroid $Z$ nearest $u$. If $Z$ is an origin centroid, subtract $I(u)$ from each link on the shortest path between $Z$ and $u$. If $Z$ is a destination centroid, add $I(u)$ to each link between $u$ and $Z$. (c) If $I(u) < 0$, find the centroid $Z$ nearest $u$. If $Z$ is an origin centroid, add $-I(u)$ to each link between $Z$ and $u$. If $Z$ is a destination centroid, subtract $-I(u)$ from each link between $u$ and $Z$. (d) Stop.

### *Discussion of Method NB1*

Method NB1 is admittedly crude, but it is fairly easy to program. The relaxed definition of "balanced" in Step 1 after iteration $k = 1$ is a recognition that exact balance for all nodes may never result from this method. Our experience indicates that, after about 20 iterations, all nodes are in approximate balance and further iterations are of little value. Therefore, Step 4—a housecleaning step—is used to avoid the creation of minicentroids by making very minor changes to the $T(i)$ and $T(j)$ values we have collected at the parking facilities and the study area boundaries.

## Method NB2: Minimum-Weight Paths

For the minimum-weight path method, we introduce link weights defined in terms of the difference between the original observed link flows $V_o$ and the improved flows $V_b$ that exist on a link during the NB process. For each link $(i,j)$,

$$d = \frac{|V_o - V_b|}{V_o} + E_o \tag{1}$$

where $E_o$ is a very small number, such as $1 \times 10^{-6}$. This second term in the expression is necessary to prevent $d = 0$ on all links at the start of the process and on any link not yet adjusted. As link flow adjustments take place, the first term begins to dominate the second.

### *Method NB2*

Step 1. Identify all nodes $j$ in the network that are not origin or destination centroids but have unbalanced flows and place them in the set of unbalanced nodes ($U$) in order of their original node numbers.

Step 2. If set $U$ is empty, stop. Otherwise, remove the first node in set $U$ and call it the "$u$-node."

Step 3. Calculate the link weights $d(i,j)$ for each link in the network using Equation 1.

Step 4. Using an appropriate shortest-path algorithm (*3*) and the link weights $d(i,j)$, find the minimum-weight paths from the current $u$-node to all centroids, treating all links as two-way links, regardless of their actual orientation.

Step 5. Identify the centroid $Z$ having the smallest path weight from the $u$-node. This path from $u$ to $Z$ has the least accumulated differences along it.

Step 6. Send one unit of flow along the path $(u,Z)$. This unit of flow will be positive or negative, depending on the sign of $I(u)$, the orientation of each link along the path, and whether the centroid $Z$ is an origin ($P$) or a destination ($A$) (see Table 1). Update $I(u)$ such that $|I(u)| = |I(u)| - 1$.

Step 7. If $I(u) = 0$, go to Step 2. Otherwise, go to Step 3.

TABLE 1 SIGN OF UNIT FLOW CHANGE FOR NB2 STEP 6

| $I(u)$ | $Z$ | Flow Change Along Path | Link Flow Change[a] |
|---|---|---|---|
| >0 | $P$ | 1 less from $Z$ to $u$ | $-1$[b] |
| | $A$ | 1 more from $u$ to $Z$ | $+1$[b] |
| <0 | $P$ | 1 more from $Z$ to $u$ | $+1$[b] |
| | $A$ | 1 less from $u$ to $Z$ | $-1$[b] |

[a]Change in flow on each link along path $(Z,u)$ or $(u,Z)$.
[b]If direction of link is opposite that of path flow change, sign of link flow change should be reversed.

An example implementation of Step 6 may be helpful at this point. Figure 1 shows the minimum-weight path from $u$ to the centroid $Z$ identified in Step 5. Let us say that $I(u) = +1$ and the centroid $Z$ is an origin ($P$) node. This path contains 2 two-way links, $(u,9)$ and $(7,Z)$, and 2 one-way links, (8,7) and (8,9). We do not know which of the four links incident to node $u$ has the faulty counts that caused $I(u)$ to be nonzero, so we will transfer this flow imbalance $I(u)$ to the nearest (in terms of link $d$-weights) centroid. Since $Z$ is a $P$-node in this illustration, Table 1 indicates that one unit of flow must be deducted from all links on this path from $Z$ to $u$, unless this direction violates a link orientation. Such a violation occurs for link (8,7), so flow on this link is increased by one unit in its only permitted direction.

In accordance with Table 1, we make the following adjustments to the link flows along the minimum-weight path in Figure 1: $V(Z,7) = V(Z,7) - 1$; $V(8,9) = V(8,9) - 1$; $V(9,u) = V(9,u) - 1$; but $V(8,7) = V(8,7) + 1$. At node 9, one less unit of flow is received from node 8, but one less flow unit is sent on to node $u$, so the previous value of $I(9)$ is preserved. At node 7, which is one end of the "backwards" link (8,7), one less flow unit is received from node $Z$, but one more unit is received from node 8, thereby preserving $I(7)$. Likewise, node 8 is preserved by sending one more unit to node 7, but one unit less to node 9. If $Z$ were a destination ($A$) centroid, the direction of flow adjustment would be reversed (see row 2, Table 1) and link (8,9) would be the backwards link. It would have its flow reduced by 1, whereas the "forward" links along the path from $u$ to $Z$ would have a flow change of $+1$. The reader is invited to verify that, in this case and for the cases of rows 3 and 4 of Table 1, the link flow changes along the minimum-weight path produce the desired

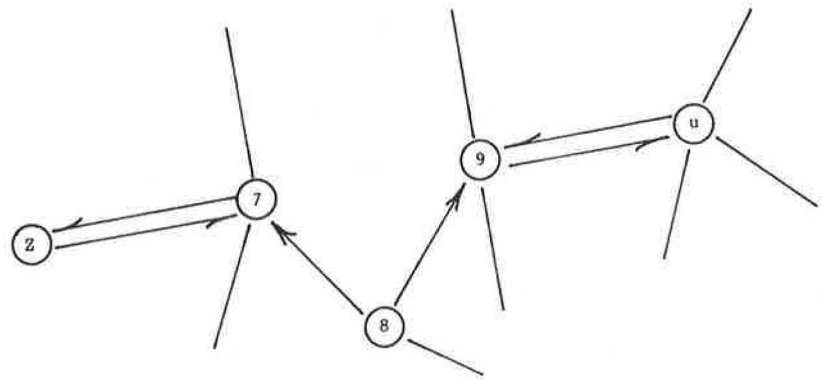


**FIGURE 1 Example of NB2 Step 6.**

results: $I(u)$ moves toward zero and $I(i)$ for all other nodes $i$ is unchanged.

### *Discussion of Method NB2*

We found several features of NB2 appealing. Once a $u$-node is balanced, it stays balanced. By sending flows through the network from the current $u$-node to a centroid in accordance with Step 6 (Table 1), every intervening node $i$—some of which may have already been balanced—has its $I(i)$ value unchanged. (See the case of node 9 in Figure 1 for the first example presented above.) Unlike NB1, where convergence is not guaranteed and the choice of the number of iterations can affect the outcome, NB2 "visits" each $u$-node only once, balances it, and moves on to the next $u$-node. We have devised an effective system for adjusting link volumes in a way that disturbs as few link counts as possible—and primarily those links with their $V_b$ values still close to their $V_o$ values. This would seem to bias the flow changes in a favorable way—toward smaller eventual network-wide error (goodness-of-fit) measure values.

A possible inefficiency in NB2 is its use of a unit flow adjustment. In cases where $I(u)$ could approach 100—any greater imbalance in our network would probably be due to a data collection or processing error—it might be wiser to use a larger flow adjustment. An adjustment of perhaps 0.5 * $I(u)$ could be used in at least the first several applications of Step 6. However, we did not adopt this procedure for two reasons:

1. A belief that too many vehicles might be sent along the smallest minimum-weight path, thereby distorting the link $d$-weights for the remainder of NB2 and precluding the best fit of $V_b$ versus $V_o$. Until an adequate investigation of the best fraction of $I(u)$ to send—and to what extent it may change from network to network—is carried out, we prefer the unit flow adjustment.
2. A desire to build from simplicity. The unit flow adjustment may be somewhat inefficient, but unless this potential flaw becomes a detriment in real applications, its current form appears suitable for comparison with other methods.

## Method NB3: Minimax Variation

The minimax variation method is designed to pay closer attention to the $d$-weights of certain individual links on the minimum-weight paths from the $u$-node to the various centroids. The idea behind NB3 is to change flows on links that have been changed relatively little earlier in the balancing process and avoid those links that have had relatively large changes. Method NB3 is a minor variation of NB2; only Step 5 is different.

### *Method NB3*

Steps 1–4. Same as those in Method NB2.

Step 5. (a) Find the link with the largest $d$-weight on each of the minimum-weight paths found in Step 4. Call these links "maxilinks." (b) Select the minimum-weight path having the maxilink with the smallest $d$-weight and call its associated centroid $Z$.

Steps 6 and 7. Same as those in Method NB2.

### *Discussion of Method NB3*

The selection rule in Step 5 can be best explained using an example. The minimum-weight paths from the $u$-node to three centroids are shown in Figure 2. Method NB2 would choose path ($u$,$z1$) at its Step 5, because that path's total weight (0.08) is smaller than 0.12 for path ($u$,$z2$) and 0.10 for path ($u$,$z3$). However, the maxilinks on these three paths have $d$-weights 0.08, 0.04, and 0.06, respectively. Thus, the second path, ($u$,$z2$), has the minimum maxilink and would be chosen by

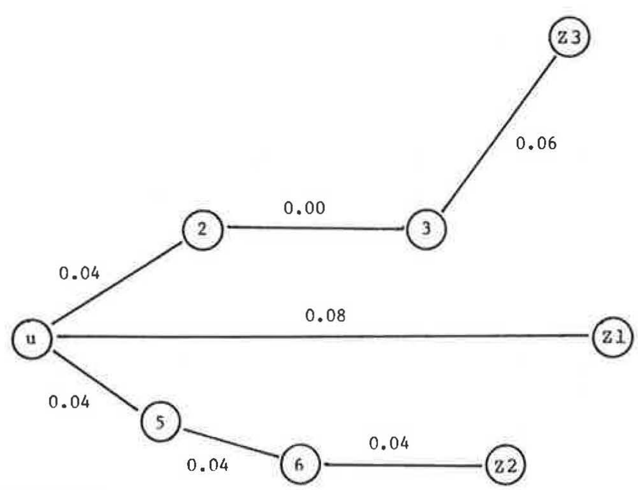


**FIGURE 2 NB3 Step 5.**

NB3 at its Step 5. Whereas Method NB2 would make changes to one link with $d$-weight 0.08, Method NB3 would cause changes on three links, each with $d$-weight 0.04 in this example. Until tests are conducted, it is not clear which method would be generally superior.

### Method NB4: Minimum $d$-Weight Links

The minimum $d$-weight link method extends the evolution begun with NB2 and continued with NB3. As in NB3, the links on the minimum $d$-weight paths found in Step 2 are modified to find a path between the current $u$-node and each centroid that has links with $d$-weights as small as possible. Whereas NB3 identifies a maxilink on each minimum $d$-weight path, NB4 is allowed to modify the minimum $d$-weight path itself if a link on that path can be replaced by a link having a smaller $d$-weight. This is accomplished through an adaptation of the "triple operation" used in Floyd's shortest-path algorithm (*3*). Floyd replaces the subpath $(i,k)$ with the subpath $(i,j,k)$ if $w(i,j) + w(j,k) < w(i,k)$, where $w(i,j)$ is the current smallest cost or distance from node $i$ to node $j$. Method NB4 replaces link $(i,k)$ on the minimum $d$-weight path with the subpath $(i,j,k)$ if $\text{Max}\{d(i,j),d(j,k)\} < d(i,k)$. This procedure modifies a good (namely, minimum $d$-weight) path to produce a path that is better with respect to a specific objective: to avoid involving links that already have large |percent $V_b$ − percent $V_o$| values in the balancing process.

Steps 1–4. Same as those in Methods NB2 and NB3.

Step 5. Along each minimum $d$-weight path from the current $u$-node to all centroids, replace a link $(i,k)$ with the subpath $(i,j,k)$ if $d(i,k) > \text{Max}\{d(i,j), d(j,k)\}$.

Steps 6 and 7. Same as those in Methods NB2 and NB3.

## COMPARATIVE ANALYSIS

As Methods NB1 to NB4 were being developed, a discussion appeared (*4*) of a computer program written in BASIC to implement the NB scheme outlined elsewhere (*1,5*). We welcomed the opportunity to test our four methods against something besides each other. We started with the example problems cited previously (*4,6*) and compared the results of our methods with those just mentioned.

### Measures of Merit

On what basis should the methods be compared? Clearly, the only frame of reference we have is the original link counts $V_o$. Although we need to balance the nodes to prepare the data for our traffic assignment model, we should do so by revising the $V_o$-values as little as necessary. Therefore, we considered several measures of merit for our comparative analysis.

1. Root Mean Square Error (RMSE). The RMSE standard measure for comparing a generated value with its target takes the form of

$$\text{RMSE} = \left|\frac{1}{n}\sum_{i,j}(V_b - V_o)^2\right|^{1/2}$$

This measure, of course, assesses disproportiontely larger penalties to greater differences between $V_b$ and $V_o$.

2. Mean of the absolute percent difference between $V_b$ and $V_o$ (mean abs % diff). For each link, the percent difference ($PD$) between $V_b$ and $V_o$ is calculated as

$$PD = \frac{V_o - V_b}{V_o} \times 100$$

which leads to

$$\text{mean abs \% diff} = \frac{1}{n}\sum_{i,j} |\,PD(i,j)\,|$$

This is a more intuitive measure of the differences occurring between $V_b$ and $V_o$ than is RMSE.

3. Maximum percent absolute difference (max % diff). This measure identifies the worst single match of $V_b$ to $V_o$ at the end of the NB procedure, based on the expression for $PD$ above.

4. Number of links with absolute difference greater than $X$ percent (links > $X$%). An NB method may produce one or two poor matches of $V_b$ to $V_o$ for measure 3 but for most links provides a good match. This measure offers a more specific description of this kind of behavior than the others. For the size of the networks we tested, we thought it best to set $X$ at approximately one-half the mean value of "max % diff" observed for the methods being tested.

5. Mean difference (mean diff). One might expect that little or no change in total link volumes would result from an NB procedure, but that is not the case. The flow adjustment process may have to add or deduct flows from links to restore conservation of flow at each node. Thus, the mean value of $(V_o - V_b)$ with the sign of this difference for each link retained is a rough indication of the change in vehicle miles of travel that accompanies the NB procedure.

6. Worst-case computational complexity. The MLM method was analyzed as having a worst-case complexity of $O(n^3)$. For NB1, it was $O(k*n)$, where $k$ equals the number of iterations needed to reach convergence. For NB2 and NB3, it was $O(W * n^2)$, where $W = \Sigma_u|I(u)|$. Finally, NB4 was evaluated at $O(W * n^3)$.

These measures have been introduced in order of their importance to us in assessing the performance of the various methods. The parenthetical abbreviations for the measures correspond to the column headings in Tables 2 and 3, which summarize our tests on the two sample problems.

### Discussion

In both problems, Methods NB3 and NB4 accomplish what they were designed to do—minimize the worst case (max % diff). However, the actual computer time for NB4 was considerably longer than that for the other methods, making it unlikely that NB4 would be practical on a microcomputer. Considering all the measures in Tables 2 and 3, Methods NB2 and NB3 performed moderately well on the smaller problem and not so well on the larger problem. What was surprising to us was the behavior of Method NB1, Automated Trial and Error, which did poorly on the small network but quite well

TABLE 2 COMPARATIVE RESULTS FOR SAMPLE PROBLEM 1

| Method | Measure of Merit (1)RMSE | (2)Mean Abs % Diff | (3)Max % Diff | (4)Links >9.5% | (5)Mean Diff |
|---|---|---|---|---|---|
| MLM | 38.5 | 10.7 | 20.0 | 8 | + 4.6 |
| NB1 | 54.8 | 14.1 | −30.2 | 10 | −14.7 |
| NB2 | 46.8 | 12.1 | −16.8 | 7 | + 5.9 |
| NB3 | 43.2 | 12.1 | −14.2 | 10 | + 4.9 |
| NB4 | 44.6 | 12.3 | −14.2 | 11 | + 7.0 |

NOTE: Six nodes (3 centroids), 12 links, mean $V_o$ = 337.8, $W$ = 631. See paper by Beagan (*4*).

TABLE 3 COMPARATIVE RESULTS FOR SAMPLE PROBLEM 2

| Method | Measure of Merit (1)RMSE | (2)Mean Abs % Diff | (3)Max % Diff | (4)Links >9.5% | (5)Mean Diff |
|---|---|---|---|---|---|
| MLM | 169.4 | 3.8 | − 8.8 | 10 | + 5.9 |
| NB1 | 169.7 | 3.7 | − 9.2 | 9 | + 0.03 |
| NB2 | 168.8 | 4.5 | −10.1 | 12 | −16.6 |
| NB3 | 178.8 | 4.6 | − 7.2 | 18 | +27.6 |
| NB4 | 186.6 | 4.3 | − 7.2 | 17 | +34.7 |

NOTE: Twenty-three nodes (5 centroids), 32 links, mean $V_o$ = 3303.7, $W$ = 2486. See presentation by Beagan (*6*).

TABLE 4 COMPARATIVE RESULTS FOR CAMPUS NE NETWORK

| Method | Measure of Merit (1)RMSE | (2)Mean Abs % Diff | (3)Max % Diff | (4)Links >10% | (5)Mean Diff | (7)Nodes with $I(u) \geq 1$ | (8)Max $I(u)$ |
|---|---|---|---|---|---|---|---|
| MLM | 37.3 | 8.2 | 64.0 | 24 | + 3.8 | 11 | +1, −1 |
| NB1(10) | 38.1 | 7.1 | − 59.1 | 20 | + 4.4 | 22 | −14 |
| NB1(100) | 68.1 | 13.2 | −143.1 | 36 | +27.4 | 24 | − 6 |
| NB2 | 48.7 | 13.8 | − 63.8 | 51 | + 1.7 | 0 | 0 |
| NB3 | 45.0 | 17.7 | 75.0 | 67 | + 5.2 | 0 | 0 |

NOTE: Fifty-four nodes (21 centroids), 128 links, mean $V_o$ = 291.69.

on the larger one. In fact, Method NB1 outperformed the MLM procedure (Table 3) in three of the first five measures.

Already some hypotheses can be formulated. These pertain to the influence of network size (number of nodes, links, and centroids), initial magnitude of $W = \Sigma_{\text{all } u} |I(u)|$, and mean link flow values ($V_o$) on the performance of an NB method. We applied the MLM method and Methods NB1 through NB3 to the network that required node balancing in the first place, Campus NE. The results are summarized in Table 4.

The format of Table 4 is a variation of that of Tables 2 and 3, brought about by the behavior of Method NB1 on Campus NE. As the number of iterations increases, the nodes become more nearly balanced. In the process, however, the difference between the improved link flows $V_b$ and the observed flows $V_o$ tends to increase. Thus, we list two versions of Method NB1: NB1(10) has gone through 10 iterations and NB1(100) has been carried through 100 iterations. Also, the measures of merit used in Tables 2 and 3 are augmented by two that reflect the true objective of the methods. These are

7. Number of nodes still unbalanced (nodes with $I(u) > 1$). We define "unbalanced" here as having an imbalance $I(u)$ of more than one vehicle.

8. Maximum absolute imbalance (max $|I(u)|$). This is thought to be an indication of the balancing work left undone and perhaps an early indication of cases in which convergence is impossible.

Comparing NB1(10) and NB1(100) without these new measures would lead us to conclude that 10 iterations are better than 100, but measures 7 and 8 indicate otherwise. These new measures also point out the superiority of methods NB2 and NB3. In one well-organized iteration, they produced a set of link flows in which each node had $I(u) = 0$. Furthermore, their first five measures are competitive with methods NB1(10) and NB1(100).

## MATHEMATICAL PROGRAMMING APPROACHES

Having acquired some experience with the criteria (measures of merit) by which NB methods might be evaluated, we began to think of each of these criteria as the basis for an NB method. The result was six mathematical programming (MP) formulations, identified as NB5 through NB10. Each MP formu-

TABLE 5 SUMMARY OF MP RESULTS FOR EXAMPLE 1

| | Measure of Merit | | | | | |
|---|---|---|---|---|---|---|
| Method | 8 | 3 | 5 | 9 | 10 | 2 |
| MLM | 58.0 | − 20.0 | 4.6 | 0 | 35.1 | 10.8 |
| NB3 | 78.0 | − 14.2 | 4.9 | 0 | 40.9 | 12.1 |
| NB5 | − 47.7 | 19.5 | 13.3 | 2.9 | 45.1 | 14.2 |
| NB6 | 78.1 | 14.0 | 14.8 | 2.9 | 41.7 | 12.2 |
| NB7 | 314.0 | 100.0 | 0 | −0.4 | 87.7 | 31.5 |
| NB8 | −404.5 | −159.3 | 26.3 | 0 | 156.2 | 49.2 |
| NB9 | −184.0 | − 73.6 | −5.3 | −3.1 | 25.4 | 9.2 |
| NB10 | 184.0 | 63.2 | 25.4 | 8.3 | 25.4 | 8.3 |

NOTE: Example 1 has 6 nodes, 12 links.

lation included the conservation-of-flow constraints. In each formulation, the objective function was written simply as min $z$. What distinguished the six formulations were the rest of the constraints, which also defined $z$ in the objective function. In the following, the objective function for each MP formulation is written in words, followed by the constraint equations that define $z$.

*NB5*: Minimize the largest absolute link volume change.

$$z \geq |V_o(i,j) - V_b(i,j)| \quad \text{for each link } i,j$$

(MP formulation NB5 has its basis in measure of merit 8.)

*NB6*: Minimize the largest percent absolute link volume change.

$$z \geq \frac{|V_o(i,j) - V_b(i,j)|}{V_o(i,j)} \quad \text{for each link } i,j$$

(See also measure of merit 3.)

*NB7*: Minimize the average link volume change.

$$c(i,j) = \frac{V_o(i,j) - V_b(i,j)}{V_o(i,j)} \quad \text{for each link } i,j$$

$$z = \sum_{i,j} c(i,j)$$

(See also measure of merit 5.)

*NB8*: Minimize the average percent link volume change.

$$c(i,j) = \frac{V_o(i,j) - V_b(i,j)}{V_o(i,j)} \quad \text{for each link } i,j$$

$$z = \sum_{i,j} c(i,j)$$

This is a variation of measure of merit 5. Let "average percent link volume change" be measure of merit 9.

*NB9*: Minimize the average absolute link volume change.

$$c(i,j) = |V_o(i,j) - V_b(i,j)| \quad \text{for each link } i,j$$

$$z \geq \sum_{i,j} c(i,j)$$

Another variation of measure of merit 5, "average absolute link volume change" becomes measure of merit 10.

*NB10*: Minimize the average percent absolute link volume change.

$$c(i,j) = \frac{|V_o(i,j) - V_b(i,j)|}{V_o(i,j)} \quad \text{for each link } i,j$$

$$z \geq \sum_{i,j} c(i,j)$$

(MP method NB10 is based on measure of merit 2.)

These six formulations were applied to Example 1, with the results summarized in Table 5. That method NB5 was designed to optimize the NB solution with respect to measure of merit 8 is borne out by the best entry (underlined) in column 2, row 3. Likewise, NB6 performs best with respect to measure of merit 3. The results for the MLM and NB3 methods used earlier in this paper are included here for comparison. This is because MLM is the "literature standard," but we prefer the objective built into NB3.

Although space limitations prevent a link-by-link listing of each solution, we can report that only NB5 and NB6 had sets of balanced link flows that were similar. There were considerable differences in $V_b(i,j)$ values produced by the various methods. Table 5 indicates that NB7 and NB8 perform poorly for any measure of merit other than their own, and this was confirmed in other tests. Unless measure of merit 5 or 9 is the *only* important one, these methods should not be used and are omitted from the remainder of this paper.

The results of the surviving methods in Example 2 are tabulated in Table 6. Again, a method performed best where it was designed to; methods NB7 and NB8 were clearly and consistently inferior elsewhere and have been omitted from Table 6, and the earlier methods (MLM and NB3) are competitive. Also, NB5 does not seem to perform well in this example, which is a simplification of a real highway corridor.

For our final example, we return to Campus NE, an exact representation of a network with 54 nodes and 128 links. Table 7 shows that NB6 performs well for its own measure of merit (3), whereas NB5 is among the two worst surviving methods for four of the five criteria for which it was *not* designed.

## CONCLUSIONS

We have sought to investigate how the NB solution for a network is affected by choice of method, which by implication also means choice of criterion. We have found that it is possible to find an optimal solution with respect to one criterion, but that the solution may be unacceptable according to other reasonable alternative criteria. On the basis of our tests, mathematical programming methods NB5 through NB8 lack versatility in this respect, whereas NB9 and NB10 do fairly well. Also doing well for most criteria are the standard MLM method and our NB3 algorithm.

It is interesting to note that NB3 and NB6 are designed to pursue the same objective: minimize percent link volume change. As an optimization routine, NB6 is always superior to NB3 for criterion 2, usually by a small margin. For most

TABLE 6 SUMMARY OF MP RESULTS FOR EXAMPLE 2

| Method | Measure of Merit 8 | 3 | 5 | 9 | 10 | 2 |
|---|---|---|---|---|---|---|
| MLM | 418.0 | −8.8 | 5.9 | 0.1 | 116.2 | 3.5 |
| NB3 | 482.0 | −7.2 | 27.6 | 0.7 | 130.0 | 4.6 |
| NB5 | 257.5 | 100.0 | 57.2 | 9.4 | 175.4 | 17.1 |
| NB6 | 489.0 | −7.1 | 7.5 | 3.0 | 135.8 | 5.2 |
| NB9 | −515.0 | 64.3 | 4.3 | 0.4 | 103.0 | 8.4 |
| NB10 | −614.0 | −12.5 | 24.7 | 0.2 | 113.5 | 2.2 |

NOTE: Example 2 has 23 nodes, 32 links.

TABLE 7 SUMMARY OF MP RESULTS FOR CAMPUS NE

| Method | Measure of Merit 8 | 3 | 5 | 9 | 10 | 2 |
|---|---|---|---|---|---|---|
| MLM | 228.0 | 64.0 | 3.8 | 1.3 | 19.4 | 8.2 |
| NB3 | 215.0 | 75.0 | 5.2 | 5.4 | 32.5 | 17.7 |
| NB5 | 172.0 | −550.0 | −1.9 | −5.8 | 66.5 | 63.8 |
| NB6 | 724.1 | 60.5 | 179.8 | 55.8 | 182.2 | 58.2 |
| NB9 | −324.0 | −241.4 | −0.3 | −0.2 | 14.0 | 8.2 |
| NB10 | 356.0 | 100.0 | 0 | 0 | 16.9 | 3.5 |

NOTE: Campus NE has 54 nodes, 128 links.

TABLE 8 SUGGESTED RANKINGS BY CRITERION

| Rank | Method by Criterion 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Best | MLM | NB3 | NB9 | MLM | NB9 | NB10 |
| Second | NB3 | MLM | MLM | NB10 | NB10 | MLM |
| Third | NB9 | NB10 | NB10 | NB9 | MLM | NB9 |
| Fourth | NB10 | NB9 | NB3 | NB3 | NB3 | NB3 |

of the other criteria, NB3 is usually superior and usually by a large margin.

Any of the four surviving methods (MLM, NB3, NB9, and NB10) has exhibited adequate versatility in our tests, but in the event that one or more of our six criteria take on special importance, we rank the four methods for each criterion in Table 8. Of course, the MP-based method formulated for a specific criterion will provide the optimal solution if the size of the problem does not exceed time or storage constraints. Another finding is that a simple algorithmic method such as NB3 seems to provide NB solutions that are of high enough quality to be comparable with the MLM. We are convinced that any competent (not necessarily advanced) computer programmer can convert the steps described for method NB3 in this paper into a usable computer code in a short time.

## ACKNOWLEDGMENT

The original draft of this paper was prepared by J.D.F. during his sabbatical leave at the Institut fuer Verkehrswesen, Universitaet (TH) Karlsruhe, West Germany, May to December 1987. He wishes to thank W. Leutzbach, Director of the Institute, and others at the Institute who provided him with the resources and support that made the completion of that draft possible.

## REFERENCES

1. H. J. van Zuylen and L. G. Willumsen. The Most Likely Trip Matrix Estimated From Traffic Counts. *Transportation Research B*, Vol. 14B, 1980, pp. 281–293.
2. L. G. Willumsen. Simplified Transport Models Based on Traffic Counts. *Transportation*, Vol. 10, 1981, pp. 257–278.
3. R. W. Floyd. Algorithm 97, Shortest Path. *Communications of the Association of Computing Machinery*, Vol. 5, 1962, p. 345.
4. D. F. Beagan. Balancing Traffic Counts on a Network. *McTrans*, Vol. 1, No. 2, Fall 1986, pp. 8–9.
5. H. J. van Zuylen and D. M. Branston. Consistent Link Flow Estimation From Counts. *Transportation Research*, Vol. 16B, No. 6, 1982, pp. 473–476.
6. D. F. Beagan. Maximizing the Information From Traffic Counts. Presented at the National Conference on Transportation Planning Applications, Orlando, Fla., 1987.