A K-shortest-paths-based algorithm for stochastic traffic assignment model and comparison of computation precision with existing methods

Xuyan QIN
Doctor student
Institute of Transportation Engineering
Tsinghua University
Beijing, 100084
P.R.China
Fax: +86-10-62795339
E-mail: qinxuyan99@mails.tsinghua.edu.cn

Huapu LU
Ph.D., Professor
Institute of Transportation Engineering
Tsinghua University
Beijing, 100084
P.R.China
Fax: +86-10-62795339
E-mail: luhp@mail.tsinghua.edu.cn

Yuanyuan WANG
Master student
Institute of Transportation Engineering
Tsinghua University
Beijing, 100084
P.R.China
Fax: +86-10-62795339
E-mail: renewang99@mails.tsinghua.edu.cn

Abstract: One of the key issues of stochastic traffic assignment (STA) is to make the flow pattern consistent with the practical results. In this paper, the author makes a summary of STOCH algorithm and puts forward a new algorithm to solve the problem of stochastic traffic assignment called k-shortest-paths-based method. First, the paper discussed the widely used STOCH algorithm, especially for "single-pass" and "double-pass" procedures. Case studies with multi-OD pairs are analyzed to demonstrate the steps and advantages in "single-pass" method. Then, the paper presents the "k-shortest-paths-based method", which not only solves all the stochastic numerical models and improve some drawbacks existed in the current STOCH algorithms, but also posses merits of flexibility, though with a modestly higher computational requirements. The detail explanation on the design idea and steps of the algorithm is given. Finally, the time complexity of various algorithms mentioned in this paper is illustrated.

Key words: single-pass STOCH algorithm, K-shortest-paths, stochastic traffic assignment, comparison of algorithm, time complexity

1. INTRODUCTION

Models for traffic assignment to transportation networks simulate how Origin-Destination demand flows affect link flows according to user path choice behavior and network performances. Traffic assignment model is classified as determinate traffic assignment (DTA) and stochastic traffic assignment (STA). In China, there is rarely transportation network analysis software with

STA methods, and say nothing of deep study into the algorithm.

This paper is to report our research efforts and results on algorithm improvement of the stochastic traffic assignment model in response to such needs in China. First, the author make a detailed comparison with various algorithm for logit-based STA, especially between “single-pass” algorithm and “double-pass” algorithm through flow pattern and flow chart structure, a sample network is used to illustrate the sameness and difference of the two methods. After case study, the author concludes that “single-pass” algorithm is better than “double-pass” algorithm. Furthermore, in order to improve some drawbacks in STOCH algorithm, a new method named “k-shortest-paths-based method” is represented in detail, the author demonstrates its algorithm flow chart, and makes a complete research with the current models.

The paper consists of the following sections: Section 2 presents commonly used STOCH algorithm in practical, i.e. “single-pass” algorithm and its comparison with original algorithm (“double-pass” algorithm), whilst Section 3 describes a new algorithm called “k-shortest-paths-based method”, with the design idea as well as steps of the algorithm. Section 4 reports time complexity of three mentioned algorithms, includes “single-pass” algorithm, “double-pass” algorithm and “k-shortest-paths-based method”. Finally, some conclusions and indication for future research work are presented in Section 5.

2. SINGLE-PASS STOCH ALGORITHM

STOCH algorithm or Dial’s method efficiently implements a Logit-based choice model at the network level. The procedure assigns choice probabilities (or traffic flows) to “reasonable” paths connecting each OD pairs, therefore, strictly speaking, STOCH algorithm is only an approximate solution to Logit model. The “reasonable” path is defined generally as the one that includes only links which take the traveler further away from the origin and closer to the destination.

Consequently, the STOCH procedure includes a preliminary phase which identifies the set of “reasonable” paths connecting each OD pairs. The OD flows are then only assigned to these paths. According to different definition of “reasonable” path, STOCH algorithm is classified into two catalogs: “double-pass” and “single-pass” algorithms. In the former one, the reasonable path is defined as we described above. However, in “single-pass” algorithm, the modified criterion of the reasonable path is that: it includes links that take the traveler further away from the origin. The second part of the original requirement that these links bring the traveler closer to the destination is dropped. The steps of “double-pass” algorithm can be referred to page 288-292 [1], but we cannot find more details there about the “single-pass” procedure which favor over the original (“double-pass”) one.

The paper puts forward detailed analysis and practical procedure on “single-pass” STOCH algorithm. The algorithm module was successfully applied in the Transport Planning and Analysis Software developed by ITE (Institute of Transportation Engineering), Tsinghua University, and it is already used to forecast the traffic patterns in several China cities, such as Jining city in Shangdong province and Yingkou city in Jilin province. In the next part, we describe how “single-pass” algorithm is prior to “double-pass” algorithm in their implementation procedure by empirical evidence. Meanwhile, this paper also represents an innovative K-shortest-paths-based algorithm for stochastic traffic assignment model from the research exploration.

In fact, because of the different criteria for reasonable path in “single-pass” algorithm, more reasonable paths can be considered in assigning the traffic flows, and there are some minor differences in realizing the “single-pass” algorithm. In a word, the “single-pass” algorithm brings more reasonable paths into consideration, but not all of these links can carry flows as the situation in original algorithm.

When STOCH algorithm is employed in a realistic-size transportation network, traffic demand is usually depicted in multi-OD pair matrix. Therefore, more noticeable differences are revealed between “single-pass” and “double-pass” algorithm at the time, which always represents a more efficient implementation in “single-pass” algorithm, owing to its modest computational requirements in finding the shortest-path. To explain the profound things in a simple way, “single-pass” algorithm can assign flows from one origin node to all corresponding destination nodes at one time, which is impossible in “double-pass” algorithm. The attributes described above is the biggest difference between the two algorithms in implementation. The Comparison of their time complexity is pointed out in the next section.

The flow chart of “single-pass” STOCH algorithm can be verified by figure 1:

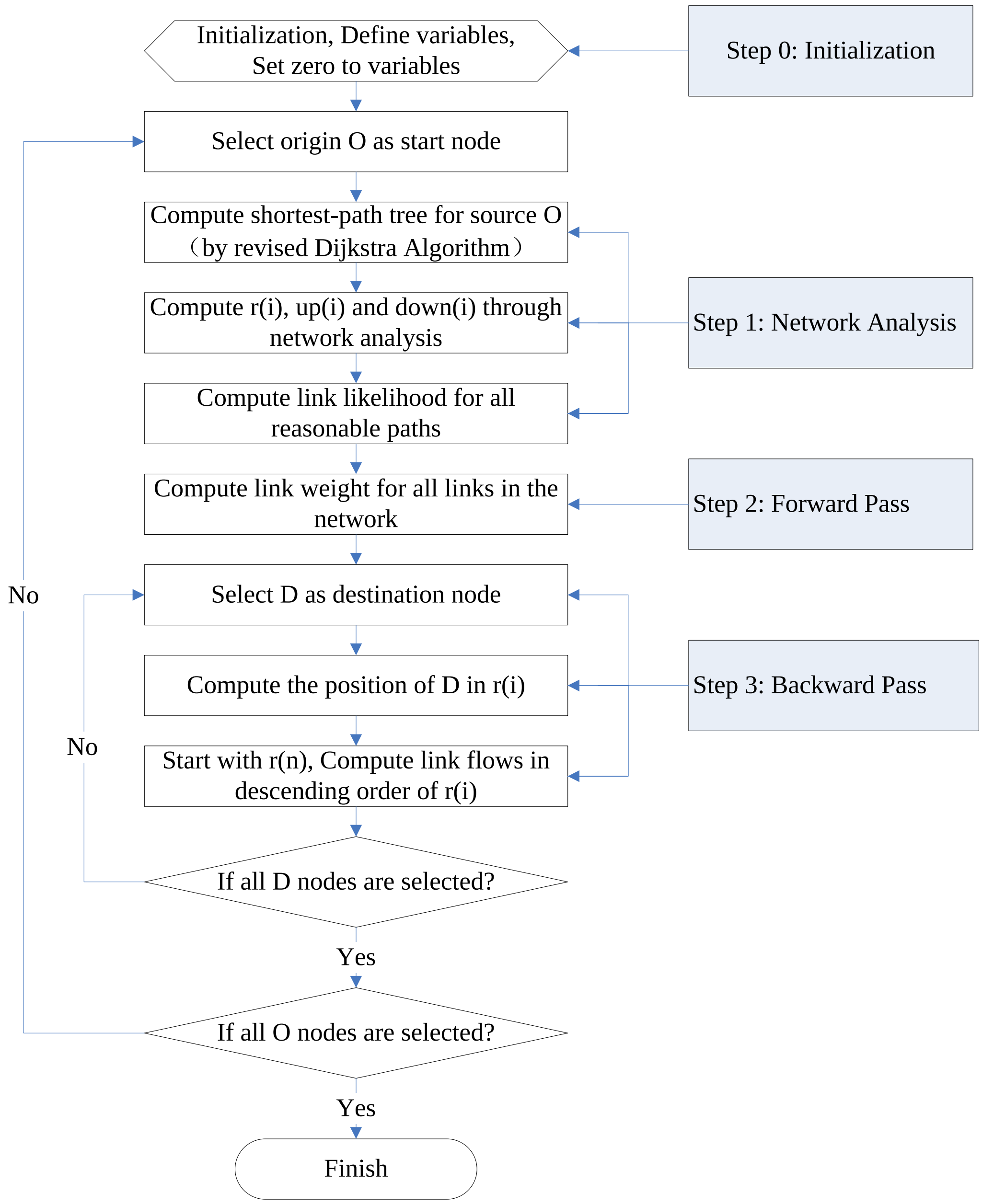

Figure1 the Flow Chart of "Single-Pass" STOCH Algorithm

The steps of the "single-pass" algorithm for multi-OD pairs are outlined below, because multi-OD pairs are more practical in the real-world and can clearly revealed the differences between 'single-pass' and "double-pass" algorithm. For the sake of understanding, each step follows a sample network. The original sample network is showed in figure 2. Since the numerical calculation is not the point in the analysis, we define the free-flow travel time as 10 of a majority of links in the network.

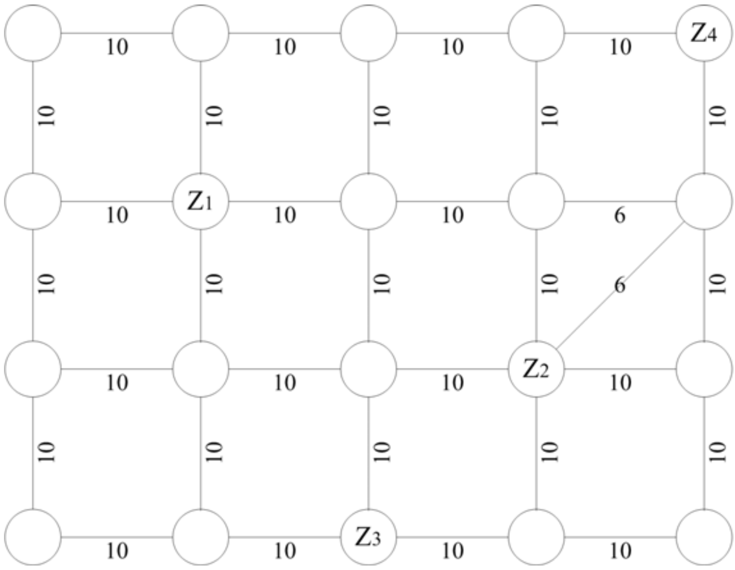

Figure2 the Original Sample Network

In the sample network, we define Z_1, Z_2, Z_3, Z_4 as centroids of TAZ (traffic analysis zone). The value of free-flow travel time is marked beside each link in figure2, we also define each link as dual-directional link, and both directions have the same free-flow travel time.

The OD flows between TAZs are listed in table1, and the traffic flow within each TAZ is not taken into consideration.

Table1 Traffic Flow between TAZs

O \ D	Z_1	Z_2	Z_3	Z_4
Z_1	0	1000	1200	1400
Z_2	1000	0	500	1200
Z_3	1200	500	0	800
Z_4	1400	1200	800	0

The steps of this algorithm for multi-OD pairs are described below:

Step 0 : Initialization

The purpose of initialization is to define the variables in the following steps, and to help initialize the network.
a) Define r(i) user for record the minimum travel time from origin node to all other nodes.
b) Define up(i) as the set of upstream nodes of all links arriving node i.
c) Define down(i) as the set of downstream nodes of all links leaving node i.

On top of initialization, we take a origin node out of the network and start the process of circulation. Please note that the procedure analyze one origin node at a time. It can be illustrated by the flow chart in figure1. When each origin node is considered, we compute the link flow for all destination nodes, which is totally different in “double-pass” algorithm. In “double-pass” algorithm, the procedure take one OD pair into consideration at a time, and it obviously consumes more time in calculation.

Step 1 : Network Analysis

We start with a determinate origin node, and then compute the following variables:
(a) Compute the shortest travel paths from origin node to all other nodes. We adopt Dijkstra shortest-path method to calculate;
(b) Compute value of r(i) for each node;
(c) According to the neighboring relationship, determine the set of up(i) and down(i) for each node;
(d) Compute the “link likelihood” L(i->j) for each reasonable link, where

$$L(i \rightarrow j) = \begin{cases} e^{\theta[r(j)-r(i)-t(i \rightarrow j)]} & \text{if } r(i)<r(j) \\ 0 & \text{otherwise} \end{cases}$$

Assume that we take Z_1 in the first circulation, Figure3 shows the link likelihoods (with θ = 1) for all the links in this example. Note that the labels beside the link are the link likelihood, and the values written in each circle are the r(i) for each node. Because each link has dual-direction, there should be link likelihoods on both directions. However, if one direction of the link belongs to a reasonable path, its link likelihood will be non-zero. Meanwhile, the link likelihood of inverse direction must have zero value. Therefore, for the simplicity, we only label the non-zero value in figure3 and its direction.

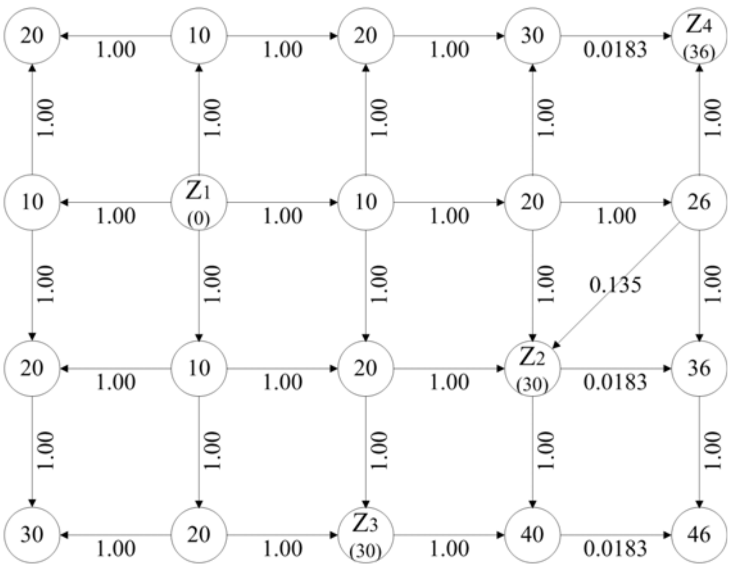

Figure3 the Link Likelihoods (for θ = 1 and Z_1 as the Origin Node)

Step 2 : Forward Pass

The procedure of forward pass is the same with "double-pass" algorithm, so the detail is omitted here. Note that consider nodes in ascending order of r(i), and start with the origin node. The link weights corresponding to the link likelihoods shown in figure3 are shown in figure4.

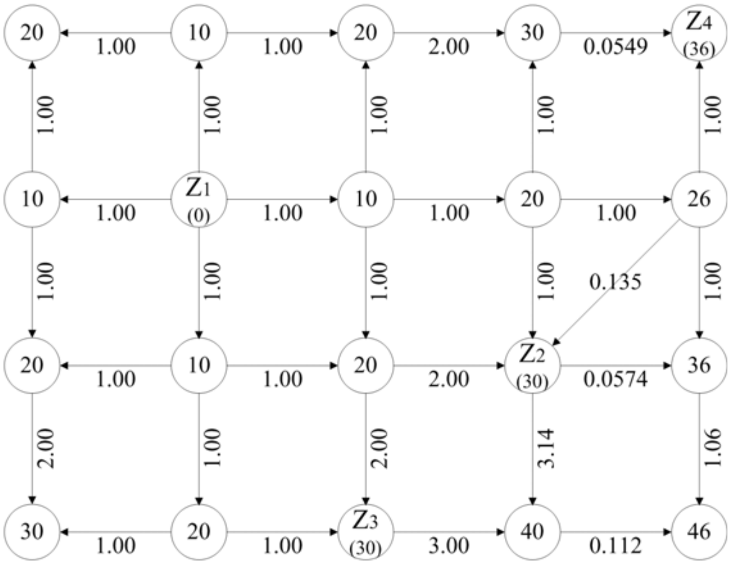

Figure4 the Link Weights After Forward Pass (for θ = 1 and Z_1 as the Origin Node)

Step 3: Backward Pass

The calculation of backward pass is totally different from the "double-pass" STOCH algorithm. In "double-pass" algorithm, we start with the destination, and then consider nodes in distant ascending order from the destination, compute the link flow. This step is applied iteratively until the origin node is reached. In "single-pass" algorithm, we could not obtain nodes order from destination, because the shortest-path only is computed once.

In order to implement backward pass, the author consider nodes in descending order of r(i). It can be illustrated that because the destination is one of the nodes in network, we could find its position in r(i), and assume that it is r(n). We start with r(n), according to the descending order of r(i), each node is considered, and the sum of the flow variables is taken over all links emanating at the upstream node of the link under consideration. When each node is considered, the computation is the same as in the "double-pass" algorithm, so the procedure is not discussed here. Assume Z_2 is considered in the first place, then the link flows for the sample network under consideration are given in Figure5, the arrow on one side of the link refers to the direction as before, and the value beside the link means traffic flow assigned to the link. As we mentioned above, the inverse direction of the arrow has zero flow.

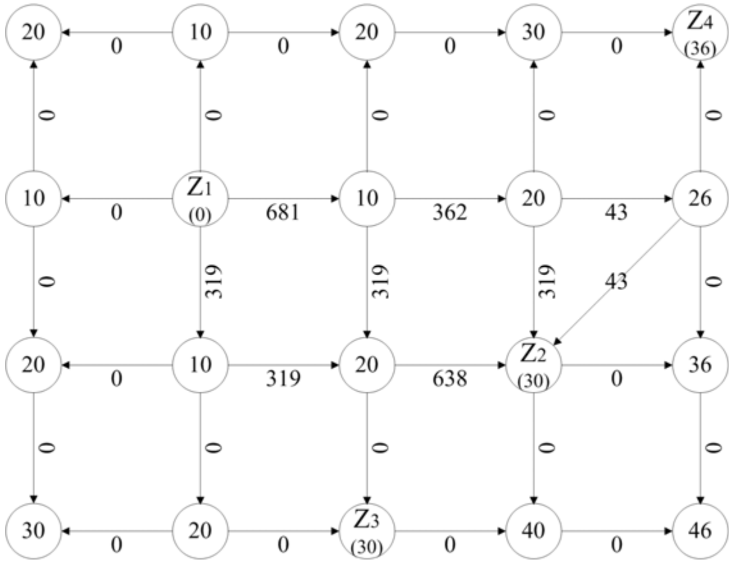

Figure5 the Link Flows (Z_1 as the Origin Node, Z_2 as the Destination Node)

A complete comparison of STOCH algorithm needs to answer the following two questions:

Question 1: if "double-pass" algorithm is employed, what is reasonable path between the OD pair Z_1 and Z_2?

Question 2: if "double-pass" algorithm is used, how about the link flows in the network? What is the main difference between the two results?

The answer to the first question is that the reasonable paths in "double-pass" algorithm are the subset of the ones in "single-pass" algorithm, that is to say those reasonable paths in "double-pass" algorithm all belongs to the set of the paths in "single-pass" algorithm. It is illustrated in figure 6. The red arrows show reasonable paths and their directions in "double-pass" procedure, while the blue arrows refer to the corresponding ones in "single-pass" procedure. Under the new definition, we find that the reasonable paths in "single-pass" procedure cover nearly all the links in the sample network than in the "double-pass" procedure. In fact, when the network becomes larger and larger, the reasonable paths cover every link. Because no link with negative travel time exists in the real road network, the link always has a direction obey the definition of the reasonable path. However, the more reasonable paths, the more calculation is executed for link likelihood and link weight, and even the more computation in sorting order of the nodes, all of the work is time-consuming. Although the "single-pass" algorithm takes advantage of the nature of minimum-path procedures and half the time cost, which is prior to calculate two minimum-path calculations for every OD pair in "double-pass" algorithm, it still requires other time consuming computations. Therefore, only carefully study further into the two methods, can we come to the conclusion about the time complexity about these mehods.

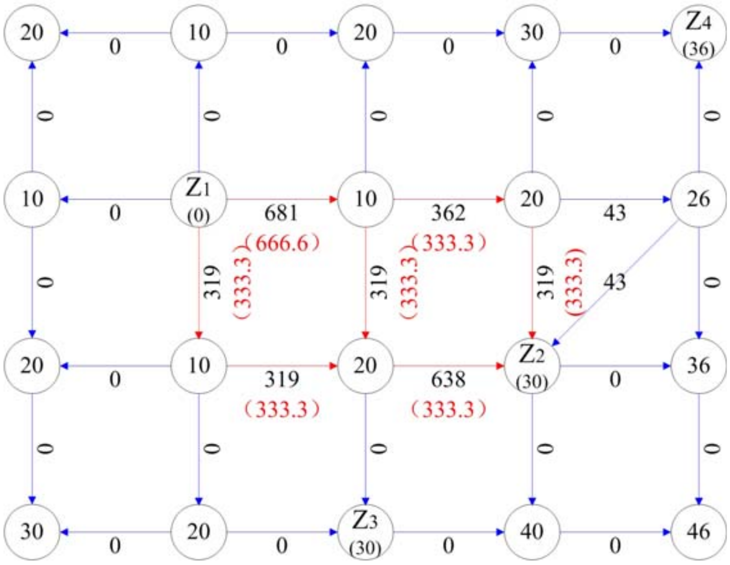

Figure6 Assignment of traffic flows to the sample network (comparison between "single-pass" and "double-pass" algorithm)

According to question 2, all reasonable paths of "double-pass" algorithm result in traffic flow on them, but this situation does not happen in the case of "single-pass" algorithm. To illustrate the problem, consider the sample network shown in figure 6. The number in black above each link shows the result of traffic flows under "single-pass" procedure, while the red number in the parentheses beside the red arrows refer to the corresponding results in "double-pass" procedure. The flow pattern of the "single-pass" algorithm differs from the one generated by the "double-pass" algorithm. According to the modified criterion, more paths are reasonable, but only a few of them obtain traffic flows. The reason of the phenomenon is due to its mechanics, that the algorithm loads the upstream links of a certain node, and assigns the OD flows on them, so the links which belong to upstream path tree of destination obtain the flows, while the other links have non flows. However, whether OD flow can be generated on a reasonable link is determined by other factors, such as location of destination and network attributes. When applied to some extreme situation of network, "single-pass" and "double-pass" may have the same results.

The above-mentioned deficiency of the "single-pass" algorithm may decrease its efficiency, because large amount of calculation on link likelihood, link weight and sorting order for each node. Why not only calculate the links that may generate traffic flows to reduce the amount of computational effort? Up to the present, we only take one origin and its corresponding destination into consideration, what we have talked about is a good solution under this single OD pair situation. In fact, when applied to the realistic-size network with multi-OD pairs, the link likelihood and link weight of the whole network generated in Step1 and Step2 are not only used for a single destination, but for all possible destinations. Anyhow, if the network information is not used by the first selected destination, it would be used by the next one, and they are useful until to the last one. In other words, most of the calculations are used due to a wide spread distribution of destination nodes in a practical network, therefore, the increased computational costs at the preliminary steps are not wasteful at all. Otherwise, if we want to avoid the consumptive computation, we need to record all the possible used paths for each destination at first, which is accomplished, however, at increased computational costs.

As mentioned above, because reasonable path defined by “single-pass” STOCH algorithm has no relationship with destination, the link likelihood and link weight computed for an origin can be employed by any destination. In the process of iteration, all destinations share the same network information generated by one origin. However, this is not the case in the “double-pass” algorithm, two shortest-path calculations for every OD pair in the network owing to the fact that the reasonable path is related with the destination, not much the same information could be executed for each destination. This problem may be solved by preceding all destinations under one origin, but attempts demonstrate that it is not effective at all.

In step 3 of “single-pass” algorithm, after one OD flow is assigned on the network, this step is applied iteratively until the last destination node is reached, and the link flows are updated to the sum of the former one and the new generated one. Note that the next origin node is considered until all destinations are reached, then repeat step1 to step3, until the last origin is selected, where is the end of the procedure. To illustrate the process, consider the sample network in figure2, after Z_2 is analyzed as destination node in step3, Z_3 and Z_4 are considered one by one. Then Z_2 , Z_3, and Z_4 are treated as origin for the circulation of step1 to step3 respectively. This concludes the description of the mechanics of “single-pass” STOCH algorithm.

3. K-SHORESTEST PATHS METHOD

Because the definition used by the STOCH algorithm has deficiencies in selection of the reasonable paths, some important paths may be missed, and no flow will be assigned to these paths, such as the rounding city expressway shown in figure 7a (marked in redline), and these paths are not considered by “double-pass” STOCH algorithm because they do not meet the requirement of reasonable path. In China, since many cities have the city-around expressway, this omission will make big effect on the traffic assignment.

The STOCH algorithm may bring some unreasonable problems. Take the red line in figure 7a for example, that it does not take the traveler closer to the destination, which is requested by “double-pass” STOCH algorithm, so it is not a reasonable path. But if deleting the node in figure 7a only, the path in red changes to figure 7b. Then, the red line will get the flow, because it meets the requirement. This phenomenon brings in problem, that the same road gets different flow under the same situation due to different judge and operation made by the planner.

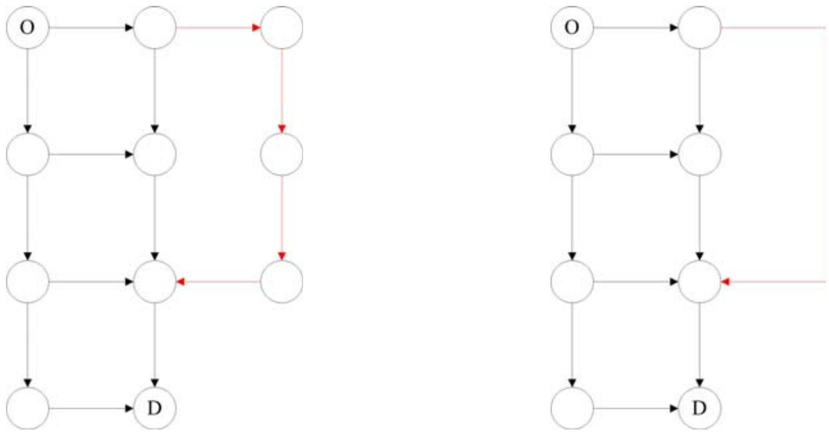

Figure 7a Figure 7b

The two questions mentioned above exist in “double-pass” STOCH algorithm only, single-pass STOCH algorithm is much better in this aspect, have no such problems.

Besides, the STOCH algorithm can not take the delay effect in a intersection into consideration, that is to say, the STOCH algorithm misses the different delay in the traffic assignment, which influences the accuracy severely. However, the delay at intersection affects the traveler’s choice much, and then the flow pattern changes. In determinate traffic assignment, the delay is considered by improving the shortest-path algorithm. But the delay effect is not considered into the STOCH algorithm up to the present, because of its particularity. Furthermore, the STOCH algorithm is not based on the path analyze, it is difficult to get the flow between OD during the assignment process, which makes complex research. Meanwhile, the STOCH algorithm is used to the Logit model only, when Logit model has some deficiencies compared with Probit model. All the questions presented above restrict the use of STOCH algorithm.

In this paper, a new method is introduced, basing on the k-shortest-paths, which attempts to get a better assignment result by changing the shortest-path method in the basic level. As it is known for all that algorithms used in STA need to calculate the shortest-path. Stochastic traffic assignment is defined assuming that each traveler may perceive different path costs, and the methods used for solve the STA, either the STOCH algorithm or the Mento-Carlo simulation, all compute the new path based on the single shortest-path. Shall we get more than one path in the process of solving the shortest-path? The answer is k-shortest-paths.

The flow chart of k-shortest-paths is shown in figure 8.

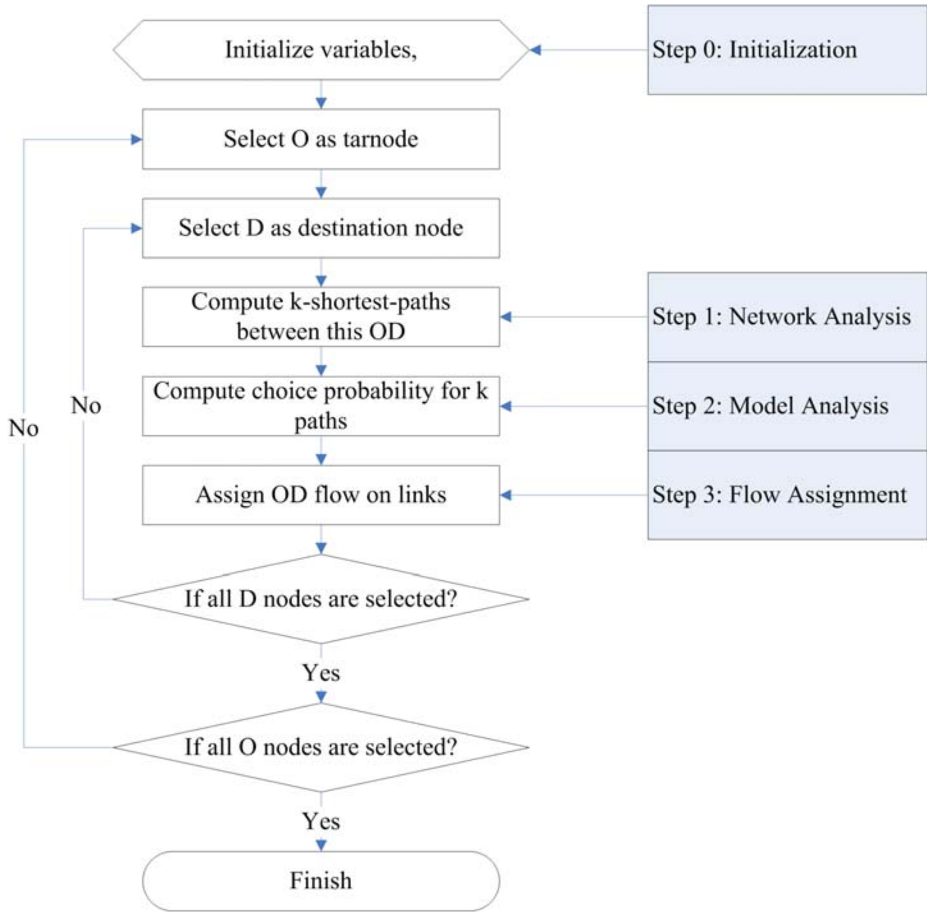

Figure 8 the Flow Chart of K-Shortest-Paths Algorithm

Step 0: Iinitialization

At this stage, initialize some variables that are used later, such as the variables used to record the k-shortest-paths, record the probability of each path, and record the flow that each path gets and so on.

Step 1: Network Analysis

The computation of the k-shortest-paths is processed at this stage. There are some methods widely used in the world, but we do not describe them in detail here, these methods can be found in [2][3][4]. Note that while solving the shortest-paths, we also get the travel time of the paths, and it will be used in the module later together with the k-shortest-paths to assign the flow to each path.

Step 2: module analysis

Define the travel time as T(i), i=1,2,…k. In this step, the probability of each path will be calculated using T(i) and the module, if Logit modle is employed, the choice probability of path i is expressed as: $$P_i = \frac{e^{T(i)}}{\sum_{l=1}^{k} e^{T(l)}}$$

Differing from the STOCH algorithm, the k-shortest-paths method is not restricted only for the Logit model, bur for various numerical models under the condition that the choice probability of each path must be solved by using T(i). Like the Logit model, the probability is easy to get because its simple mathematical formula, and it is not case for Probit model due to its complicated integration.

Step 3: Flow Assignment

Assuming the OD flow is q, then the flow of path i is: x(i) = q * Pi
After step 3, we get the value of next D node, repeat the procedure of step1~step3, until all D node is solved, then take out another O node and repeat the procedure until all O node is solved. Finally, the traffic assignment based on the k-shortest-paths is done. Notice that the iterative process may be different because of the different methods to solve the k-shortest-paths. The method presented in reference [2] can solve the k-shortest-paths between one node to any other nodes. Under this condition, there is no need to compute the k-shortest-paths analysis to each OD pair, one time for one origin node is enough.

In conclusion, the main merits in k-shortest-paths method are:

(a) Good flexibility. First, the proper k-shortest-paths method can be chosen within the development of this algorithm. Meanwhile, the method specially used to traffic network, repeat links is also under research. Second, this method is not limited to Logit model only, a variety of models can be used by it. Actually, the k-shortest-paths traffic assignment procedure presented in this paper is an open frame, lots of algorithm and model can be used in it.

(b) For a single OD pair, iteration is not needed, which is same with the STOCH algorithm. And it is essential to improve the efficiency. Mento-Carlo simulation need to repeat the iteration many time to get the flow assignment between one OD pair, which slows down its efficiency, and thus limits the use of it in large-scale road network.

(c) Since k-shortest-paths method takes the path as the analysis unit in the basic level, so the path flows, delay at intersection can be considered into the procedure.

4. ANALYSIS OF TIME COMPLEXITY

Assume a network with n nodes, m links, x origins and x destinations, the time complexity for three algorithm are listed in table 2:

Table2 the Time Complexity of Three Algorithms

double-pass	**single-pass**	**k-shortest path**

$xn\log n$	$xn\log n$	$x(m+n\log n+kn)$

Where: The time complexity of k-shortest path please refers to [2]

As for a city-sized network, the ratio between number of links and nodes is probably 2:1. Meanwhile, x and n are usually closely related, the ratio between them may reach 1:5 for a small city (i.e. n<100), and the ration becomes 1:10 to 1:100 for a large city (i.e. n>1000). After the simple substitution of x for n, the new time complexity is shown in table3.

Table3 the Time Complexity of Three Algorithms (After Substitution)

double-pass	**single-pass**	**k-shortest path**
$n^2\log n$	$n^2\log n$	$n^2\log n+kn^2$

Though the iteration process is different between "single-pass" and "double-pass" algorithm, table3 demonstrates that the same time complexity for them and it is partly because the constant is ignored in the expression. However, we come to the conclusions that "single-pass" algorithm is more efficient through their iteration and practical test, simply because the constant in time complexity of "single-pass" is smaller than that of "double-pass". For example, consider the above situations, "single-pass" executes x times shortest-path calculation in the whole assignment, while "double-pass" employs 2x times, and the ratio of sorting order computation between the two methods is 1:2, too.

The time complexity of k-shortest path is only modestly higher than that of STOCH algorithm, but it holds other advantages that can not find in STOCH algorithm, which make it favor the other two algorithms in practical use.

5. CONCLUSION

After discuss the comparison between "single-pass" algorithm and other methods in aspects of assignment results and time complexity, we conclude that "single-pass" prevails over "double-pass" when considering their results and time costs.

There are disadvantages that can not be improved through revising the STOCH algorithm. STOCH algorithm can not consider the intersection delay and acquire the path flow from the process. According to advantages of k-shortest path method, it is a promising method that could substitute the STOCH algorithm though it bears lower time efficiency. The nuclear idea of this model is to.

However, this is only an aspect of this problem. There are still many tasks left for us to study, the most important ones of which are: 1). Increase the efficiency of the k-shortest path method 2) Apply the k-shortest-path-based method for realistic-sized network and make improvement on it.

REFERENCES

1. YOSEF SHEFFI (1984) **Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods**. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632
2. David Eppstein (1997) Finding the k Shortest Paths, **SIAM Journal on Computing Volume 28, Number 2**, 652-673.
3. T. H. Byers and M. S.Waterman(1984) Determining all optimal and near-optimal solutions when solving ,shortest path problems by dynamic programming. **Operations Research** 32, 1381–1384
4. S. E. Dreyfus(1969) An appraisal of some shortest path algorithms. **Operations Research** 17, 395–412
5. Lu Huapu (1998) **Transportation Planning Theory and Method,** Tsinghua University Press**,** P.R.C., Beijing
6. Knuth D E(1973) **The Art of Computer Programming volume 1/ Fundamental Algorithms**, Wesley Publishing Company, Inc.