Problems of Optimisation of Flows Distribution in the Network

Naumova N. A.

Kuban State Technological University, Krasnodar, Russia (350072, Krasnodar, street Moskovskaya, 2-A)

Abstract The purpose of this article is to provide a stochastic model of network flows based on the Erlang time distribution for vehicles moving in succession, which allows us to describe the flows of high density. We used a graph representation and introduced a structure of matrices $A_{STREETS} \cup B_{INTERSECTION}$ to store all necessary information about the network flows for analytical modelling. In this paper, the classification of network nodes is given as well as the criteria of optimisation of flows distribution in the network nodes. We provide an algorithm of numerical method to find out the optimal parameters of control for the type 2 node. Using the graph representation of the model, we developed methods of determination of the optimal scheme of flows distribution within the network.

Keywords Transportation Network, Network Flows, Mathematical Model, Graph Representation, Node, Optimisation

1. Introduction

The accelerating increase in number of car owners results in considerable growth of traffic volume, regular traffic congestion and advance in the cost of motor trucking. Cities usually develop according to the following consistent pattern: at an early stage the directions of arterial streets are laid but later the urban transportation infrastructure itself starts dictating the directional development. Therefore, optimal planning of networks and optimization of paths gain in importance. The problem of rational employment of already existing urban transportation networks by optimal organization of traffic is also urgent.

Mathematical models applied for analyzing transportation networks vary according to the problems solved, mathematical apparatus, data used, and specification of traffic description (e.g.[1-2, 4-9, 16, 17]). The first macroscopic model was suggested by M. Lighthill and G. Whitham[8] in the middle of the last century parallel with the first microscopic models (‘follow-the-leader’ theory) which explicitly derived an equation of motion for each individual vehicle (A. Reshel, L. Pipes, D. Gazis and others).

Frank A. Haight[5] was the first to establish the mathematical investigation of traffic flow as a separate section of applied mathematics. At present there is voluminous literature on the subject.

In order to efficiently control urban traffic flows and choose the optimal solutions of designing transportation networks we should consider a wide range of the flow features as well as the laws of impact of both external and internal factors on dynamic characteristics of traffic flow (e.g.[9],[16],[17]). The problems arise due to instability of traffic flow and divergence of the criteria of traffic control quality. The travel time for a concrete path is made up of the delays at intersections and the time of motion from one intersection to another. By reducing the waiting time at intersections we could optimize the total travel time. Therefore, developing a microscopic model of transportation dynamics in the nodes and estimation of their influence on the network flows distribution seems to be relevant.

2. A Graph Representation of Transportation Network

In stochastic models a traffic flow is defined as the result of interaction of vehicles on the segments of transportation network. A network is a graph each edge of which is assigned to a certain number. A flow is a certain function prescribed for the graph edges (e.g.[12]). In the case studied, the flow in the graph is given as a function of density of arrival distribution (arrival times of service requests).

We hold that time intervals distribution in each customer flow (channel) obeying the Erlang distribution (e.g.[11]) is:

$$f^{(k)}(t) = \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!} \quad (t > 0).$$

(1)

This law allows us to describe flows of rather high density. For example, the assumption of time intervals distribution for vehicles moving in succession according to the Erlang distribution is true for the intensity of 500 vehicles per hour on each lane.
We will refer to the network flows as ‘non-conflict’ if they are not crossed in the given sector of the network, and as ‘conflict’ otherwise. We will consider the node-points – the points of sources or consumption of information and those of conflict flows crossing – to be the vertices of a graph. The node-points are formed by crossing of multichannel lines.

Let us consider a node-point (NP) in which the conflict flows are crossed as follows: a number of flows (the main ones) are freely passing the NP. The customers of the rest of flows (the secondary ones) are expecting sufficient time intervals between arrivals of main flows in order to cross the NP. We will call such a node-point a ‘type 1 node’, as in[10, 13, 15].

Let:

\[ \begin{align*}
\mathcal{E} & = \text{set of graph edges,} \\
\mathcal{V} & = \text{set of nodes.}
\end{align*} \]

Then, a edge is part of a multichannel line between two nodes. Assign the identification numbers \( W\_i \) to the lines. Then \( STREET\_i = \bigcup_{j \in W} \mathcal{E}_j \).

And the graph could be given as the following combined matrices:

\[ A_{STREETS} = (S_1, S_2, S_3, S_4, \text{Contr}, \text{Pr}, \text{Len}, \text{Col}, \text{AL}, \text{AS}, \text{AR}, \lambda_A, k_A, \ldots, \text{BL}, \text{BS}, \text{BR}, \lambda_B, k_B, \ldots) \]

where

1) \( N\_e \) = number of the matrix line \( A_{STREETS} \) corresponding to the number of the graph edge linking Nodes 1 and 2 (the number of lines equals the number of edges);
2) \( S_1 \) and \( S_2 \) = intersected lines forming Node 1;
3) \( S_3 \) and \( S_4 \) = intersected lines forming Node 2;
4) \( C \) = node type;
5) \( \text{Pr} \) = priority (main or secondary line);
6) \( \text{L} \) = length of edge;
7) \( \text{Col} \) = number of flows on the edge;
8) \( \text{AL} \) = admissibility of turning to the left from Direction A at Node 1;
9) \( \text{AS} \) = admissibility of direct motion from Direction A at Node 2;
10) \( \text{AR} \) = admissibility of turning to the right from Direction A at Node 2;
11) \( \lambda A_1, \lambda A_2, \ldots \) = parameter \( \lambda \) in Direction A;
12) \( k A_1, k A_2, \ldots \) = parameter \( k \) in Direction A;
13) \( \text{BL} \) = admissibility of turning to the left from Direction B at Node 1;
14) \( \text{BS} \) = admissibility of direct motion from Direction B at Node 1;
15) \( \text{BR} \) = admissibility of turning to the left from Direction B at Node 1;
16) \( \lambda B_1, \lambda B_2, \ldots \) = parameter \( \lambda \) in Direction B;
17) \( k B_1, k B_2, \ldots \) = parameter \( k \) in Direction B.

\[ B_{INTERSECTION} = (S_1, S_2, \lambda Cline_1, k Cline_1, \ldots, \lambda Dline_1, k Dline_1, \ldots) \]

where

1) The line number is the number of the edge connecting the type 1 and 2 nodes in the matrix \( A_{STREETS} \);
2) \( S_1 \) and \( S_2 \) = intersected lines forming Node 1;
3) \( \lambda Cline_1, \lambda Cline_2, \ldots \) = parameter \( \lambda \) of flows arriving at Node 1 in Direction C of the line crossing the given edge in Node 1;
4) \( k Cline_1, k Cline_2, \ldots \) = parameter \( k \) of flows arriving at Node 1 in Direction C of the line crossing the given edge in Node 1;
5) \( \lambda Dline_1, \lambda Dline_2, \ldots \) = parameter \( \lambda \) of flows arriving at Node 1 in Direction D of the line crossing the given edge in Node 1;
6) \( k Dline_1, k Dline_2, \ldots \) = parameter \( k \) of flows arriving at Node 1 in Direction C of the line crossing the given edge in Node 1.
3. Optimisation of Flows Distribution in the Network

3.1. Determination of the Optimal Flow Distribution in a Node

Let us consider the following problem: we must find out the optimal distribution of flows for a given node \( z_n = \text{Str} 1 \cap \text{Str} 2 \) from the known methods of the set \( \Psi = \{ \lambda A, kA, \lambda B, kB, \lambda C, kC, \lambda D, kD, \text{Contr, Prior} \} \).

Depending on the aim pursued the criteria of optimization can be:
1) \( \lambda i = \) weight of the vertex \( i \) (node-point) for the flow of the given direction;
2) \( \mu i = \) weight of the vertex \( i \);
3) \( \omega_M (z_n) = \) mean delay of request in the chosen directions.

For the type 1 node:
1) \( \bar{\mu}(z_n) = \sum_{i \in M} \frac{\lambda i}{W_{Hi}} \), where \( M \) = set of chosen directions;
2) \( \mu(z_n) = \sum_{i \in \Omega} \frac{\lambda i}{W_{Hi}} \), where \( \Omega \) = set of all directions;
3) \( \omega_M(z_n) = \frac{\sum_{i \in M} \left( \frac{\lambda i}{ki} \cdot W_{Hi} \right)}{\sum_{i \in M} \frac{\lambda i}{ki}} \), where \( M \) = set of chosen directions.

Here we used the following notation (as in[10]): \( W_{Hi} = \) mean delay (seconds) at a node of one request of a secondary direction with \( \lambda \) and \( k \) parameters of distribution:

1) \( \bar{\mu}(z_n) = \min \{ \bar{\mu}(z_n) \} \);
2) \( \mu(z_n) = \min \{ \mu(z_n) \} \);
3) \( \omega_M(z_n) = \min \{ \omega_M(z_n) \} \).

3.2. Selection of the Optimal Parameters of the Type 2 Node Operation

Let us set the following task of optimization of the type 2 node operation: to minimize the total hour delay of all requests in the given node on condition of absence of congestion for each flow:

\[ H(T_i, \lambda_i) - \frac{T_i}{h} \leq 0, \quad i = 1, 2, \ldots, n_1; \]
\[ H(T_j, \lambda_j) - \frac{T_j}{h} \leq 0, \quad j = 1, 2, \ldots, n_2, \]

where \( n_1 \) = number of flows of Line №1; \( n_2 \) = number of flows of Line №2. It is necessary also to fulfill the condition: \( T \geq 2M \), where \( M \) = the minimum time (seconds) necessary for a request to pass the type 2 node.

The problem of mathematical (non-linear) programming is:
\[ \sum_{i \in M} W(T_i, \lambda_i) + \sum_{j \in M} W(T_j, \lambda_j) \]
\[ \frac{T}{T} \rightarrow \min \]
The results obtained are the optimal values of operation parameters $T_1, T_2$.

Specify the algorithm of numerical solution to the problem (4-5) as a relaxation process — the process of construction of successive approximations $M_1, M_2, ..., M_k, ...$ for which $M_i \in \Omega$ and $Z(M_{k+1}) < Z(M_k)$.

**Step 1.** Specify initial values:

$$p^* = \frac{\sum \lambda_j / k_j}{\sum \lambda_i / k_i + \sum \lambda_j / k_j}$$

and

$$T^* = \min_{i,j} \left\{ \left( \frac{(1-k_i)}{2k_i} \right) - \left( \frac{(1-k_j)}{2k_j} \right), \left( \frac{\lambda_i - 1-p}{k_i - \frac{p}{h}} \right) - \left( \frac{\lambda_j - p}{k_j - \frac{p}{h}} \right) \right\} ;$$

and the values for fulfillment of the algorithm $\varepsilon_p = 0.01$; $\varepsilon_T = 0.5$; $\varepsilon_Z = 0.1$.

**Step 2.** Find numerically (for example, by half-division method) the solution $T^{*1}$ to the equation

$$H(T, \lambda_i) - \frac{(1-p)T}{h} = 0,$$

meeting the condition:

$$\min_{i} \left\{ \frac{\lambda_i}{k_i} \right\} .$$

**Step 3.** Check the fulfillment of other inequalities of the system with restrictions:

$$H(T^{*1}, \lambda_i) - \frac{(1-p)T^{*1}}{h} \leq 0, \quad i = 1,2,...,n_1 ;$$

$$H(T^{*1}, \lambda_j) - \frac{pT^{*1}}{h} \leq 0, \quad j = 1,2,...,n_2 .$$

**Step 4.** If the conditions of Step 3 are fulfilled, compute $Z^*(p^*, T^{*1})$ and go to Step 5. If the conditions of Step 3 are not fulfilled, then find numerically (for example, by half-division method) the solution $T^{*1}$ to the equation

$$H(T, \lambda_i) - \frac{(1-p)T}{h} = 0 ,$$

meeting the condition:

$$\min_{i} \left\{ \frac{\lambda_i}{k_i} \right\} ;$$

and check the fulfillment of other inequalities of the system with restrictions:

$$H(T^{*1}, \lambda_i) - \frac{(1-p)T^{*1}}{h} \leq 0, \quad i = 1,2,...,n_1 ;$$

$$H(T^{*1}, \lambda_j) - \frac{pT^{*1}}{h} \leq 0, \quad j = 1,2,...,n_2 .$$

Then compute $Z^*(p^*, T^{*1})$ and go to Step 5.

**Step 5.** Assume $T = T^{*1}$ (initial value $T^{*0} = p^* \cdot T^{*1}$) and find numerically the solution $T^{*1}$ to the equation:

$$\frac{\partial Z}{\partial T_1} = 0$$

$$T_1 \left( \sum \left( \frac{\lambda_i}{k_i} \right) + \sum \left( \frac{\lambda_j}{k_j} \right) \right) - T_1 \left( \sum \frac{\lambda_i}{2k_i} + \sum \frac{1-k_i}{2k_i} + \sum R_i - \sum \frac{1-k_j}{2k_j} + \sum \frac{\partial R_{ij}}{\partial T_1} \right) = 0$$

The new value $p^{*1} = \frac{T^{*1}}{T^{*1}}$.

**Step 6.** Repeat Steps 2-4 until $\Delta Z < \varepsilon_z$, $\Delta p < \varepsilon_p$, $\Delta T < \varepsilon_T$.

The author has proved that the successive approximations meet the conditions of convergence with the optimal solution $M_0$:

$$\lim_{k \to \infty} Z(M_k) = Z(M_0) .$$
3.3. Determination of the Critical Values of the Erlang Distribution Parameters for the Type 2 Node

As stated above, there is no congestion in the type 2 node under certain conditions (2-3). In the case where time intervals are distributed according to Erlang law (1), the given conditions are identical with the following inequality:

\[
\begin{align*}
\frac{\lambda_j}{k_j} T + \frac{1 - k_j}{2k_j} + R_j(\lambda_j, k_j, t) & \leq \frac{T_j}{h}, \quad j = 1, 2, \ldots, n2 \\
\frac{\lambda_i}{k_i} T + \frac{1 - k_i}{2k_i} + R_i(\lambda_i, k_i, t) & \leq \frac{T_i}{h}, \quad i = 1, 2, \ldots, n1
\end{align*}
\]

(6)

Since \( \frac{1}{4} \leq \left| R(\lambda, k, t) \right| \leq \frac{1}{2k \cdot \sin \frac{\pi}{k}} \), the approximate solution to the system for \( \lambda \) parameter (with \( k \) parameter being known) is:

\[
\lambda_i \left\{ \begin{array}{l}
\frac{T_i}{h} + \frac{k_i - 1}{2k_i} - \frac{1}{2k_i \cdot \sin \frac{\pi}{k_i}} \cdot k_i < \\
\frac{T_j}{h} + \frac{k_j - 1}{2k_j} - \frac{1}{2k_j \cdot \sin \frac{\pi}{k_j}} \cdot k_j < \\
\end{array} \right\}, \quad i = 1, 2, \ldots, n1
\]

(7)

If necessary, the value of \( \lambda \) parameter of the Erlang distribution for each flow preventing congestion in the type 2 node can be found numerically with any precision by solving the equation:

\[
H(T, \lambda) - \frac{T_i}{h} = 0 \quad \text{with} \quad H(T, \lambda) - \frac{T_j}{h} = 0.
\]

(8)

3.4. Selection of the Optimal Path for Flows in the Given Network (from the Preset Number)

Assume there are a limited number of certain routes \( WAY_i(z_0, z_p), i = 1, 2, \ldots, k \), from the node \( z_0 \) to the node \( z_p \) given by enumeration of nodes in the order of their passing by the flow requests.

Let the optimization criterion be the following function:

\[
\mu(WAY_i(z_0, z_p)) = \sum_{j \in D_i, \text{nei} V_i} \left( \tilde{\mu}(z_n) + \mu(l) \right)
\]

\( \mu(l) \) = weight of the edge \( l \) for the flow of the given direction;

\( \mu(l) \) = weight of the vertex \( z \) (node-point) for the flow of the given direction;

\( \mu(WAY_i(z_0, z_p)) = \sum_{j \in D_i, \text{nei} V_i} \left( \tilde{\mu}(z_n) + \mu(l) \right) \)

3.5. Determination of the Optimal Path Between Two Nodes

As stated above, there is no congestion in the type 2 node under certain conditions (2-3). In the case where time intervals are distributed according to Erlang law (1), the given conditions are identical with the following inequality:

\[
\begin{align*}
\frac{\lambda_i}{k_i} T + \frac{1 - k_i}{2k_i} + R_i(\lambda_i, k_i, t) & \leq \frac{T_i}{h}, \quad i = 1, 2, \ldots, n1 \\
\frac{\lambda_j}{k_j} T + \frac{1 - k_j}{2k_j} + R_j(\lambda_j, k_j, t) & \leq \frac{T_j}{h}, \quad j = 1, 2, \ldots, n2
\end{align*}
\]

Then the optimal path is determined by the solution to the following problem:

\[
WAY_{opt} = \min_{i} \left\{ \mu(WAY_i(z_0, z_p)) \right\}
\]

Information about the network necessary for computation is given in two connected matrices \( ASTREETS \) and \( B\_{\text{INTERSECTION}} \).

Lemma 2. Let:

\[
x = \text{Str1} \cap \text{Str2}, \quad y = \text{Str3} \cap \text{Str4} \quad \text{(with the order precision of } \text{Str1} \cup \text{Str2}, \quad \text{Str3} \cup \text{Str4}).
\]

The nodes \( x \) and \( y \) are adjacent only when the matrix \( ASTREETS \) has the following row:

\[
(A_{\text{STREETS}})_i = (S_1, S_2, S_3, S_4 \ldots) \quad \text{or} \\
(A_{\text{STREETS}})_j = (S_1, S_2, S_3, S_4 \ldots).
\]

Lemma 3. Let:

\[
x = \text{Str1} \cap \text{Str2}, \quad y = \text{Str3} \cap \text{Str4}, \quad z = \text{Str5} \cap \text{Str6} \quad \text{(with the order precision of } \text{Str1} \cup \text{Str2}, \quad \text{Str3} \cup \text{Str4}, \quad \text{Str5} \cup \text{Str6}).
\]

In the graph \( G(Z, L) \) there is a path \((x, y, z)\), where \( x \) and \( y \) and \( z \) are adjacent nodes only when the following conditions are fulfilled:

Case 1)

\[
\{\text{Str1}, \text{Str2}\} \cap \{\text{Str3}, \text{Str4}\} \cap \{\text{Str5}, \text{Str6}\} \neq 0;
\]

The initial data are given in the following rows of the matrix \( AS \):

\[
(A_{\text{STREETS}})_i = (S_1, S_2, S_3, S_4 \ldots) \quad ; \quad (AS)_{11} = 1 ; \quad (AS)_{21} \neq 0.
\]
Case 2)
The initial data are given in the following rows of the matrix \(A_{STREETS}^i\):
\[
\begin{pmatrix}
S_1 & S_2 & S_3 & S_4 & \ldots \\
\end{pmatrix};  \\
(\lambda B_1)^i \neq 0;
\]
\(\{\text{Str}1, \text{Str}2\} \cap \{\text{Str}3, \text{Str}4\} \cap \{\text{Str}5, \text{Str}6\} \neq 0;\)
The initial data are given in the following rows of the matrix \(A_{STREETS}^j\):
\[
\begin{pmatrix}
S_5 & S_6 & S_3 & S_4 & \ldots  \\
\end{pmatrix};  \\
(\lambda B_1)^j \neq 0;
\]

Case 3)
The initial data are given in the following rows of the matrix \(A_{STREETS}^i\):
\[
\begin{pmatrix}
S_1 & S_2 & S_3 & S_4 & \ldots \\
\end{pmatrix};  \\
(\lambda A_1)^i \neq 0;
\]
\(\{\text{Str}1, \text{Str}2\} \cap \{\text{Str}3, \text{Str}4\} \cap \{\text{Str}5, \text{Str}6\} \neq 0;\)
The initial data are given in the following rows of the matrix \(A_{STREETS}^j\):
\[
\begin{pmatrix}
S_5 & S_6 & S_3 & S_4 & \ldots  \\
\end{pmatrix};  \\
(\lambda B_1)^j \neq 0;
\]

Case 4)
The initial data are given in the following rows of the matrix \(A_{STREETS}^i\):
\[
\begin{pmatrix}
S_1 & S_2 & S_3 & S_4 & \ldots \\
\end{pmatrix};  \\
(\lambda A_1)^i \neq 0;
\]
\(\{\text{Str}1, \text{Str}2\} \cap \{\text{Str}3, \text{Str}4\} \cap \{\text{Str}5, \text{Str}6\} \neq 0;\)
The initial data are given in the following rows of the matrix \(A_{STREETS}^j\):
\[
\begin{pmatrix}
S_5 & S_6 & S_3 & S_4 & \ldots  \\
\end{pmatrix};  \\
(\lambda B_1)^j \neq 0;
\]

Case 5)
The initial data are given in the following rows of the matrix \(A_{STREETS}^i\):
\[
\begin{pmatrix}
S_1 & S_2 & S_3 & S_4 & \ldots \\
\end{pmatrix};  \\
(\lambda A_1)^i \neq 0;
\]
\(\{\text{Str}1, \text{Str}2\} \cap \{\text{Str}3, \text{Str}4\} \cap \{\text{Str}5, \text{Str}6\} = 0;\)
The initial data are given in the following rows of the matrix \(A_{STREETS}^j\):
\[
\begin{pmatrix}
S_5 & S_6 & S_3 & S_4 & \ldots  \\
\end{pmatrix};  \\
(\lambda B_1)^j \neq 0;
\]

Case 6)
The initial data are given in the following rows of the matrix \(A_{STREETS}^i\):
\[
\begin{pmatrix}
S_1 & S_2 & S_3 & S_4 & \ldots \\
\end{pmatrix};  \\
(\lambda A_1)^i \neq 0;
\]
\(\{\text{Str}1, \text{Str}2\} \cap \{\text{Str}3, \text{Str}4\} \cap \{\text{Str}5, \text{Str}6\} = 0;\)
The initial data are given in the following rows of the matrix \(A_{STREETS}^j\):
\[
\begin{pmatrix}
S_5 & S_6 & S_3 & S_4 & \ldots  \\
\end{pmatrix};  \\
(\lambda B_1)^j \neq 0;
\]

Case 7)
The initial data are given in the following rows of the matrix \(A_{STREETS}^i\):
\[
\begin{pmatrix}
S_1 & S_2 & S_3 & S_4 & \ldots \\
\end{pmatrix};  \\
(\lambda A_1)^i \neq 0;
\]
\(\{\text{Str}1, \text{Str}2\} \cap \{\text{Str}3, \text{Str}4\} \cap \{\text{Str}5, \text{Str}6\} = 0;\)
The initial data are given in the following rows of the matrix \(A_{STREETS}^j\):
\[
\begin{pmatrix}
S_5 & S_6 & S_3 & S_4 & \ldots  \\
\end{pmatrix};  \\
(\lambda B_1)^j \neq 0;
\]

Case 8)
The initial data are given in the following rows of the matrix \(A_{STREETS}^i\):
\[
\begin{pmatrix}
S_1 & S_2 & S_3 & S_4 & \ldots \\
\end{pmatrix};  \\
(\lambda A_1)^i \neq 0;
\]
\(\{\text{Str}1, \text{Str}2\} \cap \{\text{Str}3, \text{Str}4\} \cap \{\text{Str}5, \text{Str}6\} = 0;\)
The initial data are given in the following rows of the matrix \(A_{STREETS}^j\):
\[
\begin{pmatrix}
S_5 & S_6 & S_3 & S_4 & \ldots  \\
\end{pmatrix};  \\
(\lambda B_1)^j \neq 0;
\]

Case 9)
The initial data are given in the following rows of the matrix \(A_{STREETS}^i\):
\[
\begin{pmatrix}
S_1 & S_2 & S_3 & S_4 & \ldots \\
\end{pmatrix};  \\
(\lambda A_1)^i \neq 0;
\]
\(\{\text{Str}1, \text{Str}2\} \cap \{\text{Str}3, \text{Str}4\} \cap \{\text{Str}5, \text{Str}6\} = 0;\)
The initial data are given in the following rows of the matrix \(A_{STREETS}^j\):
\[
\begin{pmatrix}
S_5 & S_6 & S_3 & S_4 & \ldots  \\
\end{pmatrix};  \\
(\lambda B_1)^j \neq 0;
\]
\[(A_{\text{STREETS}})_i = (S_1, S_2, S_3, S_4, \ldots); (AL)_i = 1; (\lambda A1)_i \neq 0; (\lambda B1)_i \neq 0; (\lambda A1)_i \neq 0\].

Step 4. Select the minimum element in the field MMinus.TimeCr and record the data on a certain node in the array MPlus numbered as (n+1). Delete the data on this node from the array MMinus.

Step 5. Repeat Steps 2-4 until node (n+1) in the array MPlus coincides with the end of path. Then finish computing.

Step 6. Display the array MPlus – the list of nodes through which the shortest path between the two given points of network goes.

The given algorithm is author's modification of Dijkstra’s algorithm[3]. It is an iterative procedure where each vertex is marked – either by a permanent mark which shows the distance from this node to a particular node or by a temporary mark when this distance is estimated from above.

The implementation of the algorithm needs \(n^2\) operations \((n\) is a number of the graph vertices).

3.6. Determination of the Optimal Scheme of Flows Distribution in the Given Network (from the Preset Number)

Specify the sub-graph \(\{z_n\}_{n \in V}\) to be reorganized. Possible variants of reorganization of flow distribution in the network are presented in the matrices \((A_{\text{STREETS}}) \cup (B_{\text{INTERSECTION}}), i \in K\).

If reorganization is aimed at minimizing the delays at nodes, then the criterion for optimization is the sum of nodes weights:

\[
\mu_i(z_n)_{n \in V} = \sum_{n \in V} \mu_i(z_n).
\]

The optimal scheme of flow distribution in the network is the solution to the problem:

\[
\mu_{\text{opt}} = \min \sum_{n \in V} \mu_i(z_n).
\]

If reorganization is aimed at optimization of traffic flow along the given path \((V,D)\) in the network, then the weight of path, i.e. the time spent on passing the given path should be taken as target function:

\[
\mu_i(WAY(z_0, z_p)) = \sum_{i \in K} \sum_{n \in V} \left(\mu_i(z_n) + \mu_i(l_i) \right)
\]

where: \(\mu_i(z_n)\) = weight of vertex \(z_n\) (node-point) for the flow in the given direction;

\(\mu_i(l_j)\) = weight of edge \(l_j\) for the flow in the given direction;

\(D\) = set of edges of the path;

\(V\) = set of vertices of the path.
Then the optimal scheme of flow distribution in the network is the solution to the problem:
\[
\mu_{\text{opt}} = \min_i \left\{ \mu_i \left( \mathcal{W} \left( z_0, z_p \right) \right) \right\}
\]

4. Conclusions

The above considered optimization problems present the probabilistic model of traffic flow developed by the author (e.g. [10, 13-15]). This model is based upon the hypothesis about the Erlang time distribution for requests arriving in succession. The adequacy of this hypothesis for traffic flows was proved by experiments. The virtue of the developed model is the minimal number of initial data necessary for computing the indices of quality of the network operation. The suggested graph representation of network allows us to solve the problems on optimization of traffic flows by optimal distribution in the nodes. If there are data files for a certain road segment in the urban transportation network organized according to the matrices \( A_{\text{STREETS}} \) and \( B_{\text{INTERSECTION}} \), it will be easy to implement the algorithms of solutions to these problems using computer environment, e.g. DELPHI.

REFERENCES

[16] Semenov, V.V. Математическое моделирование динамики транспортных потоков мегаполиса (Mathematical modelling of traffic flows dynamics in the megapolis: Review), Moscow, 2003.