Applied Mathematics 2013, 3(1): 12-19
DOI: 10.5923/j.am.20130301.02

# 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 $A_{STREETS}$ и $B_{INTERSECTION}$ 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 transportatio n 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. Haigt[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 transportations 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.

* Corresponding author:
Nataly_Naumova@mail.ru (Naumova N. A.)
Published online at http://journal.sapub.org/am


## 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:

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

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].

We will call a node-point in which traffic is alternately blocked for one of the non conflict flow group for a fixed time to enable crossing a 'type 2 node', as in[10, 13, 15].

Let:

$\{l_j\}$ = set of graph edges,

$\{z_j\}$ = set of nodes.

Then, a edge is part of a multichannel line between two nodes. Assign the identification numbers $WAY\_i$, $i \in \mathrm{N}$ to the lines. Then $STREET\_i = \bigcup_{j \in W} l_j$.

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

**2.1.** $A_{STREETS} = (S_1 \quad S_2 \quad S_3 \quad S_4 \quad Contr \quad \Pr \quad Len \quad Col\ AL\ AS\ AR \quad \lambda A1 \quad kA1 \ldots \quad BL\ BS\ BR \quad \lambda B1 \quad kB1 \ldots)$

where

1) № = 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) $S_1$ и $S_2$ = intersected lines forming Node 1;

3) $S_3$ и $S_4$ – intersected lines forming Node 2;

4) C = node type;

5) Pr = priority (main or secondary line);

6) L = length of edge;

7) Col = number of flows on the edge;

8) *AL* = admissibility of turning to the left from Direction A at Node 1;

*9) AS* = admissibility of direct motion from Direction A at Node 2;

*10) AR* = admissibility of turning to the right from Direction A at Node 2;

1*1)λA1*, *λA2*, etc. = parameter λ in Direction A;

*12) kA1*, *kA2,* etc. = parameter *k* in Direction A;

*13) BL* = admissibility of turning to the left from Direction B at Node 1;

*14) BS* = admissibility of direct motion from Direction B at Node 1;

*15) BR* = admissibility of turning to the left from Direction B at Node 1;

*16)*λ1, λB2 etc. = parameter λ in Direction B;

*17) kB1*, *kB2* etc. = parameter *k* in Direction B.

**2.2.** $B_{INTERSECTION} = (S_1 \quad S_2 \quad \lambda Cline1 \quad kCline1 \quad \ldots \quad \lambda Dline1 \quad kDline1 \quad \ldots)$

where

1) The line number is the number of the edge connecting the type 1 and 2 nodes in the matrix $A_{STREETS}$;

2) S1 и S2 = intersected lines forming Node 1;

3) λCline1, λCline2 etc. = parameter λ of flows arriving at Node 1 in Direction C of the line crossing the given edge in Node 1;

4) kC line1, kC line2 etc. = parameter k of flows arriving at Node 1 in Direction C of the line crossing the given edge in Node 1;

5) λD line1, λD line2 etc. = parameter λ of flows arriving at Node 1 in Direction D of the line crossing the given edge in Node 1;

6) kD line1, kD line2 etc. = 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 $z_n = Str1 \cap Str2$ from the known methods of the set $\Psi = \{\lambda A, kA, \lambda B, kB, \lambda C, kC, \lambda D, kD, Contr, \Pr ior\}$.

Depending on the aim pursued the criteria of optimization can be:

1) $\vec{\mu}(z_n)$ = weight of the vertex $z_n$ (node-point) for the flow of the given direction;

2) $\mu(z_n)$ = weight of the vertex $z_n$;

3) $\omega_M(z_n)$ = mean delay of request in the chosen directions.

For the type 1 node:

1) $\vec{\mu}(z_n) = \sum_{i \in M} \frac{\frac{\lambda i}{ki} W_{Hi}}{3600}$, where $M$ = set of chosen directions;

2) $\mu(z_n) = \sum_{i \in \Omega} \frac{\frac{\lambda i}{ki} W_{Hi}}{3600}$, 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]): $W_H$ = mean delay (seconds) at a node of one request of a secondary direction with $\lambda$ and $k$ parameters of distribution:

For the type 2 node:

1) $\vec{\mu}(z_n) = \frac{\sum_{i \in M} W(T_a, \lambda_i)}{T}$, where $M$ = set of chosen directions, $a \in \{1;2\}$;

2) $\mu(z_n) = \frac{\sum_i W(T_1, \lambda_i) + \sum_j W(T_2, \lambda_j)}{T}$;

3) $\omega_M(z_n) = \frac{\sum_{i \in M} W(T_a, \lambda_i)}{\sum_{i \in M} H(T_a, \lambda_i)}$, where $M$ = set of chosen directions, $a \in \{1;2\}$.

Here we used the following notations (as in[10]): $W(T_i, \lambda) = \int_0^{T_1} H_\lambda(t)dt$ (requests per second) = the total delay of all requests of a given flow for one regulated cycle $T = T_1 + T_2$ ;

$H(t)$ = number of requests arriving at a given point of road for the time interval (*0; t*).

Let: the given vertex (node-point) $z_n = Str1 \cap Str2$ (with the order precision of $Str1$ u $Str2$ ). The information about incoming flows is given in he matrices $A_{STREETS}$ and $B_{INTERSECTION}$.

**Lemma 1.** The Erlang distribution parameters for incoming flows of the vertex $z_n = Str1 \cap Str2$ are given:

1) in the matrix $A_{STREETS}$ in the line $(A_{STREETS})_i = (Str1 \quad Str2 \quad Str1 \quad X \quad ...)$ – in Direction B;

2) in the matrix $B_{INTERSECTION}$ in the line $(B_{INTERSECTION})_i$ in Directions C and D;

3) in the matrix $A_{STREETS}$ in the line $(A_{STREETS})_j = (Str1 \quad Y \quad Str1 \quad Str2 \quad ...)$ – in Direction A.

The optimal flow distribution in the node is the solution to the problem (depending on the pursued objective):

1) $\vec{\mu}(z_n)\_opt = \min_{\Psi} \{\vec{\mu}(z_n)\}$;

2) $\mu(z_n)\_opt = \min_{\Psi} \{\mu(z_n)\}$;

3) $\omega_M(z_n)\_opt = \min_{\Psi} \{\omega_M(z_n)\}$.

### 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 $\mu(z_n)$ in the given node on condition of absence of congestion for each flow:

$$H(T, \lambda_i) - \frac{T_2}{h} \leq 0, \quad i = 1,2,...,n1; \tag{2}$$

$$H(T, \lambda_j) - \frac{T_1}{h} \leq 0, \quad j = 1,2,...,n2, \tag{3}$$

where $n1$ = number of flows of Line №1; $n2$ = 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:

$$Z = \frac{\sum_i W(T_1, \lambda_i) + \sum_j W(T_2, \lambda_j)}{T} \to \min \tag{4}$$

$$\Omega: \begin{cases} H(T,\lambda_i) - \dfrac{T_2}{h} \le 0, \\ H(T,\lambda_i) - \dfrac{T_1}{h} \le 0, \\ T \ge 2M, \\ T_1 + T_2 = 2M \end{cases} \quad (5)$$

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 $M_i \in \Omega$ и $Z(M_{k+1}) < Z(M_k)$.

**Step 1.** Specify initial values:

$$p^* = \frac{\sum_j \dfrac{\lambda_j}{k_j}}{\sum_i \dfrac{\lambda_i}{k_i} + \sum_j \dfrac{\lambda_j}{k_j}}$$

and

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

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 $H(T,\lambda_i) - \dfrac{(1-p)T}{h} = 0$, meeting the condition: $\min_i \left\{ \dfrac{\lambda_i}{k_i} \right\}$.

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

$$H(T^{*1},\lambda_i) - \frac{(1-p)T^{*1}}{h} \le 0, \quad i = 1,2,...,n1;$$

$$H(T^{81},\lambda_j) - \frac{pT^{*1}}{h} \le 0, \quad j = 1,2,...,n2.$$

**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 $H(T,\lambda_i) - \dfrac{(1-p)T}{h} = 0$, meeting the condition: $\min_i \left\{ \dfrac{\lambda_i}{k_i} \right\}$; check the fulfillment of other inequalities of the system with restrictions:

$$H(T^{*1},\lambda_i) - \frac{(1-p)T^{*1}}{h} \le 0, \quad i = 1,2,...,n1;$$

$$H(T^{81},\lambda_j) - \frac{pT^{*1}}{h} \le 0, \quad j = 1,2,...,n2.$$

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

**Step 5.** Assume $T = T^{*1}$ (initial value $T_1^{*0} = p^* \cdot T^{*1}$) and find numerically the solution $T_1^*$ to the equation:

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

$$T_1 \cdot \left( \sum_i \left( \frac{\lambda_i}{k_i} \right) + \sum_j \left( \frac{\lambda_j}{k_j} \right) \right) - T \cdot \sum_j \left( \frac{\lambda_j}{k_j} \right) + \sum_i \frac{1-k_i}{2k_i} + \sum_i R_{1i} - \sum_j \frac{1-k_j}{2k_j} + \sum_j \frac{\partial \widetilde{R}_{2j}}{\partial T_1} = 0$$

The new value $p^{*1} = \dfrac{T_1^*}{T^{*!}}$.

**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 system:

$$\begin{cases} \frac{\lambda_i}{k_i}T + \frac{1-k_i}{2k_i} + R_i(\lambda_i,k_i,t) \le \frac{T_2}{h},\ i=1,2,...,n1 \\ \frac{\lambda_j}{k_j}T + \frac{1-k_j}{2k_j} + R_j(\lambda_j,k_j,t) \le \frac{T_1}{h},\ j=1,2,...,n2 \end{cases} \quad (6)$$

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

$$\begin{cases} \lambda_i < \frac{\left(\frac{T_2}{h}+\frac{k_i-1}{2k_i}-\frac{1}{2k_i\cdot\sin\frac{\pi}{k_i}}\right)\cdot k_i}{T},\ i=1,2,...,n1 \\ \lambda_j < \frac{\left(\frac{T_1}{h}+\frac{k_j-1}{2k_j}-\frac{1}{2k_j\cdot\sin\frac{\pi}{k_j}}\right)\cdot k_j}{T},\ j=1,2,...,n2 \end{cases} \quad (7)$$

If necessary, the value of λ 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:

$$H(T,\lambda_i)-\frac{T_2}{h}=0 \text{ или } H(T,\lambda_j)-\frac{T_1}{h}=0\cdot \quad (8)$$

### 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 routs $WAY_i(z_0,z_p),\ i=1,2,...,k$, 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:

$$\mu(WAY_i(z_0,z_p)) = \sum_{\substack{j\in D_i \\ n\in V_i}}\left(\vec{\mu}(z_n)+\mu(l_j)\right)$$

=weight of path, i. e. time spent on passing the given path, where $\vec{\mu}(z_n)$ = weight of the vertex $z_n$ (node-point)for the flow of the given direction;;

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

$D_i$ = set of edges of the path $WAY_i(z_0,z_p),\ i=1,2,...,k$ ;

$V_i$ = set of nodes of the path $WAY_i(z_0,z_p),\ i=1,2,...,k$ .

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

$$WAY\_opt = \min_i\{\mu(WAY(z_0,z_p))\}$$

Information about the network necessary for computation is given in two connected matrices $A_{STREETS}$ и $B_{INTERSECTION}$.

### 3.5. Determination of the Optimal Path Between Two Nodes

Let us set the following task : to find the path from the vertex $z_0$ to the vertex $z_p$, meeting condition: $\mu(WAY_i(z_0,z_p)) = \sum_{\substack{j\in D_i \\ n\in V_i}}\left(\vec{\mu}(z_n)+\mu(l_j)\right) \to \min$. Unlike the problem of the previous point, only the initial vertex $z_0$ and final vertex $z_p$ of the path are specified.

Take into account that the chosen graph representation allows for each node to be adjacent to four other nodes at most. A node is shown as an intersection of two lines $Str1$ and $Str2$.

**Lemma 2.** Let:

$x = Str1 \cap Str2$, $y = Str3 \cap Str4$ (with the order precision of $Str1$ и $Str2$, $Str3$ и $Str4$).

The nodes $x$ and $y$ are adjacent only when the matrix $A_{STREETS}$ has the following row:

$(A_{STREETS})_i = (S_1 \quad S_2 \quad S_3 \quad S_4 \quad ...)$ or $(A_{STREETS})_j = (S_3 \quad S_4 \quad S_1 \quad S_2 \quad ...)$.

**Lemma 3.** Let:

$x = Str1 \cap Str2$, $y = Str3 \cap Str4$, $z = Str5 \cap Str6$ (with the order precision of $Str1$ и $Str2$, $Str3$ и $Str4$, $Str5$ и $Str6$).

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

Case 1)

$$\{Str1,Str2\}\cap\{Str3,Str4\}\cap\{Str5,Str6\} \ne 0;$$

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$(A_{STREETS})_i = (S_1 \quad S_2 \quad S_3 \quad S_4 \quad ...)$ ; $(AS)_i = 1$ ; $(\lambda A1)_i \ne 0$ ;

$\left(A_{STREETS}\right)_j = \left(S_3 \quad S_4 \quad S_5 \quad S_6 \quad ...\right)$ ; $\left(\lambda A1\right)_j \neq 0$

Case 2)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_3 \quad S_4 \quad S_1 \quad S_2 \quad ...\right)$; $\left(BS\right)_i = 1$; $\left(\lambda B1\right)_i \neq 0$

$\left(A_{STREETS}\right)_j = \left(S_5 \quad S_6 \quad S_3 \quad S_4 \quad ...\right)$ ; $\left(\lambda B1\right)_j \neq 0$;

Case 3)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_1 \quad S_2 \quad S_3 \quad S_4 \quad ...\right)$; $\left(AS\right)_i = 1$; $\left(\lambda A1\right)_i \neq 0$;

$\left(A_{STREETS}\right)_j = \left(S_5 \quad S_6 \quad S_3 \quad S_4 \quad ...\right)$ ; $\left(\lambda B1\right)_j \neq 0$;

Case 4)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_3 \quad S_4 \quad S_1 \quad S_2 \quad ...\right)$; $\left(BS\right)_i = 1$; $\left(\lambda B1\right)_i \neq 0$;

$\left(A_{STREETS}\right)_j = \left(S_3 \quad S_4 \quad S_5 \quad S_6 \quad ...\right)$ ; $\left(\lambda A1\right)_j \neq 0$;

Case 5)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} = 0$ ; $\{Str1, Str2\} \cap \{Str3, Str4\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_1 \quad S_2 \quad S_3 \quad S_4 \quad ...\right)$; $\left(AR\right)_i = 1$; $\left(\lambda A1\right)_i \neq 0$;

$\left(A_{STREETS}\right)_j = \left(S_3 \quad S_4 \quad S_5 \quad S_6 \quad ...\right)$ ; $\left(\lambda A1\right)_j \neq 0$;

Case 6)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} = 0$ ; $\{Str1, Str2\} \cap \{Str3, Str4\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_3 \quad S_4 \quad S_1 \quad S_2 \quad ...\right)$; $\left(BR\right)_i = 1$; $\left(\lambda B1\right)_i \neq 0$

$\left(A_{STREETS}\right)_j = \left(S_5 \quad S_6 \quad S_3 \quad S_4 \quad ...\right)$ ; $\left(\lambda B1\right)_j \neq 0$;

Case 7)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} = 0$ ; $\{Str1, Str2\} \cap \{Str3, Str4\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_1 \quad S_2 \quad S_3 \quad S_4 \quad ...\right)$; $\left(AR\right)_i = 1$; $\left(\lambda A1\right)_i \neq 0$;

$\left(A_{STREETS}\right)_j = \left(S_5 \quad S_6 \quad S_3 \quad S_4 \quad ...\right)$ ; $\left(\lambda B1\right)_j \neq 0$;

Case 8)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} = 0$ ; $\{Str1, Str2\} \cap \{Str3, Str4\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_3 \quad S_4 \quad S_1 \quad S_2 \quad ...\right)$; $\left(BR\right)_i = 1$; $\left(\lambda B1\right)_i \neq 0$;

$\left(A_{STREETS}\right)_j = \left(S_3 \quad S_4 \quad S_5 \quad S_6 \quad ...\right)$ ; $\left(\lambda A1\right)_j \neq 0$;

Case 9)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} = 0$ ; $\{Str1, Str2\} \cap \{Str3, Str4\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_1 \quad S_2 \quad S_3 \quad S_4 \quad ...\right)$; $\left(AL\right)_i = 1$; $\left(\lambda A1\right)_i \neq 0$;

$\left(A_{STREETS}\right)_j = \left(S_3 \quad S_4 \quad S_5 \quad S_6 \quad ...\right)$ ; $\left(\lambda A1\right)_j \neq 0$;

Case 10)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} = 0$ ; $\{Str1, Str2\} \cap \{Str3, Str4\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_3 \quad S_4 \quad S_1 \quad S_2 \quad ...\right)$; $\left(BL\right)_i = 1$; $\left(\lambda B1\right)_i \neq 0$

$\left(A_{STREETS}\right)_j = \left(S_5 \quad S_6 \quad S_3 \quad S_4 \quad ...\right)$ ; $\left(\lambda B1\right)_j \neq 0$;

Case 11)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} = 0$ ; $\{Str1, Str2\} \cap \{Str3, Str4\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_1 \quad S_2 \quad S_3 \quad S_4 \quad ...\right)$; $\left(AL\right)_i = 1$; $\left(\lambda A1\right)_i \neq 0$;

$\left(A_{STREETS}\right)_j = \left(S_5 \quad S_6 \quad S_3 \quad S_4 \quad ...\right)$ ; $\left(\lambda B1\right)_j \neq 0$;

Case 12)

$\{Str1, Str2\} \cap \{Str3, Str4\} \cap \{Str5, Str6\} = 0$ ; $\{Str1, Str2\} \cap \{Str3, Str4\} \neq 0$;

The initial data are given in the following rows of the matrix $A_{STREETS}$:

$\left(A_{STREETS}\right)_i = \left(S_3 \quad S_4 \quad S_1 \quad S_2 \quad ...\right)$; $\left(BL\right)_i = 1$; $\left(\lambda B1\right)_i \neq 0$;

$\left(A_{STREETS}\right)_j = \left(S_3 \quad S_4 \quad S_5 \quad S_6 \quad ...\right)$ ; $\left(\lambda A1\right)_j \neq 0$.

Each edge of the graph is assigned to the number $l(x, y) = \vec{\mu}(x) + \mu(l_{xy})$ – edge length; if the nodes are not linked by an edge then $l(x,y) = \infty$.

In the course of fulfillment of the algorithm, the graph vertices and edges are colored and the values ***d(x)*** are computed which equal to the **shortest** path from vertex ***s***= $z_0$ to vertex ***x*** which only includes the **colored** vertices:

$$d(x) = \min\{d(x), d(y) + l(x, y)\}.$$

In our case: *l(x, y)* = mean travel time of requests of the flow from vertex ***x*** to vertex ***y*** taking into account the delay at vertex ***x***.

The data necessary for solution to the problem will be stored in two arrays:

MPlus – data on the nodes having permanent marks;

MMinus – data on the nodes having temporal marks.

Each element of array has the following structure:

$$MPlus_i = \begin{pmatrix} Str1 \\ Str2 \\ TimeCr \\ Trassa \end{pmatrix} \text{ и } MMinus_i = \begin{pmatrix} Str1 \\ Str2 \\ TimeCr \\ Trassa \end{pmatrix}$$

Str1 = line along which traffic has been flowing to the given node;

Str2 = line, intersecting Str1;

TimeCr = travel time ***d(x)*** to the given node from the initial point of the path;

Trassa – list of the nodes passed.

***Step 1.*** Specify the start and finish of the path. The initial node is denoted as *n = 0* in the array MPlus and *4n* in the array MMinus.

***Step 2.*** Define all node-points adjacent to node $n^{th}$ (Lemma 1) and record their data in the array MMinus numbered as (4n + 1), (4n + 2), (4n + 3).

If vertices are not adjacent or traffic in this direction is banned then $l(x,y) = \infty$.

***Step 3.*** Compute the travel time from node $n^{th}$ to all adjacent nodes which are not recorded in the array MMinus.

***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 thru 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 $\left(A_{STREETS}\right)_i$ и $\left(B_{INTERSECTION}\right)_i$, $i \in K$.

If reorganization is aimed at minimizing the delays at nodes, then the criterion for optimization is he 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:

$$\mu\_opt = \min_i \left\{ \sum_{\substack{n \in V \\ i \in K}} \mu_i(z_n) \right\}.$$

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:

$$\mu_i(WAY(z_0, z_p)) = \sum_{\substack{i \in K \\ n \in V \\ j \in D_i}} \left(\vec{\mu}_i(z_n) + \mu_i(l_j)\right)$$

where: $\vec{\mu}_i(z_n)$ = 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:

$$\mu\_opt = \min_i \{\mu_i (WAY(z_0, z_p))\}$$

# 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 $A_{STREETS}$ и $B_{INTERSECTION}$, it will be easy to implement the algorithms of solutions to these problems using computer environment, e.g. DELPHI.

# REFERENCES


[1] Cox, D.R., Smith, W.L., Queues, Methuen, London, 1961.

[2] Gasis, D.C. et al. “Car following theory of steady state flow”, Operations Research, vol. 7. pp. 499-505, 1959.

[3] Dijkstra, E. W. "A note on two problems in connexion with graphs". Numerische Mathematik **1**, pp. 269–271, 1959.

[4] Drew, D.R. Traffic Flow Theory and Control, McGraw-Hill Book Company, New York, 1968.

[5] Haight, F.A. “Mathematical Theories of Traffic Flow”, Mathematics in Science and Engineering (vol.7, 1963).

[6] Inose, H., Hamada, T., Road Traffic Control, University of Tokyo Press, Tokyo, Japan, 1975.

[7] Nagel, K., Wagner, R., Woesler R. “Still flowing”, Approaches to traffic flow and traffic jam modeling, January 2, 2003.

[8] Ligthill, M.J., Whitham, F.R.S. “On kinetic waves II. A theory of traffic flow on crowded roads”, in Proceedings of the 1995 Royal Society Ser., vol. 229, no. 1178, pp. 317-345, 1995.

[9] Proceedings of 2003 3rd Swiss Transport Research Conference, 2003.

[10] Naumova, N., Danovich, L. "Modelling and Optimisation of Flows Distribution in the Network", Applied Mathematics, Vol. 2 No. 5, 2012, pp. 171-175. doi: 10.5923/j.am. 20120205. 04.

[11] Ventsel, E.S, Ovcharov, L.A. Теория случайных процессо в и ее инженерные приложения (Theory of random processes and their application in engineering),Vysshaya Shkola Publishing House, Moscow, 2000.

[12] Davydov E. G. Исследование операций (Operations research), Vysshaya Shkola Publishing House, Moscow, 1990.

[13] Dombrovsky, A.N., Naumova, N.A. Транспортные потоки на улично-дорожной сети городов: моделирование и управление: монография (Traffic Flows in Urban Transportation Networks: Modeling and Control), Yug Publishing House, Krasnodar, 2012.

[14] Naumova, N.A., Danovich, L.M. Моделирование и программная реализация движения автотранспортных средств по улично-дорожной сети: монография (Modelling and simulation of traffic in transportation networks), Yug Publishing House, Krasnodar, 2011.

[15] Naumova, N.A., Danovich, L.M. Моделирование и оптимизация распределения потоков по сети: монография (Modelling and optimisation of flows distribution in the network),Kuban State Technological University Press, 2011.

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

[17] Shvetsov, V.I.“Математическое моделирование транспор тных потоков” (“Mathematical modeling of traffic flows”), Avtomatika i telemekhanika, vol.11, 2003.