Transportation Planning and Technology
Vol. 33, No. 2, March 2010, 123–137

An improved Dial's algorithm for logit-based traffic assignment within a directed acyclic network

Bing-Feng Si[a], Ming Zhong[b]*, Hao-Zhi Zhang[a] and Wen-Long Jin[c]

[a]*State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China;* [b]*Department of Civil Engineering, University of New Brunswick, GD-128, Head Hall, 17 Dineen Drive, P.O. Box 4400, Fredericton, NB E3B 5A3, Canada;* [c]*School of Information Science and Technology, University of Science and Technology of China, Hefei, China*

(Received 21 May 2008; final version received 11 January 2010)

Dial's algorithm is one of the most effective and popular procedures for a logit-type stochastic traffic assignment, as it does not require path enumeration over a network. However, a fundamental problem associated with the algorithm is its simple definition of 'efficient paths', which sometimes produces unrealistic flow patterns. In this paper, an improved algorithm based on the route extension coefficient is proposed in order to circumvent this problem, in which 'efficient paths' *simultaneously* consider link travel cost and minimum travel cost. Path enumeration is still not required and a similar computing efficiency with the original algorithm is guaranteed. A limitation of the algorithm is that it can only be applied to a directed acyclic network because a topological sorting algorithm is used to decide the order of the sequential calculation. A numerical example based on the Beijing subway network illustrates the effectiveness of the proposed algorithm. It is found that it is able to exclude most unrealistic paths, but include all reasonable paths when compared with path enumeration and the original Dial's algorithm.

Keywords: Dial algorithm; efficient paths; assignment; logit model

1. Introduction

Dial's algorithm is one of the most effective procedures for generating 'reasonable' paths between origin–destination (O–D) pairs for a logit type stochastic assignment (Dial 1971). It can be easily applied to a large scale network since path enumeration is not required. However, it sometimes produces unrealistic flow patterns (Akamatrsu 1996) in which no flow is assigned on paths that are being used in reality because of its simple definition of 'efficient paths.'

Previous research tried to solve the above problem. For example, Bell (1995) proposes two methods to find a logit-based assignment based on the Floyd–Warshall algorithm, in which path enumeration is not required. The first method considers a finite number of paths including those with or without loops. However, its results depend on the calculation sequence and is thus not unique and stable. The second method considers all paths, which will be an *infinite* number in the presence of loops and assume that the summation sequence of the network link weight matrix

*Corresponding author. Email: ming@unb.ca

ISSN 0308-1060 print/ISSN 1029-0354 online

DOI: 10.1080/03081061003643705
http://www.informaworld.com

converges and use the reversed matrix to calculate the parameters of the logit model. Computing is a concern with the second approach with an infinite number of paths. Moreover, Wong (1999) pointed out that Bell (1995) assumed the sum of a geometric series of the weights matrix always converges for the second method, but it is only true for acyclic networks, not necessarily for general networks. Akamatrsu (1996) presents the logit type stochastic assignment that does not restrict the assignment paths and proposed two theoretical approaches for solving the model, one of which is based on the theory of the Markov Chain and results in the same procedure presented by Bell (1995). The second is based on the equivalency of the maximum entropy principle and the logit model, and he used the decomposition of the entropy function to solve the equivalent model. However, it is not effective for a large scale network. Huang and Bell (1998) tried to use network equilibrium iterative searching process to find efficient paths in order to solve the stability problem with iterative Stochastic User Equilibrium (SUE) assignment methods that incorporate Dial's algorithm, but it is difficult to implement such a method for a practical network as there would be too many possible paths. More recently, researchers in China used topology theories to study the structure of networks for more effective search of 'efficient paths.' For instance, Li and Huang (2003) and Zhu *et al.* (2000) used width-first and depth-first algorithms respectively to search all possible paths between O–D pairs. The complexity of the two algorithms is the same, but their node searching orders are different. Li *et al.* (2005) proposed to find connection paths based on a topological sort algorithm. However, his method can only exclude 'unreasonable' paths when there are loops over the network. The topology-based algorithms would have the following problems when applied to practical assignment for urban traffic networks: one is the algorithm is path-based and the other is the large amount of redundant computations associated with the solution process.

To circumvent the above problem of Dial's algorithm, this paper proposes an improved algorithm which is found to be more effective to include all 'reasonable' paths and, at the same time, keeps a similar efficiency as the original one. A numerical example based on the Beijing subway network is used to compare the proposed algorithm with the path enumeration method and the original Dial's algorithm.

2. Review of Dial's algorithm

The objective of Dial's algorithm is to obtain the link flows corresponding to the division of each set of O–D trips between a set of 'efficient' paths according to a logit-type formula:

$$p_k^{rs} = \frac{\exp(-\theta c_k^{rs})}{\sum_m \exp(-\theta c_m^{rs})}, \quad \forall k, r, s \tag{1}$$

where p_k^{rs} is the choice probability of the kth efficient path between O–D pair rs; c_k^{rs} is the travel cost of the kth efficient path between O–D pair rs; θ is dispersion parameter that is inversely proportional to the standard error of the distribution of the perceived path travel cost.

In Dial's algorithm, the paths for loading flows are restricted to the 'efficient paths,' which are defined as the path including only links that take the traveler further away from the origin and closer to the destination. Such links can be identified by associating the following two labels with each nodei: $r(i)$ and $s(i)$. The first label, $r(i)$, denotes the travel cost from the origin node r to node i along the shortest path, whereas $s(i)$ denotes the travel cost from node i to the destination node s along the shortest path. Each 'efficient path' includes only the links (I, j), which satisfies the following condition:

$$r(i) < (j) \text{ and } s(i) > s(j) \tag{2}$$

The procedures of Dial's algorithm for one O–D pair rs are outlined below (Sheffi 1985).

Step 0. Preliminaries:

(1) Compute the minimum travel cost from origin r to all other nodes. Get $r(i)$ for each node i.
(2) Compute the minimum travel cost from each node i to destination s. Get $s(i)$ for each node i.
(3) Define O_i as the set of downstream nodes of all links leaving node i.
(4) Define I_i as the set of upstream nodes of all links arriving node i.

Step 1. For each link (i, j) compute the 'link likelihood,' $L(i, j)$, where

$$L(i,j) = \begin{cases} \exp[\theta(r(j) - r(i) - t_{ij})] & \text{if } r(i) < r(j) \text{ and } s(i) > s(j) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where t_{ij} is the measured travel cost on the link (i, j). Note that $L(i, j) = 1$ for all the links on the shortest path between r and s, while $L(i, j) = 1$ for all the links that are not included in any efficient path.

Step 2. Forward pass. Consider nodes in ascending order of $r(i)$ starting with the origin, r. For each node, i, calculate the 'link weight,' $w(i, j)$, for each $j \in O_i$ (i.e. for each link starting from i), where

$$w(i,j) = \begin{cases} L(i,j) & \text{if } i = r \\ L(i,j) \sum_{m \in O_i} w(m, i) & \text{otherwise} \end{cases} \tag{4}$$

This step is applied iteratively until the destination node, s, is reached.

Step 3. Backward pass. Consider nodes in ascending values of $s(j)$ starting with the destination, s. When each node, j, is considered, compute the link flow $x(i, j)$ for each $i \in I_j$ (i.e. for each link ending at j), by following assignment:

$$x(i,j) = \begin{cases} q_{rs} w(i,j) \Big/ \sum_{m \in I_j} w(m,j) & \text{if } j = s \\ w(i,j) \sum_{m \in I_i} x(j,m) \Big/ \sum_{m \in I_j} w(m,j) & \text{otherwise} \end{cases} \tag{5}$$

where q_{rs} is the total demand from r to s and $x(i, j)$is the assignment flow on link (i, j). This step is applied iteratively until the origin node, r, is reached.

3. Limitation of Dial's algorithm

One problem of Dial's algorithm is that it restricts the assignment path set to 'efficient paths' based on Condition (2). As a result, it may produce an unrealistic flow pattern for which flow is not loaded on paths with low travel cost, but those with high cost. This problem was first observed and discussed by Akamatrsu (1996) and can be illustrated by Figure 1.

According to the preliminary procedure (Step 0 above) of Dial's algorithm, the result of $r(i)$ and $s(i)$ for each node can be computed as shown in Table 1.

It is found that two links, (2, 3) and (3, 4), do not meet the criteria of condition (2) because $r(2) = r(3) = r(4)$, thus the paths $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$ cannot be included as 'efficient paths' based on Dial's algorithm. However, the travel cost on this path is 4.5, while the travel cost of the 'efficient' path identified by the Dial's algorithm $1 \rightarrow 4 \rightarrow 5$ is 5. Obviously, such results are not reasonable.

Based on the criterion of a 'single pass' algorithm ($r(i) < r(j)$ (Sheffi 1985)), all of the connecting paths from origin r to destination s will be selected as 'efficient paths' in this example. Nevertheless, the travel cost of path $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ is 6, which is one and half times the minimum travel cost between O–D pair (r, s). In practice, we would predict that few or no travelers would choose such a path.

Through Inequality condition (2), Dial's algorithm can exclude loops from efficient paths and all road segments that do not meet the condition. The inequality also makes the calculation of link weights and flow assignments in Inequality conditions (3) and (4) easy, as they should be either zero or have been attached a value. However, regardless of whether it is a 'single-pass' or a 'double-pass' Dial algorithm, it always assumes that the probability of a route being chosen is proportional to the product of the likelihood of all composite links (Sheffi 1985), that is:

$$p_k^{rs} = K \prod_{ij} [L(ij)]^{\delta_{ij,k}^{rs}}, \quad \forall k, r, s. \tag{6}$$

This means that if a path contains more links, the probability of the route being chosen would be smaller. However, a path could consist of many short links and thus it may have a lower total travel cost at the path level. Therefore, the probability of such a path being chosen should be high. Obviously, this conclusion is contradictory to Equation (6). The reason for such an inconsistency is that the criteria for an efficient path defined by Dial's algorithm only considers the relative spatial relationship between an interim node and origin/destination (whether a node is

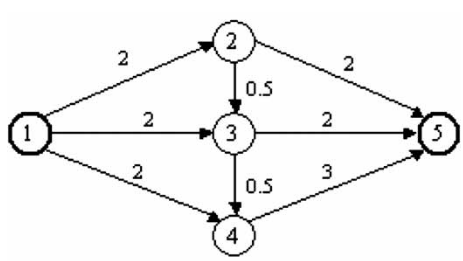

Figure 1. A simple illustrative network.

Table 1. The preliminary phase of Dial's algorithm.

Node	1	2	3	4	5
$r(i)$	0	2	2	2	4
$s(i)$	4	2	2	2.5	0

further away from the origin and closer to the destination) and does not consider the travel cost of individual links explicitly.

4. New definition of efficient paths

Generally speaking, when travelers choose a path between an O–D pair, they will not consider all connected paths, but only a portion of them that are deemed 'reasonable.' The portion under consideration should lie within a travel cost limit beyond which no paths will be considered. Obviously, a 'reasonable' or 'efficient' path should in practice not have a loop. That is, no traveler will start from a node and then comeback to the node again during the trip. Based on the above reasons, Li and Huang (2003) defined a path between O–D pair rs as an efficient path if it meets the following conditions:

(1) The path must not have a loop; and
(2) The travel cost of any effective path must be less than $(1+H)$ times that of the shortest-path. That is:

$$c_k^{rs} \leq (1+H)c_{\min}^{rs}, \quad \forall k, r, s \tag{7}$$

where $c_{\min}^{rs}$is the minimum travel cost between r and s, H is a non-negative constant with a value between zero and one, which was called the route extension coefficient (Leurent 1997).

Leurent (1997) indicates that there are random factors affecting travelers' path choices and thus we can speculate that people may choose paths with a reasonably higher travel cost than the shortest-path. It is obvious that the searching domain of efficient paths can be enlarged or reduced through adjusting the coefficient, H. In general, H can take a value of 0.16 for inter-city paths and between 0.13 and 0.15 for intra-city paths (Leurent 1997).

Studying the original Dial's algorithm and the definition of efficient path proposed by Li and Huang (2003) leads to the following two theorems:

Theorem 1. The criteria defined by Inequality (2) are neither a sufficient nor a necessary condition of Inequality (7).

Proof: first we prove that Inequality (2) is not a sufficient condition of Inequality (7). From Inequality (2), we can get:

$$c_m^{rj} \geq r(j) > r(i) \text{ and } c_n^{is} \geq s(i) > s(j) \tag{8}$$

where c_m^{rj} is the travel cost of path m from the origin r to node j through link (i, j); c_n^{is} is the travel cost of path n from node i to the destination s through link (i, j). From Inequality (8), the following can be easily derived:

$$c_m^{rj} + c_n^{is} > r(i) + s(j). \tag{9}$$

If there is a path k from the origin r to the destination s through link (i, j) which is:

$$c_k^{rs} = c_m^{rj} + c_n^{is} - t_{ij} > r(i) + s(j) - t_{ij}. \tag{10}$$

If $r(i) + s(j) - t_{ij} \geq (1 + H)c_{\min}^{rs}$, which implies $c_k^{rs} \geq (1 + H)c_{\min}^{rs}$, then it is obvious for path k between the O–D pair rs through link (i, j), which is judged as an efficient path based on Dial's algorithm (Inequality (2)), to be evaluated an 'unreasonable' path based on Inequality (7).

Next we prove that Inequality (2) is not a necessary condition of Inequality (7). For a given path k through link (i, j) connecting the O–D pair rs, the following should hold:

$$r(i) + t_{ij} + s(j) \leq c_k^{rs} \tag{11}$$

if Inequality (7) holds, then it is easy to get:

$$r(i) + t_{ij} + s(j) \leq c_{\min}^{rs} + Hc_{\min}^{rs} \tag{12}$$

Also, based on the definition of shortest path, the following should hold:

$$c_{\min}^{rs} \leq r(j) + s(j) \tag{13}$$

According to (12) and (13), the following inequality can be easily obtained:

$$r(i) \leq r(j) + Hc_{\min}^{rs} - t_{ij} \tag{14}$$

It is obvious that Inequality (14) is not equivalent to Inequality (2). For example, if $Hc_{\min}^{rs} \geq t_{ij}$, then Inequality (2) may not hold.

It can be seen from the above proof that the criteria of efficient path from Dial's algorithm (Inequality (2)) are neither sufficient nor necessary conditions of Inequality (7). This is the exact reason for unrealistic assignment patterns resulting from the original Dial's algorithm, as it does not follow travelers' decision-making processes in path choices. It is simply that the inequality criterion (2) merely considers the relative location of a given node on a path to the origin and destination to reduce the computation burden and remove the possibility of a loop, but never takes the travel cost contribution of each composite link to the whole path explicitly into account. At the same time, the second theorem can be obtained from the above proof procedure.

Theorem 2. The necessary condition for the link-based efficient path is: if a path between the O–D pair r s is efficient, all the composite links must meet Inequality (12). In other words, if a path is an efficient path defined by Inequality (7), then any link on such a path must satisfy Inequality (12).

Note that the inequality condition (12) is a necessary condition for an efficient path, but not a sufficient one. The theorem indicates that it must satisfy the inequality condition (12) for any composite link of an efficient path.

Using Figure 1 again as an example, assume we choose $H = 0.13$. It means that we count all the paths with travel cost of 4.52 or less as 'reasonable' or 'efficient' (because the minimum travel cost from node one to node five is 4.0). Because link (2, 3) satisfies the inequality condition (12), it will be included as an 'efficient' link according to new criterion presented in theorem 2, while this link is not an 'efficient' one according to Dial's criterion.

In addition, it can be found from the inequality condition (14) that for all paths that meet Inequality (12), if link (i, j) satisfies the following inequality, then it will also meet the criterion of a 'single-pass' Dial's algorithm, $r(i) \leq r(j)$:

$$Hc^{rs}_{\min} \leq t_{ij}. \tag{15}$$

That is, Inequality (15) is a sufficient condition for the single-pass Dial's algorithm. If the inequality condition (7) holds, then $r(i) \leq r(j) + Hc^{rs}_{\min} - t_{ij}$ must hold. Consequently, $r(i) \leq r(j)$ will be hold as long as $Hc^{rs}_{\min} - t_{ij}$ is negative.

Using Figure 1 as an example again, assume $H = 0.13$, then $Hc^{rs}_{\min} = 0.13 \times 4.0 = 0.52$. Since links (2, 3) and (3, 4) have a cost of 0.5, which is less than 0.52, they will not be included in any efficient paths according to Dial's algorithm.

5. Improved Dial's algorithm

This paper proposes an improved Dial's algorithm based on the new link-based condition for efficient paths. For a given link (i, j), it can only be included as part of an efficient path if the inequality condition (12) is satisfied. Inequality (12) is used to prune any links that do not meet the condition.

In step 2 of the original Dial's algorithm, the conditional probability of choosing a link ('link weight') is calculated successively in ascending order of $r(i)$ starting with the origin, r. While the link flow is assigned in ascending order of $s(j)$ starting with the destination, s, in step 3. It can be easily understood that the result depends on the order of calculation that is generated by its simple definition of 'efficient paths' according to Equation (2). According to Akamatrsu (1996), Dial's algorithm carefully avoids the cyclic flow to maintain the consistency of the sequential procedure, and that the restriction of the path set to 'efficient paths' is the derivative result. In this paper, the new definition of 'efficient paths' – Equation (12) – is used to improve Dial's algorithm and a topological sort algorithm from graph theory is used to decide the order of the sequential calculation.

The topological sort of a directed acyclic graph is a linear order of its nodes which is consistent with the link induced by the nodes i and j where node i comes before node j if there is a directed link from i to j. First, insert starting node into a queue L. Set Q_{rs} for O–D pair rs is the list that contains the nodes of a given path in topological sorted order. The procedure of topological sort used in this paper is described as follows:

Step 0. Initialization. Set a state variable, k_{ij}, for link (i, j). $k_{ij} = 0$ means the link (i, j) has not been searched; $k_{ij} = -1$ means the link (i, j) is not on 'efficient path'; $k_{ij} = 1$ means the link (i, j) is on 'efficient path.'

Step 1. Compute the minimum travel cost from original node r to each node to get $r(i)$ and compute the minimum travel cost from each node to destination node s to get $s(i)$. Set $k_{ij} = -1$ if the link (i, j) is in original node$r$ or out from destination node$s$; set $k_{ij} = -1$ if the link (i, j) is in node j or out from node j with $s(j) = \infty$; set $k_{ij} = -1$ if the link (i, j) does not meet Inequality (12); set the state variables of the other links $k_{ij} = 0$. Define K as the set of all nodes with no incoming links and Q_{rs} as an empty list that includes the sorted element; put original node r into L.

Step 2. Searching process. Implement the following iteration:
while L is non-empty do
remove a node i from K if the state variable $k_{ni} = -1$ or $1, \forall n$
insert i into Q_{rs}
for each node j with $k_{ij} = 0$ for link (i, j) do
set $k_{ij} = 1$
insert j into L
when L is empty, the procedure is stopped and a topologically sorted order is stored in the list Q_{rs}.

The improved Dial's algorithm proposed in this paper then can be described as follows:

Step 0. Find the calculation order according to the topologically sort algorithm presented above. Define O_i as the set of downstream nodes of all links leaving node i and I_i as the set of upstream nodes of all links arriving node i.

Step 1. For each link (i, j) compute the 'link likelihood,' $L(i, j)$, where

$$L(i,j) = \begin{cases} \exp[\theta(r(j) - r(i) - t_{ij})] & \text{if } k_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Step 2. Forward pass. Consider nodes in order of Q_{rs}. For each node, i, calculate the 'link weight,' $w(i, j)$, for each $j \in O_i$, where

$$w(i,j) = \begin{cases} L(i,j) & \text{if } i = r \\ L(i,j) \sum_{m \in O_i} w(m,i), & \text{otherwise} \end{cases} \tag{17}$$

when the destination node, s, is reached, this step is completed.

Step 3. Backward pass. Consider nodes in contrary order of Q_{rs}. When each node, j, is considered, compute the link flow $x(i, j)$ for each $i \in I_j$ by following the assignment:

$$x(i,j) = \begin{cases} q_{rs} w(i,j) \Big/ \sum_{m \in I_j} w(m,j) & \text{if } j = s \\ w(i,j) \sum_{m \in I_i} x(j,m) \Big/ \sum_{m \in I_j} w(m,j) & \text{otherwise} \end{cases}. \tag{18}$$

This step is applied iteratively until the origin node is reached.

It is obvious that the difference between the original Dail's algorithm and this improved algorithm is in Step 0 and Step 1.

6. Numerical example

A numerical example is used to test the improved Dial's algorithm based on the following directed acyclic network (Figure 2) abstracted from the Beijing subway network, which consists of Line 1 (thin line), Line 2 (bold line), Line 13 (dashed line), and Line 5 (dotted line). For illustration purposes, only transfer and major stations are identified. The data were obtained from the Beijing Metro Company and related surveys. The parameters used in the test are estimated with statistical methods (Beijing MTR 2007).

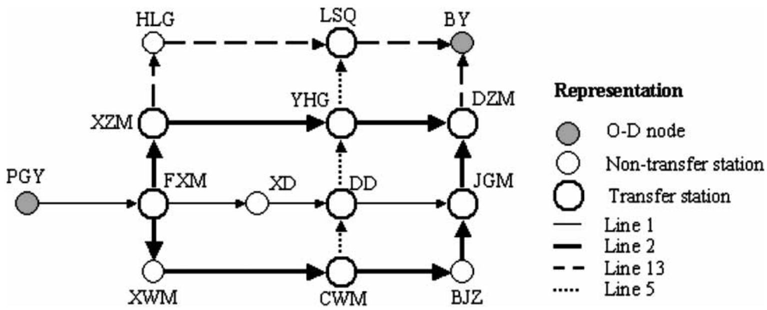

Figure 2. Beijing subway network.

The travel cost of a given link (i, j) in the above network has the following values: If node i is a transfer station and there is a transfer task required, the travel cost is the total of walking time during the transfer at the node i, waiting time, and the in-vehicle time between i and j. If node i is not a transfer station, the travel cost between node i and j is the in-vehicle time only. The basic data required in this example includes transfer walking time, in-vehicle time on a given link, average waiting time, vehicle idling time at every station, and travel demand of each O–D pair. The travel cost of the k^{th} path between the origin r and destination s is calculated as

$$c_k^{rs} = \alpha N_k^{rs} \cdot E_k^{rs} + T_k^{rs} + S_k^{rs}, \quad \forall r, s, k \tag{19}$$

where N_k^{rs} is the number of transfer tasks along the k^{th} path between r and s; E_k^{rs} is the average transfer time, including walking time and waiting time;T_k^{rs} is the total in-vehicle time of the path k; S_k^{rs} is the total vehicle idling time of the path; α is the transfer penalty coefficient (a value of 1.86 is used in this study (Beijing MTR 2007)).

The following parameters are obtained by statistically analyzing the available data.

(1) The average vehicle idling time at a non-transfer station is 0.5 min and the average waiting time at a transfer station is 2 min.
(2) There are in total eight transfer stations. The average transfer walking time between Line 2 and Line 13 at DZM station is 13 min and it is 10 min at the XZM station. The other transfer stations are for changing lines and a shorter duration of 3 min is used as the average transfer time.
(3) The train riding times between stations on Lines 1, 2, and 13 can be obtained from the data posted by Beijing MTR. Since Line 5 has just been launched, the running time between the stations on this line is calculated by dividing the distance with the average train speed obtained from the existing lines. The average train speed used is 0.58 km/min.
(4) A value of 20 is used for parameter θ in the logit model (Beijing MTR 2007).
(5) A value of 1.3 is used as the extension coefficient for effective paths.

First, the efficient paths between Station PGY and Station BY were determined based on inequality criterion (2) of the original Dial's algorithm and travel demand between this O–D pair is then assigned to the five paths obtained, as shown in Table 2.

From these results, the following path PGY–FXM –XWM–CWM–BJZ–DZM-BY was not selected as an efficient path and thus was excluded from assigning any

Table 2. Efficient paths and traffic assignment pattern between Station PGY and Station BY with original Dial's algorithm.

Path No.	Paths	Number of transfers	Travel time/ min	Travel cost/ min	Assignment proportion
1	PGY-FXM-XD-DD-YHG-LSQ-BY	2	96.10	102.98	0.6798
2	PGY-FXM-XZM-YHG-LSQ-BY	3	97.40	108.58	0.1267
3	PGY-FXM-XZM-YHG-DZM-BY	2	97.00	109.90	0.0853
4	PGY-FXM-XZM-HLG-LSQ-BY	2	99.00	110.18	0.0784
5	PGY-FXM-XWM-CWM-BJZ-DZM-BY	2	100.50	113.40	0.0298

travel demand to it. However, the travel cost of this path is 111.4 minutes, which is lower than that of path number 5 in the table. Moreover, it is known that the path is a popular choice among travelers. Obviously, it is the definition of 'efficient path' of the original Dial's algorithm that has resulted in such an unrealistic result.

Next we use path enumeration to find all efficient paths based on inequality criterion (12) with an assumed $H = 0.15$. It means that we will include all paths which have a length less or equal to 1.15 times that of the shortest path into the 'efficient path set' (It should be noted here that it would be very difficult or nearly impossible to use the path enumeration method for a large network). A network assignment was done again with the logit model, which resulted in Table 3.

Based on the proposed improved Dial's algorithm, the same results as those from the path enumeration were obtained. This observation supports the proposition that our improved algorithm can find all efficient paths as the path enumeration method

Table 3. Efficient paths and traffic assignment pattern between Station PGY and Station BY with the path enumeration method (with $H = 0.15$).

Path No.	Paths	Number of transfer	Travel time/min	Travel cost/min	Assignment proportion
1	PGY-FXM-XD-DD-YHG-LSQ-BY	2	96.10	102.98	0.4604
2	PGY-FXM-XZM-YHG-LSQ-BY	3	97.40	108.58	0.1552
3	PGY-FXM-XZM-YHG-DZM-BY	2	97.00	109.90	0.1201
4	PGY-FXM-XZM-HLG-LSQBY	2	99.00	110.18	0.1137
5	PGY-FXM-XWM-CWM-BJZ-DZM-BY	2	100.50	113.40	0.0609
6	PGY-FXM-XD-DD-JGM-DZM-BY	2	98.50	111.40	0.0897

and result in a more realistic assignment than Dial's algorithm. Nevertheless, it should be noted the proposed algorithm in general should produce more paths than the path enumeration method. However, those additional paths are expected to have negligible effects on the final assignment results (Li *et al.* 2005). The observations from this study also support this conclusion (see below for further details). Moreover, it should be noted that the new algorithm does not require searching the whole network as path enumeration. Comparison between Table 2 and 3 shows that a practical path (Path 6) was not included by the original Dial's algorithm, but by the newly proposed one and the path enumeration.

The following sections compare the assignment results over individual links between different algorithms mentioned above, including path enumeration, the original Dial's algorithm, and the proposed new Dial's algorithm. Figure 3 shows the assignment results of the original Dial's algorithm and the path enumeration for each path link, which is determined based on the percentage of the total demand between the O–D pair: PGY and BY. It is obvious that there are quite large differences between the assignment portions resulting from the two algorithms. For example, the difference on the link DD-YHG is more than 20%, where those on links JGM-DZM and YHG-LSQ are more than 10%.

Figure 4 shows the link assignment portions of the improved Dial's algorithm with different H values, in comparison to those of the path enumeration method. It can be seen from the figure, with a small H value, the improved Dial's algorithm resulted in very unrealistic assignment patterns. As the value of H increases, the assignment results of the improved algorithm are getting closer to those of the path enumeration method. However, when the value of H increases beyond a certain range (e.g. 0.15 in this case), the influence of changes with the parameter on the assignment pattern is very small. And in general, the assignment differences can be ignored as the assigned portions to newly added paths are so small. For example, when $H = 0.05$, the improved Dial's algorithm results in only one efficient path, which is the shortest path; whereas when $H = 0.1$, it results in five efficient paths. When $H = 0.15$ or 0.2, we can obtain six and seven efficient paths respectively. However, as the searching domain increases, the travel cost of newly added paths increases and the assigned portions decrease dramatically. Therefore, when the number of efficient paths is beyond a certain threshold (in this case, the number is

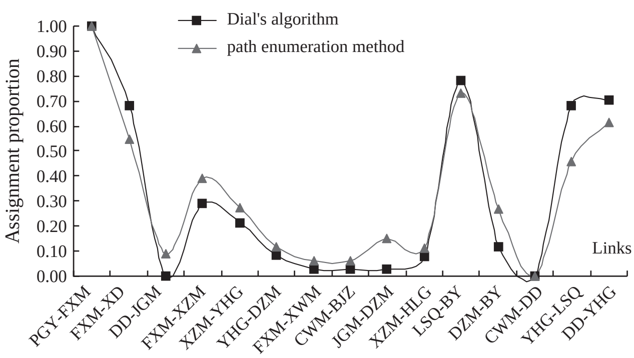

Figure 3. Comparison of assignment results of the original Dial's algorithm and the path enumeration method over individual links.

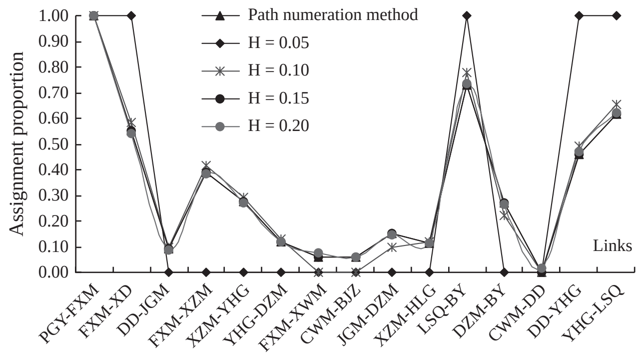

Figure 4. Comparison of assignment accuracy between the path enumeration and improved Dial's algorithm with different H values.

six), the increase in the number of efficient paths will not change the assignment pattern much and in general such influences can be ignored.

The average absolute link volume difference (AALVD) was used to evaluate the effectiveness of the different algorithms, which is defined as:

$$D = \frac{1}{m} \sum_{ij} |v_{ij}^n - v_{ij}| \tag{20}$$

Where m is the number of links in the whole network; v_{ij}^n is the assigned volume with the algorithm n on link (i, j); v_{ij} is the resulting volume from the path enumeration method for link (i, j).

Assume the total travel demand between the PGY and BY is 10,000 people per day, the AALVD between the original Dial's algorithm and the path enumeration method is 771 people, which indicates a large assignment error. AALVD was further used to evaluate the influence of different H values on the assignment results of the improved Dial's algorithm. Figure 5 presents the resulting AALVD values when different H values are used. It can be seen from the figure that it is not always the case that the algorithm results in smaller AALVD values as the H value increases. On

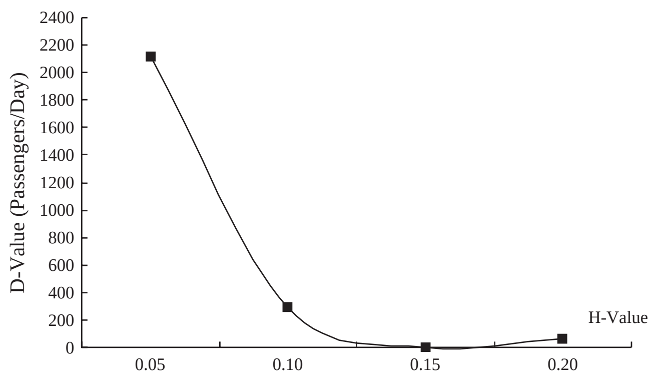

Figure 5. Resulting AALVD values of the improved Dial's algorithm with different H values.

the contrary, the parameter H with a value of 0.15 results in the lowest AALVD value. The results indicate that a reasonable H value should be obtained through either travel survey or statistical analysis of travel data. Previous research (Leurent 1997) and this study show that a value of 0.15 seems a good choice.

Table 4 presents the assigned link volumes for the different algorithms by assuming the total subway demand between PGY and BY is 10,000 people/day. It can be seen that the original Dial's algorithm and the improved Dial algorithm with small H values resulted in quite different assignment patterns than the path enumeration method, whereas the improved algorithm with large H values ($H=$ 0.15 or 0.2) simulate the path enumeration method very well, especially the one with the H value of 0.15.

7. Conclusions

Dial's algorithm is a popular method for a logit-based stochastic assignment commonly applied to non-congested networks. Because the algorithm does not require path enumeration, it has been widely applied for solving large scale network traffic assignment problems. The algorithm initializes by calculating the shortest path between each intermittent node and the origin and the destination. Then it determines whether a link can be used to establish an efficient path between the O–D pair by the condition if the starting and ending node are further away from the origin and closer to the destination. The criteria are based on the shortest-path distance of the starting and ending node to the origin and destination but never consider the cost of approaching links explicitly. Because of this limitation, Dial's algorithm was found to generate unrealistic assignment patterns, for which some low travel cost paths were not assigned traffic, whereas those with high travel cost were assigned

Table 4. Assigned link volumes from different algorithms (people/day).

Links	Path enumeration method	Dial algorithm	Improved Dial algorithm $H=0.05$	$H=0.10$	$H=0.15$	$H=0.2$
PGY-FXM	10,000	10,000	10,000	10,000	10,000	10,000
FXM-XD	5501	6798	10,000	5858	5501	5411
DD-JGM	897	0	0	955	897	882
FXM-XZM	3890	2904	0	4142	3890	3825
XZM-YHG	2753	2120	0	2931	2753	2707
YHG-DZM	1201	853	0	1279	1201	1181
FXM-XWM	609	298	0	0	609	764
CWM-BJZ	609	298	0	0	609	598
JGM-DZM	1506	298	0	955	1506	1480
XZM-HLG	1137	784	0	1211	1137	1118
LSQ-BY	7293	7849	10,000	7766	7293	7339
DZM-BY	2707	1151	0	2234	2707	2661
CWM-DD	0	0	0	0	0	166
DD-YHG	4604	6798	10,000	4903	4604	4695
YHG-LSQ	6156	7065	10,000	6555	6156	6221

(Akamatrsu 1996). This study shows the Dial's algorithm can result in very high assignment errors over individual links, compared to the path enumeration method. Some of them are over 20%.

In this study, we analyzed the problems of Dial's algorithm and proposed an improved one. We proved that the efficient path criteria used in Dial's algorithm was neither the sufficient condition nor the necessary condition of an efficient path based on the route extension coefficient, as it only looks at local node conditions. The results clearly disclose the behavioral weakness of Dial's algorithm. An improved Dial's algorithm was proposed in this study by *simultaneously* considering link travel cost and minimum travel cost of the corresponding starting node to origin and the ending node to destination, and only those paths within a smaller range beyond the shortest paths were included as 'efficient.' The improved algorithm was tested with data from the Beijing subway network and compared with the original Dial's algorithm and path enumeration method. The efficiency of the new algorithm was tested based on the different route extension coefficients and the one with $H=0.15$ was found to simulate well the results obtained from the path enumeration method.

It should be noted that in general the improved Dial's algorithm proposed in this study will produce more 'efficient paths' than the path enumeration method. However, the impact of the presence of additional paths would be minimal based on the evidence from previous studies (Li *et al*. 2005). The findings from this study support the same conclusion. Currently, the new algorithm can only be applied to the directed acyclic network. Future research will explore the possibility of extending this research to general networks.

Acknowledgements

The work described in this paper was mainly supported by grant from national natural science foundation of China (Project No. 70631001) and National Basic Research Program of China (Project Nos. 2006CB705500). Preparation of this paper is also partially funded by Natural Science and Engineering Research Council (NSERC), Canada.

References

Akamatrsu, T., 1996. Cyclic flows, Markov process and stochastic traffic assignment. *Transportation Research*, 30B, 369–386.

Beijing MTR, 2007. *Research on toll allocation model and method for Beijing subway line 4*. Beijing: Beijing Jiaotong University (in Chinese).

Bell, M.G.H., 1995. Alternatives to Dial's logit assignment algorithm. *Transportation Research*, 29B, 287–295.

Dial, R.B., 1971. A probabilistic multi-path traffic assignment mode1 which obviates path enumeration. *Transportation Research*, 5, 83–111.

Huang, H.J. and Bell, M.G.H., 1998. A study on logit assignment which excludes all cyclic flows. *Transportation Research*, 32B, 401–412.

Leurent, F., 1997. Curbing the computational difficulty of logit equilibrium assignment model. *Transportation Research*, 31B, 315–326.

Li, J., Xin, S.X., and Cai, M., 2005. Algorithm for logit network loading problem based on topological sorting. *China Journal of Highway and Transport*, 18 (4), 87–90 (in Chinese).

Li, Z.C. and Huang, H.J., 2003. Determining the efficient paths in stochastic traffic assignment. *Journal of Transportation Systems Engineering and Information Technology*, 3 (1), 28–32 (in Chinese).

Sheffi, Y., 1985. *Urban transportation networks: equilibrium analysis with mathematical programming methods.* Englewood Cliffs, NJ: Prentice-Hall.

Wong, S.C., 1999. On the convergence of Bell's logit assignment formulation. *Transportation Research*, 33B, 609–616.

Zhu, S.Y., *et al.*, 2000. Research on traffic network reliability and access road algorithm. *China Journal of Highway and Transport*, 13 (1), 91–94 (in Chinese).