Journal of Southeast University (English Edition) Vol. 21, No. 2, pp. 225 - 228 June 2005 ISSN 1003—7985

Direct computing methods for turn flows in traffic assignment

Ren Gang Wang Wei

(College of Transportation, Southeast University, Nanjing 210096, China)

Abstract: Two methods based on a slight modification of the regular traffic assignment algorithms are proposed to directly compute turn flows instead of estimating them from link flows or obtaining them by expanding the networks. The first one is designed on the path-turn incidence relationship, and it is similar to the computational procedure of link flows. It applies to the traffic assignment algorithms that can provide detailed path structures. The second utilizes the link-turn incidence relationship and the conservation of flow on links, a law deriving from this relationship. It is actually an improved version of Dial's logit assignment algorithm. The proposed approaches can avoid the shortcomings both of the estimation methods, e. g. Furness's model and Frator's model, and of the network-expanding method in precision, stability and computation scale. Finally, they are validated by numerical examples.

Key words: turn flow; traffic assignment; Dial's algorithm; directly computing method

Turn flows at intersections are critical parameters for various intersection analyses such as control-type determination, geometry design, signal setting (for signal-controlled intersections) and ramp design (for interchanges). However, they are not easy to be surveyed compared with link flows.

Even in traffic assignment algorithms that aim at predicting traffic flows, generally, only link flows are computed directly while turn flows have to be estimated from link flows by such approaches as Furness's method and Frator's method[1,2] after traffic assignment. But turn flow output by these methods are merely approximations after iterative computation. Namely they are not always consistent with the path flows that underlie link and turn flows. Moreover, these methods are unstable because the results depend on the initial values.

Another approach to computing turn flows in traffic assignment is the network-expanding method[3], where the original network with real turning movements is transformed to the expanded network with corresponding dummy links firstly, and then the dummy link flows, i. e. the original turn flows, can be calculated with the conventional traffic assignments algorithms provided these dummy links are handled as the ordinary ones. But this method increases the size of networks too much for introducing dummy links and nodes. Thus it leads to redundant computational time and computer memory unacceptable for large-scale networks.

Fortunately, only a few improvements are required for the commonly used traffic assignment algorithms covering the deterministic and stochastic models to directly compute turn flows as well as link flows, and to eliminate additional iterative estimation and network transformation as in Furness's method, Frator's method and the network-expanding method. This is just the subject this paper is to deal with.

1 Traffic Assignment Models

1.1 Notations

Let $G=(N, A)$ be a directed graph defined by a set $N=\{i\}$ of nodes and a set $A=\{(i,j)\}\subseteq N\times N$ of links. Let $R\subseteq N$ be the set of trip origins and $S\subseteq N$ be the set of trip destinations, and let $W=\{r\text{-}s\}\subseteq R\times S$ be the set of O-D pairs. Let $F(i)=\{j \mid (i,j)\in A\}$ be the set of nodes adjacent to i and $B(i)=\{j \mid (j,i)\in A\}$ be the set of nodes adjacent from i, $\forall i\in N$. Let K_{rs} denote the set of paths between O-D pair r-s and q_{rs} denote the travel demand between O-D pair r-s, $\forall r\text{-}s\in W$. The flow, travel cost and choice probability of path k between O-D pair r-s are denoted by f_k^{rs}, c_k^{rs} and P_k^{rs}, respectively, $\forall r\text{-}s\in W$, $k\in K_{rs}$. Let x_{ij} and t_{ij} denote the flow and travel cost on link (i,j) respectively, $\forall (i,j)\in A$; thus $\boldsymbol{x}$ is a vector representation of all x_{ij}. The indicator variable of the path-link incidence relationship is denoted by $\delta_{ij,k}^{rs}$, $\forall r\text{-}s\in W$, $k\in K_{rs}$, $(i,j)\in A$, where $\delta_{ij,k}^{rs}=1$ if link (i,j) is on path k between O-D pair r-s, and $\delta_{ij,k}^{rs}=0$ otherwise. The flow on turn (i,j,m) is denoted by y_{ijm}, $\forall (i,j),(j,m)\in A$.

1.2 Fundamentals

The general traffic assignment models can be categorized into user equilibrium (UE) and stochastic user equilibrium (SUE) in terms of the route choice mechanism. The foundation of UE assignment is an all-or-nothing network loading model while that of

Received 2004-10-29.

Foundation item: The National Natural Science Foundation of China (No. 50378016).

Biography: Ren Gang (1976—), male, doctor, lecturer, rengang@seu.edu.cn.

SUE assignment is a stochastic network loading model (sometimes called multiple path loading).

The above two network loading models, however, can be generalized as finding a link-flow pattern $\boldsymbol{x}$ satisfying

$$x_{ij} = \sum_{rs} \sum_{k} f_k^{rs} \delta_{ij,k}^{rs} \qquad \forall (i,j) \tag{1}$$

$$f_k^{rs} = q_{rs} P_k^{rs} \qquad \forall r\text{-}s, k \tag{2}$$

where $0 \leqslant P_k^{rs} \leqslant 1$, $\sum_k P_k^{rs} = 1$, and P_k^{rs} usually depends on c_k^{rs} that is calculated by

$$c_k^{rs} = \sum_{ij} t_{ij} \delta_{ij,k}^{rs} \qquad \forall r\text{-}s, k \tag{3}$$

For each $r\text{-}s \in W$, if set $P_m^{rs} = 1$ and $P_k^{rs} = 0 (\forall k (\neq m) \in K_{rs})$ where m is the shortest path between O-D pair r-s, Eqs. (1) to (3) become an all-or-nothing network loading model; otherwise, i. e. $\{P_k^{rs}\}$ are not determined that simply, so it is a stochastic network loading model.

The popular stochastic network loading models are of logit and probit types. In the logit model, P_k^{rs} is in the form of

$$P_k^{rs} = \mathrm{e}^{-\theta c_k^{rs}} \Big/ \sum_{l \in K_{rs}} \mathrm{e}^{-\theta c_l^{rs}} \qquad \forall r\text{-}s, k \tag{4}$$

where θ is a positive constant. But there exists no mathematical formula for P_k^{rs} in the probit model.

Once the interaction between link flow and travel cost is introduced into network loading models, namely t_{ij} is replaced by a function $t_{ij}(\boldsymbol{x})$, UE and SUE models can be formulated immediately. However, the key to the solution algorithms for UE and SUE models are still all-or-nothing and stochastic network loading algorithms. For this reason, only network loading algorithms are to be studied in this paper without loss of generality.

2 Method Based on Path-Turn Relationship

In most traffic assignment algorithms, whether path flows are explicit or implicit, link flows are essentially derived from path flows by Eq. (1), which reveals exactly the incidence relationship between links and paths.

In general, these algorithms do not give turn flows simultaneously, but we find turn flows can be computed according to the incidence relationship between turns and paths, that is

$$y_{ijm} = \sum_{rs} \sum_{k} f_k^{rs} \delta_{ij,k}^{rs} \delta_{jm,k}^{rs} \qquad \forall (i,j),(j,m) \tag{5}$$

where $\varphi_{ijm,k}^{rs} = \delta_{ij,k}^{rs} \delta_{jm,k}^{rs}$ is just an indicator variable of the path-turn incidence relationship.

Eq. (5) means that the flow on each turn is the sum of the flows on all paths going through that turn. When this method is implemented, for each O-D pair, turn flows are determined in such a way that the travel demand should be assigned onto the turns on the selected path connecting this pair as well as onto the links on this path.

This kind of direct computing method for turn flows is suitable for those assignment algorithms that can provide such detailed information on path structures as to utilize the path-link relationship, whether the path is the shortest path or not. Such algorithms include the all-or-nothing loading algorithm based on the shortest path search, the logit loading algorithm based on the column generation technique[4], and the probit loading algorithm based on the Monte Carlo simulation procedure[5].

3 Method Based on Link-Turn Relationship

For those traffic assignment algorithms that do not give detailed information of path structures, turn flows cannot be calculated by the path-turn relationship as in Eq. (5). Such algorithms are mainly the logit loading approaches proposed respectively in Refs. [6 – 8], of which Dial's algorithm is the most widely used. However, we find that Dial's algorithm only needs a small improvement before it has the ability to directly compute turn flows according to the link-turn relationship rather than the path-turn relationship.

3.1 Original Dial's algorithm

Dial's algorithm assigns O-D flows only to reasonable paths (not to all paths) connecting each O-D pair and obviates path enumeration, which is acknowledged to be the most efficient algorithm for logit network loading. In other words, path flows and link flows in this algorithm are not derived directly from Eqs. (1) to (4), but the results can exactly agree with these formulas.

For each trip origin r, the algorithm performs three steps as follows:

Step 1 Preliminaries

① Compute the shortest distance r_i from r to node i, $\forall i \in N$;

② Compute the likelihood L_{ij}, $\forall (i,j) \in A$, where

$$L_{ij} = \begin{cases} \mathrm{e}^{\theta[r_j - r_i - t_{ij}]} & \text{if } r_i \leqslant r_j \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Step 2 Forward pass to compute link weights

Process nodes in ascending order of r_i, starting with r. Namely, for each node i, compute the link weight w_{ij}, $\forall j \in F(i)$, where

$$w_{ij} = \begin{cases} L_{ij} & \text{if } i = r \\ L_{ij} \sum_{m \in B(i)} w_{mi} & \text{otherwise} \end{cases} \tag{7}$$

Step 3 Backward pass to assign link flows

Process nodes in descending order of r_j, starting with the most distant node. Namely, for each node j, compute the link flow x_{ij}, $\forall i \in B(j)$, where

$$x_{ij} = \begin{cases} 0 & \text{if } j = r \\ \left[q_{rj} + \sum_{m \in F(j)} x_{jm} \right] \dfrac{w_{ij}}{\sum_{m \in B(j)} w_{mj}} & \text{otherwise} \end{cases} \quad (8)$$

In this expression, q_{rj} denotes the travel demand between node pair r-j, which is set at zero if node j is not a trip destination.

Once the above three steps are performed for all trip origins and then link flows are added up, the whole procedure of logit loading is completed.

3.2 Improvement on Dial's algorithm for turn flows

Utilizing the conservation of flow at nodes, Dial's algorithm assigns flows by link while path variables including structures and flows have never appeared. Thus the algorithm succeeds in avoiding path enumeration but fails to directly compute turn flows by Eq. (5).

However, the inherent relationship between turns and links, a turn (i, j, k) consisting of two links (i, j) and (j, k), generates another law that network flows keep conservation on a link, which is expressed mathematically as $\forall\, r\text{-}s \in W, (i, j) \in A$, the following constraint must satisfy

$$\left.\begin{array}{ll} \sum_{m \in B(i)} y_{mij}^{rs} + q_{ij}^{rs} = x_{ij}^{rs} = \sum_{k \in F(j)} y_{ijk}^{rs} & \text{if } i = r \\ \sum_{m \in B(i)} y_{mij}^{rs} = x_{ij}^{rs} = \sum_{k \in F(j)} y_{ijk}^{rs} + q_{ij}^{rs} & \text{if } j = s \\ \sum_{m \in B(i)} y_{mij}^{rs} = x_{ij}^{rs} = \sum_{k \in F(j)} y_{ijk}^{rs} & \text{otherwise} \end{array}\right\} \quad (9)$$

where x_{ij}^{rs} and y_{ijk}^{rs} denote the share of q_{rs} assigned on link (i, j) and on turn (i, j, k), respectively, and q_{ij}^{rs} denotes the share of q_{rs} either emanating form (i, j) if $i = r$ or arriving at (i, j) if $j = s$.

The conservation of flow on links is to say that the flow on a link, the sum of flows entering this link by different turns and the sum of flows leaving this link by different turns have some relations to each other. All or part of these relations can surely be utilized to calculate turn flows. For example, starting with the left-hand equations of Eq. (9), we can make use of the known link flows in company with the link weights concerned to directly calculate the unknown turn flows. Thus an improved Dial's algorithm appears.

The improved algorithm differs from the original version only in step 3 as follows: in addition to the original operations, the turn flow y_{ijk}, $\forall\, i \in B(j), k \in F(j)$, is computed subsequently by

$$y_{ijk} = \begin{cases} 0 & \text{if } j = r \\ x_{jk} \dfrac{w_{ij}}{\sum_{m \in B(j)} w_{mj}} & \text{otherwise} \end{cases} \quad (10)$$

In fact, the relationship as for Eq. (9) is also used in estimating turn flows from link flows by Furness's and Frator's methods, but turn flows in these methods cannot be calculated uniquely and exactly due to the absence of supplementary information such as link weights.

4 Numerical Examples

For the example in Fig. 1, numerical examples to demonstrate the directly computing methods for turn flows will be given respectively. This network, coming from Ref. [9], has nine nodes with nodes 1 and 9 constituting the unique O-D pair, 12 links whose travel costs are labeled nearby and hence 14 turns. The travel demand q_{19} between the O-D pair is 1 000.

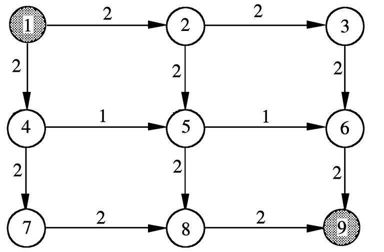

Fig. 1 Example network

Example 1 The direct computing method for turn flows based on the path-turn relationship explained here is a case of all-or-nothing network loading. First, we can get the shortest path 1-4-5-6-9 (with the shortest distance of six units) between O-D pair 1-9 by any practicable shortest path algorithm. Subsequently, according to the all-or-nothing mechanism, the flow on this path is set at 1 000 while those on any other paths are zero, and thus the flows on links (1, 4), (4, 5), (5, 6) and (6, 9) belonging to this path are set at 1 000 respectively while any other link flows are zero. In addition, the flows on turns (1, 4, 5), (4, 5, 6) and (5, 6, 9) belonging to this path are set at 1 000 respectively while any other turn flows are zero.

Example 2 The direct computing method for turn flows based on the link-turn relationship explained here is a case of the improved Dial's algorithm. The shortest distances r_i calculated in step 1 are listed in Tab. 1, and the link likelihoods L_{ij} are listed in Tab. 2. In step 2, w_{12} and w_{14} are calculated firstly according to algorithm rules, and the last one is w_{89}. All the link weights are listed in Tab. 2. In step 3, x_{69} and x_{89} are firstly calculated, followed by x_{58}, x_{78}, y_{589} and y_{789}. When x_{12}, y_{123} and y_{125} are reached, this step is completed. Thus the resulting link and turn flows are shown in Fig. 2. The consistencies between these turn flows and link flows are easy to verify, namely the relationship as in Eq. (9) holds. This indicates the validity of the improved algorithm.

Tab. 1 The shortest distances computed in step 1

j	1	2	3	4	5	6	7	8	9
r_j	0	2.0	4.0	2.0	3.0	4.0	4.0	5.0	6.0

Tab. 2 Link likelihoods and weights (for $\theta = 1$) computed in step 1 and step 2

(i, j)	(1,2)	(1,4)	(2,3)	(2,5)	(3,6)	(4,5)	(4,7)	(5,6)	(5,8)	(6,9)	(7,8)	(8,9)
L_{ij}	1.0	1.0	1.0	0.367 9	0.135 3	1.0	1.0	1.0	1.0	1.0	0.367 9	0.367 9
w_{ij}	1.0	1.0	1.0	0.367 9	0.135 3	1.0	1.0	1.367 9	1.367 9	1.503 2	0.367 9	0.638 6

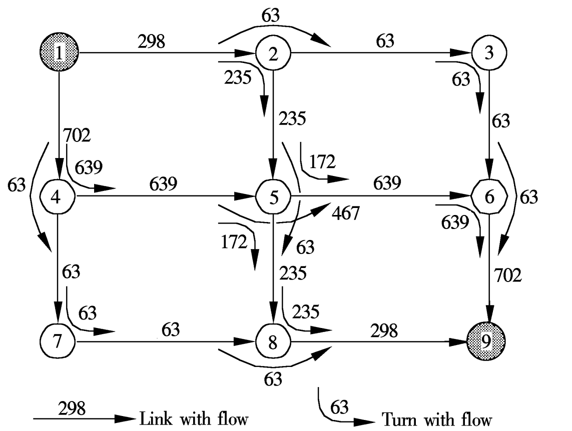

Fig. 2 Link and turn flows computed in step 3

5 Conclusion

In this paper, the two methods for directly computing turn flows as well as link flows in traffic assignment are proposed. The first method is based on the path-turn relationship and is suitable for those traffic assignment algorithms that can provide detailed path structures. On the contrary, for Dial's logit loading algorithm that cannot provide detailed path information, the second method based on the link-turn relationship is suitable.

These two methods have the following advantages: ① Avoiding shortcomings of the conventional approaches including the estimation method and the network-expanding method; ② Applicability to most of the UE and SUE algorithms with no need of large modification. Turn flows are critical parameters in such intersection analyses as control-type determination, geometrical design, and signal setting, so the proposed methods for turn flows are of great value when real data are absent and predicted results are allowed.

References

[1] Wang Wei, Xu Jiqian, Yang Tao, et al. *Theories and applications of urban transportation planning* [M]. Nanjing: Southeast University Press, 1998. 29 − 33. (in Chinese)

[2] Transportation Research Board. Highway capacity manual 2000 [R]. Washington, DC: National Research Council, 2000.

[3] Yang B, Miller-Hooks E. Adaptive routing considering delays due to signal operations [J]. *Transportation Research*, 2004, **38B**(5): 385 − 413.

[4] Wong S C, Yang Chao, Lo Hong K. A path-based traffic assignment algorithm based on the TRANSYT traffic model [J]. *Transportation Research*, 2001, **35B**(2): 163 − 181.

[5] Clark S D, Watling D P. Sensitivity analysis of the probit-based stochastic user equilibrium assignment model [J]. *Transportation Research*, 2002, **36B**(1): 617 − 635.

[6] Cheng L, Wang J Y, Shao Y H, et al. Choice probability problems in stochastic assignment [A]. In: Mao B H, Tian Z Z, Sun Q X, eds. *Proceedings of the 4th International Conference on Traffic and Transportation Studies* [C]. Beijing: Science Press, 2004. 384 − 391.

[7] Bell M G H. Alternative to Dial's logit assignment algorithm [J]. *Transportation Research*, 1995, **29B**(4): 287 − 295.

[8] Akamatsu T. Cyclic flows, Marcov process and stochastic traffic assignment [J]. *Transportation Research*, 1996, **30B**(5): 369 − 386.

[9] Sheffi Y. *Urban Transportation networks*: *equilibrium analysis with mathematical programming methods* [M]. Englewood Cliffs: Prentice-Hall Inc, 1985. 288 − 294.

交通分配中转向流量的直接计算方法

任　刚　　王　炜

(东南大学交通学院, 南京 210096)

摘要: 为了在交通分配中直接得到转向流量而不是通过事后推算或网络扩展获取, 对常规的交通分配算法稍作修改后提出了转向流量的 2 种直接计算方法. 第 1 种方法基于转向-路径关系, 其原理类似于路段流量的计算, 适用于给出具体路径构成的交通分配算法. 第 2 种方法利用了转向-路段关系及其衍生的路段上流量守恒律, 是一种改进型 Dial 算法. 这 2 种方法能避免传统的诸如 Furness, Frator 模型的事后推算法以及网络扩展法在精度、稳定性和计算量等方面的缺陷, 其有效性通过算例分别得到了验证.

关键词: 转向流量; 交通分配; Dial 算法; 直接计算方法

中图分类号: U491