Journal of Theoretical and Applied Information Technology
31st July 2021. Vol.99. No 14

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

MATHEMATICAL MODELS AND METHODS FOR SOLVING THE PROBLEM OF EVACUATION

[1]AINUR KOZBAKOVA, [2]ALIYA KALIZHANOVA, [3]FERUZA MALIKOVA, [4]ZHAZIRA AMIRGALIYEV,[5]TIMUR KARTBAYEV, [6]BYRZHAN SHARIPOVA, [7]ZHALAU AITKULOV, [8]AYGERIM ASTANAYEVA
[1,2,3,5,6]Almaty Technological University
[1, 4,7,8]Institute of Information and Computing Technologies CS MES RK, Almaty, Kazakhstan
[2]Almaty University of Energy and Communications, Almaty, Kazakhstan

E-mail: [1]ainur79@mail.ru

ABSTRACT

The problem of evacuation of people from closed premises such as universities, colleges and schools is considered. The peculiarity of this work lies in the formation of an integrated approach for organizing the evacuation process in peacetime as a training for an emergency. A conceptual diagram of an evacuation system is proposed that uses heterogeneous sources for receiving and transmitting information about the onset of an emergency. The input and output sources for receiving and transmitting information about the number of people in the building are determined. The main purpose of the system is to form an operational evacuation plan in real time. The optimal solution to the problem of maximum network flow is implemented using a game-theoretic approach. A mathematical model has been developed for the optimal distribution of the flow along the grindshill network with the analysis of the flow formation and the characteristics of the ways of people moving in closed spaces. A game-theoretic approach and mathematical methods of the theory of hydraulic networks for finding an equilibrium state in flow-distribution networks have been developed. An algorithm for solving the evacuation problem using the graph approach is proposed.

Keywords: *Mathematical modeling, graph, method, algorithm, maximum flow, evacuation.*

1. INTRODUCTION

For the evacuation planning process, a variety of methods and algorithms have been presented, in paper [1], evacuation planning was considered directly on dynamic issues related to time-varying and volume-dependent. The article [2] distinguishes between macroscopic and microscopic models of evacuation, which are able to record the movement of evacuees in time. The article [3] considers a mathematical model describing the motion of dynamic flows in a directed graph. The model parameters include the undirected graph as the building model, the initial flow values, the flow sources, and their receivers. In this paper [4] present a new mathematical model of rescue-evacuation and develop a method for a quick solution for emergency response in real time for various population groups and various means of evacuation, based on the iterative use of a modification of the planning algorithm. This article [5] describes the maximum flow in a bipartite dynamic network. The main idea behind these improvements is the rule of pushing out two arcs in the case of maximal algorithms. In this article [6] proposed and analyzed an algorithm for Dynamic Real-Time Bandwidth Sharing Routing (DRTCCR). Such an algorithm would investigate the capacity constraints of the evacuation network in real time by modeling capacity based on time series to improve current solutions to the emergency route planning (ERP) problem. The article [7] presents a highly polynomial time algorithm for calculating an approximate solution to the fastest partial contraflow problem on two terminal networks, which is justified by numerical calculations that consider the Kathmandu road network as an evacuation network. The work [8] describes the evacuation decision model proposed in this article consists of three parts: a model for predicting the distribution of pedestrians, a model for calculating pedestrian flow, a situation on the way, and a model for correcting feedback.

Evacuation is one of population protection means. It is taking out or withdrawal of people from hazardous areas. It could take place both in peacetime and wartime. Evacuation as a means of population protection used long time ago.

Actuality of evacuation as the means of population protection in wartime and peacetime during recent years not only decreased but also increased. Contemporary life experience says that the population increasingly runs into danger in the result of natural calamities, accidents and disasters in industry and transport. Think for instance of natural calamities, earthquakes, floods, snow slides, mud streams and earth falls, wild scale forest fires. In such cases, evacuation is usually unavoidable. Evacuation measures are taken at accidents at atomic power stations, emissions and flood of hazardous chemicals and biologically damaging substances, at vast fires at petrochemical and oil refineries.

In the posted task, we consider people evacuation from educational institution in the emergency situation. The main peculiarity of educational institution's buildings is instability of people distribution along internal premises connected with the lectures timetable. It requires assessing the lessons schedule with regard to organizing unobstructed movement of people. As announced earlier the topic herein is and will be acute as, unfortunately, emergencies happen increasingly frequently. To solve the given problem there used mathematical methods and model of people flows motion inside the building.

The practice of modern life shows that the population is increasingly exposed to dangers in the result of natural disasters, accidents and catastrophes in industry and nuclear power plants, earthquakes, floods, avalanches, mudflows, landslides, mass forest fires, spills of chemically hazardous substances and biologically harmful substances, large fires in petrochemical and refineries. In all those cases, it is almost necessary to resort to evacuation. The task of constructing effective evacuation measures is of paramount significance, since people's life and the preservation of material values depend on that [9,10].

Evacuation models are designed primarily for the operational conduct of evacuation processes, aimed at saving the lives of people and material assets of the enterprise, forecasting and timely determining the time of evacuation of people. Very often, such models make it possible to determine possible areas of congestion during evacuation [11].

Many existing models include features such as visualizing the flow of people, modeling human behavior, determining the best evacuation routes, etc.

The use of mathematical methods and information technologies significantly increases the efficiency of the evacuation systems, therefore, the development of new integrated and intelligent info communication approaches using sensors for receiving and transmitting information to solve the evacuation problem is currently very relevant.

At the present stage, evacuation computer models are the most effective tools for investigating and optimizing the evacuation process. By now, a large number of such computer models have been created [12].

Thus, the scientific and technical significance of the project consists in constructing an optimal plan for real-time evacuation from a different building based on the development of mathematical and information optimization models with the condition of fulfilling Nash equilibrium and software and hardware for the implementation of flow distribution for various types of buildings [13].

We offer a conceptual diagram of the evacuation task based on heterogeneous systems of reception and transmission of information (Figure 1). The development of a computer model of the information system will be carried out in accordance with the proposed conceptual schemes shown in Figures 1.

In terms of mathematical modeling, effective coverage areas for the selected type of building will be investigated (universities, schools, industrial premises, office building, business centers, etc.). As a result of the study on finding the optimal coverage, sensors for receiving information transmission will be installed, information flows will be determined, information processing technology from sensors will be proposed, warning methods for evacuation processes will be proposed, both in the training mode and in cases of emergency requiring evacuation.

The use of mathematical methods and information technologies significantly increases the efficiency of evacuation systems, therefore, the development of new integrated and info communication approaches to the evacuation problem is currently relevant.

In the scientific plan, mathematical models of multi criteria optimization are developed based on Nash conditions, there proposed models for optimal coverage of building areas, technical means for receiving information transfer,

Journal of Theoretical and Applied Information Technology
31st July 2021. Vol.99. No 14

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

technology for processing large data in the flow distribution problem, and offered the training regimes for evacuation processes. Successful implementation of these solutions ensures effective evacuation, which, surely, entails a social and economic effect.

2. ALGORITHM OF SOLVING THE TASK ON EVACUATION

Let us suppose that an emergency has happened in an educational institution bringing to the necessity to evacuate people. There are 24 classrooms with 30 students in each, and 8 stair wells and 2 exits. It is necessary to calculate the time, speed and direction of students' evacuation from the educational institution. Let us specify a graph $G=<E, V, H>$, in which direction of every arc $v \epsilon V$ identifies direction of flow motion, flow capacity of each arc equals to dv. Auditoriums are in E vertexes multiple. There identified two vertexes "start" and 'end' in E vertexes multiple. Vertex 0 is the stream source, 35 flowing. For i from E there given 2 numbers: amount of people sitting there and amount of people rushing out of there per time unit. Arcs are corridors and stir wells between the nodes.

As every arc has limited flowing capacity, the check of existing permissible flows along with their search can be fulfilled by means of the task on maximum flow and solving it with Ford-Fulkerson algorithm [14].

In the task on maximum flow, the flow is passed from one initial vertex to one final. All arcs have prescribed flowing capacity. To arrange that type of the task let us add two dummy vertexes *ii* and *kk*. Let us connect *ii* with stream source i_0. Its flowing capacity equals to $-q_{i^0}(i_0)$. Flowing c $q_i(i) > 0$ is connected with arcs to the vertex *kk*. Capacity of those arcs is $q_i(i)$ accordingly. We obtain the task on standard maximum flow and apply any known algorithm for its solving. If it turned out that maximum flow is less than $q_{i^0}(i_0)$, then initial task of one layer and accordingly the whole task has no solution. In that case, the minimal cut is beyond additional arcs [15].

If it turned out that maximum flow equals to $q_{i^0}(i_0)$, we obtain permissible flow, which is transferred to the state of equilibrium by invariant transformations.

Let us describe people's flow movement along a corridor and staircase by means of Grindshiels formula. Let us introduce following designations: L – network section length, T – time of moving along the section, x – flow having passed the road section for time unit, P – flow density, S – number of lanes, W – speed of flow, λ – average corridor length.

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

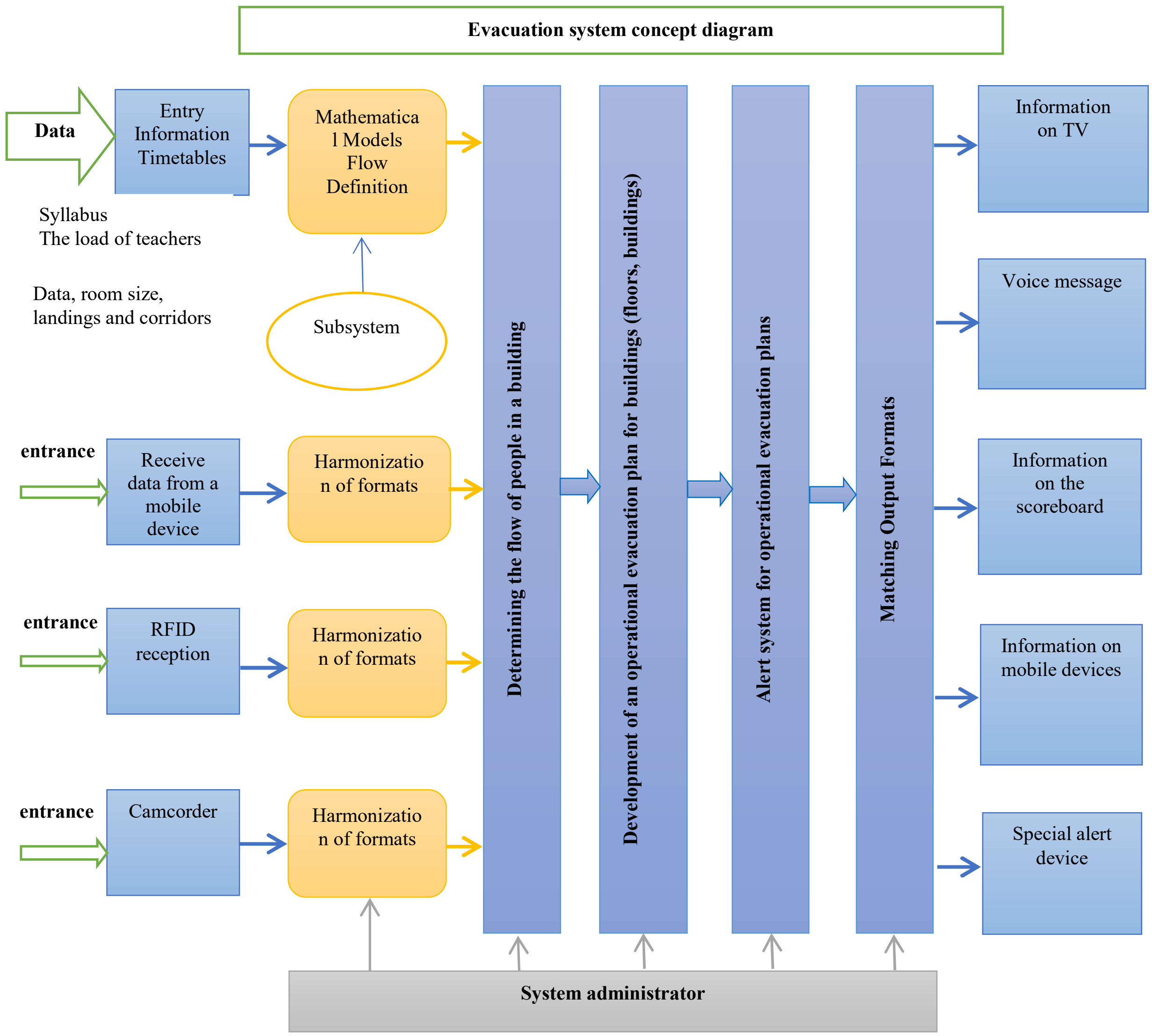

Figure 1: Conceptual Diagram Of The Evacuation System

2.1 Sections and Subsections

As defined, the density is $\rho=1/\lambda$. Assume W – student's speed, W_{max} – maximum speed. Time, which gets a man to travel a route section of λ length equals to $\tau=\lambda/v$. Amount of students per time unit will be equal to $\kappa=1/\tau$. Therefore, $x = ks = \frac{1}{\tau}S = \frac{w}{\lambda} = wps$. We'll consider that the flow speed and density are interconnected due to linear dependence $w/w_{max} + p/p_{max} = 1$ (Grindshiels formula) [16].

Therefrom $w = w_{max}(1 - p/p_{max})$, or $p = p_{max}(1 - w/w_{max})$. Let us insert it, and obtain $x = swp_{max}(1 - w/w_{max})$. Obtained function is a parabola with branches downward directed, maximum is achieved at $w = w_{max}/2$, and accordingly $x_{max} = s(w_{max}p_{max})/4$.

Thus, we obtained the magnitude of maximum flow, which can be passed through.

Let us insert instead of ρ the expression and receive the following formula $w^2 - w_{max}w + \left(\frac{w_{max}}{sp_{max}}\right)x = 0$.

According to Viete formula we obtain $w = w_{max}(1 + \sqrt{1 - x/x_{max}})/2$, taking into account that every member strives to maximize own speed.

From here we receive that the time of travelling along the network section is expressed with following dependence:

$T(x) = 2T_{min}/(1 + \sqrt{1 - \frac{x}{x_{max}}})$, where T_{min} – minimal travelling time along the section in case the flow along it equals to zero. Let us consider evacuation movement route. Based on investigation data the width can be accepted as 0, 6 м, with supposition of its small reduction for the roads with the width in several flows. Apart from that, in view of necessity, irrespective of the road width, in case of possibility of occasional opposing traffic or overdrive at traffic delay, the path with width of one flow should be accepted with some width reserve. Considering this and the necessity of flow number at existing and adaptable evacuation routes, we can give a table for defining flow number per width both of horizontal route and of staircases.

Table 1. Determination of flows number

Number of elementary flows	Width of evacuation route		
	Normal	Minimal	Maximum
1	0,9	0,9	1,2
2	1,2	1,2	1,7
3	1, 8	1,7	2,3
4	2,4	2,3	3

In practice, mass movement speed fluctuates from 5 to 75 m per minute. At sustained motion, density cannot reach physically maximum amount, therefore it is rational to accept the length of the route as calculation basis. At that, speed specified values are defined for horizontal path as 16 meters per minute, for descent down staircase as 10 meters and for ascent 20% less, that is as 8 meters per minute [17].

Flowing capacity of elementary stream per minute is defined as fraction of speed division by flow density. Total capability is defined by multiplying the obtained value by flows number at route width and by number per minute, making up evacuation duration. It is evident hereof, that such product, depending on evacuation motion factors total, cannot be constant value, as it is recommended by existing norms, but it is, to a significant extent, a variable value, depending on local conditions and increasing proportionally to increase of evacuation permissible duration. Time allowance directly influences at permissible route length.

For the first stage, the route length characterizes ultimate moving away from the exits and has importance mainly for big buildings. For the sum of the first and second stages the norms herein determine laying out of separate floors in ratio of number and location of exits to outside or to the staircases. For the sum of three stages, the same norms influence at laying out in whole limiting number of floors, and prescribing premises grouping per floors in such a way, that the first and second stages could decrease in proportion to the increase of the third one [18].

2.2 Problem on maximum flow in the network

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.

Let us assume that there is an orient graph $G=\langle E,V,H\rangle$, in which direction of every arc $v\in V$ 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.

Vertex h is the source of the flow, k – is the outflow. It requires maximum flow, which can pass from vertex h to k.

Let us denote as x_v flow level passing along the arc v.

It is obvious, that

$$0 \le x_v \le d_v \, , \, v\in V \qquad (1)$$

In every vertex $i\epsilon E$ {н, к} the incoming flow level equals to outgoing flow level. That is, following congruence is true

$$\sum_{v\in V_i^+} x_v = \sum_{v\in V_i^-} x_v \qquad (2)$$

or

$$\sum_{v\in V_i^+} x_v - \sum_{v\in V_i^-} x_v = 0 \qquad (3)$$

Accordingly to vertexes h and k there executed

$$\sum_{v\in V_i^+} x_v - \sum_{v\in V_i^-} x_v = -Q \qquad (4)$$

$$\sum_{v\in V_i^+} x_v - \sum_{v\in V_i^-} x_v = Q \qquad (5)$$

Magnitude Q is value of the flow, outgoing from vertex h and incoming into vertex k.

Problem. Define:

Journal of Theoretical and Applied Information Technology
31st July 2021. Vol.99. No 14

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

$$Q \to \max \qquad (6)$$

at delimitations (1) – (5).

Values (Q, xv , $v \in V$) satisfying delimitations (1) – (5) 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 $Q=0$, $xv=0$, $v \in V$, is the flow in the network. Problem (1) – (5) 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 $h \in E1$, $k \in E2$. Crosscut $R(E1,E2)$, separating h and k we will name such multiple $R(E1,E2) \subset V$, that for every arc $v \in R(E1,E2)$ or $h1(v) \in E1$ and $h2(v) \in E2$, or $h1(v) \in E2$ and $h2(v) \in E1$.

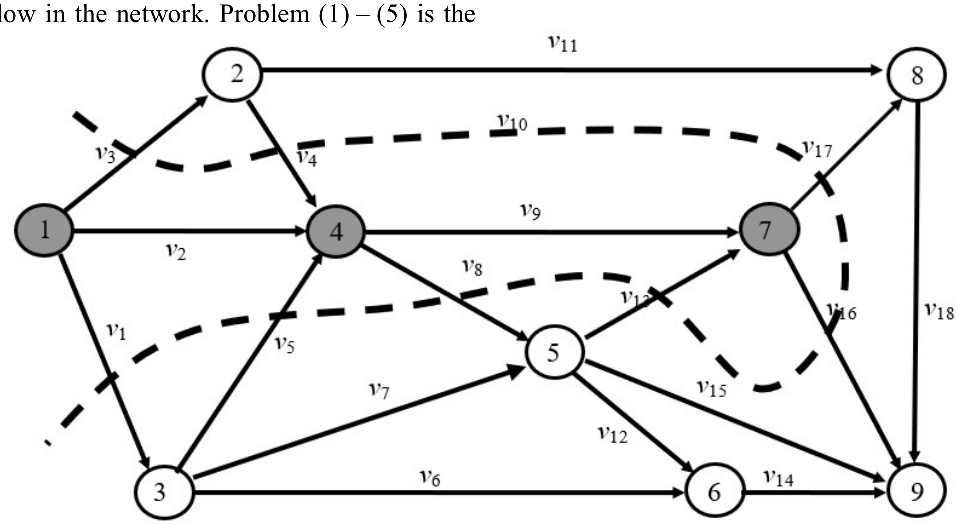

Figure 2: Search for crosscut

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:

$R+(E1,E2)=\{v \in R(E1,E2)|\ h1(v) \in E1$ *and* $h2(v) \in E2\}$,

$R-(E1,E2)=\{v \in R(E1,E2)|\ h2(v) \in E1$ *and* $h1(v) \in E2\}$.

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

$$X(E_1,E_2)=\sum_{v \in R^+(E_1,E_2)} x_v - \sum_{v \in R^-(E_1,E_2)} x_v \ .$$

Crosscut flowing capacity we will name the value

$$D(E_1,E_2)=\sum_{v \in R^+(E_1,E_2)} d_v \ .$$

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

Theorem 1. On maximum flow and minimal crosscut.

In any network the magnitude of maximum flow Q from the source h to overflow k equals the minimal flowing capacity $D(E1,E2)$ amongst all crosscuts $R(E1,E2)$, separating vertexes h and k.

Crosscut $R(\bar{E}_1, \bar{E}_2)$, with $Q=D(\bar{E}_1,\bar{E}_2)$ we will name constraining. At constraining crosscut, there is executed

$$x_v = \begin{cases} d_v, & \text{если } v \in R^+(\overline{E}_1,\overline{E}_2) \\ 0\ , & \text{если } v \in R^-(\overline{E}_1,\overline{E}_2) \end{cases}$$

Let us assume, that (Q, xv, $v \in V$) is a flow in the network, and succession $h=i0$, $v1$, $i1$, $v2$, $i2$,,vK, $iK=k$ is a circuit connecting vertexes h and k. Define on that circuit motion direction from vertex h to k. Arc vj from that circuit is called straight, if its direction coincides with motion direction from h to k, 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 h to k with value $q=min(q_1,q_2)$, where $q_1=min(dv-xv)$, minimum is taken from all straight arcs of the circuit, $q_2=min(xv)$, minimum is taken from all backward arcs of the circuit.

Theorem 2. Flow $(Q, xv, v\in V)$, 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 h to k. The search, in its turn, is based on the process of vertexes marks disposition similar to Dejkstra algorithm.

Let us add mark $P_i=[gi,vi,\theta]$ to every vertex i, where g_i – value of additional flow entered the vertex i, v_i – arc through which the flow entered, θ – sign «+», if the flow entered along the arc v_i , directed to i (along straight arc); θ – sign «–», if the flow entered along the arc v_i , 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 $P_i=[0,-,\theta]$,
- 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 $P_i=[g_i,v_i,\theta]$, where $g_i>0$,
- labelled and viewed, if the flow reached it and allowed to go further, label will have the form $P_i=[g_i,v_i,\theta]$.

Let us consider solution algorithm.

0. For all $v\in V$ assume that $xv=0$, assume that $Q=0$.

1. All vertexes are unlabeled. Vertex h is labelled, but not viewed with a label $P_h=[\infty,-,-]$. 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 $Q, xv, v\in V$ 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 $v\in V_i^-$ assume $j=h_2(v)$. If vertex j is unlabeled and $(dv-xv)>0$, then mark it with label $P_j=[q,v,+]$, where $q=min(q_i, (dv-xv))$, if $j=k$, then pass on to point 4.
- for all $v\in V_i^+$ assume $j=h_1(v)$. If vertex j is unlabeled and $xv>0$, then mark it with a label $P_j=[q,v,-]$, where $q=min(q_i, xv)$, if $j=k$, 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 $j=k$, $q=gk$ and $v=vj$.

- if θ=«+», then it is necessary to fulfill: Let us assume that $xv=xv+q$, $i=h_1(v)$, if $i=h$, 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 $xv=xv-q$, $i=h_2(v)$, if $i=h$, 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 $(Q, xv, v\in V)$. 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 $\bar{E}_1$, $\bar{E}_2=\bar{E}\setminus \bar{E}_1$. Cross cut $R(\bar{E}_1, \bar{E}_2)$ will be the sought for [19].

3. FORD AND FULKERSON ALGORITHM

Let us assume that some permissible flow has been already found. Let us ask two questions: how, having permissible flow, to define, whether it is optimal, and how to obtain permissible flow greater by value if the permissible flow thereof is not optimal.

For that purpose, it is necessary to identify, what of given below properties owns every arc in the circuit. For the first, the flow along an arc (i, j) is less than flowing capacity of an arc, (i, j), which naturally means that the flow along the arc can be increased. Let us denote multiple of such arcs in the circuit as i. For the second the flow along an arc (i, j) is positive, which means that it can be reduced. Let us denote the multiple of such arcs as R. Let us describe the procedure of Ford and Fulkerson method for labels disposition to construct the greater flow.

Step 1. Assign label to the source (vertex 1).

Step 2. Assign other labels to the vertexes and arcs proceeding from the next rules. If vertex x has a label, and vertex y has no mark and the arc $(x ,y)\epsilon I$, then label the arc (x ,y) and vertex y. In this case, the arc (x, y) is the straight direction arc. If vertex x has a label, and vertex y is unmarked and the arc (y, x) ϵR, then label the arc(y, x) and vertex y. In this case, the arc(y, x) is a backward direction one.

Step 3. Continue procedure of labels disposition until the outflow is labeled, or there are no unlabeled arcs left.

If in case of the given procedure implementation the outflow turned out to be labelled, we can say that there exists sequence of labelled arcs (name it *C*) from the source to outflow. Changing arcs flows entering *C*, we can construct the flow of greater value comparing to initial. In order to be sure, let us consider two cases: succession *C* contains only arcs of straight direction and succession *C* contains both straight and backward direction arcs.

In every case we can say how to obtain the flow of greater value comparing to the given one.

Let us consider case 1. Let $i\ (x, y)$ – a maximum value, the flow along the arc can be increased without violation of delimitation on flowing capacity. Assume that

$$k = \min_{(x,y)\in C} i(x, y)$$

Then $k>0$. In order to modify the flow upwards, let us increase values of flows on all arcs from *C* per value *k*. In this case, not a single delimitation of flowing capacity will be violated. It is easy to note, the flow preservation conditions for all vertexes will be satisfied. It follows that a new flow, on the one hand is permissible, and on the other hand, it has the value for *k* greater than the initial one.

Let us consider case 2. In this case, succession *C* contains both straight direction and backwards direction arcs. Let $r\ (x, y)$ – maximum value, the flow can be decreased along the arc (x, y). Assume that

$$k_1 = \min_{(x,y)\in C\cap R} r(x, y)$$

$$k_2 = \min_{(x,y)\in C\cap I} i(x, y)$$

Both values k_1 and k_2, and, accordingly, $min\ (k_1, k_2)>0$. In order to modify the flow upwards, let us increase flows values along all straight direction arcs from *C* for the value $min\ (k_1, k_2)$, and at all backward directions arcs from *C* decrease for the same value $min\ (k_1, k_2)$. In this case, not a single delimitation of flowing capacity will be violated. It is easy to note, the flow preservation conditions for all vertexes will be satisfied as well. Accordingly, a new flow, on the one hand, is permissible; on the other hand, it has the value less for $min\ (k_1, k_2)$ comparing to initial one.

If outflow cannot be labeled, it means that the flow is maximum. To ground this consideration, let us study the crosscut notion.

Let us select any multiple *V*, containing an outflow, but without the source. Then multiple of arcs (x, y), for which x does not belong to *V*, and $y \in V$ is called a circuit crosscut. In other words, crosscut is multiple of arcs, excluding which out of the circuit we would separate the source from the outflow. Crosscut value is the sum of flowing capacities of the arcs entering the crosscut. Crosscut is multiple of arcs removal, which brings to impossibility to pass from the source to the outflow along the remained arcs. There are several crosscuts in the circuit. Lemma 1 and lemma 2 establish connection between crosscuts and maximum flow. Lemma 1 concludes, that the value of any permissible flow from the source to the overflow is not greater than the value of any crosscut. Let us consider any crosscut, defined by multiple *V*. Assume *W* – all other circuit vertexes, not included into the multiple *V*. Let x_{ij} – value of flow for the arc (i, j), and z – overall value of the flow from the source to outflow. If to summarize conditions of flow preservation for all vertexes from the multiple *W*, then values of flows for arcs (i, j), for which vertex *i* and vertex *j* belong to the multiple *W*, will reduce, then in the result remains

$$\sum_{\substack{i\in W \\ j\in V}} x_{ij} - \sum_{\substack{i\in W \\ i\in V}} x_{ij} = z$$

Taking into account that the first sum from the given ratio is not bigger than the crosscut value, it can be concluded that Lemma 1 is true [20].

Lemma 2 lies in the fact that if the outflow cannot be labelled, then value of some crosscut equals to the flow's value. Let *V* is multiple of unmarked vertexes, and *W* is multiple of labelled vertexes. Let us consider arcs (i, j), for which $i\in W$, $j\in V$, then for them $x_{ij}=c_{ij}$ is true. It follows because, in the contrary case we could mark vertex *j* from the multiple *V* (as the arc (i, j) is the straight direction arc), which would contradict to determination of the multiple *V*.

Let us consider arcs (i, j), for which $i\in V$, a $j\in W$, then for them $x_{ij}=0$ is true. It follows because in the contrary case we could label vertex *i* from the multiple V (as the arc*(i, j)* is the backward direction arc), which would contradict to determination of the multiple V. Thus, it is seen from the ratio, that crosscut value equals to the flow value.

4. NASH EQUILIBRIUM

Nash equilibrium is the situation, upon which none of the players can increase own bending of the game, changing, on a unilateral basis, own decision. In other ways, it is the situation, at which the strategy of both players is the best reaction at opponent's actions.

Rational approach to finding the game solution supposes, that any player *i* forms an opinion on other players actions and selects as S_i own best answer. Situation S_N in the game is called Nash equilibrium, if for any player *i* and for his any strategy $s_i \in S_i$ there is fulfilled inequation $U_i(S_N) \geq U_i(S_i, S_i^*)$. Put it otherwise, S_i^* is the best reply for every player *i*. The given situation is such, that it is not beneficial for anybody to deviate from it. If others confine themselves to it.

Nash equilibrium is the main concept for solving in noncooperative case. Notion of equilibrium connects two hypotheses on players' behavior. The first – if the situation S_N is unbalanced, it cannot be considered as stable state. That is, if a player sees that deviation from S_i will bring the bigger bending game, then he/she, most likely, will deviate. It matches to rationality hypothesis. However, the player surely understands that his deviation can arouse unpredictable chain of responses from other players, final consequences of which is difficult to overestimate. Such deviation is justified only in case if there is confidence that other players keep unchanged their strategies.

The second hypothesis– if every player sees that deviations from S_i^* bring no improvement, he will maintain that strategy. Equilibrium bending of the game cannot be less than guaranteed level α_i.

Lemma 1 lies in the fact that if S_i^* - Nash equilibrium, then $U_i(S_N^*) \geq \alpha_i$ for any player- *i*.

Lemma 2 supposes that for every player there prescribed subtotals $S_i^{'} \subset S_i$. Suppose, that S_N^* - equilibrium in the game $(N, (S_i), (U_i))$, and $S_i^* \in S_i^{'}$ for any *i*. Then S_N^* is equilibrium in the game.

If $G^{'}$ is a game, obtained after iterated elimination of strongly dominated strategies, then $NE(G) \subset NE(G^{'})$. We can show that any equilibrium in the game $G^{'}$ is equilibrium in the initial game G, that is, we can record $NE(G) \subset NE(G^{'})$. The given congruence explains the sense of elimination of heavily dominated strategies. If after sequential exclusion there is one profile remained, it is in equilibrium in the initial game, but if there several profiles remained, then it is necessary to find the balanced one among them [21].

Nash theorem. Let us assume that in the game $(N, (S_i), (U_i))$ all multiples S_i are convex, and functions of bending of the game U_i are persistent and hill-shaped per variable, then there exists at least one Nash equilibrium.

5. SOFTWARE AND TOOLS

The NetBeans IDE is a free, open source, integrated development environment (IDE) that allows you to develop desktop, mobile, and web applications. The IDE supports application development in a variety of languages, including Java, HTML5, PHP, and C ++. The IDE provides integrated support for the full development cycle, from project creation with debugging, profiling, and deployment. The IDE runs on Windows, Linux, Mac OS X and Unix based systems [22].

The IDE provides comprehensive support for JDK 7 technologies and the latest Java enhancements. It is the first IDE to provide support for JDK 7, Java EE 7, and JavaFX 2. The IDE fully supports Java EE using the latest standards for Java, XML, Web Services, SQL, and fully supports GlassFish Server, a reference implementation in Java EE.

```
08.00 204 31
08.00 201 23
09.00 204 32
09.00 201 33
09.00 213 21
10.00 204 21
10.00 209 34
10.00 201 30
10.00 213 26
11.05 204 21
11.05 225 35
11.05 209 29
11.05 201 35
11.05 202 22
11.05 213 22
12.05 204 22
12.05 209 21
13.05 209 36
13.05 204 27
13.05 202 31
13.05 201 37
14.10 202 31
14.10 201 37
```

Figure 3: New txt file

Since the program will show the optimal emergency evacuation plan, it is necessary to have a building plan. The building plan of KazNRTU must be in JPEG or PNG format in order for it to appear at the beginning of the program. This floor plan shows each floor separately (1st to 10th floor).

By extracting data from the file list, we have to restore a new txt file that will contain the following information:

– lecture hall No.
– lecture (start time)
– the number of people in the hall

As you can see, in Figure 4, the chart is organized according to these three aspects and is divided by day of the week.

```
coordinates.txt
201-210,420,210,450
202-190,560,190,590
204-300,560,300,590
209-350,420,350,450
213-555,420,555,450
215-645,420,645,450
221-925,420,925,450
225-975,420,975,450
```

Figure 4: Coordinates of the lecture halls

Since the program is Java based and has the ability to draw a path from auditorium to exit, it is obvious that you need to use Canvas. In order to show the number of people in lecture halls and draw lines between objects, we are dealing with coordinates. This results in another text file with the following information:

– lecture hall No.
– coordinates along the X-axis
– coordinates along the Y axis

```
a_exit.txt   b_exit.txt   c_exit.txt
201-11.5
202-19
204-9.5
209-5.5
213-4
215-2
225-1.5
```

Figure 5: Exit priority

As you can see in Figure 5, there are 8 lecture halls on the second floor. Next to the lecture hall number are numbers, which are the X and Y coordinates, separated by commas.

In order to develop an optimal evacuation algorithm, we must have information about the "exit priorities" for each classroom, based on distance. Since we will have 4 main outputs, where there will be a stream of people, we have 4 different files that show the distances from each classroom.

The file should contain the following:

-lection (number of people),
- the distance from each lecture hall to the exits (Figure 5).

These files will be used to distribute people from the hall to the exits in the best way to avoid collisions. Based on this distance, this algorithm will calculate the number of people from each hall to the exit. This amount will vary depending on the distance to the exit.

The final product should contain an image of the floor plan and paths for each exit. Which means, if we are working in Java, our Canvas has 2 layers:

- JPEG images (building plan),
-created path with label.

In order to show a picture of the KazNITU plan, we must add it to our Canvas. We need to create a GraphicsContext object, images (Figure 6).

```
//************************************************
    Image image = new Image("guk.jpg");
    ImageView iv2 = new ImageView();
     iv2.setImage(image);
     iv2.setFitWidth(1200);
     iv2.setFitHeight(800);
     iv2.setPreserveRatio(true);
     iv2.setSmooth(true);
     iv2.setCache(true);
```

Figure 6: Creating an image object

Then, setting its width, height and other parameters in imageView, we add it to the Canvas. In order to create a Canvas, we must create an object of the Canvas class and give it a specific width and height. Then we add this Canvas to create the object's graphics context (Figure 7).

```
Scene s = new Scene(root, 1300, 800, Color.BLACK);
final Canvas canvas = new Canvas(1300, 800);
gc = canvas.getGraphicsContext2D();
        gc.clearRect(0, 0, 1300, 800);
gc.setFill(Color.GREEN);
```

Figure 7: Creating a Canvas object

The next step is to get the current date, including the time, in order to get the right number of people in the classroom. To do this, we have to use a class called "Calendar" and use simple methods that can select the time, including hour, minute, etc. (Figure 8).

```
Calendar calendar = Calendar.getInstance();
int day = calendar.get(Calendar.DAY_OF_WEEK);
if(calendar.getTime().getMinutes()<10){
time = calendar.getTime().getHours()+".0"+calendar.getTime().getMinutes();
}else{
time = calendar.getTime().getHours()+"."+calendar.getTime().getMinutes();
}

if(calendar.getTime().getHours()<10){
time = "0"+calendar.getTime().getHours()+"."+calendar.getTime().getMinutes();
}else{
time = calendar.getTime().getHours()+"."+calendar.getTime().getMinutes();
}
```

Figure 8: Current date using Calendar

Based on the current date and time, we will extract information from the files. If the day of the week is 2, we will use the monday.txt file and so on (Figure 9).

```
if(day==2){
    path = "/Users/demo/Desktop/monday.txt";
    System.out.println(path);
} if(day==3){
    path = "/Users/demo/Desktop/tuesday.txt";
    System.out.println(path);
} if(day==4){
    path = "/Users/demo/Desktop/wednesday.txt";
    System.out.println(path);
} if(day==5){
    path = "/Users/demo/Desktop/thursday.txt";
    System.out.println(path);
} if(day==6){
    path = "/Users/demo/Desktop/friday.txt";
    System.out.println(path);
} if(day==7){
    path = "/Users/demo/Desktop/saturday.txt";
    System.out.println(path);
}
```

Figure 9: Determining the target file

```
for(int i=0;i<dist.size();i++){
    double perc = (Double.parseDouble(dist.get(i).split("-")[1])/
            sum_of_percentage)*60;
    System.out.println("Percentage of "+perc);
    double origin = Double.parseDouble(temp_arrays.get(i).substring(7,
            temp_arrays.get(i).length()));
    System.out.println(origin+"----------"+perc);
```

Figure 10: Calculation of percentage

The next step is to subtract the percentage that we calculated earlier from the original amount. If the result is a negative number, it means that some people will be drawn from the next lecture hall. If the result is a positive number, it means that some people stay in this lecture, they will be directed to a different exit (Figure 11).

```
int subtr = (int)(origin - Math.round(perc))+negative_number;
int temp=0;
 if(subtr==0){
    temp = (int)(Math.round(perc));
}
if(subtr<0){
    negative_number = subtr;
    System.out.println("It is negative!!");
    System.out.println("negative number:"+negative_number);
    temp = (int)origin;
    subtr=0;

}
if(subtr>0){
    System.out.println("It is positive!!");
    System.out.println("negative number:"+negative_number);
  temp = (int)(Math.round(perc))-negative_number;
  negative_number=0;
}
```

Figure 11: Working with unnecessary people

CONCLUSION

A mathematical model of the optimal flow distribution over the grindshill network with an analysis of the formation of the flow and characteristics of the ways of people advancement has been developed. A model and an algorithm for the optimal coverage of building areas have been developed.

Offered algorithm for solving the problem of maximum flow in the network is based on searching an increasing flow in the circuit from *h* to *k*. The search, in its turn, is based on the process of vertexes marks disposition similar to Dejkstra algorithm.

Nash equilibrium is the main concept for solving in no cooperative case. Notion of equilibrium connects two hypotheses on players' behavior. The first – if the situation S_N S_N is unbalanced, it cannot be considered as stable state. That is, if a player sees that deviation from S_i will bring the bigger bending game, then he/she, most likely, will deviate. It matches to rationality hypothesis. However, the player surely understands that his deviation can arouse unpredictable chain of responses from other players, final consequences of

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

which is difficult to overestimate. Such deviation is justified only in case if there is confidence that other players keep unchanged their strategies.

In the work herein we studied models of flows distribution along the circuit using game-theoretical approach. We executed following tasks:

- given descriptive setting of the problem, with consideration of emergencies classification, norms, stages and principles of evacuation rating;
- given mathematical setting of the problem, with consideration of problem solving algorithm, task on maximum flow, method of potentials and criterion of optimality, Ford-Fulkerson algorithm, second Kirchhoff convention, Nash equilibrium and contour interrelationship;
-fulfilled search of permissible solutions of the task based on the problem on maximum flow;
-search of minimal evacuation time based on game-theoretical approach to people's flow motion modeling;
-search of the shortest path using the algorithm of finding balanced state in describing of people's flow motion model.

The programming language Java was used as a software implementation of this algorithm

Work completed on post-doctoral programs Almaty Technological University under the project "Research and development of mathematical and information models, as well as software and technical implementation of the evacuation problem". Experimental researches have been carried in the frame of the project № AP09259547 IICT CS MEC RK.

REFERENCES:

[1] D. Hartama, A. P. Windarto, A. Wanto Evacuation Planning for Disaster Management by Using The Relaxation Based Algorithm and Route Choice Model, International Journal Of Information System & Technology Vol. 2, No. 1, (2018), pp. 7-13, DOI:10.30645/ijistech.v2i1.14.
[2] H. W. Hamacher, S. A. Tjandra Mathematical Modeling of Evacuation Problems: A State of The Art Bericht, Elektronische Publikation.
[3] S. V. Malodushev, A.A. Rogov, R. V. Voronov Mathematical model for evacuation people from corridor-type buildings Vestnik of Saint Petersburg University Applied Mathematics Computer Science Control Processes 15(3):375-384,2019, DOI:10.21638/11702/spbu10.2019.307.
[4] Chi To Ng, T. C. E. Cheng, Eugene Levner, Boris Kriheli Optimal bi-criterion planning of rescue and evacuation operations for marine accidents using an iterative scheduling algorithm Annals of Operations Research 296 (1-2), January 2021, DOI:10.1007/s10479-020-03632-6.
[5] C. Schiopu Maximum flows in bipartite dynamic networks Bulletin of the Transilvania University of Bra¸sov • Vol 12(61), No. 1 - 2019 Series III: Mathematics, Informatics, Physics, 177-198 DOI:10.31926/but.mif.2019.12.61.1.14.
[6] J. Abusalama, Sa. Razali, Y.-H. Choo, L. Momani, A. Alkharabsheh Dynamic real-time capac.ity constrained routing algorithm for evacuation planning problem Indonesian Journal of Electrical Engineering and Computer Science Vol. 20, No. 3, 2020, pp. 1388~1396, DOI: 10.11591/ijeecs.v20.i3.pp1388-1396.
[7] U. Pyakurel, H. Nath, S Dempe and T. N. Dhamala Efficient Dynamic Flow Algorithms for Evacuation Planning Problems with Partial Lane Reversal, Mathematics 2019, 7(10), 993; https://doi.org/10.3390/math7100993.
[8] J. Feng and Q. Wang Emergency safety evacuation decision based on dynamic Gaussian Bayesian network IOP Conference Series: Materials Science and Engineering,Volume 688,Issue 5 Citation unning Feng and Qifeng Wang 2019 IOP Conf. Ser.: Mater. Sci. Eng. 688 055076.
[9] Belyaev S.V. Evacuation of buildings of mass use. Moscow: Publishing House of the All-Union Academy of Architecture, 1938. - 257s.
[10] Smirnova A.T. Fundamentals of life safety - AST Ltd, M., 1999 p.104-107.
[11] Kholshchevnikov V.V., Samoshin D.A., Isaevich I.I. Natural observations of human flows: textbook. allowance. Moscow: Academy of State Fire Service of the Ministry of Emergency Situations of Russia, 2009 .-- 191 p.
[12] Kholshchevnikov VV Human flows in buildings, structures and on the territory of their complexes. Abstract of a dissertation for the degree of Doctor of Technical Sciences, Moscow, MISS 1983.
[13] Amirgaliyev Ye., Kovalenko A., Kalizhanova AU, Kozbakova A.Kh Modeling of Networks Flows of Grinshilds Types, Joint issue based on the materials of the International Scientific Conference "Computational and Information Technologies in Science, Engineering and Education" Computational Technologies:

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

Bulletin of KazNU named after ... Al-farabi, 2015. - Volume XX, Part II, No. 3 (86). - P. 184-190.

[14] Kovalenko A.G., Vlasova I.A., Borisova S.P. Games theory and operations research. Samara: Publishing house «Samarsky University», 2006. – 147p.

[15] Volkov I.K., Zagoruiko Ye.A. Operations research. Moscow: Publishing house of MSTU named after Bauman N.E., 2000. – 435p.

[16] Vasin A.A., Morozov V.V. Games theory and econometric models. Moscow: MAKC Press, 2005. – 237p.

[17] Gorlach B.A. Operations research. S-Petersburg: Publishing house «Lanj», 2013. – 448p.

[18] Kosorukov O.A., Mischenko A.V. Operations research. Moscow: Publishing house «Ekzamen», 2003. – 448p.

[19] Belyayev S.V. Evacuation of mass public buildings. Moscow: Publishing house of «All-Union Academy of architecture », 1938. – 257p.

[20] Germeiyer Yu.B. Introduction into the theory of operations research. Moscow.: Nauka, 1971. – 358p.

[21] Neiman J., Morgenshtern O. Games theory and economic behavior. Moscow.: Nauka, 1970. – 244p.

[22] Kuznetsov A.I., Sakovich V.A., Kholod N.I. Mathematical programming. Minsk: Higher school, 2001. – 352p.