

# The Application of the Auction Maximum Flow Algorithm in Traffic Flow Assignment

M.Z. Tian

College of Mathematics  
Qingdao University, Qingdao  
China

C. Xu

College of Mathematics  
Qingdao University, Qingdao  
China

**Abstract-**Researching the problem of heavy traffic, by defining the degree of congestion, design the auction of maximum flow algorithm in the distribution of traffic network. Finally, establish the model to show its application in the instance.

**Keywords-**the maximum flow; the degree of congestion; the auction algorithm; the distribution of traffic network

## I. INTRODUCTION

In 1979, Bertsekas puts forward the concept of the auction algorithm for the first time, and used to solve the assignment problem. It mimics the traditional way of auction.

Nowadays, the problem of traffic is becoming more and more serious. How to choose a most efficient path has become a difficult problem. In order to save time, people will generally choice the shortest route, but they always ignore that the time we have wasted by the road congestion has already made our decision no longer be optimal.

For this, we give the auction of maximum flow algorithm. In this algorithm, we will choice the route which has the biggest capacity(has lighter congestion) relatively, and then get the optimal choice.

## II. RELEVANT DEFINITIONS

Definition 2.1<sup>[1]</sup> An undirected graph  $G$  is an ordered tuple of  $(N, E)$ , denoted by  $G = (N, E)$ .  $N = \{n_1, n_2, \dots, n_n\}$  is a nonempty set of vertices, while  $E = \{e_{ij}\}$  is a nonempty set of edges, where  $e_{ij}$  is a disordered binary set  $(n_i, n_j)$ , denoted by  $e_{ij} = (n_i, n_j)$ .  $e_{ij}$  connects  $n_i$  and  $n_j$ , while  $n_i$  and  $n_j$  called the endpoint of  $e_{ij}$ . But if we limit  $n_i$  as the tail and  $n_j$  as the head in each arc  $(n_i, n_j)$ , then  $G$  will called a directed graph.

Definition 2.2<sup>[2]</sup> We called  $c(n_i, n_j)$  as the capacity of arc  $(n_i, n_j)$ , and  $f(n_i, n_j)$  as the flow of arc  $(n_i, n_j)$ , if there is a real number  $f(n_i, n_j)$  on each arc  $(n_i, n_j)$ ,

which meets  $0 \leq f(n_i, n_j) \leq c(n_i, n_j)$ . In addition, we use  $l_{ij}$  represent the distance between  $n_i$  and  $n_j$ .

Definition 2.3 We call  $a_{ij} = \frac{1}{c(n_i, n_j) - f(n_i, n_j)}$ ,  $0 \leq f(n_i, n_j) < c(n_i, n_j)$  as the degree of congestion in each arc  $(n_i, n_j)$ , and when  $c(n_i, n_j) = f(n_i, n_j)$ , we set  $a_{ij} = 2$ . For convenience, we'll use  $c_{ij}(f_{ij})$  instead of  $c(n_i, n_j)(f(n_i, n_j))$ .

Definition 2.4<sup>[8]</sup> For each point  $n_i$  in the graph, all the arcs which from  $n_i$  point to  $n_j$  ( $i \neq j$ ) called the forward arc, and the arc which from  $n_j$  ( $i \neq j$ ) point to  $n_i$  is called the backward arc.

Definition 2.5<sup>[10]</sup> If each point is different from others in path  $P$  except the starting point  $n_1$  and ending point  $n_n$ , then we call this path as a simple path. If a path  $P$  has the same starting point and ending point, then we call it as a circle.

Definition 2.6<sup>[11]</sup> Give a graph  $G$ , if the node  $u$  has an arc  $(u, v)$  to the node  $v$ , then we call the graph  $G$  is connected. If one node  $u$  has an arc to all the nodes except itself, we call  $u$  as the root of  $G$ .

## III. THE DESIGN OF ALGORITHM

At first, we establish a directed graph  $G = (N, E, F, C, L)$ . As mentioned above,  $N = \{n_1, n_2, \dots, n_n\}$  is the set of vertices,  $E = \{e_{ij}\}$  is the set of edges,  $F = \{f_{ij}\}$  is the set of flows,  $C = \{c_{ij}\}$  is the set of arcs capacity and  $L = \{l_{ij}\}$  is the set of distance. Then we use  $a_{ij}l_{ij}$  to represent the time we'll cost between  $n_i$  and

$n_j$ . And use  $P_i$  to represent the time we need between  $n_i$  and endpoint<sup>[3,4]</sup>.

For convenience, we have the following assumptions:  
There only have original circle in the network;

Assume that each node has at least one arc to the next node, otherwise, we will establish one arc between this node to the end node, and assume the length of the regulations is infinite;

Assume that there can only have one arc between two nodes.<sup>[9]</sup>

We use  $n_1$  and  $n_n$  to represent the starting and end point.  $P = (n_1, n_2, \dots, n_i)$  represent a path from  $n_1$  to  $n_i$ , and define it as the original path. In the iterative process, we'll keep  $P$  as the original path<sup>[5]</sup>.

If  $n_{i+1} \notin P$  and  $n_{i+1} \neq n_n$ , but arc  $(n_i, n_{i+1}) \in E$ , then we use  $P = (n_1, n_2, \dots, n_{i+1})$  to instead  $P = (n_1, n_2, \dots, n_i)$ , until  $n_{i+1} = n_n$ . If  $n_1 \notin P = (n_2, n_3, \dots, n_i)$ , we'll use  $P = (n_1, n_2, \dots, n_{i-1})$  to instead it<sup>[6]</sup>.

In the iterative process, we must meet the complementary slackness<sup>[7]</sup>:

$$p_i \leq a_{ij}l_{ij} + p_j, \forall (n_i, n_j) \in E;$$

$$p_i = a_{ij}l_{ij} + p_j, \forall (n_i, n_j) \in P.$$

We assume that the original path:

$$P = (n_1), p_i = 0, \forall n_i \in N;$$

Then we make  $n_i$  be the endpoint of  $P$ . If  $p_i < \min_{(n_i, n_j) \in E} \{a_{ij}l_{ij} + p_j\}$ , we turn to the step 1. Otherwise, go to step 2.

Step 1: Make  $p_i = \min_{(n_i, n_j) \in E} \{a_{ij}l_{ij} + p_j\}$ . If  $i \neq 1$ , then we shrinkage  $P$ , go to the next round of iteration;

Step 2: We use  $n_j$  to expand  $P$ ,  $n_j = \arg \min_{(n_i, n_j) \in E} \{a_{ij}l_{ij} + p_j\}$ . If  $n_j = n_n$ , This  $P$  is the optimal choice. Otherwise, go to the next round of iteration.

In the route's extend and contract process, we can see that  $P$  is the original road between 1 to  $n_j$ . In fact, if this

conclusion is not right, we put the  $n_j$  join in the route  $P$ , then  $P$  will become a circle, and for each arc in  $P$  has  $p_i = a_{ij}l_{ij} + p_j$ . We will get the length of  $P$  is zero, and this is contradict to our assume.

#### IV. AN EXAMPLE

In order to facilitate understanding, we present a simple traffic flow graph 1. As we can see from the graph, There are three parameters in each edge: the first one on behalf of the capacity  $c_{ij}$ , the middle one means the number of flows  $f_{ij}$ , the last one present the distance between two sides  $l_{ij}$ .

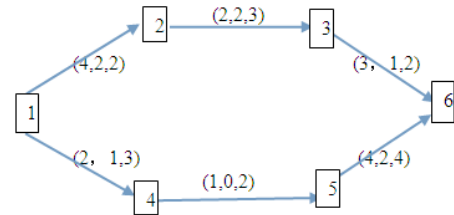


FIGURE I. THE ORIGINAL TRAFFIC FLOW GRAPH

Firstly, we simplify the graph above, then we'll get the following:

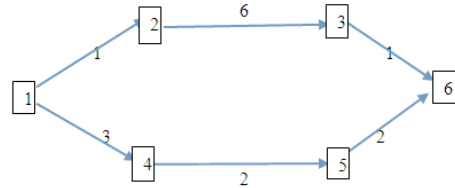


FIGURE II. THE SIMPLY TRAFFIC FLOW GRAPH

Each of this number means the new parameter we have defined above, we use  $a_{ij}l_{ij}$  to present it.

We will get the following iterative process:

TABLE I. THE ITERATIVE PROCESS

The number of iterations	The path before iteration	The $p_i$ before iteration	The iterative process
1	(1)	(0,0,0,0,0)	Contract in 1
2	(1)	(1,0,0,0,0)	Extend to 2
3	(1,2)	(1,0,0,0,0)	Contract in 2
4	(1)	(1,6,0,0,0)	Contract in 1
5	(1)	(3,6,0,0,0)	Extend to 4
6	(1,4)	(3,6,0,0,0)	Contract in 4
7	(1)	(3,6,2,0,0)	Contract in 1
8	(1)	(5,6,2,0,0)	Extend to 4
9	(1,4)	(5,6,2,0,0)	Extend to 5
10	(1,4,5)	(5,6,2,0,0)	Contract in 5
11	(1,4)	(5,6,2,2,0)	Contract in 4

<b>12</b>	(1)	(5,6,4,2,0,0)	Contract in 1
<b>13</b>	(1)	(7,6,4,2,0,0)	Extend to 4
<b>14</b>	(1,4)	(7,6,4,2,0,0)	Extend to 5
<b>15</b>	(1,4,5)	(7,6,4,2,0,0)	Extend to 6
<b>16</b>	(1,4,5,6)	(7,6,4,2,0,0)	End

At last, we get the optimal path  $P = (1, 4, 5, 6)$ .

## V. CONCLUSION

In this paper, from the heavy traffic problem, we define the degree of congestion, starting from the traffic flow network, establish the algorithm and use example to verify its correctness.

The auction algorithm is a new algorithm, from the researching for it in recent years, we have already use it to a lot of network problems, and now it is becoming a comprehensive algorithm. The auction theory has already be used to develop the software. It has broad prospects in network problem, not only in the shortest path problem, but also in transport problem and a series problems. With the continuously explore for auction theory, we'll find more excellent algorithm and apply them to a wider range of areas.

## REFERENCES

- [1] Huang G, Zhao Q. An auction-based dispatch algorithm for deregulated power systems. IEEE, 2000, 2: 1220-1225.
- [2] J.Y. Wang, W. Wang. Application of Auction Algorithm for the Shortest Paths in Traffic Flow Assignment[J]. Journal of transportation and information technology, 2007, 5(3): 16-20.
- [3] Zenkevich N.A. Auction games and pricing process. In Control Applications of Optimization 2000 vol.2:65-72.
- [4] Fujisaki M. Auction information transmission processing: U.S. Patent 4,1988,12(6):789-928.
- [5] Myerson R B. Optimal auction design[J]. Mathematics of operations research, 1981, 6(1): 58-73.
- [6] Allevi E, Gnudi A, Konnov I V. Combined methods for dynamic spatial auction market models[J]. Optimization and Engineering, 2012, 13(3): 401-416.
- [7] Lee C C, Ho P F, Hwang M S. A secure e-auction scheme based on group signatures[J]. Information Systems Frontiers, 2009, 11(3): 335-343.
- [8] Vartiainen H. Auction design without commitment[J]. Journal of the European Economic Association, 2013, 11(2): 316-342.
- [9] Jeziorski P, Krasnokutskaya E. Dynamic Auction Model with Subcontracting[J]. 2013,4(07):39-46.
- [10] Peng K. Efficient proof of bid validity with untrusted verifier in homomorphic e-auction[J]. IET Information Security, 2013, 7(1): 11-21.
- [11] Edalat N, Tham C K, Xiao W. An auction-based strategy for distributed task allocation in wireless sensor networks[J]. Computer Communications, 2012, 35(8): 916-928.