

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна

ВІСНИК

Харківського національного університету
імені В.Н. Каразіна

Серія

«Математичне моделювання.
Інформаційні технології.

Автоматизовані системи управління»

Випуск 57

Серія заснована 2003 р.

BULLETIN

of V.N. Karazin Kharkiv National University

Series

«Mathematical Modeling.
Information Technology.
Automated Control Systems»

Issue 57

First published in 2003

Харків
2023

Статті містять дослідження у галузі математичного моделювання та обчислювальних методів, інформаційних технологій, захисту інформації. Висвітлюються нові математичні методи дослідження та керування фізичними, технічними та інформаційними процесами, дослідження з програмування та комп'ютерного моделювання в наукоємних технологіях.

Для викладачів, наукових працівників, аспірантів, працюючих у відповідних або суміжних напрямках.

Наказом Міністерства освіти і науки України від 17.03.2020 № 409 наукове фахове періодичне видання Вісник Харківського національного університету імені В.Н. Каразіна серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління» включено до Категорії «Б» Переліку наукових фахових видань України за наступними спеціальностями: 113 – Прикладна математика; 122 – Комп'ютерні науки та інформаційні технології; 123 – Комп'ютерна інженерія; 125 – Кібербезпека.

Затверджено до друку рішенням Вченої ради Харківського національного університету імені В. Н. Каразіна (протокол № 8 від 01.05.2023 р.)

Редакційна колегія:

Азаренков М.О. (гол. редактор),

д.ф.-м.н., академік НАН України, проф., ІВТ ХНУ імені В.Н. Каразіна

Жолткевич Г.М. (заст. гол. редактора), д.т.н., проф., ФМІ ХНУ імені В.Н. Каразіна

Лазурик В.Т. (заст. гол. редактора), д.ф.-м.н., проф., ФКН ІВТ ХНУ імені В.Н. Каразіна

Споров О.Є. (відповідальний секретар), к.ф.-м.н., доц. ФКН ІВТ ХНУ імені В.Н. Каразіна

Замула О. А., д.т.н., доц., ФКН ІВТ ХНУ імені В.Н. Каразіна

Золотарьов В.О., д.ф.-м.н., проф., ФТІНТ імені Б.І. Веркіна НАН України

Куклін В.М., д.ф.-м.н., проф., ФКН ІВТ ХНУ імені В.Н. Каразіна

Мацевитий Ю.М., д.т.н., академік НАН України, проф., фізико-енергетичний ф-т ХНУ імені В.Н. Каразіна

Рассомахін С. Г., д.т.н., доц., ФКН ІВТ ХНУ імені В.Н. Каразіна

Руткас А.Г., д.ф.-м.н., проф., ФМІ ХНУ імені В. Н. Каразіна

Стервєдов М.Г., к.т.н., доц., ФКН ІВТ ХНУ імені В.Н. Каразіна

Толстолузька О. Г. д.т.н., с.н.с., доц., ФКН ІВТ ХНУ імені В.Н. Каразіна

Угрюмов М. Л., д.т.н., проф., ІВТ ХНУ імені В.Н. Каразіна

Шейко Т.І., д.т.н., проф., фізико-енергетичний ф-т ХНУ імені В.Н. Каразіна

Шматков С. І., д.т.н., проф., ФКН ІВТ ХНУ імені В.Н. Каразіна

Щербина В.А., д.ф.-м.н., проф., ФМІ ХНУ імені В.Н. Каразіна

Раскін Л.Г., д.т.н., проф., Національний технічний університет "ХПІ"

Стрельнікова О.О., д.т.н., проф. Ін-т проблем машинобудування НАН України

Соколов О.Ю., д.т.н., проф., кафедра прикладної інформатики, університет імені Миколая Коперника, м. Торунь (Польща)

Prof. **Harald Richter**, Dr.-Ing., Dr. rer. nat. habil. Professor of Technical Informatics and Computer Systems, Institute of Informatics, Technical University of Clausthal, Germany

Prof. **Philippe Lahire**, Dr. habil., Professor of computer science, Dep. of C. S., University of Nice-Sophia Antipolis, France

Адреса редакційної колегії: 61022, м. Харків, майдан Свободи, 6, ХНУ імені В. Н. Каразіна, к. 534.

Тел. +380 (57) 705-42-81, Email: journal-mia@karazin.ua.

Мова публікації: українська, англійська.

Статті пройшли внутрішнє та зовнішнє рецензування.

Свідоцтво про державну реєстрацію КВ № 21578-11478 Р від 18.08.2015.

The articles are present research in the field of mathematical modeling and computing methods, information technologies, information security. New mathematical methods of research and management of physical, technical and information processes, research on programming and computer modeling in science-intensive technologies are covered.

For teachers, researchers, graduate students working in relevant or related fields.

By the order of the Ministry of Education and Science of Ukraine from 17.03.2020 № 409 scientific professional periodical Bulletin of V.N. Karazin Kharkiv National University series "Mathematical modeling. Information Technologies. Automated control systems" is included in Category "B" of the List of scientific professional publications of Ukraine in the following specialties: 113 – Applied Mathematics, 122 – Computer Science and Information Technology; 123 – Computer engineering; 125 – Cybersecurity.

Approved for publication by the decision of the Academic Council of V.N. Karazin Kharkiv National University (Minutes № 8 of 01.05.2023).

Editorial Board:

Azarenkov M.O. (Chief Editor), Acad. Of the NAS of Ukraine, Dr. Sc., Prof., HTI V.N. Karazin Kharkiv National University

Zholtkevich G.M. (Deputy Editor), Dr. Sc, Prof. MCS V.N. Karazin Kharkiv National University

Lazurik V.T. (Deputy Editor), Dr. Sc, Prof. CSD HTI V.N. Karazin Kharkiv National University

Sporov O.E., (Executive Secretary), Ph.D. Assoc. Prof, CSD HTI V.N. Karazin Kharkiv National University

Zamula A.A., Ph.D. Assoc. Prof, CSD HTI V.N. Karazin Kharkiv National University

Zolotarev V.A., Dr. Sc, Prof. B. Verkin Institute for Low Temperature Physics and Engineering of the National Academy of Sciences of Ukraine

Kuklin V.M., Dr. Sc, Prof. CSD HTI V.N. Karazin Kharkiv National University

Matsevity Yu.M., Acad. Of the NAS of Ukraine, Dr. Sc., Prof., DPE V.N. Karazin Kharkiv National University

Rossomakhin S.G., Dr. Sc, Prof. CSD HTI V.N. Karazin Kharkiv National University

Rutkas A.G., Dr. Sc, Prof. MCS V.N. Karazin Kharkiv National University

Styervoyedov N.G., Ph.D. Assoc. Prof, CSD HTI V.N. Karazin Kharkiv National University

Tolstoluzka O.G., Dr. Sc, Assoc. Prof. CSD HTI V.N. Karazin Kharkiv National University

Ugryumov M.L., Dr. Sc, Prof. HTI V.N. Karazin Kharkiv National University

Sheyko T.I., Dr. Sc, Prof. DPE V.N. Karazin Kharkiv National University

Shmatkov S.I., Dr. Sc, Prof. CSD HTI V.N. Karazin Kharkiv National University

Shcherbina V.A., Dr. Sc, Prof. MCS V.N. Karazin Kharkiv National University

Raskin L.G., Dr. Sc, Prof. National Technical University "Kharkiv Polytechnic institute"

Strelnikova E.A., Dr. Sc, Prof., NASU A. Pidgorny Institute of Engineering Problems

Sokolov O.Yu., Dr. Sc, Prof. Nicolaus Copernicus University, Torun, Poland

Prof. **Harald Richter**, Dr.-Ing., Dr. rer. nat. habil. Professor of Technical Informatics and Computer Systems, Institute of Informatics, Technical University of Clausthal, Germany

Prof. **Philippe Lahire**, Dr. habil., Professor of computer science, Dep. of C. S., University of Nice-Sophia Antipolis, France

Editorial Address: 61022, Kharkiv, Svobodi sq., 6, V.N. Karazin Kharkiv National University, r. 534.

Phone. +380 (57) 705-42-81, Email: journal-mia@karazin.ua.

Language of publication: Ukrainian, English.

The articles pass internal and external review.

Certificate of state registration: KV № 21578-11478P dated 18.08.2015

ЗМІСТ

▪ Безрук Ю. Р., Яновський В. В.	6
Еволюція популяції стратегій з пам'яттю з особливою малою групою	
▪ Бєлан А. О., Васильєва Л. В.	22
Розробка гібридної рекомендаційної системи	
▪ Зінов'єв Д. В., Ткачук М. В.	32
Аналіз, класифікація та тестування інструментальних засобів для управління конфігураціями програмних мікросервісів	
▪ Ніколюк П. К.	42
Маршрутизація транспортного засобу з оптимальним за часом динамічним міським шляхом руху	
▪ Яланецький В. А.	55
Автоматизована система проклеювання деталей паперових пакетів	

CONTENTS

▪ Bezruk Y., Yanovsky V.	6
Evolution of a population of strategies with memory in the presence of a distinct small group	
▪ Bielan A., Vasilieva L.	22
Development of a hybrid-based recommendation system	
▪ Zinov'ev D., Tkachuk M.	32
Analysis, classification and testing of configuration management tools for software microservices	
▪ Nikolyuk P.	42
Routing a vehicle through city traffic by the time-optimal dynamic path	
Yalanetskyi V.	55
Automated system for gluing paper bag parts	

УДК (UDC) 519.6, 51-76

Bezruk Yurii*student**V.N. Karazin Kharkiv National University, 6 Svobody Square, Kharkiv, Ukraine, 61022**e-mail: yurii.bezruk.it@gmail.com**<https://orcid.org/0009-0005-9202-4521>***Yanovsky Volodymyr***Doctor of Physical and Mathematical Sciences, professor**V. N. Karazin Kharkiv National University, 6 Svobody Square, Kharkiv, Ukraine, 61022;**Institute of Single Crystals, National Academy of Sciences of Ukraine, 60 Nauky Ave., Kharkiv, Ukraine, 61001.**e-mail: yanovsky@isc.kharkov.ua**<https://orcid.org/0000-0003-0461-749X>*

Evolution of a population of strategies with memory in the presence of a distinct small group

Topicality. Today, the human population is constantly growing. The influence of societies and subcultures on each other and on the population as a whole is also increasing. It is necessary to expand knowledge in the field of social behavior to improve the understanding of interaction within the population.

Key goals. The purpose of the article is to study the influence of selected distinct groups on the evolution of the population. The nature of changes in evolution is studied in the presence of groups of different complexity and size in the population. This work also aims to establish the most evolutionarily advantageous composition and size of a small group and the stage at which a population will be dominated by such a group, if it is possible.

Research methods. The evolution of a population with a full set of behavioral strategies, limited only by memory depth, in the presence of a distinct group is considered. Each strategy carrier interacts with all others, themselves included, according to an iterative model of the prisoner's dilemma. Rewards are determined by payment matrices. Each subsequent generation successively loses the most disadvantageous behavioral strategies of the previous generation. Carriers who are members of a distinct group behave more aggressively towards "strangers" and more cooperative with "their own". Different variants of the initial composition of the group are considered, namely the case when the average complexity of a small group is greater than the average complexity of the population as a whole, less, or comparable to it. Variants with different initial size of the small group have been considered – 5, 10, 15, 20% of the population size.

The results. The paper shows how the presence of small groups of different complexity and size affects population dynamics. An increase in the aggressiveness of the population has been revealed if there is a small group of any composition in it. The size and composition of a small group that makes it possible to dominate the society and the stage of evolution when it takes place have been established.

Conclusion. Based on the results of the article, the main consequences of the influence of distinct small groups on the evolution of the population have been determined.

Key words: *evolution, population, society, strategy, distinct group, complexity, aggressiveness, cooperation.*

How to quote: Bezruk Y., Yanovsky V., "Evolution of a population of strategies with memory in the presence of a distinct small group" *Bulletin of V.N. Karazin Kharkiv National University, series "Mathematical modelling. Information technology. Automated control systems*, vol. 57, pp. 6-21, 2023. <https://doi.org/10.26565/2304-6201-2023-57-01>

1. Introduction

Today, research on behavior in society, development and evolution of the groups forming it are becoming more relevant [1]. The population of people is growing and the influence of societies on each other is also increasing due to the rapid development of information technologies. People from different continents and countries, different religions and beliefs, with different cultures and traditions now coexist in a single information space, and therefore constantly influence each other. The study of such interaction is a key to understanding the future of humanity not only as a society, but also as a biological species.

Modern game theory [2] is one of the areas used to study the interaction of populations. A clear example of game models used for this purpose are the models based on the so-called "prisoner's dilemma" [3]. These models represent a paradigm which has helped to discover such mechanisms of cooperative behavior as voluntary participation, punishment, heterogeneous activity and social diversity.

The model of the prisoner's dilemma game is that two players can "cooperate" with each other (help each other) or "betray" the opponent (behave aggressively). At the same time, the reward matrix is chosen in such a way that the best payoff for both players can be obtained only when they both choose cooperation, but the highest payoff for one player can be obtained if one of them chooses betrayal and the other chooses cooperation. Thus, the locally optimal solution for both players is aggressive behavior, which is due to the real-life simulation: cooperative behavior usually requires additional resources, unlike selfish behavior.

In [4-7] the evolution of populations with a full set of behavioral strategies (limited only by the depth of memory) within the framework of a generalized prisoner's dilemma has been considered. Increases in average difficulty and average memory depth over time, as well as decreases in average aggressiveness, have been analyzed. Complexity has been recognized as an evolutionarily advantageous characteristic.

In this work, we have analyzed how the course of evolution changes when a distinct group, whose members actively support each other and behave more aggressive towards others, appears in the population. The main points of interest are: to what extent the appearance of such a group will increase the level of aggressive behavior in society, and whether being in such a group is evolutionarily beneficial for the strategy carriers. Additionally, it is interesting to see how a change in the composition of a small group will affect the development of society, what composition and size of the group are sufficient for its complete dominance in society, as well as how soon such dominance will occur.

2. Description of strategies with memory

A behavior strategy is a rule which determines a move depending on the known move of the opponent. Strategies that imply only the reaction to the opponent's current move are called the strategies without memory, and strategies that allows decisions based on knowledge of the opponent's several previous moves are the strategies with memory. The memory depth is the number of the opponent's moves, over the one available at a current stage, which is taken into account when choosing a move. Therefore, the strategies without memory have a memory depth of 0, the strategies that remember an additional 1 move to the existing one have a depth of 1, and so on.

Strategies without memory choose a move based on the opponent's previous move, i.e., the one made during the previous iteration. There are two move options - cooperation (marked as 1) and betrayal (marked as 0). To denote a strategy without memory, it is necessary to specify the carrier's responses to the opponent's moves 0 or 1. Those strategy responses can be set as follows 0: 0, 1: 0, that is, "if the opponent move was 0, then we choose the move 0, and if 1, then move 0 anyway." We have decided to denote such strategies as sequences of digits $\{0,1\}$ (i.e., as numbers in binary representation). Then the bit number is the opponent's move, and the value in it is the response to such a move. For example, the strategy "if the other player moved with 0, choose move 1, and if 1, then choose move 0" can be written in the form: 01. It should be noted that in the [4-7], the strategy record is inverted, and the reaction to move is placed on the left. But we have chosen to use the notation described above for greater similarity with the binary notation and for simplifying bitwise operations in the program.

Therefore, all strategies with zero memory depth can be written as a binary two-digit number – 00, 01, 10, 11. It should be mentioned, that the choice of the first move is a unique situation, at this point there is no information about previous opponent's moves and a special procedure for such a case is required. That is why we should add one auxiliary bit to determine a first agent's move, increasing the number of strategies from 4 to 8: [0]00, [1]00, [0]01, [1]01, [0]10, [1]10, [0]11, [1]11.

Strategies with memory depth of 1 make a move taking into account the opponent's current and previous move. So, the opponent's moves will be denoted with two bits: 00, 01, 10, 11. Therefore, a reaction to the opponent's moves (i.e., description or the "name" of the strategy) must be 4 characters long. If we convert those combinations of the opponent's moves from the binary system to the decimal system: 00 - 0, 01 - 1, 10 - 2, 11 - 3, we will get the bit numbers in which the answers to the corresponding combinations of moves should be stored. For example, denoting a strategy reaction to an opponent's moves as 00: 1, 01: 0, 10: 1, 11: 0, in binary notation would be 0101 (the least significant 00th bit rightmost). So, if we need 2 bits to describe a strategy with a memory depth of 0, and 4 bits for a memory depth of 1, then we need 2^{k+1} bits to describe a strategy with an arbitrary memory depth k . 2^N strategies can be formed, where N is the number of bits. So, for an arbitrary memory depth k , there are $2^{2^{k+1}}$ unique strategies. Therefore, for memory depth of 1, we have $2^{2^{1+1}}=2^4=16$ unique strategies.

The question now is how the strategy with memory depth 1 is supposed to react if only current enemy's move is known. That is why it is necessary to add additional reaction bits – a response to the opponent's first move. These strategies should be denoted [01]1001, where the 0th bit in the brackets is a response to the first move of 0, and 1st, to the move of 1. Afterwards, there will have been sufficient data to create a 4-digit description of the strategy. Furthermore, the strategies with memory depth of 1 require an additional bit for the first move like memoryless ones. Therefore, the notation of the strategy with a memory depth of 1 is: [0][10]0110.

Similarly, the description of strategies with memory depth of 2 is a bitwise arrangement of the strategy's responses to the opponent's current move and the 2 previous ones, that is, the {0,1} sequence with the length of 3 characters (010, 101, ...). Using the method introduced above – denoting several moves as a number, converting it to decimal (101 - 5, 011 - 3, ...), and writing the response to this sequence of moves in the bit by the received decimal number, we have concluded that 8 bits to denote such a strategy are required. Moreover, the strategies with memory depth of 2 also need extra bits to record situations with insufficient data, when only the current move or the current and one previous move are known. Therefore, the notation of the strategy with memory depth of 2 is: [0][10][0110]11001011. A set of strategies with the same name but different "sub-strategies" forms a family of strategies.

3. Interaction of strategies

Since we have chosen a prisoner's dilemma game model to describe carriers' interaction, the locally optimal solution for both players is aggressive behavior. This is due to the simulation of real life: cooperative behavior usually requires additional resources, unlike selfish behavior. The matrix of rewards represents the "reaction of the world" to the strategy behavior, because, eventually, we use them to determine which behavior helps strategies to survive and which does not. As in [4-7], Axelrod's matrix [8] (Table 1) is used to describe the rewards:

Table 1 Axelrod's reward matrix

A\B	0	1
0	1	5
1	0	3

After each move, the carrier's strategy is perceived as strategy A, and its move corresponds to the matrix rows, and the opponent's strategy as strategy B, and its response corresponds to the matrix columns. The reward that the strategy receives is located at the intersection. The strategies that are not used by a small group are rewarded according to this matrix.

We can influence group behavior by changing the values of the reward matrix for group members, stimulating them to take certain actions. In this case, we are interested in a classic "sect" behavior: group members should actively cooperate with each other, help each other, but treat "strangers" aggressively. That is why we set two more reward matrices: members of the small group will be rewarded by using the first one, when they interact with "strangers", and by the second one, when they interact with "their own".

M₂ matrix shown in Table 2 is a modification of the M₁ matrix, where aggressive actions are rewarded by more points, and cooperative actions by less points. It is used for "aggressive towards strangers" interactions of the small group.

Table 2 – Reward matrix for "aggressive towards strangers" interactions

A\B	0	1
0	2	6
1	0	2

For "helpful toward one's own" interactions matrix, the M₃ matrix shown in Table 3 has been chosen, where the rewards for the aggressive behavior are reduced and for the cooperation are increased.

Table 2 – Reward matrix for "helpful toward one's own" interactions

A\B	0	1
0	0	3
1	2	5

We have used the prisoner's dilemma model iteratively – each strategy carrier interacts with all others (including themselves) a given number of times, receiving rewards according to the established reward matrices. After all players in a given generation have played with everyone, we average the scores across all strategy families, then remove the family with the lowest average score from population and start a new generation with the remained strategies.

4. Collective variables

The number of strategies with memory depth of 0 is relatively small, so it is easy to keep track of each individual strategy in a generation. But the situation changes drastically as the memory depth increases, because the number of strategies in this case grows exponentially: $2^{2^{k+1}}$ strategies for the memory depth of k . Therefore, to track the dynamics of the population as a whole, it will be appropriate to introduce some collective variables that will reflect the state of the entire population.

The average complexity of the population is the arithmetic average of the complexities of all strategies in a generation. We introduce the concept of strategy complexity as described in [4-7], so the result can be compared with the usual process of evolution. This approach is based on the principle of comparative complexity of polynomial functions, which states that polynomials of a higher degree are more complex than polynomials of a lower degree.

The sequences of symbols $\{0,1\}$ $x = x_1x_2 \dots x_n$ can be perceived as a function that for each integer value i returns the value $x_i \in \{0,1\}$. Then, to calculate the complexity of the function, the difference operator $A:M \rightarrow M$ is used:

$$y = Ax, \quad (4.1)$$

where elements of sequence $y = y_1y_2 \dots y_n$ can be defined as

$$y_i = x_{i+1} - x_i, \quad (4.2)$$

where $i=1, 2, \dots, n$ is an element number in a sequence.

If we use the cyclicity of the binary sequence, or assume that $x_{n+1} = x_1$, then it is possible to calculate the elements of the sequence y for any strategy, i.e., find the strategy into which the chosen one is switched by using the specified differential operator. We assume that the strategies into which the initial strategies are switched by using the operator are simpler. Thus, a graph of strategies is formed, with the simplest and most aggressive strategy – 00 (or 0000 or 00000000 for strategies with memory), which by using the operator are switched into itself, at the bottom. Assuming that its complexity is 0, and the complexity for the each subsequent level of such a graph increases by one, then for the strategies without memory we have the following graph:

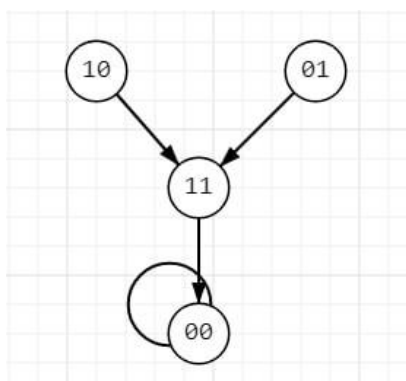


Fig. 4.1 – Complexity graph for memory depth of 0

Using such the approach, it can be determined that the strategy 11 has a complexity of 1, and the strategies 10 and 01 have a complexity of 2. In the same way, we can build a complexity graph for the strategy with any memory depth, for example, for depth 1 we have the graph shown in Fig. 2.

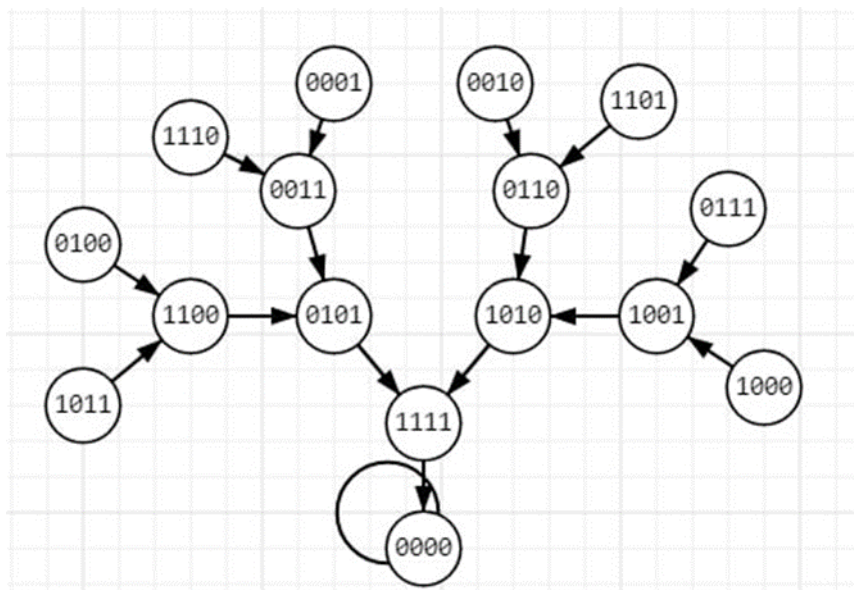


Fig. 4.2 – Complexity graph for memory depth of 1

The level of the graph corresponds to the strategy complexity. By constructing such a graph for strategies of memory depth 2, we obtain the complete complexity distribution for all carriers in the population.

The next collective variable considered is the average memory depth. Even though the strategy complexity directly depends on its memory depth, this variable is still interesting to consider, because it shows how strongly more primitive strategies behave in population of more advanced individuals and how they affect it.

A very important value is the average aggressiveness of the population, i.e., the relative part of aggressive moves that the strategy performs in each generation. Since we have stimulated the aggressive behavior of group members, aggressiveness is expected to exceed the level of the standard evolution.

It is also interesting to consider such a value as the average "earnings" – the average number of points that the strategy receives in one move. Using it, we can determine the reward "profitability" for certain modifications of the population.

Moreover, the size of the group relative to the population should also be considered – a percentage of population size in each generation. What is also interesting is the data on the size and composition of the group at the moment it is able to completely "dominate" the population displacing all other strategies, and how quickly that can happen.

5. Modelling

Since the influence of a small group on the evolution of a population can depend on the composition of a small group directly, several options for dividing strategies into large and small groups should be considered. A convenient measure for comparing groups of strategies is the average complexity. From [4], we know that the average complexity of a population with a memory depth of 2 without modifications at the beginning of the simulation is approximately equal to 7. Therefore, it would be interesting to consider how the nature of evolution can change if strategies of different complexity are adopted by a small group. Let us consider three cases, when the average complexity of the group is greater, smaller, or comparable with the complexity of the population.

Another factor that can strongly influence the nature of evolution is the size of a small group relative to the population. For each of the three variants of group complexity, we have conducted separate studies where the size of group is 5, 10, 15 and 20% of population size.

Strategies that have the same complexity may differ in other collective variables, such as aggressiveness, and therefore may affect the population differently.

For the next step, several groups with a random selection of its participants but with the same average difficulty have been considered. Averaging the results of modeling the influence of such groups on the population allows obtaining more universal dependencies. Such dependencies should not be sensitive to the composition of groups. A series of ten experiments have been conducted with identical inputs of

complexity and group size. The results have been averaged to obtain the final values of the collective variables.

5.1. The small group with greater complexity than the population

First, we have considered the case when the average complexity of a small group is greater than that of the population. That is, we have had a separate intellectual group of more "highly developed" strategies. Since the maximum possible complexity of the strategy at memory depth 2 is 8, we have created a group only from representatives of this complexity, and considered the evolution where the group size is 5, 10, 15 and 20% of the population.

At the size of 5%, a group does not make serious changes in the nature of the average complexity dynamics (Fig. 5.1). It can be seen that starting from the 217th generation, the average complexity of a small group drops, this is due to the fact that in some experiments the members of the group disappear in these generations (so its complexity is equal to zero), which lowers complexity, when averaged over 10 experiments. A size of 10% extends the survival period of the group, but still not enough to survive to the end of modelling. But the size of 15% is enough for a group to dominate in the later stages of evolution. At the same time, we can observe an increase in the average complexity of society. The increase in complexity is insignificant, which can be explained by the small difference between the complexity of the group (8) and the population (~7).

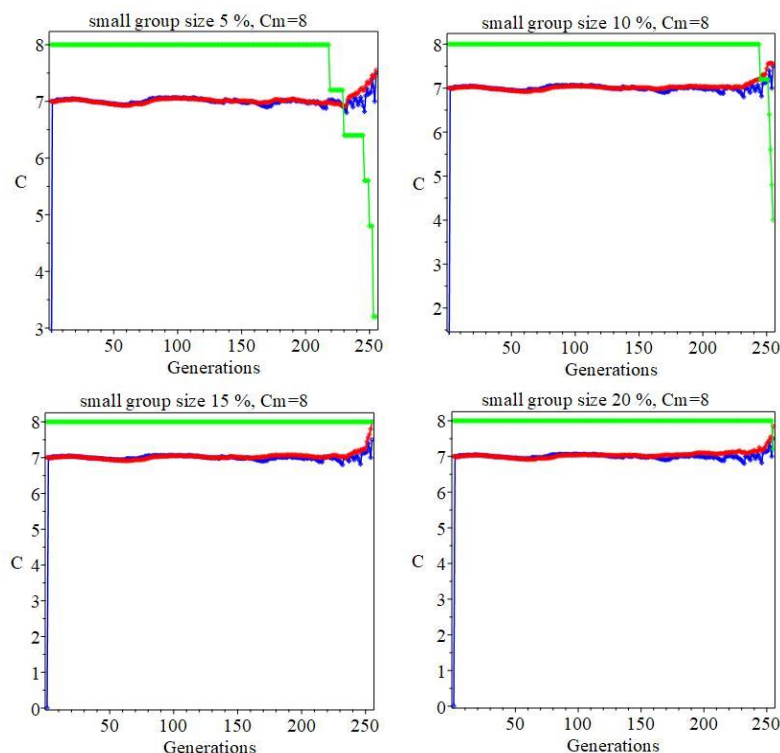


Fig. 5.1 Average complexity dynamics. Red color indicates data for the population with a small group, blue - data for the population without a small group, green - characteristics of the small group itself. The complexity of the small group is equal to 8. There is a noticeable increase in the average complexity of the population even in comparison with the change in the average complexity of the population without the presence of a group. Each picture corresponds to a certain initial size of the small group, which is indicated above it.

The average memory depth of the population slightly decreases at the later stages of evolution (Fig. 5.2) with a big group percentage (15-20%). This is because in the later stages, most of the surviving strategies of memory depth 2 are members of the group, and simpler strategies of smaller memory depths remain in the population.

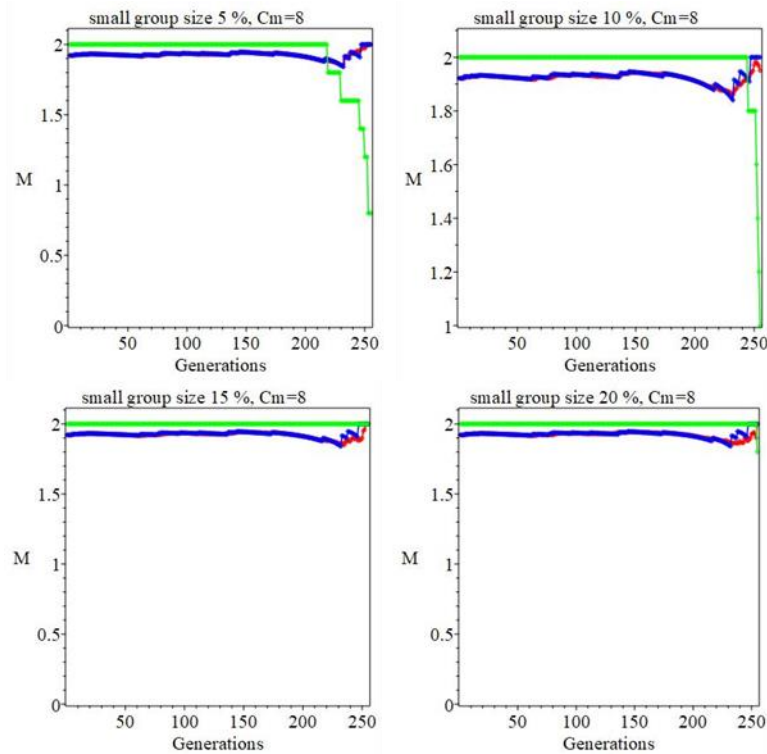


Fig. 5.2 Changes of the average memory depth in the population with a small group (red), without a small group (blue) and in the small group itself (green).

The average aggressiveness of both the population and the small group increases compared to the standard version of the population (see Fig. 5.3). As expected, the small group itself has very high aggressiveness regardless of its size. Nevertheless, its size strongly affects the aggressiveness of the population - it increases slightly at 5%, becomes increasingly larger at 10 and 15%, and at a size of 20% aggressiveness almost reaches the level of the small group itself.

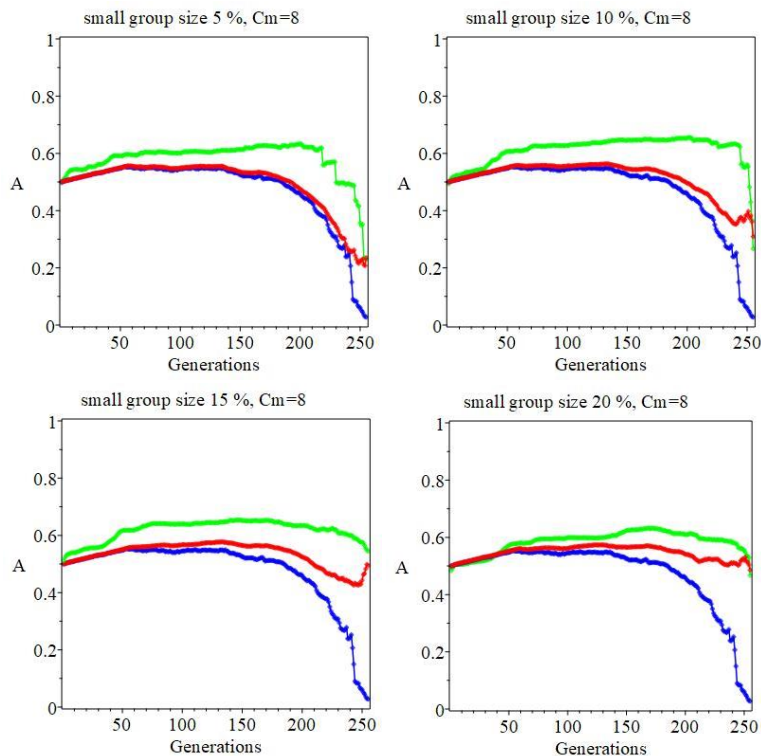


Fig. 5.3 Evolution of the average aggressiveness of the population with a small group (red), without a group (blue) and of the group itself (green).

As shown in [5], the value of the average number of points per move and the aggressiveness of the population are related by the equation:

$$P(t) = P_{\max} - \frac{(A(t) - A_{\min})^\alpha}{\lambda} \quad (5.1)$$

where $\alpha = 2$ in the absence of a small group, and λ is determined from the requirement of coincidence at the initial point.

However, it turns out that the number of small groups affects the value of α . It decreases in proportion to the initial number in the group. That dependence is in good agreement with the behavior of the average number of points obtained during modeling (see Fig. 5.4). That is, as the average aggressiveness increases, the average earnings per turn decreases, so it becomes lower and lower as the group size increases. As expected, the average earnings of the small group are higher throughout most of the evolution, but on the later stages it declines as the group strategy displaces other strategies. Thus, the number of a small group affects the relationship between aggressiveness and the average number of points per move, significantly reducing them with an increase in the number of a small group.

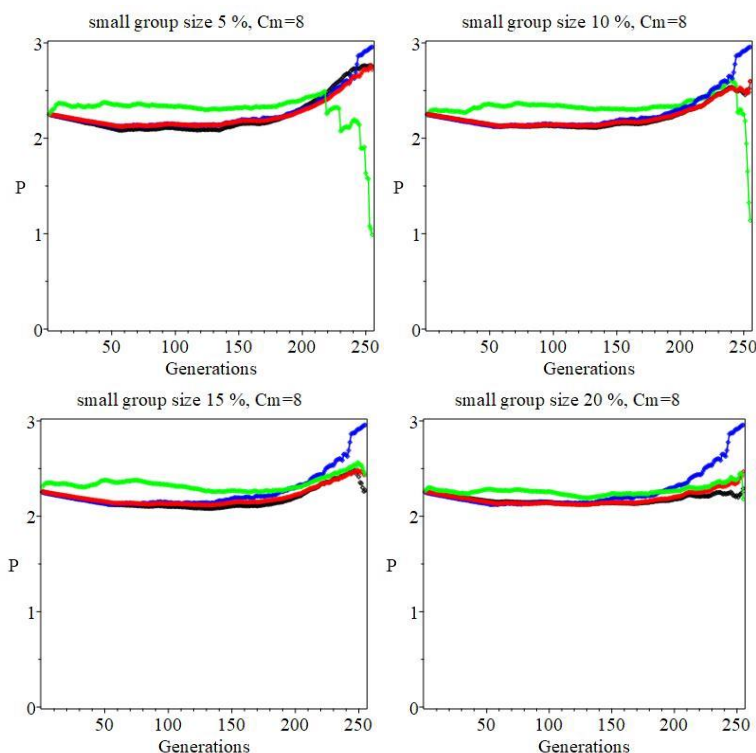


Fig. 5.4 The change in average carriers' points per turn in the population with a small group (red), without a small group (blue), in the small group itself (green). Equation (5.1) is shown in black. $\alpha = 1.6$ for a small group of 5%, $\alpha = 1.2$ for 10%, for 15% - $\alpha = 0.8$, and for 20% - $\alpha = 0.4$. The good consistency of this dependence with the simulation data (red) is noticeable.

The graphs of the relative group size (group size divided by size of the whole population) best demonstrate population changes with evolution (see Fig. 5.5) – we can see that a size of 5% for a small group does not provide much of survival advantage. A size of 10% of the population gives a slightly better result, but the group is still a minority even on the late stages. Only size of 15-20% allows a small group to dominate the society.

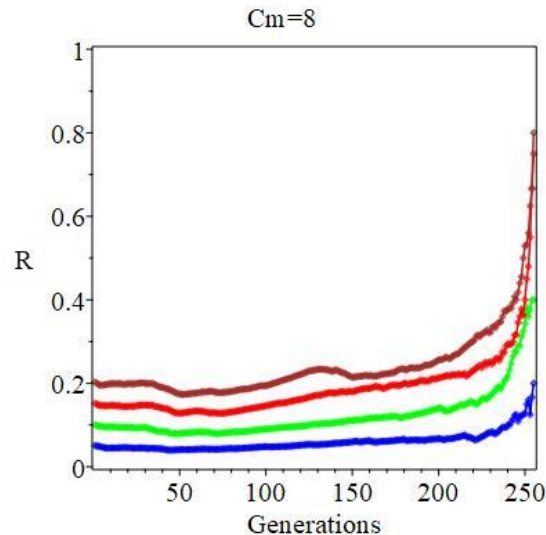


Fig. 5.5. The relative size of groups R changes during evolution for different initial sizes. Brown – initial 20%, red – 15%, green – 10% and blue – 5%. In all cases, the size of small groups is increasing.

5.2. The small group with the same complexity as the population

Let us now consider the modeling of the evolution with a different composition of the small group. This time the average complexity of the "sect" will be compared to the complexity of the population as a whole (~ 7). The group will randomly include strategies with complexity 6 and 8, but the main part of strategies will have complexity of 7. Again, in order to level the influence of the choice of strategies, the results have been averaged over ten experiments.

An interesting result is that the presence of a selected group, which has a similar "gene pool" to the population, has almost no impact (see Fig. 6.6) on the development of society, the average complexity and average memory depth remain on the same levels, as in the standard version of evolution.

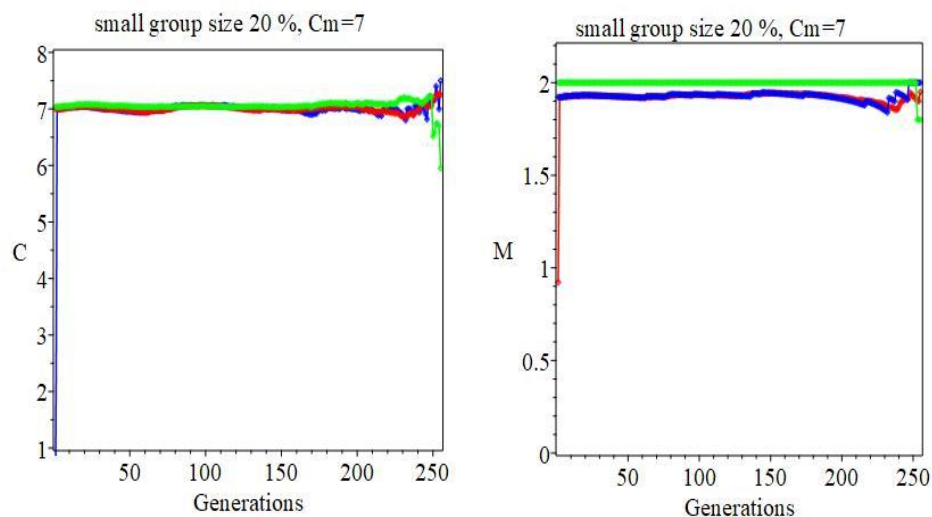


Fig. 5.6. The changes in average complexity (left) and average memory depth (right) of populations during evolution process. They are given for the largest number of the small group (20%) where the differences are most noticeable. Population with a small group marked with red, without a small group with blue, with the small group itself with green.

In such a population, the aggressiveness is the most variable value, which is shown in Fig. 5.7. When small group size is equal to 20%, the aggressiveness of the population as a whole is almost equal to the aggressiveness of a small group.

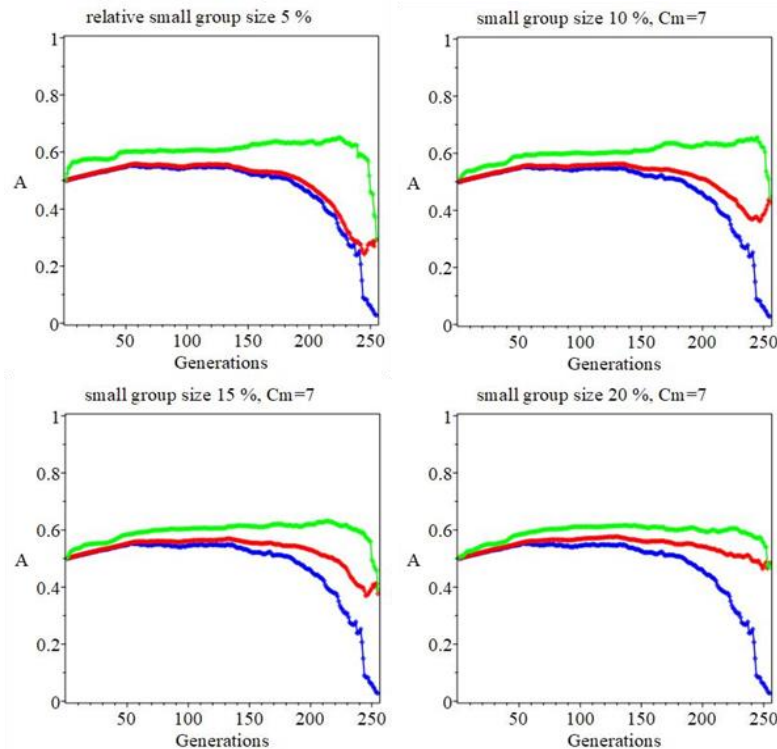


Fig. 5.7. The dynamics of the average aggressiveness of populations and groups over "time". Significant changes in aggressiveness are noticeable in comparison with the aggressiveness of the population without the small group. The difference is especially noticeable at the final stages of evolution, where the aggressiveness of a population with a small group approaches the aggressiveness of a small group itself.

The average number of points per move is roughly the same as in the case of the "more complex" small group. But it can be noticed that the group has a higher survival rate – group members more often survive to the end of the simulation, and in fewer experiments the group disappear completely at the later stages.

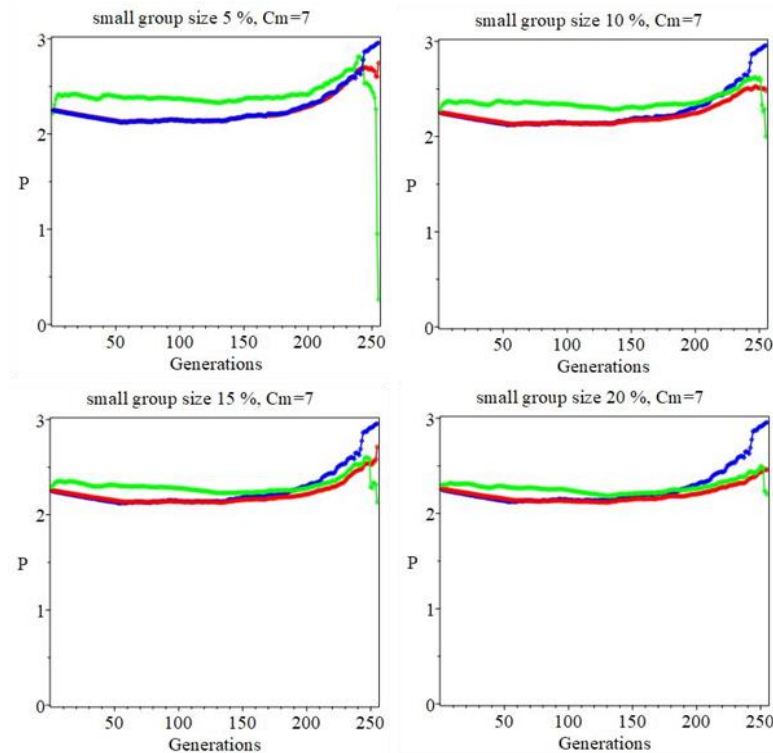


Fig. 5.8. The dynamics of the average number of points per move of populations and groups over "time". The results are better than for the previous group composition (see Fig. 5.4).

The relative size of a small group during the modeling is less (see Fig. 5.9.) than for the “intellectual” group, but then in the later stages of evolution there is a steeper “striking” – so, the group becomes a majority of the finalists by 10% of the population size.

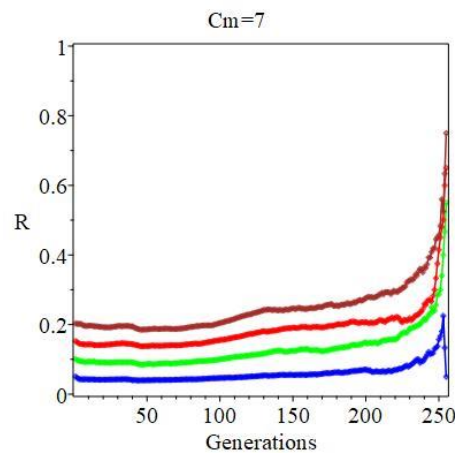


Fig. 5.9. The dependence of changes in the relative size of groups over the "time" for different initial sizes. Brown is initial 20%, red is 15%, green is 10% and blue is 5%.

5.3. The small group with lower complexity than the population

To study the impact of a more primitive group on society, we have added speakers with complexity 4, 5 and 6 to the "sect". The average complexity of the small group at the beginning of the simulation is approximately 5.45.

Observing the change in the average complexity, we can see that even a size of 5% of the population negatively affects the complexity of the whole population on later stages (compared to the standard evolution), but this size is not enough for dominating the group in all conducted experiments (Fig. 5.10). But starting with a size of 15%, the group of strategies with lower complexity succeeds in dominating the majority, which greatly reduces the average complexity of the population as a whole.

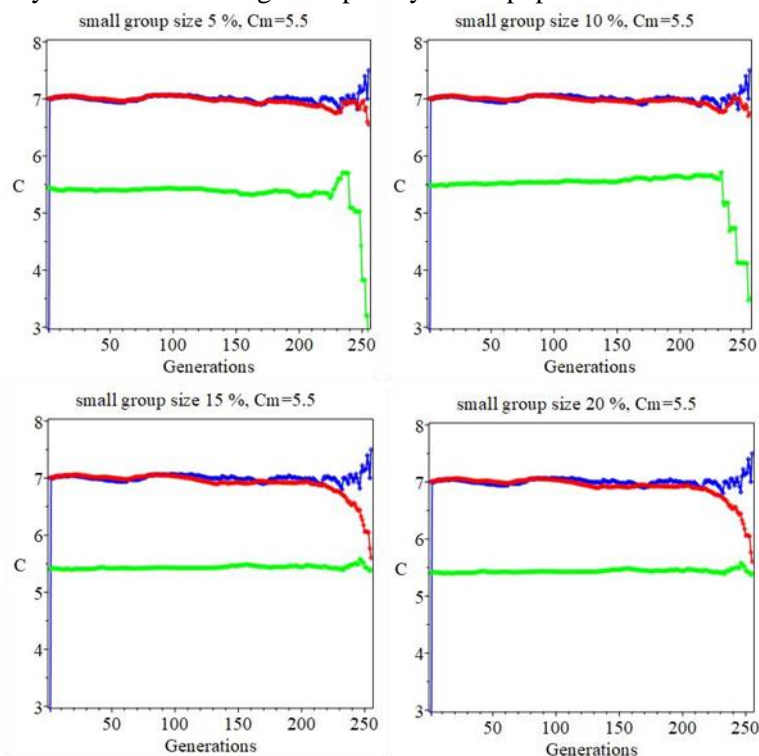


Fig. 5.10. A significant influence of a small group on the nature of the change in population complexity is noticeable. Instead of an increase in complexity, as in a population without a small group, a decrease in the complexity of the population is observed.

The same negative effects are observed with the average memory depth of the population (see Fig. 5.11). At the final stages of evolution, the nature of dynamics fundamentally changes. The difference become especially noticeable with an increase in the initial number of a small group (starting from 15%).

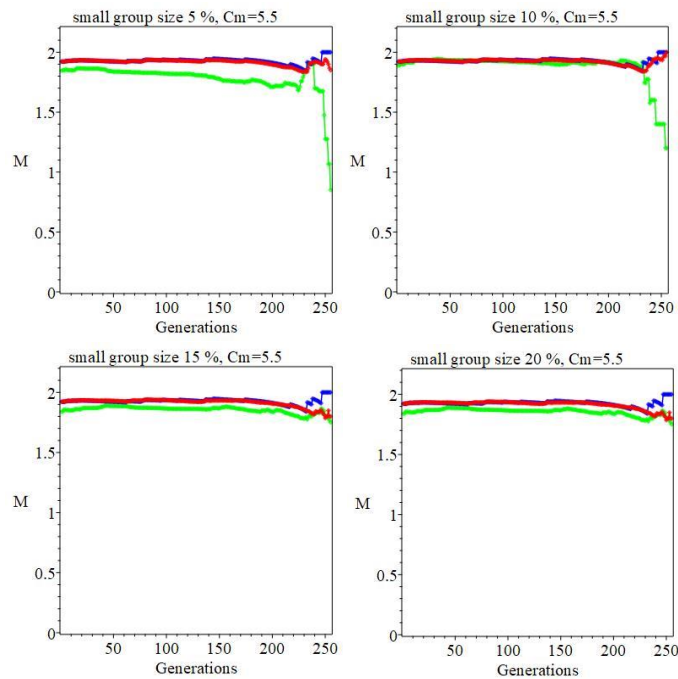


Fig. 5.11. The changes in the average memory of the population with a small group (red), without a small group (blue) and of the small group itself (green) are shown. For 15% and 20%, the average memory of the population approaches the average memory of a small group.

Aggressiveness of the behavior increases significantly. The average number of aggressive moves in the population reaches the highest values compared to previous experiments. At the size of 20% of the population, the average aggressiveness of society is almost equal to the average aggressiveness of a small group.

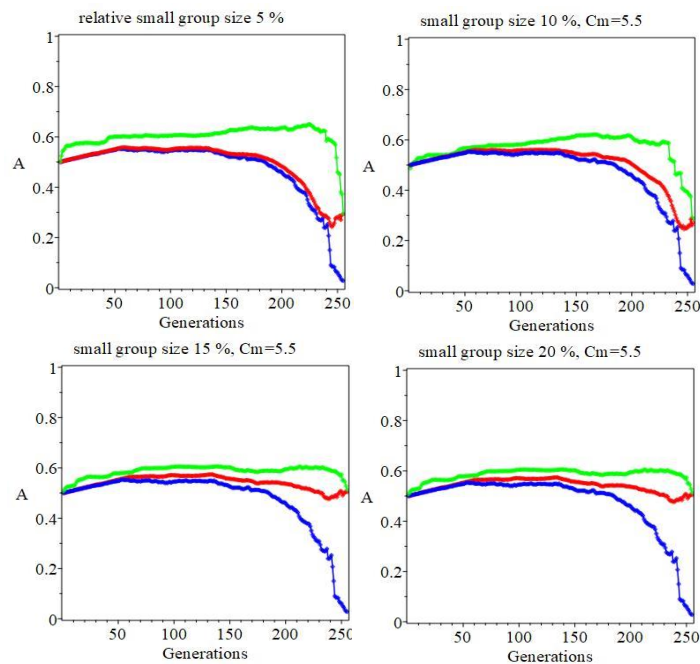


Fig. 5.12 Differences in the behavior of the average aggressiveness of the population with a small group (red), without a small group (blue) and the small group itself (green). Different pictures correspond to different sizes of the small group.

The typical changes in obtaining points per move during the evolution remain. Although the average earnings per move is lower than for the standard evolution, it still brings better results compared to previous experiments.

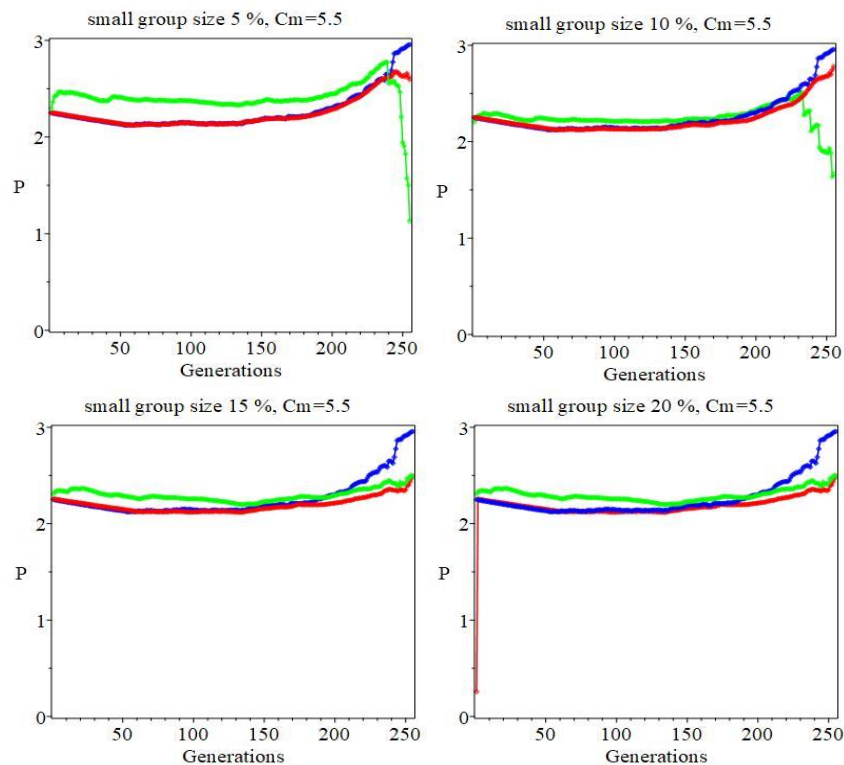


Fig. 5.13 A significant decrease in rewards is noticeable especially for a small group with an insignificant initial number of the group of 5%, 10%. With larger quantities, the rewards of group members are larger than in the population.

We can also observe that this version of the group composition has the highest survival rate among the conducted experiments. This is also confirmed by the charts of the dynamics of small group’s relative size – small group has the majority on the later stages of evolution already on 10% of the size. With a size of 20%, the number of the group is record-breaking – a group size reaches the half of the population in the 237 generation.

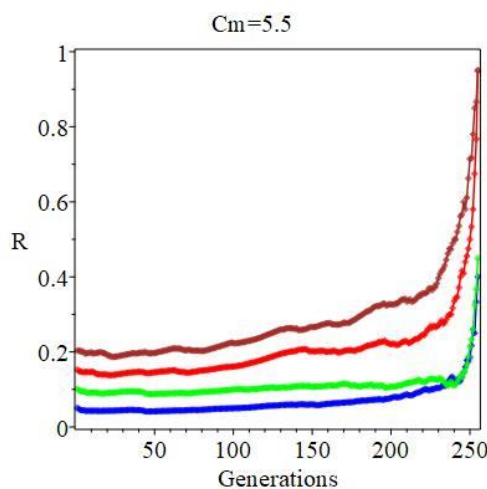


Fig. 5.14. A typical increase in the relative group size in the evolution process. Brown is initial 20%, red is 15%, green is 10% and blue is 5%.

The following chart (Fig. 5.15) shows the relative size of the population for 10 generations before the end of the simulation depending on the initial size of the group, for three variants of the composition of

such a group. We can see that such a function has a much larger angle of growth for a primitive "sect", although its initial values are smaller than in more developed versions of society.



Fig. 5.15. The relative size of the population for 10 generations before the end of the simulation depending on the initial size of the group. Blue color stands for the group with higher complexity, green for the same complexity, and red for the group with lower complexity as in population

6. Conclusions

Comparing the evolution of the population with the distinct group with the evolution of the population without any distinct groups, we can identify typical features of the influence of group presence on the evolution of the population.

As a result, we see that the presence of selected groups of strategies in the population has a significant impact on the nature of evolution. The strategies of the small group interact with each other and the strategies of the population according to different reward matrices. It should be emphasized that more cooperative interaction between group members and more aggressive towards other members of the population, as discussed in the paper, is typical for isolated groups even in real societies. "Intellectual sects" with an increased level of complexity of participants have a positive effect on the "intellectual" development of society, and groups with strategies of lower complexity impacts the population negatively. In all variants of the experiments, the aggressiveness of the population increases significantly with a small group of any composition. If a small group of strategies has a complexity comparable to the complexity of society, aggressiveness is the only indicator that shows any changes. That is, the presence of isolated groups always leads to an increase in the aggressiveness of the population. That results in decreasing points per move of the population strategies and thus the average rewards when the strategies interact. It has been determined that the relationship (5.1) between aggressiveness and points per move is preserved, but the number of a small group leads to a decrease of α dependence indicator in proportion to its number. It should be emphasized that a particularly significant decrease in points is observed for group members. An exception is the case of groups of low complexity with a relatively large initial number (15%, 20%) where the same level of rewards is achieved.

The relative group size increases in the process of evolution. An exception has been observed in the case of the group with complexity 7 with an initial concentration of 5%. An increase in the relative size of the group in the population is typical. A group of low complexity relative to the complexity of the population at an initial number of 20% even dominates the population (i.e., fully displaces all strategies that are not in the group).

An interesting fact is that it is most evolutionarily beneficial for carriers to be a part of "primitive" sect – such a group dominates the majority in society the fastest, "knocking off" the other strategies that do not have an additional "stimulus".

The issue of higher complexity of a small group (more than 8) remains open for the further research due to the operational complexity of calculations (strategy complexities of more than 8 require calculations at a memory depth of 3). It can be supposed that if the difference between the average complexity of the population and the complexity of the "intellectual" sect will be greater, being in such a group will be more rewarding than in the "primitive" one.

The impact of the reward matrix on the influence of the group on the evolution of the population also remains unexplored. Among the other interesting options, we can mention the study of cooperative

behavior towards “one’s own” without stimulating aggressive actions towards “strangers”. That should not lead to such a rapid increase of aggressiveness as has been observed in this study. The reverse option is also of interest to research.

ЛІТЕРАТУРА

1. Jans L., Leach C.W., Garcia R.L., Postmes T. “The development of group influence on in-group identification: A multilevel approach.” *Group Processes & Intergroup Relations*, 2015, vol. 18, с. 190-209.
https://www.researchgate.net/publication/264000793_The_development_of_group_influence_on_in-group_identification_A_multilevel_approach
2. Weibull J. W. “Evolutionary game theory.” Cambridge: MIT press, 1997. 265с.
<https://mitpress.mit.edu/9780262731218/evolutionary-game-theory/>
3. Nowak M., Sigmund K. “A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner’s Dilemma game.” *Nature*, 1993, vol. 364, с. 56-58.
<https://www.nature.com/articles/364056a0>
4. Куклін В.М., Приймак О.В., Яновський В.В. “Influence of memory on population evolution” *Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*, Вип. 29, 2016, с. 41-66. <https://periodicals.karazin.ua/mia/article/view/6557>
5. Куклін В.М., Приймак О.В., Яновський В.В. “The memory and the evolution of populations” *Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*, Вип. 35, 2017, с. 38-60. <https://periodicals.karazin.ua/mia/article/view/9841/9365>
6. Куклін В.М., Приймак О.В., Яновський В.В. “The evolution of strategies communities in the presence of sources”, *Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*, Вип. 36, 2017, с. 68-84. <https://periodicals.karazin.ua/mia/article/view/10098>
7. Поричанський В.В., Приймак О.В., Яновський В.В. “Alternative evolution of strategies with memory” *Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*, Вип. 44, 2019, с. 74-87. <https://periodicals.karazin.ua/mia/article/view/15775>
8. Axelrod R. “The evolution of cooperation” New York: Basic Books, 1984. 9с.
<https://ee.stanford.edu/~hellman/Breakthrough/book/pdfs/axelrod.pdf>

REFERENCES

1. L. Jans, C.W. Leach, R.L. Garcia, T. Postmes, “The development of group influence on in-group identification: A multilevel approach.” *Group Processes & Intergroup Relations*, vol. 18, 190-209, February 2015.
https://www.researchgate.net/publication/264000793_The_development_of_group_influence_on_in-group_identification_A_multilevel_approach
2. J. W. Weibull *Evolutionary game theory*. Cambridge: MIT press, 1997.
<https://mitpress.mit.edu/9780262731218/evolutionary-game-theory/>
3. M. Nowak, K. Sigmund “A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner’s Dilemma game.” *Nature*, vol. 364, 56-58, July 1993.
<https://www.nature.com/articles/364056a0>
4. V. M. Kuklin, O. V. Pryimak, V. V. Yanovsky. “Influence of memory on population evolution” *Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated control systems»*, vol. 29, 41-66, April 2016. [In Russian]
<https://periodicals.karazin.ua/mia/article/view/6557>
5. V. M. Kuklin, O. V. Pryimak, V. V. Yanovsky. “The memory and the evolution of populations” *Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated control systems»*, vol. 35, 38-60, November 2017. [In Russian]
<https://periodicals.karazin.ua/mia/article/view/9841/9365>

6. V. M. Kuklin, O. V. Pryimak, V. V. Yanovsky. "The evolution of strategies communities in the presence of sources", *Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated control systems»*, vol. 36, 68-84, December 2017. [In Russian] <https://periodicals.karazin.ua/mia/article/view/10098>
7. V. V. Porichansky, O. V. Pryimak, V. V. Yanovsky. "Alternative evolution of strategies with memory" *Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated control systems»*, vol. 44, 74-87, December 2019. [In Russian] <https://periodicals.karazin.ua/mia/article/view/15775>
8. R. Axelrod, *The evolution of cooperation*, New York: Basic Books, 1984. <https://ee.stanford.edu/~hellman/Breakthrough/book/pdfs/axelrod.pdf>

Безрук

Юрій Русланович

студент магістратури

*Харківський національний університет ім. В.Н. Каразіна, майдан
Свободи, 6, Харків, Україна, 61022*

e-mail: yurii.bezruk.it@gmail.com

<https://orcid.org/0009-0005-9202-4521>

Яновський

Володимир

Володимирович

д. ф-м. н., професор

*Харківський національний університет ім. В.Н. Каразіна, майдан
Свободи, 6, Харків, Україна, 61022*

*Інститут монокристалів, Національна Академія Наук України,
проспект Науки, 60., Харків, Україна, 61001.*

e-mail: yanovsky@isc.kharkov.ua

<https://orcid.org/0000-0003-0461-749X>

Еволюція популяції стратегій з пам'яттю з особливою малою групою

Актуальність. На сьогоднішній день популяція людей постійно зростає. Вплив суспільств та субкультур одне на одного та на популяцію в цілому теж збільшується. Необхідно розширити знання в області поведінки в соціумі для покращення розуміння процесів взаємодії всередині популяції.

Мета. Метою роботи є дослідження впливу виділених особливих груп на еволюцію популяції. Досліджується характер змін в еволюції за умови знаходження в популяції груп різної складності і кількості. Дослідження також має на меті встановити найбільш еволюційно вигідний склад і розмір малої групи і етап, на якому популяція буде захоплена такою групою, якщо це можливо.

Методи дослідження. Розглянуто еволюцію популяції з повним набором стратегій поведінки, обмежених тільки глибиною пам'яті, за наявності в ній виділеної особливої групи. Кожен носій стратегії взаємодіє з кожним, включаючи себе згідно з ітеративною моделлю дилеми ув'язненого. Винагороди визначаються за матрицями виплат. Кожне наступне покоління популяції послідовно втрачає найбільш невідгідні стратегії поведінки попереднього покоління. Носії, що є членами особливої групи поводитиметься більш агресивно до «чужинців», і більше кооперується зі «своїми». Розглянуто різні варіанти початкового складу групи, а саме випадок, коли середня складність малої групи більша за середню складність популяції в цілому, менша, або порівняна з нею. Розглянуто варіанти з різним початковим розміром малої групи – 5, 10, 15, 20% від розміру популяції.

Результати. В роботі показано, як наявність малих груп різної складності та розміру впливає на динаміку популяції. Виявлено збільшення агресивності популяції за умови знаходження в ній малої групи будь-якого складу. Встановлено кількість та склад малої групи, за яких їй вдається захопити більшість в соціумі. Встановлено етап еволюції, на якому це відбувається.

Висновки. За результатами роботи визначено головні наслідки впливу особливих малих груп на еволюцію популяції.

Ключові слова: еволюція, популяція, суспільство, стратегія, особлива група, складність, агресивність, кооперація.

Як цитувати: Безрук Ю. Р., Яновський В. В. Еволюція популяції стратегій з пам'яттю з особливою малою групою. *Вісник Харківського національного університету імені В. Н. Каразіна, сер. «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*. 2023. вип. 57. С. 6-21.

<https://doi.org/10.26565/2304-6201-2023-57-01>

УДК 004.891

Розробка гібридної рекомендаційної системи

А.О. Белан, Л.В. Васильєва

Белан**Анжеліка Олександрівна***студентка**Харківський Національний Університет імені В. Н. Каразіна,**майдан Свободи 6, Харків, 61022**e-mail: anzhelika.bielan@student.karazin.ua**<https://orcid.org/0009-0006-5473-0430>***Васильєва****Лариса Валентинівна***к.б.н., доцент;**Харківський Національний Університет імені В. Н. Каразіна,**майдан Свободи 6, Харків, 61022**e-mail: lvvasilieva@karazin.ua**<https://orcid.org/0000-0001-7926-3062>*

Сучасний інтернет і веб-сервіси переповнені великою кількістю інформації, що стає складнішим для користувачів. Системи рекомендацій мають на меті вирішити проблему перевантаження інформацією, одночасно персоналізують досвід користувача, надають точні персоналізовані рекомендації відповідно до їхніх уподобань. Головною метою цієї роботи є реалізація системи рекомендацій для складної предметної сфери та алгоритму її впровадження для предметної області пошуку закладів послуг. У статті коротко описані основні підходи та алгоритми, які використовуються в системах рекомендацій, а також висвітлені області застосування, переваги та недоліки кожного з них. У роботі наведено опис реалізації системи рекомендацій для складної предметної сфери та алгоритму її впровадження для предметної області пошуку закладів сфери послуг. Особливістю системи, що розробляється, є важливість даних про геолокацію для формування рекомендацій. Розроблено алгоритм гібридної системи рекомендацій, що об'єднує в собі підходи до розробки систем рекомендацій на основі знань та фільтрації на основі вмісту. Перевагами описаного алгоритму є відсутність необхідності зберігати та використовувати у розрахунках інформацію про попередні сеанси користувачів, вирішення проблеми холодного запуску, формування актуальних рекомендацій для поточного сеансу користувача та вирішення проблеми визначення місцезнаходження для рекомендованого закладу. Запропонований підхід може бути використано для розробки та впровадження рекомендаційного алгоритму у складних предметних сферах, де дані про геолокацію важливі для надання рекомендацій.

Ключові слова: система рекомендацій, рекомендаційний алгоритм, гібридна система рекомендацій, фільтрація за вмістом, рекомендації на основі знань.

Як цитувати: Белан А.О., Васильєва Л.В. Розробка гібридної рекомендаційної системи. *Вісник Харківського національного університету імені В. Н. Каразіна, сер. «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*. 2023. вип. 57. С.22-31.

<https://doi.org/10.26565/2304-6201-2023-57-02>

1 Вступ

У сучасному світі з динамічним розвитком другого покоління мережевих сервісів інтернету WEB 2.0 кожен має доступ до надлишково великої кількості інформації. Користувачам все складніше впоратися з цим потоком даних та виділити для себе важливу інформацію. На інтернет-платформах електронної комерції, де пропонуються мільйони товарів, звичайному споживачеві складно охопити всі можливості, і це може призвести до перенасичення інформацією. Системи рекомендацій мають на меті вирішити цю проблему перевантаження інформацією, одночасно персоналізують досвід користувача, надають точні персоналізовані рекомендації щодо предметів/продуктів користувачам відповідно до їхніх уподобань. Метою системи рекомендацій є передбачення, чи буде продукт корисним для користувача на основі відомої про нього інформації.

Застосування систем рекомендацій (СР) в останні роки постійно зростає і вони використовуються у сферах роздрібною торгівлі та електронної комерції, таких як eBay, Amazon та Netflix. Ці компанії отримують величезну кількість даних користувачів і використовують СР для задоволення потреб користувачів і покращення бізнес-процесів. За оцінками McKinsey, 35% виручки Amazon і 75% Netflix припадає саме на рекомендовані товари, і цей відсоток, швидше за все, зростатиме [1].

На сьогоднішній день сфера розробки CR переживає значний розвиток. Такі системи використовуються поряд та разом з генеративним штучним інтелектом, що стрімко розвивається саме сьогодні. Однією з ключових тенденцій є використання машинного навчання і глибокого навчання, щоб створювати більш точні та ефективні рекомендації для користувачів [2]. Використання засобів обробки природної мови, а також збільшення вимог до якості даних і етичності алгоритмів є також надзвичайно важливими аспектами в розробці CR.

Персоналізація також стала важливим аспектом розробки CR, де моделі адаптуються до індивідуальних потреб і смаків користувачів на основі їхнього минулого взаємодії. CR використовують різноманітні дані користувачів для створення більш точних і персоналізованих рекомендацій. Ці дані включають в себе історію взаємодії користувачів на платформі, такі як перегляди, оцінки, покупки, які допомагають розуміти їхні вподобання та інтереси [3]. Демографічні дані, такі як вік, стать і місце проживання, допомагають категоризувати користувачів. Дані про геолокацію важливі для рекомендацій, пов'язаних з місцевими послугами [4]. Розробники продовжують експериментувати з новими методами і техніками, а також враховувати зростаючий інтерес до прозорості та етичних питань в роботі з даними.

2 Категорії систем рекомендацій

Системи рекомендацій можна класифікувати за кількома основними категоріями, залежно від способу їхньої роботи та особливостей алгоритмів, що використовуються [5].

2.1. Системи рекомендацій на основі вмісту (Content-Based Recommendation Systems)

Підходи, спрямовані на створення CR на основі вмісту, мають за мету побудувати профіль користувача, щоб передбачити рейтинги об'єктів, які користувач ще не оцінив. Методи, базовані на вмісті, використовують для цього теги та ключові слова. Оцінка корисності фільтрації на основі вмісту зазвичай проводиться за допомогою евристичних метрик, таких як косинус подібності. Цей підхід до рекомендацій може бути застосованим в ситуаціях, коли характеристики об'єктів можна легко визначити. Однак, він неефективний у випадках, коли вимагається складна і вручну введена інформація про об'єкти, особливо в разі великої кількості нових продуктів щодня.

Фільтрація на основі вмісту не потребує даних інших користувачів, оскільки рекомендації індивідуальні для кожного користувача. Це робить цей метод масштабованим для великої кількості користувачів і не залежним від їхніх дій. Системи на основі вмісту вимагають великих обсягів даних та глибоких знань у предметній області для точних рекомендацій. Крім того, вони мають обмежену здатність до виявлення нових об'єктів або інноваційних рекомендацій, оскільки вони опираються на характеристики інших оцінених об'єктів та профілі користувачів [7].

2.2. Системи рекомендацій на основі колаборативної фільтрації (Collaborative Filtering Recommendation Systems)

Колаборативна фільтрація це метод оцінки продуктів на основі рейтингів, які надають користувачі у історичних даних. Цей метод створює базу даних уподобань користувачів до об'єктів, шляхом виявлення сусідніх активних користувачів за подібними перевагами в покупках.

Методи колаборативної фільтрації поділяються на дві основні категорії: фільтрацію на основі схожості елементів (продуктів) та фільтрацію на основі схожості користувачів [6]. Методи, орієнтовані на користувача, складаються з двох основних кроків для прогнозування рейтингів товарів для конкретного користувача: пошук користувачів, подібних до цільового користувача та отримання оцінок від користувачів, схожих на активного користувача. Ці оцінки використовуються для генерації рекомендацій.

Існує багато алгоритмів колаборативної фільтрації, які обчислюють схожість між користувачами. Зазвичай використовуються міри подібності, такі як середньоквадратична різниця, кореляція Пірсона, косинус подібності, кореляція Спірмена та скоригована косинус подібність [6]. Колаборативна фільтрація є популярним вибором для CR і не вимагає глибокого розуміння домену, оскільки вбудовується та розраховується автоматично.

CR на основі колаборативної фільтрації мають свої недоліки, включаючи проблеми холодного запуску через нестачу даних для нових користувачів і об'єктів, розрідженість даних оцінок у великих системах, складності масштабування при зростанні кількості користувачів та об'єктів.

2.3. Системи рекомендацій на основі демографічних даних (Demographic-Based Recommendation Systems)

Результати різних кількісних досліджень демонструють, що методи спільної фільтрації можуть бути покращені за допомогою демографічної кореляції. SR на основі демографії здатні генерувати рекомендації, класифікуючи користувачів за їхніми демографічними характеристиками. Ці системи використовують атрибути користувачів, такі як вік, стать, мова та інші, для надання рекомендацій [8].

Основною перевагою демографічних фільтрів SR є швидкість та простота отримання результатів на основі обмеженого обсягу спостережень. SR на основі демографії спрямовані на вирішення проблем масштабованості та холодного запуску. Вони стають особливо корисними тоді, коли обмежена кількість інформації про продукт. Проте ці методи мають свої недоліки, такі як проблеми зі збором та збереженням конфіденційної інформації користувачів та недостатня гнучкість у врахуванні зміни уподобань користувачів, що може вплинути на стабільність та ефективність рекомендаційної системи.

2.4. Системи рекомендацій на основі корисних програм (Utility-Based Recommendation Systems)

SR на основі корисних програм пропонують рекомендації на підставі розробки корисної моделі для кожного елемента, спрямованої на задоволення потреб користувача. Ці системи створюють багатоатрибутні функції корисності для користувачів та рекомендують елементи на основі явно обчисленої корисності кожного з них. SR на основі утиліт важливі, оскільки вони можуть враховувати не лише характеристики продуктів, але й інші атрибути, такі як доступність товарів та надійність постачальників. Вони здатні генерувати обчислення корисності в реальному часі, що дозволяє оцінити стан товару для користувача. Системи на основі утиліт не базуються на довгострокових узагальненнях про користувачів, а враховують актуальні потреби користувачів та доступні варіанти. Недоліком таких систем є те, що система може сховати рекомендації від користувачів, навіть якщо вони відповідають вподобанням цього користувача, якщо описові характеристики товарів обмежені [9].

2.5. Системи рекомендацій на основі знань (Knowledge-Based Recommendation Systems)

SR на основі знань використовують явні знання про продукти та користувачів для створення критерію на основі знань з метою розробки рекомендацій. На відміну від інших SR, вони не залежать від великої кількості статистичних даних про конкретні оцінені товари чи конкретних користувачів. Системи на основі знань мають декілька переваг. Наприклад, вони можуть уникнути проблеми нарощування, що часто виникає в методах машинного навчання для рекомендацій. Зазвичай зразкові системи не можуть навчатися, доки користувач не надає багато оцінок. SR на основі знань уникають цієї проблеми, оскільки їх рекомендації не залежать від історії оцінок користувача. Крім того, вони не вимагають збирання конкретних даних про користувача, оскільки рекомендації також не підпорядковані індивідуальним смакам користувача. З цих причин системи на основі знань цінні як самостійні SR і можуть доповнювати інші типи.

Одним з основних недоліків систем на основі знань є потенційна обмеженість в питанні отримання знань, яка виникає внаслідок явного визначення рекомендаційних знань. Процес отримання знань передбачає створення правил і вимог, які необхідні для системи на основі знань, і це здійснюється шляхом надання знань на основі правил, об'єктів і онтологій, побудованих на фреймах. Теорії Бейтсона були використані для керування процесом подальшого вивчення отримання знань [10].

2.6. Гібридні системи рекомендацій (Hybrid-Based Recommendation Systems)

Гібридні SR представляють собою підхід до розробки систем, який поєднує в собі різні методи та підходи з метою покращити якість та ефективність рекомендацій. Ці системи поєднують в собі два або більше основних типи SR, такі як колаборативна фільтрація, фільтрація на основі вмісту, системи на основі корисних програм, базовані на знаннях та інші. Головна мета гібридних систем це компенсація обмежень та недоліків кожного окремого методу, створюючи стійку та більш точну SR [11].

Однією з ключових переваг гібридних систем є здатність знаходити підхід до різних видів даних та користувачів. Наприклад, якщо один метод заснований на спільній фільтрації добре

працює для користувачів з великою історією оцінок, а інший метод на основі вмісту вдається для нових користувачів, гібридна система може комбінувати їхні сильні сторони для надання кращих рекомендацій незалежно від користувача.

Створення гібридних систем може бути складним завданням, оскільки потребує інтеграції різних методів та визначення оптимального способу комбінування їх результатів. Важливо також вирішити питання вагомості різних компонентів гібридної системи та розробити ефективні алгоритми комбінування рекомендацій. Гібридні СР стають є найбільш популярними завдяки їхній здатності покращувати точність та роботу СР, особливо в умовах великого обсягу даних та різноманітності користувачів.

3 Практична реалізація системи рекомендацій

Головною метою цієї роботи є реалізація системи рекомендацій для складної предметної сфери та алгоритму її впровадження для предметної області пошуку закладів послуг. Особливістю системи, що розроблюється, є можливість отримання даних про місцезнаходження користувача та доступність геоданих закладів послуг, які зберігаються в базі даних системи. Для дослідження було запропоновано виконати аналіз можливих шляхів впровадження рекомендацій у систему, виділити основні аспекти та ідеї на основі яких буде розроблена СР.

Важливим аспектом розробки алгоритму рекомендацій є визначення етапу взаємодії з системою, коли користувачу будуть надаватись рекомендації. Послідовність дій, які користувач може виконати в системі зображено на діаграмі діяльності (рис. 1). Для даної предметної області впровадження рекомендацій може виконуватись на етапі початкового перегляду доступних закладів, під час вибору параметрів для пошуку та при перегляді деталей про обраний заклад. Причому, підходи до розробки алгоритму рекомендацій можуть сильно різнитися у залежності від етапу взаємодії користувача з системою, на якому будуть надаватись рекомендації.

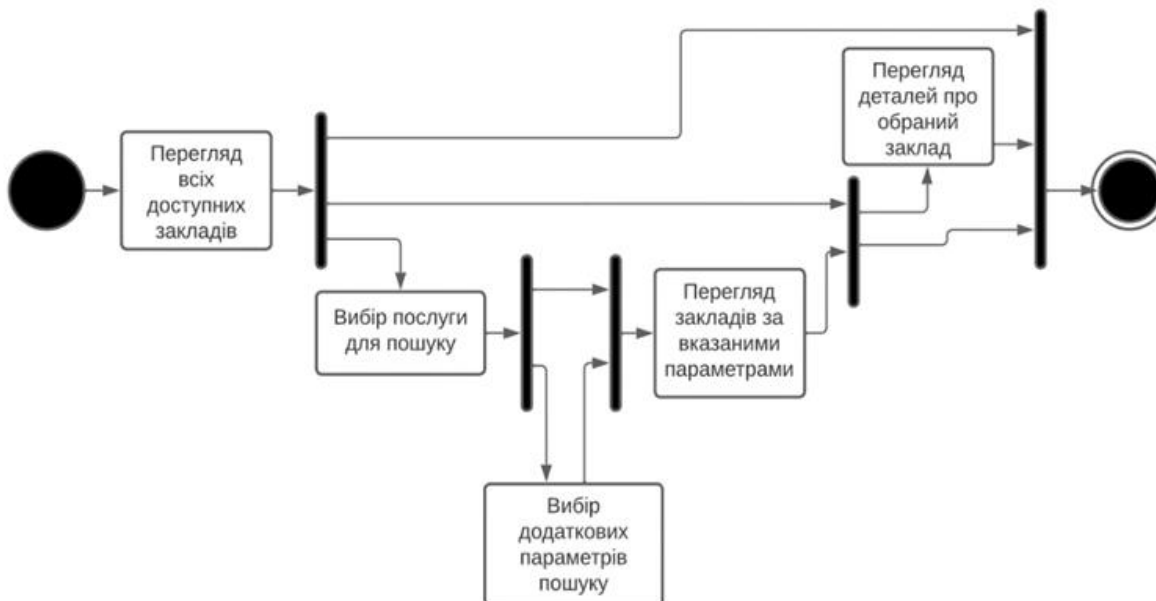


Рис.1 Діаграма активності

На етапі входу в систему та можливості перегляду всіх доступних закладів на карті користувачу можуть пропонуватися заклади за його місцезнаходженням, які надають послуги, що були оцінені користувачем у попередніх сесіях. У випадку, коли користувач вперше взаємодіє з системою, йому можуть пропонуватися найбільш популярні заклади та послуги, які вони надають, за його місцезнаходженням.

Після вибору користувачем послуги та додаткових параметри для пошуку найбільш доцільно рекомендувати заклади та послуги, що найкраще відповідають заданим параметрам, у тому числі тоді, коли в базі даних системи відсутні заклади та послуги, які повністю відповідають вказаними параметрам.

При наданні рекомендацій на етапі перегляду користувачем деталей про заклад, що надає шукану ним послугу, мають враховуватися параметри закладу та послуги. У цьому випадку для формування рекомендацій, на відміну від попередніх випадків, більше значення мають дані про місцезнаходження переглянутого закладу, ніж про місцезнаходження користувача.

Для дослідження пропонується розглянути розробку алгоритму роботи СР, впроваджену на етапі пошуку закладів послуг за вказаними параметрами та перегляду деталей про обрані заклади. Розроблений алгоритм рекомендацій має вирішувати проблеми холодного запуску та відсутності в базі даних системи об'єктів, що повністю відповідають вказаним параметрам пошуку, а також проблему визначення місцезнаходження для рекомендації.

З проаналізованих основних підходів до розробки рекомендаційних систем можна визначити, що вдало побудовані гібридні СР справляють з задачею надання рекомендацій найкраще. Отже, для вирішення проблеми відсутності в базі даних системи об'єктів, що повністю відповідають вказаним параметрам пошуку, застосовується підхід до розробки СР на основі вмісту, який враховує та аналізує характеристики закладу та послуги, які користувач шукає або переглядає. На першому етапі роботи алгоритму користувачу пропонуються об'єкти, які є найбільш схожими до так званої «ідеальної точки». Ідеальною точкою буде вважатися об'єкт із характеристиками сформованими на основі параметрів пошуку та місцезнаходження користувача або ж об'єкт, який користувач переглядає в даний момент. Для визначення «схожості» між об'єктами використовуються показники подібності для порівняння атрибутів цих об'єктів. Визначаються дві основні речі для кожного атрибута: його вага у відношенні до інших атрибутів і формула або метод, за допомогою якого оцінюється схожість між двома значеннями цього атрибута.

Для вирішення проблеми холодного запуску застосовується підхід розробки СР на основі знань, який полягає в формуванні списку вподобань користувача в рамках поточного сеансу застосування параметрів фільтрації. Так звана ідея «діалогу» застосовується на другому етапі роботи алгоритму та полягає в тому, що після визначення параметрів пошуку та перегляду певної кількості пропозицій рекомендації будуть базуватися на групі з деякої кількості останніх переглянутих закладів та послуг, які вони надають.

Крім того, для рекомендаційного алгоритму вводяться обмеження, які зменшують простір об'єктів, що можуть бути рекомендовані користувачу. Оскільки основною метою системи є пошук користувачем закладів, які надають послуги, що його цікавлять, то не доцільно рекомендувати йому заклади, які не надають послуг цієї сфери. Відповідний підхід стосується і міста, де користувач шукає заклади, що надають відповідну послугу.

Математичні основи даного алгоритму включають у себе наступне. Для кількісної оцінки схожості деякого об'єкта з еталонним використовується метод зваженої суми критеріїв. Кожен об'єкт представлено вектором критеріїв:

$$K = \{k_1, k_2, k_3 \dots k_n\} \quad (1)$$

де k_i - кількісна оцінка схожості i -ої характеристики об'єкта з i -ою характеристикою еталонного елемента.

Відносна важливість кожної з характеристик виражена вектором вагових коефіцієнтів, сума яких рівна одиниці:

$$W = \{w_1, w_2, w_3 \dots w_n\} \quad (2)$$

де w_i - відносна важливість i -ої характеристики об'єкта.

Результуюче значення схожості об'єкту розраховується за формулою:

$$Score = \sum w_i \cdot k_i \quad (3)$$

Для оцінки на основі групи елементів, оцінки (3) нового елемента, що розраховуються відносно кожного елемента у межах групи переглянутих, підсумовуються між собою для отримання кінцевого результату:

$$Score^* = \sum_{i=1}^N Score_i \quad (4)$$

де N – кількість переглянутих елементів.

Для оцінки схожості характеристик об'єкта з характеристиками еталонного об'єкта, представленими у вигляді інтервальних оцінок, використовуються функції затухання (або функції розподілу). Ці функції оцінюють певну величину, зменшуючи своє значення залежно від збільшення відстані між значенням цієї величини та певним заданим джерелом (центром мас). Ця оцінка подібна до запиту щодо включення деякого значення в заданий діапазон, з подальшим призначенням кількісної оцінки цьому значенню в залежності від близькості до центра діапазону.

Існують три основні форми функцій затухання (розподілу), які використовуються:

- Гауссіан (за функцією Гауса);
- експоненціальний розподіл;
- лінійний розподіл.

У розробці рекомендаційної системи використовується гаусівська залежність, оскільки лінійна залежність є занадто простою та приводить до однакового зменшення оцінок, при віддаленні від центрального значення, а експоненціальна залежність має дуже різкий графік зменшення значень біля центру та менш різкий за його межами. У даному випадку потрібна зворотна залежність, яку надає розподіл Гауса.

Затухання для характеристики об'єкта визначається наступним чином:

$$K_i = \exp \left(- \frac{(|value_i - origin| - offset)^2}{2 \cdot \left(- \frac{scale^2}{2 \cdot \ln(decay)} \right)} \right) \quad (4)$$

де K_i – кількісна оцінка схожості i -тої характеристики об'єкта з i -тою характеристикою еталонного об'єкта, приймає значення в діапазоні $[0;1]$;

$value_i$ – значення i -тої характеристики об'єкта;

$origin$ – значення i -тої характеристики еталонного об'єкта;

$offset$ – відстань, на якій значення характеристики буде вважатися рівним еталонній;

$scale$ – відстань від центру (+ зміщення) яка визначає діапазон прийнятної (допустимої) схожості;

$decay$ – значення схожості в яке оцінюється значення характеристики на відстані $scale$ від $origin$.

Для характеристик, які представлені на номінальних шкалах, ми можемо визначити лише подібність або різницю між об'єктами. Таким чином, розрахунок подібності для таких характеристик буде виглядати наступним чином:

– $K_i = 1$, якщо значення відповідних характеристик однакові, що вказує на повне співпадіння значень;

– $K_i = 0$, якщо значення відповідних характеристик різні, що вказує на повну відсутність співпадіння, і це оцінюється нульовим балом.

Якщо значення ж певної характеристики відсутнє, то оцінка схожості цієї характеристики між об'єктом і еталонним об'єктом буде $K_i = 0$.

Блок-схему описаного алгоритму зображено на рисунку 2. Відображення результатів роботи СР відбувається після застосування користувачем фільтрації, що передбачає вибір послуги для пошуку та інших параметрів, а також при перегляді детальної інформації про заклади, які є результатом пошуку.

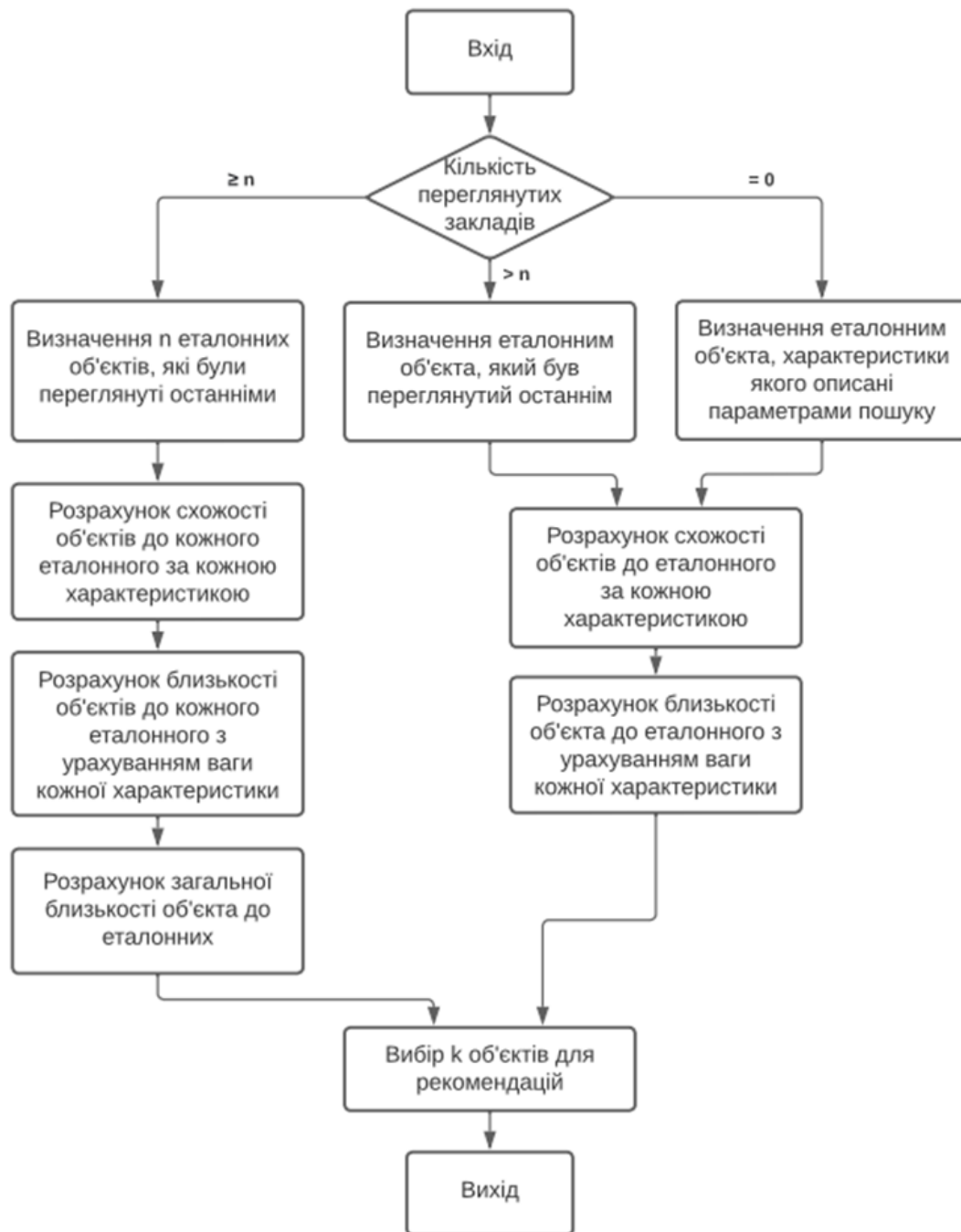


Рис.2 Блок-схема роботи алгоритму рекомендаційної системи

Спочатку алгоритм перевіряє кількість закладів, які переглянув користувач після застосування пошуку. У результаті перевірки виконується один із сценаріїв відповідно до варіантів:

- кількість переглянутих закладів є рівною нулю;
- кількість переглянутих закладів є меншою за встановлене порогове значення;
- кількість переглянутих закладів є більшою за встановлене порогове значення.

Якщо виконується перша умова, то еталонним вважається об'єкт, характеристиками якого є параметри фільтрації, а локація визначається місцезнаходженням користувача. Якщо виконується друга умова, то еталонним вважається останній переглянутий користувачем об'єкт. Для обох випадків за допомогою формул (3) і (5) обчислюються оцінки близькості інших об'єктів до вибраного центру, і вибирається k найбільш відповідних.

У випадку, коли кількість переглянутих закладів є більшою за встановлене порогове значення, останні n переглянутих оголошень вважаються «ідеальною» точкою. Потім за допомогою формул (3) і (5) обчислюються оцінки близькості інших об'єктів до кожного вибраних еталонних об'єктів. Після чого за допомогою формули (4) обчислюється остаточна оцінка близькості відносно вибраного центру, і вибирається k найбільш відповідних.

У експериментальній частині даної роботи проводиться пошук таких параметрів для розрахунків, які найкраще передбачають вподобання користувачів. Результати рекомендацій порівнюються евристично для визначення показників, які найкраще відповідають потребам користувачів. Підбір цих параметрів відбувається через аналіз правил, які діють у предметній області, а також через врахування уявлень середнього користувача. Важливо враховувати, які критерії важливі для такого користувача під час пошуку рекомендацій.

Розроблена СР для предметної області системи пошуку закладів сфери послуг використовує наступні характеристики (критерії) та їх важливість:

- сфера послуг;
- адміністративний район;
- послуга – 0.3;
- тип закладу – 0.05;
- вартість надання послуги – 0.2;
- відстань до еталонного закладу – 0.3;
- рейтинг закладу – 0.15.

Для розрахунку близькості характеристик за формулою 1.5 значення параметрів представлено в таблиці (1).

Таблиця 1. Значення параметрів розрахунку оцінки близькості

Назва критерію, одиниці вимірювання	Назва параметра		
	offset	scale	decay
Вартість, грн	5%	15%	0.5
Відстань, км	0.5	3	0.5
Рейтинг	0	1	0.5

Тестування розробленого алгоритму рекомендацій показало, що він здатен створювати релевантні рекомендації, що відповідають запитам користувачів. Алгоритм ефективно визначає бажане місцезнаходження закладу та враховує історію перекладів. Успішне тестування підкреслює готовність алгоритму до використання в реальних умовах.

4 Висновки

У роботі проаналізовано реалізацію системи рекомендацій для складної предметної сфери та алгоритм її впровадження, розглянуто різні категорії систем рекомендацій та особливості алгоритмів їх побудови, області застосування, переваги та недоліки. Для складної предметної області системи пошуку закладів сфери послуг реалізовано та впроваджено систему рекомендацій гібридного типу.

Розроблений алгоритм об'єднує в собі підходи до розробки систем рекомендацій на основі знань та фільтрації на основі вмісту. Перевагами описаного алгоритму є відсутність необхідності зберігати та використовувати у розрахунках інформацію про попередні сеанси користувачів, вирішення проблеми холодного запуску та відсутності в базі даних системи об'єктів, що повністю відповідають вказаним параметрам пошуку, формування актуальних рекомендацій для поточного сеансу користувача та вирішення проблеми визначення місцезнаходження для рекомендованого закладу.

Запропонований підхід може бути використано для розробки та впровадження рекомендаційного алгоритму у складних предметних сферах, де дані про геолокацію важливі для надання рекомендацій.

ЛІТЕРАТУРА

1. I. MacKenzie, C. Meyer, S. Noble, How retailers can keep up with consumers. McKinsey & Company, 2013. URL: <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers> (Last accessed: 29.08.2023).

2. C. C. Aggarwal, An Introduction to Recommender Systems. In *Recommender Systems*; Springer: Cham, Switzerland, 2016. URL: <https://www.springer.com/gp/book/9783642031205> (Last accessed: 30.08.2023).
3. Yang X., Guo Y., Liu Y., Steck H., A survey of collaborative filtering based social recommender systems. *Comput. Commun.* 2014, 41, pp. 1–10. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0140366413001722> (Last accessed: 30.08.2023).
4. Ding Z., Li X., Jiang C, Zhou M., Objectives and state-of-the-art of location-based social network recommender systems. *ACM Comput. Surv. CSUR* 2018, 51, pp. 1–28. URL: <https://dl.acm.org/doi/abs/10.1145/3154526> (Last accessed: 05.09.2023).
5. Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, R. Kashef, Recommendation Systems: Algorithms, Challengers, Metrics, and Business. *Appl. Sci.*, 2020. URL: <https://www.mdpi.com/2076-3417/10/21/7748#B10-applsci-10-07748> (Last accessed: 05.09.2023).
6. B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms. In *WWW '01, Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001*; Association for Computing Machinery: New York, NY, USA, 2001; Volume 1, pp. 285–295. URL: <https://dl.acm.org/doi/abs/10.1145/371920.372071> (Last accessed: 05.09.2023).
7. J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 291–324. URL: https://link.springer.com/chapter/10.1007/978-3-540-72079-9_9 (Last accessed: 07.09.2023).
8. Al-Shamri, M.Y.H., User profiling approaches for demographic recommender systems. *Knowl. Based Syst.* 2016, 100, pp. 175–187. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0950705116001192> (Last accessed: 10.09.2023).
9. Deng F., Utility-based recommender systems using implicit utility and genetic algorithm. In *Proceedings of the 2015 International Conference on Mechatronics, Electronic, Industrial and Control Engineering (MEIC-15), Shenyang, China, 1–3 April 2015*; Atlantis Press: Amsterdam, The Netherlands, 2015. URL: <https://www.atlantispress.com/proceedings/meic-15/19830> (Last accessed: 11.09.2023).
10. R. Burke, Knowledge-based recommender systems. In *Encyclopedia of Library and Information Systems*; CRC Press: Boca Raton, FL, USA, 2000; Volume 69, (Suppl. 32), pp. 175–186. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=dc133144d431fc3b75c8de27f6b21da6eb5bc1b> (Last accessed: 11.09.2023).
11. A.A. Kardan, M. Ebrahimi, A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups. *Inf. Sci.* 2013, 219, pp. 93–110. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0020025512004756> (Last accessed: 18.09.2023).

REFERENCES

1. MacKenzie, C. Meyer, S. Noble, How retailers can keep up with consumers. McKinsey & Company, 2013. URL: <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers> (Last accessed: 29.08.2023).
2. C. C. Aggarwal, An Introduction to Recommender Systems. In *Recommender Systems*; Springer: Cham, Switzerland, 2016. URL: <https://www.springer.com/gp/book/9783642031205> (Last accessed: 30.08.2023).
3. Yang X., Guo Y., Liu Y., Steck H., A survey of collaborative filtering based social recommender systems. *Comput. Commun.* 2014, 41, pp. 1–10. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0140366413001722> (Last accessed: 30.08.2023).
4. Ding Z., Li X., Jiang C, Zhou M., Objectives and state-of-the-art of location-based social network recommender systems. *ACM Comput. Surv. CSUR* 2018, 51, pp. 1–28. URL: <https://dl.acm.org/doi/abs/10.1145/3154526> (Last accessed: 05.09.2023).
5. Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, R. Kashef, Recommendation Systems: Algorithms, Challengers, Metrics, and Business. *Appl. Sci.*, 2020. URL: <https://www.mdpi.com/2076-3417/10/21/7748#B10-applsci-10-07748> (Last accessed: 05.09.2023).

6. B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms. In WWW '01, Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; Association for Computing Machinery: New York, NY, USA, 2001; Volume 1, pp. 285–295. URL: <https://dl.acm.org/doi/abs/10.1145/371920.372071> (Last accessed: 05.09.2023).
 7. J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems. In The Adaptive Web; Springer: Berlin/Heidelberg, Germany, 2007; pp. 291–324. URL: https://link.springer.com/chapter/10.1007/978-3-540-72079-9_9 (Last accessed: 07.09.2023).
 8. Al-Shamri, M.Y.H., User profiling approaches for demographic recommender systems. Knowl. Based Syst. 2016, 100, pp. 175–187. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0950705116001192> (Last accessed: 10.09.2023).
 9. Deng F., Utility-based recommender systems using implicit utility and genetic algorithm. In Proceedings of the 2015 International Conference on Mechatronics, Electronic, Industrial and Control Engineering (MEIC-15), Shenyang, China, 1–3 April 2015; Atlantis Press: Amsterdam, The Netherlands, 2015. URL: <https://www.atlantispress.com/proceedings/meic-15/19830> (Last accessed: 11.09.2023).
 10. R. Burke, Knowledge-based recommender systems. In Encyclopedia of Library and Information Systems; CRC Press: Boca Raton, FL, USA, 2000; Volume 69, (Suppl. 32), pp. 175–186. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=dc133144d431fc3b75c8de27f6b21da6eb5bc1b> (Last accessed: 11.09.2023).
- A.A. Kardan, M. Ebrahimi, A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups. Inf. Sci. 2013, 219, pp. 93–110. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0020025512004756> (Last accessed: 18.09.2023).

Development of a hybrid-based recommendation system

A.O. Bielan, L.V. Vasilieva

Bielan Anzhelika

Student

*V. N. Karazin Kharkiv National University
6 Svobody Sq., Kharkiv, Ukraine, 61022*

Vasilieva Larisa

Candidate of Biological Sciences, Associate Professor

*V. N. Karazin Kharkiv National University
6 Svobody Sq., Kharkiv, Ukraine, 61022*

Modern Internet and web services are flooded with a vast amount of information, which is becoming more challenging for users. Recommendation systems aim to address this information overload issue while personalizing the user experience by providing precise, personalized recommendations based on their preferences. The main goal of this work is implementing a recommendation system for a complex subject area and the algorithm of its integration into the sector of searching for the service providers. The article briefly outlines the main approaches and algorithms used in recommendation systems, highlighting their areas of application, advantages, and disadvantages, as well as description of the implementation of a recommendation system for a complex subject area and the algorithm for its integration into the sector of searching for the service providers. One distinctive feature of the system under development is the importance of geolocation data for the generation of recommendations. An algorithm for a hybrid recommendation system that combines knowledge-based and content-based filtering approaches has been developed. The advantages of the described algorithm include the absence of the need to store and use information about previous user sessions in the calculations, as well as addressing cold-start issues, generating real-time recommendations for the current user session, and solving the problem of determining the location for recommended establishments. The proposed approach can be used for developing and implementing recommendation algorithms in complex subject areas where geolocation data is crucial for providing recommendations.

Keywords: *recommendation system, recommendation algorithm, hybrid-based recommendation system, content-based filtering, knowledge-based recommenders*

How to quote: Bielan A., Vasilieva L., “Development of a hybrid-based recommendation system”, *Bulletin of V.N. Karazin Kharkiv National University, series “Mathematical modelling. Information technology. Automated control systems*, vol. 57, pp. 22-31, 2023.

<https://doi.org/10.26565/2304-6201-2023-57-02>

УДК (UDC) 004.051

**Зінов'єв Дмитро
Володимирович**

*старший викладач кафедри моделювання систем і технологій
Харківський національний університет імені В. Н. Каразіна, майдан
Свободи 4, м. Харків, Україна, 61022
e-mail: zinoviev@karazin.ua
<https://orcid.org/0000-0003-1862-9803>*

**Ткачук Микола
Вячеславович**

*д.т.н., професор; професор кафедри моделювання систем і
технологій
Харківський національний університет імені В. Н. Каразіна, майдан
Свободи 4, м. Харків, Україна, 61022
e-mail: mykola.tkachuk@karazin.ua
<https://orcid.org/0000-0003-0852-1081>*

Аналіз, класифікація та тестування інструментальних засобів для управління конфігураціями програмних мікросервісів

Актуальність. Розробка програмних застосунків з мікросервісною архітектурою (МСА) потребує вирішення проблем їх проектування, інтеграції, масштабування, адаптації, надійності, відмовостійкості та, в цілому, підвищення показників якості таких систем, і тому питання ефективного управління конфігураціями МСА є актуальною науково-технічною задачею.

Мета. Метою цього дослідження є аналіз функціональних особливостей існуючих інструментальних засобів для конфігурування МСА, а також їх класифікація та тестування, що уможливило їх вдосконалення шляхом розробки модельно-технологічних рішень для забезпечення адаптивного управління в таких застосунках.

Методи дослідження. Для досягнення мети дослідження визначені базові поняття та сутність процесів управління конфігураціями МСА, проведено аналіз функціональних можливостей деяких сучасних інструментальних засобів для конфігурування МСА та побудована їх класифікація. Для подальшого розгляду мотивовано обрано фреймворк Microconfig.io, програмно реалізовано конкретний приклад відповідного МСА застосунку, проведено його тестування та проаналізовано отримані результати.

Результати. Зроблено обґрунтований висновок про можливість та доцільність підвищення ефективності інструментального засобу MicroConfig.io шляхом його використання у складі перспективної інформаційної технології адаптивного управління процесом конфігурування МСА, для якої розроблена UML діаграма розміщення компонентів.

Висновки. Проаналізовані функціональні особливості сучасних інструментальних засобів для конфігурування МСА, побудована їх можлива класифікація та проведено програмне тестування однієї з типових таких систем, а саме, фреймворку MicroConfig.io. Запропоновано перспективна інформаційна технологія адаптивного управління процесом конфігурування МСА у вигляді компонентної діаграми та сформульовані напрямки подальших досліджень.

Ключові слова: програмні мікросервіси, управління конфігураціями, класифікація, тестування, адаптація, якість, метрика, інформаційна технологія, UML, діаграма компонентів, модель, засіб

Як цитувати: Зінов'єв Д.В., Ткачук М.В. Аналіз, класифікація та тестування інструментальних засобів для управління конфігураціями програмних мікросервісів. *Вісник Харківського національного університету імені В.Н.Каразіна, сер. «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління».* 2023. вип. 57. С. 32-41. <https://doi.org/10.26565/2304-6201-2023-57-03>

1 Вступ. Мета та актуальність дослідження

Сучасні процеси, що пов'язані з життєвим циклом програмних застосунків зокрема з мікросервісною архітектурою (МСА), ставлять перед розробниками питання щодо їх проектування, інтеграції, масштабування, адаптації, надійності, відмовостійкості та, в цілому, підвищення показників якості таких систем. Зростання функціональності, великі об'єми даних, вимоги до продуктивності, інтеграція з іншими системами, високий темп розвитку технологій та багато інших факторів значно ускладнюють процеси розробки, впровадження та підтримки програмного забезпечення (ПЗ) [1, 2].

Для вирішення багатьох з цих проблем при розробці використовуються методи та технології управління конфігураціями програмного продукту (Configuration Management). На даний час існує

декілька типів інструментальних засобів для управління конфігураціями ПЗ, це, наприклад системи [3, 4]: контролю версій (Version Control System); автоматизованого розгортання (Deployment Automation); управління залежностями (Dependency Management); конфігурації (Configuration Management); управління змінами (Change Management) та інш. Вони допомагають ефективно керувати різними аспектами програмних продуктів, включаючи версіонування, розгортання, масштабування, залежності, конфігурацію та контроль змін. Ці методики та інструменти досить давно використовуються для підвищення якості ПЗ із монолітною архітектурою. Для розподілених систем, а саме для ПЗ з мікросервісною архітектурою, де застосунки розгортаються, як набір незалежних компонентів, ефективне управління конфігураціями стає ключовим аспектом забезпечення стабільності, масштабованості та надійності систем [5].

Питання ефективності використання інструментів для УКПП для розподілених систем на базі МСА ще до кінця не вивчені. Тому основною задачею цього дослідження є аналіз функціональних особливостей існуючих інструментальних засобів для конфігурування програмних мікросервісів, а також їх класифікація та тестування, з метою подальшого вдосконалення шляхом розробки модельно-технологічних рішень для забезпечення можливостей адаптивного управління в таких застосунках.

2 Основні поняття, функціональність та класифікація інструментальних засобів управління конфігураціями мікросервісів

2.1 Управління конфігураціями програмного забезпечення

Управління конфігураціями ПЗ включає такі основні аспекти:

- управління версіями, яке забезпечує відстеження і контроль за різними версіями компонентів програмного продукту, включи збереження вихідного коду, бібліотек, конфігураційних файлів і документації для кожної версії;
- управління змінами, яке визначає процес заявки, оцінки, впровадження та відстеження змін, що вносяться до програмного продукту, включи управління проблемами, заявками на виправлення помилок, нововведеннями і іншими змінами;
- конфігураційне управління, яке встановлює і підтримує правильну конфігурацію програмного продукту та його компонентів, включи управління залежностями, конфігураційними файлами, налаштуваннями середовища і іншими конфігураційними елементами [2];
- забезпечення якості, до якого включені процеси тестування, перевірки та перевірки програмного забезпечення для забезпечення відповідності вимогам, стандартам і очікуванням;
- відстеження залежностей, що забезпечує відстеження залежностей між різними компонентами програмного продукту;
- управління розгортанням, що визначає процес підготовки і розгортання програмного забезпечення в різних середовищах, включаючи тестове і виробниче середовища та забезпечує контрольоване і надійне розгортання змін та версій програмного продукту.
- аудит і звітність, де забезпечуються здатність відстежувати всі зміни, вносити аудит і генерувати звіти про конфігурацію, версії та зміни в програмному продукті;
- інтеграцію з іншими процесами, що сприяє ефективному функціонуванню і успішному виконанню проекту.

2.2 Конфігураційні файли та формати

Управління конфігураціями програмного забезпечення включає в себе використання конфігураційних файлів та форматів для збереження налаштувань та параметрів програми. У роботі розглянуті деякі з популярних конфігураційних файлів, а саме:

- YAML (YAML Ain't Markup Language) – текстовий формат, який має простий для читання синтаксис, що дозволяє зберігати дані у вигляді списків, словників та примітивних типів даних [6];
- JSON (JavaScript Object Notation) – легкий текстовий формат обміну даними, який широко використовується для конфігураційних файлів і базується на синтаксисі JavaScript та дозволяє зберігати дані у вигляді пар "ключ-значення" [7];

- XML (Extensible Markup Language) – є розширюваним мовним форматом розмітки, який використовує власну синтаксичну структуру з відкриваючими та закриваючими тегами для представлення даних [8];
- INI (Initialization) файли – є простими текстовими файлами, що використовуються для зберігання конфігураційних даних у вигляді секцій та параметрів, де кожна секція містить набір ключів-значень;
- Properties файл – формат властивостей, часто використовуваний у Java-проектах, є простим текстовим форматом, в якому дані зберігаються у вигляді пар "ключ=значення";
- TOML (Tom's Obvious, Minimal Language) – є простим форматом конфігураційних файлів, який нагадує INI-файли, але має більш розширений синтаксис та підтримку більш складених структур даних;
- HCL (HashiCorp Configuration Language) – використовується в інструментах інфраструктури коду, таких як Terraform, для опису конфігурації ресурсів. HCL має простий та декларативний синтаксис, який легко зрозуміти і підтримує розширення;
- ENV (Environment Variables) – ENV-змінні зазвичай використовуються для передачі конфігураційних параметрів через зовнішнє середовище, наприклад, операційну систему або контейнер. Це гнучкий підхід, оскільки дозволяє легко змінювати параметри без зміни конфігураційного файлу;
- конфігурація бази даних оскільки деякі програми зберігають конфігураційні параметри безпосередньо в базі даних, де можна зберігати та керувати параметрами за допомогою запитів SQL або ORM (об'єктно-реляційне відображення).

2.3 Класифікація інструментальних засобів для управління конфігураціями програмних застосунків

На рисунку 2.1 наведено схему класифікації інструментальних засобів які можна використовувати для управління конфігураціями мікросервісів.

Інструменти контролю версій, такі як Git, Subversion та Mercurial, дозволяють відстежувати зміни в програмному забезпеченні та його компонентах, зберігати історію змін, відстежувати версії коду та спільно працювати над проектом у команді [9-11]. Ці інструменти допомагають ефективно керувати розробкою та забезпечувати зручну спільну роботу над програмними мікросервісами.

Інструменти, такі як Ansible, Puppet, Chef, Terraform та Kubernetes, надають засоби для автоматизації процесів розгортання та управління інфраструктурою програмних мікросервісів [12, 13]. Вони дозволяють швидко та надійно розгортати сервіси, керувати їх налаштуваннями та забезпечувати масштабованість та відновлюваність.

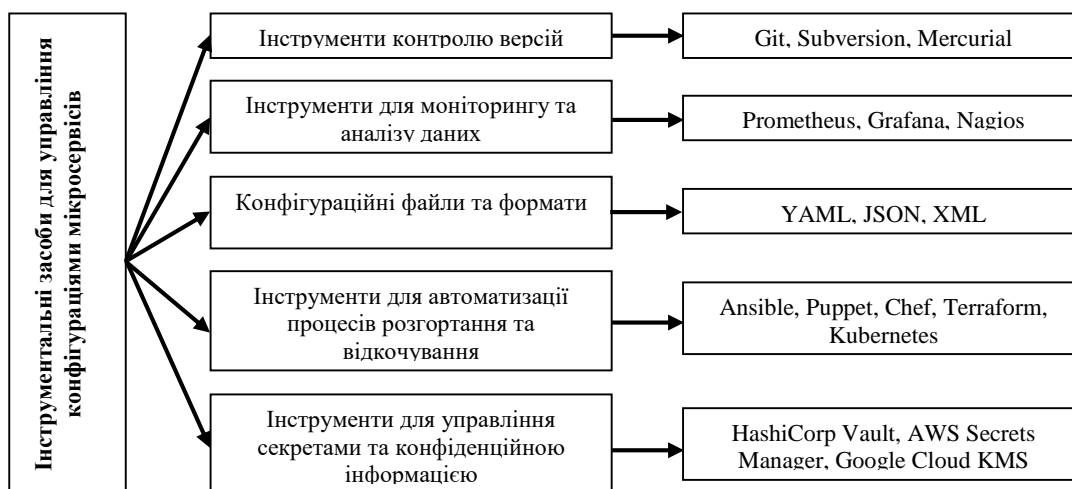


Рис. 2.1 Класифікація інструментальних засобів управління конфігураціями ПП

Інструменти Prometheus, Grafana та Nagios дозволяють здійснювати моніторинг та аналіз стану програмних мікросервісів [14, 15]. Вони забезпечують збір, візуалізацію та аналіз метрик,

журналів та інших даних, що дозволяє оперативно виявляти проблеми та оптимізувати роботу системи.

Інструменти HashiCorp Vault, AWS Secrets Manager, Google Cloud KMS допомагають забезпечувати безпеку та управління секретами, такими як паролі, ключі API та інша конфіденційна інформація, яка використовується в програмних мікросервісах [16, 17]. Вони дозволяють зберігати, керувати та забезпечувати безпеку доступу до цих секретів.

Конфігураційні файли використовуються для збереження налаштувань та параметрів програмного забезпечення або його компонентів. Формати, такі як YAML, JSON або XML, часто використовуються для представлення цих даних у конфігураційних файлах.

3 Тестування та аналіз результатів проведених програмних експериментів

3.1 Мотивований вибір та опис функціональних можливостей інструментального засобу Microconfig.io

Для вибору інструменту для управління конфігураціями програмних мікросервісів з урахуванням потреб і вимог досліджуваної системи були визначені наступні критерії: функціональні можливості, інтеграція з іншими інструментами, простота використання, розширюваність, гнучкість, надійність та безпека.

Серед існуючих інструментальних засобів для управління конфігураціями програмних мікросервісів був вибраний новий фреймворк Microconfig.io, який дозволяє вирішувати рутинні завдання з налаштування програм. Основне завдання, яке вирішує - це звести до мінімуму дублюючі параметри налаштування та спростити доставку конфігурації до мікросервісів, тим самим зменшити ймовірність додавання помилок. Цей засіб використовує підхід "конфігурація-як-код" для управління МСА, тобто дозволяє описувати конфігурації у вигляді програмного коду, що забезпечує більшу гнучкість та автоматизацію. На рисунку 3.1 показана стартова сторінка при запуску фреймворку Microconfig.io.

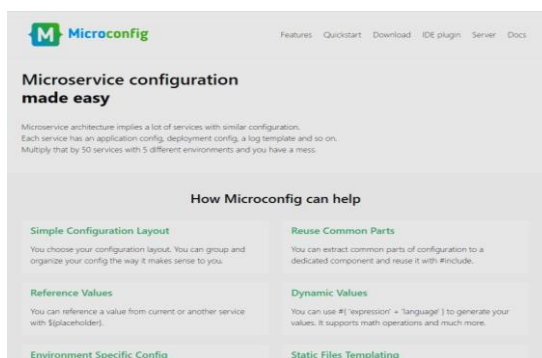


Рис. 3.1 Стартова сторінка фреймворку MicroConfig.io

До основних функцій та можливостей програми Microconfig.io відносяться:

- централізоване збереження конфігурацій, що дозволяє зберігати всі конфігураційні файли в одному місці з метою полегшення управління ними та забезпечує їх однорідність;
- версіонування та контроль доступу, що дозволяє стежити за їхніми змінами у програмі та дає можливість відновлювати попередні версії;
- інтерфейс користувача надає інтуїтивно зрозумілий інтерфейс для управління конфігураціями, що дозволяє легко переглядати, редагувати та виконувати пошук конфігураційних параметрів;
- інтеграція з автоматизованими процесами, коли є можливість легко інтегруватися з автоматизованими процесами розгортання, неперервної поставки та іншими інструментами розробки, що дозволяє автоматизувати процеси налаштування та забезпечує їхню стабільність.

Одною з головних переваг Microconfig.io у порівнянні з аналогічними застосунками є можливість інтеграції з різними інструментами та технологіями розробки, наприклад такими як:

- інструментами для автоматизованого розгортання (Ansible, Chef або Puppet);
- інструментами неперервної поставки (Jenkins, GitLab CI/CD або Travis CI);
- інструментами управління контейнерами (Docker або Kubernetes);

– інструментами системного моніторингу (Prometheus або Nagios).

Це дозволяє зручно налаштовувати кожен програмний мікросервіс незалежно, забезпечуючи гнучкість та масштабованість всієї програмної системи.

3.2 Розробка тестового мікросервісного застосунку

Для дослідження особливостей процесів конфігурування програмних застосунків за допомогою Microconfig.io був розроблений тестовий застосунок з МСА, при цьому не розглядалися питання його безпосереднього використання як програмного продукту, а основною задачею створення було використання його у якості стенда для тестування наявних засобів конфігурування.

Застосунок складався з 3 окремих мікросервісів:

- User-service відповідав за керування користувачами;
- Product-service відповідав за керування товарами;
- Order-service відповідав за керування замовленнями.

На рисунку 3.2 наведена компонентна структура тестового застосунку, з якої видно що Order-service взаємодіє з User-service та Product-service, а локальні дані від кожного сервісу зберігаються в локальну базу даних.

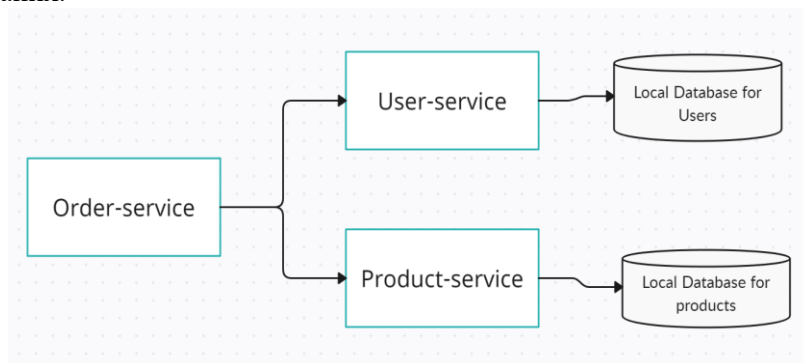


Рис. 3.2 Компонентна структура тестового МСА застосунку

На рисунку 3.3 наведена діаграма класів застосунку.

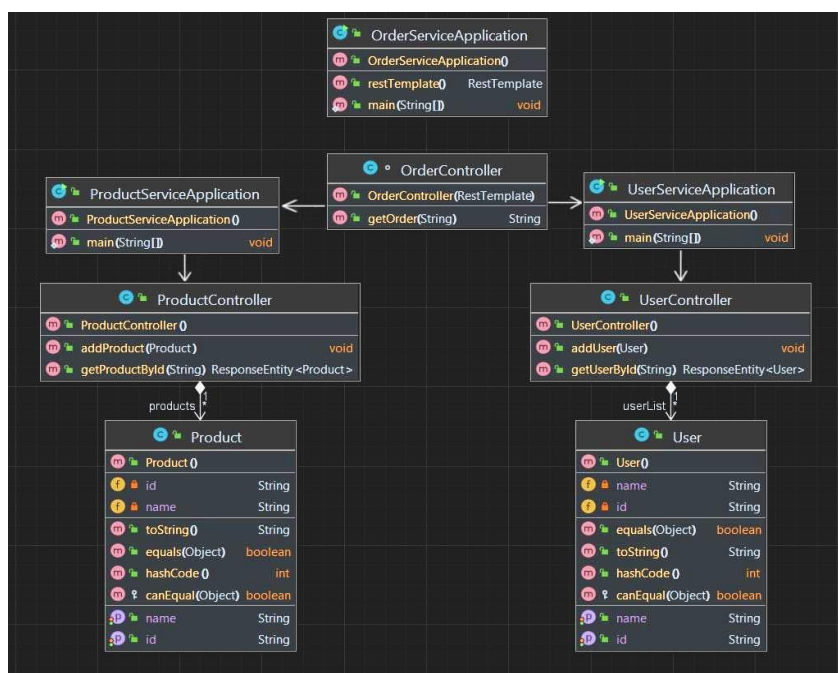


Рис. 3.3 UML діаграма класів тестового застосунку з МСА

Обрана архітектура забезпечувала модульність та розділення обов'язків між мікросервісами, що дозволяло ефективно масштабувати систему. Кожен мікросервіс міг функціонувати незалежно, що полегшувало розгортання та підтримку системи. Крім того, мікросервіси мали змогу взаємодіяти між собою для отримання інформації або виконання певних операцій.

Беручи до уваги те, що полігон розроблявся виключно для тестування, було прийняте рішення не використовувати окремі бази даних, такі як SQL або NoSQL. Був обраний варіант локального збереження файлів на комп'ютері.

3.3. Основні етапи тестування та аналіз отриманих результатів

Для тестування був створений єдиний проект з конфігураціями який включав спільні конфігурації для логування та моніторингу в Microconfig.io. Основні етапи роботи з плагіном це:

- перехід до компонента;
- вирішення заповнювача;
- попередній перегляд конфігурації;
- конфігурація збірки (через «Запуск конфігурації»).

Для генерування конфігурації для user сервісу в певному оточенні в Microconfig.io необхідно виконати команду, яка зображена на рисунку 3.4.



```
microconfig-microservices — zsh — 116x25
(base) sofiiahalamaï@Sofiias-MacBook-Pro microconfig-microservices % microconfig -r . -e dev -s user-service
Filtered 1 component(s) in [dev] env.
Copied 'user-service' template ../log/logback.xml -> logback.xml
Generated user-service/application.yaml
Generated user-service/deploy.yaml
Generated [dev] configs in 299ms
(base) sofiiahalamaï@Sofiias-MacBook-Pro microconfig-microservices %
```

Рис. 3.4 Створення конфігурації для user сервісу

Створення конфігурацій для product та order сервісів показано на рисунку 3.5.



```
Generated [dev] configs in 299ms
(base) sofiiahalamaï@Sofiias-MacBook-Pro microconfig-microservices % microconfig -r . -e dev -s product-service
Filtered 1 component(s) in [dev] env.
Copied 'product-service' template ../log/logback.xml -> logback.xml
Generated product-service/application.yaml
Generated product-service/deploy.yaml
Generated [dev] configs in 294ms
(base) sofiiahalamaï@Sofiias-MacBook-Pro microconfig-microservices % microconfig -r . -e dev -s order-service
Filtered 1 component(s) in [dev] env.
Copied 'order-service' template ../log/logback.xml -> logback.xml
Generated order-service/application.yaml
Generated order-service/deploy.yaml
Generated [dev] configs in 282ms
(base) sofiiahalamaï@Sofiias-MacBook-Pro microconfig-microservices %
```

Рис. 3.5 Створення конфігурацій для product та order сервісів

Результат створення папки збірки з конфігураціями для середовища розробки (dev) для сервісів user, product і orders за допомогою застосунку Microconfig.io показаний на рисунку 3.6. Кожен сервіс має свою власну підпапку з конфігураціями, які відповідають середовищу розробки.



Folder/Item	Created	Type
microconfig-microservices	Today, 12:27	Folder
build	Today, 12:27	Folder
order-service	Today, 12:27	Folder
order-service/deploy.yaml	Today, 12:27	YAML Document (153 bytes)
order-service/application.yaml	Today, 12:27	YAML Document (233 bytes)
order-service/logback.xml	Today, 12:27	XML Document (262 bytes)
product-service	Today, 12:27	Folder
product-service/deploy.yaml	Today, 12:27	YAML Document (157 bytes)
product-service/application.yaml	Today, 12:27	YAML Document (239 bytes)
product-service/logback.xml	Today, 12:27	XML Document (264 bytes)
user-service	Today, 12:26	Folder
user-service/deploy.yaml	Today, 12:26	YAML Document (151 bytes)
user-service/application.yaml	Today, 12:26	YAML Document (230 bytes)
user-service/logback.xml	Today, 12:26	XML Document (261 bytes)

Рис. 3.6 Результат автоматичного створення конфігураційних файлів за допомогою Microconfig.io

Таким чином, в результаті тестування засобу Microconfig.io була показана принципова можливість створення конфігураційних файлів, які включають налаштування як для кожного мікросервісу проекту, так і для всіх сервісів одночасно. Це дозволяє значно спростити процес конфігурування проекту, підвищити швидкість впровадження змін в конфігурації, забезпечує надійність у роботі проекту.

4 Перспективна технологія використання інструментального засобу Microconfig.io з можливостями адаптивного управління та моніторингу показників якості конфігурацій мікросервісів

На підставі проведеного аналізу функціональних властивостей існуючих інструментальних засобів для конфігурування програмних мікросервісів, можна стверджувати, що на теперішній час переважна їх більшість лише забезпечує автоматизацію основних рутинних операцій у цих процесах, і не підтримує такі більш складні можливості їх налаштування, зокрема, адаптацію параметрів нових конфігурацій з урахуванням стану операційного середовища виконання відповідних застосунків з МСА.

Те, що вирішення подібних проблем розробки більш ефективних засобів конфігурування та подальшої координації застосунків з МСА є актуальною науково-технічною задачею, підтверджує поява новітніх публікацій з цих питань, зокрема, в роботах [18, 19]. Для цього в них розробляються підходи із застосуванням алгоритмічних моделей динамічного управління роботою мікросервісів на основі накопичення та обробка інформації про їх взаємозв'язки (inter-service dependencies) та ланцюжки викликів (service call chains), що дозволяє, у кінцевому рахунку, зменшити обсяг необхідної пам'яті для функціонування кластерів мікросервісів та, в цілому, підняти продуктивність роботи усього МСА застосунку. Враховуючи такі тенденції у проблематиці вирішення задач підвищення якості функціонування мікросервісів, у цій роботі пропонується розглянути можливість підвищення ефективності таких інструментальних засобів як Microconfig.io шляхом їх використання у складі перспективної інформаційної технології адаптивного управління процесом конфігурування МСА. Ця технологія представлена у вигляді спрощеної компонентної діаграми в нотації UML на рисунку 4.1.

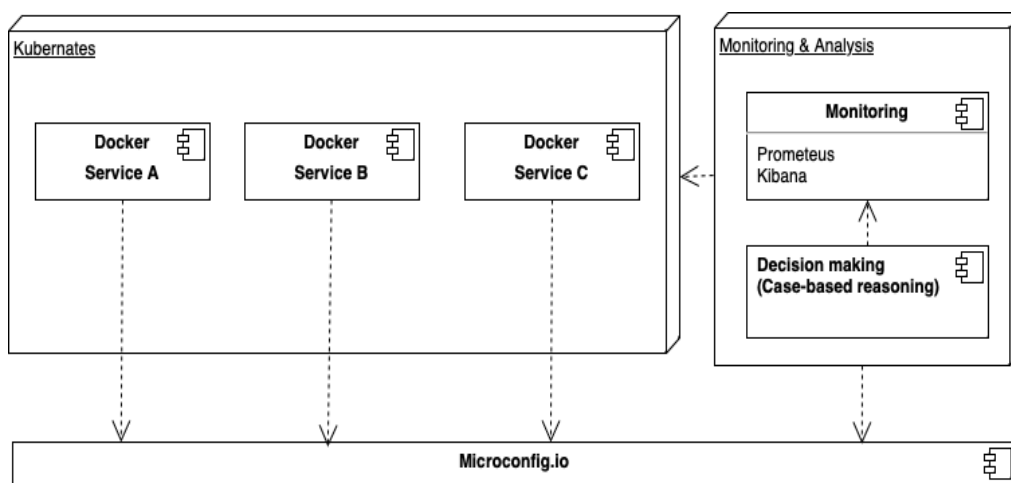


Рис 4.1 Компонентна діаграма адаптивного управління процесом конфігурування МСА

На цій діаграмі операційна платформа Kubernetes відіграє ключову роль у розгортанні та оркестровці множини контейнерів Docker, у яких розміщуються окремі мікросервіси. Кожен мікросервіс, розміщений у своєму власному Docker контейнері, може бути налаштований з використанням інструменту Microconfig.io, який спрощує процес конфігурації через можливість генерації конфігураційних файлів для різноманітних технологій та платформ, включно із Spring, Ansible, тощо (див. Розділ 3). Моніторинг та аналіз параметрів функціонування системи здійснюється завдяки інтеграції таких інструментів як Prometheus [20] та Kibana [21]. Prometheus є інструментом моніторингу з відкритим кодом, який збирає значення певних метрик з наявних конфігурацій МСА на заданих інтервалах, виводить результати та може генерувати системні повідомлення, якщо щось йде не за планом. Ці метрики включають показники від окремих

мікросервісів (Docker) та від інфраструктури їх функціонального середовища (Kubernetes), які відображають стан системи в реальному часі. В свою чергу, компонент Kibana, що є частиною фреймворку Elastic Stack, дозволяє користувачам візуалізувати дані з Elasticsearch, які потім компонент Prometheus може збирати і створювати графіки та візуальні схеми, що полегшує розуміння великої кількості інформації та прискорює процес прийняття рішень щодо процесу адаптивного конфігурування МСА застосунку. Компонент 'Decision making' у цій інформаційній технології, який має бути розробленим додатково, може використовувати методи CBR (Case-based Reasoning) для аналізу отриманих даних і, відповідно до певних правил та алгоритмів, може запропонувати коригування поточних конфігурацій окремих мікросервісів [22], що забезпечує підвищення продуктивності та надійності середовища їх функціонування.

5 Висновки та напрямки подальших досліджень

У цьому дослідженні розглянута актуальна науково-технічна задача підвищення ефективності використання існуючих інструментальних засобів для конфігурування застосунків з мікросервісною архітектурою (МСА) шляхом розробки модельно-технологічних рішень для забезпечення можливостей адаптивного управління в таких застосунках. Визначені базові поняття та сутність процесів управління конфігураціями програмних мікросервісів, проведено аналіз функціональних можливостей інструментальних засобів для конфігурування МСА і побудована їх класифікація. Для подальшого дослідження обрано систему Microconfig.io, яка представляє собою сучасний інструментальний фреймворк для автоматизації усіх основних рутинних операцій в процесах конфігурування МСА. На основі проведеної аналітичної та експериментальної роботи зроблено обґрунтований висновок про можливість та доцільність підвищення ефективності інструментального засобу Microconfig.io шляхом його використання у складі перспективної інформаційної технології адаптивного управління процесом конфігурування МСА, для якої розроблена відповідна UML діаграма компонентів.

У якості подальших досліджень у цій проблематиці можливо запропонувати розробку інформаційної технології з використанням інтелектуальних методів прийняття рішень для координації взаємодії окремих мікросервісів також на етапі їх виконання за умов обмежених операційних ресурсів, таких як оперативна пам'ять, що у кінцевому рахунку може забезпечити підвищення продуктивності МСА застосунків.

ЛІТЕРАТУРА

1. Зінов'єв Д.В., Ткачук М.В., Тріщенко І.В. Моделі та технології забезпечення якості сервіс-орієнтованих програмних систем: сучасний стан та перспективні напрямки досліджень // Матеріали міжн. науков.-техн. конференції КМНТ-2021 (м. Харків, 23-25 квітня 2021 року) – Х.: ХНУ імені В.Н. Каразіна, 2021. – С. 166-169.
2. Зінов'єв Д.В., Ткачук М.В. Розробка інструментального засобу для автоматизованої оцінки показників якості мікросервісних застосунків // «Стан, досягнення та перспективи інформаційних систем і технологій», XXIII Всеукраїнська науково-технічна конференція молодих вчених, аспірантів та студентів. Одеса, 20-21 квітня 2023 р. - Одеса, ОНТУ, 2023 р. – с. 239-240.
3. G. Singh, B. Singh, S. Srivastva. Comparative Analysis of various Software Configuration Management Tools. International Journal of Latest Engineering and Management Research (IJLEMR). Volume 1. Issue 1. November 2015. PP.01-07
4. Site: SimpliLearn. Top 12 Effective & Popular Software Configuration Management Tools. Last updated on Feb 17, 2023. [Електронний ресурс] URL: <https://www.simplilearn.com/configuration-management-tools-article> (дата звернення: 23.10.2023)
5. Chris Richardson "Microservices Patterns: With examples in Java" pp. 1–65. Manning (2019)
6. Redhat сайт. YAML. [Електронний ресурс] URL: <https://www.redhat.com/en/topics/automation/what-is-yaml#:~:text=YAML%20is%20a%20human%20Dreadable,is%20for%20data%2C%20not%20documents.> (дата звернення: 15.05.2023).
7. Hubspot сайт. JSON [Електронний ресурс] URL: <https://blog.hubspot.com/website/json-files#:~:text=What%20is%20a%20JSON%20file,the%20values%20containing%20related%20data.> (дата звернення: 16.10.2023).

8. Indeed сайт. XML [Електронний ресурс] URL: <https://www.indeed.com/career-advice/career-development/xml-file#:~:text=An%20XML%20file%20is%20an,that%20you%20wish%20to%20store>. (дата звернення: 24.10.2023).
9. Git документація. [Електронний ресурс] URL: <https://git-scm.com/> (дата звернення: 17.05.2023).
10. Subversion сайт. [Електронний ресурс] URL: <https://subversion.apache.org/> (дата звернення: 24.10.2023).
11. Mercurial сайт. [Електронний ресурс] URL: <https://www.mercurial-scm.org/> (дата звернення: 24.10.2023).
12. Ansible сайт. [Електронний ресурс] URL: <https://www.ansible.com/> (дата звернення: 25.10.2023).
13. Puppet сайт. [Електронний ресурс] URL: <https://www.puppet.com/> (дата звернення: 26.10.2023).
14. Grafana сайт. [Електронний ресурс] URL: <https://grafana.com/> (дата звернення: 24.10.2023).
15. Nagios сайт. [Електронний ресурс] URL: <https://www.nagios.org/> (дата звернення: 26.10.2023).
16. AWS Secrets Manager сайт. [Електронний ресурс] URL: <https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html> (дата звернення: 21.05.2023).
17. Cornelia Davis "Cloud Native Patterns: Designing Change-Tolerant Software" pp. 1–46. Manning (2019)
18. Wang, N.; Wang, L.; Li, X.; Qin, X. Fine-Grained Management for Microservice Applications with Lazy Configuration Distribution. *Electronics* 2023, 12, 3404. <https://doi.org/10.3390/electronics12163404>
19. Shangguang Wang , Yan Guo , Ning Zhang Peng Yang et.al. Delay-Aware Microservice Coordination in Mobile Edge Computing: A Reinforcement Learning Approach // *IEEE Transactions on mobile computing*, Vol. 20, No. 3, 2021 pp. 939-951
20. Pivotto, Julien, and Brian Brazil. *Prometheus: Up & Running*. 2nd ed., O'Reilly Media, 2023
21. Konda, Madhusudhan. *Elasticsearch in Action*. 2nd ed., Manning Publications, September 2023
22. Tkachuk M.V., Zinoviev D.V. A case-based reasoning approach to quality assurance in microservice software systems // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я: Тези доповідей XXXI міжнародної науково-практичної конференції MicroCAD-2023, 17-20 травня 2023р., / за ред. проф. Сокола Є.І. – Харків, НТУ «ХПІ». – С.1034.

REFERENCES

1. Zinov'ev D.V., Tkachuk M.V., Trishchenko I.V. Models and technologies for ensuring the quality of service-oriented software systems: the current state and promising directions of research // *Materials of the International. scientific and technical of the KMNT-2021 conference (Kharkiv, April 23-25, 2021) - Kh.: KhNU by V.N. Karazin, 2021. - P. 166-169. [in Ukrainian]*
2. Zinov'ev D.V., Tkachuk M.V. Development of a tool for automated assessment of quality indicators of microservice applications // "State, achievements and prospects of information systems and technologies", XXIII All-Ukrainian scientific and technical conference of young scientists, graduate students and students. Odesa, April 20-21, 2023 - Odesa, ONTU, 2023 - p. 239-240. [in Ukrainian]
3. G. Singh, B. Singh, S. Srivastva. Comparative Analysis of various Software Configuration Management Tools. *International Journal of Latest Engineering and Management Research (IJLEMR)*. Volume 1. Issue 1. November 2015. PP.01-07
4. Site: SimpliLearn. Top 12 Effective & Popular Software Configuration Management Tools. Last updated on Feb 17, 2023. URL: <https://www.simplilearn.com/configuration-management-tools-article> [date of access: 25.10.2023]
5. Chris Richardson "Microservices Patterns: With examples in Java" pp. 1–65. Manning (2019)
6. Site "Redhat". What is YAML? [Електронний ресурс] URL: <https://www.redhat.com/en/topics/automation/what-is-yaml#:~:text=YAML%20is%20a%20human%20Dreadable,is%20for%20data%2C%20not%20documents> [date of access: 25.10.2023]
7. Site "Hubspot". What Are JSON Files & How Do You Use Them? URL: <https://blog.hubspot.com/website/json-files#:~:text=What%20is%20a%20JSON%20file,the%20values%20containing%20related%20data> [date of access: 25.10.2023]
8. Site "Indeed". What Is an XML File? (Definition, Benefits and How To Open). URL: <https://www.indeed.com/career-advice/career-development/xml-file#:~:text=An%20XML%20file%20is%20an,that%20you%20wish%20to%20store> [date of access: 25.10.2023]
9. Site "Git. Documentation". URL: <https://git-scm.com/doc> [date of access: 25.10.2023]

10. Site “Subversion”. URL: <https://subversion.apache.org/> [date of access: 25.10.2023]
11. Site “Mercurial”. URL: <https://www.mercurial-scm.org/> [date of access: 25.10.2023]
12. Site “Ansible”. URL: <https://www.ansible.com/> [date of access: 25.10.2023]
13. Site “Puppet”. URL: <https://www.puppet.com/> [date of access: 25.10.2023]
14. Site “Grafana”. URL: <https://grafana.com/> [date of access: 25.10.2023]
15. Site “Nagios”. URL: <https://www.nagios.org/> [date of access: 25.10.2023]
16. AWS Secrets Manager”. URL: <https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html> [date of access: 25.10.2023]
17. Cornelia Davis "Cloud Native Patterns: Designing Change-Tolerant Software" pp. 1–46. Manning (2019)
18. Wang, N.; Wang, L.; Li, X.; Qin, X. Fine-Grained Management for Microservice Applications with Lazy Configuration Distribution. *Electronics* 2023, 12, 3404. <https://doi.org/10.3390/electronics12163404>
19. Shangguang Wang, Yan Guo, Ning Zhang Peng Yang et.al. Delay-Aware Microservice Coordination in Mobile Edge Computing: A Reinforcement Learning Approach // *IEEE Transactions on mobile computing*, Vol. 20, No. 3, 2021 pp. 939-951
20. Pivotta, Julien, and Brian Brazil. *Prometheus: Up & Running*. 2nd ed., O'Reilly Media, 2023
21. Konda, Madhusudhan. *Elasticsearch in Action*. 2nd ed., Manning Publications, September 2023
22. Tkachuk M.V., Zinoviev D.V. A case-based reasoning approach to quality assurance in microservice software systems // *Information technologies: science, technology, technology, education, health: abstracts of reports of XXI international scientific and practical conference MicroCAD-2023*, May 17-20, 2023, / edited by Prof. E.I. Sokol - Kharkiv, NTU "KhPI". - P.1034. [in Ukrainian]

Zinov'ev Dmitry

Senior lecturer

of the Department of Systems and Technology Modeling, V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine;

Tkachuk Mykola

Doctor of Technical Sciences, Professor;

Professor of the Department of Systems and Technology Modeling, V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine.

Analysis, classification and testing of configuration management tools for software microservices

Actuality. The development of software applications with microservice architecture (MSA) requires to solve problems of their design, integration, scaling, adaptation, reliability, fault tolerance and, in general, an improving the quality indicators of such systems, and therefore the issue of effective management of MSA configurations is an urgent scientific and technical task.

Goal. The purpose of this study is to analyze the functional features of the existing tools for configuring MSA, as well as their classification and testing, which enables their improvement by developing of model-technological solutions to ensure adaptive management in such applications.

Research methods. In order to achieve the goal of the research, the basic concepts and the essence of the management processes of MSA configurations were determined, the functional capabilities of some modern tools for MSA configuring were analyzed and their classification is elaborated. For the further research, the Microconfog.io framework was chosen, for which the specific example of the MSA application was developed, its testing has been carried out, and the obtained results were analyzed.

Results. A well-founded conclusion was made about the possibility and expediency of increasing the effectiveness of the MicroConfig.io tool by using it as a part of the perspective information technology for adaptive management of the MSA configuration process, for which a corresponding UML component deployment diagram has been developed.

Conclusions. The functional features of modern tools for MSA configuring were analyzed, their possible classification was built, and the software testing of one of the typical such systems, namely, the Microconfig.io framework, was provided. A promising information technology of adaptive management of the MSA configuration process is proposed in the form of a component deployment diagram, and the directions for further research are formulated.

Keywords: software microservices, configuration management, classification, testing, adaptation, quality, metric, information technology, UML, component diagram, model, tool

How to quote: D.V. Zinov'ev, M.V. Tkachuk, “Analysis, classification and testing of configuration management tools for software microservices.” *Bulletin of V.N. Karazin Kharkiv National University, series “Mathematical modelling. Information technology. Automated control systems*, vol. 57, pp.32-41, 2023. <https://doi.org/10.26565/2304-6201-2023-57-03>

UDC 656.11

P.K. Nikolyuk*doctor of physical and mathematical sciences, professor of information technologies department**Vasil' Stus Donetsk National University, 600-richchia str., 21, Vinnytsia-21, Ukraine, 21021**e-mail: p.nikolyuk@donnu.edu.ua***<https://orcid.org/0000-0002-0286-297X>**

Routing a vehicle through city traffic by the time-optimal dynamic path

The key goal of this study is to synchronize traffic flows, optimize the use of the transport arteries throughout the city, prevent congestion, and follow each vehicle to its destination to minimize time spent on the trip. As a result, the total time spent by cars on the road will be significantly reduced, and environmental conditions will improve accordingly.

The object of the study is a city's transportation network, represented as a weighted oriented nonplanar multigraph (WONM). The key advantage of using the graph theory to build optimal routes is based on the following considerations: 1) the graph theory has developed many algorithms for finding optimal routes (Dijkstra algorithm, Floyd-Warshall algorithm, A-star algorithm, etc.); 2) the graph theory is used as the theoretical and practical basis of logistical systems, including urban traffic. To build a route in such a multigraph an A-star algorithm has been used, which establishes an optimal time (t-optimal) route between two selected vertices of graph.

The study offers a realistic prospect for solving the problem of congestions through the use of a special software algorithm oriented towards establishing optimal routes and using graphs to represent the city's transportation network.

The fundamental issue is the representation of the city's transport network in the form of an electronic map and the display of GPS-identifiers of vehicles involved in traffic on this map. The "city traffic → electronic map" representation makes it possible to obtain data as to the level of congestion in the transport network. The use of an electronic city map allows the GPS coordinates of each vehicle to be projected onto it. Thus, the city's transport network is under the full control of the transportation management center (TMC), which has a real opportunity to interact with each vehicle and constantly adjust its route, choosing the t-optimal one. The route adjustment is carried out via General Packet Radio Service (GPRS) channel in the form of voice commands as in conventional GPS navigation. However, the specifics are as follows: 1) the navigation is carried out online; 2) t-optimal routes are plotted, taking into account the traffic situation at any given time.

Thus, a large-scale transportation urban traffic network and an associated computer program has been developed. This is an applied project, and its results can be used to effectively regulate traffic in megacities in order to minimize the travel time of each vehicle along a selected route.

Keywords: *weighted oriented nonplanar multigraph, GPS-navigation, A-star algorithm, t-optimal route, intelligent traffic.*

How to quote: P.K. Nikolyuk, "Routing a vehicle through city traffic by the time-optimal dynamic path", *Bulletin of V.N. Karazin Kharkiv National University, series "Mathematical modelling. Information technology. Automated control systems*, vol. 57, pp. 42-54, 2023.

<https://doi.org/10.26565/2304-6201-2023-57-02>

Як цитувати: Ніколюк П.К. Маршрутизація транспортного засобу з оптимальним за часом динамічним міським шляхом руху. *Вісник Харківського національного університету імені В. Н. Каразіна, сер. «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*. 2023. вип. 57. С.42-54. <https://doi.org/10.26565/2304-6201-2023-57-02>

1. Introduction

In the graph theory many algorithms have been developed to build optimal routes, including t-optimal ones [1] and this is the key advantage of the multigraph model over other traffic simulation models. Another key advantage of the multigraph model is the possibility to simulate the city's transportation network, depicting all lanes, intersections, traffic junctions, and pedestrian crossings. These features allow implementing the procedure of programmatic regulation of urban traffic, which is a particularly important result of the study. In other words, a basic framework for automating urban traffic processes has been created. The model of the urban transport network in the form of a WONM allows using graph theory algorithms to find optimal routes for vehicles participating in traffic. It is most appropriate to use the A-star algorithm, since it ensures the optimal routes between two selected positions of the graph (city transport network) and has a number of advantages over other similar algorithms.

The fundamental issue is the creation of bijective mapping principles of the "city traffic → WONM" type. In other words, the location of vehicles on urban transport arteries should be reproduced as

objectively as possible on the urban electronic map used by the city TMC. In addition, constant interaction between each vehicle, which is identified on the electronic map by a pair (O_i, D_i) is required. Each such pair is a specific vehicle identifier which can be tracked on the city's electronic map. Since each car i is under the control of TMC, tracking it along the selected route is equivalent to tracking the identifier (O_i, D_i) on the electronic city map. In turn, vehicle tracking involves online interaction with the vehicle driver via a GPRS channel. Therefore, each driver can receive accurate and timely instructions on the best way to travel. Importantly, such instructions correlate with the dynamics of urban traffic and are the result of implementing the A-star algorithm, which creates the optimal route in the graph and constantly updates the route in accordance with the traffic situation at a particular time. In other words, the route of each car is synchronized with the dynamical changes in the traffic situation.

Practically, regulating the traffic of tracked vehicles is carried out by automating traffic control at each intersection and road sections between neighboring intersections. This allows TMC to control all traffic flows in the metropolis in real time, as well as to track each vehicle along the route where the initial point is established by using a GPS navigator and the destination point is set.

The process of controlling and regulating urban road traffic is based on the fact that TMC transmits voice commands to each driver via GPRS channel highlighting the route to the destination point as in conventional GPS navigation. The distinct feature of the program is the analysis of the dynamic traffic situation throughout the city by using interactive road sensors located on the electronic map of the city in the vicinity of intersections.

The practical application of the described process of regulating the urban traffic will lead to the disappearance or significant reduction of traffic jams. The main results of this study are: 1) optimization of vehicle routes, which minimizes the travel time between origin and destination points; 2) synchronization of traffic flows; 3) the uniform redistribution of traffic flows within the city, which will lead to optimal use of transport arteries throughout the city; 4) absence or significant reduction in the duration and frequency of congestions.

2. Analytical review of the related works

The previous investigation [2] addresses the problem of traffic control in a metropolis by using the Dijkstra algorithm. The authors consider the Intelligent Transportation System (ITS) application in order to analyze the causes of congestion on urban transport arteries. ITS makes it possible to provide innovative services and enable users to be better informed and make safer, more coordinated, and 'smarter' usage of transportation networks.

A map-matching algorithm [3] allows for the correct comparison of the GPS-trajectory of the vehicle with the corresponding road segment. In particular, the comparison of GPS data with the electronic map of the city is carried out by using Automatic Vehicle Location (AVL) systems that use vehicle tracking system to track the movement of a vehicle. In turn, the information collected by AVL can be compared with electronic maps via the Internet or with help of special software. A map-matching algorithm is a significant part of any navigation system because it aligns data from the GPS with a digital road network. The map-matching algorithms are divided into simple, weight-based, and advanced ones. The advanced algorithms use different mathematical models such as probability theory, fuzzy logic, multihypothesis road tracking, a hidden Markov model, a hybrid Bayesian network and a neural network. The cited research proposes the weight-based algorithm to find a best segment of road network. In [3,5], the problem of map-matching of GPS markers (points) of vehicles with electronic maps is considered in detail. For this purpose, the so-called device-based sensors are used, which constantly record and transmit space-time information about the location and movement of the vehicle [6]. The latest applications of urban vehicle mobility visualization systems are based on the usage of Geographic Information Systems (GIS). This system allows linking data to an electronic map, in particular displaying the geolocation position of the vehicle on such a map. The study [7] shows that finding the shortest route by using an A-star algorithm facilitates the procedure of comparing a vehicle position on electronic map with road segments. That means that Dijkstra or Floyd algorithms are less effective in this case.

City traffic flows need recording in real time, since knowing the level of congestion on city streets is the starting point of solving a traffic flow regulation problem. The next step is for the TMC to select the best routes in the urban transportation network. This is key problem in terms of urban traffic strategy. Thus, in [8], a new control system using the heuristic approach is used to solve this problem. The authors propose a new integrated control algorithm that combines actions of dynamic traffic routing with on-

ramp metering to optimize the traffic flow. The algorithm can manage the system in the case of a capacity constraint in the highway network. The investigation focuses on the development of integrated traffic management strategies in road networks. The highway network is modeled by using the Lighthill-Whitham-Richards traffic flow model, which introduces the terms of flow density and average speed. An integrated control algorithm based on the methodology of feedback control and variable structure control has been developed to solve the proposed problem. Three options for selecting alternative routes to avoid congestion have been tested. The results show that the proposed algorithms can offer to the user a balance between the selected alternative routes.

A number of studies have been aimed at implementing the principles of urban traffic forecasting. Studying networks of different nature, such as neural networks, allows making forecasts. In particular, short-term traffic forecasting [9] is one of the most important elements of all active traffic control. The development of a forecasting model based on neural networks allows for short-term forecasting based on the selection of the best combination of forecast parameters. In this work, a self-correcting neural network based on genetic sorting has been used. The NSGA-II algorithm is used as a multi-objective optimizer for short-term forecasting.

The Kalman filter [10] for traffic forecasting on urban highways based on data obtained from connected vehicles is used. That allows for real-time forecasting, since the data of the connected vehicle is analyzed immediately before the forecast period. To analyze traffic data, the Vissim simulator, which registers vehicles at different speeds is used. The performance of the algorithm for different traffic situations is evaluated by using statistical methods.

The short-term forecasting of traffic scenarios [11] uses a set of specific tools and models. In particular, such forecasting models as the non-parametric k-Nearest Neighbor (kNN) regression model, the Gaussian probability maximization model, and the double seasonal exponential HolteWinters smoothing model are important components. Real highway traffic data is used to test the theoretical conceptions. The research allows predicting weekly and monthly fluctuations in average daily traffic with varying degrees of precision while being ease of use. The article uses the information entropy method and the less common Shapley method.

Forecasting of traffic processes can be carried out both for the day and for a longer-term range [12]. For this purpose, a traffic management system based on predictive information through the use of static and mobile objects is used. These objects (agents) use a methodology for collecting and transmitting data on traffic flow parameters (speed and density), spatial and temporal information on urban traffic modes in order to monitor and predict expected traffic density patterns. All that permits the vehicle driver to choose the best routes and thus ensure uninterrupted traffic flow and reduce the frequency of traffic jams. Traffic situation monitoring and forecasting is performed through the integration of NS2, Simulation of Urban Mobility (SUMO), OpenStreetMap (OSM) and the MOVE tool.

Urban traffic is a function of the capacity of a particular urban intersection or a place where traffic flows intersect. The fundamental question is how to organize an effective process of regulating the passage of vehicles through this intersection. Therefore, the main task is to create a mode of optimizing the capacity of each urban intersection [13]. For this purpose, an algorithm to control traffic signals at isolated intersections is used. In this approach, a central controller is used to collect the real-time location of communicating vehicles at certain intervals. The controlling of traffic lights is focused on the correct choice of the phase of the traffic signal, which maximizes the average vehicle speed while limiting the maximum delay that any individual vehicle can experience. The proposed algorithm is also effective when the level of interaction between connected vehicles is imperfect, when such objects account for more than 40% of the traffic flow. The results show that the proposed strategy can help to significantly reduce long delays. Intersection coordination is carried out through the implementation of a computer program for regulating vehicle flows [14]. The presented computer program allows optimizing the passage of cars through each individual intersection.

The invention of real-world applied algorithms has played an important role in creating an intelligent intersection control [15]. In particular, an intelligent traffic light can be implemented on the basis of an intelligent terminal. The intelligent terminal has a bidirectional communication with the TMC, which is used to manage urban roads and provide information services for cars, as well as provide electronic maps and route maps for the intelligent terminal. The intelligent traffic light system automatically detects the number of vehicles on the roads to adapt the travel time for each road. The system and method provided by the invention can help drivers optimize route maps and improve the efficiency of vehicle traffic.

The adaptive traffic light control system [16] operates in real time. The authors propose to consider each intersection in the urban transport network as an independent, autonomous object that can change the length of the green phase of the traffic light depending on the characteristics of local traffic. This will significantly improve the capacity of each individual intersection.

Robot cars are promising objects that are receiving special attention from researchers. Such cars operate in an autonomous mode – without drivers [17]. The advantages of this innovative approach and the real prospects for its use are becoming apparent today. This problem has recently received considerable attention from researchers around the world, and the number of publications has been growing sharply in recent years.

3. Presentation of the problem and the results of the work

It is convenient to regulate urban traffic on an electronic map of the city by means of a bi-active display of the type "urban traffic \rightarrow electronic map". In this case, urban traffic can be managed by TMC. However, the implementation of the transformation "urban traffic \rightarrow electronic map" is a significant problem [6]. To solve this problem, the following tasks have been set (Fig. 1):

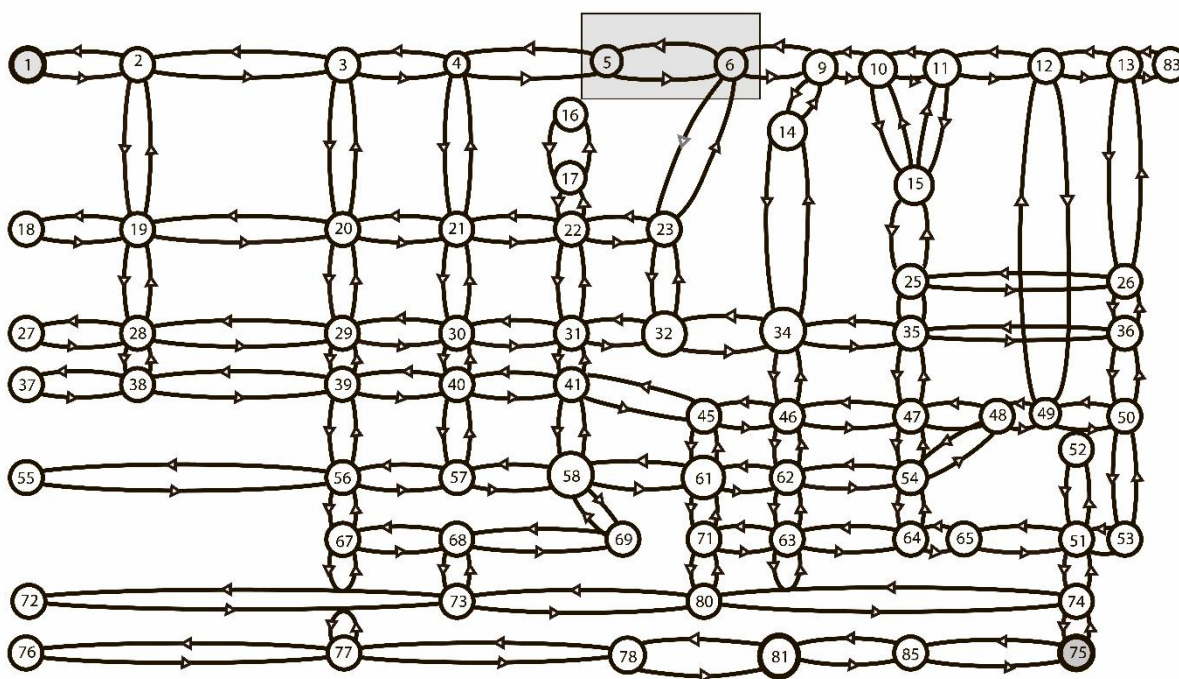


Fig.1. Weighted oriented non-planar multigraph model of the city's transport network fragment. The highlighted gray square is shown in detail on Fig. 2 below. The origin and destination points of a particular vehicle are located at nodes 1 and 75, respectively.

- to create a real-time model of the urban transport network in the form of an oriented weighted nonplanar multigraph with dynamically loaded arcs;
 - to activate the electronic map of the city in TMC and use this map to track each vehicle;
 - to create optimal time routes for all vehicles that have ordered such routes;
 - to implement in real time the software algorithm for constant route updates for each guided vehicle.
- This will allow synchronizing city traffic with the traffic management procedure on the TMC and making adjustments to the route for each accompanied vehicle.

The schema of the principle of traffic organization is shown as the diagram on Fig. 1. Urban traffic management is relied upon by the TMC, which consists of three interconnected subdivisions: the city road electronic map, which receives geo-positioning data from each vehicle in the city road network.

After processing, the data is transferred to the information and computing center. The routes calculated here send as instructions via a GPRS channel to each vehicle by traffic control center that manages each vehicle.

The creation of a model of the city's transport network by using the WONM makes the operational management of traffic flows possible. The fundamental point is that in such a graph, each arc is matched with a weight - a dynamic value correlated with the actual traffic load of each city road lane. In the terms of graph theory, this graph is a bijective (as objective as possible) representation of the distribution of vehicles on the city's transport arteries. WONM is the basic representation of an electronic traffic map of the city. GPS identifiers of vehicles involved in urban traffic are projected onto such a map [3].

The optimal routes in the graphs are constructed by using special algorithms, among which the Dijkstra and Floyd-Warschall algorithms are the most well-known [18]. The first one finds the optimal route between the selected graph vertex and all other vertices, and the second one finds the optimal route between all vertices of the graph. However, for this study, we need to build optimal routes between two selected vertices of the graph, since each driver wants to get from a certain position A to position B. In this regard, it is advisable to use the A*-algorithm to build optimal routes between two specific vertices of the graph.

The implementation of a software algorithm that builds optimal routes has a number of features, including high traffic dynamics and different levels of traffic congestion in individual lanes on the same road section and in individual streets. It is known that city central highways are usually congested during peak hours, while secondary transport arteries operate in normal or even underloaded traffic mode [6]. Therefore, one of the fundamental tasks of the proposed algorithm is to organize a predominantly uniform distribution of vehicles along the city's road arteries.

One of the issues is interacting with each vehicle in order to transfer data on the calculated optimal route. The registration of vehicles participating in traffic is carried out through the use of special sensors located on the electronic map of the city in the vicinity of intersections. These sensors interact (read information) with an electronic vehicle marker O_i, D_i when it crosses the electronic sensor. The data from the sensors act as input values to the information and computing center that implements the A-star algorithm (Fig. 1). The center plans the optimal route for each vehicle, taking into account the traffic situation in real time. For this purpose, each route is constantly corrected. Let us consider the specific details of the algorithm by using the example of the model shown in Fig. 2.

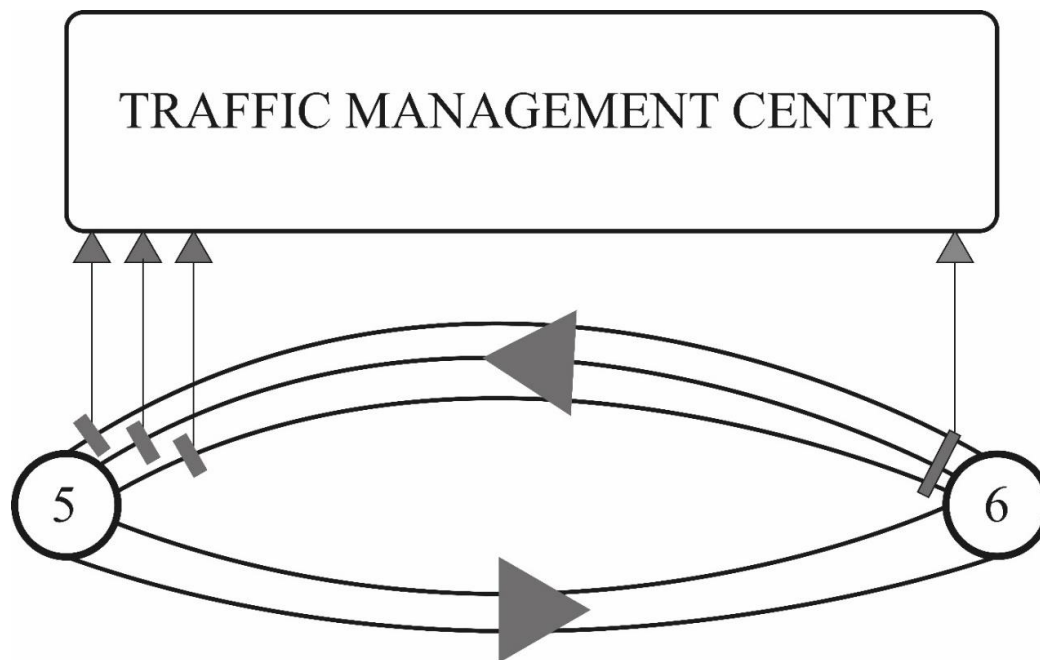


Fig. 2. Scheme of an interaction between electronic map sensors and TMC

This simple graph is an actual representation of a section of the city's transportation network. This means that the lanes between intersections are reproduced – there can be 2, 3 or more such lanes, which

corresponds to the situation with the real urban transport network. In addition, each arc of the graph is assigned a time-varying weight that corresponds to real city traffic. The graph (Fig. 1) also contains two types of traffic interchanges – the first is vertex 17, where circular traffic is organized, and the second is organized between intersections 2 and 8, which are connected by a road that runs over intersection 5. The graph (Fig. 2) models not just the road between intersections, but also reproduces individual lanes, which is a fundamentally important point, since the congestion of vehicles in different lanes, as a rule, varies significantly. Therefore, it is necessary to record the congestion of each lane separately. The fundamental issue is to build the time-optimal route for each vehicle. For this purpose, the following definition should be introduced:

Definition 1. The intersection traffic resistance as a function of the traffic observation time t in the direction of the intersection for cars located along lane h is defined as the time interval for which a car is delayed at the intersection j :

$$R_h^j(t) = S_h^j(t) \cdot T_h^j(t), \quad (1)$$

where $T_h^j(t) = \tau_j^{red}(h) + \tau_j^{green}(h)$ – the length of the traffic light switching cycle at the intersection j

(
t
h
e

l
e
n
g
t

$$S_h^j(t) = \begin{cases} \delta \\ \tau_h^j / T_h^j + \delta \\ \tau_h^{red} / T_h^j + C_h^j + \delta \\ t_h^{green} / T_h^j + C_h^j + \delta \end{cases}, \quad (2)$$

where the value $\delta \ll 1$ corresponds to the ratio (vehicle travel time through the intersection)/(the length of the traffic light cycle T_h^j). Thus, in the case of the first term, the vehicle crosses the intersection j without any delay. The value $\tau_h^j / T_h^j + \delta$ in the second term is obtained when the vehicle approaches the intersection j when the red light is on, delays for a while $\tau_h^j \leq \tau_h^{red}$, and passes the intersection immediately after the green phase turns on. The value $t_h^{red} / T_h^j + C_h^j$ in the third term consists of the fraction T_h^j plus the number of traffic light switching cycles ($C_h^j \geq 1$) during which the car is delayed at the intersection. The value $t_h^{green} / T_h^j + C_h^j$ is a part of the value of the fourth term, consisting of the fraction T_h^j plus the integer number of traffic signal cycles (C_h^j).

The delay time (intersection resistance) of a car at an intersection is given as

w
s
i
g
n
a

$$R_h^j(t) = \begin{cases} \delta \cdot T_h^j \\ \tau_h^j + \delta \cdot T_h^j \\ t_h^{red} + C_h^j \cdot T_h^j + \delta \cdot T_h^j \\ t_h^{green} + C_h^j \cdot T_h^j + \delta \cdot T_h^j \end{cases} \quad (3)$$

The weight $W^h(l_{ij}, t)$ of each arc of the graph (Fig. 2) is the sum of

i
s

$$W_h(l_{ij}, t) = R_h^j(t) \cdot V_{ij} + L_h. \quad (4)$$

i
n
c
l
u

Where l_{ij} is a lane directed from the intersection i to the intersection j ; l is a designation for the lanes named starting from the centerline as a, b, c, d, \dots (for example, the entry c_{6_5} symbolizes the third lane from the dividing line of the road, which is directed from intersection 6 to intersection 5 – Fig. 2). In the program algorithm [19], which presents the complete procedure for calculating the optimal route the antipode of expression (4) is the value of $double\ Wli_j = R() \cdot 10 + Li_j$. We assume that the average vehicle speed (in m/s) on urban transport arteries is 10. The principal task of the study is to find the minimum value of the function

$$\sum_{h=1}^H (R_h^j(t) \cdot V_{ij} + L_h) \rightarrow \min . \quad (5)$$

Where, H is the set of lanes along the route that a vehicle passes while moving from the start to the end position. Expression (5) consists of a variable component (the first term) and a constant component (the second term). The variable component is a function of time, since traffic in the city is highly dynamic.

The fundamental issue is to build a route synchronized with real, highly dynamic urban traffic. The procedure of such synchronization is implemented through the process of constantly recalculating the optimal route for each vehicle by a program. Specifically, the route is recalculated to take into account the state of city traffic at a particular moment every time the vehicle crosses the next intersection. In other words, we are talking about the route that corresponds to the traffic situation at a given time. The features of the process of implementing a dynamic software process that allows for route adjustment will be described below.

Expression (5) represents the objective function of the problem: the total weight of all consecutive lanes along the route should be minimal. The minimization of this function is performed by the A*-algorithm. The code for the graph (Fig. 1) is presented in [19].

Definition 2. The weight of the route (5) is the sum of the real and virtual components of the path:

$(\sum_{h=1}^H L_h)$ is the real component of the path that the car physically travels along the route; $(\sum_{h=1}^H R_h^j(t) \cdot V_{ij})$ is the virtual component of the path – this is the path that the vehicle would have traveled during the delay at the intersection $R_h^j(t)$, moving at an average speed V_{ij} .

Expression (5) is equivalent to the following

$$\tau^{trip} = \sum_{h=1}^H (R_h^j(t) + L_h / V_{ij}) \rightarrow \min , \quad (6)$$

where τ^{trip} is the travel time of the vehicle from its starting position to the destination. In expression (6) $R_h^j(t)$ there is a delay time of the vehicle (which was in the traffic lane h) at the intersection j , during which the vehicle is in the standby mode. To travel such a route (6) takes a minimum time.

According to (6), there is reason to talk about optimizing the car's travel along the selected route in terms of time. Accordingly, the total time spent by all vehicles participating in the traffic will also be optimal (minimal).

An important value that characterizes urban traffic is the coefficient of dynamics, which characterizes the capacity of each individual intersection

$$D_h^j(t) = m_h^j(t) / (N_h^j(t)) . \quad (7)$$

Where $N_h^j(t)$ represents the number of vehicles in the lane h in the direction of the intersection j and the value $m_h^j(t)$ is the number of vehicles that passed through the intersection j from the lane h during the green phase t_h^{green} of the traffic light. Thus, the traffic dynamics on a lane h along a road section $i \rightarrow j$ is determined by the ratio between the number of vehicles leaving this road section during

the time equal to the green phase t_h^{green} of the traffic light to the number of vehicles on the road section $i \rightarrow j$ at a given time.

The throughput $D_h^j(t)$ is related to the intersection resistance $R_h^j(t)$ as follows

$$D_h^j(t) = k / S_h^j(t), \quad (8)$$

where the coefficient k has a time dimension.

The objective function is written in the form

$$D_h^j(t) \rightarrow \max. \quad (9)$$

The analysis of the relation (8) shows that the dynamics of traffic directly depends on the capacity of the city's intersections. A number of works have been devoted to the issue of improving the throughput of urban intersections [20, 21]. In our study, the process of creating the intelligent intersection is based on the following principle.

Definition 3: Intersection intelligence is a traffic signal control process in which the length of the green light phase t_h^{green} in a direction $i \rightarrow j$ is proportional to the number of vehicles in that direction.

Creating the intelligent intersection allows maximizing both its capacity (9). In order to illustrate the operation of such object, it is advisable to conduct a model experiment by using the AnyLogic North America LLC 8.8.0 program. The purpose of the modeling procedure is to show how creating the intelligent intersection affects the value of $D_h^j(t)$.

To control traffic, it is necessary to control the values included in (6). The process of fixing the described values is organized as follows: a sensor i_h (input sensor), located on an electronic map in the vicinity of the intersection i in the direction of the intersection j (Fig. 2), registers the geoposition of vehicles in accordance with their code identifier. The spectrum of such identifiers is given by the relation

$$S = \{(O_1, D_1), (O_2, D_2), \dots, (O_k, D_k), \dots, (O_N, D_N)\}. \quad (10)$$

Where N is the number of vehicles participating in urban traffic. Each car gets its own unique token consisting of an origin-destination pair (O_k, D_k) . This pair is a specific recorder of each tracked vehicle, which allows it to be identified when it crosses the input or output sensors of the electronic map. The set of output sensors (Fig. 2) located in the vicinity of the intersection j on the lanes directed in the direction of $i \rightarrow j$ registers vehicles that have left the road section $i \rightarrow j$. The input sensors, on the opposite, register vehicles that have entered the road section $i \rightarrow j$. The ratio $D_h^j(t)$ of these values according to (7) indicates the dynamics of traffic on a particular road section in a particular direction $i \rightarrow j$. It is clear that the closer $D_h^j(t)$ is to unity, the higher the traffic dynamics on the road section in the direction of $i \rightarrow j$. The task is to find a sequence of lanes in the urban transportation network whose total weight is minimal (in terms of vehicle travel time) compared to all possible options – a t-optimal route.

Relation (7) characterizes the dynamics of vehicle movement on the road section $i \rightarrow j$. If

$$1 \leq N_h^j(t) / m_h^j(t) < 2, \quad (9)$$

we can talk about high dynamics on this section of the road.

On the contrary, if

$$N_h^j(t) / m_h^j(t) \gg 1, \quad (10)$$

it indicates low dynamics of vehicle movement in the corresponding lane h , which causes a large weight of this lane. Thus, it is necessary to determine the throughput for each individual lane separately. That is taken into account in developed software by implementing the A*-algorithm.

The functional relationship between the relations (10) and (11) and the intersection resistance $R_h^j(t)$ is represented as

$$R_h^j(t) \sim 1/D_h^j(t). \quad (11)$$

Improving the throughput of each individual intersection is a separate problem, which has been addressed in a number of studies [4,7,22,23]. If we improve the capacity of each individual intersection, then the terms $R_h^j(t)$ in the objective function (6) will be minimized. Thus, the procedure for minimizing the objective function (6) consists of two stages: 1) minimization of the total weight of the route, which is associated with the choice of the route in the graph/urban transport network; 2) minimization of the travel time through each individual intersection.

In order to simulate the weights of graph arcs, the program [19] uses a random variable generator that generates a variable part of the weight of a graph arc by using the method of *public static double R()*. In fact, this method returns a spectrum of real numbers in the range 0÷500. This range corresponds to the real traffic situation caused by the delay of a vehicle at an intersection in the range of 0÷10 times of traffic light switching cycles, where an average duration of one cycle is 50 seconds. The weight of the graph arc $W_h(l_{ij}, t)$ is represented as the sum of (4), the value of which correlates with the congestion of urban transport arteries. In the program, the weights of the graph arcs are given as follows *double Wa_{2_1} = R() · 10 + L_{2_1}*. For example, for the lane a_{2_1} we get *double Wa_{2_1} = R() · 10 + L_{2_1}*. Thus, each arc of the graph (Fig. 1) receives a time-varying weight, the value of which is constantly changing, which corresponds to the realities of urban transport arteries. Fig. 3 shows a mix of neighboring intersections. When the geopositioning tags of the vehicles cross the input sensors of the electronic map, the signal is sent to the ICC and the re-routing procedure [19] for vehicles with markers $(O_i, D_i, i \in N)$ is initiated.

The re-routing procedure is necessary to respond to changes in urban traffic, in other words, to synchronize the traffic dynamics with the routing procedure. For example, in the case of the specific vehicle (Fig. 3), the TMC program calculates a new route as soon as the specified vehicle crosses the intersection i (the geopositioning mark of the vehicle on the electronic map crosses the input sensor i_h). This process initiates the transmission of the control signal via GPRS channel and at the very beginning of the road section (that is important!) the driver of the vehicle will receive the program-calculated instructions for the further route selection. Thus, the driver has the time and opportunity to shift to the desired lane in order to proceed to the next intersection, where the re-routing procedure will be repeated. And so on until the vehicle reaches its destination.

A compact representation of the congestion of urban transport networks can be made by using matrix analysis, as shown in [22]. The elements of such a matrix are values that represent the ratio of the average speed of traffic on a road segment j to the speed of free movement on that segment.

The real-time interaction between each vehicle and the TMC is crucial for the continuous programmatic route correction. The program [19] creates specific individual routes for each vehicle: each vehicle is tracked by an individual program, as the A-star algorithm creates an optimal route between two specified graph vertices. In other words, this algorithm is tied to the geometric coordinates of the origin-destination pair.

If you run the program [19], the results will usually be different each time you run it. One of the possible variants of the program is presented below. The program facilitates the procedure of comparing the geoposition of cars on an electronic map with road segments. This means that the Dijkstra or Floyd algorithms are less effective in this case.

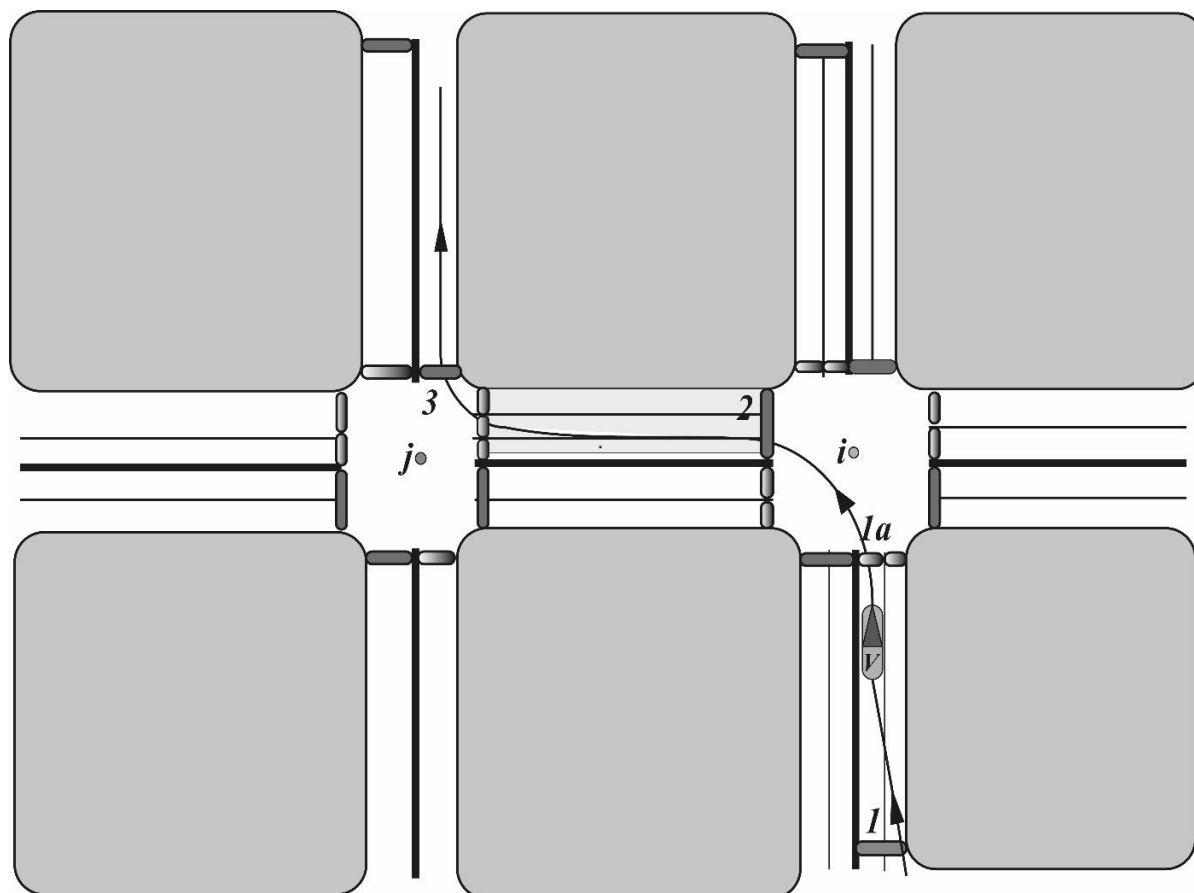


Fig.3. The trajectory of the car's route is shown. The data from the electronic map sensors are sent to the TMC (Fig. 2). The numbers 1, 2 and 3 on the input sensors mark the positions where the vehicle routes are recalculated in order to be synchronized with the new traffic realities.

Listing 1

Course-1: [a2_3-Broad Street, a3_2-Broad Street, a2_5-Heroes Street, a5_4-Short Street, a4_7-Long Street, b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]

Course-2: [a3_2-Broad Street, a2_5-Heroes Street, a5_4-Short Street, a4_7-Long Street, b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]

Course-3: [a2_5-Heroes Street, a5_4-Short Street, a4_7-Long Street, b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]

Course-4: [a5_4-Short Street, a4_7-Long Street, b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]

Course-5: [a4_7-Long Street, b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]

Course-6: [b7_9-Trump Street, a9_13-Welly Street, a13_14-Cherry Street]

Course-7: [a9_13-Welly Street, a13_14-Cherry Street]

Course-8: [a13_14-Cherry Street]

In general, the specific appearance of Listing 1 depends on the outcome of the *public static double R()* function that simulates the weights of the lanes.

The real route looks like this: a2_3-Broad Street → a3_2-Broad Street → a2_5-Heroes Street → a5_4-Short Street → a4_7-Long Street → b7_9-Trump Street → a9_13-Welly Street → a13_14-Cherry Street.

Between consecutive Courses there is a delay phase, which simulates the time required for a car to travel between two adjacent input sensors located along the vehicle route. In [19], the delay phase is programmatically implemented by using the *Thread.sleep((long) (t2-t1))* method. For a visual interpretation, let us refer to Fig. 3, where the car crosses two neighboring input sensors – 1,2 and 3 on its route. The time between the moments of crossing these sensors is the delay phase. Until the vehicle reaches the new lane, the program "waits". As soon as the vehicle crosses the input sensor of this new

lane, the program immediately plots an updated route, which, in turn, is calculated anew according to a changing traffic situation.

The tracked driver is given verbal instructions (as in conventional GPS navigation) to move to the next lane along the route (these lanes are highlighted in Listing 1). Thus, the route previously calculated is connected to the new one (that is very important!), since the new route calculation uses the second position in the route previously calculated. Thus, dynamic software approximation of urban traffic is used. That allows the program to account for the dynamics of city traffic. Therefore, the routes for all vehicles in the metropolis will be plotted, which is time-optimal. As a result, even if traffic jams do not disappear, their probability will be significantly reduced.

Let us now describe some of the features of the program presented in [19], which consists of three classes – Astar, Node, and Edge. The first class is very large, because it contains the constructors of all nodes of the graph of a type

$$\text{static Node } li_j = \text{new Node}("li_j - \text{Liberty avenue}", 1250). \quad (12)$$

Number 1250 represent the heuristic distance measured in a straight line from the intermediate node of the graph to the final node of the route. Each node of the graph is extended, i.e., it covers the adjacent lane; for example, a node $a1_2$ includes node 1 itself and lane 1_2 , which comes from this node. All that is needed in order to register cars, which, of course, are located in lanes, not at intersections. And when a real route is plotted, the starting position of the car, as well as its finish position, are set on the start position of the graph (Fig. 1).

In the *public static void initGraph()* method of the Astar class, constructors of the type $ni_j.adjacencies = \text{new Edge}[]$ are specified, which form the interaction between the edges of the graph. When forming incident edges by using such constructors, traffic rules are necessarily taken into account. For example, you cannot turn from the lane $a6_5$ (Fig. 1) to the lane $a5_2$ as it is prohibited by traffic rules. Therefore, those constructors include only such elements of the route that are not prohibited by traffic rules.

The *public static void AstarSearch (Node source, Node goal)* method of the Astar class implements the main mission – the implementation of the A-star algorithm. This algorithm uses the heuristic function $f(n) = g(n) + h(n)$, where n is the current node, $g(n)$ is the distance from the starting position of the vehicle to the current node, calculated along the arcs of the graph, and $h(n)$ is the distance from the current node of the graph to the end position, calculated along a straight line. Heuristic distances are set in the program [18] by constructors of the type (12). The peculiarity of this algorithm is that it deliberately cuts off unpromising routes by introducing a heuristic distance. Additionally, that algorithm has a minor, namely linear, algorithmic complexity.

In the field of the *Node* class, class variables are declared, and the *public Node(String val, double hVal)* method returns the value of the *value* variable.

The *Edge* class returns the values of the *target* and *cost* variables, which are used when plotting a route in the *public static void AstarSearch (Node source, Node goal)* method of the A-star class.

Thus, the study creates a realistic prospect for solving the problem of congestion through the use of a special software algorithm focused on laying optimal routes and graphs, which, in turn, model the city's transportation network. The use of an electronic map of the city allows the GPS coordinates of each vehicle to be projected onto it. Thus, the city's transport network is under the full control of the TMC, which has a real opportunity to interact with each vehicle and constantly adjust its route, choosing the t-optimal one.

4. Conclusions

1. A multigraph model that reproduces the transport network of a city simulates the real-life lanes. Each arc of the graph receives a dynamic weight that synchronously changes in accordance with changes in traffic.
2. The use of sensors located near each intersection allows for the registration of traffic flows on the electronic map, therefore, the entire transportation network of the city is under the control of the TMC. That makes it possible to track all traffic changes and calculate dynamic optimal routes for each vehicle. As a result, urban traffic enters an equilibrium state – the Nash equilibrium state making it possible to eliminate traffic jams.
3. The working program module [19] that paves the time-optimal routes in the graph, and hence in the real transportation network has been developed. It uses the heuristic A-star algorithm, a powerful

computational method of graph theory. That makes it possible to synchronize traffic flows and bring them to a qualitatively new level.

4. The study operates only with electronic services, which allows implementing the proposed method of creating optimal urban routes without significant economic costs.

5. Since all cars participating in traffic at a certain point in time will move under the control of the TMS along t-optimal routes, there will be a complete synchronization of traffic flows, which will lead to a qualitatively new state of urban traffic. That, in turn, will lead to the disappearance of traffic jams on city highways. Moreover, each driver will arrive at their destination in the shortest possible time.

REFERENCES

1. T. Xu , B. Ran , Y. Cui, Dynamic user optimal route choice problem on a signalized transportation network *Transportation Engineering* 13, 2023. <https://doi.org/10.1016/j.treng.2022.100153>
2. P. Nikolyyuk, T. Neskorođieva, E. Fedorov, E. Chioma. Intellectual algorithm implementation for megacity traffic management, *CEUR Workshop proceedings, Information Technology and Interactions*, 2021, V.2845, <https://cutt.ly/WwAwZZAp>
3. P. Dutta, S. Khatua, S. Choudhuri. DB-Corouting: Density Based Coordinated Vehicle Rerouting in Smart Environment, *International journal of intelligent transportation systems research* V.19, 2021. doi:[10.1007/s13177-021-00261-6](https://doi.org/10.1007/s13177-021-00261-6) .
4. M. Hashemi, H. A.Karimi,. A weight-based map-matching algorithm for vehicle navigation in complex urban networks. *Journal Intelligent Transportation Systems: Technology, Planning, and Operations* V. 20, 2016, DOI:[10.1080/15472450.2016.1166058](https://doi.org/10.1080/15472450.2016.1166058)
5. J. Pan, I. S. Popa, K. Zeitouni, C. Borcea, Proactive vehicular traffic rerouting for lower travel time. *IEEE Transactions on Vehicular Technology*, 62((2013). doi:[10.1109/TVT.2013.2260422](https://doi.org/10.1109/TVT.2013.2260422)
6. T. Sobral, T. Galvao, J. Bordes, Visualization of Urban Mobility Data from Intelligent Transportation Systems. *Sensors*, V.19, 2019, <https://doi.org/10.3390/s19020332>.
7. M. Quddus, S. Washington, Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. *Transportation Research Part C: Emerging Technologies*, V.55, 2015, <https://doi.org/10.1016/j.trc.2015.02.017>
8. H.Majid, C. Lu, H. Karim, An integrated approach for dynamic traffic routing and ramp metering using sliding mode control. *Journal of Traffic and Transportation Engineering (English Edition)*, V.5, 2018, <https://doi.org/10.1016/j.jtte.2017.08.002>
9. S. Rahimpour, R. Moeinfar, S. M. Hashemi, Traffic prediction using a self-adjusted evolutionary neural network, *J. Mod. Transport.* V.27, 2019, DOI: [https:// 10.1007/s40534-018-0179-5](https://doi.org/10.1007/s40534-018-0179-5)
10. A. Emami, M. Sarvi, S.A. Bagloee, Using Kalman filter algorithm for short-term traffic flow prediction in a connected vehicle environment, *J. Modern Transport.* 27, 2019, <https://doi.org/10.1007/s40534-019-0193-2> .
11. A. Pompigna, F. Rupia,. Comparing practice-ready forecast models for weekly and monthly fluctuations of average daily traffic and enhancing accuracy by weighting methods, *Journal of Traffic and Transportation Engineering (English Edition)*, V.5, 2018, <http://dx.doi.org/10.1016/j.jtte.2018.01.002> .
12. S. Chavhan, P. Venkataram, Prediction based traffic management in a metropolitan area, *Journal of Traffic and Transportation Engineering (English Edition)*, V.7, 2020). <https://doi.org/10.1016/J.JTTE.2018.05.003> .
13. X. Liang, S. Ilgin Guler, V. Gayan,. An equitable traffic signal control scheme at isolated intersections using Connected Vehicle technology, *Transp. Research Part C*, V.110, 2020, <https://doi.org/10.1016/j.trc.2019.11.005> .
14. C. Cintrano, J. Ferrer, M. López-Ibáñez, E. Alba. Hybridization of Evolutionary Operators with Elitist Iterated Racing for the Simulation Optimization of Traffic Lights Programs, *Evolutionary computation*, 31, 2023, https://doi.org/10.1162/evco_a_00314
15. C. Xu, J. Xu,. Intelligent terminal based intelligent traffic light system and method. Pat. CN104575066, 2015, <https://patents.google.com/patent/CN104575066A/en>.
16. S. S. Smith, G. G. Barlow, X.-F. Xie (2017). Smart and scalable urban signal networks: methods and systems for adaptive traffic signal control. Pat. US 9,830,813 B2. <https://www.wiomax.com/team/xie/schic/>

17. H. Yu, R. Jiang, Z. Zheng Li, R. Liu, X. Chen, Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives. J. of Intel. Transp. Systems, V.127, 2021, <https://doi.org/10.1016/j.trc.2021.103101>
18. S. Panda, A. M. Patki, K. Hushing, Traffic Management Using Swarm Intelligence and Route Selection Using Android Application// International Journal of Engineering and Innovative Technology (IJEIT), V.5, 2015, https://www.ijeit.com/Vol%205/Issue%206/IJEIT1412201512_11.pdf
19. P.K. Nikolyyuk, A-Star algorithm, 2023, https://github.com/pknikolyuk/A-Star_algorithm/blob/master/src/Astar.java
20. A. M. T. Emtenan, A. Haghghat, M. Shields, J. Shaw, P. Hawley, A. Sharma, C. M. Day, Exploratory Regression Models for Estimating Right-Turn-on-Red Volume on Exclusive Right-Turn Lanes at Signalized Intersections. Transportation Research Record, V.2677, 2022, <https://doi.org/10.1177/03611981221116370>
21. J. Tan, X. Shi, Z. Li, K. Yang, N. Xie, H. Yu, L. Wang, Z. Li, Continuous and Diskrete-Time Optimal Controls for an Isolated Signalized Intersection. Journal of Sensors, Article ID 6290248, 2017, <https://doi.org/10.1155/2017/6290248>
22. X. Ma, Y. Li, P. Chen. Identifying spatiotemporal traffic patterns in large-scale urban road networks using a modified nonnegative matrix factorization algorithm. Journal of Traffic and Transportation Engineering (English Edition), V.7, 2020, <http://dx.doi.org/10.1016/j.jtte.2018.12.002> .

Маршрутизація транспортного засобу з оптимальним за часом динамічним міським шляхом руху

**Ніколюк
Петро Карпович**

*доктор фізико-математичних наук, професор кафедри
інформаційних технологій Донецького національного університету
імені Василя Стуса, вул. 600-річчя, 21, Вінниця-21, Україна, 21021
e-mail: p.nikolyuk@donnu.edu.ua
<https://orcid.org/0000-0002-0286-297X>*

Ключова мета цього дослідження - синхронізувати транспортні потоки, оптимізувати використання транспортних артерій по всьому місту, запобігти заторам і супроводжувати кожен транспортний засіб до місця призначення так, щоб час, витрачений на поїздку, був мінімальним. В результаті загальний час, проведений автомобілями в дорозі, значно скоротиться, а екологічні показники відповідно покращаться.

Об'єктом дослідження є транспортна мережа міста, представлена у вигляді зваженого орієнтованого неплоского мультиграфа (ЗОНМ). Ключова перевага використання теорії графів для побудови оптимальних маршрутів базується на наступних обставинах: 1). теорія графів розробила багато алгоритмів побудови оптимальних маршрутів (алгоритм Дейкстри, алгоритм Флойда-Уоршалла, алгоритм А-зірки та ін.); 2). теорія графів є теоретичною і практичною основою логістики, в тому числі міського транспорту. Для побудови маршруту в такому мультиграфі використовується алгоритм А-зірка, який прокладає оптимальний за часом (t-оптимальний) маршрут між двома обраними вершинами графа.

Дослідження створює реальну перспективу вирішення проблеми заторів за рахунок використання спеціального програмного алгоритму, орієнтованого на прокладання оптимальних маршрутів та використання графів, які, в свою чергу, моделюють транспортну мережу міста.

Принциповим питанням є представлення транспортної мережі міста у вигляді електронної карти та відображення на ній GPS-ідентифікаторів транспортних засобів, що беруть участь у дорожньому русі. Відображення "транспортний рух міста → електронна карта" дає можливість отримати дані про рівень завантаженості транспортної мережі. Використання електронної карти міста дозволяє спроектувати на неї GPS-координати кожного транспортного засобу. Таким чином, транспортна мережа міста знаходиться під повним контролем центру управління транспортом (ЦУТ), який має реальну можливість взаємодіяти з кожним транспортним засобом і постійно коригувати його маршрут, обираючи t-оптимальний. Коригування маршруту здійснюється через канал General Packet Radio Service (GPRS) у вигляді голосових команд, як у звичайній GPS-навігації. Однак специфіка полягає в наступному: 1). навігація здійснюється в режимі онлайн; 2). прокладаються t-оптимальні маршрути з урахуванням дорожньої ситуації в кожен момент часу.

Таким чином, розроблено великомасштабну транспортну міську дорожню мережу у поєднанні з комп'ютерною програмою. Робота має прикладний характер, а її результати можуть бути використані для ефективного регулювання дорожнього руху в мегаполісах з метою мінімізації проїзду кожного транспортного засобу за обраним маршрутом.

Ключові слова: зважений орієнтований неплоский мультиграф, GPS-навігація, алгоритм A-star, t-оптимальний маршрут, інтелектуальний трафік.

УДК (UDC) 681.58

**Яланецький
Валерій Анатолійович***старший викладач
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Берестейський проспект (Перемоги), 37, Київ-56, Україна, 03056
e-mail: v.yalanetskyi@gmail.com
<https://orcid.org/0000-0001-6163-0258>*

Автоматизована система проклеювання деталей паперових пакетів

Актуальність. Подарункова та сувенірна продукція подається зазвичай у паперових пакетах. Дослідження і розробка практичних рішень щодо автоматизації процесу проклеювання деталей пакетів є актуальною науковою проблемою та інженерною задачею. У статті розглядається питання новітнього програмно-технічне рішення з автоматизації нанесення клею на паперові вироби різних розмірів, здебільшого незначними партіями.

Метою роботи є висвітлення інженерних рішень, що дають змогу з мінімальними витратами для бюджету підприємства, розробити автоматизовану лінію проклеювання деталей паперових пакетів.

Методи дослідження. Досліджено об'єкт автоматизації, зокрема проведено аналіз лінійки розмірів паперової продукції та продуктивності виготовлення паперових пакетів в цехах ручного збирання. Розглянуто питання побудови автоматизованих конвеєрних ліній для виконання технологічних операцій проклеювання паперових деталей для їх подальшого збирання в пакет.

Результати. Проведено дослідження останніх наукових публікацій, що підтвердили актуальність проєктування низькобюджетних автоматизованих систем для вітчизняних виробників паперових пакетів. Наведено опис геометричних параметрів пакету та траєкторії проклеювання. Розроблено структурну схему автоматизованої системи проклеювання деталей пакета та алгоритмічне забезпечення керуючої програми контролера. Наведений список основних програмно-технічних засобів системи. Визначені області допустимих значень параметрів пакета. Наведений обрахунок залежності швидкості руху каретки модуля лінійного переміщення від часу. Представлено результат фізичного впровадження системи в цеховому приміщенні підприємства-замовника.

Висновки. Впровадження автоматизованої системи забезпечить зменшення активних працівників у виробничому цеху, які дотепер виконують ручні операції проклеювання. В найближчих планах максимально швидко інтегрувати в автоматизовану систему машину, що буде подавати набір пакетів на конвеєрне полотно. Планується вдосконалення механічної конструкції над полотном конвеєра, а саме будуть додаватись механічні пристосування задля забезпечення процесу автоматичного відкривання та закривання пакету протягом його переміщення по конвеєру..

Ключові слова: автоматизована система, автоматизація, нанесення клею; паперові вироби, пакет.

Як цитувати: Яланецький В. А. Автоматизована система проклеювання деталей паперових пакетів. *Вісник Харківського національного університету імені В.Н. Каразіна, сер. «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*. 2023. вип. 57. С.55-64. <https://doi.org/10.26565/2304-6201-2023-57-05>

How to quote: V.A. Yalanetskyi, "Automated system for gluing paper bag parts". *Bulletin of V.N. Karazin Kharkiv National University, series "Mathematical modelling. Information technology. Automated control systems*, vol. 57, pp.5-64, 2023. <https://doi.org/10.26565/2304-6201-2023-57-05>

Вступ

Виробництво будь-якої цивільної, промислової та військової продукції не можливе без технологічних ліній, автоматизованих систем керування, локальних систем збору даних та програмно-технічних засобів автоматизації. Маркетингова галузь вимагає від виробничих потужностей подарункової, рекламної та сувенірної продукції. Подарункова та сувенірна продукція подається зазвичай у певній тарі, себто коробках, кейсах, пакетах. Зважаючи на критичну екологічну проблему на Землі щодо надлишку у смітті пластикової тари, маркетологи надають перевагу паперовій тарі (пакетам). В сфері виробництва сувенірної паперової продукції ключовим компонентом збирання паперових деталей в єдиний виріб є клейовий (клеючий) матеріал (далі клей). Застосування клеїв у виробництві паперових виробів значно знижує собівартість готового продукту порівняно з іншими технологіями, і не вимагає підготовку висококваліфікованих фахівців. З часів появи технології проклеювання цей процес виконується досі вручну, де якість послуги має найвищий пріоритет аніж кількість продукції.

Протягом нанесення клею на цільові поверхні деталей паперових виробів підручними засобами (проклеювання вручну) дуже складно досягти рівномірності клеєнанесення. Автоматизація клеєнанесення виключає людський фактор з цієї технологічної операції, що, в свою чергу, забезпечує підвищення якості готової продукції. Автоматизація технологічних процесів у сфері виробництва паперових виробів достатньо давно досліджена та широко представлена готовими технічними рішеннями. Виробничий процес може складатись як зі статичного автоматичного апарату, що виконує типову технологічну операцію, так і з технологічної конвеєрної лінії. Здебільшого, технологічні лінії конструктивно та фізично мають у складі спеціалізовані машини-автомати, станки, які розроблені суто на один тип технологічної операції, задля виробництва певної деталі паперового виробу. Станки для нанесення клею на паперовий виріб (папір чи картон) – це агрегати, за допомогою яких суттєво спрощується процес нанесення клею при формуванні пакувальної тари у порівнянні із проклеюванням вручну. Загальна технологія в таких станках передбачає, що паперові вироби пропускаються через роликіві (вальцеві) системи, де один із валів передає шар (іноді підігрітого) клею на поверхню виробу. Товщину шару клею можна регулювати механічно. Наступним типом клеєнаносних пристроїв є (клеєналивні) машини [1], на рамі яких горизонтально закріплена насадка (лійка або сопло) для клеєнанесення з електронасосом для безперервної подачі рідких клеїв (клейовий пістолет) під тиском.

Постановка проблеми. Клеєналивну машину з клейовим пістолетом можна інтегрувати у конвеєрну технологічну лінію. Найпростіші клейові пістолети, що подають розігрітий клей на поверхню паперового виробу, не переміщуються у просторі, позаяк мають статичне положення. А паперові вироби переміщуються конвеєром під клеєналивною машиною, клей подається безперервним потоком на поверхню паперового виробу. Ширина потоку клею може бути механічно (або електрично керовано) змінена. Безперервність потоку клею гарантує рівномірність подачі необхідної кількості на всю поверхню незалежно від геометричних вад [2]. Задача автоматизованої системи полягає у рівномірному нанесенні (підтримка густини та маси) клею на поверхню паперового виробу за чіткою, заздалегідь визначеною, геометрією. Проблема проклеювання поверхонь полягає у тому, що коли цільові паперові вироби (пакети) рухаються конвеєром, тоді такий динамічний процес вимагає точного позиціонування клейових пістолетів та задовільної траєкторії нанесення клею на поверхню виробів.

Ринок промислового обладнання представлений широким спектром готових технічних рішень, зокрема станків та клейових машин, а також комплексними роботизованими рішеннями з автоматизації нанесення клеїв на паперові вироби. Значна більшість цих систем орієнтована на досягнення максимальної виробничої продуктивності за одиницю часу (величезні однотипні партії). Переключитись на проклеювання іншої геометрії (не типових пакетів) такими системами не можливо, доведеться придбати нове обладнання або зупиняти виробництво для переналаштування лінії. Універсальними властивостями, аби швидко змінювати параметри вхідних виробів, наявні системи не володіють. Українських виробників подібних автоматизованих систем нанесення клею на жаль на ринку взагалі не спостерігається. Із урахуванням складної економічної ситуації в Україні, не завершеної пандемії та війни з росією, розробка таких автоматизованих систем доволі клопітливий процес, як із наукової, так і з практичної сторін.

Дослідження і розробка практичних рішень щодо автоматизації процесу проклеювання деталей пакетів є достатньо актуальною науковою проблемою та інженерною задачею. У статті подається новітнє програмно-технічне рішення з автоматизації нанесення клею на паперові вироби різних розмірів. Тим самим започатковується розробка таких універсальних вітчизняних систем, що не поступатимуться закордонним аналогам. Запропонована універсальна автоматизована система нанесення клею здебільшого орієнтована на незначні партії замовників сувенірних та рекламних паперових пакетів.

Аналіз останніх досліджень і публікацій. Технології виробництва продукції легкої промисловості передбачають використання з'єднувальних компонентів на базі клейових сумішей, наприклад, галузь виробництва взуття, одягу, іграшок тощо. На ринку представлені готові роботизовані комплекси нанесення клеїв на цільові поверхні, наприклад, у взуттєвому виробництві. У статті [3] описано вдосконалену роботизовану машину, яка наносить клей на робочі поверхні, зокрема, розроблено автомат для приклеювання верху та гумових вставок на підшву взуття. Система характеризується наявністю пристрою технічного зору, задля визначення положення підшов й геометрії деталей взуття. Роботизований механізм керує клейовим пістолетом з допустимою похибкою позиціонування. Також враховується перепад висоти, щоб

утримувати клейовий пістолет на однаковій відстані від підшви по всій траєкторії проклеювання. Така система має статичний характер, позаяк взуттєві деталі не рухаються, а переміщається лише сопло клейового пістолету. У роботі [4] автори презентують конструкцію та автоматизовану систему виробництва паперових пакетів, щоб зменшити вартість їх виробництва у порівнянні із поліетиленовими пакетами. Проект реалізований на базі власного мікроконтролерного пристрою, а не серійного промислового логічного контролера. Ідея авторів полягає у тому, що деталі паперових пакетів подаються конвеєром до місця розташування клейової установки. Після чого, конвеєр із деталлю пакета зупиняється, далі відбувається нанесення клею на поверхню деталі через сопла клейових пістолетів. Після проклеювання конвеєр знову вмикається в роботу і проклеєна деталь пакету рушає до збірного етапу. Велику увагу приділено питанню електромеханічних розрахунків автоматизованої системи та розробці конвеєрного столу від етапу проектування в САПР до фізичного виготовлення деталей столу і монтажних робіт. Автори роботи [5] наводять опис власної розробки автоматизованої машини для виготовлення паперових пакетів. Відмінність від попередньої роботи полягає у тому, що ця машина проклеює пакети без конвеєрної лінії. Але машина авторів здатна розкласти пакет, проклеювати його деталі та скласти ці деталі у фінальний пакет. Звичайні паперові пакети потребують спеціального дорогого паперу, що значно збільшує потім його загальну вартість. Це основна причина, через яку промисловістю виготовляється паперових пакетів менше ніж пластикових. Тому початкові інвестиції в машину, що виготовляє дорогі паперові пакети априорі завеликі. У роботі [6] автори пропонують бюджетну розробку, яку може дозволити собі навіть невелика родина для дрібносерійного виробництва пакетів. Паперовий пакет виготовляється зі звичайних газет, що значно зменшує вартість готового пакета. В розробці [7] автори пішли далі та запропонували дешеву портативну автоматизовану машину для виготовлення паперових пакетів із дешевої паперової вторинної сировини (автори підкреслюють, що можна і з будь-якої макулатури). Система також може бути впроваджена у невеликому виробництві паперових пакетів.

Виходячи із результатів аналізу публікацій можна зробити висновок, що автоматизація виробництва паперової продукції широко представлена у світі від малого, середнього і до великого бізнесу. Є великі науково-дослідні групи та поодинокі аматори, що проєктують подібні системи від складних роботизованих комплексів до невеликих автоматів. Але значна більшість підприємств розгортають автоматизовані лінії так би мовити «під ключ», у складі готових машин для виробництва деталей пакетів, їх проклеювання та пакування. Це надає підприємствам певних гарантій якості, подальшого обслуговування й супроводження та мінімізації ризиків, у випадку відмови в роботі обладнання. Великий бізнес, мета якого випуск багатотиражних пакетів, цілком у змозі придбати фірмове дороговартісне обладнання. А от для малого та середнього бізнесу придбання фірмових автоматизованих ліній – невідомою задачею. Тому, аби конкурувати на ринку із крупним бізнесом, невеликим підприємствам варто розробляти власні технічні рішення щодо автоматизації процесів виробництва паперових виробів.

Об'єктом дослідження є технологічний процес нанесення клею на деталі паперових пакетів. **Предметом дослідження** є автоматизовані конвеєрні лінії проклеювання та збирання паперових деталей у цільовий пакет. Вітчизняне підприємство ТОВ «Сувенірні вироби» (м. Бориспіль) у лиці його директора, та науково-дослідний підрозділ кафедри інформаційних систем та технологій факультету інформатики та обчислювальної техніки (КІТ імені Ігоря Сікорського), уклади партнерську угоду щодо інженерно-наукової співпраці на програмно-технічній базі підприємства. Силкові електромеханічні частини системи та конвеєрний стіл розроблено та змонтовано фахівцями інженерного підрозділу підприємства. Деталь пакета (заготовка) виробляється в іншому цеху, тож його виготовлення не входить в постановку задачі. Згідно вихідного технічного завдання було сформовано мету, задачі та календарний план робіт щодо програмних рішень в системі автоматизації процесу проклеювання деталей паперових пакетів.

Метою дослідження є висвітлення інженерних рішень, що дають змогу з мінімальними витратами для бюджету підприємства, розробити інноваційну автоматизовану лінію проклеювання деталей паперових пакетів із урахуванням найкращих світових практик, та побудовану на сучасній програмно-апаратній базі. **Постановкою задачі** є розробка та впровадження у виробництво автоматизованої системи конвеєрного типу, що виконуватиме функції нанесення клею за геометричною траєкторією згідно фактичних параметрів пакету.

Опис об'єкту автоматизації

Зважаючи на результати пошуку, огляду та дослідження практичних рішень у світі, щодо автоматизації клеєнанесення на деталі паперових виробів, можна зазначити, що такий технологічний процес є складовим етапом виробництва сувенірної продукції, зокрема пакетів. Попередньо підготовлені (не проклеєні) паперові деталі (пакети) підлягають процесу так званого «розкриття» (розкладання), далі нанесенню клею і на фінальній стадії «закриття» (складання), а значить приклеювання деталей в суцільний пакет. Серед всіх етапів в межах цього дослідження буде розглянуто саме процес нанесення клею. Механічні пристосування та пристрої, що забезпечують відкриття та закриття пакету не досліджувались. Пакети пересуваються конвеєрною лінією рівномірно та прямолінійно. Геометричні параметри пакета (рис. 1) наступні:

- довжина пакета у зібраному стані Z , мм;
- ширина виступу для формування дна пакета k , мм;
- ширина згинів для звичайних пакетів, $n = k$, мм;
- ширина пакета у зібраному стані $S = 2k$, мм;
- пряма кромка проклеювання, $m = Z - 2n$, мм;
- висота пакета L , мм;
- кут α у звичайних пакетів дорівнює 45 градусів.

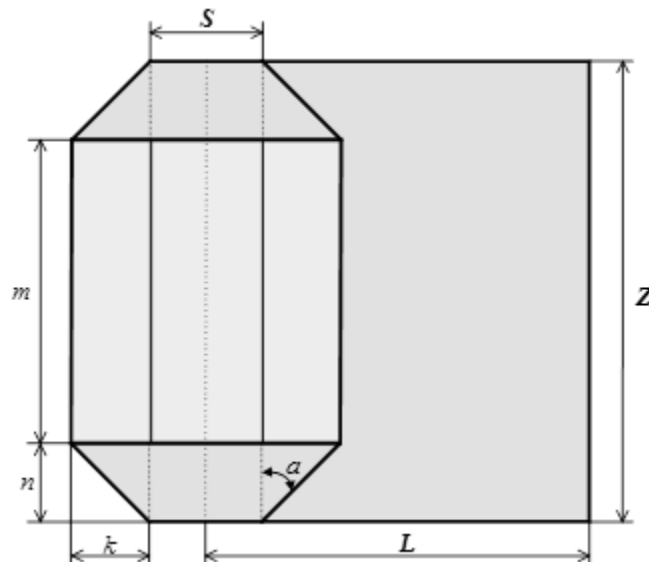


Рис.1 Геометричні параметри пакета

Пакети притискаються валками, тож швидкість їх пересування дорівнює швидкості конвеєрної лінії. Усі пакети мають однакову паперову структуру та розміри. Проклеюванню підлягають різні партії, що мають різні параметри паперових пакетів. Пакети поступають на конвеєр окремими партіями по 50-500 шт. В межах партії всі пакети однакові, та мають наперед відомі параметри. Швидкість лінії стала під час проходження партії однотипних пакетів. Пакети опиняються на конвеєрі на відстані не менш ніж 100 мм один від одного. Деякі партії пакетів мають підсилювач дна – спеціальний картонний прямокутник у відповідності до розмірів пакету, що слугує прокладкою між дном пакету та його вмістом. Технічні вимоги до автоматизованої системи проклеювання деталей пакетів узгоджувались із замовником, що фінансує цей проєкт. Попередньо розроблене технічне завдання до системи включає такі головні вимоги:

- нормативна швидкість проклеювання - 30 пакетів на хвилину;
- швидкість конвеєра (V) - від 0 мм/с до 1500 мм/с;
- похибка від траєкторії клеєнанесення не повинна перевищувати 5 мм;
- висота пакета (L) від 10мм до 900мм;
- ширина пакета (S) від 10мм до 2000 мм;
- довжина пакета (Z) визначається датчиком (G) наявності пакету на конвеєрі;
- час проходження пакета через датчик G від 0 до 30 с.;
- проклеювання із урахуванням підсилювача дна.

На вхід конвеєрної лінії подається стандартний пакет. В залежності від наявності підсилювача дна є два способи проклеювання. За наявності підсилювача дна, деталь пакета потрібно проклеїти відповідною траєкторією (рис. 2а, штрихпунктирна лінія): два діагональних відрізки (кромки) та один відрізок паралельний довжині пакету з лівої та правої сторони. Для даного способу проклейки немає різниці який край загинати першим.

За відсутності підсилювача дна, траєкторія проклеювання відрізняється, зокрема, на правій кромці проклеюються тільки діагональні відрізки (дана кромка повинна надалі приклеюватись першою), в свою чергу лівий відрізок проклеюється по всьому контуру (на рис. 2. спочатку буде загинатись права частина, а вже потім ліва). Кількість клею, що подається клейовим пістолетом на поверхню деталі пакета, стала, та задається відповідним тиском подачі клею. Розмір краплі клею має задовольняти надійному склеюванню поверхонь після їх загинання та склеювання за допомогою прикінцевих валків на конвеєрі. А також забезпечити оптимальне використання клейової суміші без його витікання за межі проклеєних деталей пакету.

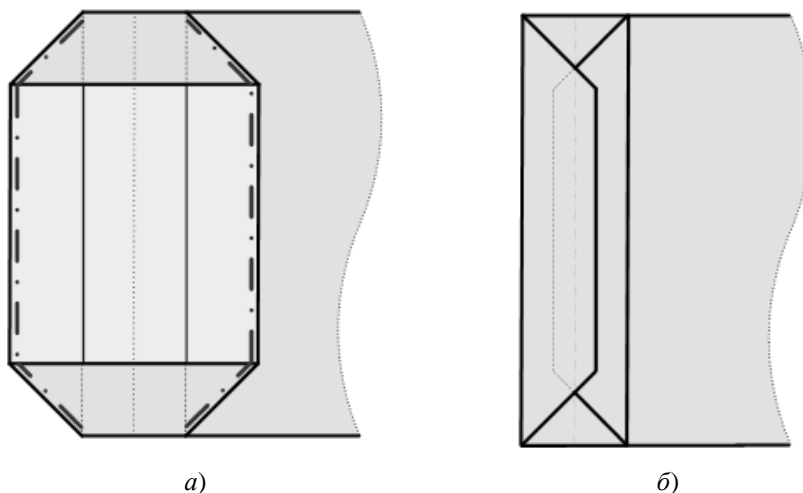


Рис. 2 Траєкторія нанесення клею з підсилювачем дна (а) та складання деталей пакету (б)

Структурна схема автоматизованої системи

Згідно технічного завдання та вимог до автоматизованої системи розроблено структурну схему (рис. 3), яка складається із наступних компонентів:

- ПЛК – промисловий програмований логічний контролер;
- ПВВ – пристрій (модуль) вводу даних;
- ПВ – пристрій виводу даних;
- ДКЛ1, ДКЛ2 – датчики крайнього лівого положення ЛМ;
- ДКП1, ДКП2 – датчики крайнього правого положення ЛМ;
- ДШК – датчик лінійної швидкості конвеєру;
- ДП – датчик наявності пакету на конвеєрі;
- ЛП1, ЛП2 – модулі лінійного переміщення;
- СД1, СД2 – серводрайвери для керування сервомоторами ЛП1 та ЛП2.

Центральний програмований логічний контролер (ПЛК) за допомогою пристрою введення (ПВВ) отримує дані від периферійних пристроїв (датчики) та передає через пристрій виведення (ПВ) керуючі сигнали до виконавчих елементів (лінійні модулі та клейові пістолети). Стислий опис периферійних компонентів системи наведений у таблиці 1. Особливості моделей клейових пістолетів і машини, що подає безпосередньо до них розігрітий клей, не впливають на розробку та програмування контролеру, тому не зазначені у таблиці програмно-технічних засобів.

Модуль виводу також формує ШІМ-сигнали керування до серводвигунів модулів лінійного переміщення, які пересувають клейові пістолети. Дані між модулями та ПЛК передаються за промисловим протоколом MODBUS.

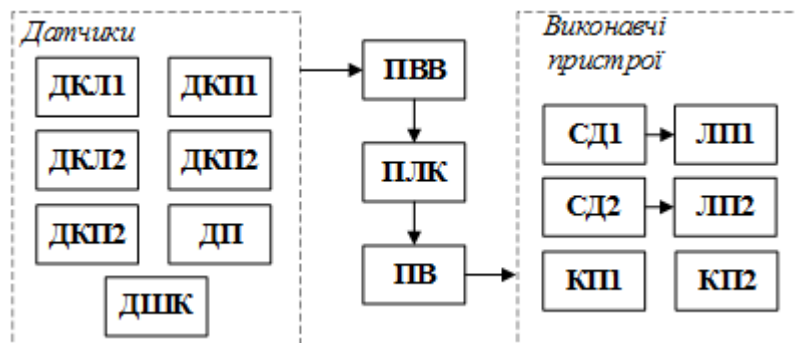


Рис. 3. Структурна схема автоматизованої системи проклеювання деталей пакета

Лінійний модуль переміщення має 1000 мм. робочого ходу каретки, що дозволяє розмістити на конвеєрі найбільш поширені за розміром пакети. Відповідно до вимог провідних виробників, висота пакету не перевищує 460 мм, а ширина 140 мм [8]. Таким чином максимальна довжина деталі стандартного пакета не перевищуватиме $460+140/2+20=550$ мм, що цілком задовольняє габаритам лінійного приводу. За електромеханічне керування лінійним модулем переміщення відповідає комплект із сервомотора та серводрайвера. Максимальна швидкість ходу каретки лінійного модуля складає 1500 мм/с. Точність позиціонування дорівнює 0.1 мм. Основні режими керування: напрям руху та швидкість руху за ШІМ-сигналом. Для вимірювання сигналів крайнього положення кареток лінійного модуля було застосовано індуктивні датчики. Наявність пакету на конвеєрі фіксує датчик дифузійного відбиття (датчик сприймає та аналізує світло, відбите від поверхні матеріалу, що проходить повз нього).

Таблиця 1. Програмно-технічні засоби системи

№	Компонент	Версія/Модель
1	Панельний ПЛК	ОВЕН СПК107
2	Модуль дискретного виводу	МУ110-8К
3	Модуль дискретного вводу	МВ110-16ДН
4	Модуль лінійного переміщення	TBD60-1000
5	Комплект із сервомоторів та драйверів	130ST-M15015
6	Індуктивні датчики крайнього положення	LJ12A3-4-Z/BX
7	Датчик наявності пакета на конвеєрі	OGT100
8	ІДЕ програмування ПЛК	CoDeSys V3.5

Фізичне розташування периферійних пристроїв на конвеєрі зображено на схематичному рис. 4. Цільовий пакет рухається від місяця (наразі ручної) подачі у сторону клейових пістолетів. Система окрім заданих користувачем параметрів, потребує обрахунок допоміжних параметрів, що залежать від конструктивного положення на конвеєрі периферійних пристроїв:

- D1 – відстань між ДП та КП1 (від 5 до 200 мм);
- D2 – відстань між КП1 та КП2 (від 200 до 400 мм);
- J – відстань між обмежувальною планкою подачі пакету та крайнім положенням КП1/КП2 (від 5 до 300 мм).

Область допустимих значень параметрів автоматизованої системи

Головним предметом математичного інтересу у даній системі виступає обрахунок траєкторії нанесення клею (проклеювання) в залежності від геометричних параметрів пакета. Для користувача, згідно з рис. 1, доступні для введення наступні параметри пакета: S , L , Z (для порівняння із підрахованим програмно), k , n , m . Кут α залишимо за замовчуванням, що дорівнює 45 градусів. Очевидно, що для задачі проклеювання висота пакета L не впливає на траєкторію, тому для обрахунків її не використовуємо.

Для початку визначимо область допустимих значень, себто опишемо неможливі параметри пакету. Даний етап безпосередньо допоможе при програмуванні контролера, оскільки дозволить відслідковувати некоректність введених оператором даних до початку запуску автоматизованої системи в роботу. Пряма кромка проклеювання $m \geq 0$. Якщо $m < 0$, тоді дно пакету стає занадто

гроздким та приймає неправильну форму. Ширина виступу для формування дна пакета $k \leq 2 * S$, інакше стає неможливим згорання деталей пакету.

Обрахунок залежності швидкості руху каретки лінійного модуля від часу

Дано: точку $(x_0; y_0)$ на прямокутній декартовій системі координат та функцію траєкторії $y = f(x)$, неперервну на проміжку $(0, x_0)$ та функцію залежності переміщення матеріальної точки по осі Ox : $x = v_x \times t$.

Задача: знайти похідну функції залежності y від часу t (функцію швидкості v).

Алгоритм розрахунку швидкості лінійного приводу від часу наступний:

КРОК 1. Дано функцію $y = f(x)$, проте вона не обов'язково проходить точку $(x_0; y_0)$, тому потрібно домножити її на коефіцієнт k . При домноженні неперервної функції на число, функція залишається неперервною: $y = k \times f(x)$.

КРОК 2. Знайти коефіцієнт k для функції в точці $(x_0; y_0)$: $k = y_0 / f(x_0)$.

КРОК 3. Підставити $x(t)$ у функцію $y = f(x)$: $y = f(t) = f(x(t))$.

КРОК 4. Знайти похідну функції y : $y' = f'(x)$.

Даний алгоритм дозволяє отримати функцію швидкості каретки ЛП, в залежності від обраної функції траєкторії та швидкості руху конвеєра.

Приклад 1. Лінійний закон переміщення при куті $a = 45$ градусів.

Дано: лінійна функція $y = x$, точка $(200; 200)$, $x(t) = 300 \times t$.

Знайдемо k : $k = y_0 / f(x_0) = 200 / 200 = 1$. Що є очікуваним, оскільки функція проходить дану точку. Підставляємо функцію $y = f(t) = f(x(t)) = 1 \times 300 \times t = 300 \times t$. Шукаємо похідну $v = y'(t) = (300t)' = 300$ мм/с. Що є очевидно правильним рішенням.

Алгоритмічне забезпечення автоматизованої системи проклеювання деталей пакетів складається із наступних кроків:

ПОЧАТОК. Ініціалізація мережевого зв'язку ПЛК із ПВВ та ПВ.

КРОК 1. Переведення кареток ЛП1 та ЛП2 до фізичного крайнього лівого положення (позиція «0»).

КРОК 2. Виконати введення фізичних розмірів пакета через пульт оператора.

КРОК 3. Виконати початкове позиціонування кареток на певну відстань у мм. до робочих положень $Q1$ (для КП1) та $Q2$ (для КП2) в залежності від параметрів пакету: $Q1 = (J + S/2 + 20)$ мм, $Q2 = (J + S + S/2 + 20)$ мм.

КРОК 4. Запустити програму сканування наявності пакета на конвеєрі.

КРОК 5. За наявного пакету пересувати КП1 та КП2 згідно цільової траєкторії нанесення клею на деталі пакета.

КРОК 6. Зупинити роботу і перехід на **КІНЕЦЬ**, або повернення до КРОК5.

КІНЕЦЬ.

На першому кроці обидві каретки ЛП1 ЛП2 рухаються одночасно до спрацювання датчика крайнього положення. чи прийшли каретки у пункт призначення. Якщо обидві каретки досягли крайнього положення, тоді відносні позиції обох кареток приймаються за фізичний нуль. Якщо прийшла одна - чекаємо на другу. Якщо каретка не приходить до позиції «0» довше ніж за 10 секунд – формується повідомлення про несправність датчика або модуля лінійного переміщення.

На третьому кроці виконується початкове позиціонування кареток, але за умови попередньо вдалої спроби калібрування крайнього положення кареток. Якщо положення кареток не відкаліброване, тоді виконати крок 1 позиціонування, а вже потім повернутись до поточного кроку. Перед початком руху кареток до робочих положень відбувається підрахунок відстаней $Q1$ та $Q2$, необхідних для пересування кареток. Далі обраховані цільові переміщення переводяться у кількість імпульсів для передачі до СД1 та СД2. Вмикається програмно генератор імпульсів і дві каретки незалежно та практично паралельно рухаються до відповідних робочих позицій. Якщо, не досягнувши робочої позиції, спрацював ДКЛ1 (ДКЛ2), це свідчить про несправність датчика крайнього положення або користувачем внесені некоректні геометричні параметри пакета (які впливають на обрахунок робочих позицій).

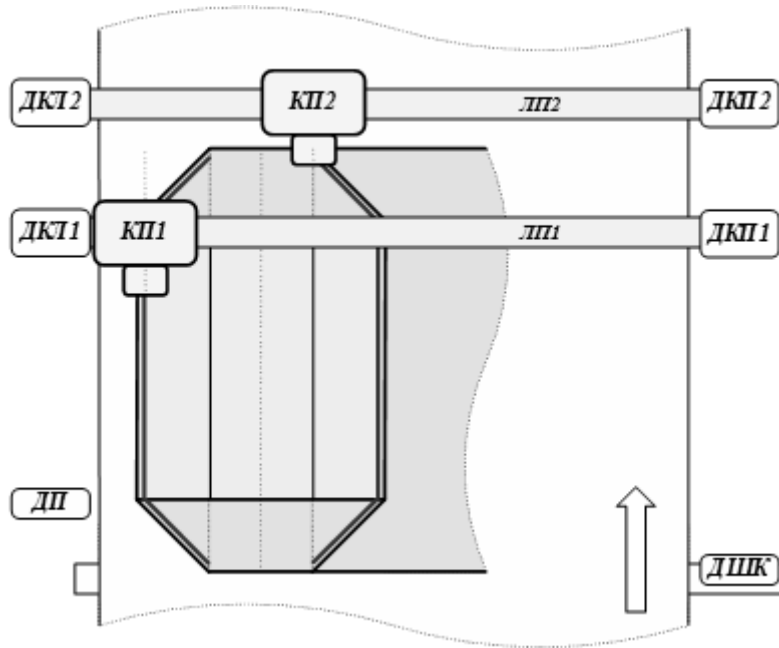


Рис. 4. Фізичне розташування периферійних пристроїв на конвеєрі

Після підготовчих етапів програма переводиться у режим сканування наявності пакетів на конвеєрі. Якщо пакет з'явився, тоді обчислюється його довжина (в залежності від швидкості конвеєру та часу виявлення пакета на конвеєрі) і КП1 та КП2 лінійно переміщуються згідно заданої траєкторії проклеювання. Після проклеювання деталей пакету КП1 та КП2 повертаються до робочих позицій і очікують на надходження чергового пакета. Виходячи із технічного завдання, де довжина пакету визначається на початку роботи системи, можемо зробити висновок, що перший пакет, в межах цієї серії проклеювання партії пакетів, буде відбракований.

Реалізація алгоритмічного забезпечення відбулася засобами інструментального програмного забезпечення для програмування промислових контролів CoDeSys V.3.5. Конвеєрний стіл, розробленої автоматизованої системи, зображений на рис. 5. На рисунку представлено фото, де зафіксовано момент проходження деталі пакета (папір білого кольору) під каретками модулів лінійного переміщення. На цьому фото змонтовані датчики, серводрайвери і лінійні модулі, але клейових пістолетів не змонтовано та відсутня клеєналивальна машина.



Рис. 5. Конвеєрний стіл автоматизованої системи в цеху підприємства

В середовищі CoDeSys було розроблено графічний інтерфейс лицьової панелі оператора та керуючу програму контролера СПК107. Засобами CoDeSys виконано попереднє імітаційне моделювання роботи автоматизованої системи. Були перевірені команди зчитування даних з датчиків, та алгоритми керування серводрайверами і клейовими пістолетами. Також у вигляді часових трендів отримано серію графіків симуляції роботи системи у псевдореальному часі. Крім того в межах цього дослідження вдалося виконати серію фізичних експериментів з нанесення графітовими олівцями (закріпленими на каретках) рисунків трапецеїдальної геометрії на тестову паперову деталь згідно заданих параметрів цільового пакету та швидкості конвеєру.

Висновки

У роботі було досліджено актуальне питання побудови автоматизованих машин та конвеєрних ліній для виконання технологічних операцій проклеювання паперових деталей для їх подальшого збирання в готовий пакет. Механізмів виготовлення деталей пакетів та складання деталей в пакет не досліджувалось. Проведено дослідження останніх наукових публікацій, що підтвердили актуальність проектування низькобюджетних автоматизованих систем для вітчизняних виробників паперових пакетів. Згідно опитувальника від замовника, наведено стислий опис об'єкту автоматизації, зокрема геометричні параметри пакету та траєкторії проклеювання. Розроблено структурну схему автоматизованої системи проклеювання деталей пакета та алгоритмічне забезпечення керуючої програми контролера. Наведений список основних програмно-технічних засобів системи. Визначені області допустимих значень параметрів пакета. Наведений обрахунок залежності швидкості руху каретки модуля лінійного переміщення від часу. Представлено результат фізичного впровадження системи в цеховому приміщенні підприємства-замовника. В результаті експериментальних досліджень та інженерних робіт вперше на вітчизняну підприємстві в м. Бориспіль розроблено конвеєрну лінію та впроваджено програму промислового контролера, що виконує задачу керування лінійним переміщенням клейових пістолетів на столі конвеєру перпендикулярно руху пакетів.

Протягом тривалої часу від концепції до фізичної реалізації проекту, як технічне завдання, так і автоматизована система набували незначних змін. Впровадження автоматизованої системи забезпечить зменшення активних працівників у виробничому цеху, які дотепер виконують ручні операції проклеювання. Але війна внесла свої корективи та не дозволила завершити повноцінне введення в штатну експлуатацію конвеєрної лінії на підприємстві. Тож науково-дослідні роботи станом на тепер заморожені з причини правового режиму воєнного стану в Україні та зухвалої імперської поведінки країни-агресора. Автор та інші учасники проекту не залишають надії повернутись до продовження досліджень та вдосконалення системи. Зокрема, в найближчих планах максимально швидко інтегрувати в автоматизовану систему машину, що буде подавати набір пакетів на конвеєрне полотно. Планується вдосконалення механічної конструкції над полотном конвеєра, а саме будуть додаватись механічні пристосування задля забезпечення процесу автоматичного відкривання та закривання пакету протягом його переміщення по конвеєру.

ЛІТЕРАТУРА

1. Тарасенко М. Особенности клеенаносящих устройств. *Журнал профессионалов ЛПК. ЛесПромИнформ*. 2018. №4 (134). URL: <https://lesprominform.ru/jarticles.html?id=5023>.
2. Shanghai Printyoung ZB 50B 2 HandBag Bottom Gluing Machine. URL: <https://www.youtube.com/watch?v=cB1KwCNxhBk>.
3. Pagano, S., Russo, R., Savino, S. A Smart Gluing Process by a Vision Guided Robotic System. In: Niola, V., Gasparetto, A. (eds) *Advances in Italian Mechanism Science. IFToMM ITALY 2020. Mechanisms and Machine Science*, 2021, vol 91. Springer, Cham. https://doi.org/10.1007/978-3-030-55807-9_47.
4. Kalai D. M. Low Cost Automated Paper Bag Making Machine. *International Journal for Research in Applied Science and Engineering Technology*. 2019, vol 7, Issue IV. <https://doi.org/10.22214/ijraset.2019.4596>.
5. Kumar A., Rawat J., Singh A., Singh H. A Research on Development & Fabrication of Automated Paper Bag Making Machine. *International Research Journal of Engineering and Technology*. 2019, vol 06, Issue 12. <https://www.irjet.net/archives/V6/i12/IRJET-V6I1295.pdf>.
6. Patil N. R., Lobo J., Madki A., Bhande R., Bodhankar S. Automated Paper Bag Making Machine. *International Journal of Innovative Research in Technology*. 2017, vol 03, Issue 12. URL: https://www.ijirt.org/master/publishedpaper/IJIRT144494_PAPER.pdf.
7. Gandhare S. N., Kaware S. M., Pandey R. S., Purse S. V., Choudhari S. P., Dable S. D., Borker S. D. Design and Fabrication of Paper Bag Making Machin. *International Journal of Advance Research and Innovative Ideas in Education*. 2019, vol 05, Issue 02. URL: http://ijariie.com/AdminUploadPdf/DESIGN_FABRICATION_OF_PAPER_BAG_MAKING_MACHINE_ijariie9646.pdf.

8. Paper packets. Format & size specification. Harrshan Products URL: <http://www.harshanproducts.com/format-size-specification.html>.

REFERENCES

1. Tarasenko M. Features of gluing devices. *Journal of timber industry professionals. LesPromInform*. 2018. №4 (134). URL: <https://lesprominform.ru/jarticles.html?id=5023>. [in Russian]
2. Shanghai Printyoung ZB 50B 2 HandBag Bottom Gluing Machine. URL: <https://www.youtube.com/watch?v=cB1KwCNxhBk>.
3. Pagano, S., Russo, R., Savino, S. A Smart Gluing Process by a Vision Guided Robotic System. In: Niola, V., Gasparetto, A. (eds) *Advances in Italian Mechanism Science. IFToMM ITALY 2020. Mechanisms and Machine Science*, 2021, vol 91. Springer, Cham. https://doi.org/10.1007/978-3-030-55807-9_47.
4. Kalai D. M. Low Cost Automated Paper Bag Making Machine. *International Journal for Research in Applied Science and Engineering Technology*. 2019, vol 7, Issue IV. <https://doi.org/10.22214/ijraset.2019.4596>.
5. Kumar A., Rawat J., Singh A., Singh H. A Research on Development & Fabrication of Automated Paper Bag Making Machine. *International Research Journal of Engineering and Technology*. 2019, vol 06, Issue 12. <https://www.irjet.net/archives/V6/i12/IRJET-V6I1295.pdf>.
6. Patil N. R., Lobo J., Madki A., Bhande R., Bodhankar S. Automated Paper Bag Making Machine. *International Journal of Innovative Research in Technology*. 2017, vol 03, Issue 12. URL: https://www.ijirt.org/master/publishedpaper/IJIRT144494_PAPER.pdf.
7. Gandhare S. N., Kaware S. M., Pandey R. S., Purse S. V., Choudhari S. P., Dable S. D., Borker S. D. Design and Fabrication of Paper Bag Making Machin. *International Journal of Advance Research and Innovative Ideas in Education*. 2019, vol 05, Issue 02. URL: http://ijariie.com/AdminUploadPdf/DESIGN_FABRICATION_OF_PAPER_BAG_MAKING_MACHINE_ijariie9646.pdf.
8. Paper packets. Format & size specification. Harrshan Products URL: <http://www.harshanproducts.com/format-size-specification.html>.

Valerii Yalanetskyi *Senior Lecturer*
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute",
37 Peremohy Ave., Kyiv-56, Ukraine, 03056

Automated system for gluing paper bag parts

Relevance. Gift and souvenir products are usually presented in paper bags. Research and development of practical solutions for automating the process of gluing package parts is an urgent scientific problem and engineering task. This article discusses the issue of the latest software and hardware solution for automating the application of glue to paper products of various sizes, mostly in small batches.

Goal. The purpose of this paper is to highlight engineering solutions that allow to develop an automated line for gluing paper bag parts with minimal costs for the company's budget.

Research methods. The object of automation is studied, in particular, the analysis of the range of sizes of paper products and the productivity of manufacturing paper bags in manual assembly shops is carried out. The issue of building automated conveyor lines for performing technological operations of gluing paper parts for their further assembly into a package is considered.

The results. A study of recent scientific publications was conducted, which confirmed the relevance of designing low-budget automated systems for domestic manufacturers of paper bags. The geometric parameters of the package and the gluing trajectory are described. A structural diagram of an automated system for gluing package parts and algorithmic support for the controller's control program have been developed. A list of the main software and hardware of the system is given. The areas of permissible values of the package parameters are determined. The calculation of the dependence of the carriage speed of the linear motion module on time is given. The result of physical implementation of the system in the workshop of the customer's enterprise is presented.

Conclusions. The introduction of the automated system will reduce the number of active employees in the production shop who still perform manual gluing operations. In the near future, we plan to integrate a machine that will feed a set of packages onto the conveyor belt into the automated system as quickly as possible. We plan to improve the mechanical structure above the conveyor belt, namely, to add mechanical devices to ensure the automatic opening and closing of the package as it moves along the conveyor.

Keywords: *automated system, automation, glue application; paper products, package.*