

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Черкаський національний університет
імені Богдана Хмельницького
MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
Bohdan Khmelnytsky
National University of Cherkasy

DOI: 10.31651/2076-5886-2020-1

ISSN 2076-5886 (Print)

ВІСНИК
ЧЕРКАСЬКОГО НАЦІОНАЛЬНОГО
УНІВЕРСИТЕТУ
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО

Серія
ПРИКЛАДНА МАТЕМАТИКА. ІНФОРМАТИКА

BULLETIN
OF THE CHERKASY
BOHDAN KHMELNYTSKY
NATIONAL UNIVERSITY

APPLIED MATHEMATICS. INFORMATICS

ВИПУСК 1
ISSUE 1

Черкаси, 2020
Cherkasy, 2020

**Засновник, редакція, видавець і виготовлювач –
Черкаський національний університет імені Богдана Хмельницького.
Свідоцтво про державну реєстрацію КВ № 16161-4633 ПР від 11.12.2009.
Свідоцтво про державну реєстрацію КВ № 23973-13813 Р від 21.05.2019.**

Журнал розрахований на математиків, спеціалістів у галузі ІТ, викладачів, науковців,
аспірантів, студентів.

Випуск № 1 наукового журналу «Вісник Черкаського національного університету імені Богдана Хмельницького. Серія «Прикладна математика. Інформатика» рекомендовано до друку та до поширення через мережу Інтернет Вченою радою Черкаського національного університету імені Богдана Хмельницького (протокол № 7 від 20.05.2020 року).

Журнал індексується Google Scholar.

Редакційна колегія серії:

Пасічний М.О., к.ф.-м.н., доц., ЧНУ ім. Б. Хмельницького (головний редактор);
Головня Б.П., д.т.н., доц., ЧНУ ім. Б. Хмельницького (заступник головного редактора);
Сердюк О.А., к.е.н., ЧНУ ім. Б. Хмельницького (відповідальний секретар);
Соловійов В.М., д.ф.-м.н., проф., КДПУ; Запорожець Т.В., д.ф.-м.н., проф., ЧНУ ім. Б. Хмельницького; Шквар Є.О., д.т.н., проф., Zhejiang Normal University (Zhejiang, China);
Ляшенко Ю.О., д.ф.-м.н., проф., ЧНУ ім. Б. Хмельницького; Дідковський Р.М., д.т.н., доц., провідний інженер, програміст (м. Черкаси, Україна); Гаєв Є.О., д.т.н., проф., НАУ;
Сторожук Н.В., к.ф.-м.н., ЧНУ ім. Б. Хмельницького; Ли́ла Д.М., к.ф.-м.н., ЧНУ ім. Б. Хмельницького; Бабенко С.В., к.ф.-м.н., ЧНУ ім. Б. Хмельницького.

За дотримання права інтелектуальної власності, достовірність матеріалів та обґрунтування висновків відповідають автори.

Адреса редакційної колегії:

18031, Черкаси, бул. Шевченка, 79
Черкаський національний університет імені Богдана Хмельницького, корпус № 3, к. 307
e-mail: ami.cdu@ukr.net

З електронною версією журналу можна ознайомитися за адресою: <http://ami-ejournal.cdu.edu.ua/>

**Founder, editorial, publisher and manufacturer –
Bohdan Khmelnytsky National University of Cherkasy.**
State registration certificate: KV No. 16161-4633 PR dated 11.12.2009.
State registration certificate: KV No. 23973-13813 R dated 21.05.2019.

This journal is meant for mathematicians, IT specialists, teachers, researchers, postgraduates and students.

Issue № 1 of the scientific journal «Bulletin of the Cherkasy Bohdan Khmelnytsky national university. Applied mathematics. Informatics» is recommended for publication and dissemination through the Internet by the Academic Council of Bohdan Khmelnytsky National University of Cherkasy (protocol number 7 dated 20.05.2020).

The journal is indexed in Google Scholar.

Editorial board of the series:

Pasichnyy M.O., Candidate of Physical and Mathematical Sciences, Associate Professor (Editor in Chief); Golovnya B.P., Doctor of Technical Sciences, Associate Professor (Deputy Chief Editor); Serdiuk O.A., Candidate of Economic Sciences (executive secretar); Soloviev V. M., Doctor of Physical and Mathematical Sciences, Professor; Zaporozhets T.V., Doctor of Physical and Mathematical Sciences, Professor; Shkvar Ye.O., Doctor of Technical Sciences, Professor; Lyashenko Y.O., Doctor of Physical and Mathematical Sciences, Professor; Didkovsky R.M., Doctor of Technical Sciences, Associate Professor; Gayev Ye.A., Doctor of Technical Sciences, Professor; Storozhuk N.V., Candidate of Physical and Mathematical Sciences; Lila D.M., Candidate of Physical and Mathematical Sciences; Babenko S.V., Candidate of Physical and Mathematical Sciences.

The authors are responsible for the observance of the intellectual property right, for the reliability of the materials and for the substantiation of the conclusions.

Editorial office address:
18031, Cherkasy, Shevchenko Blvd., 79
Bohdan Khmelnytsky National University of Cherkasy, building 3, ap. 307
e-mail: ami.cdu@ukr.net

All electronic versions of articles are available on the website edition <http://ami-ejournal.cdu.edu.ua/>

© Bohdan Khmelnytsky National University of Cherkasy, 2020
© Copyright by the contributors

СЕКЦІЯ «ПРИКЛАДНА МАТЕМАТИКА»

УДК 517.9:539.3

DOI 10.31651/2076-5886-2020-1-4-12

PACS 02.70.-c

СТЕБЛЯНКО Павло Олексійович,
доктор-фізико математичних наук,
професор, професор кафедри кібербезпеки
та інформаційних технологій, Університет
митної справи та фінансів
e-mail: caf-vmi@ukr.net
ORCID: 0000-0003-0789-4409

ДЬОМІЧЕВ Костянтин Едуардович,
кандидат технічних наук, доцент,
професор кафедри комп'ютерних наук,
Київський міжнародний університет
e-mail: demichevk@gmail.com
ORCID: 0000-0002-3428-4094

ПЕТРОВ Олександр Дмитрович,
доктор філософії, молодший науковий
співробітник, Дніпровський національний
університет ім. О.Гончара
e-mail: caf-vmi@ukr.net
ORCID: 0000-0001-8688-043X

**НЕЛІНІЙНА ФЕНОМЕНОЛОГІЧНА МОДЕЛЬ ПОВЕДІНКИ
ФУНКЦІОНАЛЬНО НЕОДНОРІДНИХ МАТЕРІАЛІВ**

У роботі сформована нелінійна феноменологічна модель, яка описує властивості сплавів з пам'яттю форми та термо-псевдо-пластичну поведінку (ТППМ) матеріалу саме в точці. Використано діаграму псевдо-пружного матеріалу, що складається з трьох криволінійних ділянок. Такий підхід призводить до нестійкої діаграми напруження-деформація, і для опису термомеханічної поведінки зразків різної форми необхідно мати рішення граничної задачі з урахуванням розвитку фронту деформації фазового перетворення. При цьому враховано не тільки температуру навколишнього середовища, але і тепло, що вивільняється в точці при фазовому переході. Розроблено числову процедуру розрахунку діаграми матеріалу, яка представляє собою криву, що огинає сімейство діаграм матеріалу, побудованих для певних законів зміни швидкості фронту розриву деформації. Побудовано інтегральну діаграму матеріалу, який знаходиться під впливом складного навантаження.

Ключові слова: феноменологічна модель, нелінійна модель матеріалу, матеріали з пам'яттю форм, термо-псевдо-пластичність, числова процедура розрахунку діаграми.

Вступ

На даний час відомий цілий ряд моделей для опису термомеханічної поведінки функціонально неоднорідних матеріалів, зокрема сплавів з пам'яттю форми (СПФ) [3; 4]. Більшість з них будуються на підставі класичних уявлень, тобто ставлять собі за мету безпосередньо описати експериментальні дані, отримані на різних макрозразках при простому і складному навантаженні. Однак, як встановлено в експериментальних дослідженнях, поведінка матеріалу в точці тіла в загальному випадку може бути

відмінною від поведінки зразка в цілому [5].

Мета роботи сформулювати **нелінійну** феноменологічну модель, яка описує властивості сплавів з пам'яттю форми та термо-псевдо-пластичну поведінку (ТППМ) матеріалу саме в точці.

Виклад основного матеріалу

Для опису пружної деформації і деформації фазового перетворення скористаємося діаграмою псевдо-пружного матеріалу, що складається з трьох **криволінійних** ділянок. Таке трактування призводить до нестійкої діаграми напруження-деформація і для описання термомеханічної поведінки зразків різної форми необхідно мати рішення граничної задачі з урахуванням розвитку фронту деформації фазового перетворення.

При цьому будемо враховувати не тільки температуру навколишнього середовища, але і тепло, що вивільняється в точці при фазовому переході. Таке трактування дозволило з єдиної позиції сформулювати нелінійну феноменологічну модель і описати цілий ряд експериментальних даних на різних зразках при різних умовах навантаження, включаючи циклічне температурне і силовий вплив. Встановлено конкретні залежності для механічних параметрів. Показано, що межа розділу фаз рухається з постійною для обраної температури швидкістю. Підтверджено, що класичні діаграми матеріалів представляють собою криву, що огинає сімейство діаграм матеріалу в точці, яке побудовано для певних законів зміни швидкості фронту розриву деформацій.

Узагальнена залежність між напруженням σ і деформацією ε , в матеріальній точці моделюється кривою, що зображена на рисунку 1. При побудові такої діаграми задаються точки $(\sigma_M, \varepsilon_M)$, $(\sigma_m, \varepsilon_m)$ і функції $\varepsilon = \psi_i(\sigma)$, $i=1, 2, 3$.

Змінна деформація фазового перетворення визначається, як різниця

$$\varepsilon_T = \psi_3(\sigma) - \psi_1(\sigma). \quad (1)$$

Функції $\psi_i(\sigma)$ діаграми мають задовольняти наступним вимогам:

$$\begin{aligned} \psi_1(0) &= 0, \psi_1(\sigma_M) = \varepsilon_M, \psi_1'(\sigma_M) > 0, \\ \psi_2(\sigma_M) &= \varepsilon_M, \psi_2(\sigma_m) = \varepsilon_m, \psi_2'(\sigma) < 0, \\ \psi_3(\sigma_m) &= \varepsilon_m, \psi_3'(\sigma) > 0. \end{aligned} \quad (2)$$

Поле переміщень в зразку повинно бути неперервним

$$u(x) = \begin{cases} \psi_3(\sigma)x, & 0 \leq x < s, \\ \psi_1(\sigma)x + \varepsilon_T, & s \leq x < L. \end{cases}$$

Відповідний кусково-однорідний розподіл деформації в стрижні дається формулою

$$\varepsilon(x) = \begin{cases} \psi_3(\sigma), & 0 \leq x < s, \\ \psi_1(\sigma), & s \leq x < L. \end{cases}$$

Залежність між збільшенням довжини $u(L) = \delta$ стрижня і напруженням визначається виразом

$$\delta = \psi_3(\sigma)s + \psi_1(\sigma)(L-s). \quad (3)$$

Тут через s позначено границю розділу фаз.

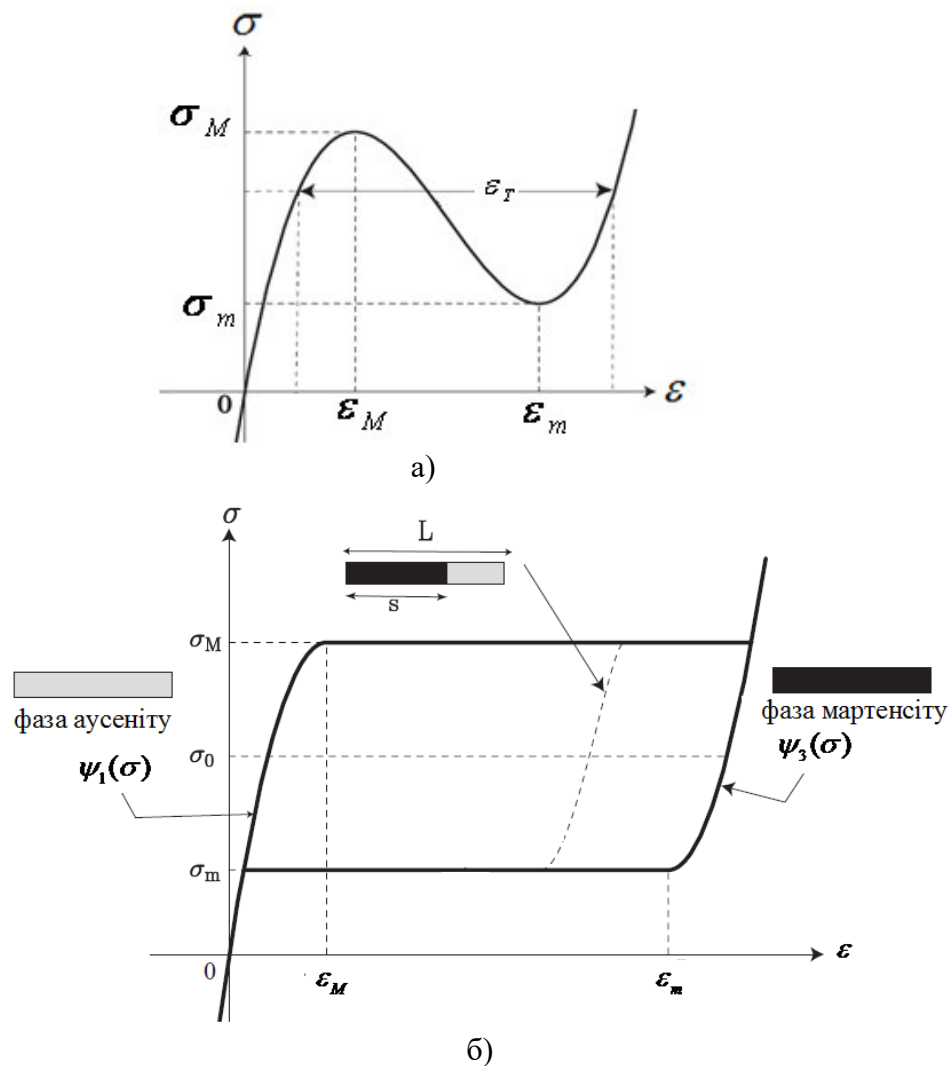


Рисунок 1. Крива напруження-деформація для двофазного матеріалу

Взагалі рівняння (3) дає змогу визначити сімейство $\delta(\sigma, s)$. Якщо δ задано, то можна знайти напруження σ , розв'язуючи відповідно до умов (2) крайову задачу нелінійної пружності. При цьому з виразу (3) не можна одразу визначити σ , оскільки невідомо положення границі розділу фаз. Для однозначного розв'язання цієї задачі необхідно мати допоміжну інформацію окрім рівняння (2.3), фізичних співвідношень $\epsilon = \psi_i(\sigma)$, $i=1, 2, 3$, граничних умов і вимог до гладкості. Перераховані рівняння достатні при відсутності нестійких ділянок діаграми, для звичайних еластичних матеріалів, для яких напруження монотонно зростає зі збільшенням деформації.

Порушення однозначності виникає в (3), якщо задано σ , а не δ . В цьому випадку проблема вирішується шляхом введення поняття рушійної сили f . Вона визначається так

$$f(\sigma) = \int_{\sigma_0}^{\sigma} \epsilon_T(\sigma) d\sigma. \quad (4)$$

Тут $\sigma_0 = (\sigma_M + \sigma_m)/2$ – напруження Максвелла. Оскільки напруження у стані рівноваги в змішаній фазі повинно знаходитися між σ_m і σ_M , то діапазон можливих значень рушійної сили буде таким $f \in [f_m, f_M]$.

Розглянемо допоміжну задачу, в якій треба визначити функції часу $\sigma = \sigma(t)$, $s = s(t)$, $\delta = \delta(t)$.

Похідна за часом від залежності (3) дає наступне нелінійне рівняння

$$\dot{\delta} = \varepsilon_T \dot{s} + (\varepsilon_T'(\sigma)s + \psi_1'(\sigma)L)\dot{\sigma}. \quad (5)$$

Тут точкою позначено похідні за часом, а штрихом похідні по напруженню σ .

В результаті з (5) в загальному випадку дотичний модуль діаграми зразка можна визначити так

$$\frac{d\sigma}{d\delta} = \frac{1 - \varepsilon_T(\sigma)\dot{s}/\dot{\delta}}{\psi_1'(\sigma)L + \varepsilon_T'(\sigma)s}.$$

При його обчисленні треба попередньо визначити функції $\sigma = \sigma(t)$, $s = s(t)$. Для розв'язання цієї задачі до рівняння (5) слід додати функцію кінетичного відгуку $\dot{s} = \Phi(f)$ (рисунок 2) і вираз для рушійної сили (4), який можна представити так

$$f(\sigma) = \int_{\sigma_0}^{\sigma} [\psi_3(\sigma) - \psi_1(\sigma)] d\sigma. \quad (6)$$

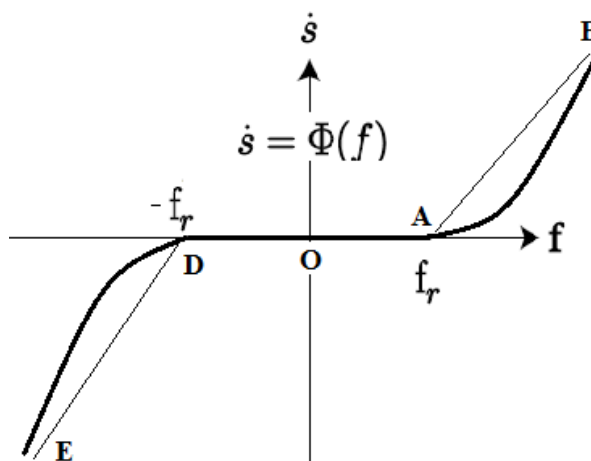


Рисунок 2. Схематичний графік нелінійної функції кінетичного відгуку.

Для триланкового двофазного матеріалу зі змінними модулями пружності і відповідно змінною трансформаційною деформацією ε_T на основі наведених вище формул (5) і (6) можна записати залежності між напруженням і збільшенням довжини пружного зразка, а також для рушійної сили.

Рівняння першої ділянки діаграми для $\varepsilon \in [0; \varepsilon_M]$, $\sigma \in [0; \sigma_M]$, має вигляд

$$\varepsilon(\sigma) = \psi_1(\sigma). \quad (7)$$

Рівняння діаграми на ділянці $\varepsilon \in [\varepsilon_m; \infty)$, $\sigma \in [\sigma_m; \infty)$, має вигляд

$$\varepsilon(\sigma) = \psi_3(\sigma).$$

Зазначимо, що час протікання процесу на другій ділянці $\varepsilon(\sigma) = \psi_2(\sigma)$ значно менший у порівнянні з перебігом деформування на першій та третій ділянках. Тому будемо вважати, що фазове перетворення протікає миттєво.

В задачах механіки, де вивчається поведінка матеріалів, в яких мають місце фазові переходи, для побудови фізичних співвідношень між напруженням та

деформацією необхідно додатково знати положення в зразку фронту фазового переходу $s = s(t)$ (рисунок 2) і функцію кінетичного відгуку $\dot{s} = \Phi(f)$.

Розглянемо випадок, коли функція кінетичного відгуку $\dot{s} = \Phi(f)$ буде мати вигляд, представлений на рисунку 2. Ця функція може бути задана аналітично.

$$\dot{s} = \begin{cases} 0; f \in [0; f_r]; \\ \dot{s}_0 + \omega(f - f_r)^n. \end{cases} \quad (8)$$

На першій ділянці (до фазового переходу) $\dot{s} \equiv 0; s \equiv 0$. В силу цього рівняння (5) набуде вигляд

$$\dot{\delta} = \psi_1'(\sigma) L \dot{\sigma}.$$

Звідси отримаємо формулу для обчислення дотичного модуля на першій ділянці діаграми

$$\frac{\partial \sigma}{\partial \delta} = \frac{1}{\psi_1'(\sigma) L}.$$

На третій ділянці (після фазового переходу) рівняння (5) з урахуванням (8) прийме вигляд

$$\dot{\delta} = \varepsilon_T(\sigma) \left[\dot{s}_0 + \omega(f - f_r)^n \right] + \frac{\varepsilon_T'(\sigma) s + \psi_1'(\sigma) L \dot{f}}{\varepsilon_T} f. \quad (9)$$

$$s = \dot{s}_0(t - t_r) + \omega \int_{t_r}^t (f - f_r)^n d\tau.$$

В результаті з (9) отримаємо наступну інтегро-диференціальну систему рівнянь відносно рушійної сили

$$s = \dot{s}_0(t - t_r) + \omega \int_{t_r}^t (f - f_r)^n d\tau, \quad (10)$$

$$\dot{f} = \frac{\varepsilon_T(\sigma)}{\varepsilon_T'(\sigma) s + \psi_1'(\sigma) L} \left[\dot{\delta} - \varepsilon_T(\sigma) (\dot{s}_0 + \omega(f - f_r)^n) \right].$$

Для випадку трилінійного двофазного матеріалу з однаковими модулями пружності результати отримані в роботах [6-8].

З метою спрощення подальшого запису введемо позначення

$$F = f - f_r$$

В результаті формули системи (10) перепишемо так

$$\dot{F} = \Phi(F; s), s = \dot{s}_0(t - t_r) + \omega \int_{t_r}^t F(\tau)^n d\tau, \quad (11)$$

$$\Phi(F; s) = \frac{\varepsilon_T(\sigma)}{\varepsilon_T'(\sigma) s + \psi_1'(\sigma) L} \left[\dot{\delta} - \varepsilon_T(\sigma) (\dot{s}_0 + \omega F^n) \right].$$

На першій ділянці (до фазового переходу) $\dot{s} \equiv 0; s \equiv 0; t \in [0; t_r]$. З формул (9) і (10) отримаєм

$$\dot{\sigma} = \frac{\mu_1}{L} \dot{\delta} \Rightarrow \sigma = \frac{\mu_1}{L} \dot{\delta} \cdot t,$$

$$\dot{f}(\sigma) = (k\sigma + b)\dot{\sigma} = \left(\frac{k\mu_1\dot{\delta}}{L}t + b \right) \frac{\mu_1}{L} \dot{\delta}.$$

Інтегруючи останній вираз, отримаємо

$$f(t) = k \left(\frac{\mu_1 \dot{\delta}}{L} \right)^2 \cdot \frac{t^2}{2} + \frac{b\mu_1 \dot{\delta}}{L} \cdot t,$$

$$t \in [0; t_r], f(0) = 0, f(t_r) = f_r < f_M.$$

На другій ділянці систему рівнянь (11) будемо розв'язувати чисельно при наступних умовах

$$s(t_r) = 0, F(t_r) = 0.$$

Для цього введемо сітку за часом і зробимо позначення

$$t_p = t_{p-1} + \tau, p = 1; 2; 3; \dots; t_0 = t_r,$$

$$s^p = s(t_p), F^p = F(t_p),$$

$$\Phi^p = \Phi(F^p, s^p), \varepsilon_T^p = \varepsilon_T(F^p).$$

В результаті диференціальне рівняння та інтеграл (11) можна замінити еквівалентними різницеvim рівняннями і сумою. Звідси отримаємо розрахункові формули, для рушійної сили F

$$F^p = F^{p-1} + \tau \cdot \Phi(F^p, s^p), s^p = p \cdot s_0 \cdot \tau + \omega \cdot J^p,$$

$$J^p = J^{p-1} + \frac{\tau}{2} (F^{p-1} + F^p), J^0 = 0, F^0 = 0. \tag{12}$$

Результати розрахунків по обчисленню дотичного модуля на кожному кроці інтегрування за часом для інтегральної діаграми матеріалу наведено нижче на рисунку 3.

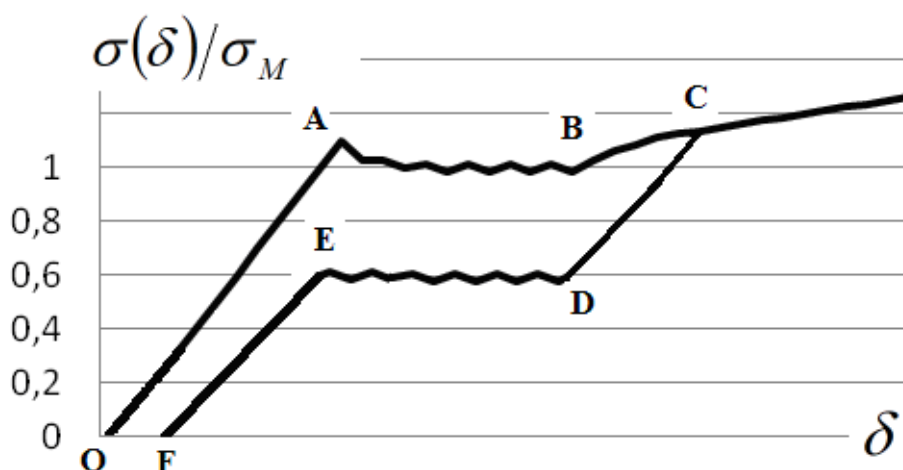


Рисунок 3. Інтегральна діаграма матеріалу

Перша ділянка інтегральної діаграми матеріалу, OA, відповідає пружній поведінці. Третя, BC, характеризує зміцнення матеріалу. Між ними міститься ділянка,

яка нагадує поведінку ідеально пластичного матеріалу. Аналогічні ділянки мають місце і при розвантаженні CDEF, але при визначених температурах.

Деформація в матеріальній точці представляється у вигляді суми пружною складовою; стрибка деформації при фазовому переході; пластичної деформації, яка підпорядковується теорії течії з кінематичним і трансляційним зміцненням; деформації, викликані температурними змінами. При цьому передбачається, що властивості матеріалу залежать від температури.

Висновки

Експериментально встановлено, що поведінка матеріалу в точці тіла в загальному випадку відрізняється від поведінки зразка в цілому. У роботі формулюється нелінійна феноменологічна модель для опису властивостей матеріалу саме в точці. Для опису пружної деформації і деформації фазового перетворення використовувалася діаграма псевдопружного матеріалу, що складається з трьох нелінійних ділянок. Таке трактування теорії призводить до нестійкої діаграми напруження-деформація і вимагає рішення граничної задачі з урахуванням розвитку фронту деформації перетворення. Це дозволило з позиції запропонованої нелінійної моделі матеріалу описати ряд експериментальних даних на різних зразках при різних умовах навантаження. Розроблено числову процедуру розрахунку діаграми матеріалу, яка представляє собою криву, що огинає сімейство діаграм матеріалу, побудованих для певних законів зміни швидкості фронту розриву деформацій. Це дозволило записати визначальні співвідношення між швидкостями напружень та деформацій в універсальній формі та застосувати до розв'язання конкретних нелінійних задач термо-псевдо-пружно-пластичності відомі методи.

Список використаної літератури:

1. Ивлев, Д. Д. Теория упрочняющегося пластического тела / Д. Д. Ивлев, Г. И. Быковцев. – Москва: Наука, 1971. – 232 с.
2. Стебляноко, П. А. Методы расщепления в пространственных задачах теории пластичности / П. А. Стебляноко – Киев: Наукова думка, 1998. – 304 с.
3. Abeyaratne, R. Evolution of phase transitions / R. Abeyaratne, J. K. Knowles – Cambridge University Press, 2006. – 258 p.
4. Shaw, J. A. Thermomechanical aspects of NiTi. / J. A Shaw, S. Kyriakides – Mechanics and Physics of Solids, 1995. – No 43, p.1243-1281.
5. Shaw, J. A. On the nucleation and propagation of phase transformation fronts in a NiTi alloy / J. A Shaw, S. Kyriakides – Acta Materialia, 1997. – No 45, p. 683-700.
6. Petrov, A. Development of the method with enhanced accuracy for solving problems from the theory of thermo-pseudoelastic-plasticity / A. Petrov, Yu. Chernyakov, P. Steblyanko, K. Demichev, V. Haydurov – Eastern-European Journal of Enterprise Technologies. 2018. Vol. 4/7 (94). P. 25-33.
7. Steblyanko, P. Phenomenological Model of Pseudo-Elastic-Plastic Material Under Nonstationary Combining Loading / P. Steblyanko, Y. Chernyakov, A. Petrov, V. Loboda – Structural Integrity, Volume 8, Theoretical, Applied and Experimental Mechanics, Springer Verlag, 2019. – P. 205-208.
8. Петров, О. Д. Комп'ютерне моделювання поведінки стрижня з трилінійного двофазного матеріалу при розтягуванні / О.Д. Петров – Інформаційні технології та комп'ютерне моделювання; матеріали статей МНПК (ISBN 978-617-7468-26-3) – 2018. – Івано-Франківськ. – 2018. – С. 234–237.

References:

1. Ivlev, D. D., Bikovtsev, G. I. (1971). The theory of a strengthened plastic body. – Moskva: Nauka. [in Russian]
2. Steblyanko, P. A. (1998). Methods of decomposition in space problems of the theory of plasticity. – Kyiv: Naukova dumka. [in Russian]
3. Abeyaratne, R., Knowles, J. K. (2006). Evolution of phase transitions. – Cambridge University Press.
4. Shaw, J. A., Kyriakides, S. (1995). Thermomechanical aspects of NiTi. J. Mechanics and Physics of Solids 43, 1243-1281.

5. Shaw, J. A., Kyriakides, S. (1997). On the nucleation and propagation of phase transformation fronts in a NiTi alloy. *Acta Materialia*.
6. Petrov, A., Chernyakov, Yu., Steblyanko, P., Demichev, K., Haydurov, V. (2018). Development of the method with enhanced accuracy for solving problems from the theory of thermo-pseudoelastic-plasticity. *Eastern-European Journal of Enterprise Technologies*.
7. Steblyanko, P., Chernyakov, Y., Petrov, A., Loboda, V. (2019). Phenomenological Model of Pseudo-Elastic-Plastic Material Under Nonstationary Combining Loading Theoretical, Applied and Experimental Mechanics, Springer Verlag.
8. Petrov, A. D. (2018). Computer modeling of the shearing behavior of trilinear two-phase material during stretching. *Information Technologies and Computer Modeling; materials of articles Annual Intern. Scientific.* [in Ukrainian]

STEBLYANKO Pavel,

Doctor of Physics and Mathematics, Professor, Professor of Department of Cybersecurity and Information Technology, University of Customs and Finance, Ukraine

DOMICHEV Konstantin,

Candidate of Technical Sciences, Associate Professor, Professor of Department of Computer Science, Kyiv International University, Ukraine

PETROV Alexander,

Doctor of Philosophy (PhD), Junior Researcher, Department of Theoretical and Computer Mechanics Dnieper National University Named After Oles Honchar, Ukraine

NONLINEAR PHENOMENOLOGICAL MODEL OF BEHAVIOR OF FUNCTIONALLY HETEROGENEOUS MATERIALS

Summary. Introduction. *In this work formed a nonlinear phenomenological model of describing the properties of alloys with shape memory and thermopseudo-plastic behavior of the material exactly at the point. A diagram of pseudo-elastic material consisting of three curvilinear sections is used. Such approach leads to unstable stress-strain diagram and in order to describe thermo-technical behavior of samples of different shapes it is necessary to have a solution to the boundary problem taking into account the development of phase transformation deformation front. In this case, not only the ambient temperature but also heat is released at the point of phase transition.*

In mechanics problems, where the behavior of materials in which phase transitions take place, to build physical relations between stress and strain, it is necessary to additionally know the position in the body of the front of the phase transition and the function of kinetic response. We consider the case when this function will have the form of a polyline consisting of two linear segments and a more general case with a nonlinear second segment, for which the second linear segment of the previous model will be an asymptote. This combined line can be set analytically, which allowed to build a more precise algorithm of problem solving.

A numerical procedure has been developed for calculating a material diagram, which is a curve that bends around a family of material diagrams constructed for certain laws of change in the velocity of the deformation rupture front. The integral diagram of the material is constructed, which is under the influence of a complex load.

For a two-phase material consisting of three curvilinear sections with different modulus of elasticity and, correspondingly, variable transformational strain on the basis of the proposed model, the dependence between the stress, length increase of the sample and the driving force is recorded. At reception of calculated formulas of the refined model the auxiliary problem connected with development of an instant thermomechanical surface on the given coordinates of four points has been considered. At active loading the total thermomechanical surface of pseudo-elastic material consists of three separate surfaces. This is the surface of the elastic part, the surface where deformations caused by phase transformation are jumping and the part of the surface for which the material continues to behave elastically (after phase transition) and then passes into the plastic state. Similarly, the thermomechanical surface is defined during unloading.

Purpose. *The aim of this work is to form a nonlinear phenomenological model that describes the properties of alloys with shape memory and thermopseudoplastic behavior (TPPM) of a point material.*

Results. *A nonlinear phenomenological model is formed in the paper to describe the properties of the material at the point. A diagram of a pseudoelastic material consisting of three nonlinear*

sections was used to describe the elastic deformation and the deformation of the phase transformation. A numerical procedure for calculating a material diagram has been developed, which is a curve enveloping a family of material diagrams constructed for certain laws of change in the velocity of the deformation rupture front.

Conclusion. It is experimentally established that the behavior of the material at a point in the body in the General case differs from the behavior of the sample as a whole. The paper formulates a nonlinear phenomenological model to describe the properties of the material at the point. A diagram of a pseudoelastic material consisting of three nonlinear sections was used to describe the elastic deformation and the deformation of the phase transformation. This interpretation of the theory leads to an unstable stress-strain diagram and requires a solution of the boundary value problem taking into account the development of the transformation deformation front. This allowed from the standpoint of the proposed nonlinear model of the material to describe a number of experimental data on different samples under different loading conditions. A numerical procedure for calculating a material diagram has been developed, which is a curve enveloping a family of material diagrams constructed for certain laws of change in the velocity of the deformation rupture front. This allowed us to record the defining relations between the rates of stresses and strains in a universal form and to apply the known methods to solve specific nonlinear problems of thermo-pseudo-elastic-plasticity [1; 2].

Keywords: phenomenological model, nonlinear model of material, materials with shape memory, thermo-pseudo-plasticity, numerical procedure of diagram calculation.

Одержано редакцією 04.03.2020 р.
Прийнято до публікації 27.04.2020 р.

УДК 519.854

DOI 10.31651/2076-5886-2020-1-12-22

PACS 02.70.Wz

LUKIANOV Petro,
PhD in Physics and Mathematics, Senior
Researcher, Associate Professor, Department
ASOIU, Faculty of Information and
Computer Techniks, NTU “KPI”
e-mail: luk_ptr@yahoo.com
ORCID: 0000-0002-7584-1491

MUKHA Iryna,
PhD in Technical Sciences, Associate
Professor, Deputy Head of the Department
ASOIU, Faculty of Information and
Computer Techniks, NTU “KPI”
e-mail: mip_kpi@gmail.com
ORCID: 0000-0002-4423-5106

BOOLEAN FUNCTION “EXCLUSIVE OR” FOR THE N VARIABLES

This paper proposes a new Boolean function – “exclusive OR” for case of n Boolean variables. The definition of this function is given; its basic properties are studied. In particular, it is established (theorem) that the “exclusive OR” of n variable is a symmetric function with respect to the permutation of arguments, and in the case $n = 2$ coincides with the “exclusive OR”. For the cases $n = 3; 4$, a disjunctive normal form (DNF), perfect disjunctive normal form (PDNF), minimal disjunctive normal form are constructed. In particular, in the case of $n = 3$, three equal minimal disjunctive normal forms were found using the Quine method. It is established that the conjunctive normal form, the perfect conjunctive normal form and the minimized conjunctive normal form coincide with the definition of the “exclusive OR” of n variable. For the cases $n = 3; 4$, Zhegalkin polynomials are constructed, which are of the order of one lower than PDNF. Post classification performed. It is established that the

"exclusive OR" for the cases of $n = 3, 4$ belongs to the class T_0 , does not belong to the classes T_1 and L , is not a monotonic and self- double function.

Keywords: Boolean function, "exclusive OR" of n variables.

Introduction

Today in the theory of Boolean functions [1], [2] there is a known function, "exclusive OR". It looks like this:

$$f(x_1, x_2) = x_1 \oplus x_2 = \bar{x}_1 x_2 + x_1 \bar{x}_2 = (\bar{x}_1 \wedge x_2) \vee (x_1 \wedge \bar{x}_2), \quad (1)$$

in which the dash above means a logical "no", ie " \neg ". In order to simplify the presentation further in this paper, the sign of Boolean multiplication and addition will be replaced by the usual multiplication and addition.

In the Venn's diagram [3] (Fig. 1), the function is denoted as two intersecting circles. The common part of these circles does not belong to the scope of this function. The remaining set is the domain of this function: left elements \bar{x}_2 , but x_1 , and right \bar{x}_1 , but x_2 . If we consider "exclusive OR" of three variables x_1, x_2, x_3 in one of the options:

$$x_1 \oplus (x_2 + x_3), \text{ or } x_2 \oplus (x_1 + x_3), \text{ or } x_3 \oplus (x_1 + x_2),$$

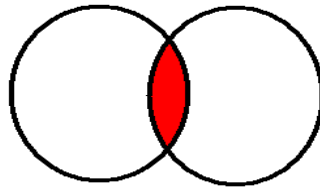


Fig. 1. Venn's diagram for the "exclusive or" of 2 variables

This type of function leads to the following expanded expressions:

$$\begin{aligned} x_1 \oplus (x_2 + x_3) &= (x_1 + x_2 + x_3)(\bar{x}_1 + \bar{x}_2 \bar{x}_3), \\ x_2 \oplus (x_1 + x_3) &= (x_2 + x_1 + x_3)(\bar{x}_2 + \bar{x}_1 \bar{x}_3), \\ x_3 \oplus (x_1 + x_2) &= (x_3 + x_1 + x_2)(\bar{x}_3 + \bar{x}_1 \bar{x}_2). \end{aligned} \quad (2)$$

As can be seen from expressions (2), for a logical function of "exclusive OR" in the case of three logical variables such a symmetric expression as for the case of two variables (1) no longer exists. And we would very much like to. To solve this problem, we introduce a new Boolean function "exclusive OR" of n variables.

The aim of the study of the paper is Boolean function "exclusive OR" of n variables and its properties.

"Exclusive OR" for n variables

Let X be a Boolean space, and let $x_1, x_2, \dots, x_n \in X$ there be a set of Boolean variables.

Definition. "Exclusive OR" of n variables is the following Boolean function:

$$f(x_1, x_2, \dots, x_n) = \bigoplus_1^n x_i = (x_1 + x_2 + \dots + x_n)(\bar{x}_1 + \bar{x}_2 + \dots + \bar{x}_n). \quad (3)$$

The symbol $\oplus_1^n x_i$ hereafter will denote the Boolean "Exclusive OR" of n variables.

Theorem 1. In the case of $n = 2$, the "Exclusive OR" of n variables is usual "Exclusive OR".

Proof. If we consider the case of a function of two variables $f(x_1, x_2)$, then expression (3) becomes:

$$f(x_1, x_2) = \oplus_1^2 x_i = (x_1 + x_2)(\bar{x}_1 + \bar{x}_2),$$

that is, usual "Exclusive OR". The theorem is proved.

Theorem 2. "Exclusive OR" of n variables is a symmetric function with respect to permutation of arguments $x_i \rightarrow x_j$.

Proof. Indeed, if in formula (3) perform a permutation of any two arguments $x_i \rightarrow x_j$, the value of the function does not change. The theorem is proved.

Corollary of Theorem 2. Boolean "Exclusive OR" of n variables is a symmetric function for any finite number of permutations of k -arguments ($k < n$).

For a more detailed study of "Exclusive OR" of n variables, consider two cases of the function $n = 3; 4$. We construct a truth table for each of these cases, perfect disjunctive and conjunctive normal forms, find minimal forms and construct Zhegalkin polynomials, perform Post classification.

1. Case $n = 3$

In this case, function (1) takes the expression:

$$f(x_1, x_2, x_3) = \oplus_1^3 x_i = (x_1 + x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3)$$

Table of truth in the case of $n = 3$ has the following form:

x_1	x_2	x_3	$x_1 \vee x_2$	$x_1 \vee x_2 \vee x_3$	\bar{x}_1	\bar{x}_2	$\bar{x}_1 \vee \bar{x}_2$	\bar{x}_3	$\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$	$(x_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$
0	0	0	0	0	1	1	1	1	1	0
0	0	1	0	1	1	1	1	0	1	1
0	1	0	1	1	1	0	1	1	1	1
0	1	1	1	1	1	0	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1
1	0	1	1	1	0	1	1	0	1	1
1	1	0	1	1	0	0	0	1	1	1
1	1	1	1	1	0	0	0	0	0	0

Table 1. Table of truth in the case of $n = 3$

If we perform the appropriate logical operations, we will have the following:

$$(x_1 + x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3) = x_1 \bar{x}_1 + x_1 \bar{x}_2 + x_1 \bar{x}_3 + x_2 \bar{x}_1 + x_2 \bar{x}_2 + x_2 \bar{x}_3 + x_3 \bar{x}_1 + x_3 \bar{x}_2 + x_3 \bar{x}_3 \quad (4)$$

In the right-hand side of equation (4), expressions $x_i \bar{x}_i = 0, i \leq 3$ are absorbed as mutually contradictory. In the end, only the following expression will remain:

$$f(x_1, x_2, x_3) = \oplus_1^3 x_i = x_1(\bar{x}_2 + \bar{x}_3) + x_2(\bar{x}_1 + \bar{x}_3) + x_3(\bar{x}_1 + \bar{x}_2), \quad (5)$$

or

$$f(x_1, x_2, x_3) = \oplus_1^3 x_i = x_1\bar{x}_2 + x_1\bar{x}_3 + x_2\bar{x}_1 + x_2\bar{x}_3 + x_3\bar{x}_1 + x_3\bar{x}_2. \quad (6)$$

Expression (6) is a *disjunctive normal form* that looks like known "Exclusive OR" $n=2$. Thus, the introduced function "Exclusive OR" of $n=3$ variables is in fact a generalization of the logical function "Exclusive OR" for the case of n variables (in the case $n=3$). The Venn's diagram, based on (5) for this function, has the form of Fig.2.

1.1. A perfect disjunctive normal form is formed using the law of addition: $x_i + \bar{x}_i = 1$ [4], [5]. So we have:

$$\oplus_1^3 x_i = x_1\bar{x}_2(x_3 + \bar{x}_3) + x_1\bar{x}_3(x_2 + \bar{x}_2) + x_2\bar{x}_1(x_3 + \bar{x}_3) + x_2\bar{x}_3(x_1 + \bar{x}_1) + x_3\bar{x}_1(x_2 + \bar{x}_2) + x_3\bar{x}_2(x_1 + \bar{x}_1)$$

Perform logical multiplication operations:

$$\begin{aligned} \oplus_1^3 x_i = & x_1\bar{x}_2x_3 + x_1\bar{x}_2\bar{x}_3 + x_1\bar{x}_3x_2 + x_1\bar{x}_3\bar{x}_2 + x_2\bar{x}_1x_3 + x_2\bar{x}_1\bar{x}_3 + x_2\bar{x}_3x_1 + x_2\bar{x}_3\bar{x}_1 + x_3\bar{x}_1x_2 + \\ & + x_3\bar{x}_1\bar{x}_2 + x_3\bar{x}_2x_1 + x_3\bar{x}_2\bar{x}_1 \end{aligned} \quad (7)$$

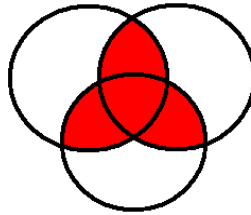


Fig. 2. Venn's diagram for the "Exclusive OR" of $n=3$ variables

In expression (7) we have twice the number of identical terms. According to the law of logical addition $x + x = x$, this expression is halved:

$$\oplus_1^3 x_i = x_1\bar{x}_2x_3 + x_1\bar{x}_2\bar{x}_3 + x_1\bar{x}_3x_2 + x_2\bar{x}_1x_3 + x_2\bar{x}_1\bar{x}_3 + x_3\bar{x}_1\bar{x}_2. \quad (8)$$

Expression (8) is a *perfect disjunctive normal form* (PDNF) of "Exclusive OR" of $n=3$.

1.2. Minimum DNF

We find the minimum disjunctive normal form using the Quine's method. Construct a corresponding table with minterms of the third rank, included in the right part (8) of PDNF (table 2).

We have 6 implicants: $x_3\bar{x}_2, \bar{x}_1x_3, x_1\bar{x}_2, x_1\bar{x}_3, x_2\bar{x}_1, x_2\bar{x}_3$. Let's make a new table in which rows will be primary implicants, and columns coincide with minterms of PDNF. Put marks in those cells where the primary implicants are part of the corresponding minterm (table 3):

Since there are two marks in each column of the table, there are no significant implicants that could be removed from the table. Therefore, it remains to choose the minimum coverage at maximum intervals. As can be seen from table 2, there are 3 *minimum DNF*:

$$1) f(x_1, x_2, x_3) = \oplus_1^3 x_i = x_1\bar{x}_2 + x_1\bar{x}_3 + x_2\bar{x}_3 + x_3\bar{x}_1;$$

$$2) f(x_1, x_2, x_3) = \oplus_1^3 x_i = x_1 \bar{x}_2 + x_1 \bar{x}_3 + x_2 \bar{x}_1 + x_3 \bar{x}_2; \quad (9)$$

$$3) f(x_1, x_2, x_3) = \oplus_1^3 x_i = x_1 \bar{x}_2 + x_1 \bar{x}_3 + x_2 \bar{x}_1 + x_3 \bar{x}_1.$$

	$x_3 \bar{x}_1 \bar{x}_2$	$x_1 \bar{x}_2 x_3$	$x_1 \bar{x}_2 \bar{x}_3$	$x_1 \bar{x}_3 x_2$	$x_2 \bar{x}_1 x_3$	$x_2 \bar{x}_1 \bar{x}_3$
$x_3 \bar{x}_1 \bar{x}_2$	1	$x_3 \bar{x}_2$				$\bar{x}_1 x_3$
$x_1 \bar{x}_2 x_3$	$x_3 \bar{x}_2$	1	$x_1 \bar{x}_2$			
$x_1 \bar{x}_2 \bar{x}_3$		$x_1 \bar{x}_2$	1	$x_1 \bar{x}_3$		
$x_1 \bar{x}_3 x_2$			$x_1 \bar{x}_3$	1		$x_2 \bar{x}_3$
$x_2 \bar{x}_1 x_3$	$\bar{x}_1 x_3$				1	$x_2 \bar{x}_1$
$x_2 \bar{x}_1 \bar{x}_3$				$x_2 \bar{x}_3$	$x_2 \bar{x}_1$	1

Table 2. Construction of minimum DNF by Quine's method

	$x_3 \bar{x}_1 \bar{x}_2$	$x_1 \bar{x}_2 x_3$	$x_1 \bar{x}_2 \bar{x}_3$	$x_1 \bar{x}_3 x_2$	$x_2 \bar{x}_1 x_3$	$x_2 \bar{x}_1 \bar{x}_3$
$x_3 \bar{x}_2$	✓	✓				
$\bar{x}_1 x_3$	✓				✓	
$x_1 \bar{x}_2$		✓	✓			
$x_1 \bar{x}_3$			✓	✓		
$x_2 \bar{x}_3$				✓		✓
$x_2 \bar{x}_1$					✓	✓

Table 3. Marking

Since in expression (9) the number of variables, the number of objections and the number of terms are the same, the preference of any of the minimized DNFs cannot be given: each of 1) -3) minimized DNFs is equal.

1.3. Perfect conjunctival normal form (PCNF), minimized CNF

Since the procedure of Quine's method is also used to construct a minimized DCNF, as can be seen from the definition of "Exclusive OR" of n variables (3), which is presented in the form of DCNF, none of the logical variables will be "absorbed". Therefore, DCNF and minimized CNF coincide directly with the definition (3) of "Exclusive OR" of n variables.

1.4. Zhegalkin polynomial for the case $n = 3$

According to the known theorem, each Boolean function of n variables can be represented as a Zhegalkin polynomial through logical operations conjunction and "Exclusive OR", $n=2$. Since for the case $n = 2$ "Exclusive OR" of n variables coincides "Exclusive OR", then the Zhegalkin polynomial is "Exclusive OR". For the case $n > 2$ the Zhegalkin polynomial must be constructed. Construct a Zhegalkin polynomial for the case for $n = 3$:

$$f(x_1, x_2, x_3) = \bigoplus_1^3 x_i = a_{123}x_1x_2x_3 \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \oplus a_{23}x_2x_3 \oplus a_1x_1 \oplus a_2x_2 \oplus a_3x_3 \oplus a_0 \quad (10)$$

We use the method of indefinite coefficients to find the unknown coefficients of the Zhegalkin polynomial. To find the coefficients use the truth table 1. Substitute the logical values x_1, x_2, x_3 of the variables corresponding to 0 and 1, and equate the Zhegalkin polynomial for these values with the values of the function $f(x_1, x_2, x_3)$ from the truth table. Remember "Exclusive OR", which is present in the expression of the Zhegalkin polynomial, is equal to logical 1 only when the logical variables take opposite values, i.e. 0 and 1. Otherwise, we have a logical 0.

$$f(0;0;0) = 0 = 0 \oplus a_0 \Rightarrow a_0 = 0;$$

$$f(0;0;1) = 0 = a_3 \oplus a_0 = 1 \Rightarrow a_3 = 1;$$

$$f(0;1;0) = 0 = a_2 \oplus a_0, a_0 = 0 \Rightarrow a_2 = 1;$$

$$f(0;1;1) = 0 = a_{23} \oplus a_2 \oplus a_3 \oplus a_0 = 1, a_0 = 0, a_2 = 1, a_3 = 1 \Rightarrow a_{23} = 1;$$

$$f(1;0;0) = a_1 \oplus a_0 = 1, a_0 = 0 \Rightarrow a_1 = 1;$$

$$f(1;0;1) = 1 = a_{13} \oplus a_1 \oplus a_3 \oplus a_0; a_0 = 0, a_1 = a_3 = 1 \Rightarrow a_{13} = 1;$$

$$f(1;1;0) = 1 = a_{12} \oplus a_1 \oplus a_2 \oplus a_0; a_0 = 0, a_1 = a_2 = 1 \Rightarrow a_{12} = 1;$$

$$f(1;1;1) = 1 = a_{123} \oplus a_{12} \oplus a_{13} \oplus a_{23} \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_0; a_0 = 0, \\ a_1 = a_2 = a_3 = a_{12} = a_{13} = a_{23} = 1, \Rightarrow a_{123} = 0$$

So, we have a Zhegalkin polynomial to "Exclusive OR" of $n=3$ in the following form:

$$x_1 \oplus x_2 \oplus x_3 \oplus x_2x_3 \oplus x_1x_2 \oplus x_2x_3. \quad (11)$$

1.5. Classification of Post

1. "Exclusive OR" of n variables, including $n = 3$, on the zero set $f(0,0,\dots,0) = 0$, therefore belongs to the class T_0 ;
2. "Exclusive OR" of n variables, including $n = 3$, does not belong to the class T_1 , since $f(1,1,\dots,1) = 0 \neq 1$;
3. "Exclusive OR" of $n = 3$ does not belong to the class of linear functions L , since the Zhegalkin polynomial for this case contains the products of variables.
4. "Exclusive OR" of $n = 3$ does not belong to the class of monotonic functions, since for values $\{1\}$ and $\{0\}$ the condition of monotonicity is not fulfilled.
5. "Exclusive OR" of $n = 3$ is not a self-double function as $f(0;0;0) = f(1;1;1)$, but condition $f(0;0;0) \neq f(1;1;1)$ must be satisfied.

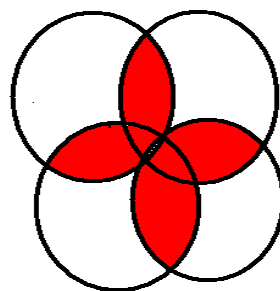
2. Case $n = 4$

For this case "Exclusive OR" of $n = 4$ variables is the next:

$$f(x_1, x_2, x_3) = \bigoplus_1^4 x_i = (x_1 + x_2 + x_3 + x_4)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4). \quad (12)$$

Since the truth table for this case contains many columns, we construct a somewhat short truth table. In which part of the terms present in table 1 (case $n = 3$) are omitted. The abbreviated truth table for function (12) has the form:

x_1	x_2	x_3	x_4	\bar{x}_1	\bar{x}_2	\bar{x}_3	\bar{x}_4	$x_1 \vee x_2 \vee x_3 \vee x_4$	$(x_1 \vee x_2 \vee x_3 \vee x_4)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)$
0	0	0	0	1	1	1	1	0	0
0	0	0	1	1	1	1	0	1	1
0	0	1	0	1	1	0	1	1	1
0	0	1	1	1	1	0	0	1	1
0	1	0	0	1	0	1	1	1	1
0	1	0	1	1	0	1	0	1	1
0	1	1	0	1	0	0	1	1	1
0	1	1	1	1	0	0	0	1	1
1	0	0	0	0	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1
1	0	1	0	0	1	0	1	1	1
1	0	1	1	0	1	0	0	1	1
1	1	0	0	0	0	1	1	1	1
1	1	0	1	0	0	1	0	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	1	0	0	0	0	1	0

Table 4. Short truth table for the case $n = 4$ Fig. 3. Venn's diagram for the "Exclusive OR" of $n = 4$ variables

Perform logical multiplication and addition, we have:

$$(x_1 + x_2 + x_3 + x_4)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4) =$$

$$= x_1(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4) + x_2(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4) + x_3(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4) + x_4(\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4).$$

After simplification of products of mutually contradictory variables we have:

$$\begin{aligned} \oplus_1^4 x_i &= x_1(\bar{x}_2 + \bar{x}_3 + \bar{x}_4) + x_2(\bar{x}_1 + \bar{x}_3 + \bar{x}_4) + x_3(\bar{x}_1 + \bar{x}_2 + \bar{x}_4) + x_4(\bar{x}_1 + \bar{x}_2 + \bar{x}_3) = \\ &= x_1\bar{x}_2 + x_1\bar{x}_3 + x_1\bar{x}_4 + x_2\bar{x}_1 + x_2\bar{x}_3 + x_2\bar{x}_4 + x_3\bar{x}_1 + x_3\bar{x}_2 + x_3\bar{x}_4 + x_4\bar{x}_1 + x_4\bar{x}_2 + x_4\bar{x}_3. \end{aligned} \quad (13)$$

Expression (13) is a *disjunctive normal form* of the "Exclusive OR" of $n = 4$ variables.

2.1. Perfect disjunctive normal form

To get a perfect disjunctive normal form, apply the law of addition:

$$\begin{aligned} &x_1\bar{x}_2(x_3 + \bar{x}_3)(x_4 + \bar{x}_4) + x_1\bar{x}_3(x_2 + \bar{x}_2)(x_4 + \bar{x}_4) + x_1\bar{x}_4(x_2 + \bar{x}_2)(x_3 + \bar{x}_3) + x_2\bar{x}_1(x_3 + \bar{x}_3)(x_4 + \bar{x}_4) + \\ &+ x_2\bar{x}_3(x_1 + \bar{x}_1)(x_4 + \bar{x}_4) + x_2\bar{x}_4(x_1 + \bar{x}_1)(x_3 + \bar{x}_3) + x_3\bar{x}_1(x_2 + \bar{x}_2)(x_4 + \bar{x}_4) + x_3\bar{x}_2(x_1 + \bar{x}_1)(x_4 + \bar{x}_4) + \\ &+ x_4\bar{x}_1(x_2 + \bar{x}_2)(x_3 + \bar{x}_3) + x_4\bar{x}_2(x_1 + \bar{x}_1)(x_3 + \bar{x}_3) + x_4\bar{x}_3(x_1 + \bar{x}_1)(x_2 + \bar{x}_2) + x_3\bar{x}_4(x_1 + \bar{x}_1)(x_2 + \bar{x}_2) = \\ &= x_1\bar{x}_2x_3x_4 + x_1\bar{x}_2\bar{x}_3x_4 + x_1\bar{x}_2x_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1x_2\bar{x}_3x_4 + x_1\bar{x}_2\bar{x}_3x_4 + x_1x_2\bar{x}_3\bar{x}_4 + \\ &+ x_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1x_2x_3\bar{x}_4 + x_1\bar{x}_2x_3\bar{x}_4 + x_1x_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3x_4 + \bar{x}_1x_2\bar{x}_3x_4 + \\ &+ \bar{x}_1x_2x_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3\bar{x}_4 + x_1x_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_3x_4 + x_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3\bar{x}_4 + x_1x_2x_3\bar{x}_4 + \\ &+ \bar{x}_1x_2x_3\bar{x}_4 + x_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3x_4 + \bar{x}_1\bar{x}_2x_3x_4 + \bar{x}_1x_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_3\bar{x}_4 + \\ &+ x_1\bar{x}_2x_3x_4 + \bar{x}_1\bar{x}_2x_3x_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3x_4 + \bar{x}_1\bar{x}_2x_3x_4 + \bar{x}_1x_2\bar{x}_3x_4 + \\ &+ \bar{x}_1\bar{x}_2\bar{x}_3x_4 + x_1\bar{x}_2x_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_3x_4 + x_1x_2\bar{x}_3x_4 + \bar{x}_1x_2\bar{x}_3x_4 + \\ &+ x_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2\bar{x}_3x_4 + x_1x_2x_3\bar{x}_4 + \bar{x}_1x_2x_3\bar{x}_4 + x_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_3\bar{x}_4. \end{aligned} \quad (14)$$

In expression (14) a significant number of terms are repeated. After absorbing the same logical terms, we finally have a *perfect disjunctive normal form*:

$$\begin{aligned} \oplus_1^4 x_i &= x_1\bar{x}_2x_3x_4 + x_1\bar{x}_2\bar{x}_3x_4 + x_1\bar{x}_2x_3\bar{x}_4 + x_1x_2\bar{x}_3x_4 + x_1x_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1x_2x_3\bar{x}_4 + \bar{x}_1x_2x_3x_4 + \\ &+ \bar{x}_1x_2\bar{x}_3x_4 + \bar{x}_1x_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3x_4. \end{aligned} \quad (15)$$

2.2. Minimum DNF

As you can see, PDNF has 14 terms. Performing the procedure of finding the minimized DNF by the method of Quine leads to a cumbersome table, in which after the gluing procedure there are 16 minterms of the 3rd order. Of course, the presentation of Quine's algorithm in the format of this work is not convenient. Therefore, to obtain the minimum DNF, we will use Carnot maps.

According to the Carnot map method, we select rectangular areas from units of the largest area, which are powers of 2, and write the corresponding conjunctions for them: K1 - $x_1\bar{x}_2$; K2 - $x_1\bar{x}_3$; K3 - $x_1\bar{x}_4$; K4 - \bar{x}_1x_2 ; K5 - \bar{x}_1x_3 ; K6 - \bar{x}_1x_4 . Combining them with the help of logical addition we have a *minimized DNF*:

$$x_1\bar{x}_2 + x_1\bar{x}_3 + x_1\bar{x}_4 + \bar{x}_1x_2 + \bar{x}_1x_3 + \bar{x}_1x_4$$

It is easy to see that, as in the case of $n = 3$, the minimized CNF for the case of $n = 4$ coincides with expression (12).

$x_1x_2 \setminus x_3x_4$	00	01	11	10
00	0	1	1	1
01	1	1	1	1
11	1	1	0	1
10	1	1	1	1

Table 5. Minimization with Carnot maps

2.3. Zhegalkin polynomial $n = 4$

We write Zhegalkin's polynomial in the form:

$$\begin{aligned}
 f(x_1, x_2, x_3, x_4) = & a_{1234}x_1x_2x_3x_4 \oplus a_{234}x_2x_3x_4 \oplus a_{134}x_1x_3x_4 \oplus a_{124}x_1x_2x_4 \oplus a_{134}x_1x_3x_4 \oplus a_{123}x_1x_2x_3 \oplus \\
 & \oplus a_{34}x_3x_4 \oplus a_{24}x_2x_4 \oplus a_{23}x_2x_3 \oplus a_{14}x_1x_4 \oplus a_{13}x_1x_3 \oplus a_{12}x_1x_2 \oplus a_{34}x_3x_4 \oplus a_{24}x_2x_4 \oplus a_{23}x_2x_3 \oplus \\
 & \oplus a_{14}x_1x_4 \oplus a_{13}x_1x_3 \oplus a_{12}x_1x_2 \oplus a_4x_4 \oplus a_3x_3 \oplus a_2x_2 \oplus a_1x_1 \oplus a_0.
 \end{aligned} \tag{16}$$

We will use the method of indefinite coefficients:

$$f(0;0;0;0) = a_0 = 0 \Rightarrow a_0 = 0;$$

$$f(1;0;0;0) = a_0 \oplus a_1 = 1, a_0 = 0 \Rightarrow a_1 = 1;$$

$$f(0;1;0;0) = a_0 \oplus a_2 = 1, a_0 = 0 \Rightarrow a_2 = 1;$$

$$f(0;0;1;0) = a_0 \oplus a_3 = 1, a_0 = 0 \Rightarrow a_3 = 1;$$

$$f(0;0;0;1) = a_0 \oplus a_4 = 1, a_0 = 0 \Rightarrow a_4 = 1;$$

$$f(1;1;0;0) = a_0 \oplus a_1 \oplus a_2 \oplus a_{12} = 1, a_0 = 0, a_1 = 1, a_2 = 1 \Rightarrow a_{12} = 1;$$

$$f(1;0;1;0) = a_0 \oplus a_1 \oplus a_3 \oplus a_{13} = 1, a_0 = 0, a_1 = 1, a_3 = 1 \Rightarrow a_{13} = 1;$$

$$f(1;0;0;1) = a_0 \oplus a_1 \oplus a_4 \oplus a_{14} = 1, a_0 = 0, a_1 = a_4 = 1 \Rightarrow a_{14} = 1;$$

$$f(0;1;1;0) = a_0 \oplus a_2 \oplus a_3 \oplus a_{23} = 1, a_0 = 0, a_2 = a_3 = 1 \Rightarrow a_{23} = 1;$$

$$f(0;1;0;1) = a_0 \oplus a_2 \oplus a_4 \oplus a_{24} = 1, a_0 = 0, a_2 = a_4 = 1 \Rightarrow a_{24} = 1;$$

$$f(0;0;1;1) = a_0 \oplus a_3 \oplus a_4 \oplus a_{34} = 1, a_0 = 0, a_3 = a_4 = 1 \Rightarrow a_{34} = 1.$$

We take into account the found coefficients, without writing them once more, then we have:

$$f(1;1;1;0) = a_0 \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_{12} \oplus a_{13} \oplus a_{23} \oplus a_{123} = 1, \Rightarrow a_{123} = 1;$$

$$f(1;1;0;1) = a_0 \oplus a_1 \oplus a_2 \oplus a_4 \oplus a_{12} \oplus a_{14} \oplus a_{24} \oplus a_{124} = 1 \Rightarrow a_{124} = 1;$$

$$f(1;0;1;1) = a_0 \oplus a_1 \oplus a_3 \oplus a_4 \oplus a_{13} \oplus a_{14} \oplus a_{34} \oplus a_{134} = 1 \Rightarrow a_{134} = 1;$$

$$f(0;1;1;1) = a_0 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_{23} \oplus a_{24} \oplus a_{34} \oplus a_{234} = 1 \Rightarrow a_{234} = 1;$$

$$f(1;1;1;1) = a_0 \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_{12} \oplus a_{13} \oplus a_{14} \oplus a_{23} \oplus a_{24} \oplus a_{34} \oplus a_{123} \oplus a_{124} \oplus a_{134} \oplus a_{234} \oplus a_{1234} = 1 \Rightarrow a_{1234} = 1$$

After finding all the coefficients, we have the following form of the Zhegalkin polynomial:

$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_1x_2 \oplus x_1x_3 \oplus x_1x_4 \oplus x_2x_3 \oplus x_2x_4 \oplus x_3x_4 \oplus x_1x_2x_3 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_2x_3x_4 \quad (17)$$

As we can see from equations (11) and (17), the Zhegalkin polynomial is an order of magnitude lower than the the "Exclusive OR" : for $n = 3$ it is a second-order polynomial, and in the case of $n = 4$ it is a third-order polynomial.

2.4. Post classification of "Exclusive OR", $n = 4$

1. $f(0;0;0;0) = 0$, therefore *belongs to the class* T_0 ;
2. $f(1,1,1,1) = 0 \neq 1$, therefore *does not belong to the class* T_1 ;
3. Zhegalkin's polynomial for $n = 4$ contains the products of variables, so the "Exclusive OR" *does not belong to the class of linear functions* L ;
4. "Exclusive OR" of $n = 4$ does not belong to the class of *monotonic functions*: for values $\{1\}$ and $\{0\}$ the condition of monotonicity is not fulfilled;
5. "Exclusive OR" of $n = 4$ is not a self-double function, since $f(0;0;0;0) = f(1;1;1;1)$.

Conclusions

1. In this paper, a new Boolean function is proposed – "exclusive OR" of n variables, its main properties are studied.

2. For cases $n = 3, 4$ the disjunctive normal form (DNF), the perfect disjunctive normal form (PDNF), the minimum disjunctive normal form are constructed. It is established that the conjunctive normal form, the perfect conjunctive normal form and the minimized conjunctive normal form coincide with the definition of "exclusive OR" of n variables.

3. The classification of the Post is performed. It is established that the "exclusive OR" for $n = 3, 4$ belongs to the class T_0 , does not belong to the classes T_1 and L is not monotonic and self-double.

References:

1. Bardachov, Yu. M., Sokolova, N. A., Khodakov, V. Ye. (2007). Discrete mathematics. – Kyiv: "Vysha shkola". – 384 p. [in Ukrainian]
2. Nikolsky, Yu. V., Pasichnyk, V. V., Scherbyna, Yu. M. (2007). Discrete mathematics. – Kyiv: BHV. – 368 p. [in Ukrainian]
3. Borisenko, O. A. (2008). Discrete mathematics. – Sumy: "Universitetska knyga". – 256 p. [in Ukrainian]
4. Bondarenko, M. F., Bilous, N. F., Rutkas, A. G. (2004). Computer discrete mathematics. – Kharkiv: "Company SMIT". – 480 p. [in Ukrainian]
5. Krivy, S. L. (2018). Problems on discrete mathematics. – Chernivtsy, Kyiv. – 456 p. [in Ukrainian]

ЛУК'ЯНОВ Петро Володимирович,

кандидат фізико-математичних наук, старший науковий співробітник НАН України, доцент, АСОІУ, ФІОТ, НТТУ "КПІ"

МУХА Ірина Павлівна,

кандидат технічних наук, доцент, заст. завідувача кафедри АСОІУ, ФІОТ, НТТУ “КПІ”.

БУЛЕВА ФУНКЦІЯ “ВИКЛЮЧАЮЧЕ АБО” ДЛЯ n ЗМІННИХ

Анотація. У даній роботі запропоновано нову булеву функцію – “виключаюче або” для n змінних. Дано означення цієї функції, вивчено основні її властивості. Зокрема, встановлено (теорема), що “виключаюче або” є симетричною функцією відносно перестановки аргументів, а у випадку $n=2$ співпадає з додаванням за модулем 2. Для випадків $n=3;4$ побудовано диз’юнктивну нормальну форму (ДНФ), досконалу диз’юнктивну нормальну форму (ДДНФ), мінімальну диз’юнктивну нормальну форму. Зокрема, у випадку $n=3$ за допомогою методу Квайна знайдено три рівноправних мінімальних диз’юнктивних нормальних форми. Встановлено, що кон’юнктивна нормальна форма, досконала кон’юнктивна нормальна форма та мінімізована кон’юнктивна нормальна форма співпадає з означенням “виключаюче або” для n змінних. Для випадків $n=3,4$ побудовано поліноми Жегалкіна, які мають порядок на одиницю нижче а ніж ДДНФ. Виконано класифікацію Поста. Встановлено, що “виключаюче або” для $n=3,4$ належить класу T_0 , не належить класам T_1 та L , не є монотонною та само двоїстою функцією.

Метою статті є введення нової функції “виключаюче або” для n змінних та вивчення її основних властивостей.

Результати та висновки. У даній роботі запропоновано нову булеву функцію – “виключаюче або” для n змінних, вивчено основні її властивості. Для випадків $n=3;4$ побудовано диз’юнктивну нормальну форму (ДНФ), досконалу диз’юнктивну нормальну форму (ДДНФ), мінімальну диз’юнктивну нормальну форму. Встановлено, що кон’юнктивна нормальна форма, досконала кон’юнктивна нормальна форма та мінімізована кон’юнктивна нормальна форма співпадає з означенням “виключаюче або” для n змінних. Виконано класифікацію Поста. Встановлено, що додавання за модулем $n=3,4$ належить класу T_0 , не належить класам T_1 та L , не є монотонною та само двоїстою.

Ключові слова: булева функція, “виключаюче або” для n змінних.

Одержано редакцією 20.02.2020 р.
Прийнято до публікації 27.04.2020 р.

УДК 519.6

DOI 10.31651/2076-5886-2020-1-22-31

PACS 02.60.-x, 02.60.Pn, 02.70.Wz

ГОЛОВНЯ Борис Петрович,

д.т.н., доцент, доцент кафедри прикладної математики та інформатики Черкаського національного університету імені Богдана Хмельницького
e-mail: bpgolovnya@gmail.com
ORCID 0000-0002-9242-3937

РАСЧЕТ ЛАМИНАРИЗАЦИИ ТЕЧЕНИЯ ПРИ СОВПАДАЮЩЕЙ ТУРБУЛЕНТНОЙ СМЕШАННОЙ КОНВЕКЦИИ НА ВЕРТИКАЛЬНОЙ ПОВЕРХНОСТИ

Аннотация. На основе анализа уравнений турбулентной смешанной конвекции предложены уравнения, описывающие изменение теплообмена в пограничном слое при наложении естественной конвекции на вынужденную. По результатам расчетов смешанной конвекции были получены корреляционные зависимости для описания упомянутого снижения теплообмена. Расчеты по этим зависимостям находятся в удовлетворительном соответствии экспериментальным данным.

Ключевые слова: турбулентность, смешанная конвекция, ламинаризация течения.

Постановка задачі

Проблема теплообмена при смешанной конвекции является важной в фундаментальном плане, поскольку она имеет самые различные технические и промышленные приложения. К ним, в частности, относятся: теплообменники, работающие в медленно движущейся окружающей среде; ядерные реакторы, охлаждаемые в режимах аварийного обесточивания главных циркуляционных насосов; этот вид конвекции имеет место в колеблющейся емкости при транспортировке жидкостей, во многих процессах химического производства.

Особый интерес вызывает теплообмен при наложении сопутствующей естественной конвекции на вынужденную турбулентную конвекцию. В этом случае наблюдается заметное снижение турбулентных пульсаций, т.е. происходит так называемая ламинаризация течения.

Задачу расчета теплообмена в ламинаризованном течении нельзя признать полностью решенной. Более того, обобщающие корреляционные зависимости получены далеко не во всех практически важных течениях.

Цель статьи

Задача работы – предложить вариант теоретического описания теплообмена при совпадающей смешанной конвекции.

Условные обозначения

$F = U/U_e$	– безразмерная скорость;
$Gr = g\beta\Delta x^x / \nu^2$	– число Грасгофа;
$Gr^* = g\beta q_w x^4 / \lambda \nu^2$	– число Грасгофа модифицированное;
$K = 0.5U_i U_i, \quad k = 0.5\overline{u_i u_i}$	– кинетическая энергия среднего течения и турбулентности;
$Nu = \alpha x / \lambda$	– число Нуссельта;
$Pr = \nu / a$	– число Прандтля;
$Pr_t = \frac{-\overline{uv} / (\partial U / \partial y)}{-\overline{vt} / (\partial T / \partial y)}$	– число Прандтля турбулентное;
$Re = U_e x / \nu$	– число Рейнольдса;
$\overline{T}, T, t T$	– Средняя, моментальная и пульсационная температура;
$\Delta T = T_w - T_e$	– разность температур;
$\overline{U}, \overline{V}, U, V, u, v$	– Средняя, моментальная и пульсационная скорость в x и y направлениях;
$UT = \frac{\beta g}{U_e^3} \int_0^\infty U(T - T_e)_m dy$	– прирост энергии среднего течения за счет архимедовых сил при наложении естественной конвекции на вынужденную;
$VT = \frac{\beta g}{U_e^3} \int_0^\infty (\overline{v't'})_m dy$	– прирост энергии турбулентности за счет архимедовых сил при наложении естественной конвекции на вынужденную;

Греческие символы

α	— коэффициент теплоотдачи;
$\delta_e = \int_0^{\infty} \frac{U}{U_{\infty}} \left(1 - \left(\frac{U}{U_{\infty}} \right)^2 \right) dy$	— толщина потери энергии;
$\Delta_k = \frac{\nu \sqrt{2 \text{Re}}}{U_e^3} \int_0^{\infty} (k_m - k_f) dy$	— снижение кинетической энергии турбулентности под воздействием архимедовых сил при наложении естественной конвекции на вынужденную; перераспределение кинетической энергии
$\Delta_{\delta} = \frac{d}{dx} 0.5(\delta_{em} - \delta_{ef})$	— среднего течения под воздействием архимедовых сил при наложении естественной конвекции на вынужденную; перераспределение энергии
$\Delta_{Uk} = \frac{1}{U_e^3} \frac{d}{dx} \int_0^{\infty} [(Uk)_m - (Uk)_f] dy$	— турбулентности под воздействием архимедовых сил при наложении естественной конвекции на вынужденную; перераспределение энергии
$\Delta_{\partial U} = \frac{\nu}{U_e^3} \int_0^{\infty} \left[\left(\frac{\partial U}{\partial y} \right)_m^2 - \left(\frac{\partial U}{\partial y} \right)_f^2 \right] dy$	— среднего течения под воздействием архимедовых сил при наложении естественной конвекции на вынужденную; перераспределение диссипации энергии
$\Delta_{\varepsilon} = \frac{1}{U_e^3} \int_0^{\infty} (\varepsilon_m - \varepsilon_f) dy$	— турбулентности под воздействием архимедовых сил при наложении естественной конвекции на вынужденную;
$\varepsilon = \overline{\nu (\partial u_i / \partial x_i)^2}$	— скорость диссипации k ;
$\varepsilon_t = \overline{\alpha (\partial t / \partial x_i)^2}$	— скорость диссипации t^2 ;
ν	— кинематическая вязкость;
$\Theta = (T - T_w) / \Delta T$	— безразмерная температура;
Индексы	
e	— в свободном течении;
f	— вынужденная конвекция;
m	— смешанная конвекция;
w	— на стенке.

Методы решения

Из экспериментальных данных хорошо известно, что при наложении естественной конвекции на вынужденное турбулентное течение происходит так называемая ламинаризация течения, т.е. снижается уровень турбулентных пульсаций, падает энергия турбулентности, что, в конце концов, ведет к снижению теплообмена.

Для объяснения этого эффекта автором была предложена следующая гипотеза. При наложении сопутствующих подъемных сил на вынужденное течение в первую

очередь это ведет к изменению профиля скорости и, в результате, возрастанию внутреннего трения. Увеличенное трение ведет к увеличению энергозатрат на его преодоление. Но в начале развития процесса прирост энергии за счет подъемных сил очень мал и не может компенсировать дополнительных расходов. В результате часть энергии потока, идущая на поддержание турбулентности, начинает убывать. Как следствие в течении начинается ламинаризация. Таким образом, ламинаризация происходит в результате перераспределения энергии между средним течением и турбулентностью. Расчеты показывают отчетливую зависимость между снижением теплообмена и турбулентной энергией (см. рис. 1). По этой причине, для прогнозирования развития ламинаризации нам в первую очередь необходимы выражения, описывающие снижение энергии турбулентности.

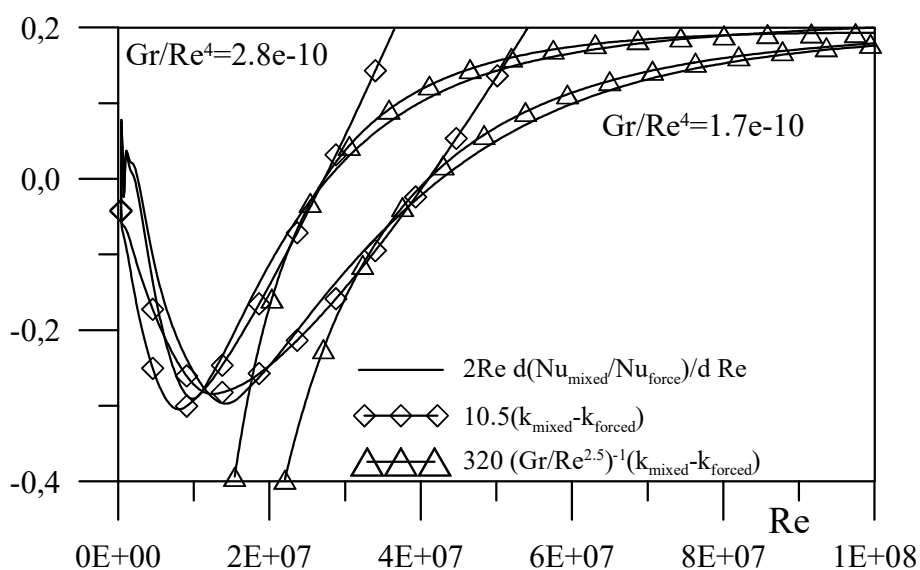


Рис. 1. Связь между изменением теплообмена и энергии турбулентности при смешанной конвекции на вертикальной пластине.

Как известно, уравнения кинетической энергии осредненного течения и кинетической энергии турбулентности в рамках пограничного слоя могут быть записаны в виде (1) и (2)

$$U \frac{\partial K}{\partial x} + v \frac{\partial K}{\partial y} = -\frac{\partial}{\partial x} U \left(\frac{P}{\rho} + \frac{U^2}{2} \right) - \frac{\partial}{\partial y} V \left(\frac{P}{\rho} + \frac{U^2}{2} \right) - U \frac{\partial}{\partial y} \overline{uv} + vU \frac{\partial^2 U^2}{\partial y^2}, \quad (1)$$

$$U \frac{\partial k}{\partial x} + V \frac{\partial k}{\partial y} = v \frac{\partial^2 k}{\partial y^2} - \overline{uv} \frac{\partial U}{\partial y} - \frac{\partial}{\partial y} \overline{v \left(\frac{p}{\rho} + k \right)} - \varepsilon. \quad (2)$$

В работе Хинце [2] на основе этих уравнений было предложено уравнение, описывающее баланс полной кинетической энергии потока в случае вынужденной конвекции в турбулентном пограничном слое.

$$\frac{d}{dx} 0.5 \delta_{ef} - \frac{1}{U_e^3} \frac{d}{dx} \int_0^\infty (Uk)_f dy = \frac{v}{U_e^3} \int_0^\infty \left(\frac{\partial U}{\partial y} \right)_f^2 dy + \frac{1}{U_e^3} \int_0^\infty \varepsilon_f dy. \quad (3)$$

Слагаемые этого уравнения, слева направо, описывают

- 1 – изменение энергии осредненного течения,
- 2 – изменение энергии турбулентности,
- 3 – диссипация энергии осредненным течением,
- 4 – диссипация энергии турбулентности.

Индекс f означает, что слагаемые относятся к случаю вынужденной конвекции.

В данной изучается воздействие сопутствующей подъемной силы на баланс энергии течения. Поэтому учтем в уравнениях (1) и (2) влияние подъемных сил, т.е. перепишем их в виде

$$U \frac{\partial K}{\partial x} + \nu \frac{\partial K}{\partial y} = -\frac{\partial}{\partial x} U \left(\frac{P}{\rho} + \frac{U^2}{2} \right) - \frac{\partial}{\partial y} V \left(\frac{P}{\rho} + \frac{U^2}{2} \right) - U \frac{\partial}{\partial y} \overline{uv} + \nu U \frac{\partial^2 U^2}{\partial y^2} - g\beta U (T - T_e)$$

и

$$U \frac{\partial k}{\partial x} + V \frac{\partial k}{\partial y} = \nu \frac{\partial^2 k}{\partial y^2} - \overline{uv} \frac{\partial U}{\partial y} - \frac{\partial}{\partial y} \overline{\left(\frac{p}{\rho} + k \right)} - \varepsilon - g\beta \overline{vt}.$$

Повторяя действия, проведенные при выводе уравнения (3) получаем

$$\begin{aligned} \frac{d}{dx} 0.5\delta_{em} - \frac{1}{U_e^3} \frac{d}{dx} \int_0^\infty (Uk)_m dy &= \frac{\nu}{U_e^3} \int_0^\infty \left(\frac{\partial U}{\partial y} \right)_m^2 dy + \frac{1}{U_e^3} \int_0^\infty \varepsilon_m dy - \\ & - \frac{\beta g}{U_e^3} \int_0^\infty U(T - T_e)_m dy - \frac{\beta g}{U_e^3} \int_0^\infty \overline{(vt)}_m dy. \end{aligned} \quad (4)$$

Новые члены уравнения, записанные во второй строке, описывают:

- 5 – прирост энергии среднего течения за счет архимедовых сил,
- 6 – прирост энергии турбулентности за счет архимедовых сил.

Индекс m у каждого слагаемого означает, что это уравнение описывает смешанную конвекцию.

Известно, что ламинаризация проявляется относительно вынужденной конвекции, но возникает при смешанной конвекции. Поэтому необходимо учитывать в одном уравнении и смешанный и вынужденный случаи. По этой причине уравнение для вынужденной конвекции (3) вычиталось из уравнения для смешанной конвекции (4). Результат (5) описывает перераспределение энергии между средним течением и турбулентностью за счет наложения на течение подъемных сил.

$$\begin{aligned} \frac{d}{dx} 0.5(\delta_{em} - \delta_{ef}) - \frac{1}{U_e^3} \frac{d}{dx} \int_0^\infty [(Uk)_m - (Uk)_f] dy &= \\ & = \frac{\nu}{U_e^3} \int_0^\infty \left[\left(\frac{\partial U}{\partial y} \right)_m^2 - \left(\frac{\partial U}{\partial y} \right)_f^2 \right] dy + \frac{1}{U_e^3} \int_0^\infty (\varepsilon_m - \varepsilon_f) dy - \frac{\beta g}{U_e^3} \int_0^\infty U(T - T_e)_m dy - \frac{\beta g}{U_e^3} \int_0^\infty \overline{(vt)}_m dy. \end{aligned} \quad (5)$$

Для упрощения записи дальнейших вычислений для слагаемых уравнения (5) введем обозначения:

- 1 Δ_δ – перераспределение кинетической энергии среднего течения;
- 2 Δ_{Uk} – перераспределение энергии турбулентности;
- 3 $\Delta_{\partial U}$ – перераспределение диссипации энергии средним течением;
- 4 Δ_ε – перераспределение диссипации энергии турбулентности;
- 5 UT – прирост энергии среднего течения за счет архимедовых сил;
- 6 VT – прирост энергии турбулентности за счет архимедовых сил.

Сопоставляя уравнение (5) с вышеприведенным качественным анализом, можно сказать следующее. Наложение архимедовых сил на течение в первую очередь ведет к уменьшению первого слагаемого за счет изменения профиля скорости. В свою очередь, так как слагаемое UT еще мало, это ведет к снижению второго слагаемого, т.е. уровня энергии турбулентности. Другими словами, начинается ламинаризация течения.

Для получения количественных характеристик процесса система модельных уравнений решалась дважды. В расчетах использовалась модель турбулентности предложенная автором в работе [6]. Расчеты проводились дважды. Первый раз для случая вынужденной конвекции, т.е. с подъемными силами равными нулю, второй раз с ненулевыми архимедовыми силами, но при тех же начальных условиях и с малым начальным значением Gr^* , т.е. так, чтобы течение оставалось в области вынужденной конвекции. По результатам расчетов находились слагаемые уравнения (5).

После введения обозначений уравнение (5) можно символически записать следующим образом

$$\Delta_\delta - \Delta_{Uk} = \Delta_{\partial U} + \Delta_\varepsilon + UT + VT . \quad (6)$$

Анализ слагаемых уравнения (6) показывает следующее.

Определяющим фактором в уравнении (6) является слагаемое UT . В случае конвекции с постоянным потоком тепла на стенке на основе уравнения переноса тепловой энергии, записанного в автомодельных переменных, для него несложно получить точное уравнение.

$$2Re \frac{\partial F\Theta}{\partial Re} + \frac{\partial V\Theta}{\partial \eta} + 2F\Theta = \frac{1}{Pr} \frac{\partial^2 \Theta}{\partial \eta^2} - \frac{\partial \overline{vt}}{\partial \eta} .$$

Интегрируем это уравнение поперек течения от 0 до бесконечности. Получаем

$$2Re \frac{\partial}{\partial Re} \int_0^\infty F\Theta d\eta + (V\Theta_{\eta=\infty} - V\Theta_{\eta=0}) + 2 \int_0^\infty F\Theta d\eta = \frac{1}{Pr} \left(\frac{\partial \Theta}{\partial \eta} \Big|_{\eta=\infty} - \frac{\partial \Theta}{\partial \eta} \Big|_{\eta=0} \right) - (vt_{\eta=\infty} - vt_{\eta=0}) .$$

Учитывая, что $\Theta_{\eta=\infty} = V_{\eta=0} = \frac{\partial \Theta}{\partial \eta} \Big|_{\eta=\infty} = vt_{\eta=0} = vt_{\eta=\infty} = 0$, $\frac{\partial \Theta}{\partial \eta} \Big|_{\eta=0} = -1$ получаем

$$2Re \frac{d Int}{d Re} + 2Int = \frac{1}{Pr} . \quad (7)$$

Здесь $Int \equiv \int_0^\infty F\Theta d\eta$.

Это уравнение имеет точное решение

$$Int = \frac{0.5}{Pr} \left(1 - \frac{Re_{trans}}{Re} \right), \quad (8)$$

здесь $Re_{trans} = \text{const}$. Отметим некоторые особенности этого уравнения и его решения.

Во первых. Уравнение (7) и решение (8) не зависят от Gr^* . Отсюда следует, что профили U и T меняются таким образом, что $Int = \text{const}$ при $Re = \text{const}$. Значение числа Gr^* никак не влияет на значение Int .

Во вторых. Уравнение (7) и решение (8) не зависят также и от режима течения (ламинарный или турбулентный), но влияние режима может быть учтено с помощью константы Re_{trans} . Из автомодельности течения в ламинарном пограничном слое, очевидно следует, что в ламинарном течении $Re_{trans} = 0$. В то же время расчеты показывают, что в турбулентном течении $Re_{trans} \neq 0$. Интеграл должен быть непрерывной функцией от продольной координаты, следовательно, Re_{trans} должно меняться от нулевого значения в ламинарном режиме до ненулевого в турбулентном. Представляется очевидным, что изменение начинается в переходном режиме. Таким образом, константа Re_{trans} должна быть связана с точкой ламинарно-турбулентного перехода.

Перепишем уравнение (6) в форме

$$\Delta_{Uk} = -UT \left[1 - (\Delta_\delta - VT - \Delta_{\partial U} - \Delta_\varepsilon) / UT \right]$$

и рассмотрим поведение его правой части, а точнее отношения в скобках в области ламинаризации. Для описания поведения введем дополнительную переменную

$\Delta_k \equiv \frac{v\sqrt{2Re}}{U_e^3} \int_0^\infty (k_m - k_f) dy$, непосредственно описывающую снижение энергии турбулентности.

Расчеты показывают, что существуют как минимум три ярко выраженных зоны развития течения (см. рис. 2). В первой из них $\Delta_{Uk} < 0$, Δ_k и отношение Nu_m/Nu_f снижаются. Точка, в которой $\Delta_{Uk} = 0$ задается соотношением $\Delta_\delta = UT + VT + \Delta_{\partial U} + \Delta_\varepsilon$ и, по результатам расчетов, хорошо описывается корреляционным соотношением $Gr^*/Re^{2.8} Pr^{1.13} = 0.09$. Отметим, что эта точка практически совпадает с точкой минимума Δ_k .

В третьей зоне $\Delta_k > 0$ и $d(Nu_m/Nu_f)/dx > 0$. Здесь течение определяется подъемными силами.

Вторая зона является переходной между первой и третьей.

Отношение в скобках в первой зоне хорошо описывается корреляцией, полученной по результатам расчетов

$$(\Delta_\delta - VT - \Delta_{\partial U} - \Delta_\varepsilon) / UT = 1.0892 (Gr^*/Re^4)^{0.00112} (Gr^*/Re^{2.8} Pr^{1.13})^{-0.006}. \quad (9)$$

Учитывая точное решение для UT , получаем корреляцию для Δ_{Uk}

$$\Delta_{Uk} = \frac{Gr^*}{Re^{2.5}} \frac{\sqrt{2}}{Pr} \left(1 - \frac{Re_0}{Re} \right) \left[1 - 1.0892 \left(\frac{Gr^*}{Re^4} \right)^{0.00112} \left(\frac{Gr^*}{Re^{2.8} Pr^{1.13}} \right)^{-0.006} \right]. \quad (10)$$

Корреляционное соотношение для Δ_k было построено в форме, аналогичной соотношению (10).

$$\Delta_k = 1.73 \left(\frac{Gr^*}{Re^4} \right)^{0.028} \frac{Gr^*}{Re^{2.5}} \frac{\sqrt{2}}{Pr} \left(1 - \frac{Re_0}{Re} \right) \left[1 - 0.993 \left(\frac{Gr^*}{Re^{2.8} Pr^{1.13}} \right)^{-0.003} \right].$$

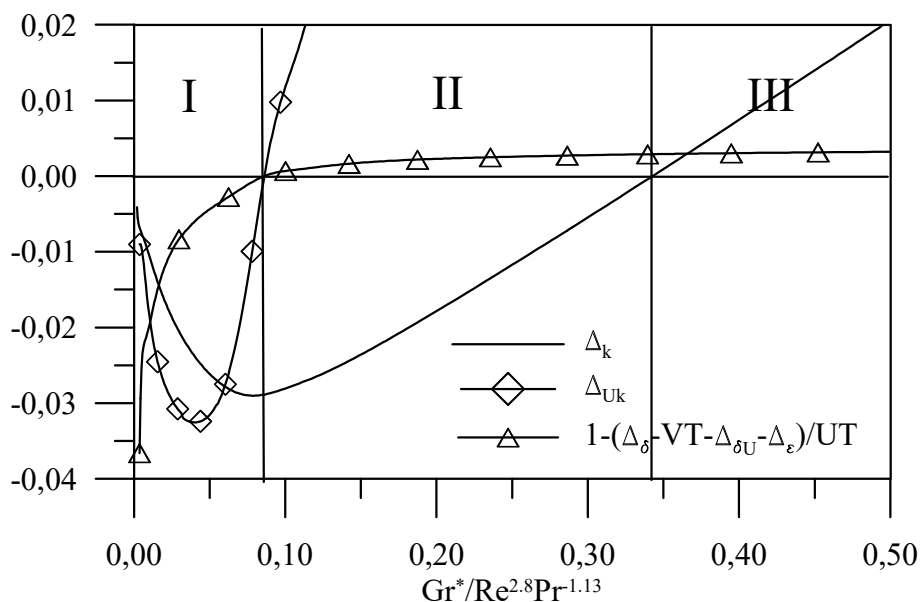


Рис. 2. Три зоны в развитии смешанной конвекции в пограничном слое на вертикальной поверхности.

Теперь мы имеем соотношение, описывающее снижение k под влиянием подъемных сил в первой зоне.

Во второй зоне расчеты показывают, что

$$\Delta_k = \Delta_{k \min} \left(1 - 37.5 \left(\frac{Gr^*}{Re^4} \right)^{1.33} \left(\frac{Re - Re_{\min}}{Pr} \right)^{1.5} \right).$$

Здесь $\Delta_{k \min}$ – минимальное значение Δ_k . Как уже говорилось, это значение достигается в точке $Gr^*/Re^{2.8} Pr^{1.13} = 0.09$. Параметр Re_{\min} соответствует числу Re в этой точке.

Третья зона начинается в точке с $\Delta_k=0$. Здесь

$$\Delta_k = 3.074 \times 10^{-6} \left(\frac{Gr^*}{Re^4} \right)^{-0.25} \left(\frac{Gr^*}{Re^{2.5}} - G_0 \right).$$

В этом выражении G_0 - значение $Gr^*/Re^{2.5}$ в точке, где $\Delta_k=0$.

Как известно, снижение теплообмена происходит вследствие ламинаризации течения, которая в свою очередь, проявляется в снижении турбулентной энергии. По этой причине естественно попробовать обнаружить связь между темпом снижения теплообмена и падением энергии.

На основе расчетов была обнаружена корреляционная зависимость между Δ_k и $d(Nu/Nu_f)/dx$. Она хорошо описывается выражением:

$$2 \operatorname{Re} \frac{d(Nu/Nu_f)}{d \operatorname{Re}} = \begin{cases} 10.5 \operatorname{Pr}^{0.3} \Delta_k & \Delta_k < 0 \\ 402 \operatorname{Pr}^{2/3} (Gr^*/\operatorname{Re}^{2.5})^{-1} \Delta_k & \Delta_k \geq 0 \end{cases}.$$

Учитывая полученные ранее корреляции, мы имеем выражение для расчета производной Nu/Nu_f .

Таким образом, мы можем рассчитать снижение числа Nu под воздействием подъемных сил. Эти расчеты были проведены автором численно. Предполагалось, что $Nu=Nu_f$ при $\operatorname{Re}=\operatorname{Re}_{\text{trans}}$. Значение $\operatorname{Re}_{\text{trans}}=1.0\text{e}+5$ выбиралось из требования наилучшего соответствия между численными и теоретическими расчетами УТ. Результаты показали, что аппроксимационные соотношения хорошо соответствуют расчетам во всем диапазоне Gr^*/Re^4 и Pr .

Для проверки полученных формул вне диапазона расчетов были использованы экспериментальные данные Семенова [7]. Результаты представлены в таблице 1. Значение $\operatorname{Re}_{\text{trans}}=8.0\text{e}+4$ было найдено из требования наилучшего соответствия теоретических и экспериментальных данных. Как видно, оно значительно меньше, чем предыдущих расчетов. По всей видимости этот факт объясняется наличием турбулизатора в экспериментах.

Re	Gr*	Gr*/Re ⁴	Nu расч.	Nu эксп.	Откл. %
1.050E+05	1.732E+13	1.425E +07	298.6	268.0	1.71
1.130E+05	2.174E+13	1.334E+07	288.8	274.0	4.13
1.200E+05	2.564E+13	1.237E +07	299.7	273.0	8.90
2.100E+05	9.315E+12	4.790E +09	450.1	453.0	0.65
2.280E+05	2.135E+13	7.901E+09	477.2	461.0	3.40
2.450E+05	1.606E+13	4.456E+09	508.4	509.0	0.51
2.480E+05	1.806E+13	4.775E+09	510.8	508.0	0.94
2.620E+05	3.520E+13	7.471E+09	530.1	489.0	7.76
2.650E+05	2.189E+13	4.439E+09	537.5	541.0	0.65
2.660E+05	2.392E+13	4.779E+09	538.6	540.0	0.25
2.840E+05	2.983E+13	4.585E+09	568.2	549.0	3.04
2.850E+05	2.909E+13	4.409E+09	567.9	578.0	1.24
3.050E+05	3.648E+13	4.216E+09	598.0	590.0	1.34
3.220E+05	4.595E+13	4.274E+09	622.8	578.0	7.20
3.450E+05	8.661E+13	3.996E+09	658.5	619.0	8.72

Среднеквадратичное отклонение=1.4883 %

Таблица 1. Сравнение результатов расчетов числа Nu при смешанной конвекции на вертикальной поверхности с экспериментальными данными Семенова [7].

Результаты исследования

На основе анализа результатов расчетов смешанной конвекции предложена методика расчета ламинаризации течения. Получаемые по этой методике значения числа Нуссельта хорошо соответствуют экспериментам Семенова [7]. Отметим, что экспериментальные данные лежат в области изменения чисел Re и Gr , не пересекающейся с областью, в которой проводились расчеты.

Список использованной литературы:

1. Петухов, Б. С. Теплообмен в движущейся однофазной среде. Ламинарный пограничный слой. / Б. С. Петухов. – М.: МЭИ, 1993. – 352 с.

2. Петухов, Б. С. Теплообмен при смешанной турбулентной конвекции. / Б. С. Петухов, А. Ф. Поляков. – М.: Наука, 1986. – 192 с.
3. Селиванов, Н. В. Теплообмен высоковязких жидкостей в емкостях. Монография. / Н. В. Селиванов. – Астрахань: Изд-во АГТУ, 2001. – 232 с.
4. Пядишюс, А. Турбулентный теплоперенос в пристенных слоях. / А. Пядишюс, А. Шланчяускас. – Вильнюс: Мокслас, 1987. – 283 с.
5. Хинце, И. О. Турбулентность. / И. О. Хинце. – М.: GIFML. 1963. – 680 с.
6. Golovnya, B. P. Model of turbulent stresses and turbulent heat fluxes transfer / B. P. Golovnya // Refrigeration technology and technology. – 2007. T.110, № 6. – С.49-54.
7. Семенов, Ю. П. Турбулентная смешанная конвекция на вертикальной поверхности. / Ю. П. Семенов // Дисс. докт. т. н. – М.: Московский лесотехнический институт, 1990.

Bibliography:

1. Petukhov, B. S. (1993). Heat exchange in a moving single-phase medium. Laminar boundary layer. – М.: MEI. – 352 p. [in Russian]
2. Petukhov, B. S., Polyakov, A. F. (1986). Heat transfer at mixed turbulent convection. – М.: Nauka, 1986. – 192 p. [in Russian]
3. Selivanov, N. V. (2001). Heat transfer of high-viscosity liquids in containers. Monograph. – Astrakhan: Publishing house of ASTU. – 232 p. [in Russian]
4. Pyadišius, A., Šlančiauskas, A. (1987). Turbulent heat transfer in near-wall layers. - Vilnius: Mokslas. – 283 p. [in Russian]
5. Khintse, I. O. (1963). Turbulence. – М.: GIFML. – 680 p. [in Russian]
6. Golovnya, B. P. (2007). Model of turbulent stresses and turbulent heat fluxes transfer // Refrigeration technology and technology. – V. 110, No. 6. – P. 49-54.
7. Semenov, Yu. P. (1990). Turbulent mixed convection on a vertical surface // Diss. doct. technical sc. – М.: Moscow Forestry Institute. [in Russian]

GOLOVNYA Boris,

Doctor of Technical Science, Professor, Department of Applied Mathematics and Informatics, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

CALCULATION OF FLOW LAMINARIZATION AT ASSISTING TURBULENT MIXED CONVECTION ON A VERTICAL SURFACE

Summary. Calculation of heat transfer at assisting turbulent mixed convection on a vertical surface cannot be considered a solved problem. There are many works devoted to the numerical study of this problem. These simulations show very good agreement with experimental data for the vast majority of parameters. But, at the same time, theoretical descriptions of the problem of flow laminarization are clearly insufficient. In this paper, a variant of the approach to solving this problem is proposed. The correlation dependences obtained on the basis of the proposed approach are in satisfactory agreement with the experimental data

Purpose. The purpose is to offer a simple and understandable explanation of the conjugate gradient method.

Results and conclusion. Note that the experimental data lie in the range of variation of the Re and Gr . Based on the analysis of the results of calculations of mixed convection, a method is proposed for calculating the flow laminarization. Using a program written on the basis of theoretical calculations, the reduction of the Nusselt number under the influence of lifting forces is calculated. The values of the Nusselt number obtained by used method are in good agreement with the experiments of Semenov [7]. Note that the experimental data lie in the range of variation of the Re and Gr numbers, which does not intersect with the range in which the calculations were performed.

Keywords: turbulence; mixed convection; flow laminarization.

Одержано редакцією 05.02.2020 р.
Прийнято до публікації 30.03.2020 р.

УДК 004.94

DOI 10.31651/2076-5886-2020-1-32-44

PACS 02.70.-c

МОРГУН Микола Геннадійович,
аспірант Черкаського національного
університету імені Богдана Хмельницького
e-mail: mykolamorhun4edu@gmail.com
ORCID 0000-0002-5520-3302

ДОСЛІДЖЕННЯ ДИНАМІКИ ГРОМАДСЬКОЇ ДУМКИ У СУЧАСНОМУ СУСПІЛЬСТВІ НА ОСНОВІ МУЛЬТИАГЕНТНОЇ МОДЕЛІ

У статті представлена вдосконалена мультиагентна модель динаміки громадської думки у сучасному суспільстві. Модель базується на взаємодії індивідів та враховує особливості процесу спілкування між ними, наявність політичних партій, засобів масової інформації, територіальне розміщення та густину населення по регіонах.

З використанням отриманої моделі, було проведено ряд чисельних експериментів за допомогою комп'ютера та зроблено ряд висновків. Чисельні експерименти проводилися для різних конфігурацій суспільства: без наявності політичних сил, з однією політичною партією та двома партіями. Для суспільства без політичних сил було досліджено на скільки відхилення від нейтральної позиції змінює ймовірність приходу суспільства до певної думки. Досліджено розвиток подій у суспільстві з однією політичною партією в залежності від характеристик останньої. Також було досліджено вплив ЗМІ та сучасних засобів комунікації на розвиток подій у суспільстві. Було розглянуто випадок, коли у суспільстві вже сформована деяка позиція, а політична партія намагається переконати суспільство у протилежному. Розглянуто проблему консенсусу та поляризації у суспільстві з двома протилежними політичними силами.

Ключові слова: модель, суспільство, суспільна думка, консенсус.

Вступ

Проблема передбачення динаміки громадської думки у суспільстві є важливою актуальною задачею. Від настроїв у суспільстві значною мірою залежать рішення, які це суспільство приймає, що, в свою чергу, впливає на майбутнє самого суспільства. Тобто маючи можливість передбачити динаміку настроїв у суспільстві можна зробити прогноз щодо його подальшого розвитку. Іншим важливим аспектом є можливість передбачення рівня згуртованості суспільства. Так, якщо у суспільстві присутні суттєві розбіжності в поглядах, то таке суспільство може розділитися або протистояння може вийти за рамки дискусії. Можливість передбачення проблемної ситуації дає змогу прийняти певні дії ще до її початку. Окрім того, маючи таку модель, можна спрогнозувати як та чи інша дія змінить ситуацію.

Питанню динаміки поглядів та консенсусу в суспільстві приділено багато уваги як з боку соціо-політичних наук [1-2], так і у сфері математичного моделювання. Для побудови моделей консенсусу та динаміки поглядів у суспільстві використовуються різні підходи: диференційні рівняння [3], клітинні автомати [4-6], мультиагентні моделі [7-11] та інші.

У статті пропонується модель громадської думки у сучасному суспільстві мультиагентного типу, яка побудована з розрахунком на подальшу комп'ютерну реалізацію. Отримана комп'ютерна модель дозволяє легко задавати різні конфігурації суспільства для його подальшого аналізу.

Мета статті

Вдосконалити модель динаміки громадської думки у сучасному суспільстві [12] та провести чисельні експерименти з нею з метою виявлення закономірностей розвитку суспільної думки при різних початкових умовах.

Виклад основного матеріалу

1. Короткий опис та доопрацювання моделі динаміки громадської думки

Описана нижче мультиагентна модель є удосконаленою версією моделі, отриманої в [12]. Розглянемо коротко основні принципові моменти побудови моделі: об'єкти, їх властивості та способи взаємодії між собою.

Основною складовою суспільства є індивід. У моделі будемо розглядати суспільство як деякий набір індивідів (агентів), кожен з яких має свої особливості. Кожен індивід має свою позицію щодо актуальних питань у суспільстві. Зазвичай, для кожної актуальної проблеми у суспільстві є кілька варіантів її вирішення. Індивіди, як члени суспільства, мають своє ставлення до того чи іншого шляху вирішення проблеми. Таке ставлення будемо називати *поглядом*, а набір поглядів – *системою поглядів*. Визначимо, що значення погляду змінюється в межах $[-1, 1]$, де $+1$ – це повна підтримка, 0 – нейтральне ставлення, -1 – повна незгода зі шляхом вирішення даної проблеми.

Якщо деяке число індивідів має спільні погляди, вони можуть утворювати групи однодумців – політичні партії. Якщо індивід входить до політичної партії, то таке членство спричиняє додатковий вплив на індивіда. Так, член політичної партії буде намагатися більше спілкуватися із однопартійцями та більше довіряти партійній пропаганді ніж інші члени суспільства.

Невід'ємним атрибутом сучасного суспільства є засоби масової інформації (ЗМІ). Вони доносять деяку інформацію до членів суспільства чим можуть впливати на їхні погляди. Усі ЗМІ поділимо на дві групи: залежні та незалежні. До залежних будемо відносити такі ЗМІ, які утворені та фінансуються деякою політичною партією. Відповідно, залежне ЗМІ дозволяє виходити в ефір здебільшого лише членам цієї партії, які будуть агітувати суспільство за свої партійні ідеї. Якщо ЗМІ є незалежним, то таке ЗМІ дозволяє висловлювати будь-яку точку зору, тобто, теоретично, будь-який член суспільства може доносити свою думку через нього. Характерною рисою будь-якого ЗМІ є односторонність його дії. Оскільки індивід не може прислухатися до всіх ЗМІ, то покладемо, що існує певна ймовірність того, що індивід сприйме інформацію від деякого ЗМІ. Причому, ймовірність того, що партійній індивід буде сприймати інформацію зі ЗМІ партії вища за ймовірність сприймання інформації з незалежного ЗМІ, і тим більше ЗМІ іншої партії.

Як згадувалося вище, кожен індивід має свої погляди, які можуть змінюватися з часом. Зміна поглядів відбувається за рахунок впливу індивідів один на одного. Індивіди спілкуються зі своїми безпосередніми сусідами, деяким колом віддалених індивідів (цьому сприяє розвиток інформаційних технологій у сучасному суспільстві), а також спілкуються з випадковими індивідами. Розглянемо процес спілкування індивідів детальніше. Покладемо, що якщо два індивіди спілкуються між собою, то вони обмінюються своїми поглядами, чинячи вплив один на одного. Дія впливу (на відміну від попередньої моделі) спрямована лише у бік значення погляду агітатора. Проте, величина впливу, який один індивід чинить на іншого, залежить від поточних значень поглядів обох співрозмовників, причому цей вплив нелінійний. Таку нелінійність добре описує косинус-функція. В результаті, у запропонованій моделі вплив агітатора на співрозмовника описується такою функцією:

$$f(a,b) = \frac{|b-a|}{2} \cdot \left(1 - \frac{\cos(\pi a) + 1}{2}\right) \cdot \frac{\cos(\pi b) + 1}{2} \cdot k \quad (1)$$

де $f(a,b)$ – величина впливу, a – значення погляду індивіда що агітує, b – значення погляду індивіда якого агітують, k – коефіцієнт впливу.

У формулі (1) перший множник характеризує залежність впливу від різниці поглядів індивідів (що не було враховано у попередній моделі), другий відповідає за вплив агітатора на співрозмовника (чим більше агітатор впевнений у своїх поглядах, тим сильніше він агітує), третій – за сприйнятливість індивіда, якого агітують (чим менше індивід впевнений у своїй позиції, тим простіше змінити його думку). На рис. 1 зображено залежність величини впливу від поглядів співрозмовників.

Окрім зміни поглядів у результаті спілкування та під впливом ЗМІ, індивіди можуть дещо змінювати свої погляди у результаті деяких своїх міркувань.

Щодо розміщення на території, кожне суспільство має свої географічні особливості. Окрім певного розташування регіонів суспільства потрібно також враховувати густину населення, домінування певних поглядів чи політичних партій у різних регіонах та межі їх спільного дотику. Для врахування вищезазначеного у моделі, пропонується територію, що займає суспільство, розбити на необхідну кількість квадратних областей, кожна з яких характеризується густиною населення, кількістю прихильників кожної з досліджуваних політичних партій, наявністю та активністю ЗМІ. Розмір такої області залежить від того, наскільки точно потрібно апроксимувати територію в моделі.

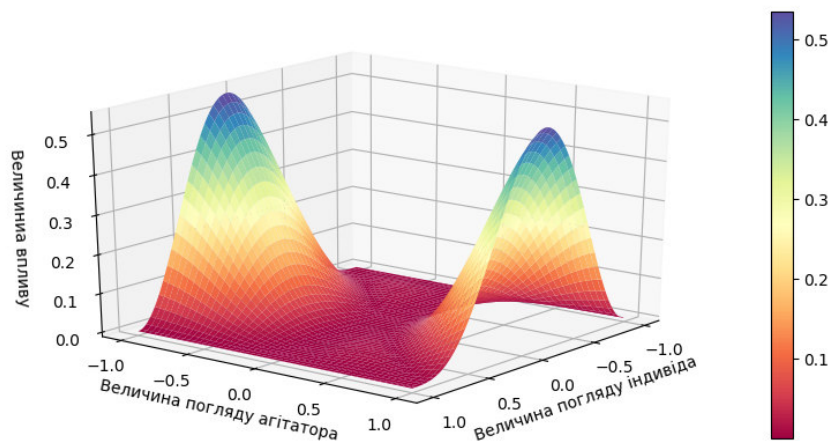


Рис. 1. Залежність величини впливу агітатора на співбесідника від величин їх поглядів ($k=1$)

Алгоритм роботи моделі складається з трьох етапів:

- Обробка вхідних даних та ініціалізація моделі.
Задаємо:
 - кількість та взаємне розташування регіонів в яких розташоване модельоване суспільство
 - питання системи поглядів
 - загальну кількість населення та його розподіл по регіонах
 - кількість політичних партій, їх погляди та кількість прихильників в кожному з регіонів
 - середнє квадратичне відхилення від заданого значення погляду для кожної з політичних партій та безпартійних
 - кількість та типи ЗМІ, частоту їх виходу в ефір
 - кількість індивідів у колі спілкування кожного з індивідів (кількість сусідів, віддалених та випадкових зв'язків)

На основі вище заданої конфігурації:

- розміщуємо індивідів по території (вибираємо координати), відповідно до заданої конфігурації
- визначаємо кількість прихильників кожної з партій та безпартійних для кожного регіону
- знаходимо відповідну кількість найближчих сусідів кожного індивіда
- задаємо постійні зв'язки кожного індивіда з віддаленими
- ініціалізуємо систему поглядів кожного з індивідів, враховуючи що у суспільстві значення кожного погляду розподіляється за Гаусом (нормальним розподілом)
- Моделювання. На кожній ітерації:
 - проводимо обмін поглядами для кожного індивіда
 - враховуємо вплив засобів масової інформації
 - проводимо флуктуацію поглядів кожного індивіда
 Ітерації продовжуємо до досягнення певного моменту в часі.
- Інтерпретація отриманих результатів. Аналізуємо динаміку поглядів у кожному з регіонів та суспільстві в цілому. На основі отриманих даних інтерпретуємо отримані результати.

Оскільки в модель включені стохастичні елементи, то для отримання достовірного результату необхідно провести моделювання деяку кількість разів та усереднити отримані результати. Емпіричним шляхом було визначено, що таких запусків повинно бути не менше 20.

2. Дослідження моделі

2.1 Суспільство без наявності політичних сил

Задамо, що у суспільстві відсутні будь-які політичні сили та початковий середній погляд рівний 0. Промодельюємо ситуацію 100 разів, кожен раз генеруючи початкову конфігурацію (рис. 2). Як видно із результатів моделювання, суспільство може рівно ймовірно прийти як до одного так і до іншого висновку щодо досліджуваного питання. Хоча, на початковий момент часу, у суспільстві немає сформованої думки, проте позиції членів суспільства дещо, хоч і не значно, відрізняються від повністю нейтрального значення. А за рахунок обміну поглядами між індивідами суспільство схиляється на той чи інший бік, приходячи до консенсусу.

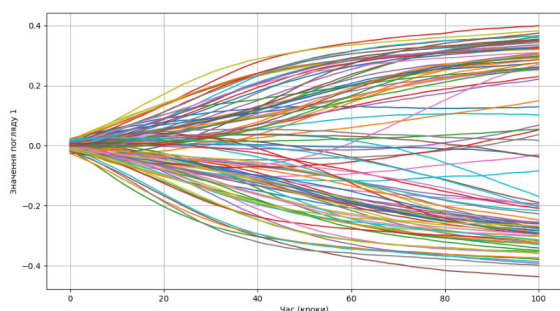


Рис. 2. Зміна середнього погляду у суспільстві з часом при відсутності політичних сил. Математичне очікування середнього значення погляду у суспільстві рівне 0.

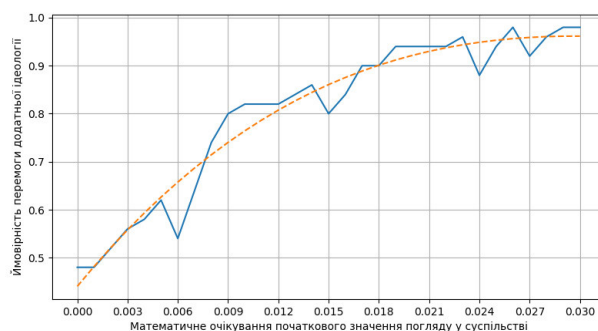


Рис. 3. Залежність ймовірності перемоги додатного погляду від початкового математичного очікування значення погляду у суспільстві.

Тепер дослідимо як зміна початкового середнього погляду у суспільстві буде впливати на ймовірність прийняття суспільством рішення. Для цього будемо

моделювати ситуацію, змінюючи середнє значення погляду в додатній бік (рис. 3). На графіку пунктиром позначена апроксимація поліномом третього степеня.

Як видно з рис. 3, навіть невелика зміна початкового значення середнього по суспільству погляду схиляє суспільство в той чи інший бік. Так, при математичному очікуванні середнього значення погляду у суспільстві 0.03, ймовірність прийняття суспільством саме цього рішення складає близько 0.95. Тобто, для того щоб схили нейтральне суспільство у якийсь бік достатньо досить незначно подіяти на нього.

2.2 Суспільство з однією політичною партією

Покладемо, що члени суспільства рівномірно розподілені по території, тобто для опису достатньо задати одну область. Проаналізуємо як впливає наявність партії на погляди у суспільстві. Для спрощення аналізу, будемо розглядати лише один погляд.

Нехай партійний погляд рівний 0.8 із середньоквадратичним відхиленням 0.033, а чисельність партійців складає 30% від загальної кількості населення. Суспільство складається із 1000 індивідів, кожен з яких спілкується з 10-ма найближчими, 5-ма віддаленими та 4-ма випадковими індивідами. Для членів партії: серед десяти найближчих – 6 партійців, а серед 5 віддалених – 3 є членами партії. Партійці також спілкуються з 4-ма випадковими індивідами. ЗМІ в суспільстві відсутні.

Змоделюємо таку ситуацію та проаналізуємо динаміку погляду в суспільстві. Результати моделювання представлені на рис. 4, з яких видно, що погляди як партійців, так і безпартійних змінюються в бік зближення позицій і рано чи пізно дійдуть до консенсусу, але зміна партійного погляду від його початкового значення є значно меншою ніж у безпартійних, незважаючи на чисельну перевагу останніх. Це пояснюється більш високим рівнем ідейності членів партії.

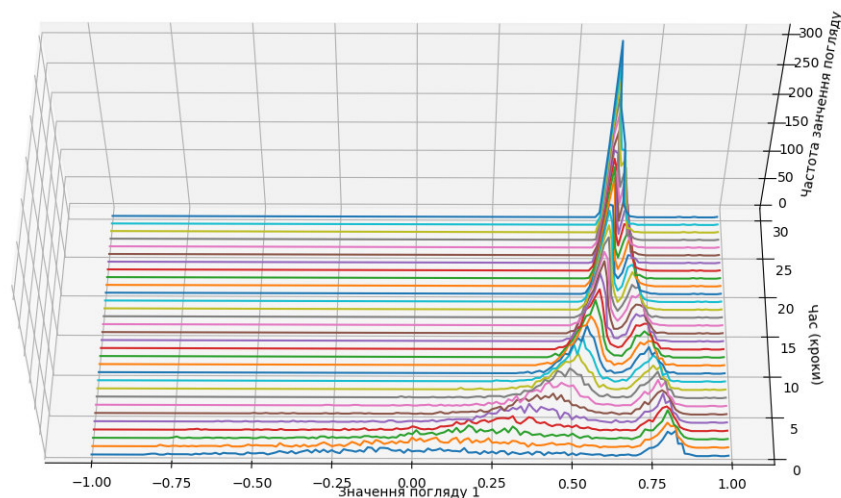


Рис. 4 Динаміка досліджуваного погляду у суспільстві з однією політичною партією

Розглянемо ситуацію, коли члени партії зосереджені в лише одному регіоні та прослідкуємо динаміку погляду на всій території модельованого суспільства. Також проаналізуємо вплив сучасних засобів комунікації на динаміку досліджуваного погляду. Для цього покладемо, що суспільство рівномірно розміщене у 8 горизонтально розташованих областях. Партія *A* присутня у крайній лівій області, величина її погляду рівна +0.9, а чисельність складає 80% від населення цієї області. Нехай в індивідів немає можливостей спілкуватися з віддаленими співбесідниками, а також відсутні ЗМІ. Такими умовами виключаємо можливість використання сучасних засобів зв'язку. Згідно алгоритму моделі початковий стан (перед початком моделювання) генерується випадковим чином. Результати моделювання представлені на рис. 5 та рис. 6.

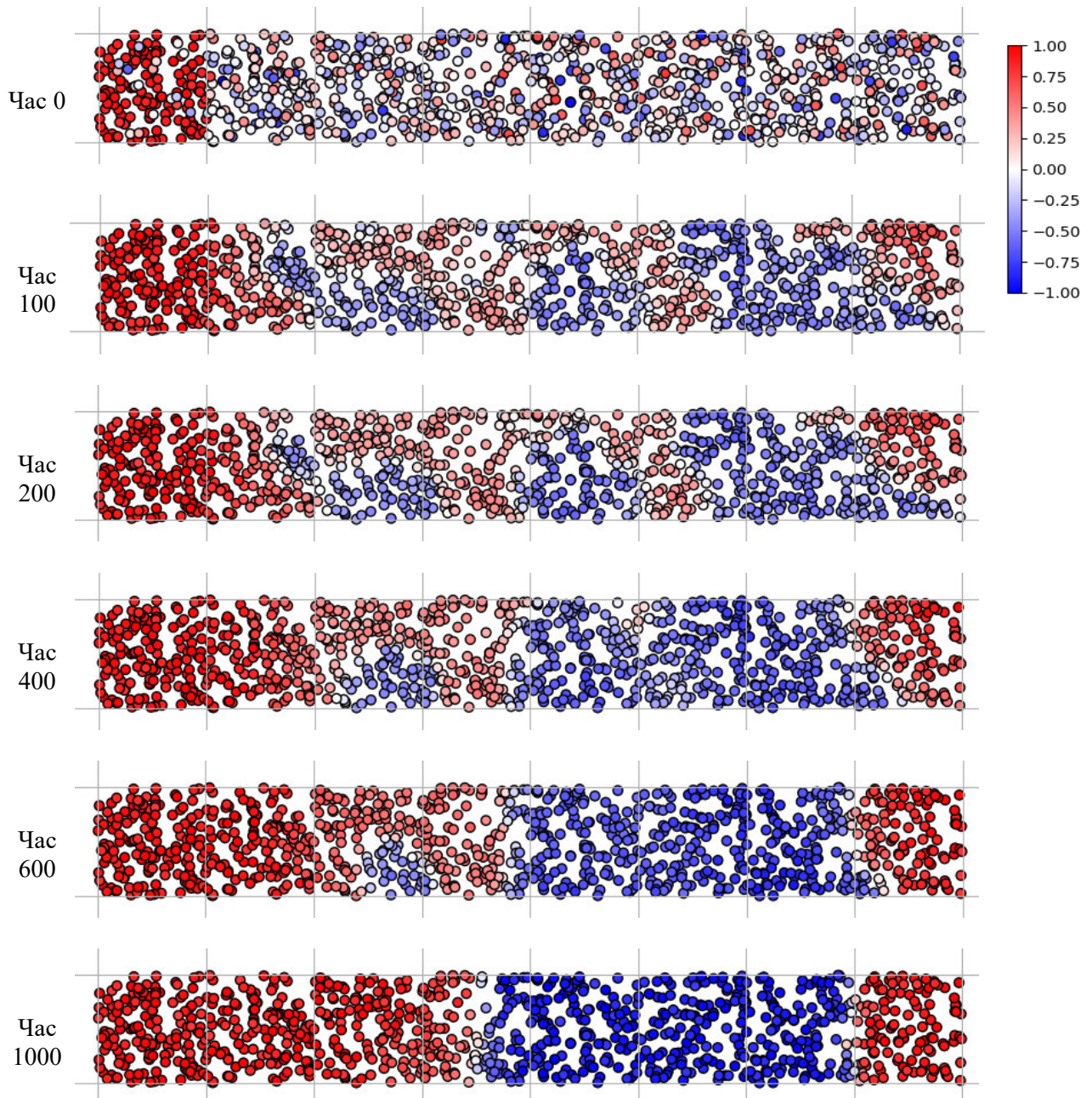


Рис. 5. Динаміка розподілу величини погляду в суспільстві по областях при відсутності віддалених зв'язків

Із результатів моделювання видно, що зміна в поглядах відбувається за час близько 1000 кроків, а потім ситуація стабілізується. Партійна ідея поширилась через умовні кордони перших двох сусідніх областей. А у віддалених областях ситуація може скластися по-різному: майже нейтральні в середньому погляди можуть перерости в протилежні – все залежить від випадкового скупчення однодумців, яке стверджує впевненість цієї групи в правоті своєї ідеї. Це може стати причиною зародження нової партії. Цікавою є також ситуація в області 4: в середньому по області (рис. 6) величина погляду встановлюється близькою до 0 (майже нейтральна позиція), але, як видно з рис. 5, насправді в цій області відбувається поляризація суспільства, що пояснюється впливом сусідніх областей з протилежними поглядами.

Отже, за відсутності засобів для віддаленої комунікації між членами суспільства, у віддалених від партійного осередку областях ситуація з досліджуваним поглядом може розвиватись непередбачувано. І межі ідеологічного протистояння можуть проходити незалежно від кордонів областей. А така розрізненість ідеологій може призвести до розколу держави за ідеологічною ознакою. Наведений вище дослід проводився неодноразово. І щоразу розповсюдження партійної ідеології зупинялося на якійсь межі, оскільки до того часу у віддалених областях стохастичним чином уже була сформована протилежна позиція.

Якщо ж при моделюванні врахувати наявність віддаленого спілкування, то, як показують чисельні експерименти, ситуація кардинально змінюється – партійна ідея доноситься до всіх членів суспільства і, з часом, партійна ідеологія поширюється на все суспільство в цілому.

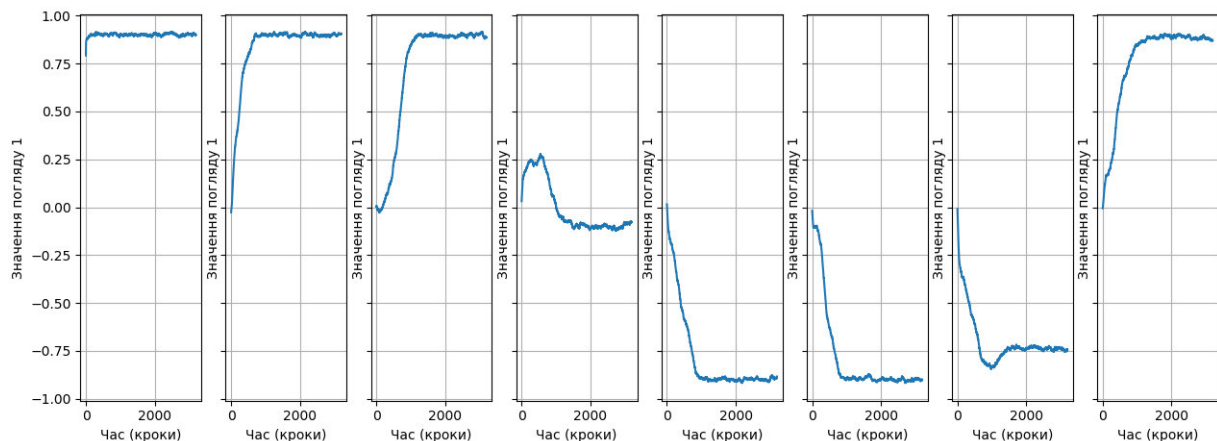


Рис. 6. Динаміка середнього значення погляду по областях при відсутності віддалених зв'язків

Дослідимо ситуацію з однією партією у суспільстві з метою визначення оптимального шляху її розвитку. Для цього проведемо серію експериментів, варіюючи як кількість членів партії, так і величину партійного погляду. Результати моделювання зобразимо у вигляді теплової карти (рис. 7), на якій побудуємо градієнтні криві з метою визначення оптимальних шляхів розвитку партії. Вважатимемо метою партії максимальне поширення своєї ідеології у суспільстві.

Із отриманих результатів можна зробити декілька корисних для партійних лідерів висновків:

- При кількості членів партії, меншій за деяку межу (в даному випадку близько 3%), ідеологія цієї партії не може суттєво впливати на погляди у суспільстві.
- Для максимального поширення партійної ідеології в суспільстві, партії вигідніше зміцнювати свою ідеологічну складову, ніж збільшувати кількість членів партії.
- Чим більша величина партійного погляду та чим більша кількість її членів, тим швидше вона досягає своєї мети.
- Орієнтуючись на градієнтні криві керівництву партії можна визначити оптимальний шлях її розвитку, виходячи зі стану партії на даний момент (кількості її членів та рівня ідеології в даний час).

У попередніх експериментах безпартійна частина суспільства мала погляд у середньому рівний нулю. Змоделюємо ситуацію, коли перед початком моделювання середнє значення суспільного погляду рівне -0.5 , погляд членів партії A має величину $+0.8$, а середньоквадратичні відхилення цих поглядів – 0.33 та 0.033 відповідно.

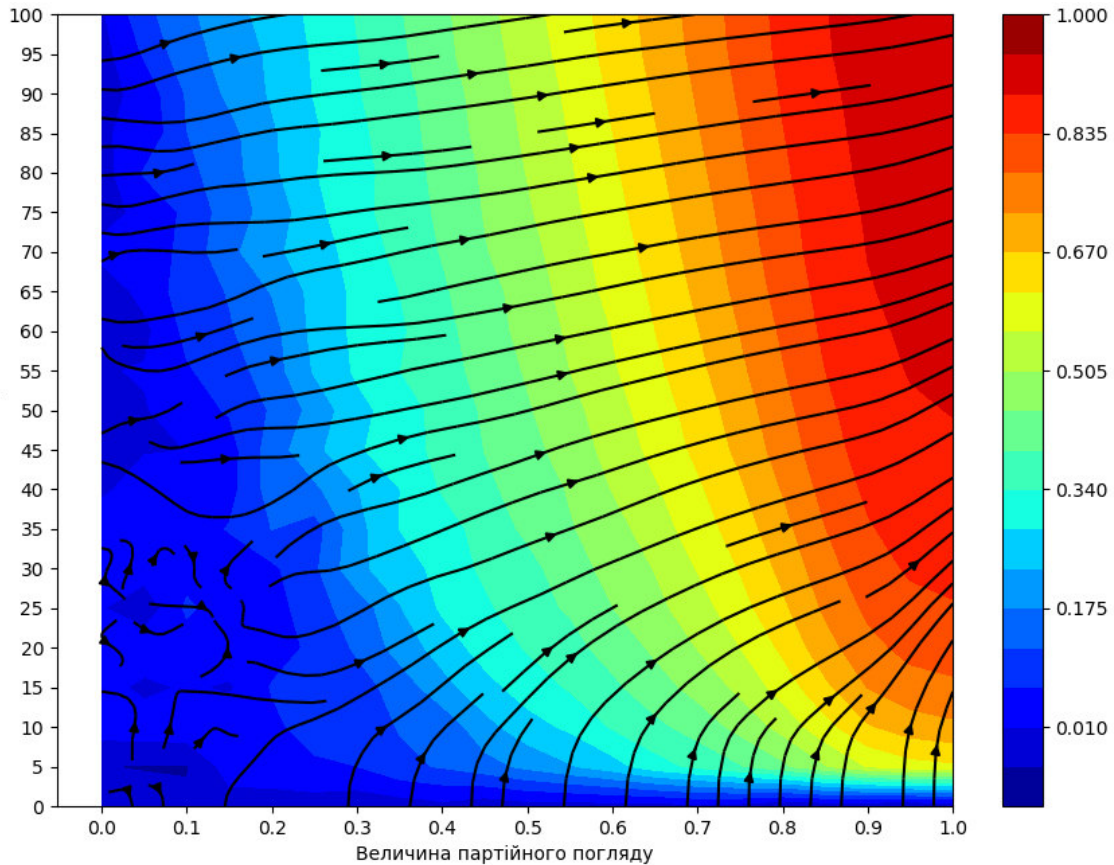


Рис. 7. Теплова карта залежності величини середнього погляду у суспільстві від співвідношення величини партійного погляду до кількості членів цієї партії.

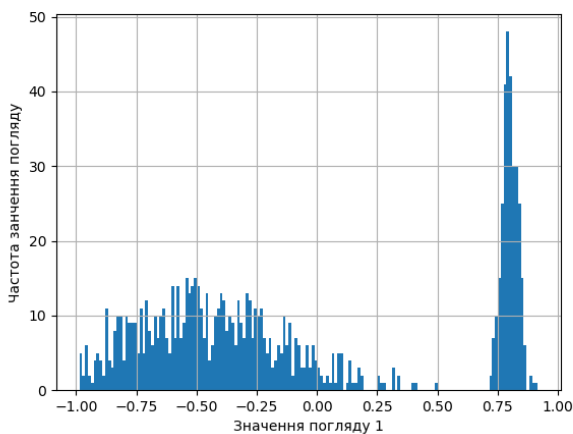


Рис. 8. Розподіл поглядів у суспільстві перед початком моделювання

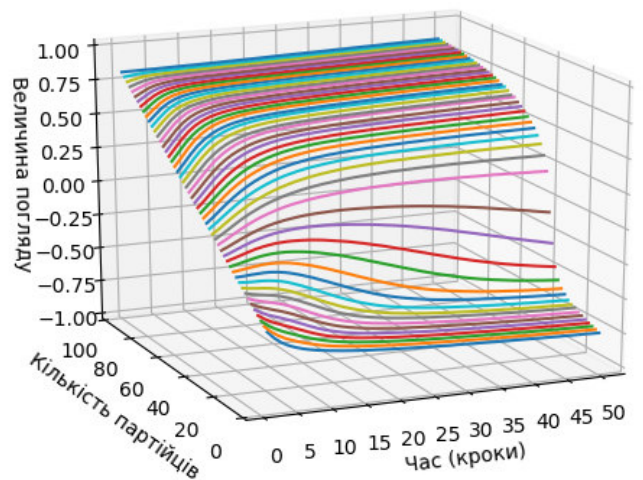


Рис. 9. Динаміка погляду в суспільстві при різній кількості партійців

Прослідкуємо динаміку середнього по суспільству погляду в залежності від кількості членів партії A . На рис. 8, для прикладу, зображено гістограму початкового розподілу погляду у суспільстві, коли чисельність партійців складає 30%. А на рис. 9 – результат моделювання (криві проведені з кроком 2% для кількості партійців).

Як видно з графіків (рис. 9) середній погляд у суспільстві змінюється по-різному в залежності від кількості партійців. Можна виділити три характерних варіанти розвитку

подій: коли величина суспільного погляду лише спадає (характерно для малої кількості членів партії A), коли середня величина суспільного погляду лише зростає (при великій кількості партійців) та коли середній по суспільству погляд спочатку зростає, а потім спадає. Саме в цьому інтервалі величина кількості членів партії суттєво впливає на кінцеву величину погляду в суспільстві. Отже про значущість партії (в сенсі впливу її ідеології на суспільство) можна говорити лише тоді, коли її чисельність перевищить деяке порогове значення. Для конкретної модельованої ситуації, коли безпартійна частина суспільства не поділяє погляду партії, для переконання останніх в правильності партійної ідеології, кількість членів партії повинна перевищувати 30%.

2.3 Суспільство з двома політичними партіями

Доволі поширеною є двопартійна політична система. Тому розглянемо деякі ситуації з двома конкуруючими партіями, які мають протилежні погляди.

У найпростішому випадку, коли суспільство рівномірно розподілене та знаходиться в одному регіоні, з часом, суспільство приходить до консенсусу. Остаточне значення погляду залежить від початкового значення поглядів та кількості прихильників кожної з партій.

Дослідимо, як залежить значення погляду при досягненні консенсусу від співвідношення кількості прихильників партій (назвемо їх A та B). Для цього, зафіксуємо початкове середнє значення погляду для партії A на рівні $+0.8$, а для партії B – на -0.8 , та будемо варіювати кількість прихильників обох партій від 0 до 50%, залишаючи всі інші параметри моделі незмінними. Результати моделювання зображені на рис. 10. Як видно з рис. 10, у випадку значення поглядів на рівні ± 0.8 , для перемоги поглядів партії у суспільстві, цій партії необхідно мати чисельну перевагу хоча б на 7-9%.

Розглянемо ситуацію з іншого боку. Зафіксуємо кількість прихильників кожної з партій на рівні 20% від загальної кількості населення у модельованому суспільстві, та будемо змінювати погляди обох партій. Значення погляду партії A будемо варіювати у межах від 0 до $+1$, а партії B від -1 до 0 (рис. 11).

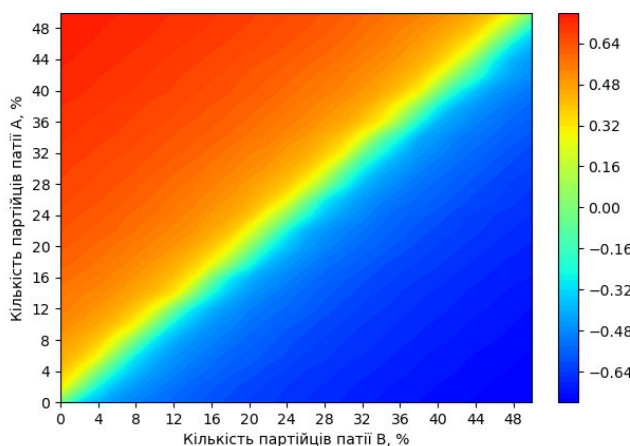


Рис. 10. Залежність величини кінцевого середнього погляду у суспільстві від співвідношення кількості членів партій A та B

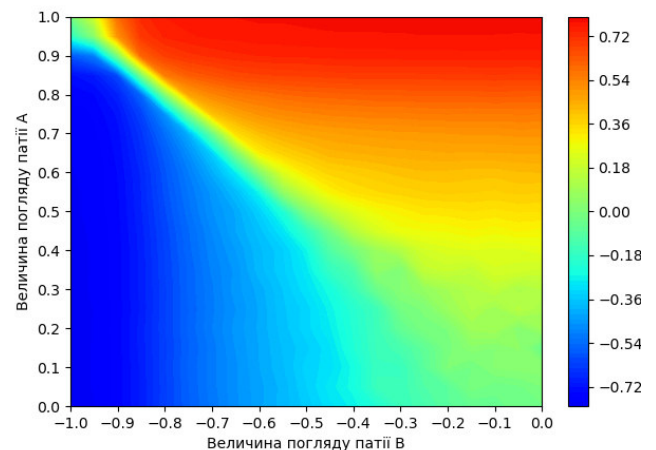


Рис. 11. Залежність величини кінцевого середнього погляду у суспільстві від співвідношення величин поглядів партій A та B

Як видно з рис. 11, для того, щоб погляди партії домінували у суспільстві, партія повинна бути високоідейною (для розглянутої ситуації абсолютне значення погляду

партії повинно бути не меншим 0.8). Проте, якщо партія-конкурент також має високий рівень ідейності, то повністю схилити суспільство на свій бік майже неможливо.

Як показують чисельні експерименти, якщо члени партій A та B , а також безпартійні рівномірно розподілені по суспільству, то таке суспільство не поляризується, а приходиться до консенсусу. У наступному експерименті покладемо, що суспільство складається з двох горизонтально розміщених регіонів. Нехай обидві партії налічують по 30% від загальної кількості членів суспільства, причому прихильники кожної з партій знаходяться в своєму регіоні. Погляди для партій задамо протилежні з абсолютним значенням 0.8. проведемо моделювання такої ситуації 25 разів та виведемо результат кожної спроби на один і той же графік (рис. 12).

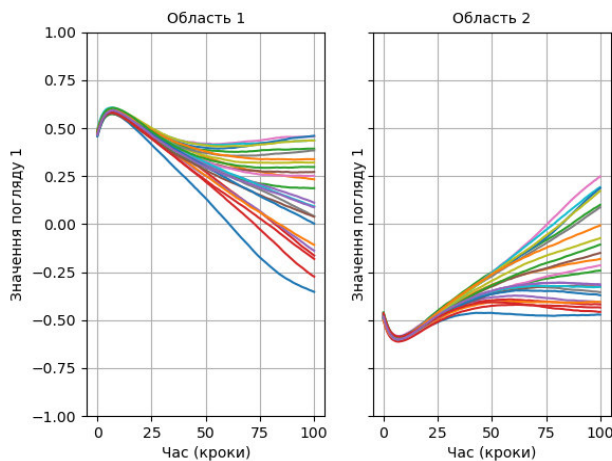


Рис. 12. Динаміка величини досліджуваного погляду по областях, в кожній з яких домінує своя політична партія

Як видно із графіків, кінцевий результат може досить значно варіюватися. Проте всі випадки мають спільну особливість: спочатку суспільство ще більше поляризується, а потім (за умови відсутності зовнішнього впливу чи розпаду на дві частини) поступово починає приходити до якоїсь спільної думки. Додаткові чисельні експерименти показали, що поляризація на початковому етапі відбувається при умові, коли абсолютне значення поглядів обох партій перевищує позначку у 0.55 (за винятком ситуації, коли модельоване суспільство майже не має безпартійних індивідів). Звідси можна зробити висновок, що якщо у суспільстві присутні дві високоідейні політичні сили, кожна з яких займає свою територію, то конфронтація у такому суспільстві неминуча.

Висновки

У статті було вдосконалено мультиагентну модель динаміки громадської думки у сучасному суспільстві. Зокрема, було покращено формулу обміну поглядами між двома індивідами шляхом додання множника, що відповідає за абсолютну різницю між поглядами індивідів. Скориговано напрямки зміни поглядів індивідів при спілкуванні. Додано флуктуації поглядів індивідів, що робить модель більш реальною.

За допомогою отриманої моделі було проведено ряд чисельних експериментів. Було змодельоване суспільство в якому відсутні політичні сили. Результати моделювання показали, що таке суспільство доволі легко може схилитися до певної думки під дією незначного ідеологічного поштовху.

Моделювання суспільства без сучасних засобів комунікації та з однією політичною силою зосередженою географічно далеко від центра суспільства показало, що така політична сила не в змозі донести свої погляди до всього суспільства, яке, в

свою чергу, утворює осередки однодумців. У випадку додання можливості віддаленого спілкування та ЗМІ ситуація кардинально змінюється: розглядувана партія може доносити свої ідеї до всього суспільства, яке згодом приходить до консенсусу.

За допомогою моделі було досліджено ситуацію, коли на суспільство впливає лише одна політична партія. Отримано результати, які показують як буде розвиватися ситуація в залежності від поточних характеристик партії. Також було знайдено, що існує певна межа (близько 3%) кількості прихильників партії, нижче якої така партія не може суттєво впливати на суспільство. Окрім того, моделювання показало, що ідеологічна складова є більш важливою для розвитку партії, ніж її чисельна складова.

Також була змодельована ситуація, коли у суспільстві вже сформована певна думка щодо вибраного питання та присутня політична партія з протилежними, до суспільних, поглядами. Було показано, що існує певна межа кількості партійців, при якій партії вдається переконати суспільство.

Моделювання суспільства з двома політичними партіями протилежних поглядів показало, що якщо ці сили рівномірно розподілені по території суспільства, то таке суспільство завжди приходить до консенсусу. Якщо ж кожна з політичних сил домінує на певній території суспільства, а також політичні сили доволі впевнені у своїй позиції, то таке суспільство поляризується ще більше. Потім, за умови відсутності розпаду, суспільство починає поступово консолідуватися.

За допомогою отриманої моделі можливо проводити моделювання і інших, не описаних у цій статті, ситуацій.

Список використаної літератури:

1. Герасіна, Л. М. Консенсус як стратегія примирення в політико-правових конфліктах / Л. М. Герасіна // Український соціум – 2006 – вип. 6(17) – с. 17-24. Режим доступу: <https://ukr-socium.org.ua/uk/archive/no-6-17-2006/konsensus-jak-strategija-primirennja-v-politiko-pravovih-konfliktah/>
2. Тупиця, О. Л. Політичний консенсус як чинник сталого розвитку сучасного демократичного суспільства / О. Л. Тупиця, К. К. Хабарева // Науково-теоретичний альманах «Грані» – 2018 – Т.21 – No 10 – с. 121-127. Режим доступу: http://nbuv.gov.ua/UJRN/Grani_2018_21_10_16
3. Агаев, Р. П. Об области сходимости дифференциальной модели достижения консенсуса / Р. П. Агаев // Управление большими системами. Вып. 36. М.: ИПУ РАН – 2012 – с.81-92.
4. Rainer, H., Andreas, F. (1998). Understanding Complex Social Dynamics: A Plea For Cellular Automata Based Modelling *Journal of Artificial Societies and Social Simulation* vol. 1, no. 3. Режим доступу: <http://jasss.soc.surrey.ac.uk/1/3/1.html>
5. Monica, S., Bergenti, F. (2014). A Stochastic Model of Self-Stabilizing Cellular Automata for Consensus Formation. 1260.
6. Nowak A., Lewenstein M. (1996). Modeling Social Change with Cellular Automata. In: Hegselmann R., Mueller U., Troitzsch K.G. (eds) Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View. Theory and Decision Library (Series A: Philosophy and Methodology of the Social Sciences), vol 23. Springer, Dordrecht. https://doi.org/10.1007/978-94-015-8686-3_14
7. Zheng, Y., Zhu, Y., Wang L. (2011). Consensus of heterogeneous multi-agent systems *IET Control Theory Appl.* Vol. 5, Iss. 16, pp. 1881-1888.
8. Suo, S., & Chen, Yu. (2008). The Dynamics of Public Opinion in Complex Networks *Journal of Artificial Societies and Social Simulation* vol. 11, no. 4 2. Режим доступу: <http://jasss.soc.surrey.ac.uk/11/4/2.html>
9. Wen, G., Duan, Z, Yu, W. (2012). Guanrong Chen Consensus in multi-agent systems with communication constraints *International Journal of Robust and Nonlinear Control* vol. 22, pp. 170-182.
10. Агаев Р.П Модели латентного консенсуса / Р.П. Агаев, П.Ю. Чеботарев // Автомат. и телемех. – 2017 – вып. 1, с. 106-120. Режим доступа: <https://publications.hse.ru/en/articles/226537418>
11. Yang, T., Meng, Z., Dimarogonas, D. V., Johansson, K. H. (2014). Global consensus for discrete-time multi-agent systems with input saturation constraints *Automatica* issue 50, pp. 499-506.
12. Моргун М. Г. Моделювання динаміки громадської думки у сучасному суспільстві / Моргун М. Г. // Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління», 2020, випуск 47, с. 64-75. Режим доступу: <https://periodicals.karazin.ua/mia/article/view/16272>.

References:

1. Herasina, L. M. (2006). Consensus as a strategy of reconciliation in political and legal conflicts. *Ukrainskyi sotsium - Ukrainian society* vol. 6(17), pp. 17-24 [in Ukrainian]. Retrieved from: <https://ukr-socium.org.ua/uk/archive/no-6-17-2006/konsensus-jak-strategija-primirennja-v-politiko-pravovih-konfliktah/>
2. Tupytsia, O. L., Khabarieva, K. K. (2018). Political consensus as a factor in the sustainable development of modern democratic society. *Naukovo-teoretychnyi almanakh «Hrani» - Scientific and theoretical almanac "Hrani"* vol. 21, No 10, pp. 121-127 [in Ukrainian]. Retrieved from: http://nbuv.gov.ua/UJRN/Grani_2018_21_10_16
3. Ahaev, R. P. (2012). On the area of convergence of the differential model of consensus. *Upravlenye bolshymy systemamy - Management of large systems* vol. 36. – 2012 – pp.81-92. [in Russian]
4. Hegselmann, Rainer & Flache, Andreas (1998). Understanding Complex Social Dynamics: A Plea For Cellular Automata Based Modelling *Journal of Artificial Societies and Social Simulation* vol. 1, no. 3. Retrieved from: <http://jasss.soc.surrey.ac.uk/1/3/1.html>
5. Monica, Stefania & Bergenti, Federico. (2014). A Stochastic Model of Self-Stabilizing Cellular Automata for Consensus Formation. 1260.
6. Nowak, A., Lewenstein, M. (1996). Modeling Social Change with Cellular Automata. In: Hegselmann R., Mueller U., Troitzsch K.G. (eds) *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View. Theory and Decision Library (Series A: Philosophy and Methodology of the Social Sciences)*, vol 23. Springer, Dordrecht. https://doi.org/10.1007/978-94-015-8686-3_14
7. Zheng, Y., Zhu, Y., Wang, L. (2011). Consensus of heterogeneous multi-agent systems *IET Control Theory Appl.* Vol. 5, Iss. 16, pp. 1881-1888.
8. Shuguang, Suo and Yu, Chen (2008). The Dynamics of Public Opinion in Complex Networks *Journal of Artificial Societies and Social Simulation* vol. 11, no. 42. Retrieved from: <http://jasss.soc.surrey.ac.uk/11/4/2.html>
9. Guanghui, Wen & Zhisheng, Duan & Wenwu, Yu (2012) Guanrong Chen Consensus in multi-agent systems with communication constraints *International Journal of Robust and Nonlinear Control* vol. 22, pp. 170–182
10. Ahaev, R. P., Chebotarev, P. Iu. (2017). Latent Consensus Models. *Avtomatyzatsiya y telemekhanika - Automation and telemechanics* vol. 1, pp. 106-120 [in Russian] Retrieved from: <https://publications.hse.ru/en/articles/226537418>
11. Yang, Tao & Ziyang, Meng & Dimarogonas, Dimos & Johansson, Karl (2014). Global consensus for discrete-time multi-agent systems with input saturation constraints *Automatica* issue 50, pp. 499-506.
12. Morhun, M. G. (2020). Model of public opinion dynamics in modern society. *Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated control systems»*, vol. 46, pp.64-75 [in Ukrainian]. Retrieved from: <https://periodicals.karazin.ua/mia/article/view/16272>

MORHUN Mykola,

PhD Student, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

ANALYZING DYNAMICS OF PUBLIC OPINION IN MODERN SOCIETY USING MULTI AGENT MODEL

Summary. Introduction. Nowadays, the problem of public opinion dynamic in a society has a great scientific interest. There are different types of models and approaches on how to solve the task. In this article the author improves his previously built model and does some situations analysis using created model.

Purpose. The aim of this article is to improve model of public opinion dynamics in modern society [12] and analyze public opinion dynamics under different conditions using the model.

Results. The model of public opinion dynamics in modern society was improved. Opinion exchange mechanism was refined. Opinion fluctuations for model agents were added, which made the model closer to real world.

By using the model, different scenarios within society were simulated. A simulation of a society without political forces showed that such a society could be inclined towards some opinion by a small impetus.

A society without modern communication technologies and mass media was simulated. The society had one political party that was located in a single region. Simulation results showed that such party cannot spread its ideology over whole society, but only to the nearest regions. However, in other regions like-minded groups of individuals started to arise. In the end of the simulation, the society was

divided by several groups. If repeat the simulation with modern communication technologies and mass media enabled, then the result will significantly differ: the party can spread its ideas over whole society and the society finds a consensus.

A case with a single political party which spread over whole society was analyzed with help of the model. It shows how resulting opinion in the society depends on the initial party parameters (number of party members and their initial opinion). The simulation showed that, in general, improving the party ideology is more important than increasing number of party members. However, there is a threshold of party members (near 3% of the society members) below which the party cannot significantly influence the society.

The situation when society already had some opinion and a political party with opposite opinion tried to change society mind was considered. It was shown, that a threshold of party members number which lets the party reach its goal exists.

Numerical experiments with two opposite political parties in a society showed, that if both parties evenly spread over the society, then such society always finds a consensus. But if each party dominates on own territory, then confrontation in the society grows. At some point it reaches its maximum and then, if the society didn't split in two, the society slowly goes towards consolidation.

Finally, obtained model can also be used for different from described in this article scenarios.

Keywords: model, society, public opinion, consensus.

Одержано редакцією 16.03.2020 р.

Прийнято до публікації 18.05.2020 р.

Уточнення до списку літератури (п.12) 24.06.2020 р.

УДК 519.688

DOI 10.31651/2076-5886-2020-1-44-60

PACS 02.70.-c

ПОРУБЛЬОВ Ілля Миколайович,
старший викладач, Черкаський
національний університет імені Богдана
Хмельницького
e-mail: ilya@vu.cdu.edu.ua
ORCID: 0000-0001-7369-3862

ДЕЯКІ АЛГОРИТМИ ПОШУКУ ЧИСЕЛ З МАКСИМАЛЬНОЮ КІЛЬКІСТЮ ДІЛЬНИКІВ

Розглянуто кілька варіантів постановки та способів розв'язування задачі пошуку чисел, що мають найбільшу кількість дільників серед чисел вказаного проміжку. Аргументовано, чому задачу для проміжку від M до N слід вважати складнішою, ніж для проміжку від 1 до N . Розглянуто узагальнення, коли максимальна кількість дільників шукається не серед усіх чисел проміжку, а лише серед не кратних деякому K . Побудовано кілька варіантів алгоритму розв'язання кількох варіантів цієї задачі, на основі істотних оптимізацій рекурсивного перебору. Наведено порівняльний аналіз швидкодії роботи розглянутих варіантів алгоритму, оцінено переваги та недоліки кожного з варіантів та наведено рекомендації про доцільність застосування розглянутих варіантів при роботі з великими числами.

Ключові слова: кількість дільників, надскладені числа, рекурсивний перебір та його оптимізації.

Вступ

Числа з великою кількістю дільників можуть мати властивості, які, зокрема, істотно впливають на час роботи деяких алгоритмів. Тому і фактичне знаходження таких чисел, і оцінка можливої кількості дільників є цікавими та мають значення.

Оскільки у світі існує дві різні трактовки натурального числа (за однією з них 0 не є натуральним числом, за іншою – є), зафіксуємо, що в цій статті дотримуємося першого варіанту (0 не є натуральним).

Означення 1. Будемо позначати кількість натуральних дільників числа n як $\tau(n)$. (Це позначення, введене, зокрема, в [4] та багатьох інших джерелах. На жаль, воно не є цілком загальноприйнятим; як стверджують [1, сторінка oeis.org/A000005] та [3], це саме можуть позначати також $d(n)$ чи $\sigma_0(n)$).

Наприклад, $\tau(18) = 6$, і цими шістьма числами є 1, 2, 3, 6, 9, 18.) \square

Означення 2. Будемо називати число n *надскладеним*, якщо кількість його дільників строго перевищує кількість дільників будь-якого меншого числа. Інакше кажучи, n надскладене, коли

$$\forall k_{1 \leq k < n} (\tau(k) < \tau(n)). \quad (1)$$

Зокрема, сайт [1] описує послідовність A002182, якій дається назва «highly composite numbers» та означення «такі числа n , що кількість дільників (n) стає рекордно великою» ([1, сторінка oeis.org/A002182]). Автору не вдалося знайти офіційного перекладу «highly composite numbers» українською та вдалося знайти російський переклад «сверхсоставные числа» ([2]), тому вжито переклад «надскладені числа». Першими десятьма надскладеними числами є: 1, 2, 4, 6, 12, 24, 36, 48, 60, 120. \square

У найстандартнішому вигляді ця задача в значній мірі вже розв'язана. Зокрема, вже згадана сторінка oeis.org/A002182 сайту [1] містить посилання на список точних значень перших 10000 надскладених чисел (10000-е число перевищує 10^{481} , чого для більшості практичних застосувань достатньо). Крім того, [3] стверджує, що у [6] теоретично доведена оцінка, що кількість дільників n асимптотично зростає досить повільно, $\forall \varepsilon_{>0} (\tau(n) \in o(n^\varepsilon))$.

Але відомі теоретичні викладки не є настільки точними, щоб отримувати з них конкретні значення надскладених чисел та кількостей їхніх дільників, а дані про конкретну послідовність A002182 не містять ні добрих описів алгоритмів, якими можна їх отримувати, ні можливостей узагальнень на інші схожі задачі. (Алгоритми, наведені безпосередньо на згаданій сторінці oeis.org/A002182 сайту [1] не є досить ефективними, бо передбачають перебір усіх чисел та знаходження кількості дільників кожного; ними нереально отримати більше, ніж перших 50–200 (залежно від швидкодії комп'ютера й того, скільки часу згодні чекати на результат) членів послідовності.)

Мета статті

Описати розроблені алгоритми для розв'язання задачі пошуку чисел з великою кількістю дільників, у таких варіантах постановки:

1. Дано натуральне число N . Для проміжку всіх натуральних чисел від 1 до N (обидві межі включно), знайти число, що має максимальну кількість дільників.

2. Дані натуральні числа M, N ($M \leq N$). Для проміжку всіх натуральних чисел від M до N (обидві межі включно), знайти число, що має максимальну серед чисел проміжку кількість дільників.

3. Дані натуральні числа N та K (K значно менше N , $K \geq 2$). Для проміжку всіх натуральних чисел від 1 до N (обидві межі включно), але ігноруючи числа, кратні K , знайти число, що має максимальну кількість дільників

- а) якщо гарантовано, що число K просте;
- б) якщо такої гарантії нема.

В усіх варіантах постановки мається на увазі, що знайти треба і максимальну серед кількостей дільників усіх чисел проміжку, і те число з проміжку, яке має цю максимальну кількість дільників.

Виклад основного матеріалу

Твердження 1. Нехай відоме канонічне розкладення числа N на прості множники

$$N = p_1^{k_1} \cdot p_2^{k_2} \cdot \dots \cdot p_m^{k_m} \quad (2)$$

(всі p_i прості та різні, всі k_i натуральні). Тоді кількість різних дільників N дорівнює

$$\tau(p_1^{k_1} \cdot p_2^{k_2} \cdot \dots \cdot p_m^{k_m}) = (k_1 + 1) \cdot (k_2 + 1) \cdot \dots \cdot (k_m + 1). \quad (3)$$

(Доведення цього твердження пропустимо як загальновідоме; в разі потреби його можна знайти, наприклад, у [5, § 9].) \square

Наслідок твердження 1. Якщо числа a та b взаємно прості, то $\tau(a \cdot b) = \tau(a) \cdot \tau(b)$; інакше, $\tau(a \cdot b) < \tau(a) \cdot \tau(b)$.

Доведення. Розглянемо довільне просте число p_i . Позначимо показник цього простого в розкладенні a як $k_{i,a}$, в розкладенні b як $k_{i,b}$. Тоді це просте привносить у $\tau(a) \cdot \tau(b)$ множники $(k_{i,a} + 1)$ та $(k_{i,b} + 1)$, а в $\tau(a \cdot b)$ один множник $(k_{i,a} + k_{i,b} + 1)$. Оскільки $(k_{i,a} + 1) \cdot (k_{i,b} + 1) = k_{i,a} \cdot k_{i,b} + k_{i,a} + k_{i,b} + 1 = (k_{i,a} \cdot k_{i,b}) + (k_{i,a} + k_{i,b} + 1)$, а $k_{i,a}$ та $k_{i,b}$ невід'ємні, то можуть бути два випадки. Якщо хоча б один з показників $k_{i,a}$ чи $k_{i,b}$ є нулем, то $k_{i,a} \cdot k_{i,b} = 0$ і внесок у $\tau(a) \cdot \tau(b)$ та $\tau(a \cdot b)$ виявляється однаковим. Якщо ж обидва $k_{i,a}$ та $k_{i,b}$ додатні, то внесок у $\tau(a) \cdot \tau(b)$ виявляється більшим, чим у $\tau(a \cdot b)$, на $k_{i,a} \cdot k_{i,b}$. Лишилося згадати, що всі множники у правій частині (3) додатні, а взаємно прості й не взаємно прості саме тим і відрізняються, що у взаємно простих (і лише в них) нема спільних додатних показників степеню при тому ж простому, й доведення завершено. \blacksquare

Надалі для цілей цієї статті зручніше трактувати розкладення трохи інакше, залишивши вимогу «всі p_i прості та різні», але дозволивши не лише натуральні, а й нульові степені, як-то $98 = 2^1 \cdot 3^0 \cdot 5^0 \cdot 7^2$. Також, для цілей цієї статті зручніше вимагати, щоб розкладення на прості множники завжди містило всі прості числа $p_0 = 2, p_1 = 3, p_2 = 5, p_3 = 7, p_4 = 11, \dots$ в порядку зростання аж до останнього реально задіяного в розкладенні (якщо якогось з цих p_i нема, вважати, що є, але з показником $k_i = 0$; нумерація з 0 не принципова, вводиться лише заради більшої схожості з більшістю сучасних мов програмування). При дотриманні всіх цих вимог, можна вважати, що розкладення однозначно задається самою лише послідовністю (k_0, k_1, \dots) . Починаючи з цього місця, скрізь у статті (де не вказано інше) мається на увазі саме таке трактування. Така зміна трактування не впливає на правильність твердження 1, бо при $k_i = 0$ права частина (3) домножується на $(0 + 1) = 1$, що не впливає на результат.

Твердження 2. Для всіх надскладених (згідно означення 2) чисел, $k_0 \geq k_1 \geq \dots$

Доведення. Припустимо, ніби в розкладенні є хоча б одна пара така, що $a < b$, але $k_a < k_b$. Згідно нашого трактування розкладення, $a < b$ означає, що $p_a < p_b$. Тоді можна, залишивши решту степенів простих (якщо такі в цьому розкладенні є) незмінними, замінити $p_a^{k_a} \cdot p_b^{k_b}$ на $p_a^{k_b} \cdot p_b^{k_a}$. Це зменшить добуток (число, розкладення якого розглядаємо), тому що $k_b - k_a > 0$ множників зменшаться з p_b до p_a , при тому, що всі множники залишаться додатними. Водночас, кількість дільників лишиться тією самою, бо в (3) множники тільки обмінюються місцями, що не впливає на значення добутку. Отже, початкове число не є надскладеним, бо ми побудували менше число з такою ж кількістю дільників. Тобто, маємо протиріччя, джерело якого — у припущенні, ніби в розкладенні надскладеного числа може порушитися вимога $k_0 \geq k_1 \geq \dots$ \blacksquare

Твердження 2 задає односторонню імплікацію, а не рівносильність. Наприклад, для $192 = 2^6 \cdot 3^1$ вимога $k_0 \geq k_1 \geq \dots$ виконується ($k_0 = 6 \geq 1 = k_1$, всі подальші $k_i = 0$), але 192 не надскладене: має $(6 + 1) \cdot (1 + 1) = 14$ дільників, тоді як менше $120 = 2^3 \cdot 3^1 \cdot 5^1$ має їх більше, а саме $(3 + 1) \cdot (1 + 1) \cdot (1 + 1) = 16$.

Наслідок 1 твердження 2. Якщо задачу слід розв'язати на проміжку від 1 до N , то оцінити згори, які прості множники мають шанс бути включеними у надскладені, а які точно не будуть, можна так: перебрати $\prod_{i=0}^k p_i$ (тобто, послідовно формувати добутки 2, потім $2 \cdot 3$, потім $2 \cdot 3 \cdot 5$, потім $2 \cdot 3 \cdot 5 \cdot 7$, ...), доки добуток не перевищить N ; повернутися до попереднього (останнього, при якому ще не перевищив). Прості з більшими номерами не зустрічатимуться у надскладених, бо якщо те просте з більшим номером входить у розкладення з деяким показником $k_i > 0$, то з не меншими показниками повинні входити також усі попередні прості, й добуток перевищить N . ■

Визначена таким способом кількість дещо завищена, бо не враховує, що фактично в розкладеннях більшості надскладених малі прості мають більші, чим одиниця, показники, й перевищення N стається на ще меншому простому. Але це правильна оцінка згори. Адекватно подати цю оцінку аналітично не здається можливим, бо нема простого аналітичного способу задати залежність величини простого числа від його номера. Грубо (додатково завищивши) це можна оцінити як «кожне просте $p_i \geq 2$, тож, щоб добуток був у межах N , кількість простих повинна бути не більшою $\log_2 N$ ».

Наслідок 2 твердження 2. Якщо бажаємо знайти надскладені, не маючи способу відразу лише їх і формувати (й тому потребуючи перебирати різні числа, щоб вибрати серед них потрібні), їх можна шукати не серед взагалі всіх чисел, а лише серед тих, для розкладень яких виконується $k_0 \geq k_1 \geq \dots$. ■

Раз ці твердження та наслідки стосуються надскладених чисел, вони можуть бути використані при розробці алгоритму вирішення задачі в постановці № 1. Також їх легко модифікувати під постановку № 3а (проміжок від 1 до N , заборонене K просте): досить викреслити K з простих, і це призведе до потрібного у цій постановці ігнорування всіх чисел, кратних K , і лише їх.

Для постановки № 3б безпосередньо вони не є правильними. Наприклад, для проміжку від 1 до 42, $K = 6$, найбільшу кількість дільників $((3 + 1) \cdot (1 + 1) = 8)$ має число $40 = 2^3 \cdot 3^0 \cdot 5^1$, для якого $k_1 = 0 < 1 = k_2$. Але далі будуть доведені твердження 4, 5 та 7, які можна вважати частковими приблизними аналогами твердження 2.

Для постановки № 2 воно теж неправильне (наприклад, на проміжку від 49 до 51 найбільшу кількість дільників $((1 + 1) \cdot (2 + 1) = 6)$ має $50 = 2^1 \cdot 3^0 \cdot 5^2$, де $k_0 = 1 < 2 = k_2$ та $k_1 = 0 < 2 = k_2$), і не вдалося знайти ніякого аналога цьому твердженню.

Загальний опис та аналіз алгоритму для постановки № 1. Можна організувати перебір рекурсивно, з такими аргументами рекурсії:

- `primeIdx` – для якого простого числа підбираємо степінь на поточному рівні рекурсії, якщо прості записані у масиві `primes`, а `primeIdx` є індексом (набуває значень 0, 1, 2, ..., внаслідок чого `primes[primeIdx]` набуває значень 2, 3, 5, ...).
- `maxDeg` – максимальний дозволений показник степені; потрібен, щоб підтримувати монотонність послідовності показників.
- `prodBefore` – який добуток усього, вже вибраного на зовнішніх рівнях рекурсії (якщо, наприклад, перебуваємо на рівні `primeIdx = 2` і на попередніх рівнях було вибрано, що $p_0 = 2$ береться у степені 4, а $p_1 = 3$ у степені 2, то повинно бути `prodBefore = 24 · 32 = 144`); потрібне, щоб запам'ятати один з результатів (яке число має максимальну кількість дільників).
- `divsBefore` – скільки дільників має `prodBefore` (у тому ж прикладі «2⁴, 3²» повинно бути $(4 + 1) \times (2 + 1) = 15$); теоретично `divsBefore` можна встановити на основі `prodBefore`, але практично для цього треба було б розкласти `prodBefore`; значно

швидше мати окремий параметр, де ця кількість готова; саме з величин `divsBefore` обирається результат-максимум.

Теоретично, аргументів `primeIdx`, `maxDeg`, `prodBefore` та `divsBefore` досить. Обмежувати перебір лише числами, меншими N , можна умовою `prodBefore <= N`. Але якщо N близьке до межі типу, дія `prodBefore * primes[primeIdx]` може давати переповнення, внаслідок чого або програма завершуватиметься аварійно, або умова `prodBefore <= N` даватиме неправильний результат. Цих проблем можна уникнути, якщо ввести ще один аргумент:

- `leftOfN` — «те, що лишилось від N »; при початковому нерекурсивному виклику надається значення N зі вхідних даних, у ході рекурсії `leftOfN` ділиться (цілочисельно) на `primes[primeIdx]` (скажімо, у тому ж прикладі « $2^4, 3^2$ », `leftOfN` при `primeIdx = 2` повинно дорівнювати $N \text{ div } 144$; якби не було переповнень, то завжди виконувалася б рівність `leftOfN = N div prodBefore`). Тоді умовою продовження рекурсії (заглиблення у рекурсію) можна взяти `leftOfN > 0`.

Тоді саму рекурсію можна організувати, як у лістингу 1. У цій реалізації змінні `globalAnsDivs` (максимальна кількість дільників) та `globalAnsVal` (число з проміжку, на якому вона досягається) глобальні. Початковий нерекурсивний виклик цієї функції може мати, скажімо, вигляд `recl(0, 100, N, 1, 1)`: починаємо з 0-го простого `primes[0] = 2`, дозволяємо йому великий показник степеню 100, шукаємо для проміжку до N , поточний добуток та поточну кількість дільників ще не накопичено (і там, і там 1, а не 0, бо це порожні добутки, а не порожні суми).

```
typedef long long LL; // Суто для скорочення назви типу, непринципово.

void recl(int primeIdx, int maxDeg, LL leftOfN, LL prodBefore, LL divsBefore) {
    if(divsBefore > globalAnsDivs) { // Якщо поточний результат кращий за досі відомий,
        globalAnsDivs = divsBefore; globalAnsVal = prodBefore; // то оновити.
    }
    for(int deg = 1; deg <= maxDeg; deg++) { // Поточне просте можна взяти
        //не менше 1 разу (0 разів вже враховано) і не більше разів, ніж попереднє просте.
        prodBefore *= primes[primeIdx]; // Добуток збільшується, а leftOfN зменшується
        leftOfN /= primes[primeIdx]; // відповідно до поточного простого primes[primeIdx].
        if(leftOfN == 0) return; // За межами потрібного проміжку, виходимо.
        else recl(primeIdx + 1, deg, prodBefore, divsBefore * (deg+1));
        // Якщо продовжуємо, то: розглядатимемо наступне просте №(primeIdx + 1);
        // обмеження на показник його степеню – кількість цього простого
        // (потрібно для монотонності, див. наслідок 2 твердження 2);
        // prodBefore вже домножений, leftOfN вже зменшений,
        // а divsBefore ще не змінений, тому домножаємо у виклику.
    }
}
```

Лістинг 1 – організація рекурсивного перебору в першому варіанті алгоритму.

Для цього рекурсивного перебору (як і для багатьох інших) важко сформулювати адекватну асимптотичну оцінку часу роботи. Можна сказати, що раз $k_0 \geq k_1 \geq \dots$, то перебираються монотонно не зростаючі послідовності довжиною m_{\max} (яку можна визначити за наслідком 1 твердження 2), значення яких є цілими числами від 0 до максимально можливого k_0 (позначимо k_{\max}). Як відомо, кількість таких послідовностей дорівнює кількості сполучень з повтореннями $\bar{C}_{k_{\max}+1}^{m_{\max}} = C_{k_{\max}+m_{\max}}^{m_{\max}}$ (можливих значень $k_{\max} + 1$, бо від 0 включно до k_{\max} включно; з них вибирається m_{\max} штук; переставляти місцями не можна, бо потрібно дотриматися монотонності; повтори можливі, бо нерівності у $k_0 \geq k_1 \geq \dots$ нестрогі). Але, як вже вказано у наслідку 1 твердження 2, є

проблеми навіть з тим, як аналітично виразити m_{\max} через N . Можна, звісно, узяти грубу верхню оцінку $m_{\max} \leq \log_2 N$, а також вивести аналогічну оцінку $k_{\max} \leq \log_2 N$ (якщо піднести просте $p_0 = 2$ до ще більшої степені, результат вийде більшим N). Але з такої грубої оцінки нема користі: вона створює хибне враження, ніби загальна кількість варіантів, які слід перебрати, співмірна з $C_{2\log_2 N}^{\log_2 N}$, що може бути оцінено нерівністю $\frac{N^2}{2^{\log_2 N + 1}} < C_{2\log_2 N}^{\log_2 N} < N^2$ (за наслідком з біному Ньютона, $C_p^0 + C_p^1 + \dots + C_p^p = 2^p$; візьмемо $p = 2 \cdot \log_2 N$ і врахуємо, що доданок $C_{2\log_2 N}^{\log_2 N}$ є максимальним, але лише одним з $2 \cdot \log_2 N + 1$ додатних доданків, які в сумі дають $2^{2\log_2 N} = N^2$).

Ця оцінка сильно завищена, бо в ній не враховано, що досить перебирати лише ті послідовності, де весь добуток (а не лише 2^{k_0}) не перевищує N . Наприклад, візьмемо $N = 10^9$ і застосуємо цю саму оцінку окремо для кожної можливої пари k_0, k_1 , де $2^{k_0} \cdot 3^{k_1} \leq 10^9$, рахуючи також окремо для кожної такої пари оцінку m_{\max} . Наприклад, при $k_0 = k_1 = 1$, максимальне значення послідовності $k_{\max} = 1$, а m_{\max} визначається тим, що $2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \approx 6,47 \cdot 10^9 > 10^9$. Отже, якщо не розглядати ні 2 та 3 (вони вже зафіксовані), ні 29 (добуток разом з ним перевищує 10^9), отримаємо $m_{\max} = 7$, $k_{\max} = 1$, $\bar{C}_{1+1}^7 = C_8^7 = 8$. Аналогічно, для $k_0 = 5, k_1 = 3$ максимальне значення подальших k_i визначається $k_1 = 3$, а m_{\max} тим, що $2^5 \cdot 3^3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \approx 1,397 \cdot 10^9 > 10^9$, тобто $m_{\max} = 5$, що дає $\bar{C}_{3+1}^5 = C_8^5 = 56$. Написавши для аналізу решти можливих поєднань k_0, k_1 програму (вона є очевидним узагальненням згаданих міркувань і тут не наводиться), отримаємо, що сумарна за всіма можливими парами k_0, k_1 кількість послідовностей становить 5373, що зовсім мало (як для комп'ютера) і значно менше $\frac{(10^9)^2}{2^{\log_2 10^9 + 1}} \approx 1,4 \cdot 10^{16}$.

Більш того, повторивши аналогічні обчислення при $N = 2^{64}$ та $N = 2^{128}$ (що перевищують можливі значення 64-бітового та 128-бітового беззнакових типів), отримаємо сумарно $\approx 6,19 \cdot 10^6$ та $\approx 8,04 \cdot 10^{11}$ послідовностей відповідно. Ці оцінки теж завищені (враховують значення $2^{k_0} \cdot 3^{k_1}$, але не значення $2^{k_0} \cdot 3^{k_1} \cdot 5^{k_2} \cdot 7^{k_3} \dots$, тому теж включають багато послідовностей, які не розглядатимуться, бо добуток перевищує N), тож можна стверджувати, що такий алгоритм досить ефективний при N у межах 64-бітового (виконання практично миттєве) і навіть 128-бітового (результатів реально дочекатися) типів. Прямий експеримент підтверджує це «з запасом»: було взято кілька значень порядку $10^{36} \dots 10^{38}$ (при $2^{128} \approx 3 \cdot 10^{38}$), і для них час виконання був до 0,1 с.

Модифікація алгоритму для постановки № 2. Якщо говорити лише про правильність та ігнорувати час роботи, можна обійтися невеликими змінами коду відносно лістингу 1: аргумент `maxDeg` стає непотрібним, бо вже не треба забезпечувати монотонність показників (див. зауваження після твердження 2); при оновленні відповіді (`globalAnsDivs = divsBefore`) слід перевірити, крім `divsBefore > globalAnswer`, також `prodBefore >= M`; показник степені `deg` слід перебирати, починаючи з 0, а не 1 (там само вже наведено приклад, коли відповідь досягається на числі $50 = 2^1 \cdot 3^0 \cdot 5^2$).

Але час роботи так модифікованого алгоритму збільшується кардинально. Головним чином (але не тільки) через можливість ситуації `deg=0`, коли всі, крім `primeIdx`, аргументи (зокрема й `leftOfN`, від якого залежить вихід з рекурсії) наступного рівня рекурсії дорівнюють відповідним аргументам попереднього рівня, тож заглиблень у рекурсію може бути дуже багато.

Це можна у значній мірі оптимізувати, використавши такі засоби.

По-перше, спробувати розв'язати задачу в постановці № 1; якщо відповідь потрапить у проміжок від M до N , то задача вирішена також і в постановці № 2. Звісно, так буде далеко не завжди.

По-друге, якщо $N - M$ досить мале, спробувати розв'язати задачу очевидним способом, тобто розкласти на прості множники кожне з чисел від M до N , знайти для кожного кількість дільників (згідно (3)) і вибрати максимум.

По-третє, ввести, «симетрично» до `leftOfN`, аргумент `leftOfM` («що лишилось від M »); якщо зменшувати нижню межу `leftOfM` діленням із заокругленням догори, наприклад, $\text{leftOfM} = (\text{leftOfM} + \text{primes}[\text{primeIdx}] - 1) / \text{primes}[\text{primeIdx}]$, то умову виходу з рекурсії можна змінити з «`leftOfN == 0`» на, наприклад, «`leftOfM > leftOfN`»; це може наставати помітно раніше, ніж «`leftOfN == 0`».

По-четверте, використати прийом відтинань гілок перебору за верхніми оцінками з методу гілок та меж (описаний в [7], [8, розд. 4.2], [9, розд. 11.3] та багатьох інших джерелах). Тобто, в кожному рекурсивному виклику будувати для поточних `prIdx`, `leftOfM`, `leftOfN`, `prodBefore`, `divsBefore` деяку евристичну оцінку вигляду «на вкладених рівнях рекурсії, набрана на зовнішніх рівнях рекурсії кількість дільників `divsBefore` не може збільшитися більш як у T разів». Тобто, потрібна додаткова функція, яка прийматиме як аргументи, наприклад, `prIdx`, `leftOfN` (чи деяку іншу підмножину параметрів рекурсії) і вертатиме деяке T , для якого виконується це твердження. Далі, якщо $\text{divsBefore} \cdot T \leq \text{globalDivs}$, то поточну гілку рекурсії можна обривати як безперспективну. Було розглянуто два варіанти такої функції. (А) Коли вже дійшли до індексу `prIdx` (кількості менших простих вже зафіксовані на зовнішніх рівнях рекурсії), в числах проміжку `leftOfM... leftOfN` може поміститися не більш як $q = \text{floor}(\log_p \text{leftOfN})$ нових простих, де $p = \text{primes}[\text{prIdx}]$, `floor` — заокруглення донизу (q не може бути більшим, ніж вказана кількість, бо добуток перевищив би `leftOfN`; уточнимо, що під цим q мається на увазі кількість, що враховує кратність, наприклад, $5^2 \cdot 7^1$ рахується як $2 + 1 = 3$). Враховуючи наслідок твердження 1, кількість дільників була б найбільша, якби ці не більш ніж q множників були різними простими; тоді кількість дільників була б $(1 + 1)^q = 2^q$. Таку оцінку відносно легко рахувати, вона є оцінкою згори (реальна кількість дільників справді не перевищує 2^q), але вона завищена: якщо хоча б частина простих різні, то хоча б частина з них більші за `primes[primeIdx]`, і їх, імовірно, поміститься менша кількість, ніж q , пораховане через логарифм. Тому є обґрунтовані сумніви, чи добре така оцінка відтинатиме безперспективні гілки. (Б) В якості функції оцінки можна використати алгоритм з лістингу 1 для постановки № 1; аргумент `leftOfM` при цьому слід просто ігнорувати, в якості `primeIdx` передавати значення `primeIdx` з поточного виклику рекурсії перебору для постановки № 2; для такого змішаного використання двох рекурсій доведеться окремо мати змінні `globalAnsDivs2` та `globalAnsVal2`, що стосуються перебору для постановки № 2, і окремо змінні `globalAnsDivs1` та `globalAnsVal1`, що стосуються «вкладеного» перебору для постановки № 1. Це не буде щоразу те саме повне розв'язання тієї самої задачі способом з лістингу 1, бо на зовнішніх рівнях рекурсії перебору для постановки № 2 могли розглядатися послідовності показників, які були б відкинуті алгоритмом з лістингу 1, але можуть бути корисними для постановки № 2. Це даватиме верхню оцінку, бо якщо для всіх можливих продовжень не вдалося знайти кращого, ніж вже відомий (`globalAnsDivs2`) розв'язок, то його тим паче нема серед лише тієї частини можливих продовжень, які враховують межу `leftOfM`. Крім того, аналогічно оптимізації «по-перше» цього переліку, можлива ситуація, коли шукали найкращий серед усіх способів, ігноруючи `leftOfM`, а виявилось, що найкращий серед усіх задовольняє це обмеження; тоді цей спосіб тим паче найкращий серед тих, на які накладене це обмеження, тобто може знайтися не просто оцінка, а точна відповідь для відповідної гілки перебору. Однак, перебір з лістингу 1, хоч і швидкий як для перебору, все ж повільний для використання його в

кожному виклику рекурсії перебору для постановки № 2. Тому потрібен певний компроміс, коли варто використовувати такий повільний спосіб знаходження верхньої оцінки, а коли ні. Це питання поки що вирішене лише евристично, без чіткої аргументації.

По-п'яте, ідею верхніх оцінок та відкидання безперспективних варіантів можна використати не лише в рекурсивному переборі, а також і в пошуку кількості дільників поточного числа через розкладення. Традиційно (описано в [10, задачі 1.1.22–1.1.23] та багатьох інших джерелах), розкладення числа n передбачає перебір усіх потенційних дільників до \sqrt{n} (включно, якщо корінь цілий), із традиційною оптимізацією «якщо число n вже зменшилося від ділень на деякі менші дільники, то далі можна брати за межу корінь зі вже зменшеного n ». Однак, все це виходить із припущення «дане число n , треба за будь-яку ціну знайти точну кількість його дільників». Зараз ситуація дещо інша: коли видно, що кількість дільників числа n точно не буде більшою за максимальну відому кількість дільників `globalAnsDivs2` числа `globalAnsVal2`, то ні розкладення n , ні точне значення кількості дільників n вже не важливі, тож цикл перебору дільників можна обірвати. Для скорочення формул, позначимо `globalAnsDivs2 / divs` як d (у скільки разів має збільшитися поточна кількість дільників `divs`, щоб їх стало більше, ніж наразі максимальна кількість).

З міркувань, аналогічних поясненню верхньої оцінки (A) абзацу «по-четверте», і вживаючи q в тому ж смислі, можна сказати, що для кількості дільників має виконатися

$$2^q \geq d; \quad (4)$$

при цьому, щоб ці q дільників помістилися в n , для мінімального з простих дільників (позначимо його як p) має виконуватися

$$p^q \leq n. \quad (5)$$

Оскільки p , q та n додатні, а функція $f(x) = x^q$ при додатних x та додатному $q = \text{const}$ монотонна, нерівність (5) рівносильна

$$p \leq n^{1/q}. \quad (6)$$

Логарифмуючи (4) за основою 2, отримуємо

$$q \geq \log_2 d; \quad (7)$$

взявши обернену величину від обох частин (7) (знак при цьому змінюється на протилежний), отримуємо

$$\frac{1}{q} \leq \frac{1}{\log_2 d}. \quad (8)$$

Враховуючи $n \geq 1$, (6), (8) та монотонність показникової функції, отримуємо

$$p \leq n^{\frac{1}{\log_2 d}}. \quad (9)$$

А враховуючи, що традиційне обмеження «до кореня» теж лишається в силі, маємо

$$p \leq \min(\sqrt{n}, n^{\frac{1}{\log_2 d}}). \quad (10)$$

При цьому слід розуміти, що:

- 1) при досить великих d , (9) накладає на p жорсткіше обмеження, ніж корінь, але при $d < 4$ меншим є, навпаки, корінь; тому, (10) обмежує перебір краще, ніж (9);
- 2) коли при пошуку розкладення вдається знайти новий дільник, зменшується не лише n , а й d (за рахунок збільшення `divs` при сталому `globalAnsDivs2`), і значення $n^{\frac{1}{\log_2 d}}$ при цьому, як правило, збільшується; тому, при знаходженні нового

дільника поточного числа, важливо оновлювати верхню межу перебору потенційних дільників, щоб не втратити правильність (не загубити розв'язок).

По-шосте, «по-четверте» та «по-п'яте» істотно залежать від оцінок, дієвість яких істотно залежить від наразі відомого `globalAnsDivs2`, тому важливо переставити рекурсивні виклики з тими самими `leftOfM`, `leftOfN`, `prodBefore`, `divsBefore` (відповідні ситуації «поточне просте число береться в 0-му степені») на останнє місце.

По-сьоме, для якнайранішого заповнення `globalAnsDivs2` досить великою правильною оцінкою, варто, якщо не спрацювало «по-перше», то розв'язати задачу у постановці № 1 для проміжку «від 1 до $N - M + 1$ », знайти число з проміжку «від M до N », кратне `globalAnsVal1` проміжку «від 1 до $N - M + 1$ » (воно гарантовано існує й лише одне, дорівнює $(N \text{ div } \text{globalAnsVal1}) \cdot \text{globalAnsVal1}$) та знайти кількість його дільників за (3). При не дуже малих $N - M + 1$ слід очікувати, що це число має чимало дільників, бо воно кратне `globalAnsVal1`, яке є надскладеним.

По-восьме, ідею «замість рекурсії, розкласти на множники кожне з чисел діапазону й рахувати кількість дільників» можна використати не лише на початку (як у «по-друге»), а й всередині рекурсії. Це особливо корисно, коли проміжок між початковими M та N завеликий для повного перебору всіх його чисел, але після кількох ділень у рекурсивних викликах `leftOfM` та `leftOfN` стають досить близькими. Такі перевірки проміжку саме всередині рекурсії гарантують, що перевірятися будуть не всі підряд числа початкового проміжку від M до N , а лише ті з них, які кратні `prodBedore`, набраному на зовнішніх рівнях рекурсії; оскільки (завдяки «по-шосте») `prodBedore` майже завжди містить добуток невеликих простих, у таких чисел досить багато дільників. Щоправда, так можуть виникати непродуктивні витрати на перевірку одного й того ж числа в різних гілках рекурсії; але цього можна уникнути, запам'ятовуючи вже перевірені числа в одній глобальній множині (наприклад, у термінах C++ STL, `set<long long> alreadyLooked`).

Врахування всіх цих (і, насправді, ще деяких) ідей дало код, що має обсяг, який вже не варто включати у друковане видання; охочі можуть ознайомитися з ним за посиланням <https://ejudge.ckipo.edu.ua/maxDivsInRange.cpp>. Асимптотично оцінити час його роботи не здається можливим, бо на рекурсивний перебір накладено дуже багато різних засобів його скорочення. Але експериментально він показав себе значно кращим за очевидний підхід (пошук кількості дільників кожного числа окремо) чи за початкову версію з мінімальними відносно лістингу 1 змінами: час роботи цих алгоритмів перевищує десятки секунд вже для багатьох вхідних даних порядку $N \approx 10^7 \dots 10^8$, а розроблений алгоритм має менший час роботи при $N \approx 10^{18}$.

У таблиці 1 наведено результати дослідження часу роботи реалізації розробленого алгоритму мовою C++ (Windows, компілятор g++, рівень оптимізацій «-O2») на комп'ютері тактової частоти 3,2 ГГц. Було проведено 192 000 вимірювань (по 32 000 у кожній з 6 груп). N завжди вибиралося випадково (з рівномірним розподілом) з проміжку $10^{15} \leq N \leq 10^{18}$. Для вибору M було використано шість різних способів, описаних у написі при таблиці; саме такі проміжки були вибрані евристично, без чіткої аргументації. Рядки були вибрані з тих міркувань, щоб наголосити в першу чергу на тривалості роботи в поганих випадках, але середнім випадкам теж приділити певну увагу. Час роботи алгоритму для кожної окремо взятої пари M, N вимірювався процесорний (засобами `GetProcessTime`), усереднений з двох спроб. Початкова похибка вимірювань кожної спроби становила 1 тик $\approx 0,015$ с. Ці фактори спричинили велику кількість значень 0,015 с та 0,008 с (які фактично означають «1 тик» та «одна спроба менше 1 тіку, інша 1 тик»).

Таблиця 1 — експериментально вимірний час роботи розробленого алгоритму для розв’язування задачі в постановці № 2, при випадкових (з рівномірним розподілом) $10^{15} \leq N \leq 10^{18}$; стовпці (1)–(6) означають спосіб вибору M : (1) $10^9 \leq M < 0,75 \cdot N$; (2) $0,75 \cdot N \leq M < 0,99 \cdot N$; (3) $0,99 \cdot N \leq M < 0,999999 \cdot N$; (4) $0,999999 \cdot N \leq M < N - 10000$; (5) $N - 10000 \leq M < N$; (6) $M = N$; у способах (1)–(5) значення M вибиралося в указаному проміжку випадково з рівномірним розподілом. Всі значення часу подані в секундах з заокругленням до тисячних.

	(1)	(2)	(3)	(4)	(5)	(6)
Середнє арифметичне	0,009	0,010	0,030	0,413	0,022	0,231
Середнє квадратичне	0,010	0,011	0,038	0,464	0,034	0,668
Максимальне значення	0,015	0,046	0,561	1,661	0,897	4,602
10-е згори значення	0,015	0,039	0,463	1,396	0,507	4,579
100-е згори значення	0,015	0,031	0,179	1,162	0,226	4,274
320-е згори значення (верхній процентиль)	0,015	0,023	0,124	1,022	0,117	3,447
3200-е згори значення (верхній дециль)	0,015	0,015	0,046	0,702	0,031	0,639
8000-е згори значення (верхній кuartиль)	0,008	0,015	0,031	0,538	0,023	0,086
16000-е згори значення (медіана)	0,008	0,008	0,023	0,328	0,015	0,015
24000-е згори значення (нижній кuartиль)	0,008	0,008	0,015	0,257	0,015	0,008

Мінімальність усіх характеристик у стовпці (1) легко пояснити тим, що для широких діапазонів відповідь для постановки № 1 часто виявляється остаточною для постановки № 2. Те, що величини стовпця (4) або максимальні, або другі максимальні у своїх рядках, легко пояснити тим, що для таких розмірів проміжку не дуже добре працюють хоч оптимізації на базі алгоритму для постановки № 1 (для них проміжок завузкий; те, що потрапляє в нього, часто сильно відрізняється від максимального для проміжку від 1), хоч оптимізації на базі перебору від leftOfM до leftOfN (для них проміжок надто широкий). Особливість стовпця (6), де медіана та кuartилі досить малі, але найбільші в своїх рядках максимум, 10-е, 100-е значення та верхній процентиль, легко пояснити тим, що з одного боку $M = N$ означає, що досить розкласти всього одне число (тому середні, медіана та кuartилі досить малі), але іноді, якщо це число просте чи є добутком двох простих, близьких до кореня, для з’ясування цього слід виконати велику кількість ітерацій, і вони не відкидаються ніякими оцінками. Отже, результати експерименту відповідають очікуванням. Також вони свідчать про результативність ідей «по-п’яте» (скорочення перебору дільників на основі формули (10)) та «по-сьоме», раз аналіз проміжків у сотні чисел відбувається швидше, ніж одночислових, хоча матсподівання кількості простих (та добутків двох простих, близьких до кореня) на проміжку, очевидно, збільшується зі збільшенням довжини проміжку.

Модифікація алгоритму для постановки № 3а. Тут досить вчинити, як зазначено відразу після твердження 2, тобто робити всі ті самі дії, що у простому й ефективному варіанті цього алгоритму для постановки № 1, але викресливши K з послідовності простих. Час роботи при цьому може або зменшитися (за рахунок того, що замість самого K та всіх подальших простих використовуються більші наступні прості, й leftOfN зменшується швидше, від чого зменшується загальна кількість варіантів, що перебираються), або лишиться незмінним (навіть якщо вказані причини не спрацювали, ніщо не збільшує час роботи у порівнянні з постановкою № 1).

Немає принципів проблем, що істотно перешкоджали б поєднати постановки № 3а та № 2 (коли і проміжок від M до N , і розглядаються лише числа, не кратні простому K). Однак, через поєднання технічної складності з сумнівною значимістю результату, таке поєднання не досліджувалося детально. Очевидно, крім виключення K з переліку простих, слід також змінити перебори від $\text{leftOf}M$ до $\text{leftOf}N$ так, щоб відкидати кратні K . Але є й менш очевидні моменти; зокрема, доведеться якось змінити оптимізацію «по-сьоме», бо навіть якщо «globalAnsVal1 проміжку від 1 до $N - M + 1$ » знайдено, враховуючи відкидання кратних K , кратність K може (але не зобов'язана) з'явитися при обчисленні $(N \text{ div } \text{globalAnsVal1}) \cdot \text{globalAnsVal1}$. Це робить час роботи такої недодослідженої модифікації алгоритму ще менш передбачуваним.

Модифікація алгоритму для постановки № 3б. Покажемо, що для постановки № 3б не можна просто відкинути якісь із простих чисел. Розглянемо $K = 40 = 2^3 \cdot 5^1$ і побудуємо (очевидним неефективним алгоритмом, не вартим детальнішої згадки) послідовність «рекордних» за кількістю дільників чисел. (Аналогічно (1) з означення 2, але з умовою ігнорування чисел, кратних $K = 40$; формально це можна виразити як $\{n \mid (n \bmod 40 \neq 0) \wedge (\forall j_{(1 \leq j < n) \wedge (j \bmod 40 \neq 0)} (\tau(j) < \tau(n)))\}$ у порядку зростання.) Перші 16 таких чисел наведено у таблиці 2, де значення розподілені у групи за причиною, чому це число не кратне 40: чи тому, що не кратне $5 = 5^1$, чи тому, що не кратне $8 = 2^3$; числа, не кратні ні 5, ні 8 включені в обидві групи.

Таблиця 2 — перші 16 значень, «рекордних» за кількістю дільників серед чисел, не кратних 40.

не кратні 5	2	4	6	12	24	36	48		144	168		336			1008	
не кратні 8	2	4	6	12		36		60			180		420	900		1260

Бачимо, що для значень, «рекордних» серед не кратних 40, ця не кратність 40 забезпечується то не кратністю $5 = 5^1$, то не кратністю $8 = 2^3$, і нема очевидного закону, який виражав би, коли яка з цих умов має місце. Природно, щоб аналогічна ситуація траплялася і для деяких інших складених, в тому числі й тих, розкладення яких мають більше простих у ненульових степенях. Тобто, гарантовано відкинути якісь із простих, що входять у розкладення K в ненульових степенях, не здається можливим. Тому, сформулюємо інші знайдені обмеження, корисні для оптимізацій перебору (плануючи його загальну логіку схожою на перебір з лістингу 1, але зі значними відмінностями). Почнемо з простих, а все ж корисних означення 3 та тверджень 3 і 4.

Означення 3. Будемо позначати z_0, z_1, \dots показники степенів, з якими прості $p_0 = 2, p_1 = 3, p_2 = 5, \dots$ входять у розкладення K , кратність якому заборонена: $K = p_0^{z_0} \cdot p_1^{z_1} \cdot \dots$. Очевидно, значення z_0, z_1, \dots цілі невід'ємні (можуть дорівнювати 0). \square

Твердження 3. Якщо в деякому рекурсивному виклику prodBefore кратний K , всю поточну гілку можна обривати виходом з рекурсії (чи, що практично те само, не заходити туди, перевіряючи кратність не всередині виклику, а ззовні (перед)).

Доведення. Перебір організований так, що якщо prodBefore поточного виклику кратний K , він буде кратний і в усьому піддереві рекурсії; все це треба проігнорувати. \blacksquare

Твердження 4. Якщо рекурсивний перебір розглядає випадок $k_j < z_j$ (бере просте p_j у степені, меншій, ніж степінь його входження у K ; технічно це визначається тим, що локальна змінна deg на рівні рекурсії $\text{prIdx} = j$ набуває значення, менше z_j , і саме з ним відбувається рекурсивний виклик; все це можливо лише для тих простих, де $z_j > 0$, тобто $p_j \in$ дільником K), то на вкладених рівнях рекурсії можна застосовувати перебір з лістингу 1, призначений для постановки № 1.

Доведення. Ми плануємо організувати рекурсію для постановки № 3б так, щоб мати всі потрібні для перебору з лістингу 1 параметри `prIdx`, `leftOfN`, `prodBefore`, `divsBefore` — отже, маємо технічну можливість у одному з вузлів однієї рекурсії запустити іншу рекурсію (тобто, організувати перебір так, щоб з деяких вузлів дерева нової рекурсії далі продовжувалися піддерева старої рекурсії, а не гілки тієї (нової) рекурсії, яка привела до цього вузла).

Єдина причина, навіщо нова рекурсія має відрізнятися від лістингу 1 — треба перебирати лише числа, не кратні K . На всіх подальших рівнях розглядатимуться лише більші прості, й результат домноження `prodBefore` на них ніколи не стане кратним K , бо степінь p_j так і лишатиметься $p_j^{k_j}$, що менше $p_j^{z_j}$, потрібного для кратності K . В цих умовах, ніщо не заважає застосувати до степенів подальших (більших) простих доведення, цілком аналогічне доведенню твердження 2. ■

Примітка до твердження 4. Йдеться лише про монотонність подальших показників $k_{j+1} \geq k_{j+2} \geq \dots$ між собою; відношення ж між k_j та k_{j+1} може бути будь-яким ($k_j > k_{j+1}$, $k_j = k_{j+1}$ чи $k_j < k_{j+1}$). □

А як перебирати, коли ще не настав перехід, описаний у твердженні 4? Щоб визначити це, сформулюємо кілька тверджень-аналогів твердження 2, які теж дозволять обмежити перебір (гірше, ніж твердження 2, але хоч якось).

Твердження 5. Для розв'язування задачі у постановці № 3б досить перебирати числа, в розкладенні яких, для будь-яких простих p_a, p_b , таких, що $p_a < p_b$ та $z_a = 0$ (K не кратне p_a), завжди (незалежно від значення z_b) виконується $k_b \leq k_a$.

Доведення. Аналогічно доведенню твердження 2, якщо розкладення деякого числа містить $p_a^{k_a} \cdot p_b^{k_b}$, де $p_a < p_b$ та $k_a < k_b$, то заміна $p_a^{k_a} \cdot p_b^{k_b}$ на $p_a^{k_b} \cdot p_b^{k_a}$ зменшує значення, не змінюючи кількості дільників, і нема потреби розглядати більше (початкове) E , коли ту ж кількість дільників можна здобути й на меншому (отриманому цією заміною) E' . Але треба ще показати неможливість поєднання умов цього твердження із ситуацією $(E : K) \wedge (E' : K)$, тобто «отримане внаслідок заміни E' стає кратним K і його слід відкинути, хоча початкове E не кратне K ». А неможливо це тому, що вказана заміна збільшує показник лише степеню з основою p_a , для якої $z_a = 0$ (тобто, якій K не кратне), тож збільшення показника саме цього степеню не може створити кратності K ; зменшення якого б не було показника теж не може створити кратності. ■

Примітка до твердження 5. Умова $z_a = 0$ важлива, бо при $z_a > 0$ існують приклади, коли менше E' стає кратним K , хоча початкове E не було. Причому, це можливо як у випадку $z_b = 0$, так і $z_b > 0$. Наприклад, $E = 18 = 2^1 \cdot 3^2$ ($k_0 = 1, k_1 = 2$) не можна зменшувати до $E' = 12 = 2^2 \cdot 3^1$ ($k_0 = 2, k_1 = 1$) ні при $K = 4 = 2^2$ ($z_0 = 2, z_1 = 0$), ні при $K = 12 = 2^2 \cdot 3^1$ ($z_0 = 2, z_1 = 1$), бо в обох випадках заміна $2^1 \cdot 3^2$ на $2^2 \cdot 3^1$ призводить до того, що число стає кратним K і його слід відкинути. І (як можна встановити перебором), хоч при $K = 4$, хоч при $K = 12$, саме 18 з 6-ма дільниками є відповіддю для всіх N з проміжку $18 \leq N < 30$, тож його пропускати не можна. □

Твердження 6. Якщо розкладення K містить $p_a^{z_a} \cdot p_b^{z_b}$ при $p_a < p_b$ та $1 \leq z_a \leq z_b$, то $p_a^{z_a}$ можна відкинути (розв'язати задачу для $K' = K / p_a^{z_a}$), і результат буде тим самим.

Доведення. Очевидно, що кожне число потрапляє в одну з чотирьох груп: (А) не кратні ні $p_a^{z_a}$, ні $p_b^{z_b}$; (Б) кратні $p_a^{z_a}$, але не кратні $p_b^{z_b}$; (В) кратні $p_b^{z_b}$, але не кратні $p_a^{z_a}$; (Г) кратні і $p_a^{z_a}$, і $p_b^{z_b}$. Враховуючи, що p_a та p_b — різні прості, кратними $p_a^{z_a} \cdot p_b^{z_b}$ є всі числа з групи (Г) і лише вони. Розглянемо, чим відрізняються задачі в

постановці № 3б для початкового K і для зменшеного $K' = K / p_a^{z_a}$. Числа з групи (Г) і там, і там треба ігнорувати; числа з груп (А) та (Б) і там, і там не треба ігнорувати (саме серед них, одних чи інших, буде шукане). Отже, створити відмінність могли б лише числа з групи (В), які треба ігнорувати для зменшеного K' , але не треба для початкового K . Але належність числа групі (В) означає, що воно містить у розкладенні $p_a^{k_a} \cdot p_b^{k_b}$, де $k_a < z_a$ (це і є не кратністю $p_a^{z_a}$) та $k_b \geq z_b$ (це і є кратністю $p_b^{z_b}$). Оскільки за припущенням цього твердження $z_a \leq z_b$, маємо $k_a < z_a \leq z_b \leq k_b$, звідки $k_a < k_b$. Аналогічно доведенню твердження 2, вигідно замінити $p_a^{k_a} \cdot p_b^{k_b}$ на $p_a^{k_b} \cdot p_b^{k_a}$ (лишаючи степені інших простих, якщо такі є, незмінними): внаслідок заміни число стає меншим, маючи ту саму кількість дільників. Щоправда, тепер слід перевірити додаткову умову: раптом зменшене число слід відкинути, бо воно кратне початковому K ? Але це не так, враховуючи $k_a < z_b$ (слідє з вписаного на кілька рядків вище ланцюжка нерівностей). Тобто, не розгляд чисел з групи (В) не заважає правильності (для кожного з них існує інше число, яке буде розглянуте й дасть не гірший результат). ■

Наслідок 1 твердження 6. Якщо розкладення K містить багато простих і ситуація $p_a < p_b$ та $1 \leq z_a \leq z_b$ з'являється кілька разів для різних пар a, b , твердження 6 можна застосувати багатократно, і гарантовано прийти до однієї з двох ситуацій: або ненульовий показник лишиться у степені тільки одного простого, або послідовність показників z_j (рахуючи лише серед ненульових) стане строго спадною (лише ці випадки роблять поєднання $p_a < p_b$ та $1 \leq z_a \leq z_b$ неможливим ні для яких a, b). Наприклад, $K = 3^3 \cdot 5^2 \cdot 11^5$: прибрати 3^3 , бо $\in 11^5$; прибрати 5^2 , бо $\in 11^5$; лишиться 11^5 . Та, наприклад, $K = 2^5 \cdot 3^3 \cdot 5^4 \cdot 7^1 \cdot 13^1$: прибрати 3^3 , бо $\in 5^4$; прибрати 7^1 , бо $\in 13^1$; лишиться $2^5 \cdot 5^4 \cdot 13^1$. ■

Наслідок 2 твердження 6. Враховуючи наслідок 1 твердження 6, навіть не маючи конкретного K та його розкладення, а лише K_{\max} (таке, що розглядатимуться лише K з проміжку $2 \leq K < K_{\max}$), можливу кількість різних простих у розкладенні зменшеного K' можна оцінити згори так: послідовно будувати $2^1 = 2$, $2^2 \cdot 3^1 = 12$, $2^3 \cdot 3^2 \cdot 5^1 = 360$, $2^4 \cdot 3^3 \cdot 5^2 \cdot 7^1 = 75600$, ..., доки добуток не перевищить K_{\max} ; коли перевищить, повернутися на один крок назад. Адже, враховуючи наслідок 1 та монотонність (при додатних аргументах) добутку, степеневі та показникової функції, саме такими є мінімальні можливі добутки степенів 1-го, 2-х, 3-х, 4-х, ... простих, що задовольняють наслідку 1 твердження 6. Аналогічно наслідку 1 твердження 2, не здається можливим виразити це аналітично скільки-небудь точно, а, завищуючи оцінку, можна говорити про не більш як $\sqrt{2 \cdot \log_2(K_{\max})}$ штук (логарифм з міркувань, аналогічних згаданим у наслідку 1 твердження 2; корінь — з того, що j різних простих повинні входити у розкладення зі степенями $j, j-1, \dots, 1$, сума становить $j(j+1)/2$, тож зворотня дія потребує (приблизно) домноження на 2 й кореню). ■

Саме тому, що твердження 5 не працює при $z_a > 0$, важливо, що твердження 6 та його наслідки дозволяють побудувати на основі K рівноцінне йому K' , яке містить мало різних простих: це зменшує кількість можливих місць порушення твердження 5). Але корисність твердження 6 та його наслідків цим не обмежується: крім того, вони також дають можливість довести наступне твердження 7.

Твердження 7. Нехай перед початком основного перебору до K застосовані дії, описані в наслідку 1 твердження 6, й перебір відбувається для зменшеного K' . Нехай перебір організований з урахуванням твердження 4. Нехай у розкладення K' входять степені $p_a^{z_a} \cdot p_b^{z_b}$, де $p_a < p_b$, $z_a > 0$, $z_b > 0$. Нехай рівень рекурсії, що вибирає показник

степеню простого p_a , розглядає випадок $k_a \geq z_a$. Тоді діапазон перебору показників степеню простого p_b можна обмежити умовою $k_b \leq k_a$.

Доведення. Припустимо, ніби розглядаємо $k_b > k_a$. Оскільки маємо не довільне K , а K' , отримане за наслідком 1 твердження 6, то з $p_a < p_b$, $z_a > 0$, $z_b > 0$ випливає $z_a > z_b$. Отже, можна зібрати ланцюжок нерівностей $k_b > k_a \geq z_a > z_b$, з якого випливає $k_b > z_b$. Тобто, степінь простого p_b не створює некратності K' . Оскільки перебір організований з урахуванням твердження 4, потрібна некратність K' не могла з'явитися ще раніше (при ще меншому простому): якби вона з'явилася вже там, то показники степенів простих p_a та p_b підбиралися б уже перебором з лістингу 1, а не тим перебором, який враховує K . Значить, можливі два варіанти: або потрібна некратність K' надалі так і не утвориться, або утвориться завдяки тому, що є хоча б одне z_c , таке, що $c > b$ та $z_c > 0$, і для нього пізніше буде вибрано $k_c < z_c$. Якщо некратність K' так і не утвориться, нема потреби перебирати зовсім безперспективні варіанти. Якщо утвориться за рахунок вказаної причини, ніщо не заважає повторити прийом з доведення твердження 2, бо некратність K' і так забезпечується степенем $k_c < z_c$ третього (не p_a і не p_b) простого p_c , і заміна $p_a^{k_a} \cdot p_b^{k_b}$ на $p_a^{k_b} \cdot p_b^{k_a}$ на це ніяк не впливає. ■

Твердження 8. Підсумовуючи твердження 3—7, перебір для постановки № 36 можна і варто організувати так:

1. Розпочати з того, що перетворити задане у вхідних даних K згідно наслідку 1 твердження 6, й надалі працювати з K' . При цьому заодно формується масив $Z_$ (показники z_0, z_1, \dots згідно означення 3), який робиться глобальним, щоб могли легко доступатися до нього з рекурсії.

2. Мати дві окремі рекурсії, одна цілком відповідає лістингу 1 й використовується тоді, коли вже забезпечено некратність K , інша — коли ще не забезпечено.

3. Друга рекурсія має всі ті самі параметри, що перша, та параметр $\max\text{DegForK}$ (максимальний показник степені для тих простих, де $z_j > 0$; використовується там, де виконуються умови твердження 7), а також використовує масив $Z_$.

4. Друга рекурсія може:

а) Обривати свої гілки згідно твердження 3.

б) Продукувати замість власних викликів виклики першої рекурсії, згідно з твердженням 4; максимальний степінь deg при цьому дорівнює $Z_[\text{prIdx}] - 1$.

в) Продукувати власні виклики для тих простих, які входять у K' ($Z_[\text{prIdx}] > 0$), значення deg мають перебувати у проміжку від $Z_[\text{prIdx}]$ до $\min(\max\text{DegForK}, \max\text{Deg})$ (обидві межі включно), бо саме цей проміжок диктується твердженнями 5 та 7, якщо врахувати, що $\text{deg} < Z_[\text{prIdx}]$ підпадають під твердження 4.

г) Продукувати власні виклики для тих простих, які не входять у K' ($Z_[\text{prIdx}] = 0$), значення deg мають перебувати у проміжку від 0 до $\max\text{Deg}$ (обидві межі включно, але 0 означає припинення подальших заглиблень), бо це визначається твердженням 5.

Доведення, як такого, нема, бо воно зводиться до того, щоб перевірити, що наведені пункти покривають собою всі можливі ситуації. ■

Отриманий код завеликий для включення у друковане видання; охочі можуть ознайомитися з ним тут: <https://ejudge.ckipo.edu.ua/maxDivsExceptDivK.cpp>.

Оцінити асимптотично час виконання цього алгоритму не здається можливим (що природно, враховуючи, що він включає в себе алгоритм з лістингу 1, адекватно оцінити який теж не вдалося), тому знову наведемо експериментальні оцінки.

Результати дослідження тривалості роботи розробленого алгоритму наведені у таблиці 3. Було проведено 180 000 (по 30 000 у кожній з 6 груп) вимірювань часу роботи реалізації цього алгоритму. Мова, компілятор, комп'ютер, засоби вимірювання часу та статистичні характеристики ті самі, що в табл. 1.

Таблиця 3 – експериментально виміряний час роботи розробленого алгоритму для розв'язування задачі в постановці № 3б, при випадкових $10^{35} \leq N \leq 3 \cdot 10^{35}$; стовпці означають спосіб вибору K : (1) $2 \leq K \leq 50$; (2) $51 \leq K \leq 1000$; (3) $1001 \leq K \leq 2 \cdot 10^9$; (4) K мають вигляд $2^{m_0} \cdot 3^{m_1} \cdot 5^{m_2} \cdot 7^{m_3} \cdot 11^{m_4}$; (5) вигляд $2^{m_0} \cdot 3^{m_1} \cdot 5^{m_2} \cdot 7^{m_3} \cdot 11^{m_4} \cdot 13^{m_5} \cdot 17^{m_6}$; (6) вигляд $2^{m_0} \cdot 5^{m_2} \cdot 11^{m_4} \cdot 17^{m_6}$.

	(1)	(2)	(3)	(4)	(5)	(6)
Середнє арифметичне	0,038	0,061	0,041	0,094	0,077	0,080
Середнє квадратичне	0,041	0,071	0,042	0,111	0,089	0,099
Максимальне значення	0,140	0,226	0,234	1,248	1,404	1,263
10-е згори значення	0,125	0,203	0,172	0,390	0,250	0,639
100-е згори значення	0,094	0,179	0,070	0,288	0,218	0,304
300-е згори значення (верхній процентиль)	0,086	0,171	0,070	0,250	0,195	0,249
3000-е згори значення (верхній дециль)	0,055	0,117	0,054	0,171	0,133	0,164
7500-е згори значення (верхній кuartиль)	0,047	0,078	0,047	0,125	0,101	0,101
15000-е згори значення (медіана)	0,039	0,047	0,039	0,094	0,078	0,078
22500-е згори значення (нижній кuartиль)	0,031	0,039	0,039	0,039	0,039	0,023

Результати та висновки

Для кожного з трьох описаних у статті варіантів постановки задачі розроблено алгоритм, значно ефективніший за очевидні. Хоча для жодного з розроблених алгоритмів не вдалося побудувати адекватну асимптотичну оцінку, але швидкодія кожного з них досліджена експериментально. Для постановок № 1 та № 3а час не перевищував десятих секунди при $N \approx 10^{38}$; для постановки № 2, час не перевищував кількох секунд при $N \approx 10^{18}$ (детальні результати подано у таблицях, наведених у статті); для постановки № 3б час не перевищував кількох секунд при $N \approx 10^{35}$.

Список використаної літератури:

1. Сайт «The On-Line Encyclopedia of Integer Sequences». Режим доступу <https://oeis.org/A002182>
2. Стаття «Сверхсоставное число» в російськомовному розділі вікіпедії. Режим доступу https://ru.wikipedia.org/wiki/Сверхсоставное_число
3. Стаття «Divisor_function» в англomовному розділі вікіпедії. Режим доступу https://en.wikipedia.org/wiki/Divisor_function
4. Бухштаб А. А. Теория чисел. / А. А. Бухштаб – М.: Просвещение, 1966 – 392 с.
5. Дирихле П. Г. Л. Лекции по теории чисел. В обработке и с добавлениями Р. Дедекинда. / П. Г. Л Дирихле. – М.: Объединённое научно-техническое издательство НКТП СССР, 1936. – 404 с.
6. Apostol, Tom M. (1976), *Introduction to analytic number theory*, Undergraduate Texts in Mathematics, New York-Heidelberg: Springer-Verlag.
7. Little J. D. C. An algorithm for the traveling salesman problem. / Little J. D. C., Murty K. G., Sweeney D. W., Karel C. // *Operations Research* – Vol. 11 (1963) – P. 972-989.
8. Гудман, С. Введение в разработку и анализ алгоритмов. / С. Гудман, С. Хидетниemi. – М.: Мир, 1981 – 368 с.
9. Порублёв И. Н., Алгоритмы и программы. Решение олимпиадных задач / И. Н. Порублёв, А. Б. Ставровский – М., СПб-К., Диалектика, 2007. – 480 с.

10. Шень А., Программирование: теоремы и задачи / А. Шень. – М., МЦНМО, 2007 – 296 с.

References:

1. Web-site "The On-Line Encyclopedia of Integer Sequences". Retrieved from: <https://oeis.org/A002182>
2. The Article "Highly composite number". Retrieved from: https://ru.wikipedia.org/wiki/Сверхсоставное_число. [in Russian]
3. The Article "Divisor function". Retrieved from: https://en.wikipedia.org/wiki/Divisor_function
4. Bukhshtab, A. A. (1966). Number theory. – М.: Prosveshcheniie. [in Russian]
5. Dirichlet, P. G. L. (1936). Lectures on number theory. In processing and with additions by R. Dedekind. – М.: NKTP of the USSR. [in Russian]
6. Apostol, Tom M. (1976). Introduction to analytic number theory // Undergraduate Texts in Mathematics, New York-Heidelberg: Springer-Verlag.
7. Little, J. D. C., Murty, K. G., Sweeney, D. W., Karel, C. (1963). An algorithm for the traveling salesman problem. Operations Research, Vol. 11.
8. Goodman, S., Hideniemi, S. (1981). Introduction to the development and analysis of algorithms. – М.: Mir. [in Russian]
9. Porublyov, I. N., Stavrovsky, A. B. (2007). Algorithms and programs. Solving Olympiad problems. – М.-St. Petersburg-K., Dialektika. [in Russian]
10. Shen', A. (2007). Programming: theorems and problems. – М., MTsNMO. [in Russian]

PORUBLYOV Ilya,

senior lecturer, Bohdan Khmelnytsky National University of Cherkasy, Ukraine

SOME ALGORITHMS FOR SEARCHING NUMBERS WITH MAXIMAL QUANTITY OF DIVISORS

Summary. Introduction. Introduced some versions of the problem of searching, among numbers of some range, numbers with maximal quantity of divisors. Introduced reasons, why searching such numbers among range between arbitrary M and N should be considered harder, that among range between 1 and N . Considered generalizations, when maximal quantity of divisors is searched not among all numbers in range, but among numbers, which aren't multiples of some given K , only. Some versions of algorithms, solving some versions of the problem, were designed, based on significant optimizations of brute force approach. A comparative analysis of performance of the algorithms was carried out, the differences and shortcomings for the each of the variants were estimated, and the recommendations about the efficiency of the algorithms were made.

Purpose. The purpose of the article is to describe the designed algorithms for solving problem of searching numbers with maximal quantity of divisors, in such problem formulation versions:

1. Given a positive integer N , for all integers in range between 1 and N (both inclusive), find number having maximal (among numbers in the range) quantity of divisors.
2. Given positive integers M, N ($M \leq N$), for all integers in range between M and N (both inclusive), find number having maximal (among numbers in the range) quantity of divisors.
3. Given positive integers N and K (K significantly less N , $K \geq 2$), for integers in range between 1 and N (both inclusive), but excluding multiples of K , find number having maximal (among the said numbers) quantity of divisors.

To achieve the goal, a number of definitions and statements are given that allow to implement more efficient algorithms compared to the standard ones. Based on the given theoretical information, algorithms are developed and described, which are further implemented in C++.

Results and conclusions. For each problem formulation mentioned in previous section, an algorithm, significantly more effective than obvious one, was developed. Attempts to build adequate asymptotic estimate of their work duration failed (for each of them), but work time duration of each of them was studied experimentally. For searching the divisors on the interval from 1 to N , duration didn't exceed some decimals of second for $N \approx 10^{38}$; for searching the divisors on the interval from some integers M to N , duration didn't exceed some seconds for $N \approx 10^{18}$ (the article provides detailed tables with the results of the algorithms); for searching the divisors on the interval from some integers M to N , if it is not known whether the number M is prime, duration didn't exceed some seconds for

$N \approx 10^{35}$. Thus, the obtained results allow us to assert the greater efficiency of the developed algorithms compared to the standard ones.

Keywords: quantity of divisors, highly composite numbers, brute force and its optimizations.

Одержано редакцією 24.01.2020 р.
Прийнято до публікації 30.03.2020 р.

СЕКЦІЯ «ІНФОРМАТИКА»

УДК 519.6:004.8

DOI 10.31651/2076-5886-2020-1-61-68

PACS 02.60.-x, 02.60.Pn, 02.70.Wz

ПШЕНИШНИЙ Олександр Вадимович,
студент спеціальності
«Інформаційні системи та технології»
Черкаського національного університету
імені Богдана Хмельницького
e-mail: X3morph@gmail.com

**КРАСНОШЛИК Наталія
Олександрівна**,
кандидат технічних наук, доцент,
доцент кафедри прикладної математики та
інформатики Черкаського національного
університету імені Богдана
Хмельницького
e-mail: wlik007@ukr.net
ORCID 0000-0003-4661-6997

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ПОПУЛЯЦІЙНИХ АЛГОРИТМІВ ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧ ГЛОБАЛЬНОЇ ОПТИМІЗАЦІЇ

У роботі розглянуто алгоритми кажанів, світлячків та зозулі для розв'язування задачі глобальної оптимізації. Дані метаевристичні алгоритми відносяться до популяційних алгоритмів, які інтенсивно розвиваються в останні роки. Метою роботи є реалізація та дослідження ефективності їх роботи при розв'язуванні оптимізаційних задач. Досліджено залежність точності й ефективності роботи розглянутих популяційних алгоритмів від параметрів, які задаються, та знайдено їх оптимальні значення. Проведено обчислювальні експерименти з порівняння точності знаходження глобального мінімуму унімодалних і мультимодальних тестових функцій.

Встановлено, що розглянуті популяційні алгоритми є ефективною альтернативою класичним методам розв'язування оптимізаційних задач, оскільки не накладають жодних обмежень на вигляд цільової функції.

Ключові слова: популяційний алгоритм, ройовий інтелект, глобальна оптимізація, алгоритм кажанів, алгоритм світлячків, алгоритм зозулі.

Постановка проблеми

Розв'язування багатьох актуальних технічних, економічних, фізичних, біомедицинських та інших задач зводиться до задач глобальної оптимізації. Чисельні методи оптимізації можна умовно розділити на детерміністичні та стохастичні. Детерміністичні методи, як правило, передбачають знаходження градієнта цільової функції і залежать від вибору початкових значень. Широкого розповсюдження набувають стохастичні методи, які почали інтенсивно розвиватися в 1980-х рр. Відповідні методи в літературі називають метаевристичними, популяційними, багатоагентними і алгоритмами ройового інтелекту (Swarm intelligence) [1].

Метаевристичні алгоритми передбачають одночасну обробку декількох можливих варіантів розв'язку оптимізаційної задачі. Ще однією особливістю подібних алгоритмів

є їх біонічна природа. Популяційні алгоритми описують динаміку поведінки груп різних соціальних тварин, птахів, комах та інших агентів деякої біологічної системи, які локально взаємодіють між собою і з навколишнім середовищем. Колективна поведінка групи агентів дозволяє їм вирішувати різні складні практичні задачі у природі, що свідчить про ефективність алгоритмів ройового інтелекту, які моделюють їх поведінку [2]. Тому досить актуальною є задача розробки та дослідження популяційних алгоритмів для розв'язування оптимізаційних задач.

Серед різноманіття популяційних алгоритмів можна виділити наступні [3]:

- алгоритми, інспіровані живою природою (nature-inspired algorithms);
- алгоритми, інспіровані неживою природою;
- алгоритми, інспіровані людським суспільством та інші.

До найбільш відомих алгоритмів, натхненних живою природою, відносяться: мурашиний алгоритм (Ant Colony Optimization), метод рою частинок (Particle Swarm Optimization), бактеріальна оптимізація (Bacterial Optimisation), алгоритм бджолиного рою (Bees Algorithm), алгоритм світлячків (Firefly Algorithm), пошук косяком риб (Fish School Search), алгоритм зозулі (Cuckoo Search), алгоритм кажанів (Bat Algorithm) та ін.

Алгоритми, інспіровані неживою природою, імітують процеси, що відбуваються у фізиці, хімії і т.п., зокрема це: стохастичний дифузійний пошук (Stochastic Diffusion Search), диференціальна еволюція (Differential Evolution), електромагнітний алгоритм (ElectroMagnetism-like Algorithm), гармонійний пошук (Harmony Search), алгоритм гравітаційного пошуку (Gravitational Search) та ін.

До алгоритмів, інспірованих людським суспільством, належать: алгоритм еволюції розуму (Mind Evolutionary Computation), культурний алгоритм (Cultural Algorithm) та ін.

Об'єктом даного дослідження є процес розв'язування задач глобальної оптимізації за допомогою популяційних алгоритмів. Предметом дослідження виступають популяційні алгоритми світлячків, кажанів та зозулі.

Метою даної статті є реалізація та дослідження ефективності популяційних алгоритмів кажанів, світлячків та зозулі для розв'язування задач глобальної оптимізації.

Методи розв'язання

Розглянемо наступну задачу. Задано цільову функцію $f(x)$, яка визначена на множині допустимих розв'язків $X \subseteq R^m$.

Потрібно знайти глобальний умовний мінімум функції $f(x)$ на множині X :

$$f(x^*) = \min_{x \in X \subseteq R^m} f(x), \quad (1)$$

де $x = (x_1, x_2, \dots, x_m)^T$, $X = \{x \mid x_i \in [x_{\min}, x_{\max}], i \in \overline{1, m}\}$.

Задача пошуку максимуму функції $f(x)$ зводиться до задачі пошуку мінімуму:

$$f(x^*) = \max_{x \in X \subseteq R^m} f(x) = -\min_{x \in X \subseteq R^m} (-f(x)).$$

Алгоритм світлячків (Firefly algorithm) був запропонований Xin-She Yang у 2008 р. Даний алгоритм використовує наступну модель поведінки світлячків [4]:

– світлячки породжують світло з інтенсивністю I , яка обернено пропорційна квадрату відстані. Ця здатність свідчить про те, що світлячок є видимим на обмеженій відстані. Світло допомагає йому притягувати їжу та партнерів;

– привабливість світлячка пропорційна яскравості, тому у випадковій парі світлячків менш яскравий буде прагнути до більш яскравого;

– яскравість зменшується зі збільшенням відстані. Якщо не має більш яскравого світлячка, то він буде рухатися випадковим чином. Яскравість ототожнюється з величиною цільової функції.

Привабливість визначається двома факторами: величиною цільової функції і відстанню до світлячка. Яскравість залежно від відстані до джерела визначається відповідно до закону:

$$I = \frac{I_0}{1 + \gamma r^2},$$

де I_0 – інтенсивність джерела, r – відстань до нього, γ – заданий коефіцієнт (визначає поглинання світла).

В якості функції I можуть бути використані і інші монотонно спадаючі функції, наприклад:

$$I = I_0 e^{-\gamma r^2}.$$

Тоді привабливість світлячка i -го для j -го визначається формулою:

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2},$$

де β_{ij} – взаємна привабливість світлячків, $r_{ij} = \sqrt{\sum_{k=1}^m (r_k^i - r_k^j)^2}$ – відстань між ними.

Алгоритм світлячків для розв’язання задачі глобальної мінімізації полягає у наступному:

- 1: Ініціалізація початкової популяції (задати початкове положення x_i кожного i -го світлячка, $i = \overline{1, N}$) та параметрів алгоритму (α і γ).
- 2: **while** ($t \leq tMax$)
- 3: **for** $i = \overline{1, N}$
- 4: **for** $j = \overline{1, N}$
- 5: **if** $f(x_j) < f(x_i)$
- 6: Переміщення j -го світлячка у напрямку i -го за формулою:

$$x^j = x^i + \beta_0 e^{-\gamma r_{ij}^2} \cdot (x^i - x^j) + \alpha \cdot \varepsilon^j.$$
- 7: **end if**
- 8: Оцінюємо нові рішення та інтенсивність оновлення світла.
- 9: **end for**
- 10: **end for**
- 11: Знаходимо краще поточне положення світлячка.
- 12: Збільшити лічильник кількості ітерацій $t \leftarrow t + 1$.
- 13: **end while**

Алгоритм кажанів (Bat Algorithm) був запропонований Xin-She Yang у 2010 р. і є потенційно більш потужним, ніж алгоритм рою часток або генетичний алгоритм. Основними складовими алгоритму є: особливості переміщення кажанів у просторі та характеристики звукових сигналів, які вони генерують.

Робота алгоритму ґрунтується на наступній моделі поведінки кажанів [5-7]:

– усі кажани використовують ехолокацію, щоб визначити відстань, а також розрізнити їжу/здобич і перешкоди.

– поточне положення кожного кажана позначимо x_i , з нього він переміщується випадковим чином зі швидкістю v_i . У процесі руху кажани генерують звукові сигнали, що мають частоту ω_i і гучність A_i . При цьому вони можуть змінювати як частоту, так і інтенсивність звукових імпульсів $r_i \in [0, 1]$, залежно від близькості до цілі.

– гучність звукового сигналу змінюється від більшого початкового значення A_0 до меншого заданого A_{\min} .

Алгоритм кажанів для розв'язання задачі глобальної мінімізації полягає у наступному:

1: Ініціалізація початкової популяції (задати початкове положення x_i кожного i -го кажана, $i = \overline{1, N}$) та параметрів алгоритму (ω_i – частота, r_i – інтенсивність, A_i – гучність звукових імпульсів кажанів).

2: **while** ($t \leq tMax$)

3: **for** $i = \overline{1, N}$

4: Переміщення кожного i -го кажана за формулами:

$$\begin{aligned}\omega_i &= \omega_{\min} + (\omega_{\max} - \omega_{\min}) \cdot \beta, \\ v_i^{(t+1)} &= v_i^{(t)} + (x_i^{(t)} - \bar{x}) \cdot \omega_i \\ x_i^{(t+1)} &= x_i^{(t)} + v_i^{(t+1)},\end{aligned}$$

де $\omega_i \in [\omega_{\min}, \omega_{\max}]$ – частота; \bar{x} – положення кажана, в якому знайдено кращий глобальний розв'язок; \bar{x}_i – положення кажана, в якому знайдено кращий розв'язок даним кажаном окремо; β – випадкове число з інтервалу $[0, 1]$;

5: **if** ($rand > r_i$)

6: Здійснення локального пошуку в околі поточного положення шляхом випадкового блукання за формулою:

$$x_i^{(new)} = x_i^{(old)} + \varepsilon \cdot \bar{A}(t),$$

де $\bar{A}(t)$ – середнє значення гучності звукових сигналів всіх кажанів; ε – випадкове число з інтервалу $[-1, 1]$.

7: **end if**

8: **if** ($rand < A_i$ and $f(x_i^{(new)}) < f(\bar{x})$)

9 : Прийняти нове положення кажана.

10: Зменшити A_i і збільшити r_i за формулами:

$$A_i^{(t+1)} = \alpha \cdot A_i^{(t)}, \quad r_i^{(t+1)} = r_i^{(0)} \cdot (1 - e^{-\gamma t}),$$

де α, γ – задані коефіцієнти.

11: **end if**

12: Оновити \bar{x} .

13: **end for**

14: Збільшити лічильник кількості ітерацій $t \leftarrow t + 1$.

15: **end while**

Алгоритм зозулі (Cuckoo search) був запропонований Xin-She Yang та Suash Deb у 2009 р. Натхненням для його створення послужив гніздовий паразитизм деяких видів зозуль, що підкладають свої яйця до гнізд інших птахів.

Робота алгоритму ґрунтується на наступній моделі поведінки даних птахів [8]:

– кожна зозуля відкладає одне яйце за один раз, і підкидає його в гніздо, яке вибирає випадково;

– найкращі яйця з найкращою оцінкою (потрібними розв'язками) переходять на наступне покоління;

– число доступних гнізд фіксовано, а яйце зозулі може бути знайдене господарем з ймовірністю $p \in [0;1]$. Виявленні розв'язки виключаються із подальшого розв'язування.

Алгоритм зозулі для розв'язання задачі глобальної мінімізації полягає у наступному:

1: Ініціалізація початкової популяції (задати початкові положення гнізд x_i , $i = \overline{1, N}$) та параметрів алгоритму (λ – параметр розподілу Леві, p – частина гнізд, які видаляються, α – параметр зміни кроку).

2: **while** ($t \leq tMax$)

3: **for** $i = \overline{1, N}$

4: Переміщення зозулі випадковим чином, відповідно до польотів Леві

$$x_i^{(t+1)} = x_i^{(t)} + \frac{\alpha}{t+1} \cdot Levy(\lambda),$$

де $Levy(\lambda)$ – випадкова величина, що генерується розподілом Леві; α – величина кроку.

5: Вибрати випадковим чином гніздо x_j .

6: **if** $f(x_i) > f(x_j)$

7: Замінити гніздо з номером j на нове.

8: **end if**

9: Частину гнізд, виявлених із ймовірністю p , видалити із популяції.

10: Згенерувати таку ж кількість нових гнізд.

11: **end for**

12: Знаходимо краще поточне гніздо.

13: Збільшити лічильник кількості ітерацій $t \leftarrow t + 1$.

14: **end while**

Результати дослідження

Для проведення обчислювальних експериментів обрали наступні тестові функції:

1) функція де Джонга: $f_1(x) = \sum_{i=1}^m x_i^2$, $-10 \leq x_i \leq 10$. Глобальний мінімум функції

$f_1(x^*) = 0$ при $x^* = (0, 0, \dots, 0)$.

2) функція Розенброка: $f_2(x) = \sum_{i=1}^{m-1} ((x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2)$, $-5 \leq x_i \leq 5$. Глобальний

мінімум функції $f_2(x^*) = 0$ при $x^* = (1, 1, \dots, 1)$.

3) функція Растрігіна: $f_3(x) = \sum_{i=1}^m (x_i^2 - 10 \cos(2\pi x_i)) + 10m$, $-5 \leq x_i \leq 5$. Глобальний мінімум функції $f_3(x^*) = 0$ при $x^* = (0, 0, \dots, 0)$.

Популяційні алгоритми світлячків, кажанів та зозулі реалізовані у середовищі MATLAB R2012b. Для кожного алгоритму експериментально встановлені значення констант і параметрів, при яких знайдено порівняно найкращі розв'язки.

Для алгоритму світлячків обрано наступні значення: $\alpha = 0.15$, $\gamma = 1$, $\beta_0 = 0.1$. Для алгоритму кажанів обрано параметри і початкові значення: $\alpha = 0.9$, $\gamma = 0.9$, $A_i^{(0)} = 0.35$, $r_i^{(0)} = 0.4$. А у алгоритмі зозулі обрали: $p=0.25$, $\lambda=1.5$, $\alpha=1$. У всіх алгоритмах покладали $tMax = 1000$, $N = 20$.

Алгоритми кажанів, світлячків та зозулі містять ряд варійованих параметрів, від значення яких, може сильно залежить їх ефективність. Крім того, вплив даних параметрів сильно залежить від виду цільової функції. Тому оптимальні значення вхідних параметрів для розглянутих алгоритмів повинні вибиратися в кожному конкретному випадку індивідуально.

При порівнянні результатів розв'язування задачі (1), кожен алгоритм застосовувався для кожної функції 50 разів. Отримані результати для набору тестових функцій наведено в табл. 1.

Таблиця 1

Результати обчислювальних експериментів для тестових функцій

Тестова функція	Розмірність простору, m	Алгоритм	Краще отримане значення цільової функції	Середнє серед отриманих значень цільової функції
$f_1(x)$	5	світлячків	1.5678e-03	5.6784e-01
		кажанів	7.5366e-03	3.4367e-01
		зозулі	8.8369e-03	9.4356e-01
	10	світлячків	1.0420e-01	5.1940e-00
		кажанів	2.5723e-01	5.6863e+00
		зозулі	1.6783e-02	6.4344e-01
$f_2(x)$	5	світлячків	7.5366e+00	5.5676e+01
		кажанів	4.8717e+00	7.8534e+01
		зозулі	6.8576e-01	0.8943e+01
	10	світлячків	5.7401e+01	8.8465e+02
		кажанів	4.8663e+01	6.8372e+02
		зозулі	3.8672e+01	6.9383e+02
$f_3(x)$	5	світлячків	8.6573e+00	4.7822e+01
		кажанів	9.2335e+00	7.1244e+01
		зозулі	1.9023e-01	6.3476e+00
	10	світлячків	3.6533e+01	9.7634e+01
		кажанів	4.2676e+01	9.7346e+01
		зозулі	9.9703e+00	7.8352e+01

Результати обчислювальних експериментів показали, що в окремих випадках алгоритм зозулі є більш ефективним у порівнянні з іншими алгоритмами для розв'язування задачі глобальної оптимізації.

В цілому, розглянуті популяційні алгоритми є ефективною альтернативою класичним методам розв'язування оптимізаційних задач, оскільки є нескладними для

програмної реалізації, невибагливими до виду цільової функції та дозволяють знайти розв'язок з порівняно високою точністю.

Висновки

У роботі розглянуто задачу знаходження глобального мінімуму цільової функції за допомогою алгоритмів, інспірованих поведінкою кажанів, світлячків та зозулі, які відносяться до популяційних алгоритмів.

Здійснено програмну реалізацію даних алгоритмів, досліджено їх ефективність для розв'язування задачі глобальної оптимізації числових функцій. В результаті обчислювальних експериментів встановлено, що в окремих випадках алгоритм зозулі є більш ефективним у порівнянні з іншими. При цьому кожен популяційний алгоритм містить ряд варійованих параметрів, від значення яких, може сильно залежить їх ефективність. Тому не можна визначити конкретний алгоритм, який завжди дає найкращі результати для всіх функцій.

Завдяки використанню різних тестових функцій виявлено, що популяційні алгоритми дозволяють знайти глобальний мінімум функції будь-якої складності, не накладаючи додаткових обмежень на неї. Розглянуті алгоритми є досить конкурентоздатними та ефективними для оптимізації складних функцій при розв'язуванні практичних задач.

Список використаної літератури:

1. Карпенко, А. П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой : учебное пособие / А. П. Карпенко. – Москва : Издательство МГТУ им. Н. Э. Баумана, 2014. – 448 с.
2. Субботін, С. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейронмережних моделей : монографія / С. О. Субботін, А. О. Олійник, О. О. Олійник ; під заг. ред. С. О. Субботіна. – Запоріжжя : ЗНТУ, 2009. – 375 с.
3. Карпенко, А. П. Популяционные алгоритмы глобальной поисковой оптимизации. Обзор новых и малоизвестных алгоритмов / А. П. Карпенко // Приложение к журналу «Информационные технологии». – 2012. – № 7. – С. 1-32.
4. Yang, X. S. Firefly algorithm, stochastic test functions and design optimization / X. S. Yang // International Journal of Bio-inspired Computation. – 2010. – Vol. 2 (2). – P. 78–84.
5. Yang, X. S. A new metaheuristic bat-inspired algorithm / X. S. Yang // Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). – 2010. – Vol. 284. – P. 65-74.
6. Yang, X. S. Bat algorithm for multi-objective optimization / X. S. Yang // International Journal of Bio-Inspired Computation. – 2011. – Vol. 3. – № 5. – P. 267-274.
7. Красношлык, Н.А. Решение задачи глобальной оптимизации модифицированным алгоритмом летучих мышей / Красношлык Н.А. // Радиоелектроніка. Інформатика. Управління. – 2015. – № 4(35). – С. 96-103.
8. Yang, X. S. Cuckoo search via Lévy flights / X. S. Yang, S. Deb // World Congress on Nature & Biologically Inspired Computing (NaBIC 2009) IEEE Publications. – 2009. – P. 210–214.

References:

1. Karpenko, A. P. (2014). Modern search engine optimization algorithms. Nature-Inspired Algorithms: A Tutorial / A. P. Karpenko. – Moskva : Izdatelstvo MGTU im. N. E. Bauman [in Russian].
2. Subbotin, S. O., Oliinyk, A. O., Oliinyk, O. O. (2009). Non-iterative, evolutionary and multiagent methods of synthesis of fuzzy and neural network models. – Zaporizhzhia : ZNTU [in Ukrainian].
3. Karpenko, A. P. (2012). Population algorithms for global search engine optimization. Review of new and little-known algorithms. Prilozheniye k zhurnalu «Informatsionnyye tekhnologii», 7, 1-32 [in Russian].
4. Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimization. International Journal of Bio-inspired Computation, Vol. 2 (2), 78-84.
5. Yang, X. S. (2011). A new metaheuristic bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Vol. 284, 65-74.
6. Yang, X. S. (2011). Bat algorithm for multi-objective optimization. // International Journal of Bio-Inspired Computation, Vol. 3(5), 267-274.

7. Krasnoshlyk, N.A. (2015). Solution of the global optimization problem by the modified bat algorithm. Radioelektronika. Informatyka. Upravlinnia, № 4(35), 96-103 [in Ukrainian].
8. Yang, X. S., Deb, S. (2009). Cuckoo search via Lévy flights. World Congress on Nature & Biologically Inspired Computing (NaBIC 2009) IEEE Publications, 210–214.

PSHENISHNIY Oleksandr,

Student, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

KRASNOSHLYK Nataliia,

PhD, Senior Lecturer, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

INVESTIGATION OF THE EFFICIENCY OF POPULATION ALGORITHMS FOR SOLVING GLOBAL OPTIMIZATION PROBLEMS

Introduction. *The solution of many technical, economic, physical, biomedical and other problems comes down to global optimization. Meta-heuristic methods are widely used, which involve the simultaneous processing of several possible solutions to the optimization problem. Relevant methods in the literature are called population, multi-agent and swarm intelligence algorithms.*

Purpose. *The purpose of this article is to implement and investigation the effectiveness of population methods bat algorithm, firefly algorithm and cuckoo search to solve global optimization problems.*

Results. *Population algorithms bat algorithm, firefly algorithm and cuckoo search are implemented in MATLAB R2012b environment.*

These algorithms contain a number of varied parameters, the value of which may greatly affect their effectiveness. In addition, the influence of these parameters strongly depends on the type of objective function. Therefore, the optimal values of the input parameters for the considered algorithms must be selected in each case individually.

The results of computational experiments have shown that in some cases the cuckoo algorithm is more efficient than other algorithms for solving the global optimization problem.

In general, the considered population algorithms are an effective alternative to the classical methods of solving optimization problems, as they are simple for software implementation, unpretentious to the type of objective function and allow to find a solution with relatively high accuracy.

Conclusion. *The software implementation of bat algorithm, firefly algorithm and cuckoo search is performed and their efficiency for solving the problem of global optimization of numerical functions is investigated. Each population algorithm contains a number of varied parameters, the value of which may greatly affect their effectiveness. Therefore, it is not possible to determine a specific algorithm that always gives the best results for all functions.*

Due to the use of various test functions, it was found that population algorithms allow to find the global minimum of a function of any complexity, without imposing additional restrictions on it. The considered algorithms are quite competitive and effective for optimization of complex functions in solving practical problems.

Keywords: *population algorithm, swarm intelligence, global optimization, bat algorithm, firefly algorithm, cuckoo search.*

*Одержано редакцією 12.02.2020 р.
Прийнято до публікації 27.04.2020 р.*

УДК 004.85:519.6

DOI 10.31651/2076-5886-2020-1-69-77

PACS 02.70.Wz, 07.05.Kf, 07.05.Mh,
07.05.Tr

МОТОРНА Ярослава Сергіївна,
магістрантка спеціальності «Прикладна
математика» Черкаського національного
університету імені Богдана Хмельницького

**КРАСНОШЛИК Наталія
Олександрівна,**
кандидат технічних наук, доцент,
доцент кафедри прикладної математики та
інформатики Черкаського національного
університету імені Богдана Хмельницького
e-mail: wlik007@ukr.net
ORCID 0000-0003-4661-6997

ПІСКУН Олександр Варфоломійович,
кандидат технічних наук, доцент,
завідувач кафедри прикладної математики
та інформатики Черкаського національного
університету імені Богдана Хмельницького
e-mail: piskun@ukr.net
ORCID 0000-0001-5334-6337

РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ АЛГОРИТМУ RANDOM FOREST ДЛЯ РОЗВ'ЯЗУВАННЯ ЗАДАЧ КЛАСИФІКАЦІЇ

У роботі проведено оцінку можливості застосування алгоритму випадкового лісу до розв'язання задач класифікації даних. Подано загальну постановку задачі класифікації даних, описано поняття дерева рішень та наведено основні алгоритми побудови таких дерев. Детально розглянуто алгоритм CART та описано розрахунок основного індексу, на основі якого проводиться побудова дерева для розв'язання задачі класифікації. Розглянуто використання алгоритму CART у ансамблевих методах, на основі яких будується випадковий ліс. Описано основні поняття випадкового лісу, реалізацію випадкового лісу та набір тестових даних, на основі яких проводилась перевірка реалізованого програмного продукту. Результати роботи реалізованого алгоритму випадкового лісу показали дієвість застосованого підходу до розв'язання задач класифікації.

Ключові слова: машинне навчання, задача класифікації, дерево рішень, Random Forest.

Вступ

Дерева рішення є одним з досить часто використовуваних способів побудови алгоритмів розпізнавання в сучасних системах обробки інформації у зв'язку з їх численними перевагами, а саме: забезпеченням можливості роботи як з суттєво значними ознаками, так і з категоріальними ознаками, гарну стійкість до аномальних значень спостережень, можливістю наочної інтерпретації одержуваного розв'язувального правила і пояснення досліджуваної закономірності у даних для їх застосування у експертних системах. однією з задач, що розв'язуються за допомогою експертних систем, є задача класифікації, яка полягає у визначенні класу об'єкта, тобто, співставленні аналізованого об'єкта з одним з відомих експертній системі класів.

Можливості застосування окремого виду ансамблю дерев рішень – випадковому лісу – для розв’язування задачі класифікації й присвячена дана стаття.

Метою статті є реалізація і дослідження алгоритму Random Forest та його застосування до розв’язування задач класифікації.

Виклад основного матеріалу

1. Постановка задачі класифікації даних

Задачі класифікації досить часто зустрічаються у найрізноманітніших областях діяльності людини. Прикладами таких задач є проблеми медичного діагностування, оцінювання кредитного ризику, визначення тенденцій на фінансових ринках тощо.

Задача класифікації належить до задач машинного навчання з учителем. Її метою є поділ деякої множини об’єктів на заздалегідь визначену кількість класів. При цьому для підмножини об’єктів, яку називають навчальною вибіркою, клас об’єкта відомий заздалегідь.

Нехай X – множина характеристик об’єктів, Y – множина номерів (або найменувань) класів. Існує невідома “цільова залежність” – відображення, значення якого відомі тільки на об’єктах навчальної вибірки $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$. Для розв’язання задачі потрібно побудувати алгоритм $\alpha : X \rightarrow Y$, що апроксимує цільову залежність [1].

2. Древа рішень і алгоритми їх побудови

Древа рішень (також можуть називатися деревами класифікації або регресійними деревами) використовуються в області машинного навчання, статистики та аналізу даних у якості прогнозуючих моделей [2].

Структура дерева містить “листя” і “гілки”. На ребрах (гілках) дерева рішень записані атрибути, від яких залежить цільова функція, у листі записані значення цільової функції, а в інших вузлах – атрибути, за якими розрізняються об’єкти. Кожен внутрішній вузол відповідає одній з вхідних змінних. Щоб класифікувати новий об’єкт, треба спуститися по дереву до листа і отримати відповідне значення. Приклад дерева рішень наведено на рис. 1.

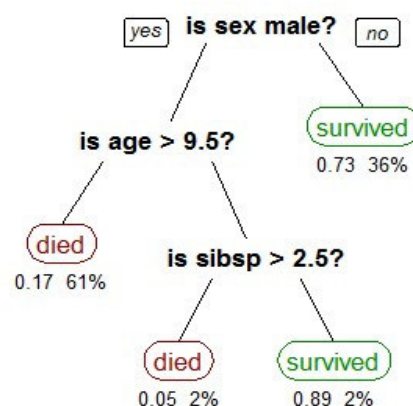


Рис. 1. Приклад дерева рішень

Дерево рішень отримується поділом вхідних наборів змінних на підмножини на основі тестування значень атрибутів. Цей процес повторюється на кожній з отриманих підмножин. Рекурсія завершується тоді, коли підмножина у вузлі має ті ж значення

цільової змінної. Такий процес, що йде зверху вниз, називається індукцією дерев рішень (TDIDT), і є прикладом жадібного алгоритму, що на сьогоднішній день є найбільш поширеною стратегією побудови дерев рішень для даних.

Алгоритми побудови дерева рішень

Загальна схема застосування дерева рішень для класифікації об'єктів тестової вибірки має наступний вигляд.

Листами дерева рішень є класи. Щоб класифікувати об'єкт за допомогою дерева рішень, потрібно послідовно спускатися по дереву, вибираючи напрямок розгалуження, опираючись при цьому на значення предикатів, що містяться у вузлах дерева, шляхом застосування їх до об'єктів, які потрібно класифікувати. Шлях від кореня дерева до листа можна трактувати як пояснення того, чому той чи інший об'єкт віднесений до якого-небудь класу.

Ключовою особливістю алгоритмів побудови дерева рішень є спосіб вибору чергового атрибуту. Існують наступні алгоритми [3]:

- алгоритм ID3, де вибір атрибуту відбувається на підставі приросту або на підставі індексу *Gini*;
- алгоритм C4.5 (покращена версія ID3), де вибір атрибуту відбувається на підставі нормалізованого приросту інформації;
- алгоритм CART і його модифікації – IndCART, DB-CART.

У загальному вигляді алгоритм побудови дерева рішень подано у лістингу псевдокоду 1.

Псевдокод 1

обчислюємо ентропію вхідної множини s_0

якщо $s_0 = 0$, то:

- 1) всі об'єкти вхідного набору, належать до одного класу;
- 2) зберігаємо цей клас як лист дерева;

якщо $s_0 \neq 0$, то:

- 1) шукаємо предикат, який розбиває дану множину таким чином, щоб зменшилося середнє значення ентропії;
- 2) знайдений предикат є частиною дерева рішень, зберігаємо його;
- 3) розбиваємо вхідну множину на підмножини, відповідно до даного предикату;
- 4) повторюємо процедуру рекурсивно для кожної підмножини.

Щоб знайти предикат для розбиття на підмножини, потрібно для кожного елемента перебирати всі його атрибути. Для кожного елемента генеруємо предикат, який розбиває вхідну множину на дві підмножини. Потім розраховуємо середнє значення ентропії та обчислюємо ΔS . Обираємо предикат з найбільшим значенням ΔS .

У найпростішому випадку можна використовувати предикати, які відносяться тільки до значення якого-небудь атрибуту (наприклад « $x \geq 12$ », або «колір == жовтий»).

Одне з питань, яке виникає у алгоритмі побудови дерева рішень, – це оптимальний розмір кінцевого дерева. Так, невелике дерево може не охопити ту чи іншу важливу інформацію про об'єкти вибірки. При цьому важко визначити, коли алгоритм повинен зупинитися, тому що неможливо спрогнозувати, додавання якого вузла дозволить значно зменшити загальну похибку. Виникає потреба у регулюванні глибини отриманого дерева рішень.

Регулювання глибини дерева – це техніка, яка дозволяє зменшувати розмір дерева рішень, видаляючи ділянки дерева, які мають невелику вагу. Тобто, загальна стратегія обмеження дерева реалізується шляхом видалення вузлів, якщо вони не несуть додаткової інформації.

Скорочення дерева може здійснюватися зверху вниз або знизу вгору. Зверху вниз – обрізка починається з кореня, а знизу вгору – скорочується кількість листів дерева. Один з найпростіших методів регулювання – починаючи з листів, кожен вузол замінюється на найпопулярніший клас. Якщо така заміна не впливає на точність прогнозування, то вона приймається.

Алгоритм побудови дерева рішень CART

Алгоритм CART (Classification and Regression Tree) призначений для розв'язування задач класифікації та регресії. Він розроблений у 1974-1984 роках чотирма професорами статистики: Лео Брейманом (Leo Breiman, Берклі), Джеромом Фрідманом (Jerome H. Friedman, Стенфорд), Чарлзом Стоуном (Charles Stone, Берклі) і Річардом Олшеном (Richard A. Olshen, Стенфорд) [4].

У алгоритмі CART кожен вузол дерева рішень має два нащадки. На кожному кроці побудови дерева правило, сформоване у вузлі, ділить задану множину об'єктів (навчальну вибірку) на дві частини: частина, у якій виконується правило (правий нащадок – right), і частина, у якій правило не виконується (лівий нащадок – left). Для вибору оптимального правила використовується функція оцінки якості розбиття.

Оціночна функція, яка використовується алгоритмом CART, базується на інтуїтивній ідеї зменшення невизначеності у вузлі. Мається на увазі таке розбиття, при якому у вузлі буде якомога більше прикладів одного класу та як можна менше усіх інших. Це поняття близьке до ентропії, але тут використовується інша міра невизначеності, для якої використовується термін “не чистий вузол”.

У алгоритмі CART ідея “не чистого вузла” формалізована в індексі *Gini* [5]. Якщо набір даних T містить дані n класів, тоді індекс *Gini* визначається як

$$Gini(T) = 1 - \sum_{i=1}^n p_i^2,$$

де параметр p_i – ймовірність класу i в T .

Якщо набір T розбивається на дві частини, T_1 і T_2 з кількістю прикладів у кожній N_1 і N_2 відповідно, то показник якості розбиття буде дорівнювати

$$Gini_{split}(T) = \frac{N_1}{N} Gini(T_1) + \frac{N_2}{N} Gini(T_2).$$

Найкращим вважається те розбиття, для якого $Gini_{split}(T)$ мінімальне.

Позначимо через N кількість прикладів у вузлі, через L і R – кількість прикладів у лівому і правому нащадках відповідно, l_i і r_i – кількість екземплярів i -го класу у лівому/правому нащадку. Тоді якість розбиття оцінюється за наступною формулою:

$$Gini_{split} = \frac{L}{N} \left(1 - \sum_{i=1}^n \left(\frac{l_i}{L} \right)^2 \right) + \frac{R}{N} \left(1 - \sum_{i=1}^n \left(\frac{r_i}{R} \right)^2 \right) \rightarrow \min.$$

Щоб зменшити обсяг обчислень, формулу для $Gini_{split}$ можна переписати інакше:

$$Gini_{split} = \frac{1}{N} \left(L \left(1 - \frac{1}{L^2} \sum_{i=1}^n l_i^2 \right) + R \left(1 - \frac{1}{R^2} \sum_{i=1}^n r_i^2 \right) \right).$$

Так як множення на константу не відіграє ролі при мінімізації, то

$$Gini_{split} = L - \frac{1}{L} \sum_{i=1}^n l_i^2 + R - \frac{1}{R} \sum_{i=1}^n r_i^2 \rightarrow \min,$$

$$Gini_{split} = N - \left(\frac{1}{L} \sum_{i=1}^n l_i^2 + \frac{1}{R} \sum_{i=1}^n r_i^2 \right) \rightarrow \min,$$

$$\tilde{G}_{split} = \frac{1}{L} \sum_{i=1}^n l_i^2 + \frac{1}{R} \sum_{i=1}^n r_i^2 \rightarrow \max.$$

Отже, найкращим буде те розбиття, для якого величина \tilde{G}_{split} максимальна. Рідше в алгоритмі CART використовуються інші критерії розбиття: *Twoing*, *Symmetric Gini* та ін. [4].

Правила розбиття

Вектор предикторних змінних, що подається на вхід дереву, може містити як числові (порядкові), так і категоріальні змінні. У будь-якому випадку, у кожному вузлі розбиття здійснюється лише по одній змінній. Якщо змінна має числовий тип, то у вузлі формується правило виду $x_i \leq c$, де c – деякий поріг, який найчастіше вибирається як середнє арифметичне двох сусідніх упорядкованих значень змінної навчальної вибірки. Якщо змінна має категоріальний тип, то у вузлі формується правило $x_i \in V(x_i)$, де $V(x_i)$ – деяка непорожня підмножина множин значень змінної x_i у навчальній вибірці. Отже, для n значень числового атрибуту алгоритм порівнює $n - 1$ розбиття, а для категоріального – $(2^{n-1} - 1)$. На кожному кроці побудови дерева алгоритм послідовно порівнює всі можливі розбиття для всіх атрибутів і вибирає найкращий атрибут і найкращу розбивку для нього [4].

3. Алгоритм Random Forest

Ансамбль алгоритмів – це певна сукупність алгоритмів, які об'єднуються для розв'язання однієї спільної задачі і утворюють єдине ціле.

Беггінг (Bagging від Bootstrap aggregation) – це один з перших і найпростіших видів ансамблів [6]. Він базується на статистичному методі бутстрепа.

Метод бутстрепа полягає у наступному. Нехай ϵ вибірка X розміру N . Рівномірно візьмемо з вибірки N об'єктів з поверненням. Це означає, що ми будемо N разів вибирати довільний об'єкт вибірки (вважаємо, що кожен об'єкт “досягається” з однаковою ймовірністю $1/N$), причому кожного разу ми обираємо з усіх початкових N об'єктів. Відзначимо, що внаслідок наявності повернень серед них можуть бути повтори. Позначимо нову вибірку через X_1 . Повторюючи процедуру M разів, згенеруємо M підвбірок X_1, \dots, X_M .

Для побудови ансамблю алгоритмів на основі беггінгу за допомогою бутстрепа генерують вибірки, на кожній з яких навчають свій класифікатор $a_i(x)$. Результуючий класифікатор буде усереднювати відповіді усіх алгоритмів (у разі класифікації це відповідає голосуванню):

$$a(x) = \frac{1}{M} \sum_{i=1}^M a_i(x).$$

Беггінг дозволяє знизити дисперсію класифікатора і запобігає перенавчанню. Ефективність беггінга досягається завдяки тому, що базові алгоритми, які навчалися на різних підвбірках, отримуються досить різними, і їх помилки взаємно компенсуються при голосуванні. Крім того, об'єкти-викиди можуть не потрапити до деяких навчальних підвбірок.

Random Forest (випадковий ліс) – один з найбільш ефективних алгоритмів машинного навчання, запропонований Лео Брейманом [7]. Він являє собою множину дерев рішень. У задачі регресії їх відповіді усереднюють, а у задачі класифікації приймається рішення голосуванням за більшістю.

Всі дерева ансамблю будуються незалежно один від одного за наступною процедурою:

- згенерувати випадкову підвбірку розміром n з навчальної вибірки;
- побудувати дерево рішень, причому в ході створення чергового вузла дерева розглядають не всі ознаки, а лише m випадково обраних ознак, на основі яких буде проводитися розбиття;
- дерево будується до повного вичерпання об'єктів підвбірки і не піддається процедурі відсікання гілок.

Алгоритм Random Forest може бути використаний у задачі оцінки важливості ознак. Для цього необхідно навчити алгоритм на вибірці і під час побудови моделі для кожного елемента навчальної вибірки порахувати out-of-bag-помилку. Потім для кожного об'єкта така помилка усереднюється по всьому випадковому лісі. Щоб оцінити важливість ознаки, його значення перемішуються для усіх об'єктів навчальної вибірки і out-of-bag-помилка обчислюється знову. Важливість ознаки оцінюється шляхом усереднення різниці показників out-of-bag-помилки по всіх деревах до і після перемішування значень. При цьому значення таких помилок нормалізуються на стандартне відхилення.

4. Реалізація і дослідження алгоритму Random Forest

Будемо розглядати задачу класифікації. Це формалізована задача, в якій задана множина об'єктів, розділених деяким чином на класи. Задана кінцева множина об'єктів, для яких відомо, до яких класів вони відносяться. Ця множина називається вибіркою. Класова належність інших об'єктів не відома. Потрібно побудувати алгоритм, здатний класифікувати довільний об'єкт з початкової множини. Класифікувати об'єкт – означає вказати номер (або назву) класу, до якого відноситься даний об'єкт [1].

Алгоритм Random Forest був реалізований у середовищі Octave. Реалізацію дерева рішень здійснювали за алгоритмом CART [8]. Реалізовані програми дозволяють розв'язувати як задачу бінарної (двокласової класифікації), так і багатокласової класифікації.

Для дослідження алгоритму Random Forest розглядали наступні набори даних [9]:

- бінарна класифікація:
 - Titanic (класифікація пасажирів);
 - Seeds (класифікація ядра пшениці);
 - Telecom_churn (класифікація відтоку клієнтів телеком оператора);
 - CreditScoring (класифікація видачі кредитів);
- багатокласова класифікація:
 - Iris (класифікація ірисів);
 - Wine Quality (класифікація винних виробів).

Наведемо опис розглянутих наборів даних.

Набір даних Titanic (класифікація пасажирів) містить дані про справжніх пасажирів на кораблі “Титанік”. Кожний рядок представляє одну особу. Колонки описують різні атрибути про особу: вік (age), клас (pclass), номер квитка (ticket), вартість проїзду, яку заплатили (fare). Пасажирів класифікують за двома класами: 1 – вижили, 2 – не вижили.

Набір даних Seeds (класифікація ядра пшениці) містить дані про ядра, які належать до двох різних сортів пшениці: “Елегія”, “Олеся”. Для побудови даних були виміряні шість геометричних параметрів ядра пшениці: периметр, компактність, довжина ядра, ширина ядра, коефіцієнт несиметрії, довжина канавки ядра.

Набір даних Telecom_churn містить дані про клієнтів телеком оператора, які належать до двох різних класів: втрата клієнта (відтік), клієнт залишився. Для побудови класифікатора застосовувались такі ознаки: загальна кількість дзвінків вдень, загальна кількість дзвінків вночі, загальна кількість дзвінків увечері, загальна кількість міжнародних дзвінків, загальна сума оплати за міжнародні дзвінки, число звернень у сервісний центр. Клієнти класифікуються за двома класами: 1 – втрата клієнта, 2 – клієнт залишився

CreditScoring (класифікація видачі кредитів). Наявні дані про клієнтів банку, які класифікуються на два класи: банк видасть кредит, банк не видасть кредит. Для побудови даних були взяті наступні ознаки: вік клієнта, термін зайнятості, кількість дітей, заробітня плата, чи має клієнт нерухомість, скільки активних кредитів. Клієнти класифікуються за двома класами: 1 – видати кредит, 2 – не видати.

Iris (класифікація ірисів). Набір містить дані про види ірисів. Він включає в себе три види ірисів, по 50 кожного виду, а також деякі властивості кожної квітки: довжина пелюстки, ширина пелюстки, довжина чашолистика, ширина чашолистика. Іриси класифікуються за трьома класами: 1 – Iris-setosa, 2 – Iris-versicolor, 3 – Iris-virginica.

Wine Quality (класифікація винних виробів). Наявні дані про різноманітні сорти білих вин, які належать до трьох різних класифікацій якості: хороша, середня та найгірша. Для побудови даних були виміряні властивості вина: фіксована кислотність, залишковий цукор, рН, сульфати, летюча кислотність, вміст алкоголю. Сорти класифікуються за трьома класами: 1 – хороша якість, 2 – середня якість, 3 – найгірша якість

Якість роботи алгоритму оцінювалась за часткою правильних відповідей на відкладеній вибірці. Для набору даних Titanic отримано правильних відповідей 89.04%, для Seeds – 94.11%, для Telecom_churn – 91.66%, для CreditScoring – 90.47%, для Iris – 95.94%, для Wine Quality – 93.65%. На рис. 2 наведено приклади виклику алгоритму Random Forest для наборів даних Titanic і CreditScoring.

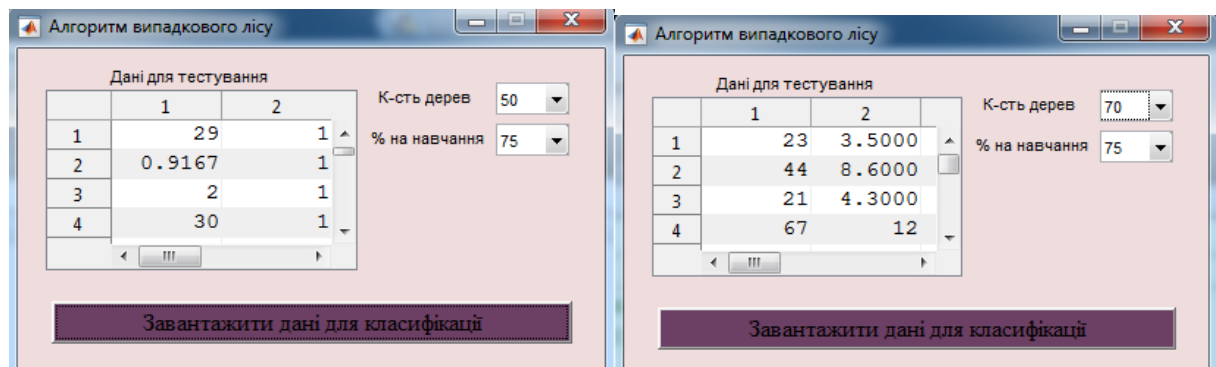


Рис. 2. Приклад застосування алгоритму Random Forest до класифікації даних

Як видно з поданого вище, отримані результати показують дієвість застосування ансамблю дерева рішень до задач класифікації.

Висновки

У результаті проведеної роботи отримано підтвердження дієвості застосування ансамблевих методів на основі дерев рішень, які будуються за допомогою алгоритму CART. Тестування показало точність результатів, більшу за 90%, що свідчить про високу якість роботи алгоритму. Цілком зрозуміло, що така якість отримується завдяки використанню кількох різних дерев рішень, результати роботи яких узагальнюються при виведенні остаточної оцінки.

Список використаної літератури:

1. Классификация [Електронний ресурс]. – Режим доступу: <http://www.machinelearning.ru/wiki/index.php?title=Классификация>.
2. Chelliah, I. Understanding Decision Trees in Machine Learning [Електронний ресурс]. – Режим доступу: <https://medium.com/better-programming/understanding-decision-trees-in-machine-learning-86d750e0a38f>
3. Деревья решений: общие принципы [Електронний ресурс]. – Режим доступу: <https://loginom.ru/blog/decision-tree-p1>
4. Breiman, L. Classification and Regression Trees / L. Breiman, J. H. Friedman, R. A. Olshen, C. T. Stone. – Chapman and Hall/CRC, 1984. – 368 p.
5. Андреев, И. Деревья решений – CART математический аппарат. Часть 1. [Електронний ресурс]. – <https://basegroup.ru/community/articles/math-cart-part1>
6. Rocca, J. Ensemble methods: bagging, boosting and stacking [Електронний ресурс]. – <https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205>
7. Breiman, L. Random Forests / Leo Breiman // Machine Learning. – 2001. – Vol. 45. - № 1. – P. 5 – 32.
8. Patel, F. Decision Tree the CART Algorithm. [Електронний ресурс]. – <https://medium.com/analytics-vidhya/decision-tree-the-cart-algorithm-28c481d28813>
9. Datasets [Електронний ресурс]. – <https://www.kaggle.com/datasets>

References:

8. Klassifikatsiya [Classification]. (2011). Retrieved from [http://www.machinelearning.ru/wiki/index.php?title= Classification](http://www.machinelearning.ru/wiki/index.php?title=Classification). [in Russian]
9. Chelliah, Indhumathy. (2020). Understanding Decision Trees in Machine Learning. Retrieved from: <https://medium.com/better-programming/understanding-decision-trees-in-machine-learning-86d750e0a38f>
10. Derevia resheniy: obshchiye printsipy [Decision trees: general principles]. (2019). Retrieved from <https://loginom.ru/blog/decision-tree-p1>. [in Russian]
11. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. T. (1984). Classification and Regression Trees. Chapman and Hall/CRC.
12. Andreev, I. (2005). Derevia resheniy – CART matematicheskiiy apparat. Chast 1. [Decision trees – CART mathematical apparatus. Part 1]. Retrieved from <https://basegroup.ru/community/articles/math-cart-part1>. [in Russian]
13. Rocca, J. (2019). Ensemble methods: bagging, boosting and stacking. Retrieved from <https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205>
14. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.
15. Patel F. (2020). Decision Tree the CART Algorithm. Retrieved from <https://medium.com/analytics-vidhya/decision-tree-the-cart-algorithm-28c481d28813>
16. Datasets. (2021). Retrieved from <https://www.kaggle.com/datasets>

MOTORNA Yaroslava,

student, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

KRASNOSHLYK Natalia,

Candidate of Technical Sciences, Associate Professor, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

PISKUN Oleksandr,

Candidate of Technical Sciences, Associate Professor, Head of Department of Applied Mathematics and Informatics, Bohdan Khmelnytsky National University of Cherkasy, Ukraine

IMPLEMENTATION AND RESEARCH OF THE RANDOM FOREST ALGORITHM TO SOLVE CLASSIFICATION PROBLEMS

Summary. Introduction. Solution trees are one of the most commonly used ways to build recognition algorithms in modern information processing systems due to their many advantages, in particular, the possibility of their application in expert systems. One of the problems solved by expert systems is the classification problem, which is to determine the class of an object from a set of classes known to the expert system. This article is devoted to the possibility of using one type of decision tree ensemble - a random forest - to solve the classification problem.

The purpose of this paper is to implement and study the Random Forest algorithm and its application to solving classification problems.

Results. The problem of classification by means of a random forest is considered in the work. This is a formalized task that specifies a set of objects that are divided into classes in some way. A finite set of objects is specified for which classes they are known. The Random Forest algorithm is implemented in the Octave environment. The decision tree is implemented according to the CART algorithm. Implemented programs allow us to solve both the problem of binary (two-class classification) and multi-class classification.

To study the Random Forest algorithm, the following data sets were considered: passenger classification, seed classification, mobile operator customer outflow classification, credit issuance classification, iris classification, wine classification. The quality of the algorithm was evaluated by the share of correct answers in the deferred sample: 89.04% received correct answers for the Titanic data set, 94.11% for Seeds, 91.66% for Telecom_churn, 90.47% for CreditScoring, 95.94% for Iris, and 93.65% for Wine Quality.

Conclusion. As a result of this work, the effectiveness of the application of ensemble methods based on decision trees, which are built using the CART algorithm, was confirmed. Testing showed the accuracy of the results, more than 90%, which indicates the high quality of the algorithm. It is clear that this quality is obtained through the use of several different decision trees, the results of which are summarized in the derivation of the final assessment.

Keywords: machine learning, classification problem, decision tree, Random Forest.

Одержано редакцією 20.12.2019 р.
Прийнято до публікації 24.02.2020 р.

УДК 378:517

DOI 10.31651/2076-5886-2020-1-77-86

PACS 02.60.-x

ТАРАСЕНКОВА Ніна Анатоліївна,
доктор педагогічних наук, професор,
завідувач кафедри математики та
методики навчання математики,
Черкаський національний університет
імені Богдана Хмельницького
e-mail: ntaras7@ukr.net
ORCID: 0000-0002-6418-6380

СЕРДЮК Зоя Олексіївна,
кандидат педагогічних наук, доцент,
доцент кафедри математики та методики
навчання математики,
Черкаський національний університет
імені Богдана Хмельницького
e-mail: serdyuk_z@ukr.net
ORCID: 0000-0002-9376-4346

**ОСОБЛИВОСТІ ВИКЛАДАННЯ КУРСУ МАТЕМАТИЧНОГО АНАЛІЗУ
ДЛЯ ФАХІВЦІВ З АНАЛІЗУ ДАНИХ**

У статті розглянуто специфіку організації вивчення математичного аналізу студентами – майбутніми фахівцями з аналізу даних з урахуванням особливостей освітнього контенту спеціальності 126 «Інформаційні системи та технології»; зазначено загальні компетентності з даної навчальної дисципліни для зазначеної спеціальності та вказано шляхи їх досягнення, зокрема схарактеризовано особливості застосування схематизації під час вивчення та закріплення теоретичного матеріалу в курсі математичного аналізу студентами спеціальності 126 «Інформаційні системи та технології»; наведено конкретні приклади завдань із застосуванням різноманітних схем.

Ключові слова: математичний аналіз, аналіз даних, схематизація, схеми, студенти-програмісти.

Вступ

Курс математичного аналізу є базовим курсом циклу математичних дисциплін, які вивчають студенти більшості нематематичних спеціальностей класичних, педагогічних, технічних ЗВО. Наприклад, у класичних університетах – це спеціальності «Інформаційні системи та технології», «Прикладна математика», «Програмна інженерія», «Системний аналіз», «Фізика та астрономія», «Прикладна фізика та наноматеріали» тощо. Даний курс спрямований не лише на засвоєння студентами основних математичних понять і фактів диференціального та інтегрального числення функцій однієї чи кількох змінних та їх застосування до дослідження функцій, обчислення довжин кривих, площ поверхонь, моментів інерції та статичних моментів, знаходження кратних і контурних інтегралів тощо, а й на формування в них спроможності доказово і несуперечливо міркувати, аналізувати, порівнювати, узагальнювати тощо, загалом, уміння робити правильні висновки та будувати реалістичні прогнози, застосовувати отримані знання, навички й уміння до розв'язування різноманітних практичних задач. Деякі аспекти вивчення даного курсу висвітлено у працях [1; 2; 3]. Проте специфіка організації вивчення математичного аналізу з урахуванням особливостей освітнього контенту спеціальності 126 «Інформаційні системи та технології» ще не була предметом розгляду і залишається у фокусі наших наукових розвідок.

У Стандарті Вищої Освіти України для галузі знань 12 «Інформаційні технології» спеціальності 126 «Інформаційні системи та технології» першим серед програмних результатів навчання зазначено наступний: «ПР 1. Знати лінійну та векторну алгебру, диференціальне та інтегральне числення, теорію функцій багатьох змінних, теорію рядів, диференціальні рівняння для функції однієї та багатьох змінних, операційне числення, теорію ймовірностей та математичну статистику в обсязі, необхідному для розробки та використання інформаційних систем, технологій та інфокомунікацій, сервісів та інфраструктури організації» [4, с. 8].

Цей програмний результат навчання сприяє формуванню у студентів таких загальних компетентностей:

- КЗ 1. Здатність до абстрактного мислення, аналізу та синтезу;
- КЗ 2. Здатність застосовувати знання у практичних ситуаціях;
- КЗ 3. Здатність до розуміння предметної області та професійної діяльності;
- КЗ 6. Здатність до пошуку, оброблення та узагальнення інформації з різних джерел [4].

Під час вивчення математичного аналізу якраз і створюються комфортні умови для ефективного формування у студента зазначених компетентностей. Вивчення курсу математичного аналізу сприяє не лише оволодінню студентами основними теоретичними знаннями, практичними навичками й уміннями, а й забезпечує досягнення такого рівня засвоєння матеріалу, який був би достатнім для їхньої професійної діяльності.

Забезпечення належного рівня математичної освіти набуває на сучасному етапі розвитку суспільства особливого значення. Математична компетентність для фахівців різних напрямів, зокрема інженерів-програмістів – основне підґрунтя для засвоєння фахових дисциплін і майбутньої професійної діяльності. У ході вивчення математичного аналізу закладають не тільки методологічний, а й психофізіологічний фундамент системного, логічного та критичного мислення, що є життєво необхідним.

Мета статті – розглянути особливості організації вивчення курсу математичного аналізу студентами спеціальності 126 «Інформаційні системи та технології» з опорою на схематизацію як специфічну знаково-символічну діяльність.

Виклад основного матеріалу

Аналіз даних – розділ математики, що займається розробкою методів обробки даних незалежно від їх природи. Аналіз даних включає виконання послідовних, логічних дій з інтерпретації зібраних даних (наприклад, відповідей респондентів) та їх перетворення на статистичні форми, потрібні для ухвалення маркетингових та керівницьких рішень.

Можна виділити такі етапи аналізу даних: отримання даних, обробка, аналіз та інтерпретація результатів обробки.

Останнім часом значного поширення набувають нові технології і методи аналізу даних, зокрема методи інтелектуального аналізу даних (Data Mining), які використовують для виявлення прихованих закономірностей у великих масивах даних, та нейроінформатики, а також методи й засоби статистичного контролю за якістю на виробництві та в управлінні організаціями. Основні процедури аналізу даних найчастіше реалізують за допомогою сучасних комп'ютерних технологій [5].

Аналіз даних по суті є прикладним розділом математичної статистики, а як відомо, то більшість фактів, формул, теорем математичної статистики базуються на теоретичних знаннях та практичних навичках з математичного аналізу. Тому під час підготовки майбутніх інженерів-програмістів – студентів спеціальності 126 «Інформаційні системи та технології», – важливо, щоб вони отримали міцну теоретичну базу та набули практичних навичок з тих чи тих тем математичного аналізу, які будуть необхідні їм у подальшому професійному становленні, зокрема: «Числові ряди», «Степеневі ряди», «Інтегральне числення функції однієї змінної», «Інтегральне числення функцій кількох змінних» тощо.

На жаль, за останні роки кількість годин, що відводиться на вивчення даного курсу значно зменшилося, проте вимоги до змістового наповнення суттєво не змінилися. Тому доцільно змінити підхід до структурування змісту курсу та відпрацювання навичок й умінь студентів у нових «складніших» умовах. Наразі структура навчальної дисципліни «Математичний аналіз» для студентів спеціальності 126 «Інформаційні системи та технології» є такою, як показано у таблиці 1.

Оскільки курс математичного аналізу зазвичай насичений великою кількістю досить складних математичних фактів (теорем, формул тощо), які студенти спеціальності 126 «Інформаційні системи та технології» насамперед повинні вміти застосовувати у своїй безпосередній професійній діяльності, то ми маємо за мету не лише їх «бездумне» запам'ятовування, адже тренувати пам'ять можна й іншими способами, а саме вміння їх застосовувати в тій чи тій практичній ситуації та для розв'язання різних прикладних задач. Згідно з нашими спостереженнями, засвоєння та, найголовніше, застосування тих чи тих фактів у процесі розв'язування задач часто зумовлюють появу труднощів у студентів-програмістів. Тому методика їх вивчення має будуватися як специфічна для такої категорії студентів.

Таблиця 1

Структура навчальної дисципліни

Форма навчання	Семест у межах дисципліни	Кількість кредитів	Загальна кількість годин	Аудиторна робота		Самостійна робота		Форма підсумкового контролю
				лекції	практичні	розрахункові роботи	підготовка до занять	
денна	перший	5	150 год	50 год /33 %		100 год /67 %		іспит
				26 год	24 год	30 год	70 год	
денна	другий	5	150 год	50 год /33 %		100 год /67 %		іспит
				26 год	24 год	30 год	70 год	

Ураховуючи останні тенденції освітньої політики та специфіку підготовки фахівців зі спеціальності 126 «Інформаційні системи та технології», а саме – схильність до алгоритмізації у розв’язанні тих чи тих практичних завдань, то доцільним, на нашу думку, буде використання схематизації під час вивчення математичного аналізу вищезазначеними студентами – і під час опанування теоретичного матеріалу, і в ході розв’язування задач.

Схематизація як знаково-символічна діяльність у навчанні математичного аналізу студентів-програмістів посідає значне місце. Як зазначає Н. А. Тарасенкова, її сутність полягає в тому, що навчальне пізнання здійснюється з опорою на певну схему, що відображає структуру реальності мовними чи немовними засобами [6].

У ході ознайомлення студентів з новим матеріалом та під час застосування знань можна виділити кілька різновидів навчальних ситуацій, в яких вони стикаються з діяльністю схематизації. Розглянемо їх детальніше.

Перший випадок організації схематизації ми пов’язуємо з використанням відомих схем діяльності для отримання суб’єктивно нових знань загального характеру. Такі ситуації виникають, наприклад, під час застосування певного методу доведення теорем з курсу математичного аналізу чи окремого прийому встановлення математичного факту тощо. При цьому важливо, щоб студенти не просто слідували за викладом лектора, а просувалися по зарані розставлених орієнтирах [7]. Сукупність таких орієнтирів і виступатиме схемою діяльності студентів.

Не менш важливу роль схематизація може відігравати у розвитку візуального мислення студентів. Для цього потрібно, щоб схема діяльності фіксувалася графічними засобами, істотні зв’язки між елементами схеми були виведені назовні й так, щоб утворювалися можливості декодувати інформацію поза її вербалізації.

До наступного виду схематизації ми відносимо використання плану викладу змісту певної навчальної теми. При цьому схеми діяльності можуть подаватися учням як у розгорнутій словесній формі, так і за допомогою немовних знаково-символічних засобів [6].

Наприклад, план викладу може бути пред’явлений студентам у формі опорного конспекту. Так, під час введення означення визначеного інтеграла доцільно скористатися наступним покроковим планом-конспектом.

План-конспект

Нехай функція $y = f(x)$ визначена на відрізку $[a; b], a < b$ (рис. 1).

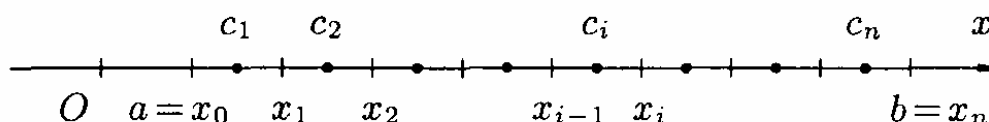


Рис. 1

Виконаємо наступні дії.

1. За допомогою точок $x_0 = a, x_1, x_2, \dots, x_n = b$ ($x_0 < x_1 < \dots < x_n$) **розіб'ємо** відрізок $[a, b]$ на n **частинних** відрізків $[x_0; x_1], [x_1; x_2], \dots, [x_{n-1}; x_n]$.
2. У кожному частинному відрізку $[x_{i-1}; x_i]$, $i = 1, 2, \dots, n$ **виберемо довільну точку** $c_i \in [x_{i-1}; x_i]$ й обчислимо значення функції в ній, тобто знайдемо $f(c_i)$.
3. **Помножимо** знайдене значення функції $f(c_i)$ на довжину відповідного частинного відрізка $\Delta x_i = x_i - x_{i-1}$, тобто $f(c_i) \cdot \Delta x_i$.
4. **Складемо суму** S_n усіх можливих добутоків:

$$S_n = f(c_1)\Delta x_1 + f(c_2)\Delta x_2 + \dots + f(c_n)\Delta x_n = \sum_{i=1}^n f(c_i)\Delta x_i. \quad (1)$$

Сума S_n виду (1) називається **інтегральною сумою** функції $y = f(x)$ на відрізку $[a; b]$.

5. **Знайдемо границю інтегральної суми** S_n (1) за умови, що $n \rightarrow \infty$ так, щоб $\lambda \rightarrow 0$, де λ – довжина найбільшого частинного відрізка, тобто $\lambda = \max \Delta x_i$ ($i = 1, 2, \dots, n$).
6. Якщо при цьому інтегральна сума S_n має границю, яка не залежить ні від способу розбиття відрізка $[a; b]$ на частинні відрізки, ні від вибору точок в них, то вона називається **визначеним інтегралом** від функції $y = f(x)$ на відрізку $[a; b]$ і позначається

$$\int_a^b f(x)dx = \lim_{\substack{n \rightarrow \infty \\ (\lambda \rightarrow 0)}} \sum_{i=1}^n f(c_i)\Delta x_i \quad (2)$$

Якщо опорний конспект формувати разом зі студентами протягом вивчення нової теми, зокрема визначеного інтеграла, й відновити у процесі актуалізації знань, тоді цей опорний конспект цілком може використовуватися в якості плану вивчення матеріалу і в подальшому, а саме – під час введення й засвоєння понять інших видів інтегралів: подвійного, потрійного, криволінійного 1 роду, криволінійного 2 роду, поверхневого 1 роду тощо. Поняття кратних, криволінійних та поверхневих інтегралів вводяться вже в другому семестрі вивчення курсу математичного аналізу, коли студенти є більш підготовленими і можуть брати активну участь у складанні плану-конспекту введення означень даних понять, а деякі з них можуть робити це й самостійно. Доцільно в плані-конспекті виділити *слова (фрази)-індикатори*, щоб студентам було легше засвоїти вищезазначені математичні поняття. У наведеному нами плані-конспекті ці індикатори виділені жирним курсивом. Для інших понять зазначені індикатори будуть такими ж і допоможуть студентам у кращому засвоєнні даних означень.

Більш того, результати наших спостережень показали, що така форма введення нових понять студентам спеціальності 126 «Інформаційні системи та технології» у

вигляді плану-конспекту виявилася найбільш ефективною, тобто відтворити означення даного поняття студентам саме у запропонованому вигляді найлегше.

Крім планів-конспектів під час вивчення нового матеріалу доцільно пропонувати студентам також короткі схеми, у яких зазначено основні формули чи випадки того чи того математичного факту.

Наприклад, під час пояснення теми «Частинні похідні вищих порядків функції кількох змінних» варто запропонувати студентам наступну схему (рис. 2).

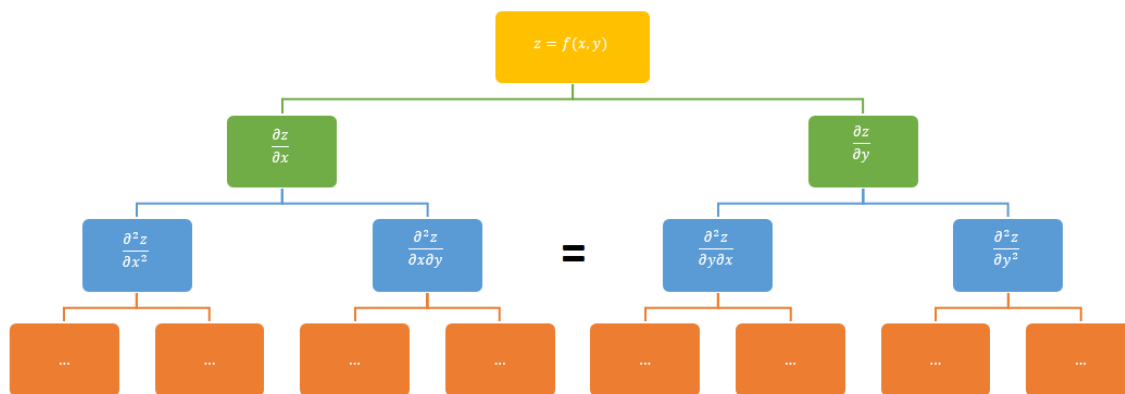


Рис. 2

Як показали наші спостереження, за допомогою саме такої схеми студентам набагато легше запам'ятати послідовність (алгоритм) відшукування похідних вищих порядків на прикладі функції двох змінних, а також, що є особливо важким для запам'ятовування – форма запису частинних похідних та спосіб їх відшукування відповідно до візуального образу.

Наприклад, для закріплення даного матеріалу доцільно запропонувати студентам наступне завдання.

Завдання 1. Довести, що $\frac{\partial^2 z}{\partial x \partial y} = \frac{\partial^2 z}{\partial y \partial x}$, якщо $z = e^x \cos y + e^y \cos x$.

Розв'язання будемо проводити за наступною схемою:

1. Знаходимо $\frac{\partial z}{\partial x} = e^x \cos y - e^y \sin x$.
2. Знаходимо $\frac{\partial^2 z}{\partial x \partial y} = -e^x \sin y - e^y \sin x$.
3. Знаходимо $\frac{\partial z}{\partial y} = -e^x \sin y + e^y \cos x$.
4. Знаходимо $\frac{\partial^2 z}{\partial y \partial x} = -e^x \sin y - e^y \sin x$.
5. Звідси маємо, що $\frac{\partial^2 z}{\partial x \partial y} = \frac{\partial^2 z}{\partial y \partial x}$, що й треба було довести.

Наведемо ще один приклад використання схем. Під час вивчення методів інтегрування, а саме методу інтегрування частинами, у студентів особливі утруднення виникають під час застосування цього методу в розв'язуванні завдань, наприклад, коли потрібно правильно вибрати функцію для заміни через змінну u . Студенти зазвичай роблять це несвідомо, або ж методом підбору, що не є правильним. І через певний час, або ж під час застосування в інших завданнях, студенти не можуть пригадати, яку саме

заміну треба зробити, тобто не можуть відтворити отримані знання в іншій ситуації. Тому для кращого засвоєння пропонуємо наступну схему (рис. 3).

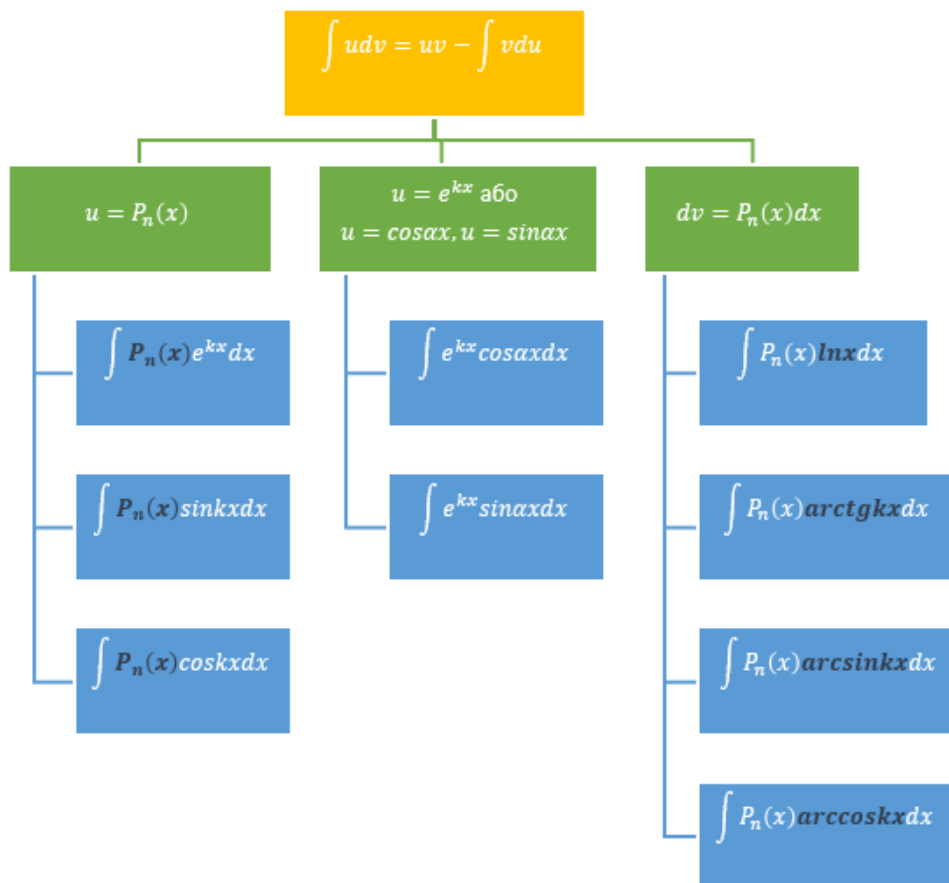


Рис. 3

Розв’язування задач алгоритмічного чи евристичного характеру є другим випадком навчальних ситуацій, коли діяльність студентів є схематизацією. Тут, використовуючи відомі схеми, студенти здебільшого отримують математичні результати окремого характеру.

Під час вивчення теми «Невласні інтеграли», наприклад, невластні інтеграли першого роду розв’язуються на основі засвоєних раніше операцій інтегрування та граничного переходу. Відповідні правила для студентів є об’єктами засвоєння. Отже, розв’язуючи такі завдання з опорою на відповідні правила, студенти виконують діяльність схематизації. Розглянемо етапи розв’язування, наступного завдання.

Завдання 2. Обчислити невластний інтеграл або встановити його розбіжність: $\int_1^{+\infty} \frac{dx}{x^2}$

Під час введення поняття невластного інтеграла першого роду ми запропонували студентам наступну схему (рис. 4) для того, щоб вони могли краще розібратися, як саме застосувати дане означення до розв’язування конкретних завдань.

Розв’язування завдання 2 доцільно супроводжувати такими роз’ясненнями:

1) перепишемо даний невластний інтеграл у вигляді границі, згідно з означенням:

$$\int_1^{+\infty} \frac{dx}{x^2} = \lim_{t \rightarrow +\infty} \int_1^t \frac{dx}{x^2};$$

2) обчислимо визначений інтеграл, користуючись методом безпосереднього інтегрування (за допомогою таблиці інтегралів): $\int_1^t \frac{dx}{x^2} = \int_1^t x^{-2} dx = -\frac{1}{x} \Big|_1^t = -\left(\frac{1}{t} - 1\right) = -\frac{1}{t} + 1$;

3) наступним кроком обчислимо границю отриманого виразу, враховуючи, що $t \rightarrow \infty$: $\lim_{t \rightarrow \infty} \left(-\frac{1}{t} + 1\right) = 0 + 1 = 1$;

4) на основі отриманого значення границі (у нас це 1) робимо висновок про те, що даний невластний інтеграл є збіжним і дорівнює 1.

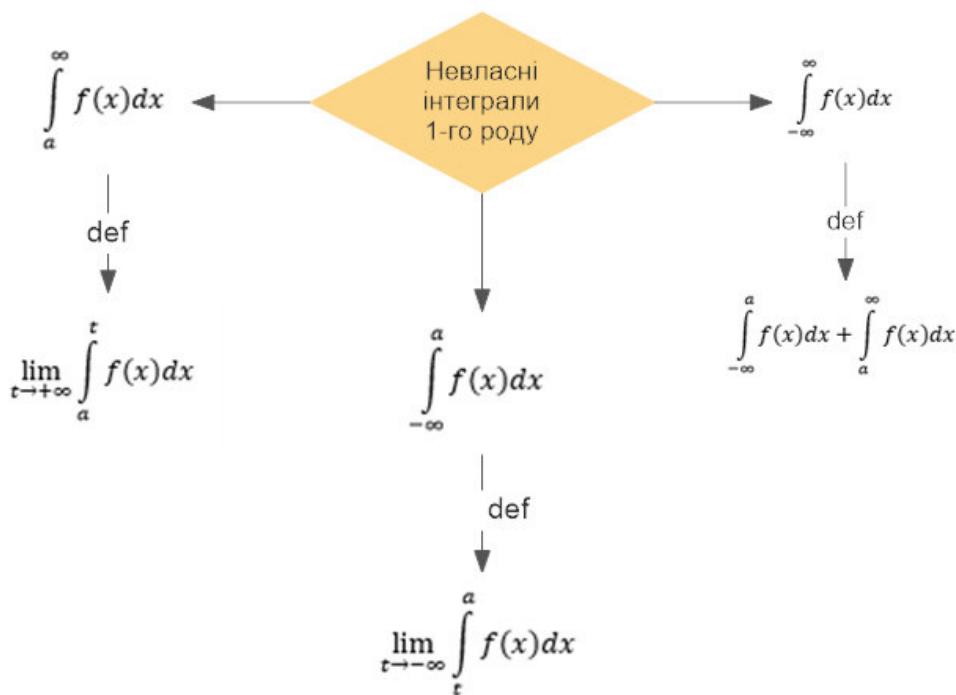


Рис. 4

У наведеному прикладі навчальної ситуації відома студентам схема діяльності представлена у словесній формі – через пояснення і обґрунтування.

Під час вивчення курсу математичного аналізу доцільно акцентувати увагу студентів на застосуванні отриманих ними теоретичних основ дисципліни, а саме – розв’язування задач. Достатньо посутню частку всіх розглядуваних задач складають задачі з геометричним та фізичним (чи то механічним) змістом. Їх розв’язування викликає у студентів значні утруднення, оскільки тут перетинаються дві складові – якісна та кількісна (правильний вибір і вміння застосовувати ті чи ті математичні формули). Тому спочатку доцільно добре відпрацювати саме вміння студентів працювати з математичними формулами.

Висновки. Подальші дослідження ми вбачаємо у розробці компетентісно орієнтованої потемної системи завдань з математичного аналізу з урахуванням специфіки підготовки майбутніх фахівців з аналізу даних.

Список використаної літератури:

1. Сердюк З. О. Математичний тезаурус як інтелектуальний засіб навчання студентів фізичних спеціальностей ВНЗ / З. О. Сердюк, Т. М. Христенко // Засоби і технології сучасного навчального середовища // Матеріали всеукраїнської науково-практичної конференції (20-21 травня 2011 року, м. Кіровоград). – С. 78-79.
2. Сердюк З. О. Особливості вивчення навчальної дисципліни «Математичний аналіз» для студентів

- фізичних спеціальностей ВНЗ / З. О. Сердюк // Актуальні проблеми і перспективи дидактики фізики // Матеріали Всеукраїнської науково-практичної конференції (26-28 квітня 2012 року, м. Черкаси). – Черкаси, ЧНУ ім. Б. Хмельницького. – С. 51-52.
3. Сердюк З. О. Реалізація компетентнісного підходу під час вивчення курсу математичного аналізу в ВНЗ. – Вісник Черкаського університету, Випуск № 8 (341): серія «Педагогічні науки». – Черкаси: Вид. від. ЧНУ ім. Б. Хмельницького, 2015. – С. 101-106.
 4. Стандарт вищої освіти України: перший (бакалаврський) рівень, галузь знань 12 – Інформаційні технології, спеціальність 126 – Інформаційні системи та технології / [Електронний ресурс]. – Режим доступу: <https://mon.gov.ua/ua/npa/pro-zatverdzhennya-standartu-vishoyi-osviti-za-specialnistyu-126-informacijni-sistemi-i-tehnologiyi-dlya-pershogo-bakalavrskogo-rivnya-vishoyi-osviti>.
 5. Бахрушин В. Є. Методи аналізу даних : навчальний посібник для студентів / В.Є. Бахрушин. – Запоріжжя : КПУ, 2011. – 268 с.
 6. Тарасенкова Н. А. Використання знаково-символічних засобів у навчанні математики : [монографія] / Н. А. Тарасенкова. – Черкаси : Відлуння-плюс, 2002. – 400 с.
 7. Талызина Н. Ф. Педагогическая психология / Н. Ф. Талызина. – М. : Академия, 1999. – 288 с.

Bibliography:

1. Serdiuk, Z. (2011). Mathematical thesaurus as an intellectual means of teaching students of physical specialties of higher education. Means and technologies of modern educational environment. Proceedings of the All-Ukrainian scientific-practical conference, 78-79. [in Ukrainian]
2. Serdiuk, Z. (2012). Peculiarities of studying the discipline "Mathematical analysis" for students of physical specialties of higher education. Actual problems and prospects of didactics of physics. Proceedings of the All-Ukrainian scientific-practical conference, 51-52. [in Ukrainian]
3. Serdiuk, Z. (2015). Implementation of the competence approach during the study of the course of mathematical analysis in higher education. Bulletin of Cherkasy University, 8 (341): series "Pedagogical Sciences", 101-106. [in Ukrainian]
4. Standart of higher education of Ukraine: first (bachelor's) level, branch of knowledge 12 Information technologies, specialty 126 Information systems and technologies, (2020). Retrieved from <https://mon.gov.ua/ua/npa/pro-zatverdzhennya-standartu-vishoyi-osviti-za-specialnistyu-126-informacijni-sistemi-i-tehnologiyi-dlya-pershogo-bakalavrskogo-rivnya-vishoyi-osviti> [in Ukrainian]
5. Bakhrushin, V. (2011). Methods of data analysis: a textbook for students. [in Ukrainian]
6. Tarasenkova, N. (2002). The use of sign-symbolic means in teaching mathematics: [monograph] [in Ukrainian]
7. Talyzina, N. (1999). Pedagogical psychology. M.: Akademiya. [in Russian]

TARASENKOVA Nina,

Doctor of Science (Pedagogical Sciences), Professor, Head of the Department of Mathematics and Methods of Learning of Mathematics, Cherkasy Bohdan Khmelnytsky National University, Ukraine

SERDIUK Zoia,

PhD (Pedagogical Sciences), Associate Professor of the Department of Mathematics and Methods of Learning of Mathematics, Cherkasy Bohdan Khmelnytsky National University, Ukraine

PECULIARITIES OF TEACHING THE COURSE OF MATHEMATICAL ANALYSIS FOR SPECIALISTS IN DATA MINING

Summary. Introduction. *The course is aimed not only at mastering basic mathematical concepts and facts of differential and integral calculus of functions of one or more variables and their application to the study of functions, calculation of curve lengths, surface areas, moments of inertia and static moments, finding multiple and contour integrals, etc., but also to form in them the ability to reasonably and consistently reason, analyze, compare, summarize, etc., in general, the ability to draw correct conclusions and make realistic predictions, apply the acquired knowledge, skills and abilities to solve various practical problems. However, the specifics of the organization of the study of mathematical analysis, taking into account the peculiarities of the educational content of the specialty 126 «Information Systems and Technologies» has not yet been considered and remains the focus of our scientific research.*

Purpose – *to consider features of the organization of studying of a course of the mathematical analysis by students of a specialty 126 «Information systems and technologies» by means of schematization.*

Originality. *In this paper, given the latest trends in educational policy and the specifics of training specialists in the specialty 126 «Information Systems and Technologies», namely – the*

tendency to algorithmization in solving certain practical problems, it is advisable to use schematization when studying mathematical analysis by the above students – and during mastering the theoretical material, and in solving problems. Schematization as a sign-symbolic activity in the teaching of mathematical analysis to students-programmers occupies a significant place. During the acquaintance of students with the new material and during the application of knowledge, several types of learning situations are identified in which they encounter the activities of schematization.

Conclusion. We see further research in the development of a competency-based dark system of tasks in mathematical analysis, taking into account the specifics of training future professionals in data analysis.

Keywords: mathematical analysis, data mining, schematization, schemes, students-programmers.

Одержано редакцією 03.12.2019 р.
Прийнято до публікації 24.02.2020 р.

УДК 004.85:519.6

DOI 10.31651/2076-5886-2020-1-86-100

PACS 02.70.Wz, 07.05.Kf, 07.05.Mh,
07.05.Tr

КОВАЛЕНКО Олена Сергіївна,
магістрантка спеціальності «Прикладна
математика» Черкаського національного
університету імені Богдана
Хмельницького

СЕРДЮК Олександр Анатолійович,
кандидат економічних наук, старший
викладач кафедри прикладної математики
та інформатики Черкаського
національного університету імені Богдана
Хмельницького
e-mail: serdyuk@ukr.net
ORCID 0000-0002-3919-4661

ВИКОРИСТАННЯ КЛАСИЧНИХ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ТЕКСТІВ У ПРОГРАМАХ ГЕНЕРАЦІЇ АВТОМАТИЧНИХ ВІДПОВІДЕЙ

У статті подано опис методів машинного навчання для розв'язання задач класифікації текстів з метою їх подальшого використання у програмах автоматичної генерації відповідей на основі аналізу контексту питань користувача. Розглянуто кроки підготовки та проведення класифікації текстів разом з фрагментами програмного коду мовою Python з використанням бібліотеки обробки природної мови. Проведено попередній огляд автоматичних генераторів відповідей та вибрано метод автоматичної генерації відповідей. Розроблено структуру програми автоматичної генерації відповідей та визначено напрямки подальшої роботи.

Ключові слова: машинне навчання, класифікація текстів, чат-боти, програмування, Python.

Вступ

Методи машинного навчання наразі широко використовуються для задач обробки природної, чи людської, мови (Natural Language Processing) з метою її розпізнавання, розуміння, інтерпретації та генерації комп'ютерними алгоритмами. У сучасному світі інформаційних технологій постійно збільшується спектр задач, що можуть використовувати результати інтерпретації людської мови, зокрема це автоматична класифікація текстів для класифікаторів електронних бібліотек, рубрикація новин,

розпізнавання питань користувача у автоматичних будинках та генерація відповідей тощо. Широке розповсюдження месенджерів для передачі повідомлень між користувачами разом з постійним збільшенням кола користувачів відкрило ще один напрямок застосування задач класифікації текстів: розробка автоматичних помічників для генерації повідомлень. Для розв'язання такої задачі необхідно не лише вміти обробляти природну мову, а й генерувати адекватні відповіді, чи, загалом – речення для діалогу. Саме на огляд методів класифікації текстів для їх наступного використання при генерації автоматичних відповідей спрямована стаття.

Метою статті є дослідження методів та алгоритмів, а також демонстрація програмних фрагментів роботи класичних методів машинного навчання для класифікації текстів і застосування результатів класифікації у програмах автоматичної генерації відповідей.

Подальший виклад матеріалу має наступну структуру:

1. Огляд умов відбору даних.
2. Огляд методів попередньої обробки тексту.
3. Огляд методів векторизації тексту.
4. Огляд деяких алгоритмів класифікації тексту.
5. Ознайомлення з поняттям автоматичної генерації відповідей.
6. Розробка структури програми автоматичної генерації відповідей на основі попередньої класифікації текстів.

Виклад основного матеріалу

Відбір даних та вимоги до них

Набір даних, його обсяг та якість у машинному навчанні мають найбільше значення, адже саме від цього залежить результат тренування будь-якої моделі. Потрібний обсяг зразків даних залежить, перш за все, від типу задачі, яку розв'язує дослідник: так, вважається, що прості, класичні моделі машинного навчання, наприклад, лінійної регресії, не потребують так багато даних, як штучні нейронні мережі. Однак, цього не можна сказати про якість даних, адже те, наскільки добре вони структуровані й розмічені, буде вирішальним фактором у роботі моделі незалежно від її складності. Навіть звичайна лінійна регресія може дати дуже гарні результати, що нам доводить Google зі своїми результатами у тренуванні “розумних моделей” [15].

Нами для роботи було використано стандартний набір даних під назвою 20 newsgroups text dataset з бібліотеки Scikit-learn, що включає 18000 різних текстів, розподілених за 20 темами [14].

Вибір середовища розробки та його налаштування

Для роботи з попередньою обробкою, векторизацією та класифікацією текстів було використано пакет бібліотек і програм для символної та статичної обробки природної мови – NLTK, доступної для роботи на операційній системі Windows [10].

У якості середовища розробки було обрано інтерактивне середовище Jupyter Notebook, що працює у браузері. Цей інструмент зручний у використанні для роботи з даними, статистичного моделювання та машинного навчання [12].

Для того, щоб запустити Jupyter Notebook у операційній системі Windows, спочатку було встановлено дистрибутив з відкритим кодом Anaconda (рис. 1), що містить у собі набір вільних бібліотек та найбільш популярних модулів для мови Python, зокрема, NumPy, SciPy, Astropy та ін. [2].

Попередня обробка текстів

Роботі з кластеризацією чи класифікацією текстів зазвичай не обходиться без їх попередньої обробки, адже для адекватного сприйняття даних моделлю потрібно спочатку очистити їх від небажаних символів та слів, тегів, повернути словам їх початкову форму, тобто лематизувати, видалити небажані закінчення, перетворити всі літери до нижнього регістру, прибрати цифри, пробіли, ASCII-символи тощо.

На поточний момент існує багато бібліотек для природної обробки мови (Natural Language Processing), які вже мають у собі весь вищеперахований набір функцій, наприклад SpaCy, gensim, NLTK. У якості альтернативного варіанту обробку можна виконати самостійно, звертаючись до певних функцій, класів, методів бібліотеки Scikit-learn чи інших популярних бібліотек для мови Python.

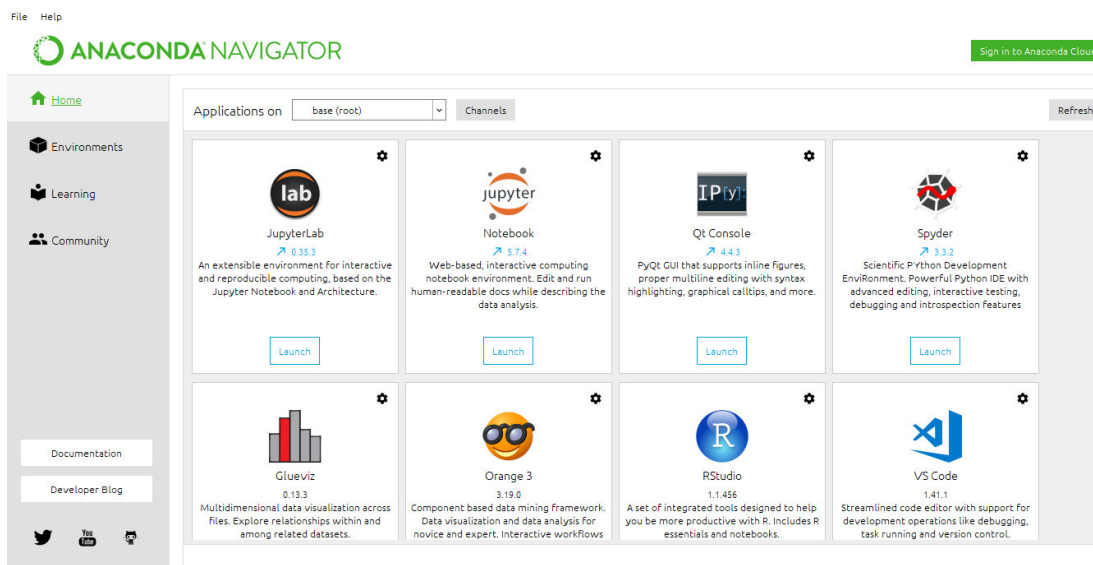


Рис. 1. Початкова сторінка інструментального середовища Anaconda

Підготовка корпусу

Корпус тексту – це великі структуровані набори текстів, подані у електронному вигляді. Корпус тексту може бути одномовним або багатомовним. Існує кілька найрозповсюдженіших видів структури корпусу текстів: найпростіша з них – це ізольовані тексти, що не мають якоїсь визначеної організації, а представляють собою лише набір текстів. Наступним видом корпусу текстів є тексти, зібрані по категоріях за жанрами, джерелом, авторами, мовою і т. ін. Ще одним видом структури текстів є тексти, де присвоєні їм категорії можуть перемижуватись: наприклад, у випадку з визначенням теми тексту один і той же текст може належати до кількох тем. Іноді структура текстів може змінюватись з часом; зразком такого виду текстів може бути набір новин. Останній вид текстів належить до категорії за вступною адресою (з англ. Inaugural Address). Вище перераховані види текстів проілюстровано на рис. 2.

Для роботи з текстами не обов'язково мати власний корпус: його можна завантажити з бібліотеки NLTK, яка налічує більше, ніж 25000 електронних книг, що знаходяться у вільному доступі з ресурсу Gutenberg [13]. Щоб завантажити їх, потрібно спочатку імпортувати пакет NLTK, а потім скористатися однією з функцій, що дає доступ до корпусу текстів, наприклад `nltk.download('gutenberg')`:

```
import nltk
nltk.download('gutenberg')
```

У разі, якщо необхідно скористатись власним набором текстів, можна зробити це за допомогою `PlaintextCorpusReader` з бібліотеки NLTK, присвоюючи змінній

corpus_root адресу розташування у мережі відповідного набору текстів:

```
from nltk.corpus import PlaintextCorpusReader
corpus_root = '/usr/share/dict'
```

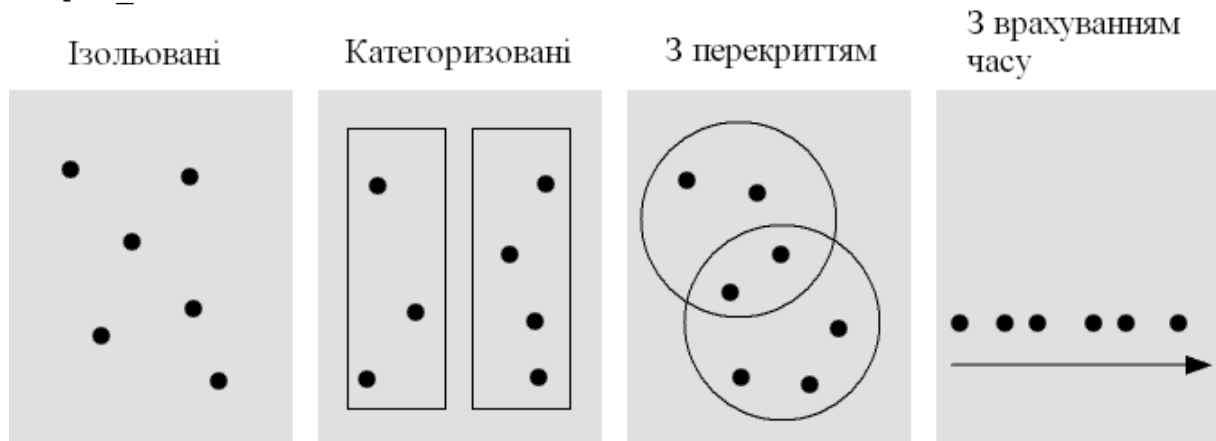


Рис. 2. Види структури корпусу текстів

Видалення пунктуації та тегів

Після завантаження набору текстів проводиться їх попередня обробка з метою підготовки їх до векторизації. Обробка починається з видалення пунктуації, тегів, ASCII-символів:

```
from nltk.tokenize import RegexpTokenizer
tokenizer = RegexpTokenizer(r'\w+')
tokenizer.tokenize()
```

Також на даному етапі усі слова у наборі тексту перетворюються до нижнього регістру, що можна зробити за допомогою наступного виразу:

```
sorted([w for w in set(sent7) if not w.islower()])
```

Видалення стоп-слів

Видалення стоп-слів є досить важливим етапом очищення тексту перед векторизацією, адже більшість займенників, прийменників та артиклів у англійській мові на кшталт “it, his, for, the” не несуть вагомого значення для класифікації текстів за темами, натомість лише заплутуючи алгоритм. Видалити ці слова можна за допомогою наступних рядків:

```
nltk.download(stopwords)
from nltk.tokenize import word_tokenize
example_sent = str([twenty_train])
stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(example_sent)
filtered_sentence = [w for w in word_tokens if not w in stop_words]
filtered_sentence = []
for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)
```

Лематизація

Лематизація означає перетворення слова в його словникову форму, тобто – лему. Цей етап застосовується для того, щоб спростити роботу алгоритму й відкинути видозмінені форми слів, що заважатимуть класифікації і не несуть вагомого значення. Лематизація виконується за наступним прикладом:

```
from nltk.stem import WordNetLemmatizer
```

```
lemmatizer = WordNetLemmatizer(twenty_train)
```

Стематизація

Кінцевим етапом обробки тексту у багатьох випадках є стематизація слова, але існує ще багато способів, якими можна підготувати текст до тренування моделі, що, звісно, залежать від мети наукового дослідження чи практичного завдання. До них, зокрема, належать Part-of-Speech tagging, Named-entity recognition та інші.

Під стематизацією розуміють процес знаходження основи кожного слова, тобто, відсікання закінчень та суфіксів слова, залишаючи лише його корінь:

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
ps = PorterStemmer()
words = ["program", "programs", "programer",
         "programing", "programers"]
```

Векторизація текстів

Необхідною умовою того, щоб комп'ютер міг працювати з текстами, що складаються зі слів природної мови, є їх векторизація. Векторизація – процес перетворення слів у цифрові вектори. Для цього використовуються спеціальні моделі, найбільш популярними з яких є: Bag of Words, TF/IDF, Word2Vec, Sent2Vec та інші. Нижче подано два типи векторизації текстів, кожен з яких має свої переваги та недоліки.

Bag of words

У перекладі з англійської Bag of words означає «Мішок слів», що ґрунтується на головній ознаці цього методу: порядок слів у цій моделі векторизації не береться до уваги, і головною ознакою, що визначатиме вагу слова, є кількість появ слова у корпусі текстів. На основі корпусу текстів складається його словник, що містить усі унікальні слова, наявні в усіх текстах.

Наприклад, корпус нараховує загалом 24 слова, а після вилучення повторюваних слів його словник матиме 10 унікальних слів. Оскільки відома кількість унікальних слів, що входять до словника, вектор кожного речення у наборі текстів матиме 10 позицій, де кожна позиція буде нулем або одиницею, що відповідатиме наявності або відсутності кожного слова зі словника у реченні:

```
1 "it was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
2 "it was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
3 "it was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]
```

Якщо корпус текстів складається з десятків книг художньої літератури, то словник такого корпусу може містити мільйони слів і вектор кожного речення, таким чином, може мати до мільйона координат з нулями та одиницями, де нулі, очевидно, займатимуть майже всі позиції.

Для того, щоб застосувати метод Bag of Words до певного корпусу попередньо обробленого тексту, можна використовувати наступний фрагмент коду:

```
for sentence in allsentences:
    words = word_extraction(sentence)
    bag_vector = numpy.zeros(len(vocab))
    for w in words:
        for i,word in enumerate(vocab):
```

```
if word == w:
    bag_vector[i] += 1
```

TF/IDF-векторизація

Метод TF/IDF-векторизації складається з двох термінів, які визначають її роботу: TF (Term Frequency) та IDF (Inverse Document Frequency), що допомагає визначати вагу слова у всьому наборі даних.

Term Frequency визначає те, наскільки часто слово з'являється у документі чи корпусі текстів. Оскільки кожне речення має різну довжину, це є ймовірністю появи слова у довгих реченнях порівняно з короткими.

Формула розрахунку частоти слів наступна:

$$TF = \frac{\text{кількість появ слова у документі}}{\text{загальна кількість слів у документі}}$$

Inverse Term Frequency використовується для розрахунку вагомості слова. Поняття базується на думці про те, що слова, які рідше зустрічаються, є більш важливими. Формула для визначення IDF має вигляд:

$$IDF = \log_{10} \frac{\text{кількість документів}}{\text{кількість документів з заданим словом}}$$

Таким чином, якщо загалом є, наприклад, 10 документів, а якесь слово зустрічається у даному наборі документів лише 2 рази, вагомість такого слова розраховується як $10/2 = 5$. Такий розрахунок дозволяє додавати ваги словам, які зустрічаються рідко, на протипагу тим, що наявні майже у кожному документі [8].

Отримані значення TF та IDF перемножуються для кожного слова і результат використовується у подальшій роботі.

Завантажити TF/IDF-векторизатор з бібліотеки Scikit-learn можна наступним чином:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tf = TfidfVectorizer()
text_tf = tf.fit_transform(data['Phrase'])
```

Класифікація текстів з учителем

Класифікація текстів – це процес визначення відповідного класу для тексту, який передається класифікатору. Якщо розглядати базові задачі класифікації, тобто, класифікацію з учителем, то тут кожен вихідний текст сприймається класифікатором незалежно від усіх інших текстів, адже ці тексти попередньо були розмічені [11]. Серед деяких задач класифікації можна виділити наступні:

- визначити тип повідомлення у електронному листі: спам, чи ні;
- визначити тему статті з новинами, вибравши її з попередньо визначеного набору тем, наприклад: “спорт”, “технології”, “політика”;
- визначити, яке з існуючих значень певного слова, наприклад, слова “коса”, мається на увазі у реченні: як назва жіночої зачіски, як сільськогосподарське знаряддя праці чи як місце суші, оточене водою.

Класифікація за допомогою Decision Trees

Дерево рішень – це проста блок-схема, яка вибирає мітки для вхідних значень. Ця блок-схема складається з вузлів рішення, які перевіряють важливість ознак, та вузлів-листіків, які присвоюють мітки. Щоб у блок-схемі вибрати мітку для вхідного значення, робота починається зі стартового вузла розв'язку, відомого як його кореневий вузол.

Цей вузол містить умову, яка перевіряє одну з ознак введеної одиниці даних і вибирає гілку на основі значення цієї ознаки. Слідом за гілкою, яка описує вхідне значення, відбувається перехід до нового вузла розв'язку з новою умовою для перевірки значення вхідної ознаки. Такі дії переходу до нових гілок, вибраних згідно умов кожного вузла, продовжуються, поки не буде досягнуто вузол-лист, що містить мітку для введеного значення. На рис. 3 продемонстровано приклад моделі дерева рішень для задачі з розпізнавання статі за її назвою [13].

Побудоване дерево рішень далі використовується для розмітки нових вхідних значень. Однак, при цьому залишається відкритим питання щодо того, як побудувати дерево рішень, яке змоделює існуючий набір даних. Перед тим, як сконцентруватись на природі алгоритму побудови дерева рішень, потрібно розглянути одну простішу задачу – вибір найкращого остова рішень (з англ. decision stump) для корпусу текстів.

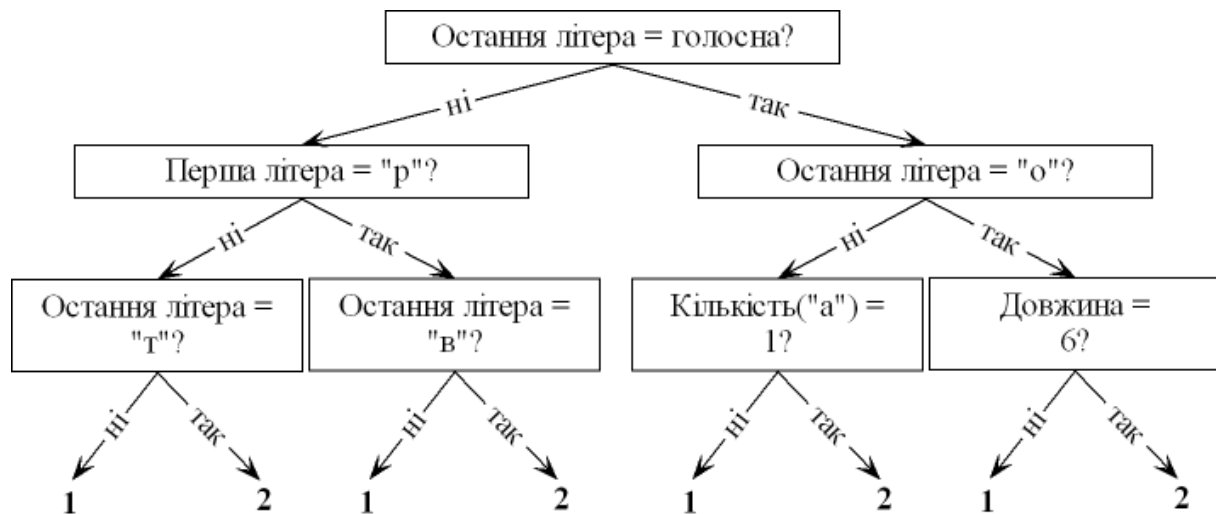


Рис. 3. Приклад дерева рішень

Остів рішень – це дерево рішень, яке має лише один вузол-листок, що розмічає вхідні дані на основі лише однієї ознаки. Загалом остів рішень має один листок для кожного можливого значення ознаки, визначаючи клас, який має бути присвоєно вхідним даним, зважаючи на цю ознаку. Щоб побудувати остів рішень, спершу необхідно визначити, яка ознака має бути використана. Найпростіший шлях – побудувати остів рішень для усіх можливих ознак і простежити, яка з них матиме найбільшу точність на тренувальному наборі даних, після чого можна побудувати остів рішень, присвоюючи мітку кожному листку і опираючись при цьому на те, яка з ознак превалює для вибраних зразків з тренувального набору даних [7].

Після отримання алгоритму для вибору остова рішень, будується алгоритм для дерева рішень. Його робота починається з вибору найкращого остова рішень для задачі класифікації. Потім перевіряється кожен листок дерева на тренувальному наборі даних. Листки, що не досягають достатньої точності, замінюються новими остовами рішень, тренуваними на корпусі текстів, що обирається шляхом до листка. Наприклад, можна побудувати дерево рішень з рис. 3, замінивши листки, що розташовуються зліва, новим остовом рішень, тренуваним на підмножині набору даних, що не починаються з “р” та не закінчуються на голосну.

Програмний код для класифікації за допомогою Decision Trees після попередньої обробки має вигляд:

```

from sklearn import tree
unique_words = []

```

```

for sentence in tagged_tokenized_comments_corpus:
for word in sentence[0]:
    unique_words.append(word)
unique_words = set(unique_words)

dictionary = {}
i = 0
for dict_word in unique_words:
    dictionary.update({i, dict_word})
    i = i + 1
train_target = []
train_data = []
for sentence in tagged_tokenized_comments_corpus:
    train_target.append(sentence[0])
    train_data.append(sentence[1])

clf = tree.DecisionTreeClassifier()
clf.fit(train_data, train_target)

test_data = """Beautiful Keep it up..
                this far is the most usable app editor..
                it makes my photos more beautiful and alive.. """

test_words = tokenizer.tokenize(test_data)
test_tokenized_sentence = []
for test_word in test_words:
    if test_word not in stop_words:
        test_tokenized_sentence.append(
            lemmatizer.lemmatize(test_word.lower()))

print("predicting the labels: ")
print(clf.predict(test_tokenized_sentence))

```

Класифікація за допомогою Naïve Bayes

У класифікації за допомогою Naïve Bayes для кожної ознаки розраховується прогноз того, яку мітку має отримати дане вхідне значення. Щоб вибрати мітку для вхідного значення, класифікатор Naïve Bayes спочатку розраховує попередню ймовірність того, що цьому зразку буде присвоєна певна мітка, що визначається перевіркою того? як часто така мітка зустрічається у тренувальному наборі даних. Потім розраховується вплив кожної ознаки у поєднанні з попередньою (апріорною) ймовірністю, щоб визначити ймовірність для кожної мітки. Мітка, що отримує найбільшу ймовірність, присвоюється вхідному значенню (рис. 4).

Рис. 4 ілюструє процес, використовуваний класифікатором Naïve Bayes, щоб обрати тему для вхідного документа. У тренувальному наборі даних більшість документів стосуються теми “наука”, тому класифікатор визначає свою першу точку розрахунку у місці, наближеному до відповідного кута трикутника. Але потім він починає зважати на значення усіх наявних ознак. Конкретно у поданому прикладі вхідний документ містить слово “райтинг”, що є слабким індикатором наближеності до тем “політика” та “спорт”, але в ньому також наявне і слово “рахунок”, що має велику вагу ознаки, яка відповідає темі “спорт”. Після того як для кожної наявної ознаки розраховане значення, класифікатор перевіряє, для якої ознаки це значення найбільш вагоме, і присвоює цю ознаку вхідному документу.

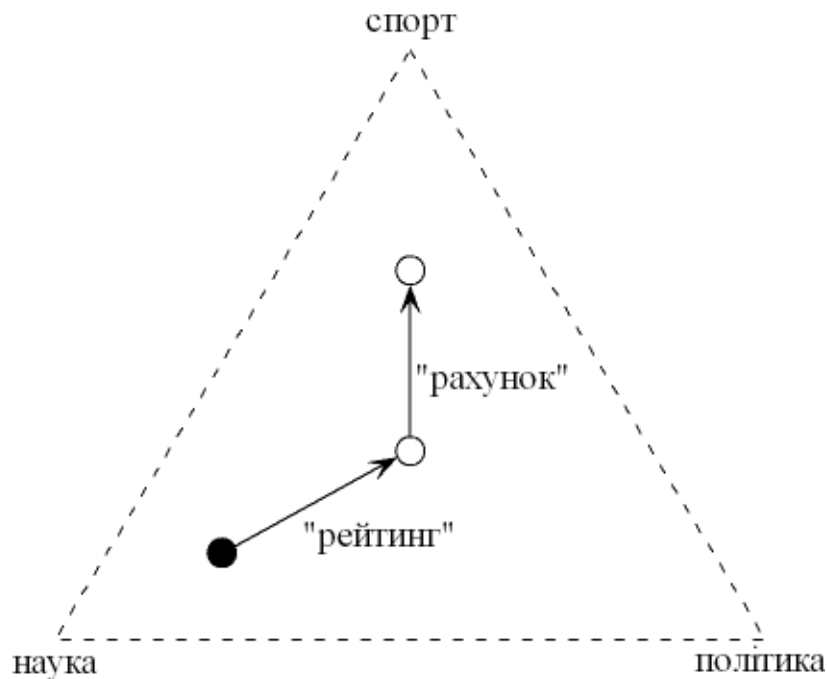


Рис. 4. Приклад оцінки ймовірності мітки

Окремі ознаки впливають на розраховане значення завдяки тому, що “голосують проти” міток, які не дуже часто присвоюються якійсь ознаці у наборі тренувальних даних. Зокрема, якщо слово “рейтинг” зустрічається у 12% документів про спорт, у 10% документів про політику, та лише у 2% документів про науку, то ймовірність того, що документ належатиме до теми “спорт”, множиться на 0.12, до теми про політику – на 0.1, а до теми про науку – лише на 0.02. У результаті ймовірність присвоєння вхідній ознаці теми про спорт буде трохи вищою, аніж для теми про політику, і значно меншою для мітки про науку порівняно з двома попередніми. Кожен етап розрахунку ймовірностей для міток можна прослідкувати на рис. 5.

Програмний код для класифікації за допомогою Naïve Bayes на стандартному наборі текстів з бібліотеки NLTK “Movie reviews” має наступний вигляд:

```
from nltk.corpus import movie_reviews

documents = [(list(movie_reviews.words(fileid)), category)
for category in movie_reviews.categories():
for fileid in movie_reviews.fileids(category)]:
random.shuffle(documents)
all_words = nltk.FreqDist(w.lower())
for w in movie_reviews.words():
word_features = all_words.keys()[:2000]

def document_features(document):
document_words = set(document)
features = {}
for word in word_features:
features['contains(%s)' % word] = (word in document_words)
return features

featuresets = [(document_features(d), c) for (d,c) in documents]
train_set, test_set = featuresets[100:], featuresets[:100]
classifier = nltk.NaiveBayesClassifier.train(train_set)
print 'Accuracy: %4.2f' \
```

```
% nltk.classify.accuracy(classifier, test_set)
```

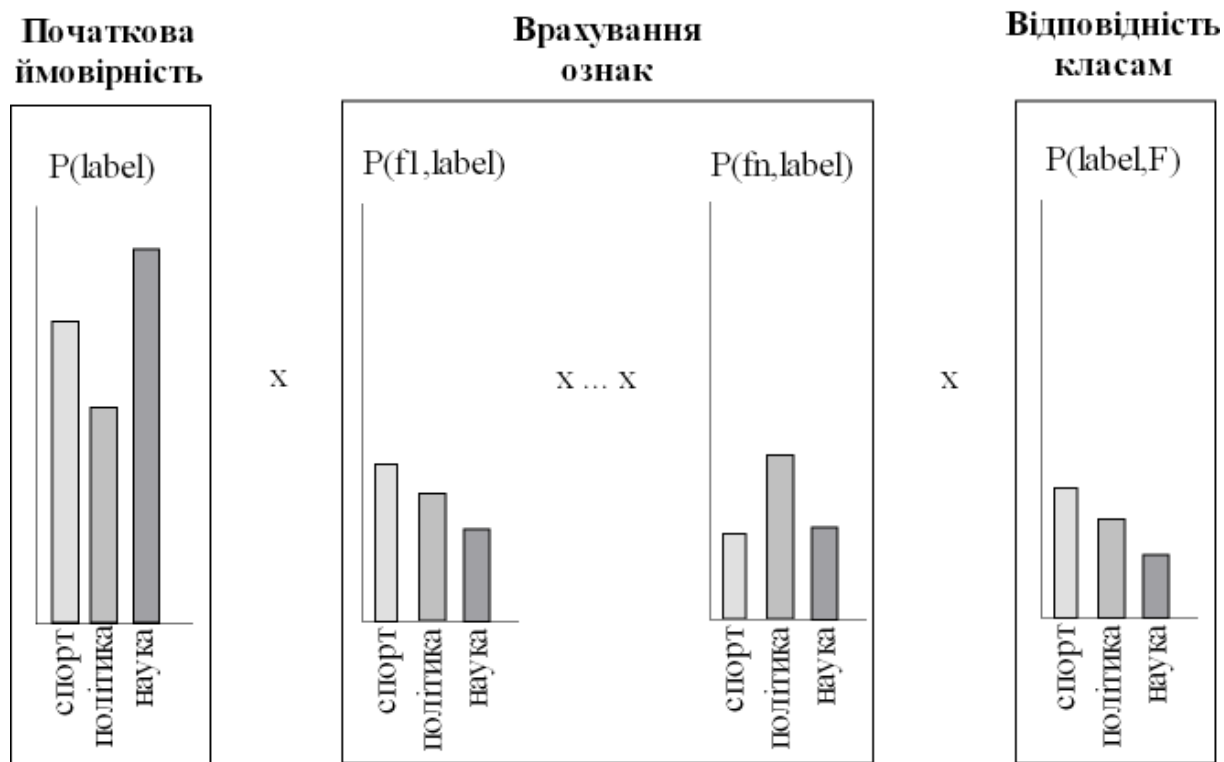


Рис. 5. Розрахунок ймовірностей для кожної мітки

Поняття автоматичного відповідача (чат-бота)

Чат-боти – це такий вид програмного забезпечення, що використовує алгоритми штучного інтелекту та обробки природної мови, щоб зрозуміти, яку команду хоче дати йому людина, і надає їй бажаний результат з найменшою кількістю витраченого часу з боку користувача. Цей вид програмного забезпечення почав набувати популярності з 2016 року, коли Фейсбук відкрив свою розробницьку платформу [5] і показав світу, якого результату можна досягти за допомогою їхнього Messenger App. Трохи пізніше Гугл створив свого голосового асистента. З того часу кількість чат-ботів почала надзвичайно стрімко зростати, вони вбудовувалися у веб-сайти, додатки, соціальні мережі, використовувалися для служби підтримки клієнтів та у багатьох інших застосуваннях.

Чат-боти можна побудувати різними способами. Технологія, що лежить у їх основі, може дещо відрізнятися залежно від використовуваного методу. Загалом, типи чат-ботів, що існують на поточний момент, можна розділити на дві категорії:

- чат-боти на основі правил, тобто ті, що у своїй роботі керуються лише певними правилами і не мають сценарію поведінки, який міг би відрізнятися від наявного набору правил, або на який може впливати контекст розмови з користувачем.
- чат-боти на основі штучного інтелекту, тобто ті, що автоматично донавчаються у процесі роботи після проходження попереднього етапу навчання розробником на основі певного набору даних.

Чат-боти на основі правил працюють за допомогою встановлених для них правил, що прописані розробником, опираючись на певні ключові слова. Такі чат-боти не запрограмовані на зміну повідомлень у відповідності зі змінами у мові та мають

визначену структуру діалогу, у якій користувач отримує відповіді на потрібні йому питання, співвідносячи ввід користувача з підготовленими зарані відповідями.

Існують різні види архітектури чат-ботів, що обирається залежно від основної мети розробки. Є два способи того, як чат-бот може надати відповідь: він може або створити відповідь за допомогою моделей машинного навчання, або використати евристики, щоб обрати потрібний варіант з бібліотеки зарані визначених відповідей.

Генеративна модель архітектури

Генеративна архітектура чат-боту (рис. 6) використовується для розробки розумних ботів, що можуть самі спілкуватися з користувачем, не використовуючи правил. Цей вид чат-ботів дуже рідко використовується, адже потребує застосування комплексу алгоритмів. Генеративні моделі відносно важко побудувати та розробити. Тренування цього типу ботів потребує дуже багато часу та зусиль на збір мільйонів тренувальних даних, адже саме за допомогою даних модель глибокого навчання може ефективно спілкуватись з користувачем. Однак, незважаючи на складність використовуваних алгоритмів, достовірність відповідей, які генерує модель, не є стовідсотковою [3].

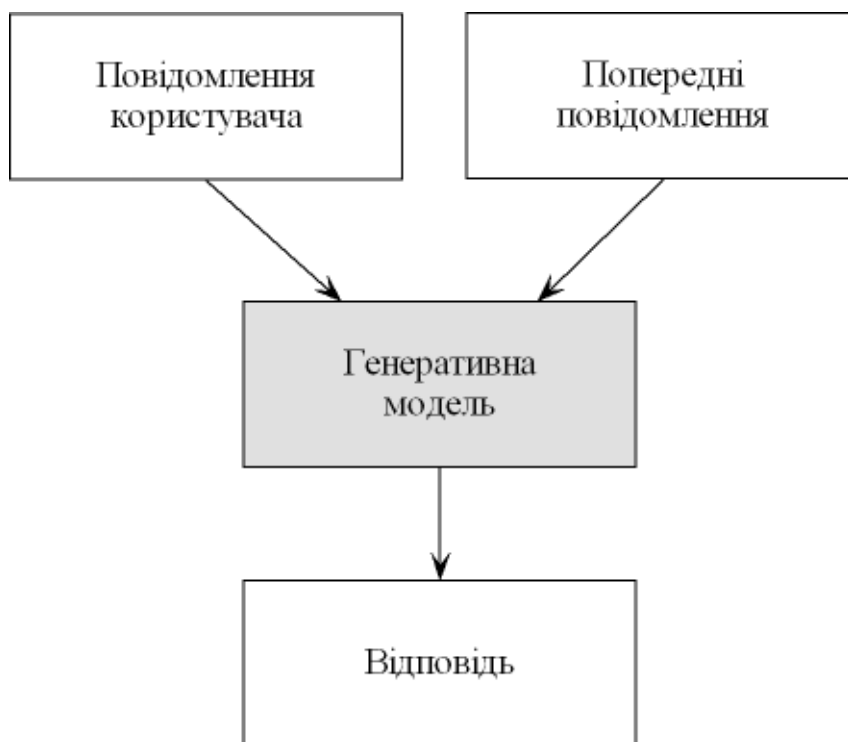


Рис. 6. Генеративна архітектура чат-боту

Моделі на основі пошуку (Retrieval-based)

Цей тип архітектури моделі чат-боту легше побудувати, ніж генеративний, а також він набагато надійніший. Хоча не можна бути стовідсотково впевненим у достовірності відповідей такого чат-боту, можливі типи його відповідей зарані відомі, за рахунок чого можна забезпечити відповідь чат-боту за потрібною темою [4].

Моделі на основі пошуку (рис. 7) є найбільш популярними на даний момент. Наразі розробникам доступні кілька алгоритмів та API, щоб побудувати чат-бота на основі цієї моделі архітектури. Такий чат-бот розпізнає повідомлення та його контекст, щоб вибрати найкращу відповідь з-поміж списку зарані визначених повідомлень.

Застосування попередньої обробки та класифікації текстів для побудови автоматичних відповідачів

Найпростіший варіант генерації відповіді чат-бота використовує if-else умови чи навчання машинних класифікаторів для визначення набору правил із заздалегідь визначеними шаблонами, які виступають умовою для оформлених правил. Найбільшою популярністю для формування таких шаблонів та відповіді у процесі розробки чат-боту використовується мова розмітки Artificial Intelligence Markup Language (AIML) [1, 9].

З правильно налаштованою обробкою тексту за допомогою NLP та попередньо визначеними добре продуманими шаблонами, AIML можна використати для побудови розумних чат-ботів. Такі боти роблять аналіз повідомлення користувача, знаходять у ньому синоніми та теми, розмічають його по членам речення та визначають, який з множини попередньо заданих шаблонів підходить для запиту користувача. Однак, ці боти не використовують алгоритми машинного навчання або будь-які інші API за замовчуванням [5].

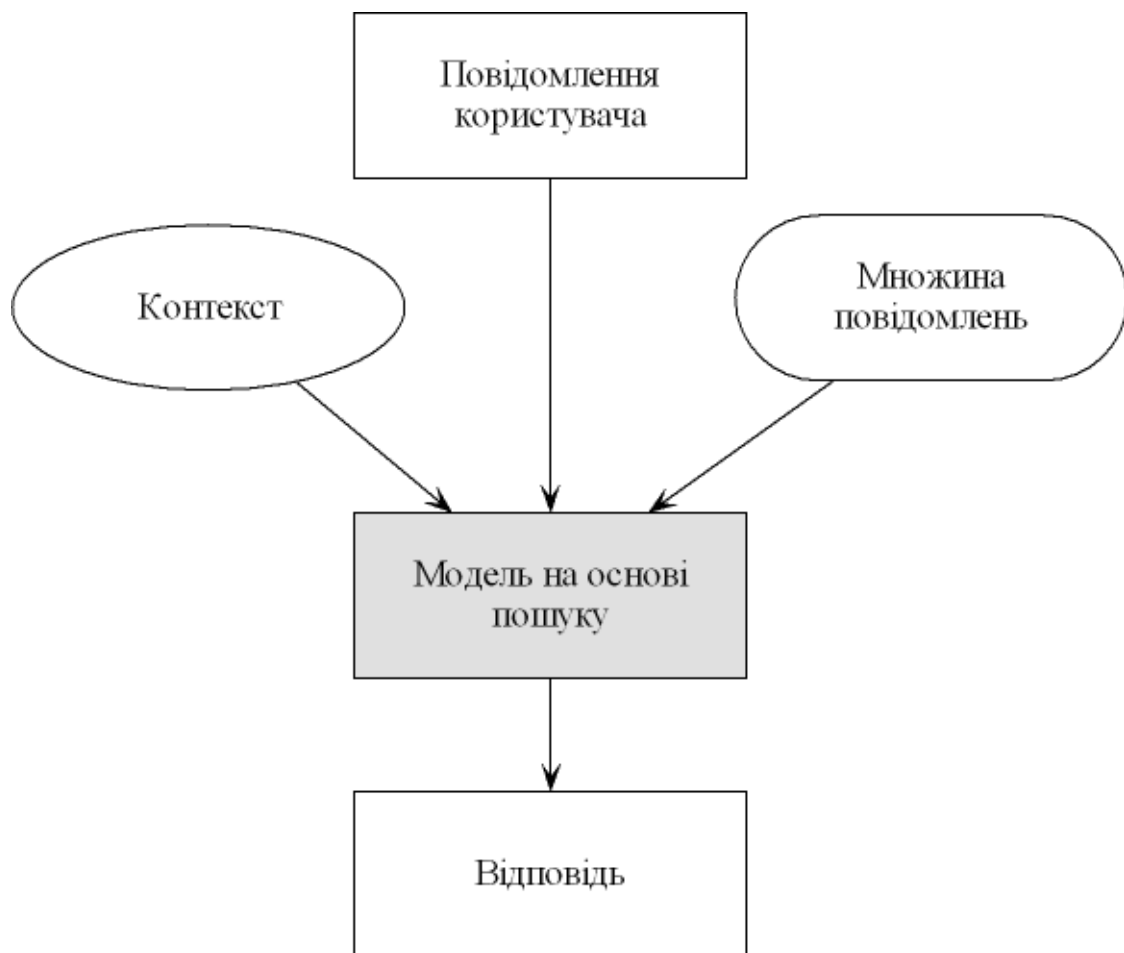


Рис. 7. Архітектура чат-боту на основі пошуку

Хоча евристика на основі зразків приносить досить непогані результати, проблема полягає в тому, що всі шаблони спілкування з користувачем потрібно прописувати вручну. Це досить кропітке завдання, особливо якщо чат-бот повинен розпізнавати сотні намірів користувачів, що можуть мати різні сценарії діалогів.

Класифікація намірів – завдання, побудоване цілком на технології машинного навчання, що дозволяє тренувати ботів. За допомогою набору тренувальних даних, що

містять тисячі прикладів комунікації, які найімовірніше знадобляться у спілкуванні з користувачем, чат-бот навчиться виділяти ознаки у даних і вдосконалюватись.

Якщо чат-бот зміг зрозуміти запит користувача, наступним кроком є генерація відповіді. Одним зі шляхів є генерація простої статичної відповіді. Іншим способом є отримання шаблону на основі наміру користувача та додавання до нього деяких змінних [6]. Наприклад, чат-бот для прогнозування погоди, що використовує API, щоб отримати інформацію про погоду за певною геолокацією, може мати кілька варіантів відповіді про одну й ту ж подію “сьогодні швидше за все дощитиме”, “сьогодні буде дощовий день” або “ймовірність дощу 80%, тому захопіть з собою парасольки”.

Стиль відповіді варіюється залежно від користувача. У цьому випадку бот може дослідити та вивчити ознаки попередніх діалогів, щоб створити персоналізовані відповіді для користувача.

Таким чином, на основі поданого вище матеріалу нами пропонується наступна структура алгоритму роботи автоматичного генератора відповідей з використанням корпусу повідомлень користувача та попередньо виконаної обробки текстів повідомлень:

1. Аналіз повідомлень користувача за допомогою методів обробки природної мови. Метою аналізу є створення словника користувача, а також бази знань, що містить відповіді користувача на різні запити. База у подальшому використовуватиметься як з метою оцінки питання користувача, так і для генерації відповіді.

2. Робота системи у діалговому режимі, тобто: питання – відповідь.

- 2.1. Після отримання запитання проводиться його аналіз для визначення теми запитання на основі бази знань.

- 2.2. Вибір множини можливих відповідей на основі бази знань та оцінка їх апріорної ймовірності.

- 2.3. Генерація остаточної відповіді користувачу на основі вибраної множини відповідей з використанням власних евристик.

Зважаючи на концептуальність поданої структури алгоритму, у кожному з його етапів можна виділити окремі підзадачі, що самі по собі можуть бути напрямками для подальшої роботи. Зокрема, нами визначено напрямки можливих подальших досліджень.

1. Задача попередньої обробки текстів досить добре розв’язана для англійської мови та споріднених з нею. У той же час окремі етапи попередньої обробки для української мови ставлять нові задачі у зв’язку з іншим способом словотворення та побудовою речень, що є менш формалізованою порівняно з англійською мовою.

2. Задача побудови бази знань на основі повідомлень користувача вимагає попередньої оцінки як структури програмних одиниць, так і структури синтаксичних та лексичних одиниць природної мови.

3. Вимагають додаткового дослідження евристики вибору можливих варіантів відповідей та генерації остаточної відповіді користувачу. Для цього можуть використовуватись як зокрема методи машинного навчання, так і загалом методи штучного інтелекту, такі як обробка баз знань, автоматична генерація суджень тощо.

Висновки

Отже, у ході проведеного дослідження розглянуто основні вимоги до вибору даних та методи попередньої обробки тексту, такі як: видалення пунктуації, тегів, стоп-слів, приведення тексту до нижнього регістру. Розглянуто методи, за допомогою яких

векторизується природна мова, зокрема: Bag of Words, TF/IDF. Досліджено методи класифікації за допомогою Naïve Bayes та Decision Trees, розглянуто окремі приклади коду, що дозволяють реалізувати вказані класифікатори й отримати точність розподілення попередньо розмічених текстів за групами. Подано опис поняття автоматичного відповідача та наведено дві найпоширеніші моделі архітектури таких відповідачів: генеративну модель та модель на основі пошуку. Побудовано концептуальний алгоритм роботи автоматичного відповідача з використанням методів машинного навчання для попередньої класифікації повідомлень користувача.

Список використаних джерел

1. Allen J.F. Analyzing intention in utterances / J.F. Allen, C.R. Perrault // Artificial Intelligence. – 15(3). 1980. Pp. 143-178.
2. Anaconda | The World's Most Popular Data Science Platform / [Електронний ресурс] – Режим доступу: <https://www.anaconda.com/>
3. Britz D. Deep learning for chatbots / [Електронний ресурс] – Режим доступу: <http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/>
4. Cassell J. Human conversation as a system framework: designing embodied conversational agents, Embodied conversational agents / J. Cassell, T. Bickmore, L. Campbell, H. Vilhjálmsón. MIT Press, Boston, 2000. – P. 29-63.
5. Facebook for developers / [Електронний ресурс] – Режим доступу: <https://developers.facebook.com/>
6. Galitsky B.A. Chatbot with a discourse structure-driven dialogue management / B.A. Galitsky, D. Ilvovsky // EACL Demo E17-3022. Valencia, Spain. – 2017 / [Електронний ресурс] – Режим доступу: <https://developers.facebook.com/>
7. Jurafsky D. Speech and Language Processing / D. Jurafsky, J. H. Martin. – Stanford: Pearson Prentice Hall, 3rd edition, 2019. – 605 p.
8. Kurdi M.Z. Natural Language Processing and Computational Linguistics: speech, morphology, and syntax / M. Z. Kurdi. – London: Wiley ISTE, 2016. – 296 p.
9. Lee C. Example-based dialog modeling for practical multi-domain dialog system / C. Lee, S. Jung, S. Kim, G.G. Lee // Speech Comm 51:466, 2009.
10. Natural Language Toolkit / [Електронний ресурс] – Режим доступу: <https://www.nltk.org/>
11. Powers D. Machine Learning of Natural Language / D. Powers, C. Turk. – New York: Springer Verlag, 1989. – 358 p.
12. Project Jupyter | Home / [Електронний ресурс] – Режим доступу: <https://jupyter.org/>
13. Steven B. Natural Language Processing with Python / B. Steven, E. Klein, E. Loper. – Sebastopol, CA: O'Reilly Media, 2009. – 504 p.
14. The 20 newsgroups text dataset / [Електронний ресурс] – Режим доступу: <https://scikit-learn.org/>
15. The Size and Quality of a Data Set / [Електронний ресурс] – Режим доступу: <https://developers.google.com/machine-learning/data-prep/construct/collect/data-size-quality>

Bibliography:

1. Allen, J.F., Perrault, C.R. (1980). Analyzing intention in utterances. Artificial Intelligence, 15(3):143-178.
2. Anaconda | The World's Most Popular Data Science Platform. Retrieved from: <https://www.anaconda.com/>
3. Britz D. Deep learning for chatbots. Retrieved from: <http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/>
4. Cassell, J., Bickmore, T., Campbell, L., Vilhjálmsón, H. (2000). Human conversation as a system framework: designing embodied conversational agents, Embodied conversational agents. MIT Press, Boston, pp 29–63.
5. Facebook for developers. Retrieved from: <https://developers.facebook.com/>
6. Galitsky, B.A., Ilvovsky, D. (2017). Chatbot with a discourse structure-driven dialogue management. EACL Demo E17-3022. Valencia, Spain. Retrieved from: <https://developers.facebook.com/>
7. Jurafsky, D., Martin, J.H. (2019). Speech and Language Processing. Stanford, Pearson Prentice Hall, 3rd edition.
8. Kurdi, M.Z. (2016). Natural Language Processing and Computational Linguistics: speech, morphology, and syntax. London: Wiley ISTE.
9. Lee, C., Jung, S., Kim, S., Lee, G.G. (2009). Example-based dialog modeling for practical multi-domain dialog system. Speech Comm 51:466.
10. Natural Language Toolkit. Retrieved from: <https://www.nltk.org/>
11. Powers, D., Turk, C. (1989). Machine Learning of Natural Language. New York: Springer Verlag.

12. Project Jupyter | Home. Retrieved from: <https://jupyter.org/>
13. Steven, B., Klein, E., Loper, E. Natural Language Processing with Python. Sebastopol, CA: O'Reilly Media.
14. The 20 newsgroups text dataset. Retrieved from: <https://scikit-learn.org/>
15. The Size and Quality of a Data Set. Retrieved from: <https://developers.google.com/machine-learning/data-prep/construct/collect/data-size-quality>

KOVALENKO Olena,

Student, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

SERDIUK Oleksandr,

Candidate of Economic Sciences, Department of Informatics and Applied Mathematics, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

USING THE CLASSICAL METHODS OF MACHINE LEARNING FOR TEXT CLASSIFICATION IN THE AUTOMATIC ANSWERS GENERATION PROGRAMS

Summary. Introduction. *Machine learning methods are now widely used for natural language processing to recognize, understand, interpret, and generate natural language by computer algorithms. In the modern world of information technology, the range of tasks that can use the results of human language interpretation is constantly increasing, including automatic texts classification, news rubrication, recognition of user questions in automatic houses and generation of answers. The widespread use of messengers to transmit messages between users, along with the increasing number of users, has opened up another area of application for text classification tasks: the development of automatic assistants for generating messages. To solve this task, it is necessary not only to be able to process natural language, but also to generate adequate answers, or, in general, sentences for dialogue. The article is aimed at reviewing the methods of text classification for their using in generating automatic answers.*

The purpose of the article is to study methods and algorithms and demonstrate program fragments of classical machine learning methods for text classification and application of classification results in automatic answer generation programs.

To achieve the goal, the following issues are considered in the article: review of data selection conditions; review of word processing methods; review of text vectorization methods; review of some text classification algorithms; introducing the concept of automatic answers generation; developing the structure of the program for automatic answers generation using the corpus of processed texts.

Based on the study, the following conclusions were obtained. The problem of pre-processing texts is quite well solved for English and related languages. At the same time, some stages of pre-processing for the Ukrainian language pose new challenges due to a different way of word formation and sentence construction, which is less formalized compared to English. The task of building a knowledge base based on user messages requires a preliminary assessment of both the structure of program units and the structure of syntactic and lexical units of natural language. Required also additional research on the heuristics of the choice of possible answers and generate a final answer to the user. For this purpose, both machine learning methods and artificial intelligence methods in general, such as processing knowledge bases, automatic generation of judgments, etc. can be used.

Keywords: *machine learning, text classification, chat-bots, programming, Python.*

*Одержано редакцією 09.01.2020 р.
Прийнято до публікації 30.03.2020 р.*

ЗМІСТ

СЕКЦІЯ «ПРИКЛАДНА МАТЕМАТИКА»

П. О. Стеблянко, К. Е. Дьомічев, О. Д. Петров

НЕЛІНІЙНА ФЕНОМЕНОЛОГІЧНА МОДЕЛЬ ПОВЕДІНКИ
ФУНКЦІОНАЛЬНО НЕОДНОРІДНИХ МАТЕРІАЛІВ 4

P. V. Lukianov, I. P. Mukha

BOOLEAN FUNCTION “EXCLUSIVE OR” FOR THE N VARIABLES 12

Б. П. Головня

РАСЧЕТ ЛАМИНАРИЗАЦИИ ТЕЧЕНИЯ ПРИ СОВПАДАЮЩЕЙ
ТУРБУЛЕНТНОЙ СМЕШАННОЙ КОНВЕКЦИИ НА ВЕРТИКАЛЬНОЙ
ПОВЕРХНОСТИ 22

М. Г. Моргун

ДОСЛІДЖЕННЯ ДИНАМІКИ ГРОМАДСЬКОЇ ДУМКИ У СУЧАСНОМУ
СУСПІЛЬСТВІ НА ОСНОВІ МУЛЬТИАГЕНТНОЇ МОДЕЛІ 32

І. М. Порубльов

ДЕЯКІ АЛГОРИТМИ ПОШУКУ ЧИСЕЛ З МАКСИМАЛЬНОЮ
КІЛЬКІСТЮ ДІЛЬНИКІВ 44

СЕКЦІЯ «ІНФОРМАТИКА»

О. В. Пшенишний, Н. О. Красношлик

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ПОПУЛЯЦІЙНИХ АЛГОРИТМІВ ДЛЯ
РОЗВ’ЯЗУВАННЯ ЗАДАЧ ГЛОБАЛЬНОЇ ОПТИМІЗАЦІЇ 61

Я. С. Моторна, Н. О. Красношлик, О. В. Піскун

РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ АЛГОРИТМУ RANDOM FOREST ДЛЯ
РОЗВ’ЯЗУВАННЯ ЗАДАЧ КЛАСИФІКАЦІЇ 69

Н. А. Тарасенкова, З. О. Сердюк

ОСОБЛИВОСТІ ВИКЛАДАННЯ КУРСУ МАТЕМАТИЧНОГО АНАЛІЗУ

ДЛЯ ФАХІВЦІВ З АНАЛІЗУ ДАНИХ

77

О. С. Коваленко, О. А. Сердюк

ВИКОРИСТАННЯ КЛАСИЧНИХ МЕТОДІВ МАШИННОГО НАВЧАННЯ
ДЛЯ КЛАСИФІКАЦІЇ ТЕКСТІВ У ПРОГРАМАХ ГЕНЕРАЦІЇ
АВТОМАТИЧНИХ ВІДПОВІДЕЙ

86

CONTENTS

APPLIED MATHEMATICS SECTION

P. O. Steblyanko, K. E. Domichev, A. D. Petrov	
NONLINEAR PHENOMENOLOGICAL MODEL OF BEHAVIOR OF FUNCTIONALLY HETEROGENEOUS MATERIALS	4
P. V. Lukianov, I. P. Mukha	
BOOLEAN FUNCTION “EXCLUSIVE OR” FOR THE N VARIABLES	12
B. P. Golovnya	
CALCULATION OF FLOW LAMINARIZATION AT ASSISTING TURBULENT MIXED CONVECTION ON A VERTICAL SURFACE	22
M. H. Morhun	
ANALYZING DYNAMICS OF PUBLIC OPINION IN MODERN SOCIETY USING MULTI AGENT MODEL	32
I. M. Porublyov	
SOME ALGORITHMS FOR SEARCHING NUMBERS WITH MAXIMAL QUANTITY OF DIVISORS	44

INFORMATICS SECTION

O. V. Pshenishniy, N. O. Krasnoshlyk	
INVESTIGATION OF THE EFFICIENCY OF POPULATION ALGORITHMS FOR SOLVING GLOBAL OPTIMIZATION PROBLEMS	61
Y. S. Motorna, N. O. Krasnoshlyk, O. V. Piskun	
IMPLEMENTATION AND RESEARCH OF THE RANDOM FOREST ALGORITHM TO SOLVE CLASSIFICATION PROBLEMS	69
N. A. Tarasenkova, Z. O. Serdiuk	
PECULIARITIES OF TEACHING THE COURSE OF MATHEMATICAL ANALYSIS FOR SPECIALISTS IN DATA MINING	77

O. S. Kovalenko, O. A. Serdiuk

USING THE CLASSICAL METHODS OF MACHINE LEARNING FOR TEXT
CLASSIFICATION IN THE AUTOMATIC ANSWERS GENERATION
PROGRAMS

86