DEVELOPMENT OF MATHEMATICAL MODELS AND SOFTWARE OF FLOW DISTRIBUTION: THE PROBLEM OF EVACUATION

[1]AMIRGALIYEV YEDILKHAN, [2]KALIZHANOVA ALIYA, [3]KOZBAKOVA AINUR, [4]KENSHIMOV CHINGIZ, [5]SHAMILUULU SHAHRIAR

[1,3,5]Institute of Information and Computational Technologies CS MES RK, Almaty, Kazakhstan

[2,4]Al-Farabi Kazakh National University, Almaty, Kazakhstan

E-mail: [1]amir_ed@mail.ru, [2]kalizhanova_aliya@mail.ru, [3]ainur79@mail.ru, [4]kenshimov.chingiz@gmail.com

ABSTRACT

The paper herein is dedicated to mathematical models and software development for evacuation problems in emergency situations at educational organization, there is offered the problem solution method and algorithm, allowing to structure the evacuation optimal plan, changing in real time, according to a time-table and people amount. The optimization task is proposed and algorithms to find maximum flow on limited base are proposed.

Keywords: *Algorithm, Evacuation, Stream Of People, Optimal Plan, Software.*

1. INTRODUCTION

Elaboration of contemporary safety means in the mass flock spots demands developing the new research methods, and, in particular, the methods of evacuation process modeling and optimization upon the emergency situations.

Most models include such features as: people's flow motion visualization, human behavior modeling, the best evacuation routes, etc.

Elaboration of mathematical and informational flow distribution models in the networks for evacuation tasks in emergency situations is acute.

The experience evidences that the sufficient potential of evacuation system functioning efficiency is behind the use of mathematical methods and information technologies. Nowadays the emergency situation danger is increasingly grows, but calculating and forecasting methods, protection means are being created and updated with sufficient delay.

Contemporary life practice shows, that the population increasingly runs into danger due to natural calamities, accidents and disasters in industry and transport, such as: earthquakes, floods, snow slides, mud streams, landslides, massive forest fires. In such cases the evacuation is unavoidable [1].

Evacuation measures are possibly happened under accidents at atomic power stations, chemically and biologically hazardous substances spill, at big fires at oil and gas and processing plants [2].

Calamities forecasting shortages, late delivery of vital means demand perfecting of evacuation process management methods under emergency situations.

Evacuation models are designated mainly for people evacuation time definition. Very frequently such models allow defining potential areas of people flock at evacuation [3].

Most models include such peculiarities as people stream motion visualization, human behavior modeling, defining the most efficient evacuation routes, etc. [4].

Mathematical methods and information technologies use sufficiently increase evacuation system functioning performance, therefore developing the new complex and info-communication approaches to evacuation problems solution is currently important [5].

Different matters of health and safety theory and practice are found in the works of Rakishev B.R., Meljnikov N.V., Suleyev D.K., Mutanov G.M. and others.

Development of such important scientific guidelines as systemic analysis and control theory, complex technical systems simulation modeling was contributed by the works of Buslenko N.P., Rogov Ye. I., Moiseyev N.N., Moldabekov M.M., Zharaspayev M.T., Galiyev S.Zh. and others.

Timely and unobstructed evacuation upon emergency situations demands scientifically-substantiated evacuation plans.

Evacuation plans assessment is fulfilled using mathematical modeling of people’s flow motion inside the building, theoretical bases of which were founded by professor Belyayev S.V. Further researches are connected with the names of Milinsky A.I., who have worked out graphic-analytical calculation method of evacuation total time and Predtechensky V.M., who obtained empirical dependences of people’s motion speed on people’s flow density.

2. MATHEMATICAL STATEMENT OF THE PROBLEM

Multiple of all subjects thereof form multiple players I . Every γ∈I player’s strategy ξγ is a path from the source i to the outflow j out of all strategies’ multiple - paths χγ., connecting those nodes. Every subject’s criterion is motion time from i to j, its target is to minimize motion time of every transport unit, i.e. of all streams transport units. The criterion value is influenced by other players which path intercrosses with the γ player path. They increase the stream intensity at intercrossing sections thereby reducing motion speed and accordingly increasing time. As a result we obtain the game in the normal form:

$$G = \left\langle I;\ \chi_\gamma, \gamma \in I;\ \ \varphi_\gamma(\xi_1, \xi_2, \xi_3, \ldots \xi_\gamma, \ldots, \xi_{|I|}) \underset{\xi_\gamma \in \chi_\gamma}{\Longrightarrow} \min, (\xi_1, \xi_2, \xi_3, \ldots \xi_\gamma, \ldots, \xi_{|I|}) \in \chi = \prod_{k \in I} \chi_k, \gamma \in I \right\rangle$$

Assume **Nesh** balance as the balance in the game herein [4].

Thereafter let us assume one more condition that I players multiple is infinite, of continual nature. In compliance with that the streams Qij will be broken down into the streams x_{ij}^{γ} , no matter how small is the magnitude.

Arc flow circulation study

It is obvious that in order the motion time all over the way has been minimal it is necessary that every participant’s motion speed has been maximum. But the speed magnitude of the motion participant herein is spontaneously affected with other participants negative interference. They as well strive to speed maximizing upon selecting motion variables. Stream growth brings to reducing of the considered participant’s motion speed leading to increase of time thereof.

Let us consider the motion only along one arc, therefore miss out all indices in respect of the arc. Let’s introduce following designations: L – network section length, T – motion time along the section, x – stream – quantity of cars having passed through the road section per time unit, ρ - stream density – quantity of cars per length unit at one lane, s – amount of lanes on the road, w – stream speed, λ - average section length per one car on one lane.

According to the definition density is ρ=1/λ. Assume w – motion participant speed, wmax – speed. Time spent by the participant upon driving along the path of length λ equals to τ=λ/v. Participants amount per time unit is κ=1/τ. Therefore $x = \kappa s = \frac{1}{\tau} s = \frac{v}{\lambda} = w\rho s$. Assume the stream speed and density are inter connected through linear dependence $w/w_{max} + \rho/\rho_{maa} = 1$. It is the known ***Grinshield’s*** formula.

This implies that $w = w_{maa}(1 - \rho/\rho_{maa})$, or $\rho = \rho_{max}(1 - w/w_{maa})$. Let’s substitute it into the stream formula and obtain $x = s\ w\ \rho_{max}(1 - w/w_{max})$. Resulting function is the parabolic curve branched downward, maximum is achieved at $w = w_{max}/2$ and accordingly at $x_{max} = s(w_{max}\ \rho_{max})/4$. Thus we obtained maximum magnitude of the stream which can be passed along the arc.

Let’s substitute instead of ρ its value expression and we obtain $w^2 - w_{max} w + (w_{max}/(s\rho_{max}))x = 0$. From Vieta formula we get $w = w_{max}\left(1 + \sqrt{1 - x/x_{max}}\right)/2$ with account of everyone’s strive to minimize its speed. It follows from here that motion speed along network section is expressed through the dependence herein: $\tau(x) = 2\tau_{min}/\left(1 + \sqrt{1 - x/x_{max}}\right)$, where τmin – minimum motion speed along the section in case the stream thereupon equals to zero. Function graph is given for illustrative purposes of that function: $\tau(x) = 1/\left(1 + \sqrt{1 - x}\right)$.

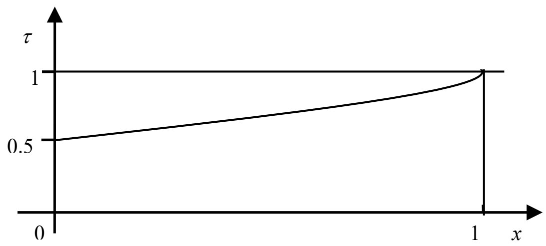

Figure 1: Timetable for calculating time

2.1 Layer concept, layer balance correlations and layer streams invariant transformation

Basic designations of graph theory used in the work herein

Assume $G = \langle E, V, H \rangle$ as an oriented graph, *E* and *V* end multiples, *H* – function *H: V→E ×E*. Let us name multiple components *E* the graph nodes, multiple components of *V* as arcs. For each arc *v∈V* the showing *H (v)=(h1(v), h2(v)), h1(v)* – the arc onset *v, h2(v)* – the end. Let's denote $V^{+}(i) = \{v \in V \mid h2(v) = i\}$ as arcs multiple entering the node *i*, $V^{-}(i) = \{v \in V \mid h1(v) = i\}$ - arcs multiple leaving the node i.

2.2 System decomposition – breakage of transport streams into layers

Assume i0 is the top the streams from other vertexes enter into. For instance, the enterprise where move in together its employees, a big shop, cinema, etc., is in i0. It is an exit from the building the evacuated subjects are directing to. If there are several exits let's introduce additional vertex. Streams out coming from the vertex i∈E and moving to the vertex i0∈E denote as qi(i0), for the vertex $q_{i_0}(i_0)$=0. Total volume of the stream inletting into i0 will be $Q_{i_0} = \sum_{i \in E} q_i(i_0)$. Let's denote $x_v(i_0)$ - stream along v moving to the vertex i0. Total $S(i_0) = \langle G; i_0; q_i(i_0), i \in E; x_v(i_0), v \in V \rangle$ will be called the layer i_0. I.e., the circuit parts united into a single target to deliver the stream into the vertex i_0 enter into the layer. For every vertex $\sum_{v \in V^{+}(i)} x_v(i_0) + q_i(i_0) = \sum_{v \in V^{-}(i)} x_v(i_0)$ is just, flow volume along entering arcs of plus stream out coming from the vertex itself equals to flow volume moving along out coming arcs. Therefore

$$\sum_{v \in V^{+}(i)} x_v(i_0) - \sum_{v \in V^{-}(i)} x_v(i_0) = -q_i(i_0), \quad i \in E \tag{1}$$

Following below is just for the vertex i0 :

$$Q_{i_0} = \sum_{i \in E} q_i(i_0) \tag{2}$$

Correlation (1) is the First Kirchhoff law for the circuit.

Let's denote the total stream $X_v = \sum_{i_0 \in E} x_v(i_0)$ -the total stream flowing to the arc v. From (1) it follows

$$\sum_{v \in V^{+}(i)} X_v - \sum_{v \in V^{-}(i)} X_v = -\sum_{i_0 \in E} q_i(i_0), \quad i \in E \tag{3}$$

Without restricting the generality let's assume that every vertex forms the layer, if for any vertex *i0* it does not exist, it means that $q_i(i_0) = 0, i \in E$. In the city traffic network (1), (2) are fulfilled for i0∈E. From non-accomplishability (3) there does not follow correctness (1)

2.3 Single layer system study- problem of evacuation

Search of initial permissible flows by means of maximum flow in the network. Balance algorithm idea is in searching initial permissible streams in the circuit with their subsequent transformation into the balance state. As every arc has restricted traffic capacity then we fulfill the existence of permissible streams with their search by means of maximum flow problem and solve with Ford-Fulkerson algorithm.

In the maximum flow problem the stream is allowed to pass from one initial vertex to another end vertex. All arcs have designed carrying capacity. To deduce the problem to this type let's add two fictitious peaks *ii* and *kk.* Let's connect ii to a stream source i_0. For it the carrying capacity equals to Q_{i_0}. Run-off from $q_i(i) > 0$ link via arcs with a vertex *kk*. For those arcs the carrying capacity equals to $q_i(i)$ accordingly. We obtain standard maximum flow problem, to solve it we apply any known algorithm. If it turns out that maximum flow is less than Q_{i_0}, one layer initial problem as well as the total problem accordingly have no solution. In that case minimal cut is beyond additional arcs.

If it has turned out that maximum flow equals to Q_{i_0} we obtain permissible flow which we transfer to balance state via invariant

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

transformations.

2.4 Invariant transformation of layer streams

Let's consider a random cycle C. Let's pose a random sense of rotation coinciding with a certain arc in a cycle u. Let's construct generic function $sign_C^u(v)$:

$$sign_u(v) = \begin{cases} 0, \text{ if } v \notin C \\ +1, v \in C, \text{arc direction coincides with the cycle by-pass direction}, \\ -1, v \in C, \text{arc direction is opposite to the cycle by-pass direction}. \end{cases}$$

Assume xv, v∈ V satisfies correlation (1). Let's take a random number θ, for all v∈ V , $\overline{x_v} = x_v + sign_C^u(v)\theta$, i.e. for cycles arcs the direction of which coincides with the one of rotation sense add θ to the magnitude of the stream xv, for cycle arcs the direction of which is contrary to pass-by direction deduct xv. Then $\overline{x_v}, v \in V$ meets the correlation (1).

2.5 Second Kirchhoff law for a layer

Let's consider one layer of traffic flow from the source with a number i0 into stream flow number j_0. Assume that at the motion herein the streams fall apart at vertex i and converge at vertex j. Assume some streams move the arcs which meet provision of First Kirchhoff law. There are minimum two ways of the stream thereof deliver from ***i*** to ***j***, let's denote them P1 and P2. Assume that the time of motion t1 along the path 1 is more than the time of motion t2 along the path 2. Then a part of the stream switches to the path 2. There will increase the stream along the path 2 and accordingly the time along the path 2. Simultaneously the stream magnitude along the first path decreases and accordingly the motion time along it. Switching ends when congruence t1=t2, or t1-t2=0 is reached. In general terms, there shall be performed congruence $\sum_{v \in V} sign_C^u(v)(\tau_v(x_v)) = 0$, In hydraulic circuit theory it is called as the Second Kirchhoff law. Its literary wording: In balance state the sum of the stream motion time change along the cycle equals to zero.

At random streams the congruence does not work. Using invariant transformation of streams can be expressed as

$$NB_u(\theta) = \sum_{v \in V} sign_C^u(v)\left(\tau_v\left(x_v + sign_C^u(v)\theta\right)\right)$$

Cycle balancing problem is in the definition of ***θ***, that $NB_u(\theta) = 0$.

Framework construction, fundamental cycles system.

It is known that if Kirchhoff law is applied to fundamental cycles system, it is performed at any cycle graph. Framework is a random tree, tops of which coincide with the ones of the assumed graph G, a framework example is given. Tree can be constructed by any algorithm, for instance, constructing the tree of the shortest paths with Dejkstra algorithm. Shortest paths mean the path lengths to the root. Arcs beyond the tree are chords. Chord and tree's arcs form a fundamental cycle C(u) – a cycle, formed with a chord u. Let's set by-pass direction to every cycle coinciding with a chord. Function construction for the cycle $sign_u(v)$ is not complicated

Reckoning of function argument limits of variation $NB_u(\theta)$.

Let's break down $NB_u(\theta)$ in to three parts: $NB_u(\theta) = NB_u^{\ 0}(\theta) + NB_u^{\ +}(\theta) + NB_u^{\ -}(\theta)$, The first part $NB_u^{\ 0}(\theta)$ consists of additive components with v, for which $sign_u(v) = 0$. The second part $NB_u^{\ +}(\theta)$ additive components with v, for which $sign_u(v) = 1$, the third part $NB_u^{\ -}(\theta)$ consists of additive components with v, for which $sign_u(v) = -1$.

1. It is evident that $NB_u^{\ 0}(\theta) = 0$.

2. $NB_u^{\ +}(\theta) = \sum_{v \in V,\, sign_u(v)=1} \tau_v(x_v + \theta)$. As $\tau_v(x\ ) > 0$ and ascendant per x (see Fig. 1), as $NB_u^{\ +}(\theta) > 0$ ascendant per ***θ***. For all $v \in V, sign_u(v) = 1$ is performed as $0 \le (x_v + \theta) \le x_{\max}$, or $-x_v \le \theta \le x_{\max} - x_v$. Thus we obtain that

$$\theta \in [\underline{\theta^+}, \overline{\theta^+}], \tag{4}$$

where

$$\underline{\theta^+} = \max_{v \in V,\, sign_u(v)=1} (-x_v), \quad \overline{\theta^+} = \min_{v \in V,\, sign_u(v)=1} (x_{\max} - x_v)$$

3. In a similar way to 2 we obtain $NB_u^{\ -}(\theta)$ as an ascendant.

$$\theta \in [\underline{\theta^-}, \overline{\theta^-}], \tag{5}$$

where

$$\underline{\theta^-} = \max_{v \in V,\ sign_u(v)=-1} (x_v - x_{\max}), \quad \overline{\theta^-} = \min_{v \in V,\ sign_u(v)=-1} x_v$$

From considered cases 1-3 A (4), (5) it

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

proceeds that the function $NB_u(\theta)$ is ascendant and defined at the section $\theta \in [\underline{\theta}, \overline{\theta}]$, where $\underline{\theta} = \max(\underline{\theta^-}, \underline{\theta^+})$, $\overline{\theta} = \min(\overline{\theta^-}, \overline{\theta^+})$.

4. Searching algorithms of city transport system balance state

Given designs allow applying algorithms of consequent cycles balancing to search balance states of one layer. For example, we search the arc for which $NB_u(0) > \varepsilon$ (sufficiently small number), if there is no such an arc we stop the layer balancing and solve the task $NB_u(\theta) = 0$ for the problem thereof and pass over to the algorithm execution over again. For multilayer systems the arc search is fulfilled along all layers and inside the layer accordingly.

2.6 Problem on maximum flow

In many network problems, it is meaningful to consider the arcs as certain communication having definite flowing capacity. In this case, as a rule, there considered the task of some flow maximization, directed from the selected vertex (source) to some other vertex (outflow). Such type of task is called the problem of maximum flow.

Suppose we are given orient graph $G=\langle E,V,H \rangle$, in which direction of every arc v∈V denotes the flow motion direction, flowing capacity of each arc equals to dv. At vertexes of multiple E there distinguished two vertexes: start and end. Accordingly, vertex н is the source of the flow, к – is the outflow. It requires maximum flow, which can pass from vertex н to к.

Let us denote as xv flow level passing along the arc v. It is obvious, that

$$0 \le x_v \le d_v,\ v \in V \tag{6}$$

In every vertex, the incoming flow level equals to outgoing flow level. That is, following congruence is true

$$\sum_{v \in V_i^+} x_v = \sum_{v \in V_i^-} x_v \tag{7}$$

or

$$\sum_{v \in V_i^+} x_v - \sum_{v \in V_i^-} x_v = 0 \tag{8}$$

Accordingly, to vertexes *н* and *к* there executed

$$\sum_{v \in V_i^+} x_v - \sum_{v \in V_i^-} x_v = -Q \tag{9}$$

$$\sum_{v \in V_i^+} x_v - \sum_{v \in V_i^-} x_v = Q \tag{10}$$

Magnitude Q is value of the flow, outgoing from vertex н and incoming into vertex к.

Problem. Define:

$$Q \to \max \tag{11}$$

at delimitations (6) – (10).

Values (Q, xv , v∈V) satisfying delimitations (6) – (10) will be named as flow in the network, and if they maximize the magnitude Q, then as maximum flow. It is easy to see that values Q=0, xv=0, v∈V, is the flow in the network. Problem (6) – (10) is the task of linear programming and can be solved applying simplex algorithm.

Let us break multiple of vertex E into two nonintersecting parts E1 and E2 in such a way, that н∈E1, к∈E2. Crosscut R(E1,E2), separating н and к we will name such multiple R(E1,E2)⊂V, that for every arc v∈R(E1,E2) or h1(v)∈E1 and h2(v)∈E2, or h1(v)∈E2 and h2(v)∈E1.

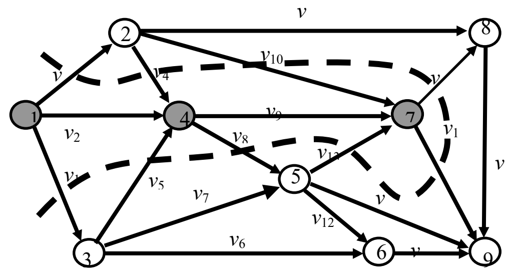

Figure 2: Search for crosscut

There is multiple E1={1,4,7} on Fig.1, these vertexes have dark filling. E2={2,3,5,6,8,9}. Crosscut R(E1,E2) represent arcs, which dotted line went through.

Let us break multiple R(E1,E2) into two parts as follows:

R+(E1,E2)={v∈R(E1,E2)| h1(v)∈E1 and h2(v)∈E2},

R–(E1,E2)={v∈R(E1,E2)| h2(v)∈E1 and h1(v)∈E2}.

Elements of the multiple R+(E1,E2) we will name straight arcs, they lead from multiple E1 to E2. Elements of the multiple R–(E1,E2) are backward arcs, they lead from multiple E2 to E1. Flow through the crosscut we will name the value

Crosscut flowing capacity we will name the value

It is obvious that 0≤X(E1,E2)≤D(E1,E2). Next theorem is true.

Proposition 1. In any network the magnitude of maximum flow Q from the source н

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

to overflow к equals the minimal flowing capacity D(E1,E2) amongst all crosscuts R(E1,E2), separating vertexes н and к.

Crosscut R(Ē1, Ē2), with Q=D(Ē1,Ē2) we will name constraining. At constraining crosscut, there is executed

Let us assume, that (Q, xv, v∊V) is a flow in the network, and succession н=i0, v1, i1 , v2, i2, vK, iK=к is a circuit connecting vertexes н and к. Define on that circuit motion direction from vertex н to к. Arc vj from that circuit is called straight, if its direction coincides with motion direction from н to к, and backward, if not. Circuit will be called flow increasing circuit, if for straight arcs of the circuit v (dv– xv)>0 and for backward xv>0. Through the circuit thereof it is possible to pass additional flow q from н to к with value q=min(q1,q2), where q1=min(dv – xv), minimum is taken from all straight arcs of the circuit, q2=min(xv), minimum is taken from all backward arcs of the circuit.

Proposition 2. Flow (Q, xv, v∊V), is maximum, then and only then, there is no way to increase the flow. Offered algorithm for solving the problem of maximum flow in the network is based on searching an increasing flow in the circuit from н to к. The search, in its turn, is based on the process of vertexes marks disposition similar to Dejkstra algorithm.

Let us add mark Pi=[gi,vi,θ] to every vertex i, where gi – value of additional flow entered the vertex i, vi – arc through which the flow entered, θ – sign «+», if the flow entered along the arc vi , directed to i (along straight arc); θ – sign «–», if the flow entered along the arc vi , directed from i (along backward arc),

Let us say that vertex i:

- is not labelled, if the additional flow does not reach it, the label will have the form Pi=[0,–,θ],

- is labelled, but not viewed, if the flow has reached it, but has not been allowed to go further, the label will have the form Pi=[gi,vi,θ], where gi>0,

- labelled and viewed, if the flow reached it and allowed to go further, label will have the form Pi=[gi,vi,θ].

Let us consider solution algorithm.

0. For all v∊V assume that xv=0, assume that Q=0.

1. All vertexes are unlabeled. Vertex н is labelled, but not viewed with a label Pн=[∞,–,–]. It means that the unlimited volume flow enters that vertex.

2. Search labelled but not viewed vertex. If it is not available, then the found flow Q, xv, v∊V is maximum and algorithm completes its function. If such vertex is found, i – is its number, then pass on to 3.

3. View vertex i:

- for all assume j=h2(v). If vertex j is unlabeled and (dv–xv)>0, then mark it with label Pj=[q,v,+], where q=min(qi, (dv–xv)), if j=к, then pass on to point 4.

- for all assume j=h1(v). If vertex j is unlabeled and xv>0, then mark it with a label Pj=[q,v,–], where q=min(qi, xv), if j=к, then pass on to point 4.

- label vertex i as viewed and pass on to point 2.

4. Pass additional flow. Let us assume that j=к, q=gк and v=vj.

- if θ=«+», then it is necessary to fulfill: Let us assume that xv=xv+q, i=h1(v), if i=н, then pass on to point 1, otherwise put j=i and pass on to v=vj),

- if θ=«–», then it is necessary to fulfill: Let us assume that xv=xv–q, i=h2(v), if i=н, then pass to point 1, otherwise put j=i and pass on to v=vj).

Because of the algorithm execution there will be obtained the flow (Q, xv, v∊V). To search the crosscut with minimal flowing capacity part of vertexes should be labelled and viewed at the final stage of algorithm operation in point 2, we include these vertexes into multiple Ē1, Ē2=Ē\ Ē1. Cross cut R(Ē1, Ē2) will be the sought for [7].

3. PROBLEM SETTING

Hereby we specify a graph G=<E, V, H>, in which each arc direction vϵV indicates the flow motion direction, each arc capacity equals to dv. There are class rooms at vertex set E. At vertex sets E there differentiated two vertexes: start and end. Vertex 0 is the flow source, 35 outlet. For i from E there are given 2 numbers: amount of people sitting there and number of people running out from there. Arcs are corridors and stairwells between constructions [6].

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

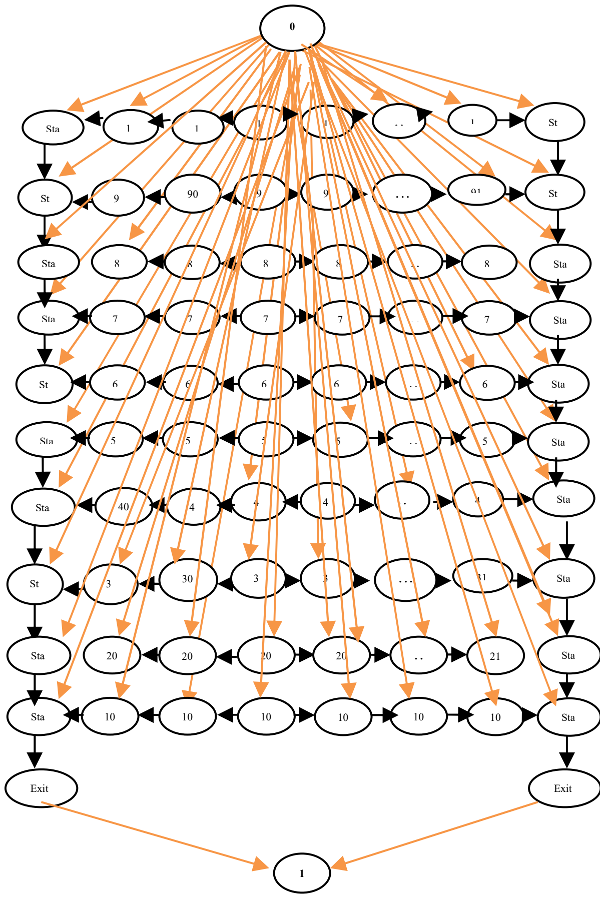

Figure 3: KazNRTU main academic building in the form of a diagram.

There is presented deca-floors academic institution. Let's say that there is the necessity to evacuate people in connection with the occurred emergency situation. As the emergency situation alarmed during the academic studies, accordingly all class rooms are occupied. There is a definite students' amount in every class room. There are from 25 to 35 class rooms at every floor. Between the floors there are 4 stairwells. The building has 4 exits, two of which are main, two emergencies.

It is very important for the people under emergency situations (fire, earthquake, etc.) to leave promptly the building they are in.

To find a maximum efficient evacuation plan we have the following data:

1. Model (object) – Kazakh National Research Technical University named after Satpayev K.I. (indoor premises model is in AutoCAD, figure 4).

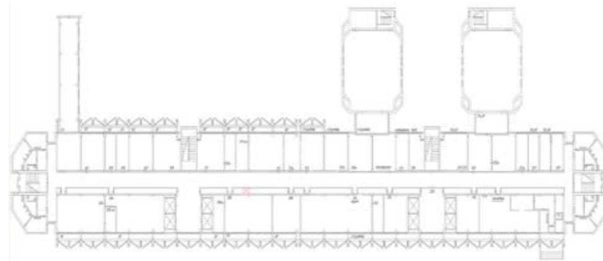

Figure 4: KazNRTU building plan (main academic building, 2nd floor)

2. KazNRTU current time-table (figure 5).

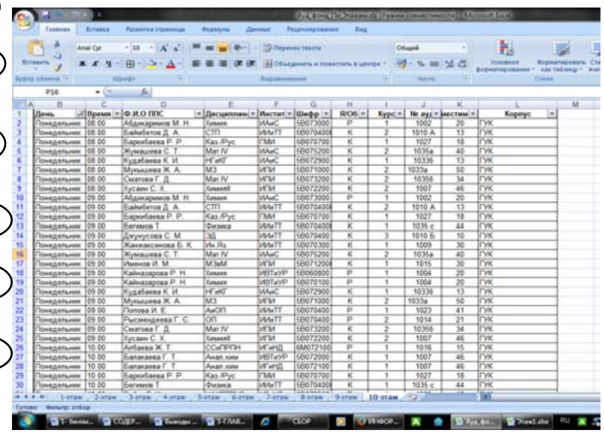

Figure 5: KazNRTU time-table

For optimal flow distribution, to avoid any congestion during evacuation process it is necessary to take into account the number of people in the classrooms which differs dependent on the time and schedule.

Such approach secures quick evacuation without congestion, which gains time and reduces human life risks.

Principal tasks are:

- To calculate the distance from every class room to every exit.
- To calculate the factor of exits capacity.
- In compliance with a time-table to calculate the number of people in every class room.
- To organize every exit (if there are more than 2) per every class room priority.
- To execute an algorithm with account of people flow distribution.
- To find an optimal and efficient route to avoid congestion and to secure quick evacuation.

4. SOFTWARE AND TOOLS

Medium NetBeans IDE is free, with an open initial code, integrated development

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

environment (IDE), which allows elaborating the desktop, mobile and web applications. IDE supports developing applications in different languages, including Java, HTML5, PHP and C++. IDE represents an integrated support of complete cycle development, starting from creating the project assisted with debugging, profiling and scanning. IDE operates at the systems based on Windows, Linux, Mac oS X and Unix [13].

IDE creates extensive support for JDK 7 technologies and the latest Java advanced. It is the first IDE, which supports JDK 7, Java EE 7 and JavaFX 2. IDE completely supports Java EE by means of the latest standards for Java, XML, web services, SQL and GlassFish Server, reference implementation execution in Java EE.

As the program will show emergency evacuation optimal plan, it is necessary to have the building plan. KazNRTU's building plan shall be represented in format JPEG or PNG, for the aim of its appearance in the beginning of the program.. The building plan herein features every floor separately (from the 1st to the 10th floor).

After extraction the data from the list a new file shall be reconstructed txt (figure-6), containing the following information:

- Class room number,
- lecture (start time),
- number of people in the classroom.

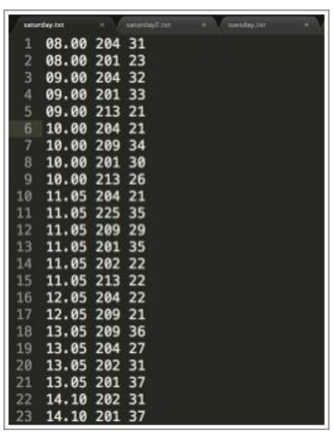
```
1  08.00 204 31
2  08.00 201 23
3  09.00 204 32
4  09.00 201 33
5  09.00 213 21
6  10.00 204 21
7  10.00 209 34
8  10.00 201 30
9  10.00 213 26
10 11.05 204 21
11 11.05 225 35
12 11.05 209 29
13 11.05 201 35
14 11.05 202 22
15 11.05 213 22
16 12.05 204 22
17 12.05 209 21
18 13.05 209 36
19 13.05 204 27
20 13.05 202 31
21 13.05 201 37
22 14.10 202 31
23 14.10 201 37
```

Figure 6: New file txt

As the Figure 6 shows, the diagram is constructed according to the three aspects and broken down into week days.

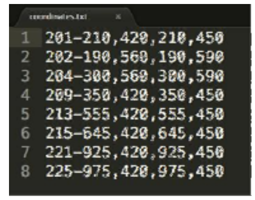
```
1 201-210,420,210,450
2 202-190,560,190,590
3 204-300,560,300,590
4 209-350,420,350,450
5 213-555,420,555,450
6 215-645,420,645,450
7 221-925,420,925,450
8 225-975,420,975,450
```

Figure 7:Class rooms coordinates

As far as the software developed in Java and has a potential to draw the route from a class room to an exit, it is obvious that, Canvas shall be used. To show number of people in lecture halls and draw lines between the objects there used coordinates, which lead to another text file with the following information:

– lecture hall number,
– X axis coordinates,
– Y axis coordinates

For instance, on the picture-6, there are 8 classrooms on the second floor. Next to a classroom number there figures, which are coordinates along X and Y comma separated.

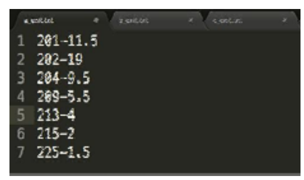
```
1 201-11.5
2 202-19
3 204-9.5
4 209-5.5
5 213-4
6 215-2
7 225-1.5
```

Figure 8: Priority exit

To develop optimal evacuation algorithm it is necessary to be aware of «priority exits» for each classroom, be based on the distance. As there stream of people, we have 4 different files, showing the distance from every classroom.

File contains following data:

-lecture (amount of people),

-distance from every classroom to the exits (figure 8).

The files thereof will be used for people distribution from a hall to an exit in the most optimal way in order to avoid congestion. Based on that distance the algorithm thereof allows calculating number of people from every classroom to an exit. The sum will vary dependent on the distance to an exit.

Final product shall contain building plan picture and routes to each exit. It means that Canvas has 2 layers:

-picture JPEG (building plan),

-created with a mark.

Next step is receiving current data, time exclusive, in order to get wanted amount of people

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in the classroom. For that purpose better to use the class under the name "Calendar" and simple methods, which can choose time, including hour, minute, etc.

To receive current number of people in the classroom, it is necessary to extract data from monday.txt and add all lines in the array list (figure ок 9).

```
try {
    InputStream instream = new FileInputStream(path);
    InputStreamReader isr = new InputStreamReader(instream,Charset.forName("UTF-8
    BufferedReader bufread = new BufferedReader(isr);
     String line = "";
    while((line=bufread.readLine() )!=null){
       schedule.add(line);
    }

} catch (Exception ex1) {

}
ArrayList<String> ttt = new ArrayList<>();
for(int i=0;i<schedule.size();i++){
    ttt.add(schedule.get(i).split(" ")[0]);
```

Figure 9: Data retrieval from monday.txt.

Further to add to the same array list current time and sorting. Orienting oneself at the current index in the choice list, the previous one will become the start time, and that line will show flux density in the classrooms (figure 10).

```
ttt.add(time);
Collections.sort(ttt);
for(int i=0;i<ttt.size();i++){
  //  System.out.println(ttt.get(i));
}
begin_time = ttt.get(ttt.indexOf(time)-1);
```

Figure 10: Current time adding

By means of coordinates.txt number of people should be denoted at the number of picture. For that reason lecture halls coordinates shall be extracted from txt file and the information shall be denoted through the use of previous choice lists.

Algorithm starts from the cycle for, which is performed 4 times, as there are 4 main exits, if it equals to the null, it needs to consider lecture halls with «a» priority exit, if i equals to 1, then exit «B», etc. (figure 11).

```
for(int i=0;i<4;i++){

    if(i==0){
        negative_number=0;
        temp_array = new ArrayList<>();
        for(int j=0;j<rooms_priority_people.size();j++){
            if(rooms_priority_people.get(j).charAt(3)=='a'){
            temp_array.add(rooms_priority_people.get(j));
            }

        }
```

Figure 11: «a» exit priority.

Each exit has definite number of people, which are appropriate to avoid some clashes. The number herein equals to 60. When amount of people in each lecture hall with a priority «a» constitutes over 60, there is applied the method, organizing the way out of definite people amount from a lecture hall, which equals to 60 sharp. Therefore we shall calculate the first sum (figure 12).

```
//calculate sum of students
for(String line : temp_array){
// System.out.println(line);
sum = sum + Integer.parseInt(line.substring(7, line.length()));
}
System.out.println("sum is!-!_!_!_!_!_!"+sum);
//check if its more that 60
```

Figure 12: People amount calculation.

In case the sum is less than 60, it is needed to receive all people from a lecture hall hall, direct them on the given exit, maintaining in the choice list, which is called final_results_for_a.

In case the sum is more than 60, then we fulfill getDistances, which chooses lecture hall parameter and priority exit.

As soon as we execute getDistances, by means of the second parameter we define corresponding file in the format txt. For example, if priority equals to «a», we define the file a_exit.txt.

To calculate an exact number of people from every lecture hall, it needs to concentrate on the file content. txt. As mentioned above, it already has defined the distance from a lecture hall to exits. Having determined every distance it is possible to calculate an exact number of people from every lecture hall.

Next step: subtract from the initial sum the percent, calculated before. If in the result there is a negative number, it means that some people will be added from the next lecture hall. If in the result there is a positive number, it means that some people remain in the lecture hall thereof, they will be directed to another exit.

Next step: to save the result in the array list. Every exit has its own array list. Having added the percent sum to a lecture hall, we'll have one line, which appears approximately as:

201/15, 202/3, 204/19, 209/15, 213/14, 215/20, 221/31, 225/25.

Subsequent to the result saving in the corresponding choice list it is necessary to replace a global array list (rooms_priority_people), which contains total information, by the new array list by means of replace method. Next step: check that nobody remained in the lecture hall. In case there rooms_priority_people with 0 people, it shall be deleted. For that purpose there should be used remove_zeros method.

Last step: to perform priority exit. After 60 people way out, move ahead using other exits. which means, that an exit «a» switches over at the last place and an exit «B» will be on the first one.

It continuous until the cycle reaches the last exit. Every exit repeats above denoted actions and saves them to reduce to the corresponding choice list And finally, the latest result will have the following image, as it is shown in the picture 13.

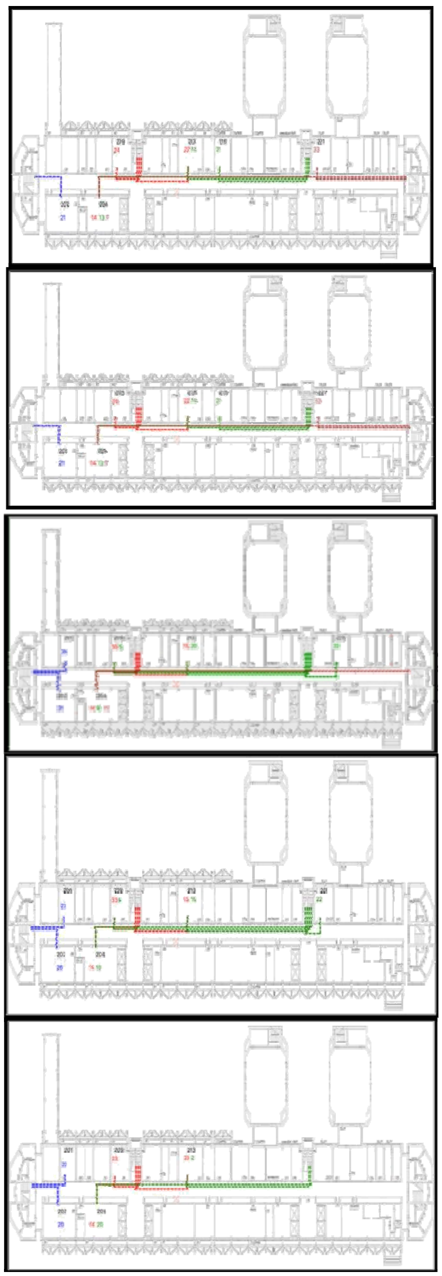

Figure 13:People load flow according to the time table.

Time to consume for standard and extended evacuation

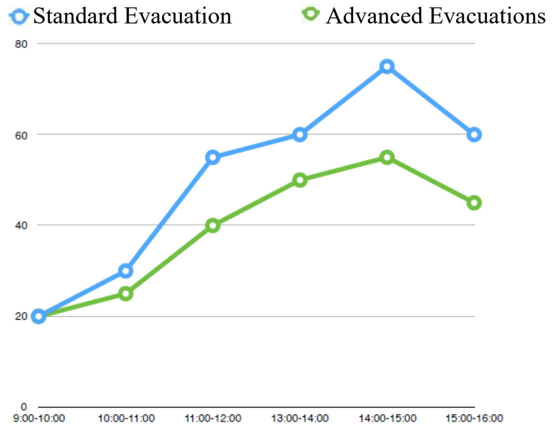

Figure 13: Time to consume for standard and extended evacuation

CONCLUSION

In this work paper questions of mathematical modeling of a stream which in turn, are applied to the solution of a problem of evacuation with scheduling were considered.

Namely, the following tasks are carried out:

- the mathematical problem definition were given where algorithm of the solution of a task, a task about the maximum stream, a method of potentials and criterion of an optimality, Ford and Falkerson's algorithm were considered.
- search of admissible solutions of a task on the basis of a task about the maximum stream is carried out,
- search of the shortest way with use of algorithm of search of an equilibrium state in the description of model of the movement of flows of people.

For mathematical representation of a problem of evacuation questions of information modeling with possibility of obtaining density are provided further in audiences from the educational schedule of university, and also as more reliable method for determination of density of people in the building - use of wireless touch systems

Natural calamities can result in substantial damage for the human society and, above all, constitute the risk for human life. In each case of unexpected emergency situation it is important to evacuate people as soon as possible.

Developed software is designed for quick and maximum efficient people evacuation from an academic institution and can be used for other types of buildings.

REFRENCES:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[1] Neiman J., Morgenshtern O. Games theory and economic behavior. Moscow.: Nauka, 1970. – 244p.
[2] Zamkov O.O., Tolstopyatenko A.V., Cheremnykh Yu.N. Mathematical methods in economics . Moscow: MSU, 1997. – 366p.
[3] Kuznetsov J.A., Sakovich V.A., Kholod N.I. Mathematical Programming. Minsk: Higher School,2001. – 352p.
[4] Amirgaliyev Y, Kovalenko A, Kalizhanova A, Amirgaliyeva Zh Efficient Algorithm for Evacuation Problem Solving. 2015 15th International Conference on Control, Automation and Systems (ICCAS 2015), 2015. - P. 1587-1592.
[5] Amirgaliev E.N., Kalizhanova A.U., Kozbakova A.Kh. Mathematical methods and models in an emergency evacuation // Vestnik KazNRTU after Satpaev K.I. - Almaty. - 2015. - №#6 (112). – p.p. 231-235.
[6] Kovalenko A.G., Vlasova I.A., Borisova S.P. Games theory and operations research. Samara: Publishing house «Samarsky University», 2006. – 147p.
[7] Volkov I.K., Zagoruiko Ye.A. Operations research. Moscow: Publishing house of MSTU named after Bauman N.E., 2000. – 435p.
[8] Vasin A.A., Morozov V.V. Games theory and econometric models. Moscow: MAKC Press, 2005. – 237p.
[9] Gorlach B.A. Operations research. S-Petersburg: Publishing house «Lanj», 2013. – 448p.
[10] Kosorukov O.A., Mischenko A.V. Operations research. Moscow: Publishing house «Ekzamen», 2003. – 448p.
[11] Belyayev S.V. Evacuation of mass public buildings. M: Publishing house of «All-Union Academy of architecture », 1938. – 257p.
[12] Germeiyer Yu.B. Introduction into the theory of operations research. M.: Nauka, 1971. – 358p.
[13] https://netbeans.org/features/index_ru.html.