

DOI: 10.31432/1994-2443-2023-18-2-33-47

Development the Information Systems With Fuzzy Logic Algorithms and Network Optimization

Sinitsyn Alexander Vladimirovitch

Ph.D. (Math.), Head of Department^a,
Docent of the Department of Information Processes and Systems^b,
e-mail: a@sinitsyn.info, web-site: <https://alexander.sinitsyn.info>

Lisay Nikolay Yurievitch

Cand. Sci. (Engineering Sciences), Deputy Director for International Projects^c,
e-mail: lisainik@rambler.ru

Selivanov Sergey Alexandrovich

Cand. Sci. (Engineering Sciences), Deputy Director for Information Technology^a,
Docent of the Department of Information Processes and Systems^b,
e-mail: selivanov@inevm.ru

Sinitsyn Artemy Alexandrovitch

M.D. Science, engineer-developer^a, postgraduate student^d,
e-mail: art@sinitsyn.info, web-site: <https://art.sinitsyn.info>

^a International Center for Informatics and Electronics (ICIE),
17-1, Presnensky Val, 123557, Moscow, Russia

^b Federal State Budget Educational Institution of Higher Education «MIREA — Russian Technological University» (RTU MIREA), 86 Vernadsky Avenue, 119571, Moscow, Russia

^c National Center «Kurchatov Institute» — Institute for System Research, Ploshchad' Akademika Kurchatova, 1, 123098, Moscow, Russia

^d Lomonosov Moscow State University, 1, Kolkogorov str., 119234, Moscow, Russia

Abstract. The article is devoted to the problems of construction information systems, consisting of typical parts. The first one is a system for collecting and analyzing geolocation data with fuzzy logic in disasters. The second one — a network to be optimized in terms of total traffic transmission capacity by increasing the power of network equipment with restrictions on their number.

Keywords: fuzzy logic, system prototype, client-server architecture, switching matrix, evolutionary algorithm, method for finding the maximum traffic, graph theory, disaster relief

Citation: Sinitsyn Alexander V., Lisay Nikolay Yu., Selivanov Sergey A. Sinitsyn Artemy A. Development the information systems with fuzzy logic algorithms and network optimization // Information and Innovations 2023, T.18, №2. p. 33-47. DOI: 10.31432/1994-2443-2023-18-2-33-47

Разработка информационных систем с алгоритмами нечеткой логики и оптимизация сети

Синицын Александр Владимирович

кандидат-физико-математических наук, начальник отдела^а,
доцент кафедры Института информационных технологий^б
e-mail: a@sinitsyn.info, web-site: <https://alexander.sinitsyn.info/>

Лисай Николай Юрьевич

кандидат технических наук, заместитель директора по международным проектам^с,
e-mail: lisainik@rambler.ru

Селиванов Сергей Александрович

кандидат технических наук, заместитель директора по информационным технологиям^а, доцент кафедры Института информационных технологий^б,
e-mail: selivanov@inevm.ru

Синицын Артемий Александрович

магистр науки о данных, инженер-разработчик^а, аспирант^д,
e-mail: art@sinitsyn.info, web-site: <https://art.sinitsyn.info/>

^а Международный центр по информатике и электронике (ИнтерЭВМ),
Россия, 123557, Москва, Пресненский Вал, дом 17, стр. 1

^б Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА – Российский технологический университет»,
Россия, 119571, Москва, пр-т Вернадского, 86 с2

^с НИЦ «Курчатовский Институт» — НИИ Системных Исследований,
Россия, 123098, Москва, пл. Академика Курчатова 1

^д Московский государственный университет им. М.В. Ломоносова,
Россия, 119234, Москва, ул. Колмогорова 1

Аннотация. Статья посвящена проблемам построения информационных систем, состоящих из типовых частей, в качестве одной рассматривается система сбора и анализа данных о местоположении с нечеткой логикой при бедствиях, в качестве другой — система передачи данных, подлежащая оптимизации по пропускной способности передачи трафика за счет увеличения мощности сетевого оборудования при ограничениях на их количество.

Ключевые слова: нечеткая логика, искусственный интеллект, прототип системы, клиент-серверная архитектура, коммутационная матрица, эволюционный алгоритм, метод нахождения максимального трафика, теория графов, помощь при бедствии

Цитирование публикации: Синицын А.В., Лисай Н.Ю., Селиванов С.А., Синицын А.А. Разработка информационных систем с алгоритмами нечеткой логики и оптимизация сети // Информация и инновации. 2023, Т.18, № 2. с. 33-47. DOI: 10.31432/1994-2443-2023-18-2-33-47



1. Introduction

Typical information systems, including, firstly, a system for collecting information, which is sometimes incorrect, so it could be considered fuzzy logic algorithms, and secondly, a system for transmitting information from multiple clients (gadgets) to central servers, which requires optimization communication network to ensure maximum power at the lowest cost of its operation.

This article consider typical difficulties for constructing each of these parts. For easy understanding, the authors restricted by a brief summary of the theory, giving specific examples of really working systems that are useful to developers. The description of these developments are published in [1,2] and [3-5], where is shown both new algorithms with fuzzy logic and well-known algorithms for network optimization as searching for maximum traffic paths and genetic algorithms.

The main significance of this work is not only to combine the previous results, but also presentation of really working prototypes developed jointly with bachelors and masters of the Federal State Budget Educational Institution of Higher Education "MIREA — Russian Technological University" (RTU MIREA) [4].

2. Fuzzy logic in disaster relief system

The target of the first part is construction a system for collecting and analyzing fuzzy data to increase the efficiency of providing medical and humanitarian relief to victims of disasters. "Disasters" are natural or man-made disasters and other events. The system must solve the following tasks:

- collection, transmission and processing of information about the victims to provide relief;
- ensuring the supply of humanitarian and medical relief in the required volume and goods nomenclature;

- organization of applications and their control in an interactive mode;
- management of rehabilitation routes.

Further development of the functionality of the system prototype will be the basis for the creation of a global information system that organizes the storage, processing of special data to solve disaster relief problems that cannot be overcome without the involvement of modern techniques and new algorithms.

It is assumed that information is collected on the client side using drones or a mobile gadgets of a volunteer or victims, and then it is processed on the server side.

2.1 Fuzzy logic algorithm

The fuzziness of data collection arises from the difficulty of data interaction when counting, when the same data or part of the data can be duplicated by different messages. For example, three drones collect information about objects by exploring overlapping areas of the earth's surface, then, as shown in Fig. 1, the central object located in the intersection of the research areas of all three drone will be reported three times, each objects are reported twice from the intersection of the research areas of every two drones and so on.

Let's denote the "fuzzy" number of founded objects as N , the research area of each message as circles C , where D is the distance between the centers.

In the "fuzzy" formulation of the problem, linguistic variables were introduced for the radius and quantity $\{R_i, N_i\}$.

The problem is to develop an algorithm in standard logic of with fuzzy logic problems [6]:

- 1) determination of a fuzzy value $>$ reduction to fuzziness (fuzzification);
- 2) fuzzy logical inference (according to the rule base);

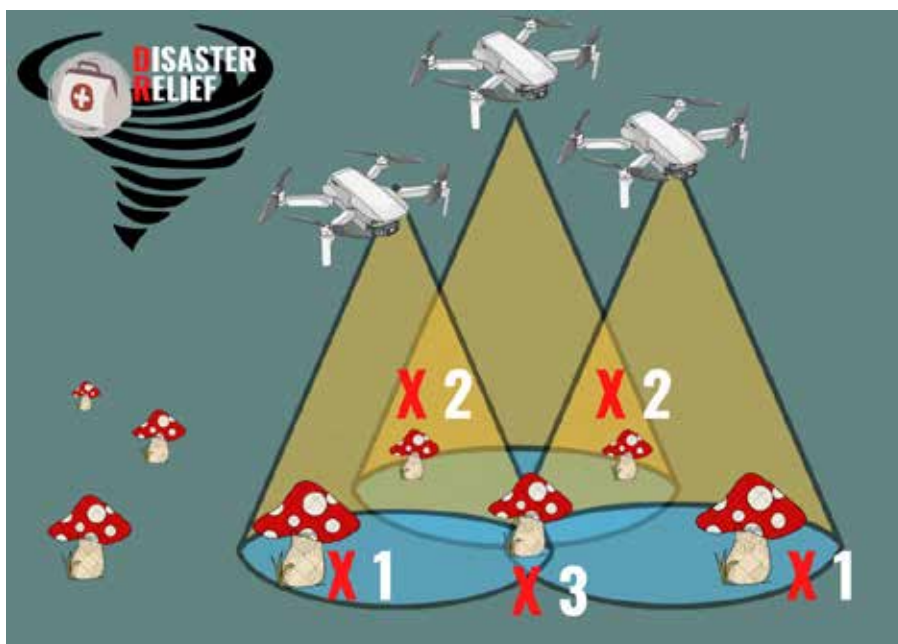


Fig. 1. Fuzzy geolocation data collection

Source: developed by the authors

3) reduction to clarity > clear value (defuzzification).

Such an algorithm has been developed. It consists of the following steps, involving pairwise reduction of the quantity in the research areas:

- 1) input N_1, N_2, \dots, N_n ;
- 2) fuzzy sum calculating $N_0 = \sum_{i=1}^n N_i$;
- 3) geolocation calculation of distances between areas centers D_{ij} ;
- 4) calculation of the matrix (c_{ij}) of pairwise intersections of circles;
- 5) loop start on i and j : pairwise comparison and quantity reduction for measurements N_i and N_j according to the following basic rules:
 - 6) computing the counts of a pair of i and j regions in an intersection $N_i(2) = c_{ij} * \min(N_i, N_j)$, $N_j(2) = c_{ij} * \min(N_i, N_j)$;
 - 7) calculation of quantities outside the intersection $N_i(1) = N_i - N_j(2)$, $N_j(1) = N_j - N_i(2)$;
 - 8) calculation of new "clear" values, taking into account "fuzziness" $N_i = 1/1 * N_i(1) + 1/2 * N_i(2)$, $N_j = 1/1 * N_j(1) + 1/2 * N_j(2)$;

9) loop end;

10) calculating the total amount $N = \sum_{i=1}^n N_i$ and reduction percent $100 * (1 - N/N_0)$.

A numerical experiment of a fuzzy logic algorithm (which was written by Python, the code of which is freely available at <https://github.com/asinitsyninfo/Fuzzy-Disaster-Relief.git>) gave the results shown in Fig. 2. The digits in the center of each research area indicate the collected a "fuzzy" value and a "clear" value found from the rule base. The values could be reduced from 6% to 70% depending of the increase in overlapping research areas.

2.2 Client part of the system prototype

Fig. 3 shows a general description of the client side of the system. In the disaster area, a volunteer or a victim contacts the system data center using a drone, mobile application or web form (Fig. 3-1) to transmit geolocation (Fig. 3-2). Then the application offers to select the type of disaster: tsunami, fires, mudflows, nuclear power plant accidents,

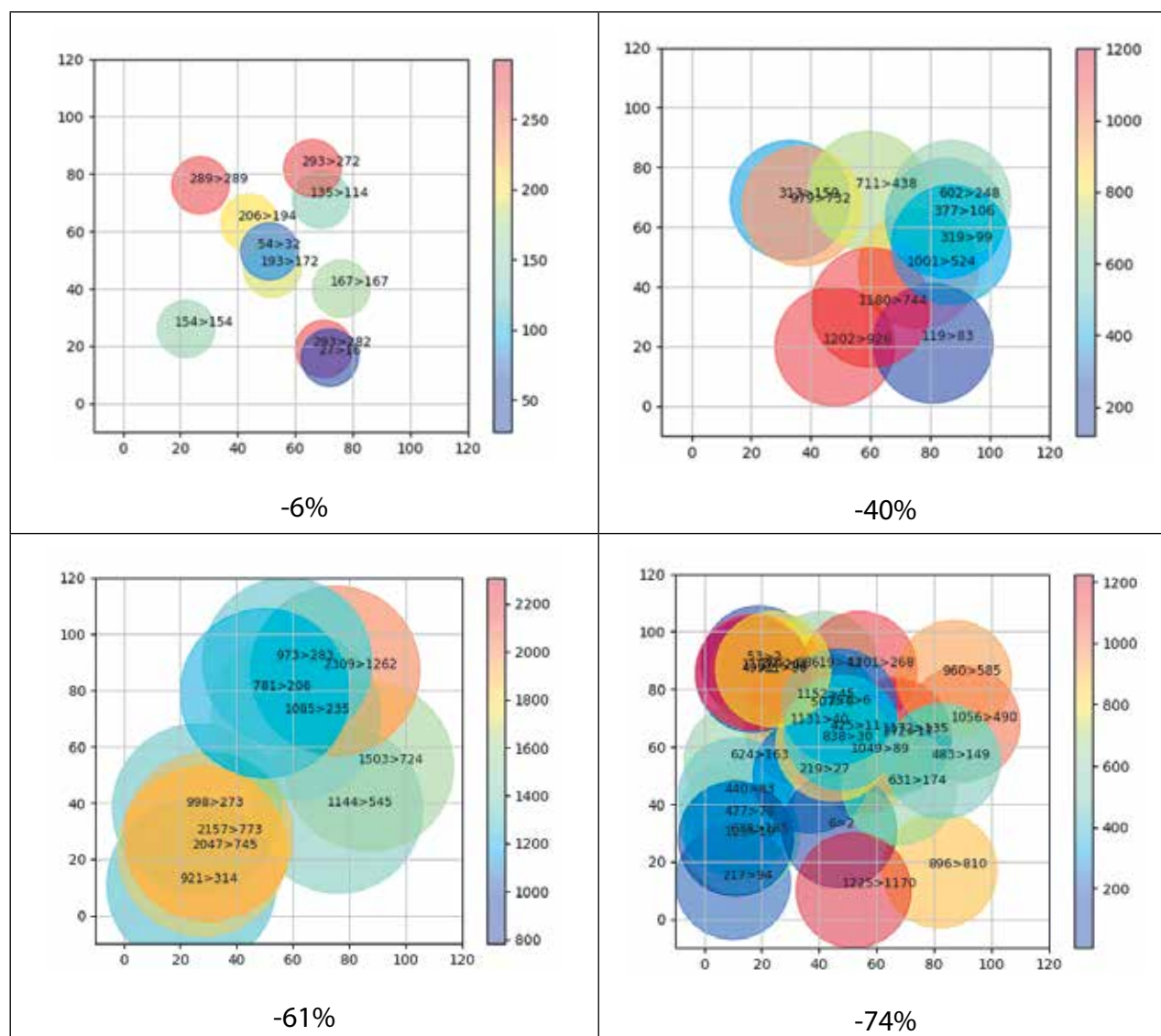


Fig. 2. The fuzzy and clear total values difference for different cases of overlapping research areas and size of radius

Source: developed by the authors

earthquakes, floods, hurricanes, volcanoes (Fig. 3-3), estimate the number of victims in the visible radius of meters (Fig. 3-4), which can be detailed by types of damage to health (Fig. 3-5). The mobile device's IMEI or IP and GPS data are transmitted to a central server to calculate the number of victims and the need for relief.

Note that the definition of the visibility radius and the definition of details is highly

dependent on the terrain, for example, in a city it could be 10-20m, in a forest — up to 50m, in a field — several hundred meters, and in the sea — up to several kilometers.

The client part is a fast and accessible web application for a smartphone browser or any other gadget. To collect information about the victims, only a simple mobile phone with Internet access and GPS is needed to report a natural disaster.



Fig. 3. The client side of the prototype to disaster relief system

Source: developed by the authors

The MVVM pattern — Model-View-ViewModel (Fig. 4) was chosen as the architecture for developing the client part, which clearly delimits the areas of responsibility between the layers and consists of the following parts:

- the model contains the functional business logic of the application;

- view is used to display the data received from the model;

A view-model is an abstraction of a view and a wrapper around the data from the model. HTML and CSS technologies, as well as the Bootstrap framework, were used to develop the user interface, and Javascript was used to make requests to the server side.

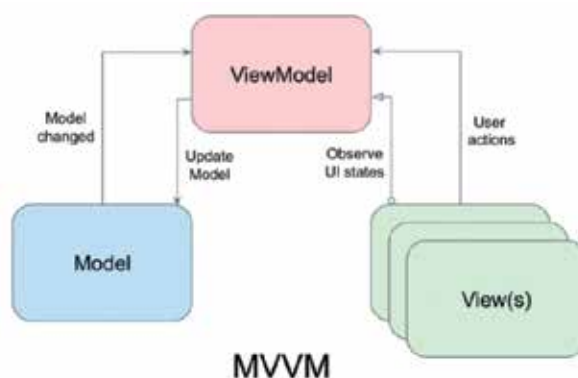


Fig. 4. MVVM pattern schema

Source: developed by the authors

2.3 Server part of the system prototype

The server receives data from numerous sources (gadgets), processes the geolocation and the number of victims and their needs (Fig. 2-1), based on the fuzzy logic algorithm [1], makes a disaster map

(Fig. 2-2), which can be built using the OpenStreetMaps API (or Google Maps, or Yandex Maps etc.), for reporting in PDF, CSV, XML formats (Fig. 2-3) for use (Fig. 2-4) by affiliated organizations organizing disaster relief (Fig. 2-5).

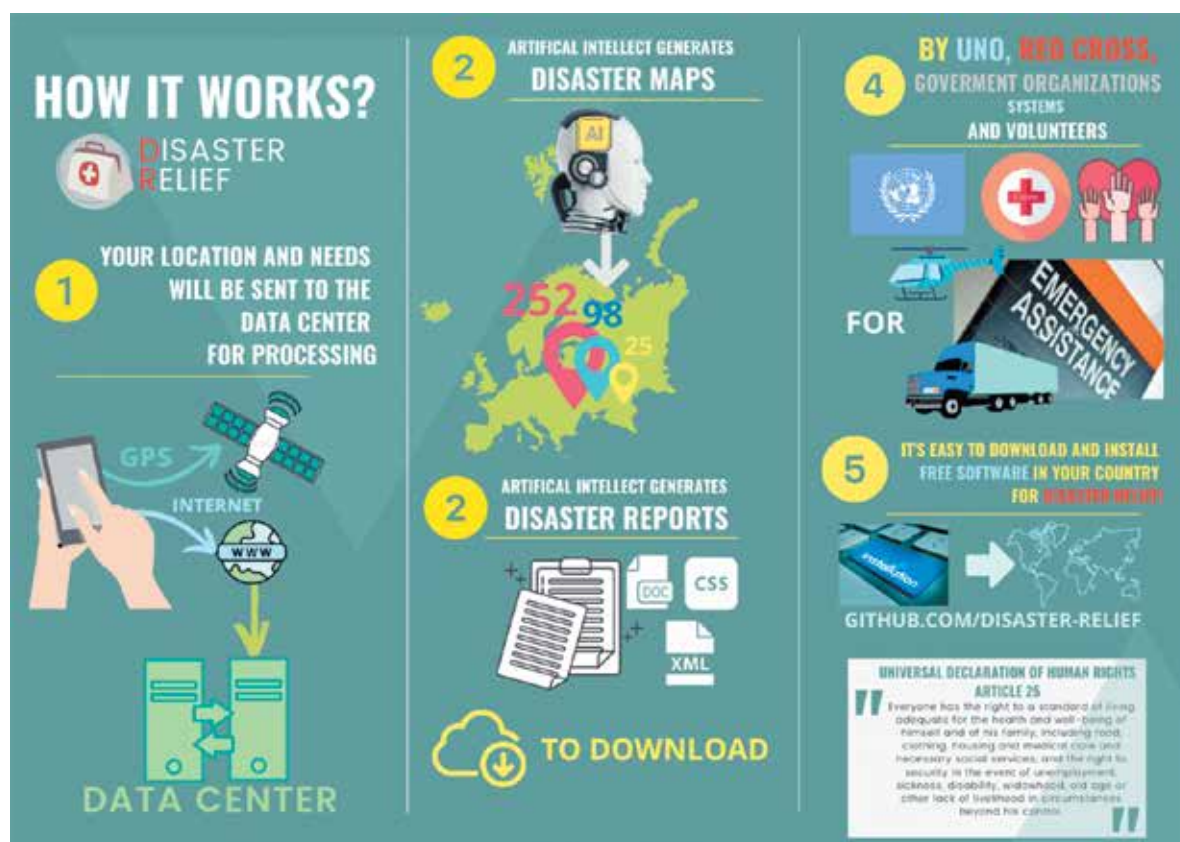


Fig. 5. The server side of the prototype to disaster relief system

Source: developed by the authors

The development of the server part follows the principles of the microservice architecture (Fig. 6):

Data sent to the system from the disaster area passes through the Gateway — a proxy server that provides access to data through a single software interface.

Gateway receives requests from the client and forwards them to microservices. In total, there are 4 microservices in the server part:

- the emergency message processing module receives information about disasters that have occurred, saves them in the

database and, if necessary, initiates the process of calculating victims needs. It also provides access to the history of emergency situations, allows you to get information about the actions taken;

- the victim needs calculation module determines the need for assistance depending on the number of victims. The module also makes use of emergency regulations and contacts of volunteers and aid organizations;

- analysis and statistics module generates reports on current and past emergencies in PDF, CSV, XML formats;

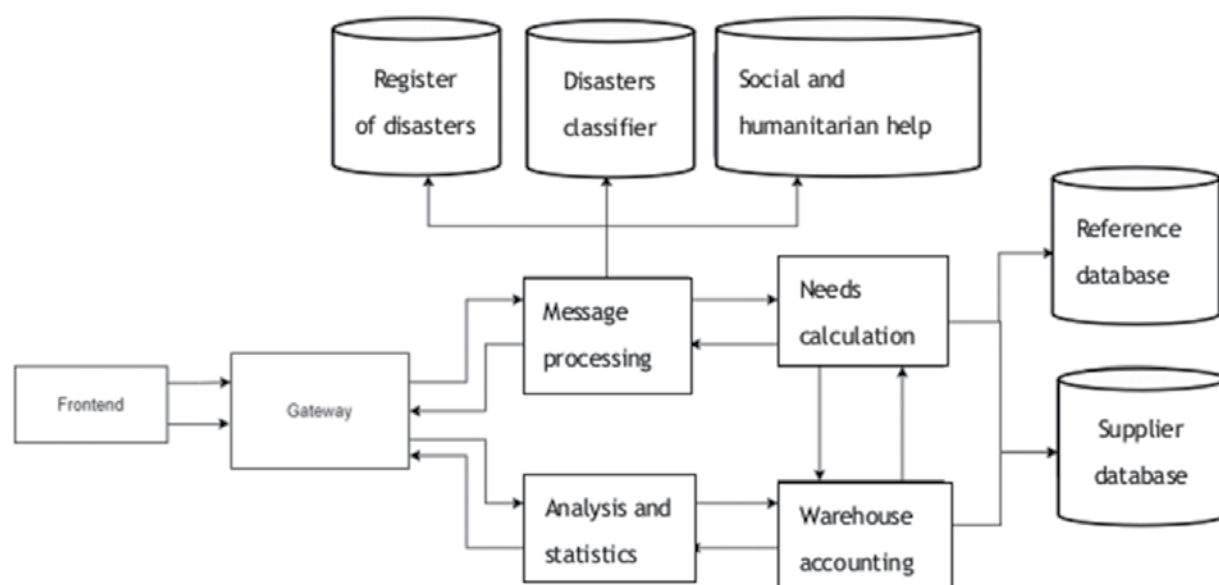


Fig. 6. Block diagram of the server part of the system

Source: developed by the authors

- the warehouse accounting module stores and analyzes information on the amount of resources in the warehouses of various organizations for disaster relief.

To develop the server part of the system, the Go programming language was chosen with high performance and a wide range of tools for creating highly loaded web applications.

The onion architecture in the standard backend application layout was chosen as the architecture for developing microservices, dividing each microservice as follows:

- Layer of business entities represented by classes, their fields and methods for interacting with themselves;
- Layer with the business logic of the application;
- Data Access Object layer for exchange with databases;
- Layer of interfaces.

Third-party open source libraries and third-party service APIs were used for development:

- go-mssqldb to work with the MS SQL Server database management system;
- go-elasticsearch to work with Elasticsearch databases;
- go-minio to work with the MinIO object store.

- Library GoFPDF is used to generate PDF disaster reports.

OpenRoute Service API for determining the area of a disaster area and building detour routes.

To determine the disaster area and build a route around it, the application exchanges a request with the OpenRoute service in JSON format.

The received response is used by the server to create a map showing the disaster area. Fig. 7.1 shows an example of a map generated by the server.

When hovering the mouse over the cursor (Fig. 7.2), the distance of a specific point from the epicenter of the disaster area and the address are shown, when hovering over the area, the "fuzzy" and "clear" number of geographic coordinates affected by it are shown.

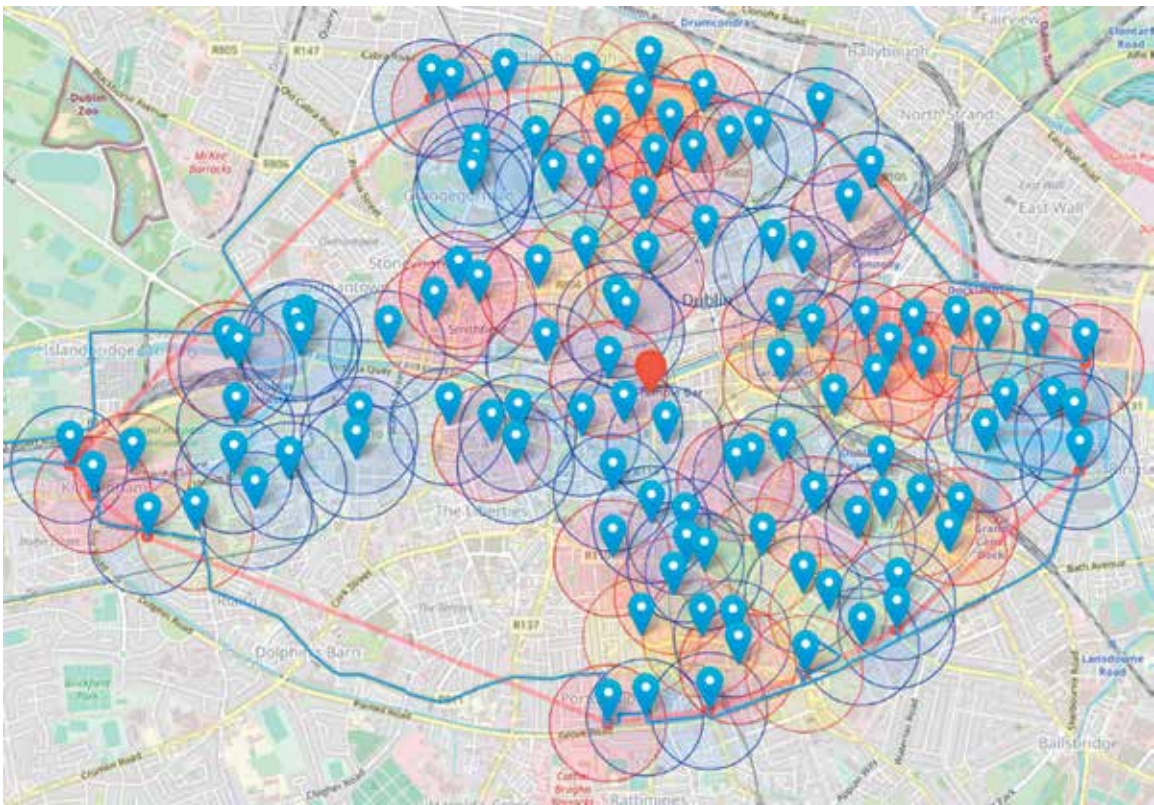


Fig. 7.1. Disaster area on the map with its convex shell (red line) and enveloping logistic route (blue line)
Source: developed by the authors



Fig. 7.2. Details of the disaster area on the map
Source: developed by the authors

The development and implementation of such a system on an international scale will provide to organize effective interaction

and coordination of work at a new level for any geographical location and scale of the emergency.

2.4 Outline of the global disaster relief information system

The prototype presented in the previous paragraphs is the basis for the development

of a global level system, which scheme is shown in Fig. 8.

The scheme has the following links between the numbered blocks:

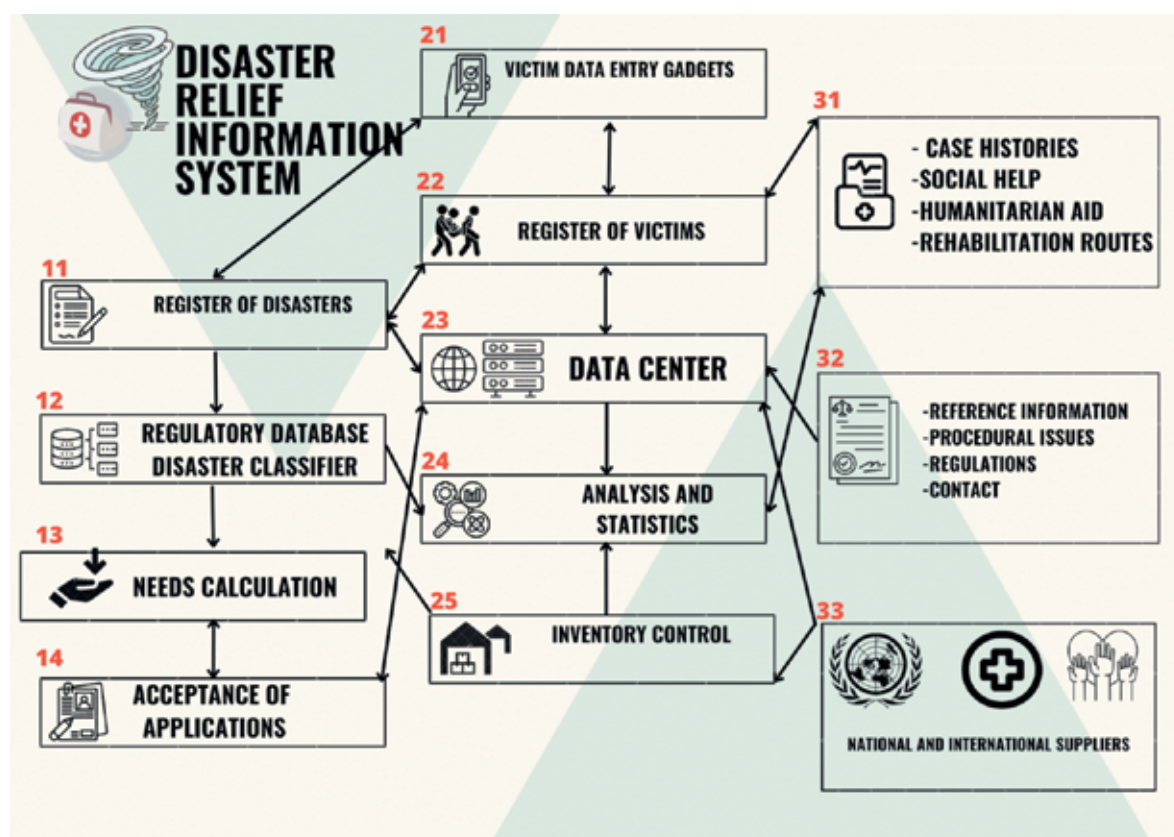


Fig. 8. General scheme of the global information system for disaster relief

Source: developed by the authors

- 21-11 — data of the research area;
- 11-23 and 22-23 — data on the disaster and victims;
- 11-22 — clarification of the scale of the disaster and the number of victims;
- 33-34 — supply tickets and proposals;
- 33-25 — relief;
- 25-13 — data on actual deliveries.

The proposed global information system for the management of medical and social relief for victims of disasters has no analogues so far and will significantly increase the effectiveness of measures aimed at pro-

viding relief to victims and eliminating the consequences of emergencies.

Improving the effectiveness of humanitarian and medical relief for victims of disasters is possible using a global information system built on the basis of such recent advances in information technology, as presented in approach of [7-8].

The result of the article is a working prototype of an information system for relief to victims in emergency situations, the development and construction of the following objects:

- client-server system;

- interaction with normative documents;

- an integrated database.

Such a system allows:

- to estimate the number of victims using fuzzy logic;

- to estimate the geography of the disaster;

- get basic reporting.

The authors express their hope for further development of their developments at new levels.

3. Network calculus traffic optimization

The purpose of the second part of the article is to optimize the capacity of the communication network with minimal equipment costs, which is important for highly loaded applications. Such optimization is necessary for the functioning of any client-server systems, such as described above. The communication network is represented by a graph, network nodes — by vertices, channels — by edges. All channels and nodes are equal in priority, and traffic between nodes is regular and uniform. The control consists in changing the capacity of a certain edge. To increase the throughput of the physical network, the network equipment is replaced with a more powerful one (Fig. 9).

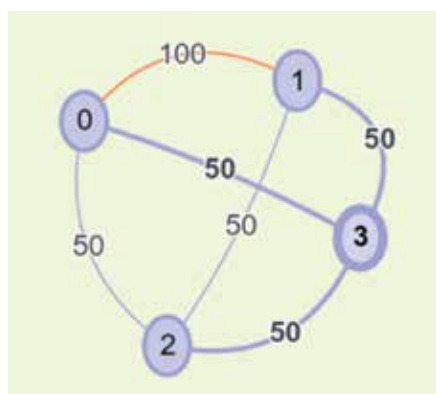


Fig. 9. Network representation as a graph with one optimized edge (red)

Source: developed by the authors

3.1 Functional definition

Let us define the functional as the total throughput, according to the previously unknown paths of maximum traffic between the nodes.

To determine the maximum traffic $T(i, j)$ between two i and j nodes, we use well-known Ford-Fulkerson algorithm [9] (or the Edmonds-Karp algorithm [9], or the Dinitz algorithm [10]).

We will be interested in the total bandwidth of the network represented by the graph, namely the sum of all $T(i, j)$:

$$T = \frac{1}{m} \sum_{i=1}^n \sum_{j=i}^n T(S(i, j)) \rightarrow \max, \quad (1)$$

calculated for $T(i, j) = T(S(i, j))$ along corresponding path $S_{ij} = S(i, j)$ for nodes determined by Ford-Fulkerson algorithm.

3.2. Formulation of the problem

The communication network is represented by an undirected connected planar graph:

$$G = G(N, E, C), \quad (2)$$

where $N \in (N_1, \dots, N_n)$ is the set of graph vertices or communication nodes (Node),

$E \in (E_1, \dots, E_m)$ — is the set of graph edges or communication channels (Edge) и $C \in (C_1, \dots, C_l)$ — and is the set of network capacities (Capacity).

Power constraints are represented by the inequality

$$C = \sum_{i=1}^l C_i \leq C_0 \rightarrow \min, \quad (3)$$

where the functional C depends on the total cost of the network equipment for the physical network. Inequality (3) can be considered as cost constraints by given constant C_0 .

Finally, the classical optimization problem is to find on the graph (2) optimal dis-

tribution of edges with increased capacity to maximize (1), while minimizing (3).

3.3. General scheme of the method

The optimal distribution of network capacities (3) is found by a combination of the genetic algorithm [11] and the Ford-Fulkerson Algorithm [9] for the graph (2). The general scheme consists of steps [4-5]:

- The initial representation of the graph;
- Initialization of graph nodes by placing equipment with minimal performance characteristics;

- Determining the amount of resources in the maximum characteristics to be placed on the column (2);
- Application of an adapted genetic algorithm;
- Calculation of the functional (1) by the Ford-Fulkerson method;
- Obtaining the final optimal distribution of resources on the graph (2).

This method was implemented in [4] and [5] using the C++ and Python programming languages, respectively. The tools shown in Fig. 10 allowed the development of a public service for network optimization (Fig. 10).

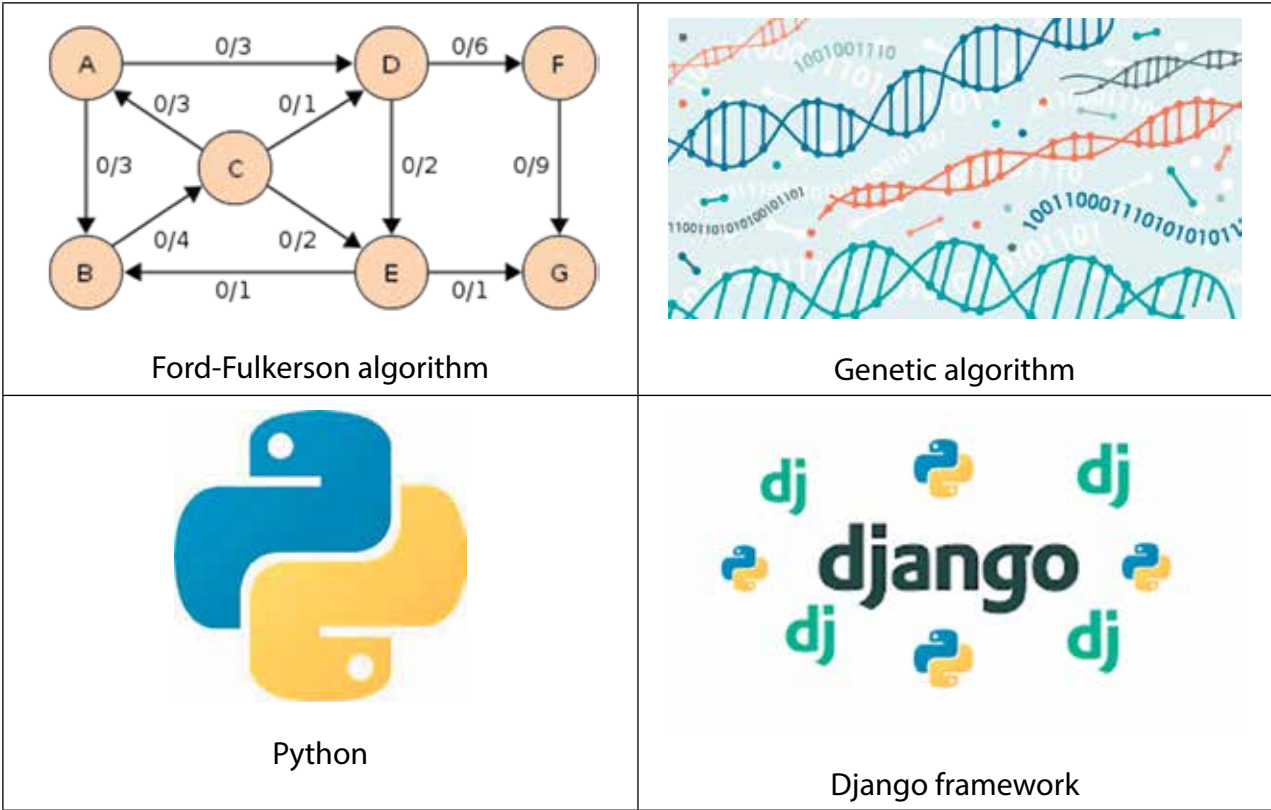


Fig. 10. Development tools for Web Optimizer of Network Capacity
Source: developed by the authors

The public service provides the input of an arbitrary graph configuration up to 12x12, the input of a new capacity for edges, the input of the number of edges to be optimized, and the generation of the optimiza-

tion result. On Fig. 11 shows an interesting case when the algorithm independently generates a “ring” structure, often used in building communication networks for complete connectivity and fault tolerance.

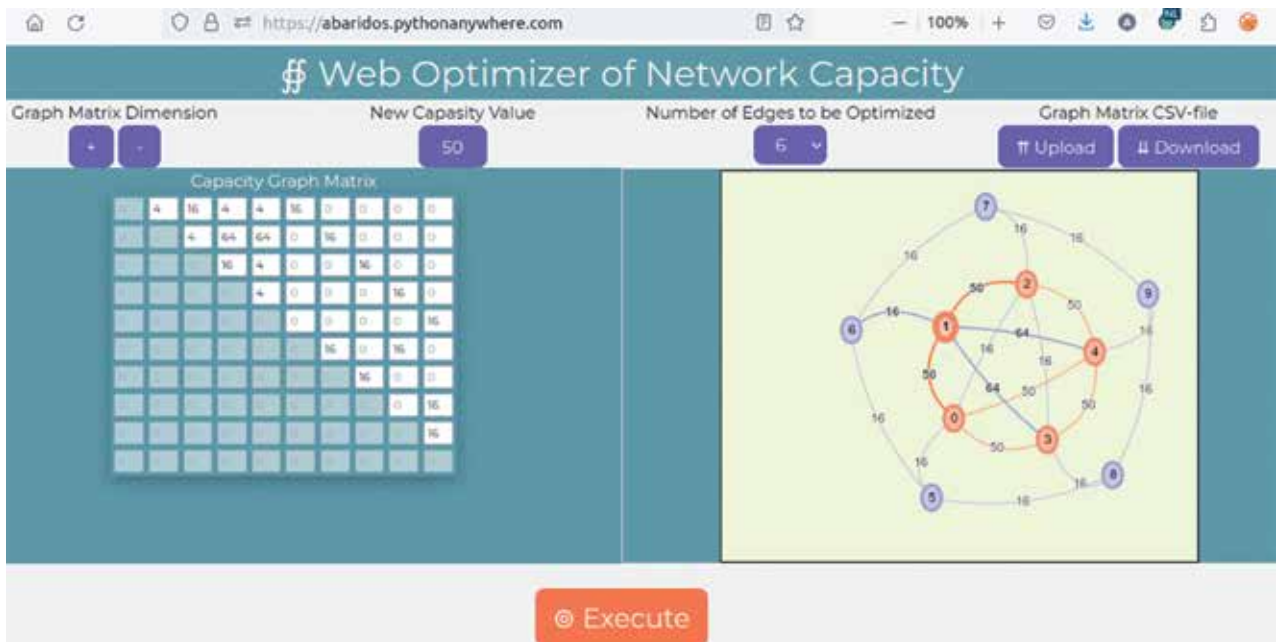


Fig. 11. Network optimization public web service
Source: developed by the authors

3.4. Approach application in software defined networks

To control the bandwidth of networks, the modern approach of Software-Defined Networks [12] is used, in which the network is controlled by virtual controllers. Software Defined Controller Services allows you to add functionality using a Virtual Network Function [13], which can dynamically control the bandwidth of individual graph edges, which will lead to a sharp increase in functionality. The software basis of such a virtual network function can be the service shown in Fig.11.

4. General development scheme

The development process can be described by the general scheme shown in the Fig. 12.

First of all, the needs of society should be investigated by scientists who study the general formulation of the problem and publish their solutions. Then the development of a prototype system, which could

be presented as an open-source solution, is given to students for implementation. After finding sponsors, a professional development teams will be included to the development for creation a global information system. Those team also support and develop the open-source solution together with the community. Proprietary and open-source solutions mutually develop each other at their levels, which leads to the satisfaction of the needs of society.

5. Results

The basic source code of system prototype for both parts of article could be found in [14-15]. Public web-service (web-optimizer of network capacity) of first part of article is shown in [16].

6. Conclusion

The article presents typical problems of constructing information systems. Specific examples show the difficulties of development such systems. These are the difficul-

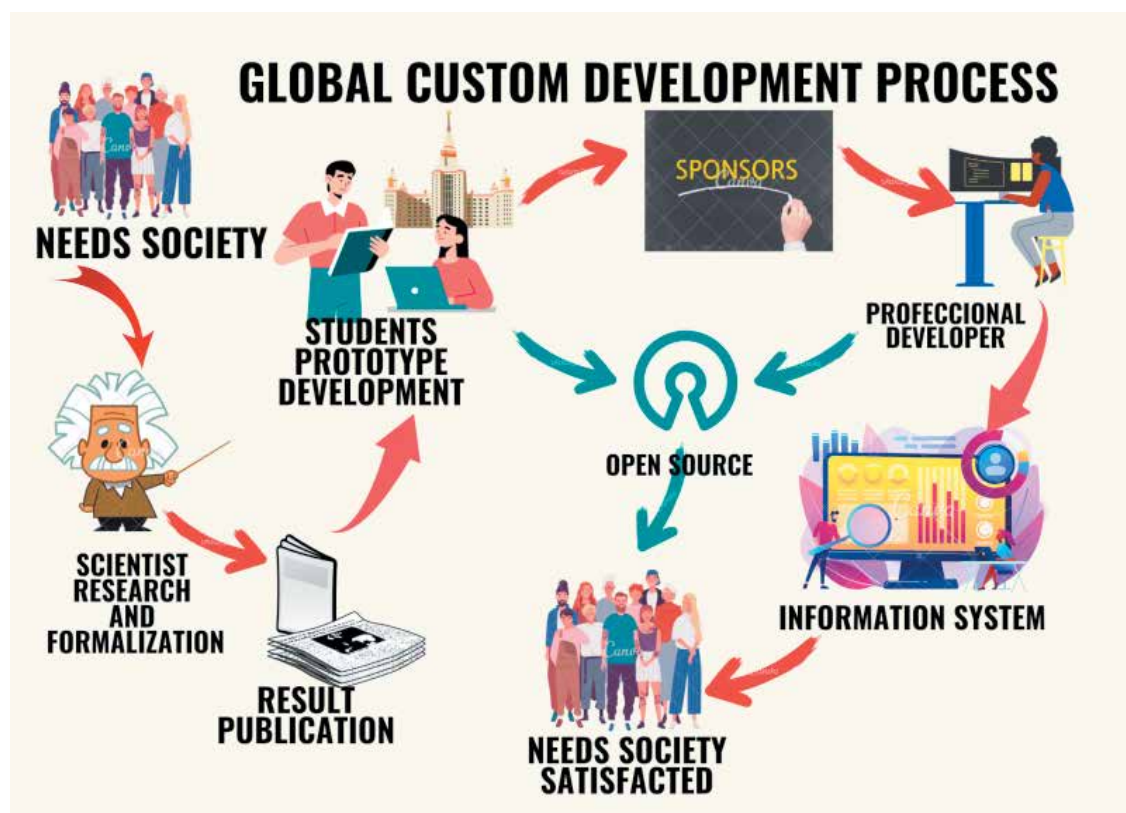


Fig. 12. Global custom development process

Source: developed by the authors

ties of determination the input information, which are solved by using new algorithms of fuzzy logic, which is shown in the first part of the article. And these are difficulties of the second part of the article associated with the uncertainties of optimization in the network calculus, which are successfully solved using a combination of known algorithms. All parts of the article are not limited to theoretical calculations, but end with writing the source code for real prototype systems. The study can be useful both for students of software engineering, and for scientists and professional developers of complex global systems.

Funding

The study had no funding.

Conflict of interest

There is no conflict of interest.

REFERENCES

1. Development of an algorithm for collecting and analyzing data with fuzzy logic for management decision support systems / Sinitsyn A.V., Lisay, A.A., Sinitsyn N.Yu., Buikin A.Yu. // Informatization and communication. - 2022. - No. 6. - P. 33-43. — DOI 10.34219/2078-8320-2022-13-6-33-43. — EDN UZZWIE.
2. Loginov L.R., Lisay N.Yu., Sinitsyn A.V. Development of a system for meeting the need for resources in emergency situations, Proceedings of the International Congress "Modeling of complex technical systems", conference "Modeling in engineering", Moscow, MSTU them. N.E. Bauman, 2023 (UDC 599.722).
3. Genetic Algorithm Application for Optimal Distribution of Network Resources for Best QoS under Financial

Constraints / Voronkov I.M., Sinitsyn A.V., Mukhamadiev V.I., Nazarov M.A. // 3rd International Science and Technology Conference "Modern Network Technologies 2020", MoNeTec-2020 — Proceedings: 3, Moscow, October 27-29, 2020. — Moscow, 2020. — P. 9258168. — DOI 10.1109/MoNeTeC49726.2020.9258168. — EDN KCFDLS.

4. Sinitsyn A.V. Multiservice network traffic optimization with communication matrix / Sinitsyn A.V., Selivanov S.A., Sinitsyn A. A. // Modern Network Technologies (MoNeTec-2022): Proceedings of the 4th International Conference (short and poster presentations), Moscow, October 27–29, 2022. — Moscow: Moscow State University named after Lomonosov M.V. Publishing House (printing house), 2022. — P. 43-47. — EDN KWQEEW.

5. The Ford-FULKERSON method and a genetic algorithm for optimal distribution of network resources under financial constraints / Voronkov I.M., Mukhamadiev V.I., Nazarov M.A., Sinitsyn A.V. // Informatization and Communication. — 2021. — No. 1. — pp. 7-14. — DOI 10.34219/2078-8320-2021-12-1-7-14. — EDN XCEPGW.

6. Novak, V., Perfilieva, I., Mockor, J. Mathematical Principles of Fuzzy Logic. Fizmatlit, 2006. — 352 c. — ISBN 0-7923-8595-0.

7. Betin V.N., Ivashchenko V.A. & Suprun A.P. Eliciting and the Use of Information Concerning Regular Structures in the Formalism of Functional Neural Networks in Decision-Support Systems. Autom. Doc. Math. Linguist. 56, 179–186

(2022). URL: <https://doi.org/10.3103/S0005105522040021>.

8. Betin V.N., Luk'yanov S.E., Suprun A.P. A Mechanism for a Solution Search within the Formalism of Functional Neural Networks. Autom. Doc. Math. Linguist. 54, 124–129 (2020). URL: <https://doi.org/10.3103/S0005105520030024>.

9. Thomas H. Kormen and etc. Algorithms: Constraction and Analisys, Introduction to Algorithms. — Editon 2. — M.: «Williams», 2006. — p.1296. — ISBN 0-07-013151-1.

10. Korte B.H., Jens Vygen. 8.4 Blocking Flows and Fujishige's Algorithm // Combinatorial Optimization: Theory and Algorithms (Algorithms and Combinatorics, 21) (eng.). — Springer Berlin Heidelberg, 2008. — P. 174—176. ISBN 978-3-540-71844-4.

11. Bessaou M. and Siarry P. A genetic algorithms with real-value coding to optimize multimodal continuous functions, J. Structural Optimization 23, pp.63-74.

12. Smelyansky R.L., Antonenko V.A., Concepts of Programmatic Management and Virtualization of Network Services in Modern Data Transmission Networks: (textbook) - Moscow: KURS, 2020. - 160 p. - (Modern network technologies).

13. SDN-controller RunOS 2.0 URL: <http://arccn.github.io/runos/docs-2.0/eng/index.html>

14. URL: <https://github.com/asinitsyn-info/Fuzzy-Disaster-Relief.git>

15. URL: <https://github.com/asinitsyn-info/QoSMatrix-GA-FFA>

16. URL: <https://abaridos.pythonanywhere.com/>

